Aarhus University

Challenges in Bioinformatics

Quarter 1, October 2013

Crossover and Gene Conversion Discovery using Local Phasing

Author:
Stefania Bjarney
Olafsdottir

Supervisor:
Thomas Maillund, Phd
Professor:
Jotun Hein, Phd

October 29, 2013

Abstract

There are two main phenomena which explain the variability of populations; mutations and crossovers. We will discuss an implementation of an algorithm to detect crossovers using locally phased trios, and using the discoveries of crossovers, we want to find gene conversion.

By using only local phasing, we rely on a single read, a pair of reads, or a chain of pairs of reads, to cover regions containing two single nucleotide polymorphisms (SNPs). We thus form phased pairs of SNPs for all individuals in the trio and compare the strands of the offspring with the strands of the parents. We will go through the combinatorics involved in the strand comparison.

We present results from testing our algorithm on a chimpanzee trio and a orangutan trio, although our results were not fruitful. For the chimpanzee trio the main problem was extremely low coverage of the genome. The orangutan trio turned out to be faulty, as the alleged DNA of the mother was DNA from another orangutan.

There are improvements to be made to the algorithm in several ways. Some of them are quite straight forward and mainly need programming time. Others need a little more thought, such as combining local phasing with long range phasing to yield greater coverage.

Acknowledgements

I collaborated with Thomas Maillund and Soren Besenbacher. They preprocessed the sequencing data to the form of SNP pairs, as well as being part of the thought process through this project.

Brainstorming sessions with Jotun Hein were fruitful, and he was a source of ideas for new angles and approaches.

Contents

1 Introduction

Our goal is to use locally phased trios (parents + offspring) to find crossovers and gene conversions. Let's introduce some concepts to be able to discuss this process.

1.1 Genetical concepts

The genetic information, DNA, of an individual is written in the individuals *genome*, which consists of 23 paired *strands*. The strands are called chromosomes and each strand in a pair is inherited from one parent. The parent created the strand from its own two strands, i.e. an offspring's maternal strand is a mixture of the maternal and the paternal strands of the offspring's mother. The alphabet used to write the DNA is A, C, G, and T (refering to the nucleotides adenine, cytosine, guanine, and thymine, respectively) and a letter is refered to as a base of the genome.

The strands are mostly identical between individuals, and thereby also the two strands inherited from each parent, but out of ~3 billion bases (EMBL-EBI and SangerInstitute (2013)), a very small fraction is variable between humans. In our method we utilize a kind of *variable site* called *single nucleotide polymorphisms (SNPs)*, which are variable in the sense that a part of the population may for example have the *variant* A while an other part has the variant G. According to the latest dbSNP build (138, august 2013) there are less than 300 million submitted SNPs (NCBI (2013)) in the human populations, which less then 1% of the genome. We will later see why this sparsity of SNPs is limiting for our method.

To describe all of the 23 strands (note, not the pairs of strands), like they are in most of the population, there is a consensus for a so-called *reference sequence* for species. On the variable sites, one of the bases will be the one in reference sequence, here on refered to as the *reference base*. Note, that the fact that there is a consensus about a site being variable, e.g. the reference base being A and the alternative base being G, does not necessarily mean that each individual contains both variants.

When a site in an individual contains two kinds of bases, e.g. A and G, we will refer to the site as being a *hetero* (meaning different) site. If the site only contains one kind of base, e.g. A and A, or G and G, we will refer to it as being *homo* (meaning same).

To determine an individuals genome the current most accurate method is Next Generation Sequencing, discussed in the next section.

1.2 Next Generation Sequencing

Without going into details of the *Next Generation Sequencing (NGS)* technology, the output data comprises millions of so-called *short paired-end reads*. 'Read' refers to a string of the nucleotide alphabet, describing the DNA on a particular region, the region being unknown at the time of output. 'Short' in this context varies from ~30 to ~250 bases. 'Paired-end' refers to the fact that the reads come in pairs which we know come from the same strand, i.e. the same parent. We will later see why this pairing is convenient.

The output reads are not positioned on the above mentioned reference sequence by default; algorithms are needed to compare the reads to the reference sequence and *align* them to the reference sequence, so that they are correctly positioned in the genome. We then have a stack of aligned reads, as seen in figure 1. For each position on the reference sequence, we then have a pile of bases, but at this point we cannot identify from which parent each base is. What we do know, is that each read, and furthermore each pair of reads, comes from the same parent, and that we can utilize, as well as the variable sites, to determine a parent of origin of each base.

1.3 Local phasing

The process of establishing which letters go together is called *phasing*, which yields us data that is *phased* into the two strands it is originated from. As we see in figure 1, most positions only have one letter in their pile, but some of them have two, and have been colored red. On those positions, the strands of the individual are not the same. By looking at a pair of reads containing two variable sites, we can establish which letters on two variable sites go together, i.e. are from the same parent. This way we are *locally phasing*.

To see an example of how the local phasing of an individual would work, let's look at figure 1 again and see what information we can extract from it. Let's label the (red) variable sites from 1 to 4, and tabulate informative reads (reads which contain a variable site) such that a row represents a read and a column represents a variable site.

	SNP_1	SNP_2	SNP_3	SNP_4
$read_1$	G	-	-	-
$read_2$	-	A	-	-
$read_3$	C	G	G	-
$read_4$	C	G	G	-
$read_5$	G	A	C	C
$read_6$	C	G	G	A

$read_7$	G	A	C	C
$read_8$	C	-	G	A
$read_9$	-	-	C	-
$read_{10}$	-	A	-	-
$read_{11}$	-	A	C	-
$read_{12}$	-	G	G	A
$read_{13}$	-	A	C	C
$read_{14}$	-	-	C	C

Table 1: Reads containing variable sites, indexed according to their row of appearance in figure 1, where variable sites are colored red.

Our goal here is to combine the variabilities into two separate strands. In table 1 we have tabulated all combinations of letters the variable sites in this region yield. In this case, it is very straight forward to realize what the separate strands look like. We combine reads 1, 2, 5, 7, 9, 10, 11, 13, and 14 into one strand, and reads 3, 4, 6, 8, and 12 into the other strand, yielding the strands in table 2.

	SNP_1	SNP_2	SNP_3	SNP_4
$reads_A$	G	A	C	C
$reads_B$	C	G	G	A

Table 2: Reads of table 1, combined into their unity of variabilites: reads A: {1, 2, 5, 7, 9, 10, 11, 13, 14}, and reads B: {3, 4, 6, 8, 12}, yielding the two separate strands in this location.

With this, we have locally phased this region into its two strands. Note that we do not know from which parent each strand comes, we only know that, in these locations of the genome, G, A, C and C go together and C, G, G, and A go together. To be exact, the strands look like this:

```
... GCTTATGTCCACGACCAGCC ...
... CCTTATGTCCACGGCGAGCA ...
    ^           ^ ^   ^
```

But we are merely interested in the variable sites, the ones marked with a '^', since those are the only positions on the strands that give us any information on crossovers and gene conversions.

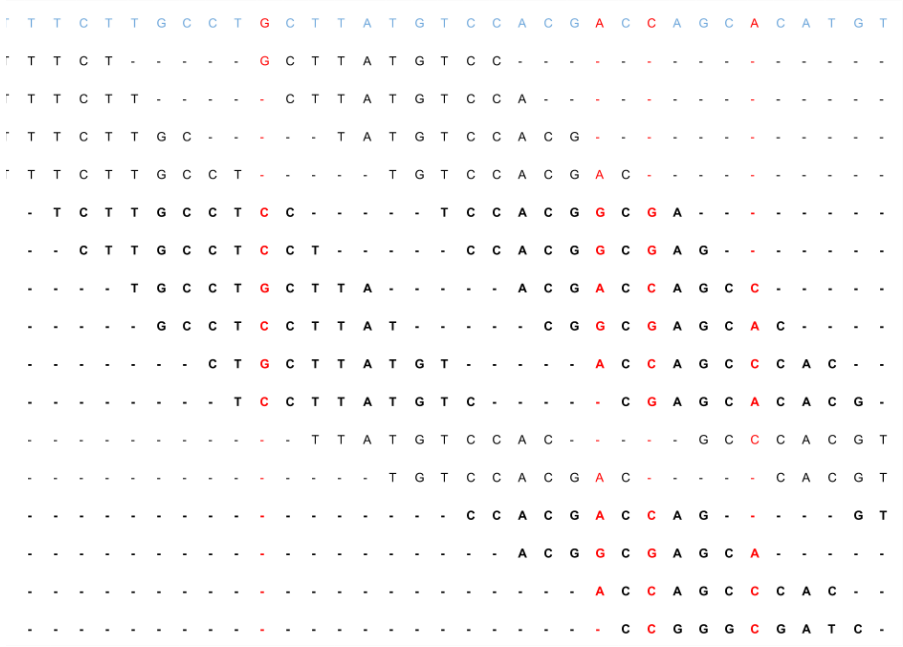

Figure 1: Aligned paired-end short reads. The abovemost blue string is a reference sequence, where each letter has its own reference position. Each of the black strings are one read. Two reads in the same line represent paired reads, so we know that they come from the same strand in a pair of strands, i.e. from the same parent of that individual. Looking closely, we see that the The SNPs (the variable sites) have been colored read. In our actual data set, both the reads, and the interval between them are longer; this image is simulated for visualization of the process.

1.4 Parent of origin

To assign the phased strands to the parents, we need the region to be phased in at least one of the parents. It is intuative that when the maternal nucleotide in an offspring, is identical to the paternal one, there is no way of telling which one comes from which parent. We have thus so far established, that to determine the parent of origin, (a) we need a variable site, (b) we need information on the parents, and (c) we need the parents to differ.

1.5 Crossovers and gene conversions

Crossovers are the points on which a parent switches from its maternal strand to its paternal strand in the creation of the strand which is to be given to its offspring. See figure 2. *Gene conversions* are a more complicated biological process which we will not go into detail here, but as for sequencing data, their manifestiation is the same as two crossovers in a small region. According to Sasaki, Sugino, and Innan (2013) the regions are ~70-1000 bases.

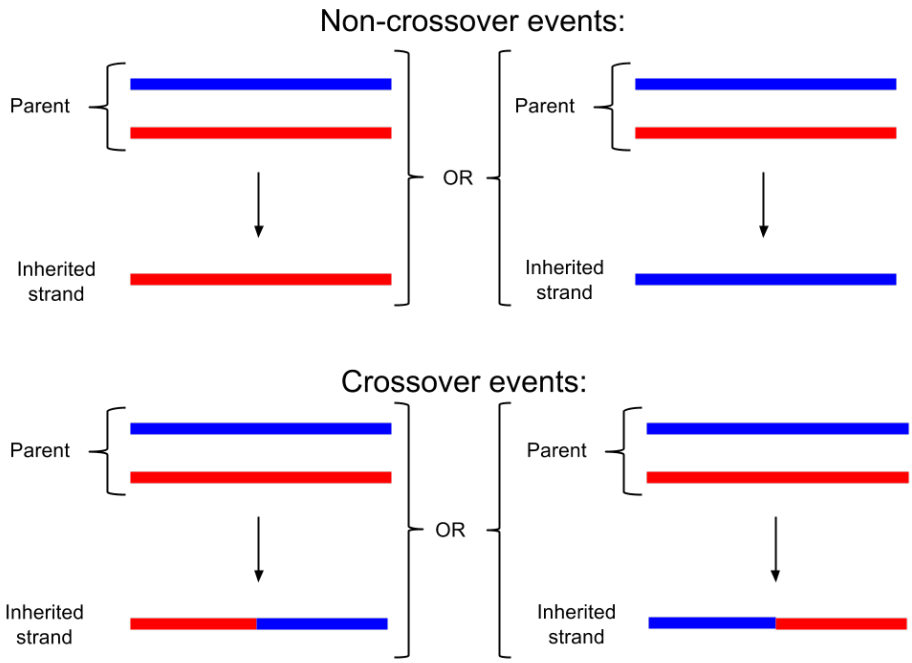

Figure 2: All possible inheritance instances on a region, given that there was no mutation. The lower ones represent the crossover inheritance instances, and the points where blue goes to red, or red goes to blue, are the points of the crossover event.

1.6 Detecting crossovers and gene conversions

To be able to see a crossover, (a) we need two variable sites, on each side of the crossover, in a (b) phased region in the offspring and (c) the parent needs to be phased in the same region. Not all crossovers can be seen, despite these conditions being fulfilled, we will discuss that in the Combinatorics chapter.

Since we restrict ourselves to local phasing, we are limited to analysing regions containing at least two variable sites caught by a read, a pair of reads, or a chain of pairs of reads. We will miss out on crossovers in regions which are not covered by a pair of variable sites, as showcased in figure 3.

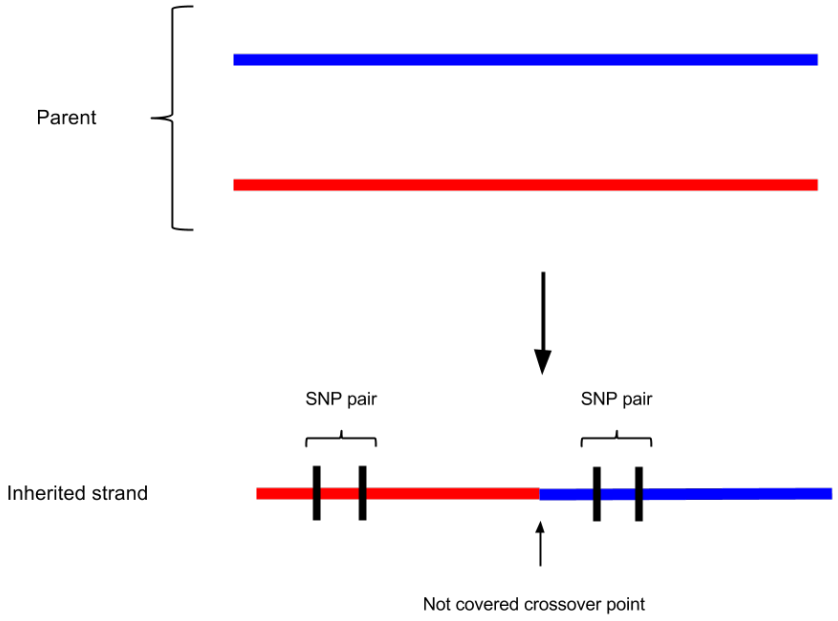

Figure 3: A crossover in a region which is not covered by SNP pairs.

The human genome has been measured to be ~30 Morgans (Kong et al. (2002)), where a Morgan is a measuring unit, defined as the region needed to see 1 crossover event. This means that the human genome should contain ~30 crossovers per inheritance in an individual, and ~60 in total per individual, i.e. a crossover every 100 million bases, from each parent.

With the sparse distribution of SNPs in the genome, discussed in the genetical concepts section, we have to be quite lucky to find crossovers this way, let alone to see two instances of crossovers close to each other, to be able to assume gene conversion.

Clearly, having the whole genome phased would make this easier, since we would always see a crossover if we could compare a phased offspring to a phased parent. So far it is not trivial to phase the whole genome. It has been done with a few statistical inference methods, among them is the *long range phasing* method developed by Kong and Masson (2008), utilizing the well known pedigree of Icelanders.

2 Combinatorics in crossover search

Let's go through the combinatorics involved in comparing locally phased sequencing data between all individuals in a trio. First let us introduce some concepts to make it easier for us to refer to this comparison.

In our discussions and assumptions in this chapter, we assume only two possible events, non-crossover or crossover, i.e. we dismiss new mutations and sequencing errors.

2.1 Boolean for each SNP

Since the two possible bases of one SNP are independent of the two possible bases of another SNP, we can treat the bases one SNP as a boolean variable, where 'true' represents having the reference base and 'false' represents having the alternative base, see figure 4. This way we will use a 2x2 matrix to describe possible SNP pair combinations.

2.2 Possible SNP pair combinations

When we define a pair of SNPs this way, we can have $2^4 = 16$ possible combinations of bases:

```
0 0     0 0     0 1     0 0     1 0     1 0     0 0     0 1
0 0     0 1     0 0     1 0     0 0     0 1     1 1     0 1

1 1     1 1     1 0     1 1     0 1     0 1     1 1     1 0
1 1     1 0     1 1     0 1     1 1     1 0     0 0     0 0
```

Remember that these matrices represent a phased pair of SNP, so a column represents a SNP and a row represents a strand.

To find crossovers, we need this form of data for all individuals in a trio. If we needed to look at all possible combinations for all individuals, we would have $(2^4)^3 = 4096$ possibilities, but fortunately we can simplify these cases somewhat (at least for a discussion such as this one, although the algorithm for detecting the cases can not be simplified to the same extent).

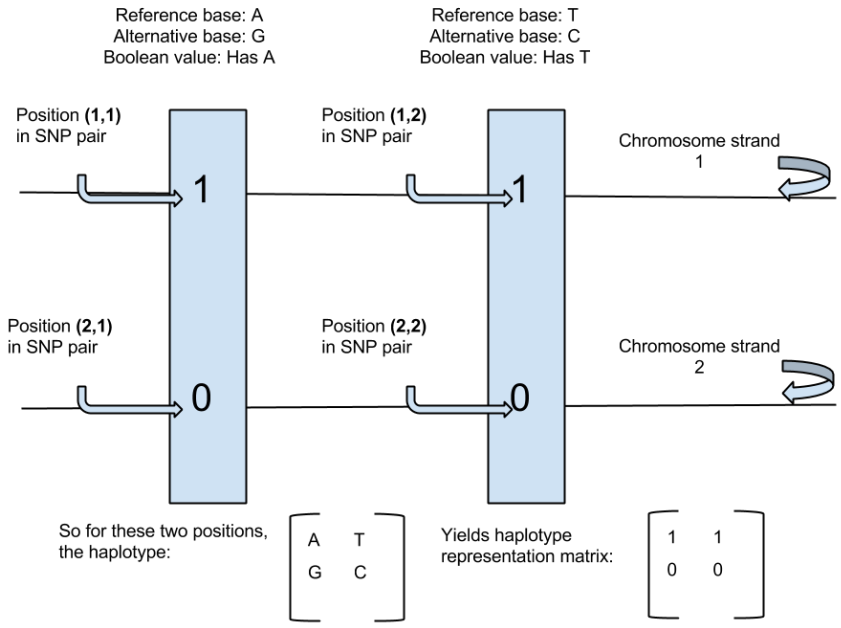

Figure 4: A boolean representation of a pair of SNPs in an individual. The left box represents one SNP and the right one another SNP. Note that the '1' in the left SNP does not represent the same letter as the '1' in the right SNP. Generally, the '1' stands for having the reference base on the discussed position, and oppositely the '0' stands for having the alternative base of the discussed position. The upper black line represents one strand and the lower black line the other strand.

Let's start by splitting the 16 cases up into three groups:

- Both SNPs are hetero sites. (See table 3).
- Both SNPs are homo sites. (See table 4).
- One SNP is a homo site and the other one is a hetero site. (See table 5).

Let's introduce the writing I for a homo site and X for a heterosite. We then get four cases of two homo sites (II), four cases of both sites being hetero (XX), and eight cases of one site being homo and the other hetero (IX and XI):

XX_1	XX_2	XX_3	XX_4
1 1	1 0	0 1	0 0
0 0	0 1	1 0	1 1

Table 3: Four cases of both SNPs being hetero sites. We say that the individual is XX.

II_1	II_2	II_3	II_4
0 0	0 1	1 0	1 1
0 0	0 1	1 0	1 1

Table 4: Four cases of both SNPs being homo sites We say that the individual is II.

IX_1	IX_2	IX_3	IX_4	XI_1	XI_2	XI_3	XI_4
0 1	0 0	1 1	1 0	1 0	1 1	0 0	0 1
0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1

Table 5: Eight cases of one SNP being a homo site and the other being a hetero site. We say that the individual is IX in the first four cases and XI in the latter four cases.

2.3 Possibilities of creating a new strand

Let's now look at what each of the 16 cases could yield as a strand to give to its child, with and without a crossover event. In figure 5 we can see how each of the four possible strand inheritance instances can come about, independent of what the strands of the parent contain.

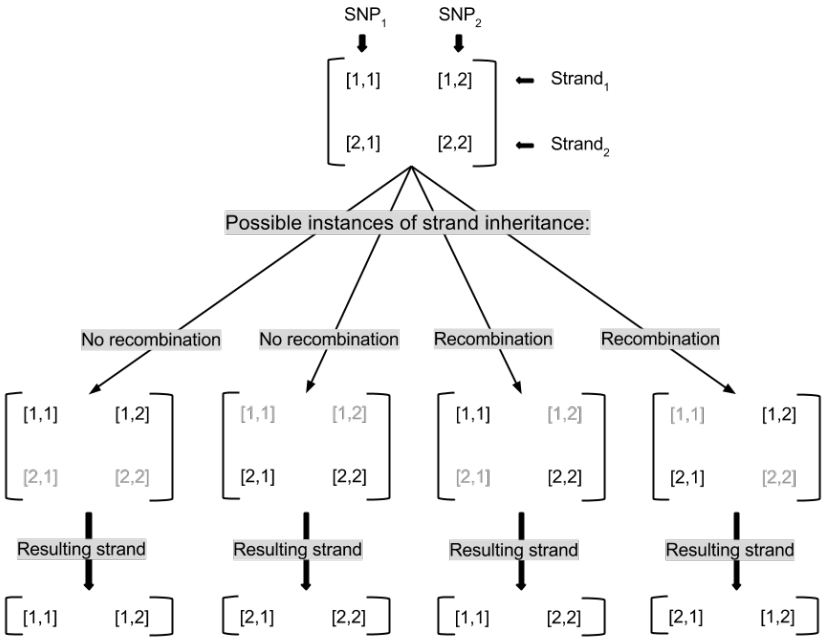

Figure 5: Diagram of how strands can be formed from a pair of phased SNPs. The small brackets represent the indices of the original base. Note that the left one represents the strand index and the right one represents the SNP index.

Base indices	XX_1	XX_2	XX_3	XX_4	Crossover	Crossover seen
$(1,1)$ and $(1,2)$	1 1	1 0	0 1	0 0	No	-
$(2,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	No	-
$(1,1)$ and $(2,2)$	1 0	1 1	0 0	0 1	Yes	**Yes**
$(2,1)$ and $(1,2)$	0 1	0 0	1 1	1 0	Yes	**Yes**

Table 6: Possible strand inheritance instances of cases XX_i. They appear in different ways for all four inheritance cases and make up all possible combinations of 0 and 1.

Base indices	II_1	II_2	II_3	II_4	Crossover	Crossover seen
$(1,1)$ and $(1,2)$	0 0	0 1	1 0	1 1	No	-
$(2,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	No	-
$(1,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	Yes	No
$(2,1)$ and $(1,2)$	0 0	0 1	1 0	1 1	Yes	No

Table 7: Possible strand inheritance instances of cases II_i. They appear in only one way since the strands of the individual creating the new strand are identical.

Base indices	IX_1	IX_2	IX_3	IX_4	Crossover	Crossover seen
$(1,1)$ and $(1,2)$	0 1	0 0	1 1	1 0	No	-
$(2,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	No	-
$(1,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	Yes	No
$(2,1)$ and $(1,2)$	0 1	0 0	1 1	1 0	Yes	No

Table 8: Possible strand inheritance instances of cases IX_i. They appear in two ways; identical to the two strands they can be created from.

Base indices	XI_1	XI_2	XI_3	XI_4	Crossover	Crossover seen
$(1,1)$ and $(1,2)$	1 0	1 1	0 0	0 1	No	-
$(2,1)$ and $(2,2)$	0 0	0 1	1 0	1 1	No	-
$(1,1)$ and $(2,2)$	1 0	1 1	0 0	0 1	Yes	No
$(2,1)$ and $(1,2)$	0 0	0 1	1 0	1 1	Yes	No

Table 9: Possible strand inheritance instances of cases XI_i. They appear in two ways; identical to the two strands they can be created from.

In tables 6-9, we confirm that to be able to see a crossover event we need the

parent to have two hetero sites (XX), which is covered in table 6, while tables 7-9 show that if the parent has two homo sites (II), or one homo site with one hetero site (IX and XI), we will not be able to see the crossover event. The hidden crossovers are because they appear exactly like the two possible non-crossover events, and in those cases we will have to assume that the crossover did not occur, since a crossover is far less likely than a non-crossover.

2.4 Possibilities for seeing crossovers

Since we have deducted that to see a crossover we need the parent to have two hetero sites, we see that we have three remaining cases of two parents uniting and yielding two strands in a child, on which we can see that there was a crossover (see table 10).

	Parent 1	Parent 2
Parent combination 1	XX	II
Parent combination 2	XX	XX
Parent combination 3	XX	XI or IX

Table 10: Parent type combinations, for which we have a possibility of seeing crossover events, in some sub-cases.

We can now also show that a parent having two hetero sites (XX) is not sufficient for seeing a crossover event, because it depends on what the other parent has.

2.4.1 Parent 1 is XX and parent 2 is II

Always informative for parent 1:

Let's look at tables 6 and 7. First, in table 7, we see that when parent 2 has two homo sites (II), then all strand yielding possibilities look the same. This means there should not be a doubt about which strand in the child belongs to parent 2 (assuming no mutation), but we will always have to assume a non-crossover. This leads to the fact that we can establish which case of strand formation we have from the other parent (displayed in table 6.)

2.4.2 Both parents are XX

Informativeness depends on parent sub-type combination:

Let us categorize the XX_is into horizontally symmetric and asymmetric ones, putting XX_1 together with XX_4, and XX_2 together with XX_3.

Generally, we have three *event-cases*: (1) Crossover in neither parent, (2) crossover in one parent, and (3) crossover in both parents.

Not mixing symmetric and asymmetric:

When we do not mix the symmetric category with the asymmetric one, we will always the crossovers, because they look the same for both XX_i within the category. If we only see one crossover, we cannot assign it to a parent, but we will see it.

Mixing symmetric and asymmetric:

When we mix the symmetric category with the asymmetric one, i.e. we take one parent from each of the categories, there are limitations to when we can see crossovers.
One crossover:
Take XX_1 and XX_2, for example, and event-case 2 (one crossover). We will see strands, both of which look like a non-crossover from one of the parents. We then assign one of the strands to that parent, and are left with a strand which we have to assign to a crossover from the other parent. We will therefore see a crossover without being sure of which strand was the crossover.
No crossover or two crossovers:
If when then take event-cases 1 (no crossover) and 3 (two crossovers), we will see two strands. Strand 1 looks like a non-crossover from one parent A, and a crossover from parent B. Strand 2 looks like a non-crossover from one parent B, and a crossover from parent A. In this case we will always assume two non-crossover events rather than two crossover events.

2.4.3 Parent 1 is XX and parent 2 is IX or XI

Informativeness depends on parent sub-type combination:

Remember from table 6 that if the parent has two hetero sites, any combination of 0 and 1 can come out as a strand, and we will always see whether it was a crossover event. Remember also from tables 8 and 9 that if the parent has one hetero site and one homo site, only two combinations of 0 and 1 can come out as a strand, namely exactly the ones the parent has. We thus have the following subcases.

Both strand possibilities of parent 2 visible:

If both strand possibilities of parent 2 are visible in the offspring, and one of them could be a non-crossover from parent 1, then we assign the non-crossover strand to parent 1 and the other strand to parent 2, as non-crossover. We cannot say whether there was a crossover.

If both strand possibilities of parent 2 are visible in the offspring, and none of them could be a non-crossover from parent, we can see that there was a crossover from parent 1, but not on which strand in the child.

One strand possibility of parent 2 visible:

If only one of the strand possibilities of parent 2 are visible in the offspring, we assign that strand to parent 2 and the other strand to parent 1, knowing with certainty whether there was crossover in parent 1.

2.4.4 Summary

We can simplistically summarize the informative parent type combinations this way:

	Parent 1	Parent 2	Information
Parent combination 1	XX	II	Always informative for parent 1
Parent combination 2	XX	XX	Sometimes informative
Parent combination 3	XX	XI or IX	Sometimes informative for parent 1

Table 11: Summary of parent type combinations, for which we have a possibility of seeing crossover events, if they occur.

3 Methods for crossover search

The above combinatorics can be used to implement an algorithm to not only find crossovers, but keep count of the SNP pairs where we know that we would see the crossover event, were it the case. Like explained in the Combinatorics chapter, crossovers can sometimes occur without the data displaying it, i.e. when strand formations look the same for crossover events as for non-crossover events, in which case we will assume a non-crossover event, since that is more likely than a crossover-event. Implementing an algorithm that would also keep count of the parent combinations that *could* display a crossover event, could make a difference in the proportion of crossovers to the locations compaired.

3.1 Current implementation

The author's first stable implementation of crossover search, does not utilize, to the full extent, the results from the Combinatorcs chapter. It compares the pairs of the trio, assumes non-crossover if that is a possible option, and assumes

crossover if that is the only remaining solution to creating the strand of the child from the strands of the parent. It does not keep count of the parent combinations that could display a crossover event, in the purpose of getting a more accurate proportion of crossovers out of the covered region.

3.2 Pseudo code

The code's function can be simply explained in the following pseudo code:

```
for Strand in PairOfStrandsInChild do
    if Strand matches a strand in PairOfStrandsInMom then
        label Strand as PossibleMatchToMom
    if Strand matches a strand in PairOfStrandsInDad then
        label Strand as PossibleMatchToDad

    if Strand found no match in PairOfStrandsInMom then
        if Strand could be Crossover from Mom then
            label Strand as PossibleCrossoverFromMom
    if Strand found no match in PairOfStrandsInDad then
        if Strand could be Crossover from Dad then
            label Strand as PossibleCrossoverFromDad
```

Now we have mininum 0 and maximum 2 labels per strand. The label contains information on what parent it is from, and if it was created with or without crossover. We now need to assign parents to the strands.

```
if Strand_1 has one label and Strand_2 has one label then
    if the labels are not from the same parent then
        assign the labels
    else
        return error for this SNP pair

if Strand_1 has one label and Strand_2 has two labels then
    assign label of Strand_1 and appropriate Parent
    assign label of Strand_2 from the other Parent

if Strand_1 has two labels and Strand_2 has one label then
    assign label of Strand_2 and appropriate Parent
    assign label of Strand_1 from the other Parent

if both strands have two labels then
    assign Strand_1 to Mom
    assign Strand_2 to Dad
```

There is a certain bias in the very last if-statement here, that if both strands can come from both parents then we merely assign strand 1 to the mother and strand 2 to the father. We could be choosing a combination of two crossovers where we have the option of choosing two non-crossovers. This is an unlikely situation, so we won't worry too much about this for this first round of implementation, but it is quite a trivial extra step which will be fixed in the continued processing of the algorithm.

3.3 Genome coverage calculations

We define a region as covered if it is between SNP pairs. We calculate the coverage by summing over all covered regions, taking into account that some regions are covered by two SNP pairs. We assume the chromosome lengths from the reference sequence used to align the NGS reads which yielded the data we use in the crossover search. The coverage is therefore a count of bases covered. The ratio of this count to all bases is also calculated.

4 Experiments and future testing

We tested the algorithm on a orangutan trio and a chimpanzee trio, while waiting for human trios, which we are expecting. There were complications along the way:
The mother in the chimpanzee trio yielded few phased SNP pairs, resulting in the coverage being way too low a proportion of the genome, to be able to discover crossovers.
It turned out that the DNA from the mother in the orangutan trio belongs to another orangutan, due to an unfortunate mix up in the sequencing lab. The results for the orangutan trio are therefore not valid. When the experiments of the algorithm were being run, the report author did not realize that this trio was faulty, so the tests for it were still run. The orangutan trio yielded greater coverage than the chimpanzee trio, but clearly we cannot infer anything about the crossover rate, as we need to compare an actual trio for that.

4.1 Chimpanzee trio

The unity of sites called as hetero in all individuals, only reached a total of 111 SNP pairs, mainly because only 1161 hetero site pairs were found for the mother. This resulted in $2,246/(2,246+3,288,036,651) \approx 0\%$ coverage of the genome, and therefore we picked up no crossovers for the chimpanzee trio, let alone gene conversions. The coverage can be seen in table 12.

Chromosome	Covered bases	Not covered bases
chr1	63	228,333,808
chr2a	0	113,622,374
chr2b	84	247,518,394
chr3	77	202,329,878
chr4	254	193,494,838
chr5	129	182,650,968
chr6	54	172,623,827
chr7	136	161,824,450
chr8	4	143,986,465
chr9	0	137,840,987
chr10	15	133,524,364
chr11	143	133,121,391
chr12	0	134,246,214
chr13	0	115,123,233
chr14	33	106,544,905
chr15	0	99,548,318
chr16	15	89,983,814
chr17	83	82,630,359
chr18	0	76,611,499
chr19	125	63,644,868
chr20	0	61,729,293
chr21	34	32,799,076
chr22	0	49,737,984
chrx	89	156,848,055
Total	2,246	3,288,036,651
Ratio	**~0**	**~1**

Table 12: Coverage of the chimpanzee genome.

We include the event summary, as to see that the algorithm picked up a right amount of events per SNP pair. This should be

$$\text{(non-crossovers + crossovers)} = 2 * \text{SNP pairs}$$

since a SNP pair contains an event from both parents. This is confirmed in table 13. In figure 6 we can see the non-crossover events per chromosome.

Chromosome	SNP pairs	Non-crossovers	Crossovers
chr1	3	6	0
chr2B	3	6	0
chr3	3	6	0
chr4	11	22	0
chr5	4	8	0
chr6	4	8	0
chr7	11	22	0
chr8	1	2	0
chr10	2	4	0
chr11	4	8	0
chr14	1	2	0
chr16	1	2	0
chr17	2	4	0
chr19	7	14	0
chr21	1	2	0
chrX	5	10	0
Total	**111**	**222**	**0**

Table 13: Event count per trio, i.e. crossovers and non-crossovers from both parents to the offspring. Per SNP pair compared, we should have two events which are crossover or non-crossover.

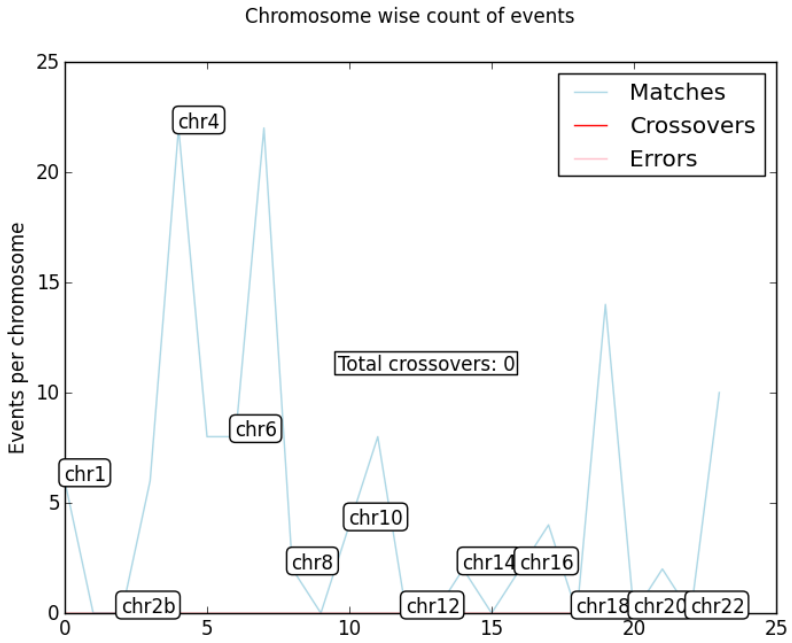

Figure 6: Chimpanzee trio: Events per chromosome.

4.2 Orangutan trio

The coverage of the orangutan trio was higher but still only 0.2% of the genome, as seen in table 14. We did not find previously published data on the genome of orangutans, but if orangutans have a similar crossover rate to humans, we should expect to find 0.2% of 60 crossovers, which is none. The algorithm turned out to find a total of 60 crossovers (table 15), all from the 'mother'. The auther manually checked a large portion of the crossovers and they stem. Clearly we can not infer anything from these crossovers, other than that they are probably there because we are comparing unrelated individuals. In figure 7 we see the distribution of non-crossovers and crossovers over the genome, and in figure 8 see see the distribution of crossovers.

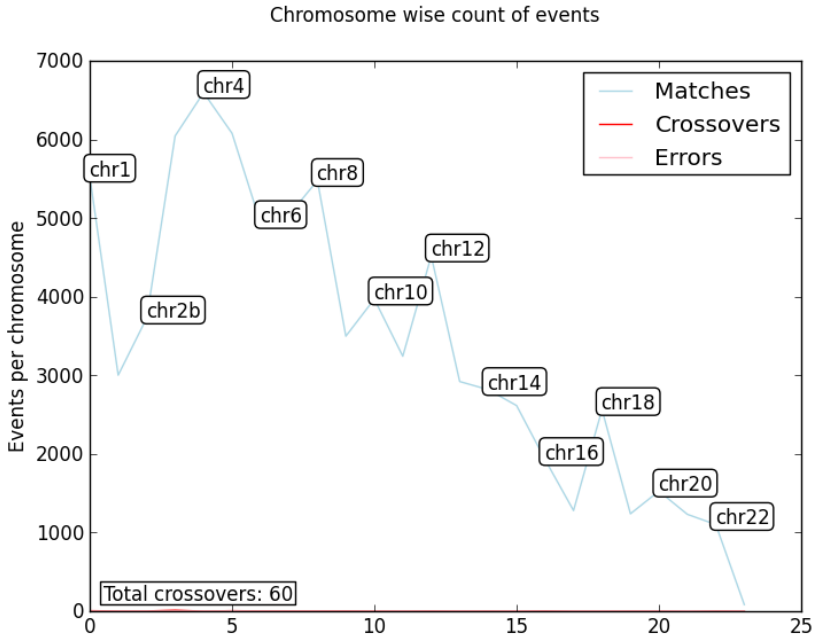

Figure 7: Orangutan trio: Events per chromosome.

Chromosome	Covered bases	Not covered bases
chr1	50,425	229,891,592
chr2a	25,806	113,002,850
chr2b	34,259	134,966,035
chr3	56,783	202,083,449
chr4	60,624	198,271,594
chr5	55,975	183,896,687
chr6	44,904	174,165,527
chr7	45,917	157,503,354
chr8	48,951	153,433,398
chr9	31,908	135,159,618
chr10	35,470	133,374,587
chr11	29,685	132,078,286
chr12	41,617	136,345,848
chr13	26,345	117,068,804
chr14	25,430	108,843,169
chr15	23,966	99,128,057
chr16	16,007	77,784,209
chr17	10,945	73,201,508
chr18	23,541	94,027,349
chr19	10,157	60,704,683
chr20	13,209	62,723,140
chr21	11,013	48,383,497
chr22	9,009	46,526,543
chrx	468 156,	194,831
Total	736,789	3,101,176,487
Ratio	**0.0002**	**0.9998**

Table 14: Coverage of the orangutan genome.

Chromosome	SNP pairs	Non-crossovers	Crossovers	Other
chr1	2,769	5,532	6	0
chr2a	1,502	3,002	2	0
chr2b	1,865	3,728	2	0
chr3	3,028	6,044	12	0
chr4	3,298	6,596	0	0
chr5	3,043	6,078	6	1
chr6	2,472	4,942	2	0
chr7	2,520	5,037	3	0
chr8	2,740	5,477	3	0
chr9	1,751	3,498	4	0
chr10	1,986	3,970	2	0
chr11	1,622	3,243	1	0
chr12	2,260	4,517	3	0
chr13	1,461	2,921	1	0
chr14	1,410	2,818	2	0
chr15	1,308	2,615	1	0
chr16	966	1,928	4	0
chr17	640	1,280	0	0
chr18	1,282	2,564	0	0
chr19	620	1,239	1	0
chr20	764	1,527	1	0
chr21	618	1,232	2	1
chr22	552	1,102	2	0
chrX	42	84	0	0
Total	**40785**	**81506**	**60**	**2**

Table 15: Event count, total and per chromosome, per trio, i.e. this counts crossovers from both parents to the offspring. Per SNP pair compared, we should have two events which are crossover or non-crossover. The events under 'Other' neither found a match or a crossover possibility, and then the trio-pair was dismissed.

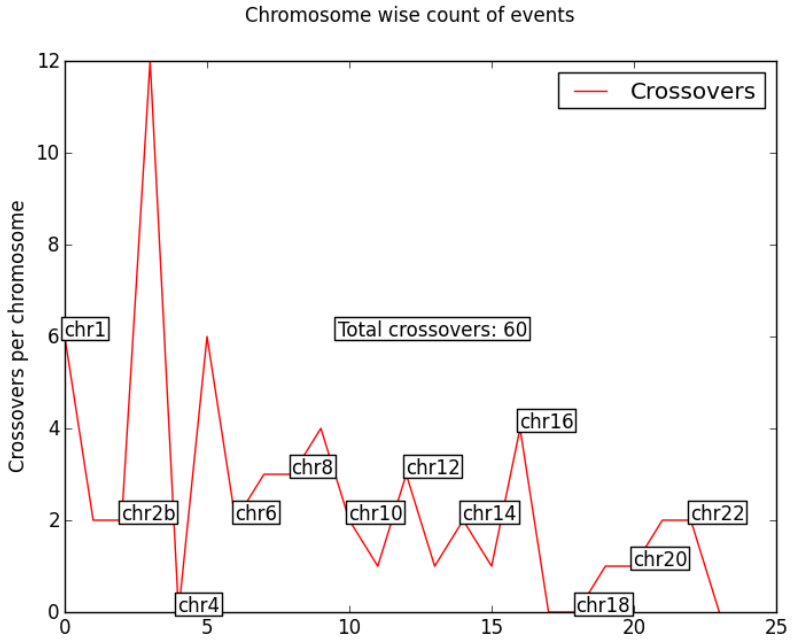

Figure 8: Orangutan trio: Crossovers per chromosome.

Assuming we had actual crossovers, the next step for us would be to find gene conversions. As we discussed in the Introduction chapter, the manifestation of gene conversion in our data is the same as two crossovers.

To see the locations of the crossovers, we made histograms of crossover events for each chromosome separately, where we binned the chromosome into regions. In the purpose of finding gene conversions, we are interested in crossovers that occur with bases of the order of a thousand or less until the next one.

Chromosome 3 (figure 9) was the chromosome that came closest to us yielding exciting results, where we saw an instance of two crossovers in the same bin. When we looked these crossovers up, we saw that they were an instance of two SNP pairs covering the same crossover. When we looked at the 12 crossovers that occured on chromosome 3, we saw that none of them occured with less than ~2 million bases until the next one. This turned out to be the case for the crossovers of the other chromosomes as well, i.e. the interval between them was to great for us to be able to assume gene conversion. We therefore found no gene conversion from these given crossovers.

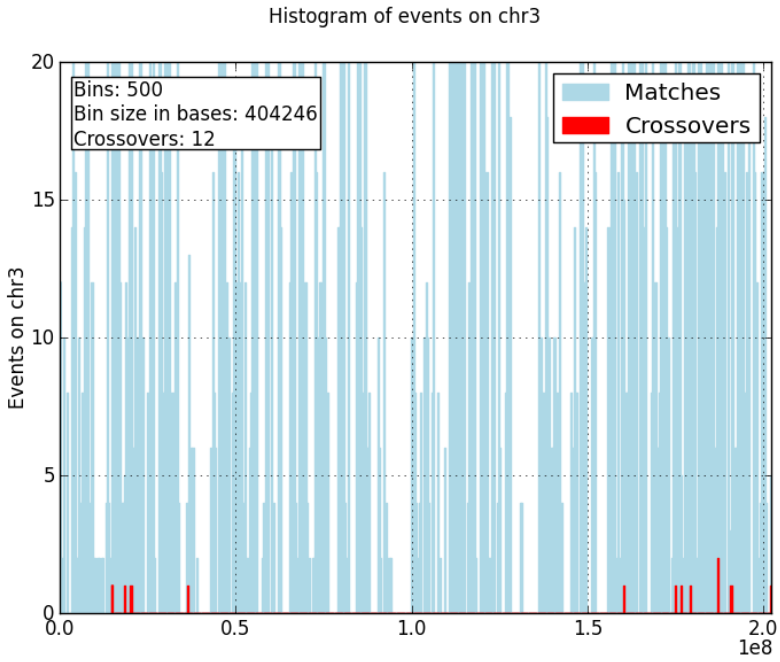

Figure 9: Histogrom of events per chromosome 3.

4.3 Future testing

We expect ten sequenced Danish human trios in the near future, which should soon grow to 50 trios. We will test the algorithm on those as the algorithm development progresses. The author has also received an invitation to test the algorithm on over 200 Icelandic trios.

5 Improvement and thoughts

5.1 Instant improvement

5.1.1 Currently only using hetero sites

The tools used to obtain the data used in the current run of results allowed us to find all hetero sites in each individual of the trio. From that we made pairs of all adjacent SNPs for each of the three individuals. We then merged the hetero site pairs, ending up with the intersection of hetero sites present in all three individuals. We are therefore are only comparing sites where all of the individuals in the trio are hetero.

In the Combinatorics chapter, we saw that the homozygosity of one parent can assist with analysing whether the other parent yielded a crossover. Instead of merging on the hetero sites, we could preprocess the SNP pairs data before we start comparing the strands, in the following manner.

When we come upon a hetero site in one of the individual, but missing data in another individual, we start by looking up if the reason for the missing data, is merely that the particular location is a homo site in the individual. That instance would namely not yield information in the first process of the tools used, but we have the data to look it up, and determine if the homozygosity is the only reason. (Other reasons could be new mutations or sequencing errors).

5.1.2 Keep count of sites where we *could* catch crossover

Like previously discussed, we should keep count of whether a crossover from the given parent strand pair combination could even be seen, were it there. To get a proper idea of crossovers, we should namely not count crossovers as a proportion of the trio-pairs we manage to compare, but we should count the crossovers as a proportion of trio-pairs we could see crossovers in, had they occured - which are not all comparable trio-pairs, as discussed in the combinatorics chapter. Without doing this, we might underestimate the amount of crossovers.

5.2 Less trivial improvement

5.2.1 Sparsity of SNPs limit ranges of local phasing

Like discussed in the complications section of the phasing chapter, having only locally phased data results in missing information. We certainly don't need a connected region of phased data to find crossovers - *if* they happen to occur on regions we manage to phase and cover. If, on the other hand, they occur on regions we have not locally phased, we will of course not see these crossovers using local phasing. In this purpose, we should consider implementing some sort of a combination of the algorithm discussed here, and perhaps the long-range-phasing algorithm, developed by Kong and Masson (2008) at deCODE genetics. A brief underdeveloped idea for this implementation mixture is using the longe range phasing method to have complete phased genomes of a trio, but then locally phase them as well, and thus somehow connect the accurate locally phased regions to increase information. This combination could yield better opportunities to discover gene conversions, since they need accurately phased data.

5.3 Can we infer a distribution in unphased regions

It is an interesting question whether or not we have any reason to infer anything about the regions where we don't have enough SNP density to be able to do analysis. There are namely crossover hotspots and mutation hotspots, and it is not straight forward to assume those are not connected. Therefore, if we, for example, assume a certain crossover frequency for the regions we are able to analyse, we cannot trivially superimpose this frequency to the regions that do not contain enough SNPs for analysis.

6 References

EMBL-EBI, and SangerInstitute. 2013. Whole Genome — Ensembl,. http://Sep2013.archive.ensembl.org/Homo_sapiens/Location/Genome.

Kong, Augustine, and Gisli Masson. 2008. Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature Genetics* 40, no. 9 (September).

Kong, Augustine, Daniel F. Gudbjartsson, Jesus Sainz, Gudrun M. Jonsdottir, and Sigurjon A. Gudjonsson. 2002. A high-resolution recombination map of the human genome. *Nature Genetics* 31, no. 241-247 (June).

NCBI, dbSNP. 2013. National Center for Biotechnology Information, dbSNP - Short Genetic Variations. http://www.ncbi.nlm.nih.gov/SNP/snp_summary.cgi.

Sasaki, Eriko, Ryuichi Sugino, and Hideki Innan. 2013. The Linkage Method: A Novel Approach for SNP Detection Haplotype Reconstruction from a Single Diploid Individual Using N Generation Sequence Data. *Molecular Biology and Evolution, Oxford Journals* 30, no. 10 (September).