Hypothesis-free detection of genome-changing events in pedigree sequencing

Kiran V Garimella

Green Templeton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2015

Dedication

Acknowledgements

Acknowledgements

Abstract

In high-diversity populations, a complete accounting of *de novo* mutations can be difficult to obtain. Most analyses involve alignment of genomic reads to a reference genome, but if the haplotypic background upon which a mutation occurs is absent, events can be easily missed (as reads have nowhere to align) and false-positives may abound (as the aligner forces the reads to align elsewhere). In this thesis, I describe methods for *de novo* mutation discovery and genotyping based on a so-called "pedigree graph" - a de Bruijn graph where all available sequencing data (trusted and untrusted data alike) is represented. I constrain genotyping efforts to locations containing "novel kmers" - sequence present in the child but absent from the parents. I then apply Dijkstra's shortest path algorithm to perform the genotyping, even in the presence of sequencing error. In simulation, this approach provides a vastly more sensitive and specific set of *de novo* variants than traditional methods.

In Chapter 1, I use part of a real dataset, progeny from the crossing of two *Plasmodium falciparum* parasites, to demonstrate the pitfalls of the reference-based approach. I also introduce de Bruijn graphs for genome assembly.

In Chapter 2, I present a review on mutational mechanisms that generate *de novo* mutations, their rates, factors that influence their generation, and known events in various species.

Chapters 3 and 4 detail the software packages I have written for this work, the former including descriptions of the realistic variant read simulations, the latter detailing the graph genotyping algorithm. Results from the application of the algorithm to the complete *P. falciparum* dataset are presented in Chapter 5.

Finally, Chapter 6 discusses the work in a larger context and details various improvements that can be made in future work.

Contents

List of Figures

1 *Introduction*

Pathogens

2 Background

2.1 How genome changes

2.1.1 Cross-over

2.1.2 Gene conversion

2.1.3 Point mutations

2.1.4 Structural variants

2.1.4.1 Small (indels)

2.1.4.2 Large (fusions, NAHR)

2.1.4.3 Chromosomal changes

2.2 Rates

2.3 Factors influencing

2.3.1 Replication time

2.3.2 Mat/pat age effects

2.3.3 Biases/locality

2.4 Known events in species

2.4.1 P.f.

2.4.2 Human

2.4.3 Chimp

2.4.4 Others

3 Detection

3.1 Basic design

3.2 Current state of the art

3.2.1 Cross-over

3.2.2 Gene conversion

3.2.3 Point mutations

3.2.4 Structural variation

3.2.5 Chromosomal

3.3 Potential for de novo assembly to detect events

3.3.1 Why assembly should theoretically work

It is conceivable that a mapping-based approach and dozen different variant callers, each purpose-built for dealing with a particular mutational class, could result in a complte callset of variation between samples. However, such a baroque processing pipeline has several drawbacks. In particular, one is beholden to a particular read mapper's determination of a read's unique home in the reference genome. This poses significant problems when reads are of insufficient length to span a repetitive region in the genome, and when the sample being analyzed is sufficiently divergent from the reference such that no plausible home for the read can be found. In these cases, each algorithm is required to go out of its way to adjust read mappings to be more consistent with each other across a locus, or align reads to alternative reference sequences that hopefully contain a closer version of the sequence in question. The result is a hodge-podge of representations of data mapped to a single reference genome in multiple ways, or to multiple reference genomes, callsets with different output formats, etc.

One might consider a different method entirely: *de novo* assembly. In this approach, rather than mapping reads to a reference genome, we dispense with the idea that there should be a single reference genome at all against which variation should be described. Instead, the next-generation sequencing reads are used to produce a "directed acyclic graph" from the data, which is simply a set of vertices (a string representing some amount of genomic sequence) and edges (a set of connections between one vertex and others, representing sequence overlaps). The graph is "directed" in the sense that the connections between vertices are given an orientation that specifies the direction of travel required to obtain linear sequence in the 5′ or 3′ direction. The graph is "acyclic" in the sense that repetitive sequence manifesting as loops will get collapsed into a single loop; the number of times one should traverse the loop in order to recapitulate the input sequence is not stored. Formally, we define this graph as $G = V, E$, a set of vertices and edges between them. The overlaps between reads are detected, and by examining these overlaps, we can walk the graph to detect long contiguous regions of sequence, or "contigs". In theory, with perfect data and long reads that span repetitive regions, the entire linear sequence of a chromosome can be reconstructed in this manner.

Computing the overlaps is a difficult computational challenge to overcome, and there are a variety of methods for doing so. We shall focus on one of the most straightforward methods, effectively amounting to a hashtable approach: construction of a de Bruijn graph. Each read is decomposed into small subwords of a fixed length, k, typically referred to as *kmers*. These kmers become the vertices in graph (which, in most implementations, is basically a hashtable). For every kmer in a read, we record the base immediately preceeding and following it. As we process the first (last) kmer in a read, the edge information for the preceeding (following) base is absent. However, if the read overlaps with another one, the overlapping read should contain the same kmer with the missing edge information. We proceed in this manner until all reads are processed and the hashtable is fully populated with every kmer in the dataset and corresponding connection.

For multiple samples, we still construct the graphs separately, but the results can be stored together by introducing the concept of a "color" for each edge. Thus instead of storing a single edge (from a single sample), we store an array of edges, indexed by an integer (or "color") representing each sample. This has the convenient property of allowing us to quickly look up any single kmer in the genomes of the N samples (essentially an $O(1)$ operation) and find edge information for each sample separately. Variants between the samples will manifest with different edge information for the same kmer, and traversing the graph from where those edges diverge to where they converge again should reproduce the alleles of the variant.

There are tremendous benefits of approaching the problem of finding *de novo* variants this way. First, we are not reliant on the genomes having low divergence from a reference sequence in order to facilitate read mapping. Instead, the reads are effectively aligned to each other, and variants are identified via graph traversals. Thus, in regions of high genomic divergence between samples, we needn't worry that the reads had no place in the reference to map. This gives us a much more complete usage of the data. Second, the graph-based approach is a much cleaner representation of the biology we're attempting to describe. By constructing a "pedigree" graph, that is, a graph representing child, mother, and father, we can quickly identify variants in a child but not in either parent. Furthermore, the graph naturally encodes the parental origin of a *de novo* variant. Should such a variant occur on a haplotype transmitted by the mother, but completely absent in the father, the graph will encode the edge conflict between child and mother and the complete absense of the kmer and edge in the father. This is exceptionally difficult to achieve with a mapping-based approach, even if one tries to map reads to multiple reference genomes and disntangle the results later.

3.3.2 *Haploids/perfectly assembled diploids*

Assuming one has perfect data, the assembly of haploid genomes is more-or-less straightforward. The graph is constructed as indicated above. Because the genome is haploid, any kmer with multiple edges in a sample must represent a bit of repeated sequence. Variants will manifest as simple graph bubbles between the samples.

Diploid data is a little trickier, as variants will manifest within a single sample. Should a *de novo* variant occur proximal to a variant between the diploid copies of a chromosome in a sample, it may impair the ...

3.3.3 *What do we look for*

...

3.4 Limitations of read data

With perfect data, long reads, and uniform coverage, it is theoretically possible to reconstruct entire chromosomes from $5'$ to $3'$ telomeres. However, data is never perfect, NGS reads tend to be short, and coverage fluctuates.

3.4.1 Sequencing errors

Sequencing error can manifest from a variety of sources. Most commonly for Illumina data, error takes the form of single base errors in reads. Rarely, insertion and deletion events may also be present. An understanding of how the sequencing platform works can shed light on these behaviors.

Illumina sequencing is an implementation of so-called "sequencing by synthesis", wherein a small region of the genome is sequenced by providing a single-stranded DNA (ssDNA) template for which the complementary strand must be synthesized. Each nucleotide is labelled with a fluorophore (four different fluorophores for each of the bases, A, C, G, and T) and a "reverseable terminator" which prevents the incorporation of more than one nucleotide per template at a time. The sequencer monitors the synthesis with a camera, taking an exposure each time a nucleotide is incorporated into the fragment. With each cycle of the sequencer, a nucleotide is incorporated, four pictures are taken (each one with a filter that should measure the intensity of flourophores corresponding to a particular nucleotide), and reverseable terminators washed away to prepare for the next cycle. To achieve an adequate signal to noise ratio (SNR), each template will exist in a cluster of thousands of identical templates, and each should incorporate identical nucleotide in identical positions.

Several issues occur in practice. First, nucleotides are labelled with fluorophores that emit light across a range of wavelengths, and the spectra for each overlap slightly. While a nucleotide-specific filter should block out most of the light from the other three, these filters are not perfect, and some light leakage from another set of fluorophores can occur. This "cross-talk" may impair the base-caller's ability to assign the proper label to an event, thus occassionally miscalling nucleotides. Furthermore, the miscalls will be biased towards their cross-talk partners.

Second, with each cycle of the chemistry, a wash step removes the reversable terminator and prepares the flowcell for the next set of labelled nucleotides. However, the wash itself is rather abrasive, and can dislodge templates from the flowcell itself. Thus, with each cycle, the clusters of identical templates get smaller and the SNR decays. This can result in diminished ability to call a nucleotide properly.

Third, on occassion, not every template in a cluster will incorporate a nucleotide. More rarely, a template might incorporate two (though the reversable terminator for a nucleotide would have to be absent for this to occur). In either case, subsequent cycles will find some templates out of phase with others in the same cluster. Signal from the cluster is reduced once again, leading to basecalling problems.

Fourth, PCR amplification is used to produce the clusters of identical templates. If an early iteration of the PCR should produce a mismatch, that mismatch will be exponentially propagated to other templates in the cluster, resulting in a consistent error in all of those templates, manifesting as a high-quality single nucleotide mismatch in the read. If PCR amplification is used to amplify the original input DNA as well, this error may be compounded.

Fifth, the wash step that should remove the flourophores from nucleotides in preparation for the next cycle may be inefficient, resulting in a mix of signal from different nucleotides. This effect is clearly observed at the end of homopolymer sequences, where the next nucleotide is often miscalled as whichever base was in the immediately preceeding homopolymer run. Additionally, early versions of the Illumina chemistry (which much of our data will have been sequenced with) suffered from an inefficient wash step for T nucleotide fluorophores, resulting in decreased SNR after Ts are incorporated.

Sixth, inverted repeats can cause single-stranded templates to fold in on themselves, effectively blocking out positions for the labeled nucleotides to anneal. This would force the synthesis to skip ahead several bases, resulting in an apparent deletion in the read.

(TBD: should I talk about error cleaning / correcting in this section?)

3.4.2 *Read length / repeat length*

Genomes often contain repetitive regions, originating by a variety of mechanisms (tandem duplication, short tandem repeat expansion, etc.). In order to unambiguously reconstruct such regions, read lengths must be long enough to span the repeat and be anchored on either end in unique sequence. In some applications, a paired-end read with sufficiently large insert size might be able to span the repetitive region. Some assemblers (e.g. McCortex) are capable of leveraging such long-distance information to span repetitive regions and provide information on how many times a graph loop should be traversed. Effectively, this makes the read length one considers when determining the ability to span a repeat to be the mean fragment size of input data, rather than the read length itself.

3.4.3 *Coverage fluctuations*

Ideally, a sequencing experiment will yield uniform coverage across the genome. However, in practice this is never the case. Coverage fluctuations occur for a variety of reasons. In particular, high GC content can cause problems during any PCR steps. G and C nucleotides are bound with three hydrogen bonds, rather than the two that form between As and Ts. The ideal temperature to achieve complete denaturation of the strands during amplification may not be precisely known at the time of library preparation, and it may be impractical to measure it for each sample.

Furthermore, if library complexity is low (that is, if the sequencing library is constructed from very little input DNA), an aliquot from this library may suffer from a random sampling bias wherein some sequences are overrepresented (duplicate reads - reads that start that the exact same position in the genome and are effectively multiple copies of the exact same genomic fragment) or underrepresented (little to no coverage over some regions of the genome).

Finally, a genome might not be sequenced to sufficient coverage in order to ensure that all regions of the genome can effectively be sampled. This can occur for certain precious samples where a large amount of DNA cannot be obtained easily and experimenters are forced to proceed with whatever they can get. We can employ Lander-Waterman statistics, which provides estimates for read coverage based on a Poisson model, to compute the amount of genome we might expect to miss when sequencing to a target depth of coverage. Let $a = (N/G) \times L$, where N is the number of reads, G is the genome size (in the case of the *P. falciparum* genome, $\approx$ 23 million bp), and L is the read length (for most of our samples, 76 bp). Then the probability that there are no read starts in some interval I is:

$$p = e^{-a} \tag{3.1}$$

and the probability that there are ≥ 1 read start in some interval is:

$$q = 1 - e^{-a}. \tag{3.2}$$

```r
G = 23e6;
L = 76;

lw = NULL;

for (a in seq(5, 100, by=5)) {
    nReads = a*(G/L);
    nNucleotides = a*G;
    pctGenome = 100 * (1 - exp(-a));

    if (is.null(lw)) {
        lw = c(a, nReads, nNucleotides, pctGenome);
    } else {
        lw = rbind(lw, c(a, nReads, nNucleotides, pctGenome));
    }
}

colnames(lw) = c("coverage", "numReads", "numNucleotides", "pctGenome");

kable(lw, row.names=FALSE);
```

coverage	numReads	numNucleotides	pctGenome
5	1513158	1.150e+08	99.32621
10	3026316	2.300e+08	99.99546
15	4539474	3.450e+08	99.99997
20	6052632	4.600e+08	100.00000
25	7565789	5.750e+08	100.00000
30	9078947	6.900e+08	100.00000
35	10592105	8.050e+08	100.00000
40	12105263	9.200e+08	100.00000
45	13618421	1.035e+09	100.00000
50	15131579	1.150e+09	100.00000
55	16644737	1.265e+09	100.00000
60	18157895	1.380e+09	100.00000
65	19671053	1.495e+09	100.00000
70	21184211	1.610e+09	100.00000
75	22697368	1.725e+09	100.00000
80	24210526	1.840e+09	100.00000
85	25723684	1.955e+09	100.00000
90	27236842	2.070e+09	100.00000
95	28750000	2.185e+09	100.00000
100	30263158	2.300e+09	100.00000

(Note: Surprising - I would have expected the need for a higher coverage to capture the genome fully... double-check these results)

3.4.4 Rare vs error

...

3.4.5 Algorithm error can mimic real biology

Finally, the assembly algorithm employed might make errors that mimic real biology, confounding the results. For example, an overzealous error-correction algorithm may attempt to correct a low-frequency nuclotide in the read data under the assumption that its sparsity reflects an error process, when it fact it is derived from a repetitive sequence that has slightly deviated from the original ancestral sequence, and simultaneously has poor coverage that masks the problem. This would hide the divergence of one of the repeat's copies.

Additionally, in an error-cleaning step that should look for short branches in the graph that are likely derived from sequencing error and discard them, the algorithm may "over-clean" (too many events are thrown away) or "under-clean" (too many events are retained). These may conspire to artificially reduce effective contig length by either removing critical kmers or presenting too many ambiguities in the graph to effectively complete the traversal.

Finally, some assembly algorithms will attempt to leverage long-range information to scaffold short contigs into longer "supercontigs". This relies on the ability to map the original paired-end reads back to the contigs and detect read pairs anchored in two contigs. Should the alignment be in error (due to aforementioned sequencing errors, coverage fluctuations, etc.), the scaffolder may join two contigs that do not represent a real sequence in the genome. This may manifest as a structural variant or recombination event where none has occured.

3.5 Outline of the work

4 Methods

4.1 Overview

4.1.1 Start with NGS data from mother, father, child

4.1.2 Need to identify relevant motifs within data

4.1.2.1 Discovery/exploration

4.1.2.2 Validation

4.1.2.3 Interpretation

4.2 Discovery/exploration

4.2.1 Assembly

4.2.2 Annotation of kmers and links

4.2.3 "Fishing"

4.2.4 Visualization

4.3 Validation

4.3.1 In silico

4.3.1.1 Contig decoration

4.3.1.2 Decision (trust / not trust)

4.3.1.3 Simulations

4.4 Empirical

4.4.1 Known AHRs

4.4.2 Known NAHRs

4.4.3 Comparison of 3D7 (Illumina) to 3D7 (ref), using 3D7 (PacBio) to adjudicate

4.5 Experimental

5 Pf

5.1 Lit review

5.1.1 Review of Kong et al., 2002

Augustine Kong et al. discuss a new genetic map of recombination rates using genotyping information from 869 individuals in 146 Icelandic families. This is the first such map made after the sequencing of the human genome, and is thus able to leverage the new reference sequence in order to correctly order the genotyped markers. It is a substantially higher-resolution map than provided by the former gold-standard, the Marshfield map. The Marshfield map contained data on only 188 meioses, whereas the Kong et al. map contained data on 1, 257. The new map reveals marked differences in recombination rates between males and females (e.g. the recombination rate in female autosomes is a factor of 1.65 higher than that observed in males) for reasons beyond sequence features.

6 Chimp

6.1 Lit review

6.1.1 Review of Kong et al., 2002

Augustine Kong et al. discuss a new genetic map of recombination rates using genotyping information from 869 individuals in 146 Icelandic families. This is the first such map made after the sequencing of the human genome, and is thus able to leverage the new reference sequence in order to correctly order the genotyped markers. It is a substantially higher-resolution map than provided by the former gold-standard, the Marshfield map. The Marshfield map contained data on only 188 meioses, whereas the Kong et al. map contained data on $1,257$. The new map reveals marked differences in recombination rates between males and females (e.g. the recombination rate in female autosomes is a factor of 1.65 higher than that observed in males) for reasons beyond sequence features.

7 Discussion

7.1 Lit review

7.1.1 Review of Kong et al., 2002

Augustine Kong et al. discuss a new genetic map of recombination rates using genotyping information from 869 individuals in 146 Icelandic families. This is the first such map made after the sequencing of the human genome, and is thus able to leverage the new reference sequence in order to correctly order the genotyped markers. It is a substantially higher-resolution map than provided by the former gold-standard, the Marshfield map. The Marshfield map contained data on only 188 meioses, whereas the Kong et al. map contained data on $1,257$. The new map reveals marked differences in recombination rates between males and females (e.g. the recombination rate in female autosomes is a factor of 1.65 higher than that observed in males) for reasons beyond sequence features.

References