Analysis and Improvements to Existing Clonality Tools

# Abstract

This is a small extracurricular research project executed in 180 hours throughout semester 2, 2017. Due to time constraints, the background research is not extensive but this report is written for an expert audience.

During the analysis of 5 case studies on intestinal metaplasia in the stomach, it was discovered that existing clonality tools do not provide the correct phylogenetic tree in particular circumstances. Thus, the phylogenetic trees had to be solved in a fairly manual process.

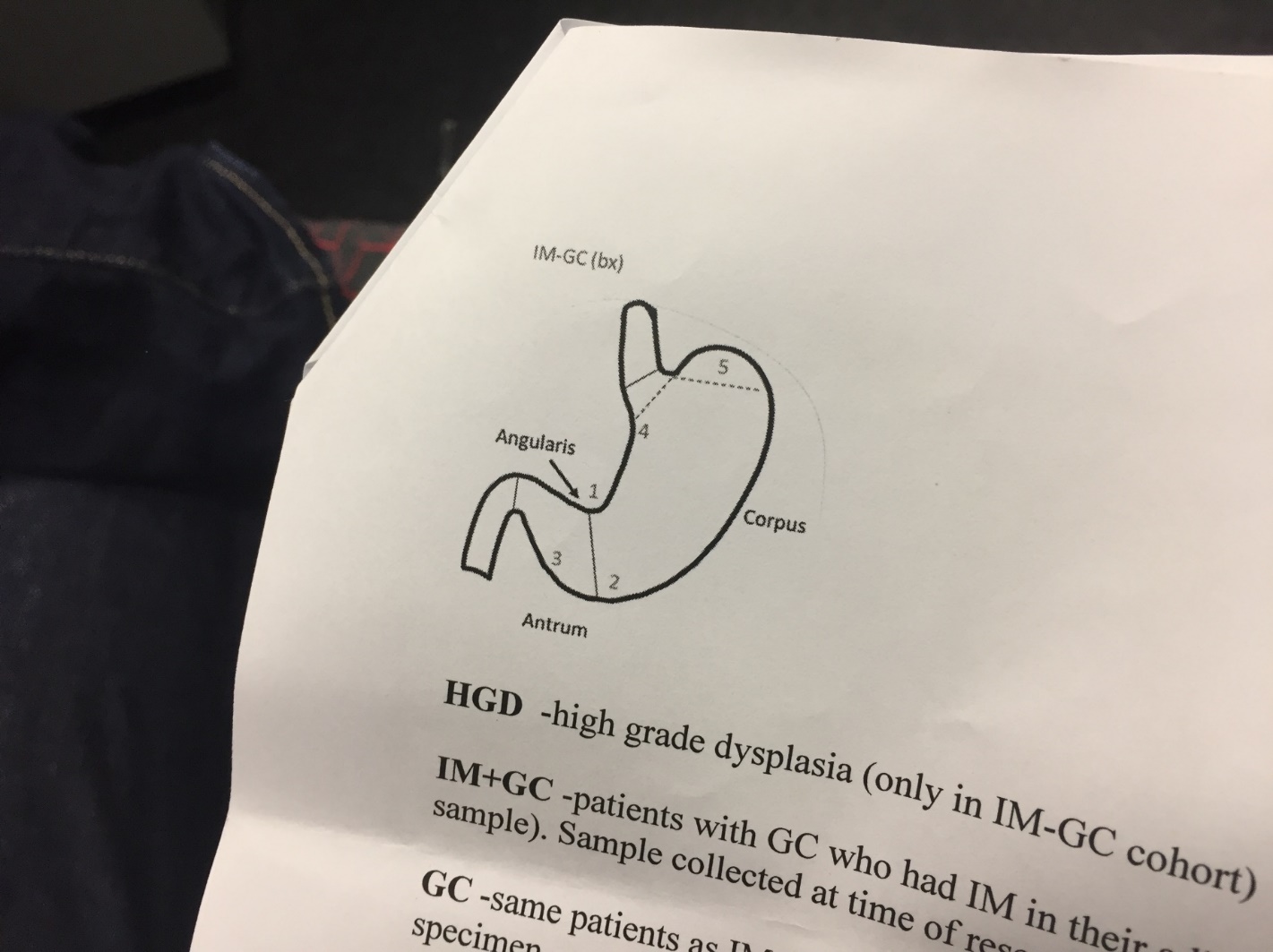
This project attempts to solve this existing problem in two parts. Firstly, the study of existing clonality tools used and the determination of their shortcomings. In particular, in the context of the intestinal metaplasia case studies in which the solving of the phylogenetic trees was done by hand and then compared against superFreq results. Secondly, the creating of a rough prototype code that attempts to quantify and solve these shortcomings and thus automate the current manual process.

# Background and Motivation

The determination of a phylogenetic tree of mutations in an individual is a useful tool in researching the triggers of cancers and potential future treatments. In order to achieve this, clonalities are required to be determined for the mutations (SNVs and CNVs) and then clustered into specific clones and the relationship between the clones. This project analyses and improves upon the clustering section in the determination of a phylogenetic tree.

## Case Studies

There are 5 case studies analysing the stomach of patients with intestinal metaplasia which is a premalignant condition where the stomach cells resemble intestinal cells. This then progresses to high grade dysplasia and in some patients’ eventually gastric cancer resulting in stomach resection. Samples were taken at multiple locations of the stomach as seen in **figure 1** and over multiple time points. These samples were then histologically categorised into intestinal metaplasia, high grade dysplasia and gastric cancer and then exome sequenced as discussed in implementation section.



**Figure 1:** Diagram of Stomach is sites samples were taken.

## Existing Programs

### HD clone-Whole Genome Sequencing

Determines subclone properties through treating them as hidden variables in a hidden markov model using CNV, BAF and SNV data. Where the hidden states are:

CNA- total number of chromosome copies in each subclone with a set cmax=7 typically

BAF- genotypes of minor allele at originally heterozygous sites

SNV- genotypes of the somatic mutations at each locus

With n clones, the number of states is (cmax+1)n. Thus increasing cmax or n greatly increases the complexity of the model.

### superFreq-Whole Exom Sequencing

Allele specific distinguishing AAB from ABB between samples.

Solves CNAs and clonalities then determines clonalities of somatic SNVS through running through these three scenarios:

SNV subclone of CNA -> upper limit on clonality of SNV

CNA subclone of SNV-> lower limit on clonality of SNV

CNA and DNV disjoint-> upper limit on clonality of SNV as sum of SNV and CNA cannot exceed 1

If only one solution is possible, it is chosen. Otherwise SNV errors are increased to cover all possibilities.

Mutations (SNVs and CNAs) with similar clonalities in all samples are then clustered into clones. This is done in a pairwise manner until no two sets of mutations are sufficiently similar. The clone is then assigned a clonality and uncertainty based on weighted mean of clustered mutations. Phylogenetic relationship is then determined between clones. This is done through determining if clone A and B are either subclones of each other or independent. If two solutions are possible, subclones are preferred over disjoint clones.

### Treeomics- Whole Genome Sequencing

Can distinguish true mutations from sequencing artefacts whereas conventional tools 7% of variants were misclassified due to artefacts skewing the phylogenies.

### PyClone- Whole Genome Sequencing

Bayesian clustering -hierarchical Bayes statistical model

Uses deeply sequenced 1000x data

Assumes no site mutation can occur more than once and mutations do not disappear or revert.

Clustering done in a pairwise manner using a posterior similarity matrix. Then hierarchical clustering was done using average linkage and the resulting dendrogram was used to find the clustering that optimises specific criteria.

## Current deficiencies in these programs: Motivations

### Dealing with high error SNVs and CNVs

It has been theorized that high error SNVs and CNVs can pose potential issues in how existing programs are defining the distance. For instance the below **equation 1** is one way to calculate distance between two normal distribution. Currently only treeomics attempts to distinguish true mutations from sequencing artefacts and thus overall dealing with the issue of high error mutations.

**Equation 1:** Unpaired test statistic, where z is a quantile from the standard normal distribution.

 It can be seen in **equation 1** that high error SNVs and CNVs will tend to cluster with other high error SNVs and CNVs. Thus eventually absorbing all the SNVs and CNVs into one big cluster which is not representative of the actual clonal relationships.

The solution employed in this project is to filter out high error SNVs and CNVs and cluster the low error/more reliable data. Then introduce back the high error SNVs and CNVs and only allow them to cluster to the existing clusters.

### Calculation of distances over multiple samples

Depending on how other programs calculate the overall distance across multiple samples, they can encounter an issue. For instance if multiple sample distance was calculated through **equation 2**

**Equation 2:** Euclidean distance for n samples (s)

It can be seen in **table 1** that case 1 and 2 produce the same distance. This poses an issue where high individual distances in one sample can be masked by another sample having a low distance. Whilst, overall this is not a big issue with small sample sizes such as n=2, this problem grows with increasing sample size.

|  |  |  |  |
| --- | --- | --- | --- |
|  | z1 | z2 | d |
| Case 1 | 0 |  |  |
| Case 2 | 1 | 1 |  |

**Table 1:** Example of 2 cases where different individual distances at sites z1 and z2 result in the same overall distance

Thus an equation was required to empathize the individual distances through amplifying them. **Equation 3** is how this project resolved this issue.

**Equation 3:** Solution for calculation of distances over multiple samples

### Updating of the distance matrix

Some of the existing programs use an initial distance matrix and then cluster accordingly based on maximum, minimum or average distance. For this problem, average distance is the most appropriate. However, it would be more accurate to recalculate the distance matrix each time a new cluster is created, which will give the real average distance between two clusters. The grouping of two clusters and the recalculation of their standard deviation and mean is shown in **equation 4**.

Part 1

Part 2

Part 3

**Equation 4:**

### Filtering out small CNVs

|  |  |  |
| --- | --- | --- |
|  | Germline | Cancer Line |
| Case 1 | 19AB CNV called | no 19AB CNV called |
| Case 2 | Normal | 19AB and 39AB CNVs called |

**Table 2:** Summary of 2 case studies’ CNV plot calls of high CNV genotypes

When analysing the CNV plots, it was discovered that there were consistently small regions that would appear in germline and only some cancer lines as seen in **table 2: case 1** or some cancer lines and not germline as seen in **table 2: case 2.** These small length CNVs also had absurd genotypes such as 19AB and 39AB. Current programs filtering is based on removing CNVs that are present in both germline and cancer lines. However, slight errors/variations in readings cause CNVs to have widely different genotypes, such as one sample would state a normal genotype, another will state 19AB and another would state 39AB. Thus while by ‘eye’ the germline and cancer samples have an anomaly, programmatically this is not the case.

This issue only occurs for small sized CNVs, thus to solve this a size filter has been applied. Based on empirical evidence the size filter threshold has been set to 10kb.

### Using temporal and spatial data to aid in the determination of subclones and clones

Currently none of the existing clonality tool use spatial or temporal data of the given samples to construct constraint equations which can improve the accuracy and efficiency of determining the clonal history of cancer.

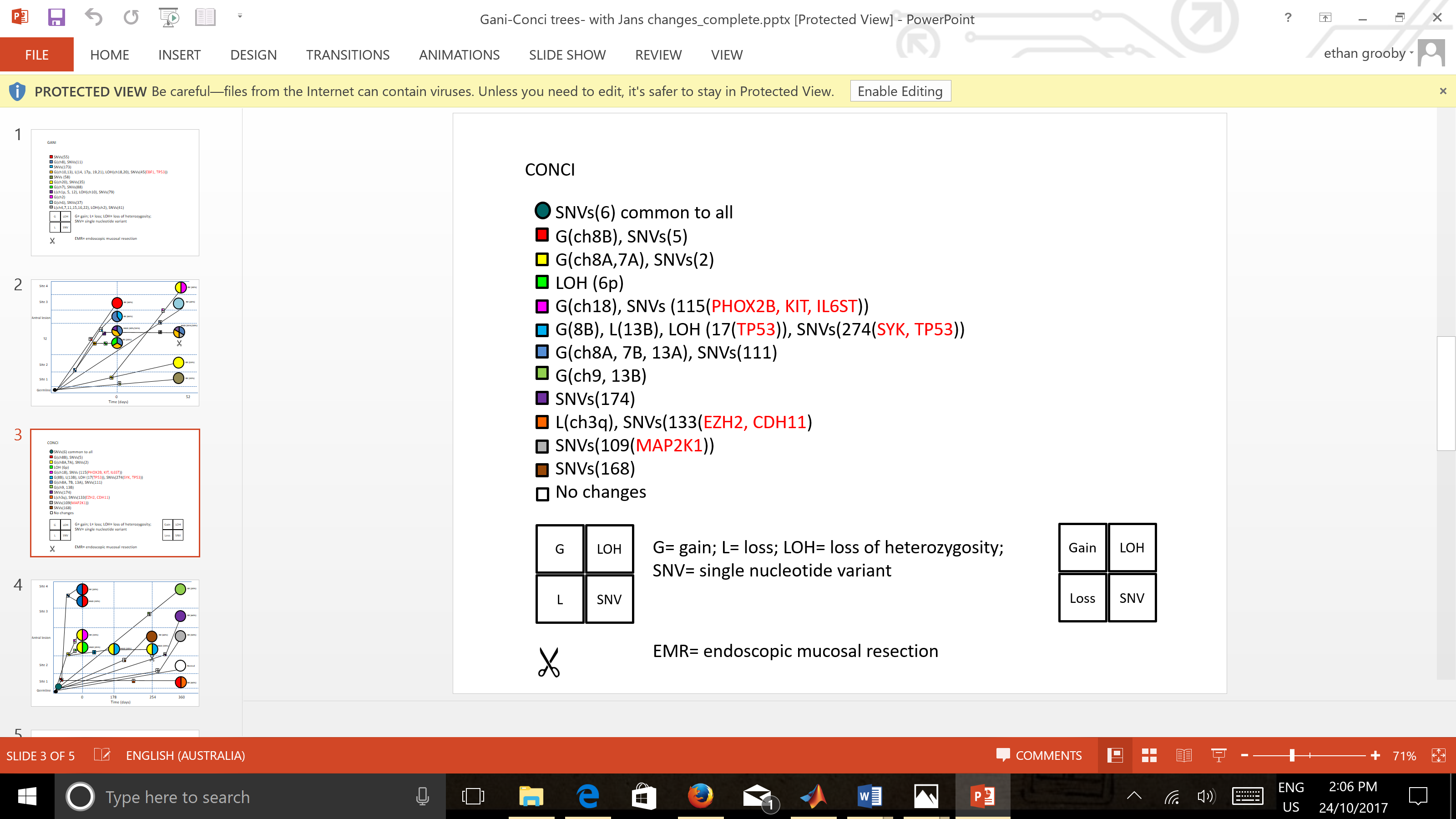
Temporal data could be used to form a constraint that if a sample at t0 has clone x with subclone y. Then the sample at t1 must also have clone x and subclone y present. Note that this constraint is not always true and is in the relationship between clones discussion.

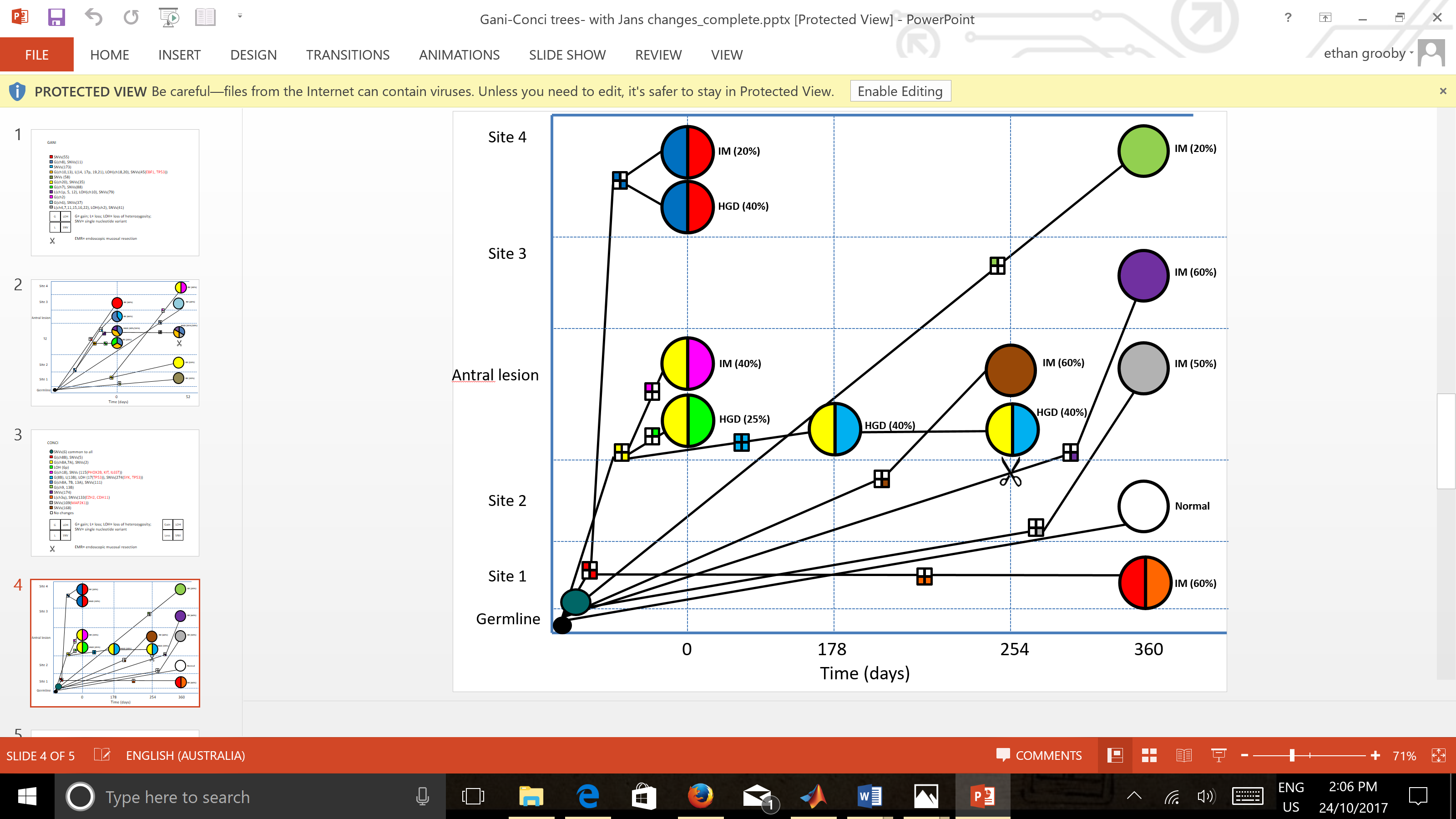
A suggested way that spatial data could be used is determining whether a CNV that has occurred in two samples means that this CNV occurred in a common ancestor or if it occurred in two separate events.

Existing programs work under the assumption that any mutation that occurs must only happen once in the phylogenetic tree. Whilst this assumption is almost certainly upheld with SNVs as the probability of two independent events occurring is next to zero. This is not the case with CNVs.

For example a copy number gain of chromosome 8 is a common mutation leading up to gastric cancer. The probability of the occurrence of gain in chromosome 8 independently is high due to natural selection. A gain of chromosome 8 confers a selective advantage for them cells, thus prolonging their survival and proliferation. Additionally, a gain in chromosome 8 results in genomic instability allowing more mutations to aggregate and thus the progression of cancer.

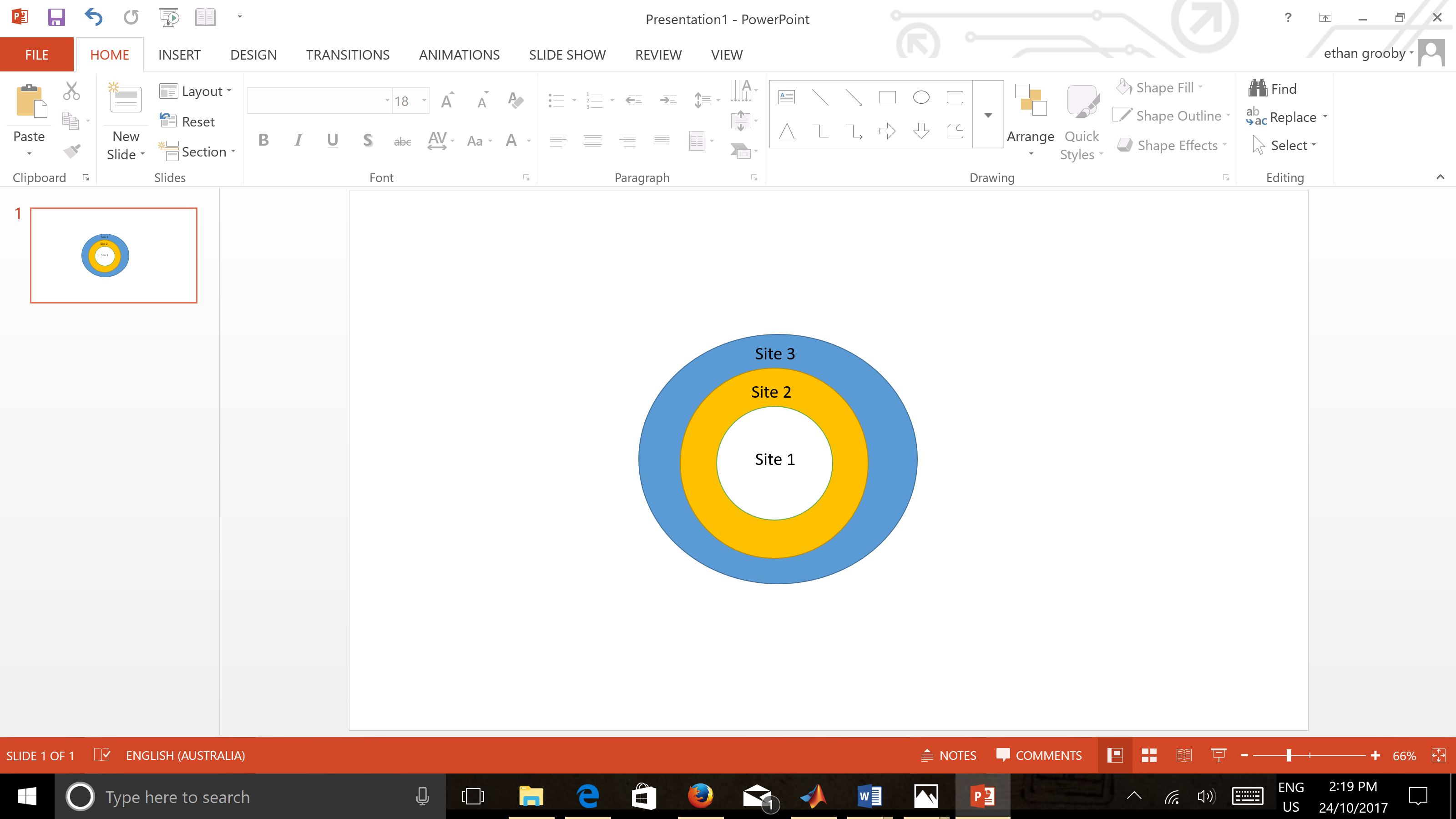
In case study 5 it was found that this gain in chromosome 8 had occurred independently in two parts of the genetic tree as shown below in **figure 2**.





**Figure 2:** Case study 5 phylogenetic tree showing a gain in chromosome 8 separately in the dark blue and yellow clones.

The way that spatial data can be used to solve this issue is shown in **figure 3**. If site 1 is related to site 3, then they must also be related to site 2. Thus using the **table 3** below it can be determined whether the gain in chromosome 8 occurred due to a common ancestor or independent events.



**Figure 3:** Relationship between sites 1-3 spatially

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Site 1 | Site 2 | Site 3 | Result |
| Case 1 | Gain Chr 8 | Gain Chr 8 | Gain Chr 8 | Common Ancestor |
| Case 2 | Gain Chr 8 | - | Gain Chr 8 | Independent Events |

**Table 3:** Gain of chromosome 8 due to common ancestor or independent events

### Relationship between clones

All the current programs also make the assumption that if a mutation occurs that it remains in the genetic tree. Whilst this is typically true, in the case studies analysed here, some of the patients received treatments or had their stomachs partially resected. Thus meaning whole clones could disappear. However, for some individuals there were surviving clones which then proliferate and cause gastric cancer to reappear. Thus, in future, the allowing of an input to state which samples have had treatment or resection, which would allow the removal of this assumption will lead to more accurate phylogenetic trees.

Additionally, there is the potential of disjoint clones due to tumour removing old genetic changes or one or more clones pushing out other clones through selection. Both these cases can lead to losses in mutations in subsequent samples. This issue has not been dealt with and poses an issue with the temporal constraint mentioned previously.

### Differentiation between AAB and ABB

Many of the existing programs with the exception of superFreq do not differentiate between which specific chromosome has been gained or lost. Thus scenarios such as AAB and ABB are deemed equivalent and grouped accordingly.

Solutions involve tracking heterozygous points along the chromosome of interest and analysing them across all samples. This can lead to the distinguishing between ABB and ABB.

# Implementation

## By hand method

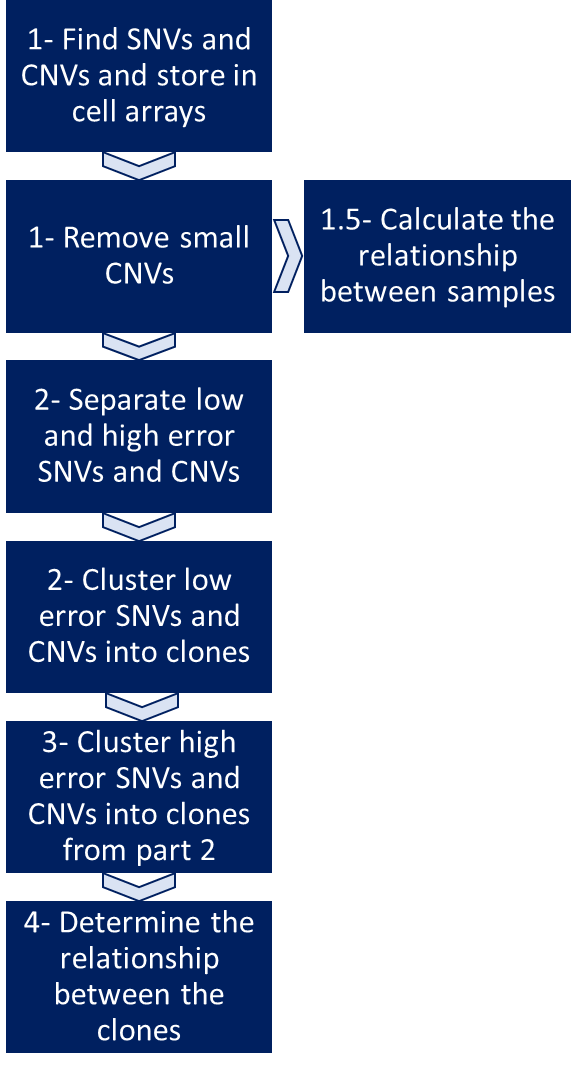
To determine the accuracy of the pre-existing program superFreq and this project, the 5 case studies were solved by hand. This was done through several steps and predominantly focused on CNV clones.

All sample CNV plots were viewed and clonality and CNV calls were determined rounded to the nearest 5%. Any small CNVs with high copy number calls were removed.

Inputting this into a table on excel, for ease of viewing. A phylogenetic tree is constructed with the aid of spatial and temporal data as well as superFreq results.

## Program Method

The code was run on 5 case studies. These case studies samples were run using Agilent 100x coverage exome. This was then processed by superFreq with this program using the clonality calls to determine the phylogenetic tree.



**Figure 4:** Flowchart of project’s code

## Break down of Matlab code

### Part 1: Getting CNV and SNV data

**locateSNV:** reads through the river data outputted from superFreq to find the location of the SNVs and CNVs.

Additionally filters out all small length CNVs which is currently defined as anything equal to or smaller than 10kb

**reconstruct:** uses the location of SNVs and CNVs to construct the SNV and CNV matrices which contain all the information required for all SNVs and CNVs respectively

### Part 1.5: SNV binary

**SNVrelationship2:** computes the SNV binary

SNV binary uses all the SNVs to calculate the probability any combination of samples being related. For example, if there are three samples S1, S2 and S3. **Table 4** shows the combinations possible.

|  |  |  |  |
| --- | --- | --- | --- |
| S1, S2 | S1, S3 | S2, S3 | S1 ,S2, S3 |

**Table 4:** Combinations of how the samples can be related

The program runs through each SNV and calculates the probability that it is present in S1 and S2 using **equation 5**

**Equation 5:** Calculation of if samples S1 and S2 are related using all SNVs

The purpose of 2a in **equation 5** is to differentiate SNVs that all have mean=0 but different standard deviations Thus a max input between 0.05-0.1 is used.

The overall purpose of the SNV binary was to use all SNVs before high error SNV filtering to determine the relatedness between all samples. This would then be used for two cases. Firstly, to study if the spatial constraint mentioned before is true. Secondly, if spatial constraint is not always true, then the SNV binary would become a way of determining if the same mutation occurred independently or in the same common ancestor. This would be achieved by looking at the probability that the two samples are related. If the probability is small then the same mutation occurred independently, otherwise they share a common ancestor.

### Part 2: SNV and CNV filtering and clustering

**SNVrelationship2:** filters out all germline mutations

**SNVcluster:** removes high error SNVs and clusters the remaining SNVs

Once an error cut off has been defined. The program checks to see what percentage of SNVs are going to be filtered out. If greater than 50% of the data is going to be filtered out, it will readjust the error cut off by increasing it in increments of 0.05. Thus, the maximum amount of SNVs that can be filtered out is 50%. The purpose of this condition is to insure there is enough SNVs left to be able to sufficiently cluster accurately.

A distance matrix is created using two equations:

1. **Equation 1-** to define the individual distance between two SNVs at a particular sample
2. **Equation 3-** which takes the magnitude of **equation 1** and compiles all the individual distances from multiple samples together.

This initial distance matrix is size where *m* is the number of SNVs.

This hierarchical clustering is applied to this distance matrix. To do this, the program searches for the smallest distance. If the smallest distance is larger than the maximum distance allowed then the program ends. Otherwise, it will cluster the two SNV clusters together.

This new cluster has its mean and standard deviation calculated by **equation 4.**

The distance matrix is then recalculated and the process is iterated to find the next smallest distance pair of SNV clusters. This program runs until either the smallest distance is greater than the maximum distance or all SNVs have been clustered.

**Extractinfo:** reads the output of SNVcluster to present the current clones, their clonality and their standard deviation

**CNVfilter:** filters out all germline mutations and high error CNVs

**CNVcluster:** clusters the CNVs

**extractinfoCNV:** reads the output of CNVcluster to present the current clones, their clonality and their standard deviation

**overallgroupwithouterror:** displays the current clones in SNVs and CNVs in graph form

### Part 3: Regrouping high error SNVs and CNVs

**SNVhigherror:** Using the current clones as determined by SNVcluster, the high error SNVs are reintroduced and clustered into the clones

Clustering is the same method as SNVcluster, however, with a constraint. High error SNVs are only allowed to be clustered into the pre-existing clones as determined by SNVcluster. Additionally, the pre-existing clones cannot cluster with each other. Thus this clustering step is determining which pre-existing clone should the high error SNV be placed. If none of the pre-existing clones are suitable as determined by the maximum distance cut off, then the high error SNV is removed. The reasoning behind this is that there is the possibility that some SNV’s clonalities are incorrectly called. Thus, if there is no logical way to place them into clones without creating contradictions or hindering the accuracy of the phylogenetic tree, then the SNV is removed.

**CNVhigherror:** Using the current clones as determined by CNVcluster, the high error CNVs are reintroduced and clustered into the clones

**Overallgroup:** displays the clones of SNVs and CNVs in graph form

### Part 4: Grouping of SNVs & CNVs and subclone analysis

**Subclone:** Groups each CNV clone to the closest SNV clone using **equation 1** given that it is not greater than the maximum distance. It then runs through all the clones to determine which ones are subclones of each other and which ones are independent.

This is examples using clone A and B

* If any of the clonalities in one sample sum to greater than 1, then clone A and B cannot be independent. Whichever clonality is smaller is the subclone.
* Otherwise check the ratio of clonality A divided by clonality B in all samples.

### Part 5: End

**Overallrelationship:** Runs the entire program

### Part 6: Comparison with superFreq

**Comparison:** Compares the clonality call of superFreq and then project’s results. This is displayed in two forms. Firstly, the percentage composition of each of this project’s clones with superFreq’s clones. Secondly, the percentage of superFreq’s clones within each of this project’s clones.

# Results and Discussion (2 pages)

## Results for the 5 case studies

**Refer to result data document for full results**

Case Study: P1003801

* Successful size filtering and germline filtering consistent with superFreq and by hand solution
* Overall clonal structure consistent with superFreq and by hand solution
* Whilst not determine with the current distance cut off, this program successfully can detect a correct subclone not detected by superFreq.
* Issue in not detecting a particular clone due to insufficient number of mutations in the clone

Case Study: P1003701

* Successful filtering consistent with superFreq
* Clonal structure consistent with superFreq and by hand solution
* Issue not detecting a particular clone due to insufficient number of mutations in the clone

Case Study: P080017

* Overall CNV clonal structure consistent with superFreq and by hand solution
* Detects two additional clones with provide more detail about phylogenetic tree subclone relationship.
* Issue not detecting two clones due to insufficient number of mutations in the clones.

Case Study: P14010

* Overall CNV clonal structure consistent with by hand solution
* Some issues with distinguishing between two particular clones
* Large number of CNVs with low clonalities formed into one clone

Case Study: P14006

* Overall CNV clonal structure consistent with by hand solution
* Large number of CNVs with low clonalities formed into one clone

## SNV Binary

The method provided in section **1.5 of implementation** is an accurate way of calculating the probability that any combination of samples are related. However, the computational time for this is long. Thus only one case study could be solved with this current method.

Several solutions are:

1. Only calculate the probability of two samples being related instead of all the combination.
2. Main rate limiting factor is the calculation of all the integral due to intervals of size ∞. Thus changing the intervals to an appropriate large number instead of ∞ may save computational time.
3. Computer with more computing speed

## Determination of error cut-offs and distance cut-offs

The current program is reliant on several key inputs. These include:

* The error cut-offs for SNVs and CNVs when doing the initial first round of clustering
* Maximum distance to determine when to stop clustering SNVs or CNVs in predominately the first round of clustering, but also the second round of clustering when high error mutations are reintroduced
* CNV size cut-off

Implemented and future solutions are discussed below.

### Calibration

Using key pieces of information about the input, a method of calculating these key inputs is required.

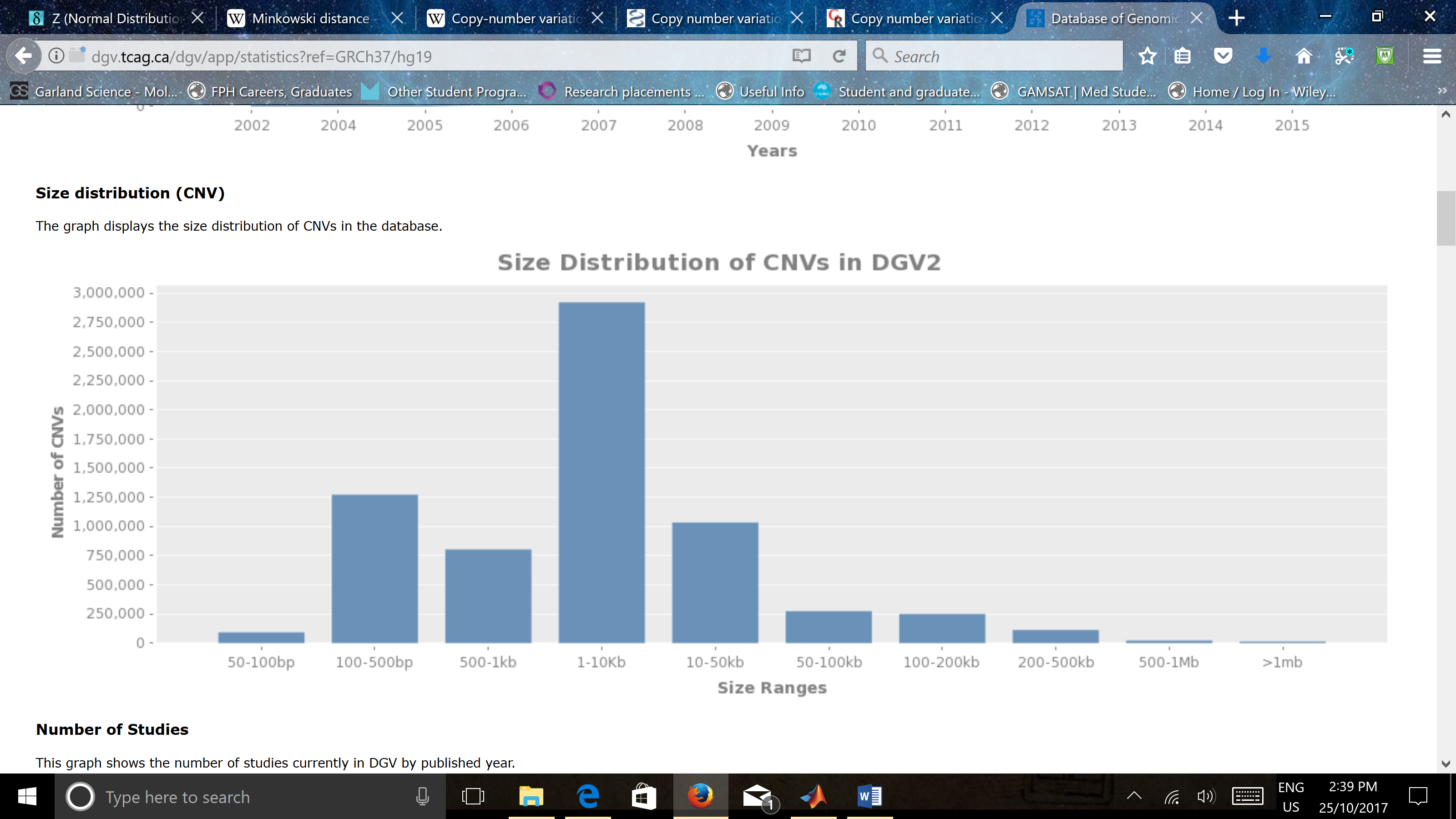
For maximum distance this could be solved using **equation 6.**

2.

**Equation 6:** Max distance cut off

As individual distance is calculated using **equation 1** which is an unpaired t test statistic. Given the number of samples (degrees of freedom), the distance/t value can be determine for when p=0.05. This distance will represent when two SNVs or CNVs are significantly different and thus should not be clustered together. The second part of the equation is translating the individual max distance cut-off value would equal when done across all n samples.

CNV size cut-off would be dependent on the coverage and accuracy of obtaining bin size reads. Another method, is using the distribution of all discovered CNVs sizes as seen in **figure 5**.



**Figure 5:** Database of Genomic Variants: Size distribution of CNVs <http://dgv.tcag.ca/dgv/app/statistics?ref=GRCh37/hg19>

It can be seen that that there exists a large number of discovered CNVs between the sizes of 1-10kb. Thus a potential cut-off could be 1kb using this data.

However, further information such as the odds that a discovered CNV has a particular size would be useful. Since it may be that CNVs of sizes 1-10kb are rare or potentially occur in specific circumstances.

Another solution would be to remove the CNV size filter altogether and approach the problem of small incorrect CNVs in a more advance method. Under the assumption that all CNVs detected are present. Then CNVs of size less than 10kb, instead of being filtered out, can be further analysed. This analysis would involve viewing the genotype called for that CNV and the error associated with it. Additionally, small focal increases in CNV producing genotypes of 39AB may be indicative of amplification of a specific gene. Thus the analysis of what gene is located in that section, viewing the gene sequence to confirm that gene in entirety is copied and potentially viewing the transcriptome levels to confirm that gene is amplified can lead to the conclusion of if the CNV is real or an error.

For error cut off currently this is a manual input, where you can filter no more than 50% of the data. A potential way of determining more accurately determining the error cut off would be to generate a density plot of the errors and then create a method of filtering out the high error outliers.

### Density plots

Whilst not implemented for error cut-off, density pots were produced for the initial distance matrices in the first and second rounds of clustering to analyse what cut-off would be appropriate. The desired shape of the density plot for the first round of clustering would be an exponential decaying function as shown in **result data document** thus there is a distinct range of distances tamper off quickly which would be appropriate for the maximum distance. However, the determination of which exact value is still an issue.

Additionally, there are density plots based on SNV and CNV clustering distances. These plots would be analysed in two ways. Firstly, the same process as mentioned in the previous paragraph. Secondly, to analysed the order of clustering and see if there is a specific point in which clustering distances increase markedly. This sudden jump in clustering distance would indicate that two unrelated clones are getting clustered together and thus would be appropriate to stop clustering at this point.

### Formula derivation

Overall key aim is to create a formula that determines a good initial input for maximum distance and error cut-offs. Then programmatically check to see if this is the case. If it is not, then a method of recalculating a more accurate cut-off from the results for the initial input. This is an example of where some form of probability maximization could be employed.

## Discussion of results

Overall this project can successfully cluster and thus infer the phylogenetic history of the progression of various mutations in an individual. There are four key issues about the results.

Firstly, for all the case studies the cut-off determination was successful and determined all the clones required for CNV clones. However, for SNVs the cut-off may need to be increased as there was consistently more clones in this project compared to the superFreq solution.

Secondly, the program missed on average 1 clone per case study. This was due to the filtering that clones needs to be of a sufficient size. While this seems like a sensible filtering constraint, it was discovered there are large whole chromosome CNVs with minimal error that occur in isolation of any other mutations. Thus the editing of this constraint to allow sufficiently large and low error mutations to occur by themselves in a clone would solve this problem.

Thirdly it was discovered that larger data set case studies that the largest CNV clone had clonalities of less than 5% in all samples. While this is the correct clustering, this clone should be filtered out in the future as the read depth and error associated with this clone suggests that it is highly likely all the mutations detected are due to error/chance and thus cannot be confidently stated as a clone.

Finally, the clustering of CNVs within a single SNVs clone was unsuccessful in some case studies with multiple CNV clones clustering to the same SNV clone. This should not occur as these CNV clones are distinct and thus have not been previously clustered together.

# Conclusion

|  |  |
| --- | --- |
| Shortcomings Discovered | Solved with code? |
| Dealing with high error SNVs and CNVs | Yes |
| Calculation of distances over multiple samples | Yes |
| Lack of updating the distance matrix | Yes |
| Filtering of small CNVs | Yes |
| Lack of use of temporal and spatial data | No |
| Lack of use of current treatments or resections occurring to the patient | No |
| Differentiation between AAB and ABB | No |

## What has been achieved

Critical analysis of existing clonality tools and the determination of areas of improvement. This project has then achieved a relatively successful clustering and phylogenetic tree determination program. It has added high error filtering which is currently only used in the treeomics clonality tool and size filtering which is not sued in any of the existing clonality tools. Additionally, created a novel way of calculating distances for multiple samples with updating the distance matrix to get average distance more accurately.

## Future potential

In future work, the solving of calibration issues will be a key focus. Then the making of a program that reassigns the clonality and genotype calls of outlier SNVs and CNVs to optimize the clusters instead of filtering them out completely. Finally, the adding temporal, spatial and treatment data to aid in the proper analysis of the phylogenetic relationship between clones.

# Appendix

## Code git hub

<https://github.com/ethangrooby/Analysis-and-Improvements-to-Existing-Clonality-Tools>