read_the_docs_test Documentation

Release 1

Ad115

Mar 20, 2017

Contents

Contents:

CHAPTER 1

About the ICGC-data-parser

Scripts to automate parsing of data from the International Cancer Genome Consortium data releases, in particular, the simple somatic mutation aggregates.

Download and installation

Scripts

To download the files in this repo, you can use the green Download button on the GitHub repository or enter the following in a Unix terminal

```
git clone https://github.com/Ad115/ICGC-data-parser.git
```

Data download

The base data for the scripts is the ICGC's aggregated of the simple somatic mutation data. Which can be downloded using

```
wget https://dcc.icgc.org/api/v1/download?fn=/current/Summary/simple_somatic_mutation.
↪aggregated.vcf.gz
```

To know more about this file, please read *About the ICGC's simple somatic mutations file*

Requisites installation

The main scripts are written in Perl and use the Ensembl Perl API for which detailed instructions of installation are in *How to install the Ensembl Perl API*.

Plotting and analysis of results are implemented in Wolfram Mathematica notebooks. *TODO:* Change it to a free platform or add alternate scripts in a free platform.

The scripts in the *SunGridEngine-scripts* folder, as it's name indicates, are designed for running in a Sun Grid Engine cluster and thus will break if tried to run on a typical personal computer. Despite this, they are mostly a convenience and their functionality is still in other alternative scriptsfor each pipeline. See the *Usage* section for more details.

Usage

The scripts are divided in workflows or pipelines. The pipelines currently implemented are the following:

- **Mutation recurrence count**: Automates the process of extracting reccurrence of mutations across patients, it answers the question: *How many mutations appear in multiple patients?* or, specifying further, *how many mutations are repeated in 'n' different patients in a given cancer project and a given gene?* This workflow is further documented in *The mutation recurrence workflow*.
- **Locating mutations in the genome**: Automates the process of searching where does each mutation fall relative to a gene. In particular, it answers the questions: *How many (and which) mutations fall in INTRONIC, EXONIC or INTERGENIC regions?* and *if a mutation falls in an exon, which base of the codon does it affects? TODO:* This pipeline is further documented in The mutation locating workflow.
- **Distribution of the mutations in the genes**: Automation of the extraction of the distribution of mutations in the genes. It answers the question of *how many genes contain 'x' number of mutations in a given gene or project? TODO:* This is further documented in *The mutations distribution workflow*.
- **Simple Ensembl Perl API convenience scripts**: These are small convenience scripts constructed with the intention to test the Ensembl Perl API but serve as integral programs on their own. These are found in the ensembl_API folder . *TODO:* These are further documented in The Ensembl Perl API scripts.

TO DO

- [ ] Cleanup every workflow and document it.

CHAPTER 2

How to install the Ensembl Perl API

Most of the scripts used are written in Perl using the Ensembl Perl API to access easily to the data in their databases. So, those libraries must be installed, and, in turn, they depend on BioPerl, Expat and mySQL! n(O.O)n

What follows are the instructions to install these dependencies. Note this instructions are for Ubuntu-based Linux only, to use it in another OS you might better read the instructions from the Ensembl Perl API webpage.

Installing mySQL

Just typing `sudo apt-get install libmysqlclient-dev` in a terminal should be enough.

Installing Expat

(*Reference*: expat)

This is required by BioPerl (I'm not shure whether it's essential but I installed anyway), to install it, you can download the source from source-forge or, if you prefer the console terminal:

```
wget http://downloads.sourceforge.net/project/expat/expat/2.1.1/expat-2.1.1.tar.bz2
```

Then decompress it with:

```
sudo tar -jxvf expat-2.1.1.tar.bz2
```

And this will install it:

```
sudo ./configure && make && sudo make install
```

Installing BioPerl

(Reference: BioPerl)

- In the console line type: `cpan`, or: `perl -MCPAN -e shell`, and if it's the first time you excecute that, cpan will enter a configuration process. Preferably say yes to all :P .
- Once there type:

```
cpan>  install Module::Build
cpan>  o conf prefer_installer MB
cpan>  o conf commit
cpan>  install CJFIELDS/BioPerl-1.6.924.tar.gz
```

It most surely will take a while executing lots of tests. You'll only have to be patient and say yes to everything it prompts. When it's done, type Ctrl-D to exit cpan.

That's it! You now have BioPerl installed on your machine! :D

Installing Ensembl Perl API

(Reference: Ensembl API Installation)

Now, what we are all here for:

- First, choose a folder in which to install the library. In the following, I'll assume that folder is ~/ensembl-api. In that folder you will install and decompress the files:

```
       cd ~/ensembl-api
       wget ftp://ftp.ensembl.org/pub/ensembl-api.tar.gz
       sudo tar -zxvf ensembl-api.tar.gz

- Now, you need to tell Perl where to find those files, so you must type the
↪following:
```

```
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl-compara/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl-variation/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl-funcgen/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl-io/modules
PERL5LIB=${PERL5LIB}:${HOME}/ensembl-api/ensembl-tools/modules
export PERL5LIB
```

And to avoid having to type that every time you start a terminal window, add those lines at the end of the file `.bashrc` in your home.

Aaaaanddd... We're done! :D

CHAPTER 3

About the ICGC's simple somatic mutations file

This is about the infamous `simple_somatic_mutations.aggregated.vcf` file presented in each ICGC Data Release which contain an aggregated of the information of all simple somatic mutations found accross all patients in all cancer projects is found.

The top 500 lines of this file (ICGC data release 22) can be found at the ssm_head500.vcf file in the `sample-data` folder of this repo.

Download

This file can be downloaded from the ICGC site data releases site or using:

```
wget https://dcc.icgc.org/api/v1/download?fn=/current/Summary/simple_somatic_mutation.
↪aggregated.vcf.gz
```

To resume an interrupted download use the `-c` switch on the previous command.

Then, the file can be extracted with the `gunzip` command.

Structure

The `simple_somatic_mutations.aggregated.vcf` file, from now on referred as the SSM file, is a VCF file as specified in HTS format specifications, and in particular, the SSM files are created using the SnpEff annotation tool.

In general, the format is similar to a TSV file in which the comments are marked with `##` and the headers line with `#` and there is one line per simple-somatic-mutation found.

Fields and Header lines

Next are the 13 heading lines from a SSM file (data release 22):

```
##fileformat=VCFv4.1
##INFO=<ID=CONSEQUENCE,Number=.,Type=String,Description="Mutation consequence␣
↪predictions annotated by SnpEff (subfields: gene_symbol|gene_affected|gene_
↪strand|transcript_name|transcript_affected|protein_affected|consequence_type|cds_
↪mutation|aa_mutation)">
##INFO=<ID=OCCURRENCE,Number=.,Type=String,Description="Mutation occurrence counts␣
↪broken down by project (subfields: project_code|affected_donors|tested_
↪donors|frequency)">
##INFO=<ID=affected_donors,Number=1,Type=Integer,Description="Number of donors with␣
↪the current mutation">
##INFO=<ID=mutation,Number=1,Type=String,Description="Somatic mutation definition">
##INFO=<ID=project_count,Number=1,Type=Integer,Description="Number of projects with␣
↪the current mutation">
##INFO=<ID=tested_donors,Number=1,Type=Integer,Description="Total number of donors␣
↪with SSM data available">
##comment=ICGC open access Simple Somatic Mutations (SSM) data dump in VCF format
##fileDate=2016-08-16T16:32:17.882-04:00
##geneModel=ENSEMBL75
##reference=GRCh37
##source=ICGC22-12
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO
```

This is what we can see in those lines:

- *fileformat*: A line specifying the VCF version (4.1).
- *INFO*: Six lines breaking down each part of the INFO field.
- *comment*:A general description of the file (*ICGC open access Simple Somatic Mutations (SSM) data dump in VCF format*).
- *fileDate*: The creation date of the file (August 2016).
- *geneModel*: A specification of the gene model (ENSEMBL75). This is the nameset used for the annotations in the INFO field. In particular the identifiers and names in the CONSEQUENCE subfield. **WATCH OUT!** Some genes, transcripts and identifiers annotated may have changed for the current release and so may not be found in a direct query. At the moment of writing, the most recent build is Ensembl 87.
- *reference*: The genome assembly version used for the positions in the reference genome (GRCh37). **WATCH OUT THIS!** The positions may have dramatic changes from one assembly to another. At the moment, the most recent version of the human genome reference is the GRCh38 assembly.
- *source*: The data source (ICGC Data release 22)
- **The column headers**: See section *The column headers*.

The column headers

The data is split in these fields:

- **CHROM**: The chromosome the mutation is in.
- **POS**: The position in the chromosome of the start of the mutation. This is in the reference assembly specified in the initial comments.
- **ID**: The current mutation's ICGC identifier.
- **REF**: The sequence found in the reference.
- **ALT**: The sequence found in the mutated sample, so that the mutation definition is REF>ALT.

- **QUAL**: The quality of the read. As a general rule, a quality <10 is unreliable.
- **FILTER**.
- **INFO**: Additional annotation on the mutation consequences, and occurrence along patients and projects. It is further commented on *The INFO Field*

The INFO field

This field annotates predicted consequences, and seen occurrences of the current mutation. The consequences are as seen by the SnpEff package.

There may be multiple consecuences and occurrences of the same mutation, and those need to be clearly specified. Thus the complex form of this field.

In the file, *the parts are separated with a semicolon (* `;` *), and each part may have itself subfields, which are separated with pipes (* `|` *). Alternative parts* (e.g. different consequences for the mutation or occurrences in different cancer projects) *are separated by a comma (* `,` *).*

- **CONSEQUENCE:** Mutation consequence predictions annotated by SnpEff. Which has itself the next subfields:

1. *gene_symbol,*
2. *gene_affected,*
3. *gene_strand,*
4. *transcript_name,*
5. *transcript_affected,*
6. *protein_affected,*
7. *consequence_type,*
8. *cds_mutation,*
9. *aa_mutation*

- **OCCURRENCE**: Mutation occurrence counts broken down by project. Which has itself the next subfields:
    1. *project_code,*
    2. *affected_donors,*
    3. *tested_donors,*
    4. *frequency*
- **affected_donors**: Total number of donors with the current mutation.
- **mutation**: Somatic mutation definition, in the form BEFORE>AFTER.
- **project_count**: Number of projects with the current mutation.
- **tested_donors**: Total number of donors with SSM data available.

Interpreting a sample mutation

Now we come to try to read an example mutation from the data.

The mutation

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO
1       100000022       MU39532371      C       T       .       .       ␣
↪CONSEQUENCE=||||||intergenic_region||,RP11-413P11.1|ENSG00000224445|1|RP11-413P11.1-
↪001|ENST00000438829||upstream_gene_variant||;OCCURRENCE=SKCA-BR|1|70|0.01429;
↪affected_donors=1;mutation=C>T;project_count=1;tested_donors=10638
```

The interpretation

We can see the data for the mutation **MU39532371**, which is in the chromosome number *1*, at the position *100000022*, and is defined as *C>T*, with no quality or filtering information available. We can also see in the INFO that this mutation has two consequences: one as a mutation ocurring in an intergenic region, and one as a mutation that affects the *ENSG00000224445* gene and it's *ENST00000438829* transcript provoking an *upstream_gene_variant*. Besides, it was found in a sample from the Great Britain's skin cancer ICGC project (*SKCA-BR*) with *1* patient affected out of the *70* in the project and of the *10638* accross all projects.

CHAPTER 4

The mutation recurrence workflow

Documentation for the mutation recurrence workflow. Including description, basic structure logic, description of the implementations and results.

Description

In general, the mutation recurrence workflow automates the process of extracting reccurrence of mutations across patients, it answers the question: *How many mutations appear in multiple patients?* or, specifying further, *how many mutations are repeated in ''n'' different patients in a given cancer project and a given gene?*

Basic structure logic

This workflow explodes the fact that the ICGC data already provides, for each mutation, the number of affected patients per project and accross all projects. So, the steps involved are the following:

1. **Fetching of the recurrence data (no. of affected donors) for a cancer project and/or gene of interest** for each mutation from the raw mutation data (gene "all genes" and/or project "all projects" are allowed). This narrows the scope to only those mutations that are present in the given gene and the given project and from those only get the data that may be useful.
2. **Counting of the recurrence data**. Specifically, counting how many mutations recurr in `n` patients (the workflow question) for each `n` in `1, 2, 3, ...`. In this process the mutation identities are lost, and we are only left of the distribution of mutation recurrence across patients.
3. **Display (plotting) and analysis of the results**. This step involves plotting the resulting distribution (table) and doing analysis and interpretation of the results.

Results

The following specifies the outputs seen at each step. This output serves as input to the next step.

1. **Fetching of the recurrence data (no. of affected donors) for a cancer project and/or gene of interest**
2. **Counting of the recurrence data**. Specifically, counting how many mutations recurr in *n* patients (the workflow question) for each `n` in `1, 2, 3, ...`. In this process the mutation identities are lost, and we are only left of the distribution of mutation recurrence across patients.
3. **Display (plotting) and analysis of the results**. This step involves plotting the resulting distribution (table) and doing analysis and interpretation of the results.

Appendix I: Implementation(s)

WARNING: Subject to change in the near future.

Step 1. Fetching of the recurrence data (no. of affected donors) for a cancer project and/or gene of interest

This is implemented in the script **filter_gene_project.pl** which fetches important data from the raw data.

The script retrieves the next fields:

- MUTATION_ID
- POSITION
- MUTATION
- TOTAL_AFFECTED_DONORS
- PROJ_AFFECTED_DONORS
- CONSEQUENCES

Usage

The user provides the gene to search for, the project the input file (ICGC's SSM file) and the desired output file.

Example output

For the command `filter_gene_project.pl -g TP53 -p BRCA-EU -i $ICGC_DATA` the first 7 lines of output are:

This script is also important as the first step of other workflows, to learn more about it, read ***TODO:*** The filtering script

Step 2 Counting of the recurrence data.

TODO

Step 3 Display (plotting) and analysis of the results.

TODO: *distribution-plots.nb*

Complete workflow up to counting.

The complete workflow up to the plotting is implemented in serial form in Perl/Bash language by the script `get_genes_info.pl` ***TODO***

CHAPTER 5

On the frequency of simple somatic mutations in cancer

Abstract

Genetic instability is a landmark of cancer, but simple patterns may be hidden behind it's complex evolution laws. This paper presents the results of analysing the distribution of mutations accross genes of the ICGC simple somatic mutations data.

Background

Cancer is a set of diseases characterized by genetic instability, frequently found as simple-somatic mutations genome wide. The study of this mutations may lead to new insights related to cancer origins, onset and propagation. The International Cancer Genome Consortium (ICGC) is an international scientific group, founded in 2008, whose goal is to obtain a comprehensive description of genomic, transcriptomic and epigenomic changes accross different tumor types of clinical and societal importance across the globe. For this goal, it has coordinated the analysis of thousands of samples of more than 10,000 patients at genetic, epigenomic and transcriptomic level. Although some of it's data has restricted access, other is publicly available, such as simple somatic mutation data, copy-number mutation data, patients treatment data, etc. The ICGC releases it's public data as Data Releases at an approximate rate of 2 per semester. The data is organized in Cancer Projects, which separate data from different cancer types and different countries. At the date, the most recent version is Data Release 23 (Dec 2016). Although the ICGC Data Releases are complete and useful data for many analysis purposes, it may be obtuse for simple information retrieval as is mostly raw data from the samples. The ICGC data parser is a suite of programs whose purpose is to facilitate the retrieval of simple, purposeful data from the ICGC Data Release archives, particularly from the simple somatic mutation (ssm) data. One of the questions that surges when looking at the ssm data is how many mutations are present in an average gene, and more generally, Are the mutations randomly distributed? and What is the distribution of mutations like accross genes?

Approach

We analyze the simple somatic mutations(ssm) data from the ICGC Data Releases to find the distribution of mutations accross genes. To keep things simple, we analyse only ssm data from the BRCA-EU project, which analyzes ductal breast cancer samples from the European Union.

Methods

We made a Perl script as part of the ICGC-data-parser suite to find, in the BRCA-EU project, how many mutations were present in each gene. This is documented as the Mutation Distribution Workflow.

Results

The resulting distribution is in *Figure 1*.

We can see that most genes fall in the range of 10-50 mutations in a very sharp peaked distribution leaned towards zero. To find a probability distribution to interpret the data, several distributions where tested with the aid of the Mathematica software. With the best fit showing a mixture of the Negative Binomial distribution (74%, $r = 4$, $p = 0.185$) and the Geometric Distribution (25%, $p = 0.0053$), fitted with a p value of 7.72×10^{-16}. This is shown in *Figure 2*

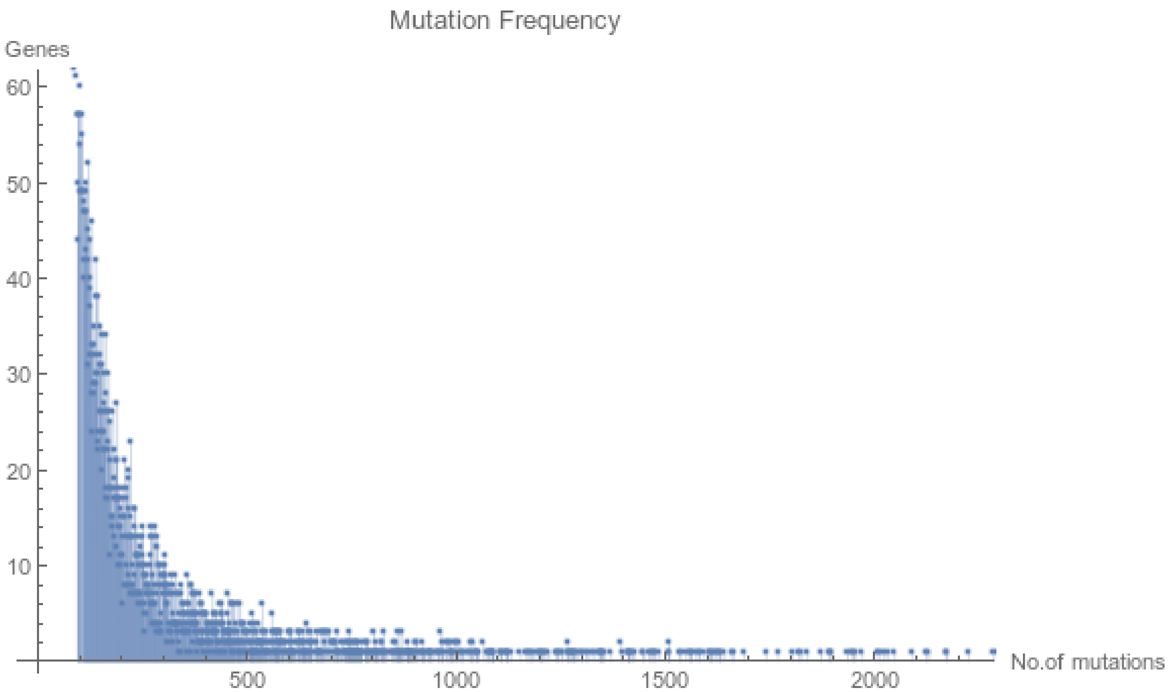
Besides that, we plotted the positions of the most representative genes in breast cancer according to *[Yu2016]*. This is shown in *Figure 3*.

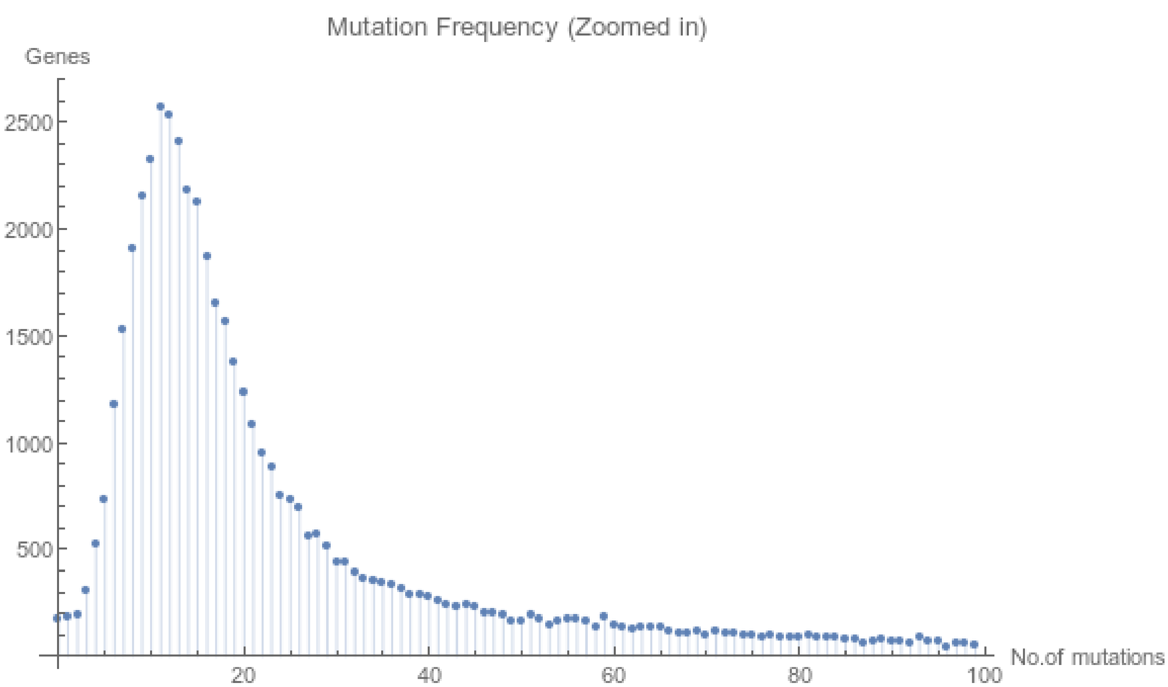
The plot shows the genes are widely distributed along the tail of the distribution, and so, they themselves may suffice to characterize the distribution.

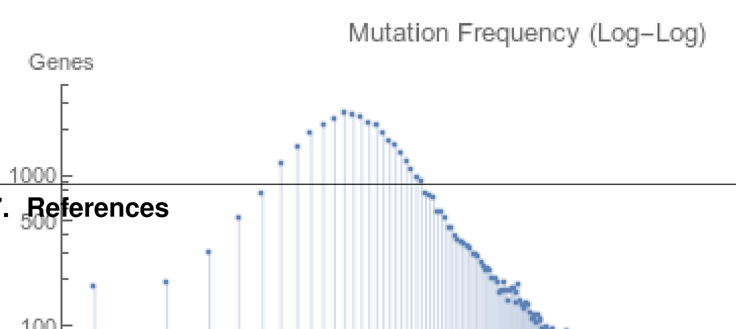
Conclusions

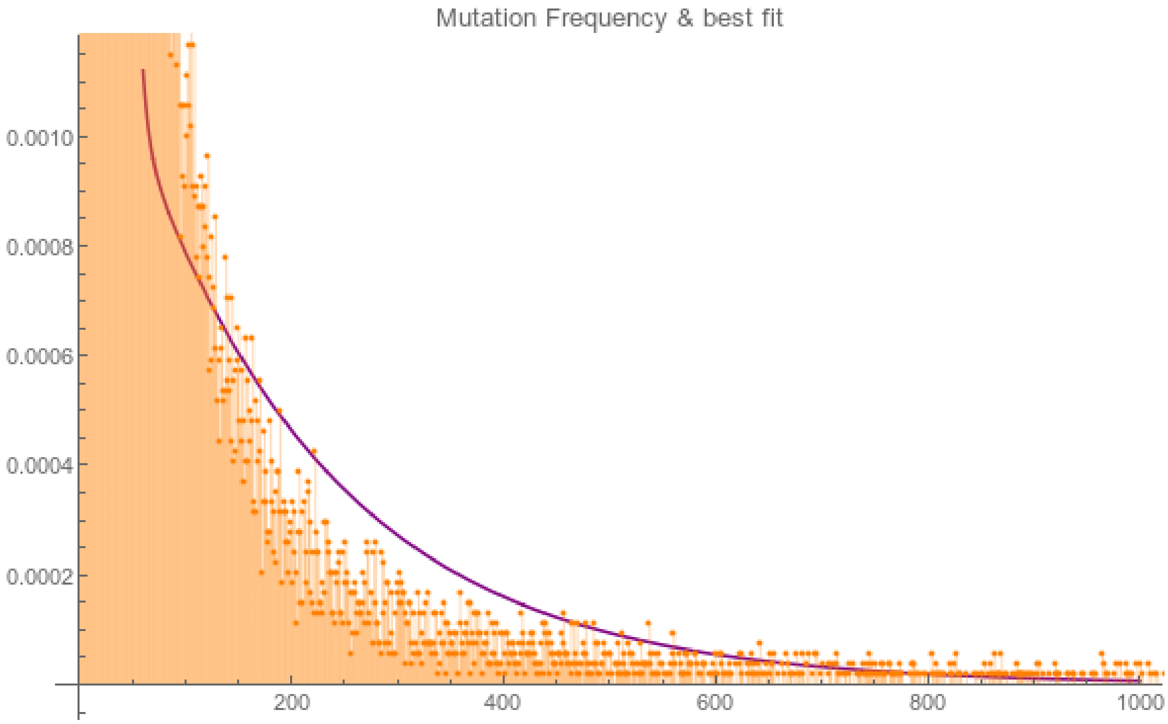
The data shown gives a snapshot of the mutations that may be found in a tumour, and show that the mutation probabilities follow simple laws as shown by the appeareance of the Negative Binomial and the Geometric distribution.

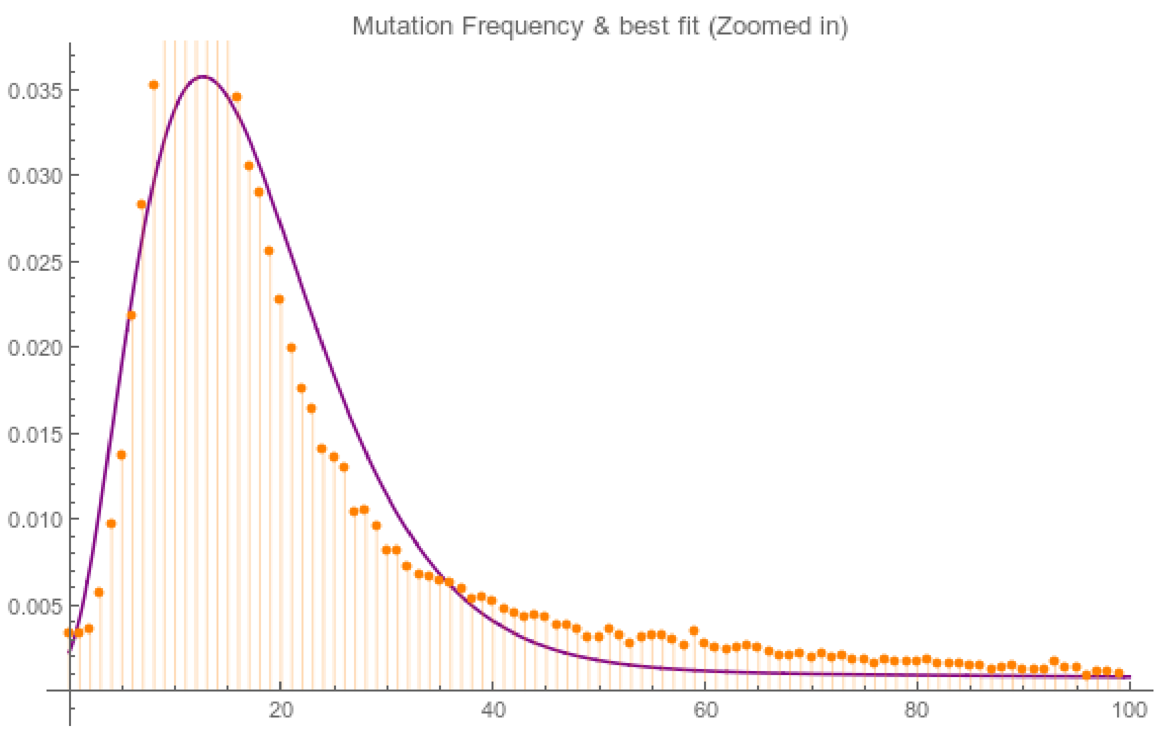
References

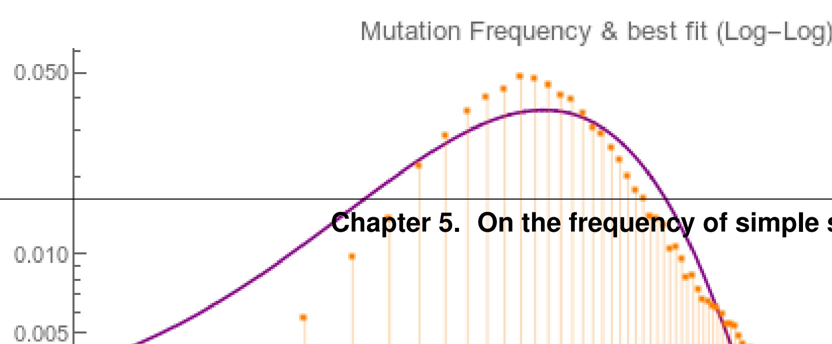

Mutation Frequency
Genes
60
50
40
30
20
10
500
1000
1500
2000
No.of mutations


Mutation Frequency (Zoomed in)
Genes
2500
2000
1500
1000
500
20
40
60
80
100
No.of mutations


Mutation Frequency (Log–Log)
Genes
1000
500
100


Mutation Frequency & best fit
0.0010
0.0008
0.0006
0.0004
0.0002
200
400
600
800
1000


Mutation Frequency & best fit (Zoomed in)
0.035
0.030
0.025
0.020
0.015
0.010
0.005
20
40
60
80
100


Mutation Frequency & best fit (Log–Log)
0.050
0.010
0.005

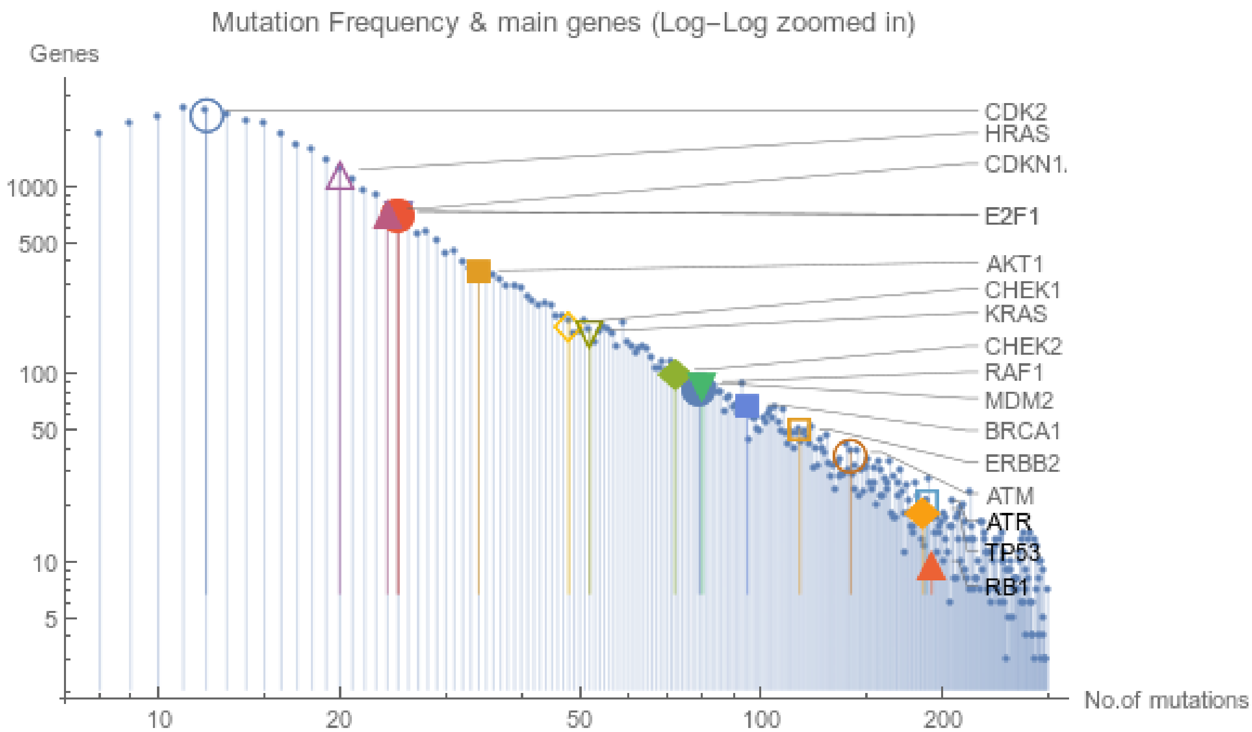


Fig. 5.3: **Figure 3:** *Positions in the distribution of the most representative genes in breast cancer*

Bibliography

[Yu2016] Chong Yu, Jin Wang. A Physical Mechanism and Global Quantification of Breast Cancer. PLOS ONE 11, e0157422 (2016).