Marker Development for Bitter-Tasting-Saponin Gene in Quinoa (*Chenopodium quinoa*)

Final report

Author:
Willem van Erp
930917228020
Plant Sciences - Breeding and Genetic Resources

Date and Place:
Wageningen – The Netherlands
24 October, 2016

Supervisors:
Dr. EN van Loo
Dr. Ir. LM (Luisa) Trindade
Dr. TJA (Theo) Borm

WAGENINGEN UR
For quality of life

Marker Development for Bitter-Tasting-Saponin Gene in Quinoa (Chenopodium quinoa)

Internship course

Name:	MSc Thesis Plant Breeding
Code:	PBR-80439
Credits:	39 credits
Submission date:	24 October 2016

Author

Name:	Wilhelmus Adrianus Nicolaas van Erp (W.A.N van Erp)
Registration number:	930917228020
Study program:	MSc Plant Sciences - Breeding and Genetic Resources
Phone:	+31655511759
E-mail:	willem1.vanerp@wur.nl / wan.van.erp@hotmail.com

Supervisors

Name:	dr. EN (Robert) van Loo
E-mail:	robert.vanloo@wur.nl
Phone:	+31317480879
Info:	www.wageningenur.nl/en/Persons/dr.-EN-Robert-van-Loo.htm
Name:	dr.ir. LM (Luisa) Trindade
E-mail:	luisa.trindade@wur.nl
Phone:	+31317482127
Info:	www.wageningenur.nl/en/Persons/Luisa-Trindade.htm
Name:	dr. TJA (Theo) Borm MSc
E-mail:	theo.borm@wur.nl
Phone:	+31317480877
Info:	www.wageningenur.nl/en/Persons/drs.-TJA-Theo-Borm.htm

Institute data

Name:	Laboratory of Plant Breeding - Wageningen UR
Address:	6708 PB, Wageningen

Summary

Quinoa is an economically important crop, which is consumed by both humans and animals. One of the issues with quinoa for human consumption is the production of saponins which have a bitter and soapy taste. Currently there are saponin free lines on the market, but there is no tightly linked marker (<5cM) for this trait yet. Using marker technology could speed up the breeding process of quinoa. By skim sequencing an F2-mapping population and whole genome sequencing their two parents, large amounts of read data were obtained. A k-mer analysis was done on both parental lines to validate the read data and coverage. Afterwards, the reads were mapped against the reference genome of quinoa and variants were called. Variants with a simple segregation in the mapping population – those that were single locus and polymorphic between the parents Atlas and Carina Red for which the parents were homozygous - were mapped onto a linkage map. Additionally, a homolog search was performed, in which 87 saponin related genes were blasted onto the genomic region which was associated with the non-bitter locus.

The k-mer analyses confirmed that there was enough data to support variant calling. Mapping the reads onto the reference genome of quinoa showed that the first part of the contig 3489 (<=4Mb) contained the non-bitter locus. Mapping the SNPs on the linkage map yielded a maker, 3489x00463139y16 (context 500), at 0.0 cM of the non-bitter locus. Near that locus no candidate genes directly involved in the saponin biosynthesis pathway were found.

In the aftermath of this MSc-thesis, in a collaborative research with King Abdullah University for Science and Technology (KAUST), in the region of the locus of the non-bitter trait a transcription factor (a bHlH-gene) was found that is known to regulate the terpene pathway from which saponins are made. This gene shows mutations that make the gene dysfunctional. In Atlas, apparently at least two mutations occur: one SNP that causes a severe amino acid change causing the gene and an insertion. Both cause the gene product to be non-functional. The full saponin pathway proved to be almost fully downregulated according to Jarvis *et al* (1).

Table of Contents

Acknowledgement

This master thesis was carried out for the Bio Based Economy group, Laboratory of Plant Breeding of the Wageningen University and Research Centre as part of the MSc study Plant Sciences – Plant Breeding and Genetic resources. Before working on this thesis I had hardly any experience with bio informatics, gene discovery programs and programming itself. Luckily I had the chance to work on this project and to get to know so many new techniques, I have learned a lot. Of course I could not have done this without the help of my supervisors and fellow students. Therefore, I would like to thank my supervisors, dr. EN (Robert) van Loo, dr. TJA (Theo) Borm and dr.ir. LM (Luisa) Trindade for the help with the analyses, feedback and discussion during the project. Furthermore, I would like to thank all the students and researchers in the Breeders Hollow/Hall for all the discussions and feedback.

I hope you all will enjoy reading the report

Chapter 1. Introduction

Chapter 1.1. *Chenopodium quinoa*

Quinoa (*Chenopodium quinoa*) is an allotetraploid dicotyledonous annual plant species with 18 chromosomes and is part of the *Amaranthaceae* family. The *Amaranthaceae* family includes several economically important crops such as sugar beet and spinach. Quinoa finds its origin in the Andes region of South America and has been used as a food crop for centuries by the Incas (2). To this day, quinoa is still produced in South America, with Bolivia (77,354 tonnes in 2014) and Peru (114,343 tonnes in 2014) being the main producers (3). In both countries, production and export increased over the last couple of years (Figure 1.1). The increase in caused partly by the superfood hype, but mainly by quinoa's high nutritional values and agricultural benefits (4).

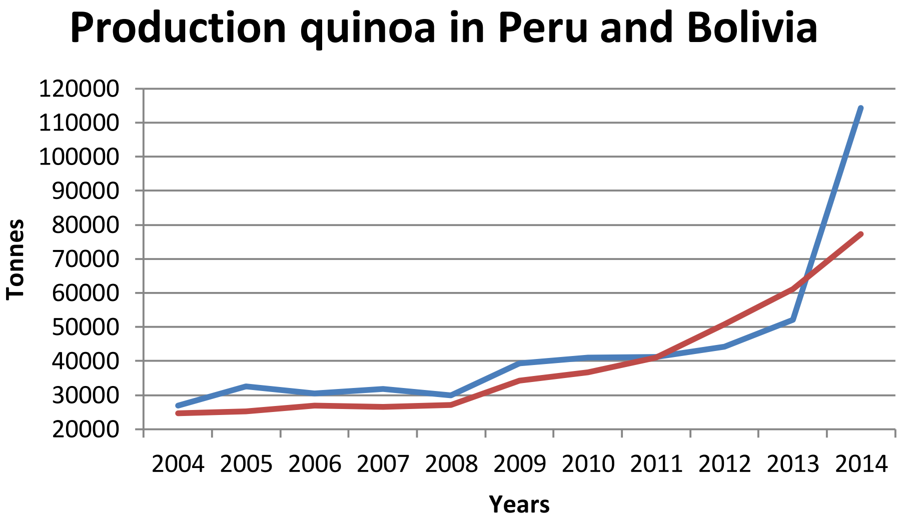

Figure 1.1: Production of quinoa in tonnes over the year 2004 to 2014 for Peru (blue) and Bolivia (red) (3).

Quinoa is gluten free, rich in iron, vitamin B1, B6 , E and has high levels of protein and fibre content compared to other well-known types of food (Figure 1.2). These favourable characteristics makes quinoa an ideal source for dietary needs (5) and is therefore used in many products, ranging from animal feed to medicines (6). Additionally, quinoa is also interesting from a farmer's perspective, due to its agricultural benefits. Benefits such as tolerances to downy mildew, nematode genus *Nacobus*, stalk rot, shelling, lodging, frost, hail and its resistance to drought (2). These favourable agricultural and nutritional traits make quinoa an interesting plant species to study.

Currently, both research and production of quinoa are increasing, to > 5,000 ha in the EU and > 250 ha in the Netherlands. In 1990 Wageningen University (Wageningen UR) started their research program on quinoa, in which Wageningen UR assessed the possibilities of cultivating quinoa in the Netherlands. From that point on, many studies followed and between 2003 and 2007 four cultivars, being Atlas, Carmen, Pasto and Riobamba, were released by Wageningen UR. Nowadays, Wageningen UR is developing varieties with an improved mildew resistance, salt tolerance, new flower colours and enhanced taste (7). This master thesis research focusses on the taste of quinoa, more specifically, on the bitter tasting saponins.

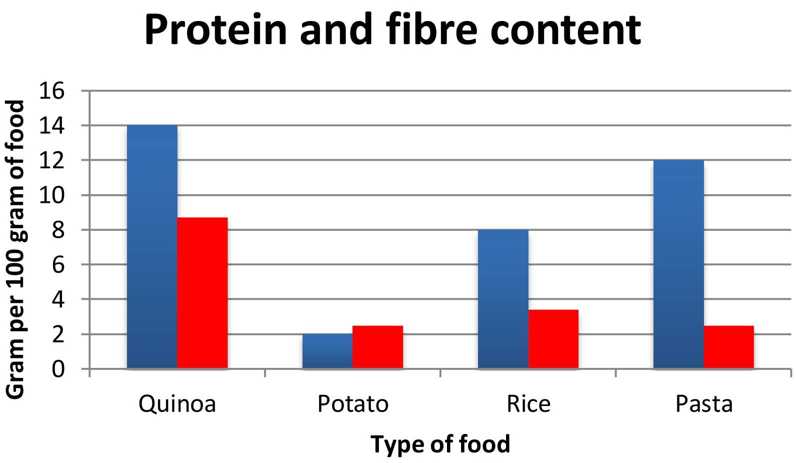

Figure 1.2: Protein and fibre content per 100 gram dry weight of quinoa, potato, rice and pasta. The blue bar indicates the protein content and the red bar indicates the fibre content (8).

Chapter 1.2. Saponin – A Bitter Taste

Triterpenoid saponins are secondary plant metabolites which are mainly found in dicotyledons (9). Until now, 20 different kind of saponins have been identified in quinoa. High levels of saponins concentration serve as a defence mechanism against pathogens and herbivores (10). Unfortunately, high concentrations of saponins cause a bitter and soapy taste, which is unfavourable upon human consumption. Concentrations of saponins vary between genotypes, with sweeter varieties ranging from 0.2 to 0.4 grams and bitter varieties ranging from 4.7 to 11.3 grams saponin per kg dry matter (11). Saponin levels are influenced by many factors, such as water and nutrient availability, light irradiation, but also depends on the plants developmental stage and architecture (11). To understand how these factors influence the saponin production and which genes are involved, one should study the biosynthesis pathway.

Currently, there is a lack of knowledge about the genes and their key roles in the saponin biosynthesis pathway in quinoa. There are many plant species which produce saponins, but still no complete biosynthesis pathway has been unravelled for one species. A concept pathway suggests that saponins are derived from the phytosterol pathway (Figure 1.3). Saponins are formed by the condensation of a 3-isopentenyl pyrophosphate, C5 (IPP) and a dimethylallyl pyrophosphate, C5 (DMAPP). Condensation is the reaction in which two molecules combine to form a larger molecule. Then, IPP and DMAPP are synthesized to from acetyl-CoA via the mevalonate pathway or from pyruvate and phosphoglyceraldehyde via the plastid-localized 2-C-methyl-d-erythritol 4-phosphate (MEP) pathway. IPP and DMAPP condense to form geranyl pyrophosphate, C10 (GGP). Then, a second IPP unit binds to GPP and a farnesyl pyrophosphate, C15 (FPP) is formed. Then a second FPP unit binds to the previous FPP to form squalene, C30. The squalene is then epoxygenated to a 2,3-oxidosqualene, C30, which is the last common precursor of the titerpenoid saponins, phytosterols and steroidal saponins (12). After the common phytosterol pathway, the multigene families of cytochromes P450 (P450s), family 1 UDP-glycosyltransferases (UGTs) and oxidosqualene cyclases (OSCs) are responsible for the formation of the specific saponins. OSCs is the largest family and therefore subdivided into: accurate (and putative) b-amyrin synthases, accurate lupeol synthases, accurate dammarenediol synthases, accurate (and putative) cycloartenol synthases, accurate (and putative) lanosterol synthases, moderate accurate OSCs and multifunctional OSCs. In literature, 77 genes are known to be involved in OSC production, four genes in P450 production and six in the UTG production (Attachment 2). The complexity of the gene families make identification of new relevant genes difficult. Expected is that there are multiple QTLs involved in the regulation of the saponin pathway of quinoa, but whether or not the pathway is active, is controlled by one single gene (12).

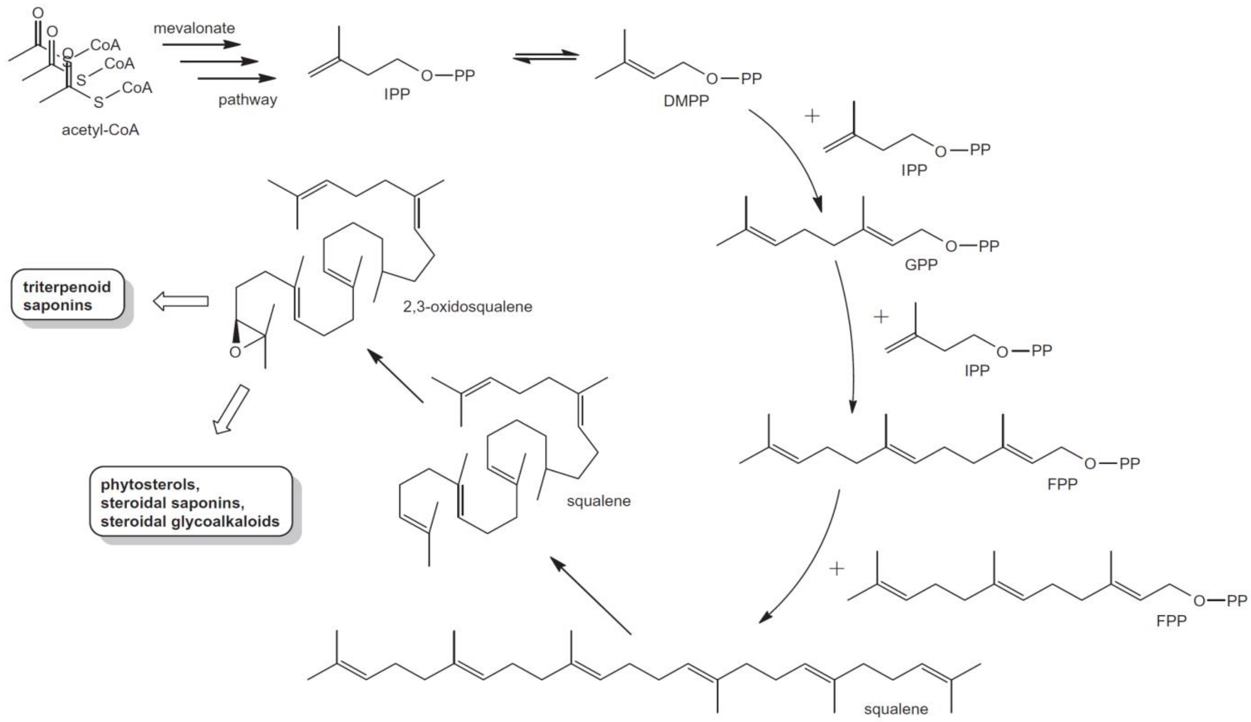

Figure 1.3: Early steps in phytosterol pathway which leads to 2,3-oxidosqualene. IPP = Isopentenyl pyrophosphate, DMPP = dimethylallyl pyrophosphate, GPP = geranyl pyrophosphate and FPP = farnesyl pyrophosphate (12).

The single dominant diploid segregating gene seems to control the whole biosynthesis pathway of saponins in quinoa, functioning like a master switch. A study by Gandarillas in 1948 showed a 3:1 segregation in bitter versus non bitter varieties and thereby confirming the presence of a dominant allele (12). Furthermore, research by Ward showed that two recessive alleles on the bitter saponin locus could inhibit the saponin synthesis in two Bolivian quinoa genotypes (14). Also known is that there is no significant difference in the level of saponin between plant which are heterozygous and homozygous for the single dominant gene (14). The position of this saponin regulatory gene on the genome is not known yet. There is a marker for the gene, but the marker is not tightly linked (<5 cM) at 9.4 cM from the non-bitter locus (14). Therefore, more genetic research has to be conducted to pinpoint the locus associated with the non-bitter phenotype and find the causal allele(s).

Chapter 1.3. Aim of Research

Quinoa is an economically important crop, which is consumed by both human and animals. Breeders are challenged to improve quinoa as a crop, to secure the world's supply of quinoa and therefore research and breeding are required. Traditional breeding of quinoa is a time consuming and costly process. Therefore, breeding programs are often assisted by molecular techniques, such as marker-assisted-selection (MAS). MAS is objective and selection can be done with a high reliability compared to phenotypic screening. Also, screening can be done at seedling stage, which is especially important in plant species with a long generation time. Unfortunately, no marker has been tightly linked to the dominant gene (locus) controlling the saponin biosynthesis pathway yet (<5cM). Therefore the objective of this research is to identify candidate contigs/regions and markers closely associated with this switch for production of bitter tasting saponins in quinoa (*Chenopodium quinoa).*

To identify the non-bitter locus, skim-sequencing of 94 F2 quinoa lines obtained from a cross between a bitter and a non-bitter quinoa line in a bulk segregant analyses (BSA) can be used to quickly discover trait-associated sequence variants. Using the available reference genome the sequence context of these variants can be retrieved. As the 94 lines were skim-sequenced individually and not as a pool, imputation can be used to infer genetic variant segregation patterns and construct a genetic map ab-initio as well.

Furthermore, a k-mer analyses will be performed which will give an insight on the DNA composition of quinoa, such as genome size and its tetraploid characteristics (15). Also 87 genes, involved in the saponin biosynthesis pathway which are summarized by Augustin et al, 2011, will be blasted against the candidate contigs from the reference genome. Using this strategy, homologs of OCSs, p450s or UTGs from other plant species could be identified in quinoa and possibly connected to SNPs found in the bulk segregant analyses.

Chapter 2. Materials and Methods

2.1 Description of the data - Whole Genome Sequencing (WGS)

In total 96 plants were sequenced, the non-bitter parental line Atlas, bitter parental line Carina red and 94 non-bitter F2 plants of the Atlas x Carina red BSA population (Figure 2.1). A cross was made between Atlas and Carina Red, the F1 was selfed and a 3:1 ratio was found for the non-bitter trait. Of these 800 plants 200 plants were non-bitter and 94 lines were selected for WGS. Phenoptying the F2 population for bitter saponins was done by a foam layer analyses (16). The selected F2 plants were grown in the field located at the Grebendijk, South-West of Wageningen and were regularly checked for disease and watered if needed. From the adult plants, samples were taken of the youngest leaves and stored in liquid nitrogen to prevent the DNA from degrading. During isolation about 300 mg of leaf material was used to create the DNA stock solution used for WGS.

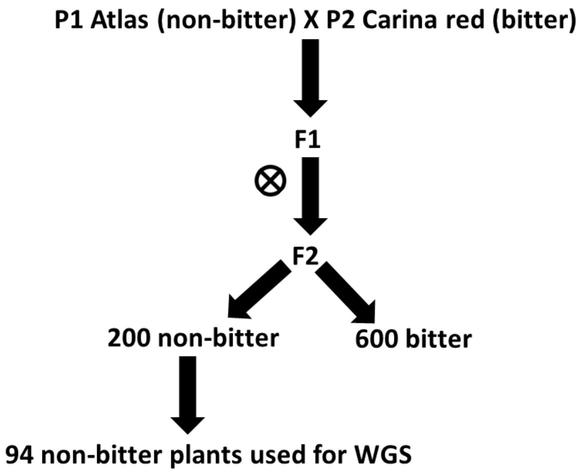
P1 Atlas (non-bitter) X P2 Carina red (bitter)
F1
⊗
F2
200 non-bitter
600 bitter
94 non-bitter plants used for WGS

Figure 2.1: Crossing scheme for the F2 BSA population derived from a cross between Atlas and Carina red.

The Illumina HiSeq 2500 System was used for whole genome sequencing (WGS), generating paired end reads of 2 x 125 bases, with a target coverage of 40x for both parental lines and an average 0.6x per F2 progeny genotype. The analyses described below were performed on the calculation cluster of the Wageningen UR Plant Breeding.

2.2 Analyses of Sequence Data – K-mer Analyses

A k-mer analyses was performed on merged read data consisting of the forward and reverse reads of one parental line, creating merged files for both Atlas and Carina Red. Jellyfish (Version 2.2.3) was used for partitioning the 125 base pair reads into k-mers with a length of 31 base pairs. Jellyfish is a fast memory efficient counting tool for k-mers in DNA sequence (17). The count command in Jellyfish counts the k-mer occurrences and outputs these in a mer_counts.jf file by default. Note that the k-mer and other analyses were submitted to the computer cluster via the Sun Grid Engine (SGE) job scheduler.

```
#!/bin/bash
#$ -q stat.short
#$ -cwd
Jellyfish count -m 31 -s 1500M -t 2 -C merged_file_Atlas.fast
```

-m = k-mer length
-s = amount of elements in the hash
-t = the amount of processors which will be used
-C = jellyfish will count canonical

The output file mer_counts.jf was converted to a .txt file with the histo command and then used to compute a k-mer histogram.

```
Jellyfish histo mer_counts.jf
```

Based on the k-mer occurrences, tetraploid peaks of both parental lines were estimated by plotting the curve with a normal distribution using the solver option in Microsoft Excel. Furthermore, the genome size was estimated based on the total number of putative error free k-mers divided by the multiplicity of the second peak (maximum coverage depth). The amount of putative error free k-mers was calculated by subtracting the putative error k-mers from the total amount of k-mers. Putative error k-mers occur at a frequency which is below the expectation of three k-mer occurrences.

2.3 Analyses of Sequence Data – Mapping Reads, Variant Calling and Linkage Maps

Pre-processing

The raw zipped data from the sequencing provider was unzipped and prepared for mapping. The sequence data consisted of the reads of the two parental lines Atlas and Carina red and of the F2 population which consisted out of 94 lines. Furthermore, no data trimming was done before mapping, because the sequenced samples were derived from DNA and not RNA samples. Illumina sequencing of DNA samples is less prone to random errors compared to RNA samples.

Mapping

Before mapping, the reference genome was indexed using the BWA index command (Attachment 1: Indexing reference genome). The reference genome (version 3.1, consisting of 3480 contigs) was made available by the King Abdullah University of Science & Technology (KAUST) in Saudi Arabia. The mapping of the parental lines and the 94 F2 plants was done by Burrows-Wheeler Aligner – MEM (BWA-MEM Version 0.7) (Attachment 1: Mapping of Atlas and Carina red reads and Mapping of BSA population reads)[18] . BWA outputs a .SAM file which was converted to a coordinate sorted .BAM file by Picard Tools – SortSAM (SAMtools Version 0.1.18) . The .SAM files were removed after sorting to save disk space. The last step in the mapping process was to check if the mapping was done correctly by generating statistics on the .BAM file with FlagStat (SAMtools Version 0.1.18) (18). Next the variants were called.

Variant discovery

Sites which displayed variation to the reference genome were called. Unfortunately not all variants were true SNPs, because of mapping errors and sequencing artefacts. Therefore, to bring balance between sensitivity and specificity the variant discovery was divided into two steps, namely variant calling and variant filtering.

Before variant calling, CreateSequenceDictionary.jar (picard-tools-1.90) and faidx (SAMtools Version 0.1.18) were used to create a description and index file of the reference genome (18,19). Also the reference genome was split up into 35 files each containing randomly selected 100 contigs to speed up the variant discovery done by Mpileup (SAMtools Version 0.1.18) (Attachment 1: Variant calling with Mpileup) . MpileUp produced information on chromosomal position of the called SNPs and outputs these in .vcf files. By using bcftools call (bcftools-1.2) the .vcf files were converted to .bcf files to save memory space (18).

Using a filtering-script written by Loo, 2016 (Attachment 1: Splitting VCF files) the .bcf files were split into three files: combi.out, context.out and geno.out. Combi.out would contain the genotype calls of the combinations of ./. to 3/3 for the first parent, second parent and bulk population. Context.out describes the counts of genotype scores per context and geno.out contained the genotype counts per allele. The filtering-script combined genotype counts of 50 variants and 500 variants. Adding up 50 variants or 500 variants would increase the coverage over these variants and therefore it would be easier to filter true SNPs from false SNPs. These true SNPs were than used to construct a linkage map.

Linkage map

A genetic map was constructed with JoinMap (Version 4.1). JoinMap is a high quality tool which creates genetic linkage maps in experimental populations [20]. Only markers which had sufficient coverage (4000/40,000 reads per context 50/context 500), 95% Atlas reads, a diploid segregation (1:2:1) and showed correlation of 0.98 to the up following marker (CORREL function excel) were used as input data for Joinmap.

2.3 Homologs Saponin Biosynthesis Genes in Contig of Interest

The 87 OSCs, P450s and UGTs genes described by Augustin et al., 2011 were blasted against a possible contig of interest of the reference genome of quinoa (Attachment 2). TBLASTX (BLAST+ executables version 2.3.0) was used to search for homologs. TBLASTX searches in a translated nucleotide database using a translated nucleotide query. The formatdb command created a database containing the 87 protein sequences.

```
Formatdb -i gene_1.fasta gene_2.fasta gene_3.fasta -p F -n
gene_seq_quinoa
```

-i = input FASTA files
-p = type of data file; either F for nucleotides and T for protein
-n = output file name

By default, the command formatdb created three output files: gene_seq_quinoa.nhr, gene_seq_quinoa.nsq and gene_seq_quinoa.nin. Afterwards, TBLASTX was run.

```
#!/bin/bash
#$ -q stat.short
#$ -cwd
Blastall -p tblastx -i reference_genome_quinoa.fasta -d gene_seq_quinoa -
e 1E-4 -o TBLASTX output quinoa.txt -m 8
```

-p = program to run; in this case TBLASTX
-i = input FASTA file of reference genome of quinoa
-d = subject of database created with formatdb
-evalue = the minimum e value a homolog must have to be a ‘hit’
-o = output file name
-m 8 = this option gives a tabular output file which summarizes the information for each hit

The TBLASTX hits were filtered based on the ‘Percent Identities’ and ‘E-value’. Hits with a minimum Percent Identity or average nucleotide identity (ANI) of 95% and an E-value of $1x10^{-4}$ were considered a significant match. Additionally, the sequence of the marker from Ricks, 2005 at 9.4 cM was blasted against the reference genome.

Chapter 3. Results

3.1 Analyses of Sequence Data – K-mer Analyses

In the k-mer histogram of Atlas three peaks can be distinguished (Figure 3.1). The first peak, represents the read errors. These k-mers only occur once in the WGS dataset, which, given its genome coverage of approximately 40x is unlikely. In total, 7% of all k-mers were present in this first peak. The second peak was found at k-mer frequency ~20, representing the k-mers which are unique for a certain sub genome. The third peak is located at k-mer frequency ~39 and represents the k-mers that occur in both sub genomes. The frequency of the third peak at 39 is double that of the first peak, as one would expect. Fitting normal distributions to these peaks yields that ~36.2% of the data are contained third peak. Note that the standard deviation of this fitted normal distribution was 24.2, indicating that the third peak was much wider compared to the second peak. Furthermore, a fourth peak was observed at k-mer depth 700, which are caused by contaminants. Additionally, there are 22,478,197,952 putative error free k-mers divided by the multiplicity of the second k-mer peak (20) yields a genome size of 1.14 Gb

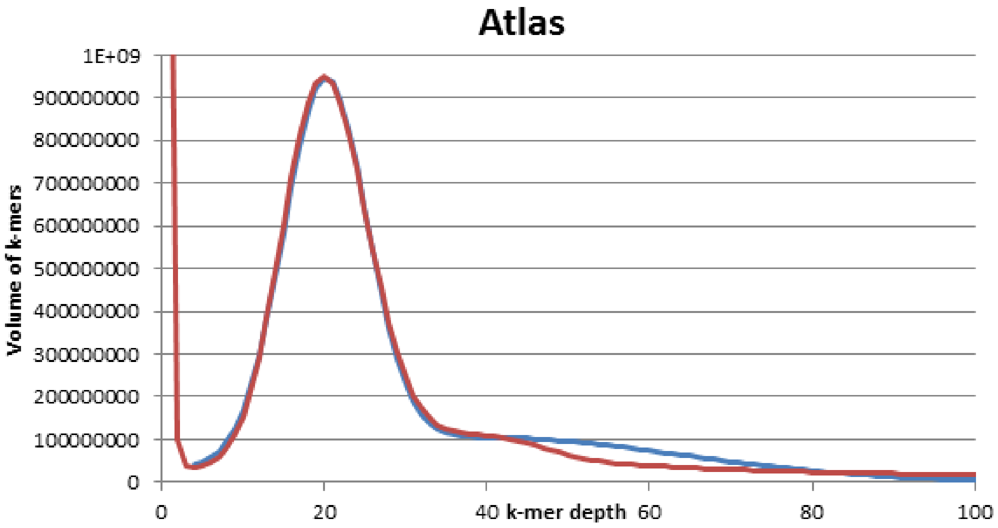

Figure 3.1: The k-mer histogram of parental line Atlas (blue). The k-mer volume is the 'amount of different k-mers' times the 'occurrence of that k-mer'. For example, if there are 20 different k-mers which each have five copies in the sequence data, the volume would be 100 k-mers. Also the fitted normal distribution is shown (red).

In the second k-mer histogram of Carina red also three peaks were observed (Figure 3.2). The first peak, same as for Atlas, contained the putative error read k-mers and was 7.9% of the total amount of k-mers. The second peak at depth 20, similar to the first histogram, represented the k-mers which are unique for a sub genome. Although the peaks were at the same depth of both parental lines, there was variation in the k-mer volume at depth 20, with Atlas having 9.5E8 k-mers and Carina red 8.1E8 k-mers. The third peak, containing k-mers obtained from genomic regions where the two sub genomes were identical (at K=31), located at a k-mer frequency of 39, represents 38.0% of the total amount of k-mers with a standard deviation of 34. Furthermore, similar to the histogram of Atlas, there was a peak at depth 700. Additionally, 26,436,352,296 putative error free k-mers which were divided by the multiplicity of about 20 to yield a genome size of 1.34 Gb.

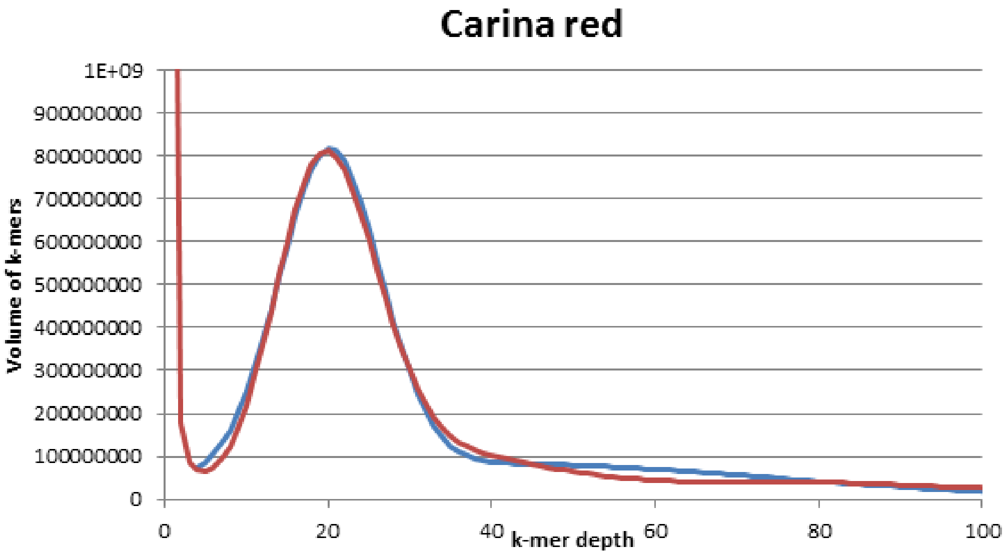

Figure 3.2: The k-mer histogram of parental line Carina red (blue). The k-mer volume is the ‘amount of different k-mers’ times the ‘occurrence of that k-mer’. For example, if there are 20 different k-mers which each have five copies in the sequence data, the volume would be 100 k-mers. Also the fitted normal distribution is shown (red).

3.2 Analyses of Sequence Data – Quality of Mapping

The Flagstat output showed that there was variation between the percentage of properly paired reads between Atlas and Carina red (Figure 3.3). Here 96.6% and 96.4% of the Atlas reads mapped onto the reference genome, compared to 84.7% and 83.7% of Carina red reads. As for the F2 plants, on average 93.9% of the reads were mapped, with genotypes 32, 43 and 55 mapping less than 80% and genotypes 47, 62 and 74 mapping over 97%. Additionally, the difference in percentage between the mapped reads and properly paired reads could be observed. There are two to five percent difference between them, indicating that two to five percent of the reads could be mapped on different places on the reference genome, for example on the other sub genome.

Although variation was observed between parental lines and F2 plants and between mapped reads and properly paired reads, the overall percentage of properly paired reads was sufficient to call variants between the genomes.

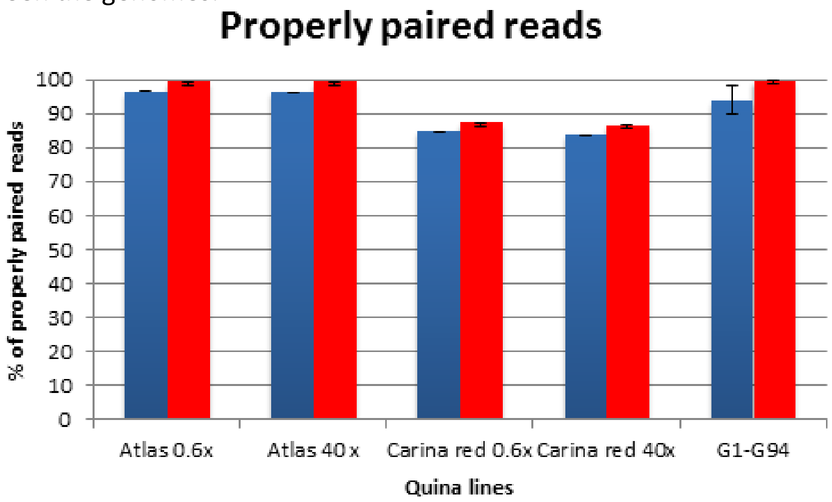

Figure 3.3: Statistics on the mapped reds (red) and percentage of properly paired reads (blue) of Atlas (0.6x and 40x), Carina red (0.6x and 40x) and an average of all the F2 plants. Included for G1-G94 is the standard deviation calculated for the average percentage of properly paired reads (4.2).

3.3 Analyses of Sequence Data – The Non-Bitter Saponin Locus

In total 1,721,100 variants were called between the 96 genomes and reference genome on 3480 contigs spread over 34,422 contexts. Of the 34,422 contexts, 70 contexts were >= 95% Atlas reads and especially contig 3489 drew attention. In total 31 contexts of contig 3489 were >= 95% Atlas reads, spanning over 3 Mb (Figure 3.4). These areas of Atlas are interesting because the non-bitter locus is inherited via the Atlas background. Furthermore, contigs 3387 and 1480 yielded eight and nine contexts with >= 95% Atlas reads. Additionally, the sequence of marker ss530859734, which was related to the non-bitter locus, had a significant BLAST hit in contig 3489.

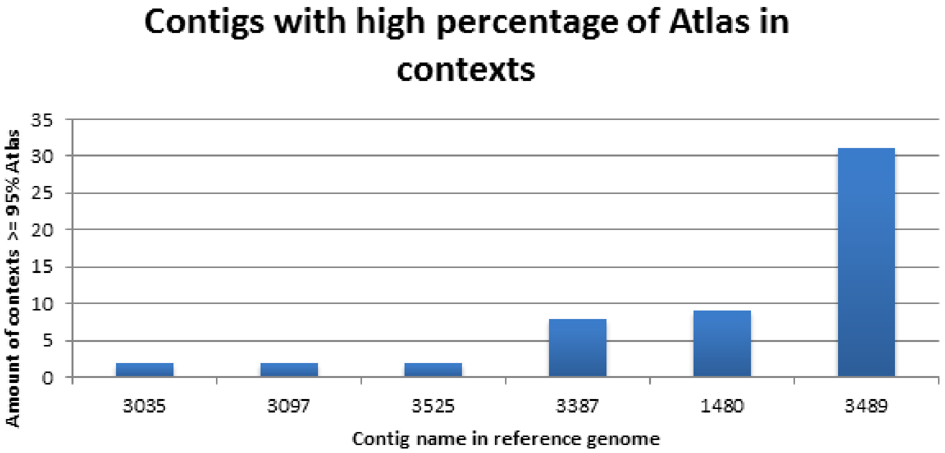

Figure 3.4: Context with >= 95% of Atlas alleles are shown in this figure, with on one side the amount of contexts per contig with show >= 95% and the corresponding contig name in the reference genome.

In a segregating population with individual accessions selected for non-bitterness, given the fact that non-bitterness is a recessive trait, the causal locus should exhibit extremely skewed segregation – with 100% Atlas allele. This contrasts with the expectation that unrelated/unlinked loci will exhibit a 1:1 ratio of Atlas:Carina red alleles approximately. Given the limited number of meiotic recombination expected in a single generation (as a rough rule of thumb ~ 1 recombination per chromosome arm per generation), this skewed 1:0 segregation is expected to decay relatively slowly, over tens of megabases to the 1:1 ratio signifying non-linkage. Figure 3.5 shows the decay in segregation skew over contig 3489, showing slow decay up until ~ 3 MB, followed by a sharp drop-off, contradicting this model.

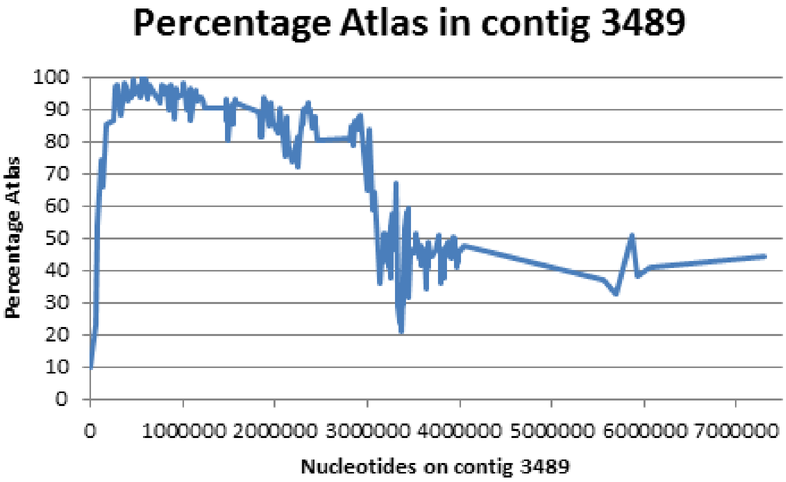

Figure 3.5: The segregation pattern of contig 3489 expressed in the percentage of Atlas alleles for each nucleotide positions on the contig.

For the construction of the linkage map, after filtering, in total 423 SNP markers were used of which 107 were mapped in the same linkage group as the non-bitter locus. In total 65 markers were mapped in front of and 40 markers after the non –bitter locus (Figure 3.6). Markers flanking the non-bitter locus are M3489.1 at 6.2 cM and M1480.1 at 6.8 cM (Table 3.1). The contigs 1480 and 3489 were also the contigs which had the highest amount of context with >=95% Atlas alleles (Figure 3.4). The closest marker to the non-bitter locus was found at 6.2 cM, which is still not in the preferred range of <5 cM. Therefore, following up the linkage map of the context 50 linkage map of context 500 was constructed.

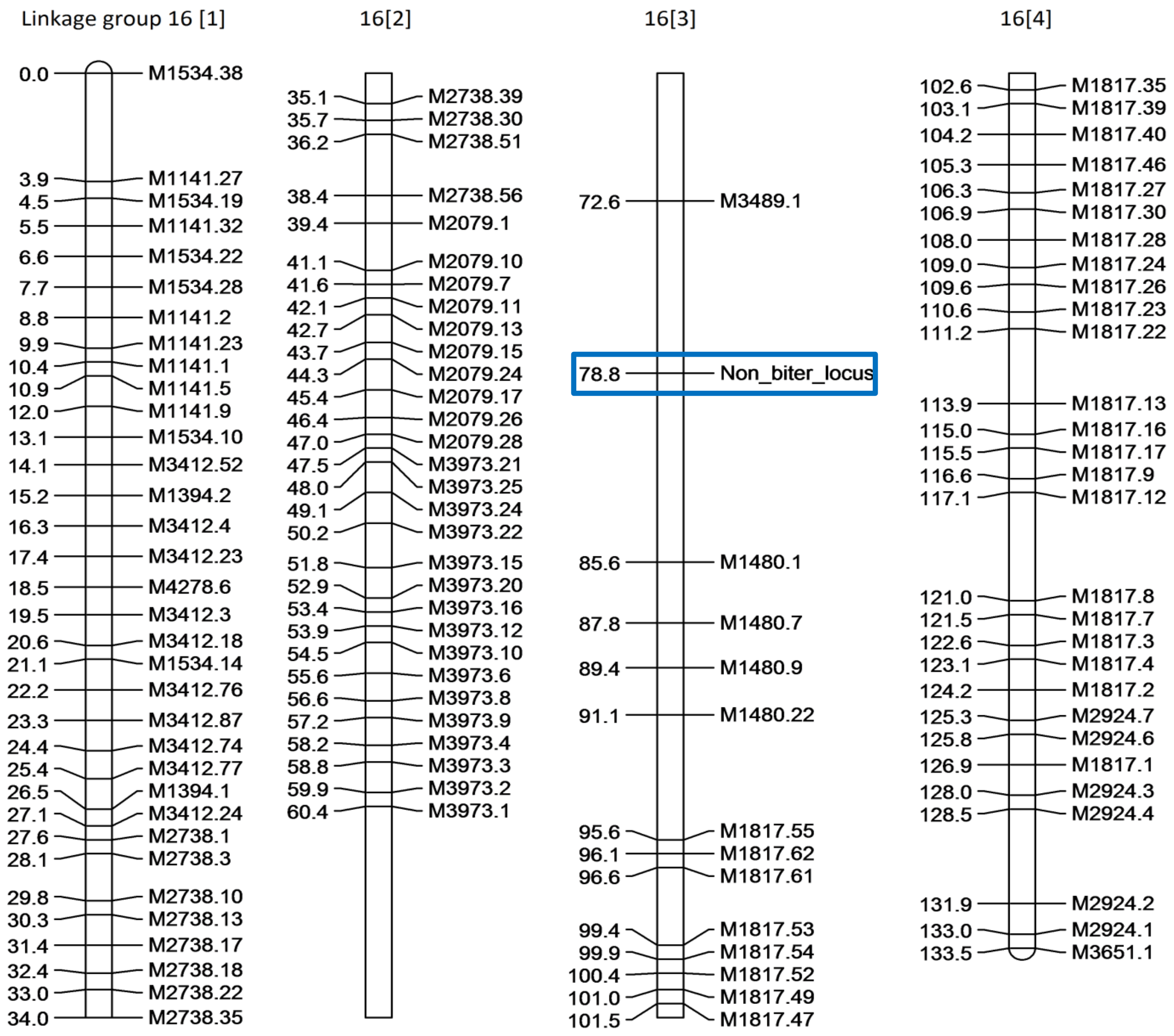

Figure 3.6: Linkage map of linkage group 16 displaying the context 50 markers. The traits of interest is shown at 77.8 cM distance and is indicated with the blue square.

Table 3.1: Markers close to the non-bitter locus from the linkage map of context 50markers.

Marker name	Contig	Starting position on reference genome	Position on map (cM)
M3489.1	3489	1,872,148	72.6
M1480.1	1480	356,463	85.6

For the context 50 markers 110 SNPS were mapped onto linkage group 16 (Figure 3.7). A total of 66 markers were mapped upstream and 31 markers downstream the non-bitter locus. The remaining markers 13 were mapped 0.0 cM from the non-bitter locus (Table 3.2). Again, a marker from contig 3489 is present, but this is a different maker from M3489.1. Marker 3489.1 starts at position 1,872,148 bp and covers the genome until 1,892,963 bp whereas marker 3489x00463139y16 starts at 463,139 bp until 571,213 bp. Flanking the bitter locus are again the regions from contig 1480 with ten markers and the 3489 regions with four markers. Additionally, comparing the context 50 and the context 500 map showed that more noise was found around the non-bitter locus in the context 500 map.

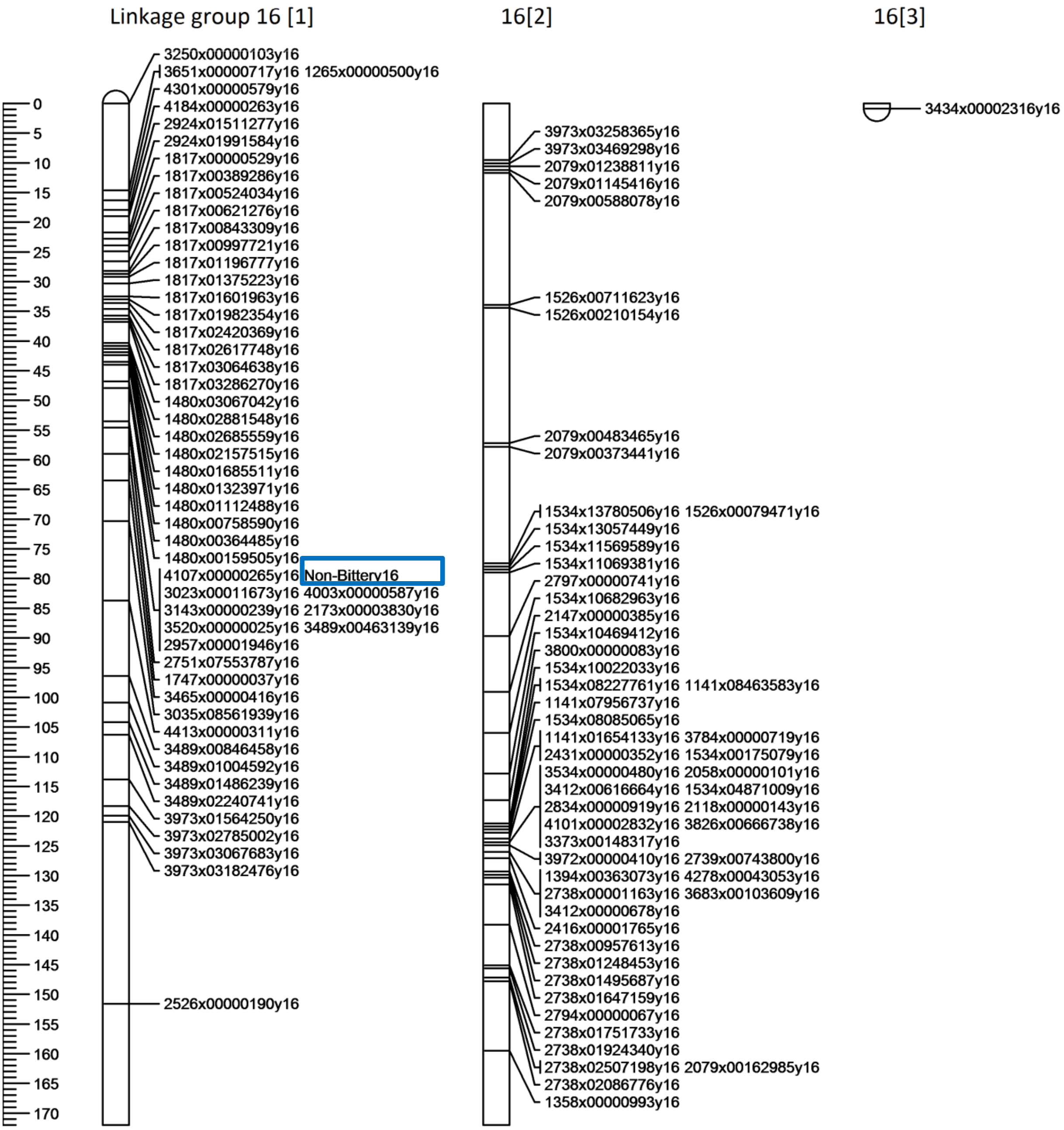

Figure 3.7: Linkage map of linkage group 16 displaying the context 500 markers. The traits of interest is shown at 53.5 cM distance and is indicated with the blue square.

Table 3.2: Markers 0.0cM from the non-bitter locus from the linkage map of context 500 markers.

Marker name	Contig	Starting position on reference genome	Position on map (cM)
4107x00000265y16	4107	265	53.5
3023x00011673y16	3023	11,673	53.5
4003x00000587y16	4003	587	53.5
3143x00000239y16	3143	239	53.5
2173x00003830y16	2173	3830	53.5
3520x00000025y16	3520	25	53.5
3489x00463139y16	3489	463,139	53.5
2957x00001946y16	2957	1946	53.5

3.4 Homologs Saponin Biosynthesis Genes in Contig of Interest 3489

No homologs were found in quinoa using BLAST with thresholds ANI >=95% and E-value 1E-4 on contig 3489. Therefore the threshold was lowered to an ANI of >= 80%. Note that when the ANI threshold is lowered the chance of false hit will increase. Eight homologs were found on contig 3489, but not all of them seem promising (Table 3.3). The aligned lengths in combination with the number of mismatched positions shows that only 126 to 129 nucleotides were covered of genes which are about 2500 nucleotides long. Therefore, on average the homolog hits only covers 5% of the original gene sequence. Furthermore, the plant species on which the hits was found were not closely related to quinoa.

Table 3.3: the homolog hits found in the reference genome of quinoa. The indicated gene length, aligned length and number of mismatched positions are in nucleotides.

Homolog hit	Gene length	Percent identities (ANI)	Aligned length	Number of mismatched positions	E-value
AB181245.1: Lotus japonicus OSC3 mRNA for lupeol synthase	2268	83.72	129	21	2.00E-17
HM623871.1: Kalanchoe daigremontiana lupeol synthase	2368	83.33	126	21	8.00E-17
AB055511.1: Betula platyphylla OSCBPW mRNA for lupeol synthase	2471	83.72	129	21	8.00E-17
FJ790411.1: Gentiana straminea beta-amyrin synthase	2268	80.95	126	24	1.00E-16
AB009030.1: Panax ginseng OSCPNY1 mRNA for beta-Amyrin Synthase	2589	80.95	126	24	1.00E-16
AB206469.1: Euphorbia tirucalli EtAS mRNA for beta-amyrin synthase	2532	80.95	126	24	2.00E-16
AB025343.1: Olea europaea OEW mRNA for lupeol synthase	2546	80.95	126	24	3.00E-16
AB116228.1: Glycyrrhiza glabra GgLUS1 mRNA for lupeol synthase	2657	80.95	126	24	4.00E-16

Chapter 4. Discussion

The objective of this research was to identify candidate contigs/regions and markers which are responsible for the production of non-bitter saponins in quinoa (*Chenopodium quinoa).*To answer this question a number of analyses have been performed on the sequence data.

The k-mer analyses is able to produce important information prior to mapping. The k-mer results confirm that there is sufficient data to call variants on and that these should exhibit mainly diploid segregation. These markers were to be used for the construction of the linkage map, later on. Additionally, the genome size was estimated and a large difference between Atlas and Carina red was observed. The difference of 0.20 Gb is caused by an under estimation of the total amount of putative error-free k-mers as seen in figure 3.1. Here a gap can be observed between the fitted line (red) and the k-mer count line (blue). On the other hand, the genome size of Carina red, 1.34 Gb, was quite similar to the genome size of the 1.35 Gb reference genome used by the genome assembler (version 3.1 KAUST 2015).

Following up the k-mer analyses, was the mapping of the reads. As described in chapter 3.2, the output of Flagstat shows a mapped reads percentage of >= 99% for Atlas and the F2 plants. Only about 87% of the Carina red reads mapped against the reference genome, and this difference is putatively due to the fact that both the reference and Atlas have a comparable Chilean background. As opposed to Carina red, which is derived from Peruvian low-land variety (1). This difference is also seen back in the percentage of properly paired reads.

After mapping and variant calling, contigs which had a high percentage of Atlas were considered candidate contigs for the non-bitter locus, in this case contig 3489. Contig 3489 being a candidate was confirmed by the mapping of the non-bitter-related-marker ss530859734 onto the first part of contig 3489 (21). Within contig 3489 a strange distortion was observed in the segregation pattern. It was later found that contig 3489 was a combination of three other contigs, 3489.A spanning from 0 to 4 Mb, 3489.B from 4 to 6 Mb and 3489.C 6 to 7.4 Mb. Therefore, the genomic region of the non-bitter locus is present in contig 3489.A. The linkage map of both the context 50 and context 500 confirmed that contig 3489.A on linkage group 16 contains the non-bitter locus.

Additionally, several genes which are known to be in the biosynthesis pathway of saponins were blasted against the 3489 contig, but without results. Even when using a more flexible threshold only a few homologs were found which only covered small stretches of the gene sequences. Due to the independent evolution of the biosynthesis pathway of saponins there has been an increase in the variation in the sequences of these genes (12). The variation in the genes would give saponin-producing plants the ability to have a wide variety of defence mechanisms. Therefore only smaller stretches of the saponin related gene sequences were recovered from the genome sequence of quinoa (12).

In future research a couple of different strategies could be used and current experiments could be extended. The k-mer analyses could be extended by using 63-mers. A larger k-mer size could help resolve differences between sub-genomes, but an increase in k-mer length will also mean an increase in memory use (17), as well as reduce the total number of k-mers available for analysis.

Furthermore, contig 3489 should be split in three parts, 3489.A to 3489.C. Additionally, during sequencing not all the data was received in one data set. Therefore there is still data left which could be mapped against the reference genome to increase coverage. An increased amount of data would especially interesting for contig 3489, once the extra data is mapped, a smaller genomic region could be assigned to the non-bitter locus.

At this point a marker has been identified at 0.0 cM from the non-bitter locus. Marker 3489x00463139y16 is derived from a set of 500 consecutive variants which were added up to increase the statistic power. To precisely map the gene onto a genetic map, fine mapping has to be done. Fine mapping the non-bitter gene would involve the sequencing of the genomic region of contig 3489 in an increased number of F2 plants. Thereby increasing the amount of recombination and so to identify SNP markers between every gene. With this strategy one could confirm the marker within a population of bitter and non-bitter lines.

As for the homolog search, although it seems difficult to find homologs in quinoa, alternative blasting strategies could be used. One could take the starting positon of a homolog and go 1000 nucleotides up and downstream the 3489 contig and blast this sequence against the NCBI database. Possible homologs could be found in other plants species from the *Amaranthaceae* family, such as Spinach (*Spinacia oleracea*) or Sugar beet (*Beta vulgaris*).

After this MSc-thesis work, the collaborative work on the genome sequence with KAUST led to the discovery of a candidate gene at the non-bitter locus: a bHlH transcription factor that is known to regulate gene expression at the start of the saponin pathway. All genes in this pathway were shown to be down-regulated in a second mapping population in the group of non-bitter genotypes while the whole pathway was normally in the developing seeds of the bitter genotypes (Jarvis et al. (joint paper of KAUST, Bringham Young University and Wageningen UR, submitted to Nature). An SNP in the KAUST/BYU population was found in the non-bitter genotypes that causes the gene product to be non-functional and two variants were found in Atlas, the non-bitter parent of the Wageningen UR mapping population (the same SNP and also a second mutation which was shown to be an insertion also rendering the gene non-functional).

Chapter 5. Conclusion

The objective of this research is to identify candidate contigs/regions and markers which are responsible for the production of non-bitter saponins in quinoa (*Chenopodium quinoa).* Using the k-mer analyses confirmed the allotetraploid character of quinoa and it was estimated that there was enough data to support variant calling. Mapping the reads onto the reference genome of quinoa yielded that contig 3489 contains the non-bitter locus, more specifically, the first part of contig 3489 (<=4Mb). From the context 500 marker map a co-segregating marker (3489x00463139y16) was found 0.0 cM distance from the non-bitter trait. Lastly, there were no homologs found on contig 3489 which are involved in the saponin biosynthesis pathway. This work was carried out in collaboration with KAUST using a yet unpublished PacBio/Bionano/Dovetail genome assembly. On the basis of this work, further work revealed that a bHlH-gene (known to regulate the full saponin pathway in other species) at the non-bitter locus is the probably candidate gene that has mutations causing the gene or gene product to be non-functional.

References

1. Jarvis et al. Unpublished genomic data Quinoa. 2016;

2. Kenwright P. Breeding the Andean grain crop quinoa (*Chenopodium quinoa*) for cultivation in Britain. PhD thesis University of Cambridge. 1989.

3. FAO STAT. Production Data Quinoa 2004-2014 [Internet]. 2014. Available from: http://faostat3.fao.org/browse/Q/*/E

4. Hudson L. Quinoa Prices Fall (Finally) Due to Rise in Production. 2015; Available from: http://spendmatters.com/2015/04/27/quinoa-prices-fall-finally-due-to-rise-in-production/

5. Repo-Carrasco R, Espinoza C, Jacobsen S-E. Nutritional value and use of the andean crops quinoa (*Chenopodium quinoa*) and kañiwa (*Chenopodium pallidicaule*). Food Rev Int. 2003;19(1–2):179–89.

6. Vega-Gálvez A, Miranda M, Vergara J, Uribe E, Puente L, Martínez EA. Nutrition facts and functional potential of quinoa (*Chenopodium quinoa* willd.), an ancient Andean grain: A review. J Sci Food Agric. 2010;90(15):2541–7.

7. WUR. Quinoa cultivation in the Netherlands [Internet]. 2014. Available from: https://www.wageningenur.nl/en/article/Quinoa-cultivation-in-the-Netherlands.htm

8. D.Q Group. Onze quinoa [Internet]. 2014. Available from: http://www.dqg.nl/onze-quinoa/

9. Sawai S, Saito K. Triterpenoid biosynthesis and engineering in plants. Front Plant Sci [Internet]. 2011;2(JUN). Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84892719565&partnerID=40&md5=7781ce2adcfe7dab16d93ecc141ce5e6

10. Sparg SG, Light ME, Van Staden J. Biological activities and distribution of plant saponins. J Ethnopharmacol. 2004;94(2–3):219–43.

11. Mastebroek HD, Limburg H, Gilles T, Marvin HJP. Occurrence of sapogenins in leaves and seeds of quinoa (Chenopodium quinoa Willd). J Sci Food Agric. 2000;80(1):152–6.

12. Augustin JM, Kuzina V, Andersen SB, Bak S. Molecular activities, biosynthesis and evolution of triterpenoid saponins. Phytochemistry. 2011;72(6):435–57.

13. Augustin JM, Kuzina V, Andersen SB, Bak S. Molecular activities, biosynthesis and evolution of triterpenoid saponins. Phytochemistry. 2011;72(6):435–57.

14. Ward SM. A recessive allele inhibiting saponin synthesis in two lines of Bolivian quinoa (chenopodium quinoa Willd.). J Hered. 2001;92(1):83–6.

15. Bhargava A, Shukla S, Ohri D. Evaluation of foliage yield and leaf quality traits in *Chenopodium* spp. in multiyear trials. Euphytica. 2007;153(1–2):199–213.

16. van Raamsdonk LW., Pinckaers V, Ossenkopple J, Houben R, Lotgering M, Groot M J. Quality assessments of untreated and washed Quinoa (*Chenopodium quinoa*) seeds based on histological and foaming capacity investigations. Microscopy: Science, Technology, Applications and Education. 2010;

17. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics. 2011;27(6):764–70.

18. Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics. 2010;26(5):589–95.

19. Broad Institute. Picard Tools [Internet]. 2015. Available from: http://broadinstitute.github.io/picard/

20. Liu BS Y, Yuan J, Hu X, Zhang H, Li Z, Chen Y, et al. Estimation of genomic characteristics by analyzing k-mer frequency in de novo genome projects. Available from: https://arxiv.org/ftp/arxiv/papers/1308/1308.2012.pdf

21. Ricks MD. Genetic Mapping of the Bitter Saponin Production Locus (BSP Locus) in (*chenopodium quinoa* Willd.). Brigh Young Univ Dep Plant Anim Sci.

Attachment 1: Scripts

Indexing reference genome

```
#!/bin/bash
#$ -q stat.short
#$ -cwd

bwa index -a bwtsw -p polished_assembly_V2.1 polished_assembly_V3.1.fasta
```

Mapping of Atlas and Carina red reads (submit script)

```
#!/bin/bash
#$ -cwd

for dn in `ls /media/scratchpad_01/erp016/seq_dataV2/run0183IL`
do

echo $dn

qsub /media/scratchpad_01/erp016/qsub_scra_01/mapping/A_CR_bwa_map.sh
/media/scratchpad_01/erp016/seq_dataV2/run0183IL $dn

done
```

Mapping of Atlas and Carina red reads (A_CR_bwa_map.sh)

```
#!/bin/bash
#$ -q stat.long
#$ -pe maxslots_patty 8
#$ -cwd

# A_CR_bwa_map.sh basisdir_fq_files_en_output $dn
# $1 = basisdir_fq_files_en_output
#    nu: /media/scratchpad_01/erp016/seq_dataV2/run0183IL
# $2 = $dn (is de subdirectory die in de submitloopfile $dn wordt genoemd)

fqdir=$1
subdir=$2

echo A_CR_bwa_map $fqdir $subdir

#genome must be indexed upfront

genomedr=/media/scratchpad_01/erp016/seq_dataV3/VER31
genome=$genomedr/CQ31
bamoutdr=/media/scratchpad_01/erp016/seq_dataV3/BAM_output
samoutdr=/media/scratchpad_01/erp016/seq_dataV3/SAM_output
bwa0712dr=/media/bulk_01/programs/bwa-0.7.12

cd $fqdir/$subdir/

for fn1 in `ls *_R1_001.nophix.fastq`
```

```
do
          fn2=$[fn1%_R1_001.nophix.fastq[_R2_001.nophix.fastq
          sambn=${fn1%_R1_001.nophix.fastq}

echo fqdir/subdir $fqdir/$subdir
echo fn1 $fn1
echo fn2 $fn2
echo samoutdr/sambn $samoutdr/$sambn
echo genome $genome

          $bwa0712dr/bwa mem \
            -M -t 8 \
            $genome $fn1 $fn2 > $samoutdr/$sambn.sam

     java -Xmx4g -Djava.io.tmpdir=/tmp -jar \
            /media/bulk_01/programs/picard-tools-1.96/SortSam.jar \
            SORT_ORDER=coordinate \
            INPUT=$samoutdr/$sambn.sam \
            OUTPUT=$bamoutdr/$sambn.bam \
            VALIDATION_STRINGENCY=LENIENT \
            CREATE_INDEX=true

samtools flagstat $bamoutdr/$sambn.bam > $bamoutdr/$sambn.flagstat.txt

done
```

Mapping of BSA population reads (submit script)

```
#!/bin/bash
#$ -cwd

for dn in `ls /media/scratchpad_01/erp016/seq_dataV3/run0182IL`
do

echo $dn

qsub /media/scratchpad_01/erp016/qsub_scra_01/mapping/bwa_97_V3.sh
/media/scratchpad_01/erp016/seq_dataV3/run0182IL $dn

done
```

Mapping of BSA population reads (bwa_97_V3.sh)

```
#!/bin/bash
#$ -q stat.short
#$ -cwd                    # change to current working directory

# bwa_97.sh basisdir_fq_files_en_output $dn
# $1 = basisdir_fq_files_en_output
#    nu: /media/scratchpad_01/erp016/seq_data/Quinoa-96
# $2 = $dn (is de subdirectory die in de submitloopfile $dn wordt genoemd)
```

```
fqdir=/media/scratchpad_01/erp016/seq_data/Quinoa-96
subdir=BSA_merge

echo bwa_97 $fqdir $subdir

#genome must be indexed upfront

genomedr=/media/scratchpad_01/erp016/seq_data
genome=$genomedr/polished_assembly
bamoutdr=/media/scratchpad_01/erp016/seq_data/Quinoa-96/BAM_output
samoutdr=/media/scratchpad_01/erp016/seq_data/Quinoa-96/SAM_output
bwa0712dr=/media/bulk_01/programs/bwa-0.7.12

cd $fqdir/$subdir/
        fn1=$fqdir/$subdir/merged_R1.fastq
         fn2=$fqdir/$subdir/merged_R2.fastq
        sambn=bsa_merged

        bn1=$fn1
        bn2=$fn2

echo fqdir/subdir $fqdir/$subdir
echo fn1 $fn1
echo fn2 $fn2
echo samoutdr/sambn $samoutdr/$sambn
echo genome $genome

        $bwa0712dr/bwa mem \
          -M -t 1 \
          $genome $bn1 $bn2 > $samoutdr/$sambn.sam

    java -Xmx4g -Djava.io.tmpdir=/tmp -jar \
          /media/bulk_01/programs/picard-tools-1.96/SortSam.jar \
          SORT_ORDER=coordinate \
          INPUT=$samoutdr/$sambn.sam \
          OUTPUT=$bamoutdr/$sambn.bam \
          VALIDATION_STRINGENCY=LENIENT \
          CREATE_INDEX=true

samtools flagstat $bamoutdr/$sambn.bam > $bamoutdr/$sambn.flagstat.txt
```

Variant calling with Mpileup (submit script)

```
#!/bin/bash
#$ -cwd

for fn in `ls /media/scratchpad_01/erp016/seq_dataV2/contig_list_files`
do
```

```
qsub mpileup_split.sh $fn

done
```

Variant calling with Mpileup (mpileup_split.sh)

```
#!/bin/bash
#$ -q stat.short
#$ -cwd

## cd /media/scratchpad_01/erp016/seq_dataV2/BAM_output
## for fn in `ls /media/scratchpad_01/erp016/seq_dataV2/contig_list_files`; do qsub
mpileup_split.sh $fn; done

SAMTOOLS=/media/bulk_01/programs/samtools-1.2/samtools
BCFTOOLS=/media/bulk_01/programs/bcftools-1.2/bcftools
VCFTOOLS=/media/bulk_01/programs/vcftools-vcftools-78add55/src/cpp/vcftools

fn=$1

cd /media/scratchpad_01/erp016/seq_dataV2/BAM_output

set=${fn#contig_list_split.}

for dn in `ls *.bam`
do
	$SAMTOOLS view -b -L
/media/scratchpad_01/erp016/seq_dataV2/contig_list_files/contig_list_split.$set $dn >
split/$set.$dn
	$SAMTOOLS index split/$set.$dn
done

$SAMTOOLS mpileup -d 10000 -f
/media/scratchpad_01/erp016/seq_dataV2/polished_assembly_V2.1.unix.fasta -g -t DP,DPR,DV,DP4
split/$set.*.bam > split/$set.raw.bcf
$BCFTOOLS call -o split/$set.called.bcf -O b -f GQ,GP -v -m split/$set.raw.bcf
```

Splitting VCF files

```
#!/usr/bin/perl
use strict;
use warnings;

# use File::Basename;
# use File::Spec;

# author: E.N. van Loo

# What does it do:
# reads genotype call output from bcftools -call with counts per genotype
# counts for P1, P2 and BSA-merged for combinations of 0/0, 0/1, 1/1, 0/2, 1/2, 2,2 and so on if more alleles;
# uses the field description to create hashes of arrays

##########################################
# check if arguments have been given
##########################################

if ($#ARGV ne 1)
# print usage if not enough arguments given:
  {
    print STDERR "Too few arguments for Filter...pl: required: first argument - basename indicating vcf/bcf inputfile; second argument - context_length\n";
    print STDERR "\nusage: \nbcftools view your_file_basename.bcf | perl filter.pl your_filteredcounts_basename context_length \n" ;
    print STDERR "input is a streaming vcf file either from cat vcf or from bcftools view\n";
    print STDERR "argument 1 (ARGV[0]) = your filename for output file with read counts per genotype and per allele per context\n";
    print STDERR "argument 2 (ARGV[1]) = context length\n";
    print STDERR "genotype counts per allele go to your_filename.geno.out\n";
    print STDERR "counts of combinations of 0/0 to 3/3 for P1, P2 and BSA_bulk go to your_filename.combi.out\n";
    print STDERR "counts of genotype scores (phased alleles) per set of variants (=contexts) go to your_filename.context.out\n";
    die "\n";
  }
else {print STDERR "number of arguments OK; $ARGV[0] $ARGV[1]\n"};

# first argument of call specifies genotype scores output
my $fn = $ARGV[0];

# filenames based on $fn
open(GENO, '>', "$fn.geno.out") or die "Could not open file $fn.geno.out $!";
open(CONTXT, '>', "$fn.context.out") or die "Could not open file $fn.context.out $!";
open(COMBI,'>', "$fn.combi.out") or die "Could not open file $fn.combi.out $!";
open(NAME,'>', "$fn.name.lst") or die "Could not open file $fn.name.lst $!";

print STDERR "Files geno.out, context.out combi.out and name.lst opened\n";

# second argument of call specifies the length of the context
# number of variants that passed filtering for which read counts are added up to be able to a genotype for each BSA_genotype
my $size_context=$ARGV[1];

print CONTXT "read counts summed up in context.out for $size_context consecutive variants\n";

my @hd_chrom; # array for reading header line with #CHROM
my @name; # array of sample names (genotypes)
my $chrom_read=0; # check on whether #CHROM was encountered
my $double_hashtag_read=0; # 0 when double hash tag lines not yet encountered, 1 when at least one ## line has been read
my $previous_contig=""; # flag to check on whether a new contig has started - for summing up
genotype combinations

my $m=0; # counter to check how many variants were summed up

######################################################################################
# set initial number at 0 for all genotype combinations (first time outside while VCF then for each
contig re-initiliased within while VCF
#my @genotypes=("./.","0/0", "0/1", "1/1", "0/2", "1/2", "2/2", "0/3", "1/3", "2/3", "3/3","0/4",
"1/4", "2/4", "3/4", "4/4"); # checked for quinoa: no situations with 5 alleles including REF
my @genotypes=("./.","0/0", "0/1", "1/1", "0/2", "1/2", "2/2", "0/3", "1/3", "2/3", "3/3"); # checked
for quinoa: no situations with 5 alleles including REF

my %genotype3;

my @data; # array with genotype data in called.bcf
my @subfields; # array with the subfields for each column with genotype data
my @format; # the string in data[8] is split on : to get GT, PL etcetera in the elements
my %geno_per_pos; # hash with genotype data split into genotype columns and subfields per
genotype according to FORMAT
my $contig; # contig name

my %counts_per_context; # hash with read counts of P1 and P2 alleles summed up over n
consecutive variants

my $new_context= 1; # flag at 1 for first contig to be able to initialise within the sub sum_up and
then set to 1 each time a context is finished
my $new_contig = 0; # will be set to 1 the first time a read of a new contig is read - for finishing the
last sum up of counts
my $first = 0; # before first context was summed up

my $A_GT; # genotype of P1
my $CR_GT; # genotype of P2
my $BSAm_GT; # genotype of BSA_merged

my $pos_start=0; # start position of context
my $pos=0; #  position of variant - initial value - changed for new contexts - improve: initialise after
first filter_pass?

my $previous_pos=0; # init value
my $filter_passed=0; # 0 when not passed and 1 when passed

################################################
# subroutine adds up counts for n consecutive variants
# called with &sum_up_per_context
# uses global array for all genotypes with counts of P1 and P2 alleles
# until n reaches context size
# then prints a line with read counts and file with genotype calls
# then resets the counts to zero for new context
################################################

sub sum_up_init {
```

```
          for my $sel_geno (0 .. 98)
          {
            for my $sv (0 .. 1) # from 0 to 1

# was:      for my $sv (0 .. $#{ $geno_per_pos{$sel_geno}{'DPR'} }) # from 0 to last field in DPR (so for all alleles, but now printed only for first for 2 alleles)
            {
              $counts_per_context {$sel_geno}[$sv] = 0;
            }
          }
    $pos_start = $pos;
    $m=0;
return;
}

# prints output of context sum_up if end of context, contig or end of file
sub sum_up_print {

     my $pos_end;
     my $print_contig;

     if (($new_contig eq 1) or (eof(STDIN)))
     {
       $pos_end=$previous_pos;
       $print_contig=$previous_contig;
     }
     else # if context finished
     {
       $pos_end=$pos-1;
       $print_contig=$contig;
          }

          print CONTXT "$print_contig $pos_start","-",$pos_end;

          # For all genotypes: print
          for my $sel_geno (0 .. 98)
           {
            print CONTXT " $counts_per_context{$sel_geno}[0] $counts_per_context{$sel_geno}[1]";
           } # end print summed up counts for all genotypes

    print CONTXT "\n"; # Print newline:
return; # here you can return a value from the sub through the paroefameter list
&count_per_context (@output when array);
}

sub sum_up_per_context {

    $m++; # counter that counts number of variants summed up - print when $m reaches $n and then reset

#    print "m:",$m,"\n";
```

```
    if ($m > $size_context)
    {
#       print "m > size_context", "m: $m size_context: $size_context \n";
      &sum_up_print;
      &sum_up_init;
    }

    # $AR flips scores of ALT to P1 if P1 is 1/1 and vice versa REF to P2 when P1 is 1/1 -- no flip when
P1 is 0/0 (Ref/Ref)
    my $AR = 1; # if P1 is 0/0;
    if ($A_GT eq "1/1") { $AR = 0;} # set to zero when P1 is 1/1

    for my $sel_geno (0 .. 98)
    {

# for my $sv (0 .. $#{ $geno_per_pos{$sel_geno}{'DPR'} }) # $# is length of DPR field (number of
alleles -1, now only for 2 alleles)

          {
      my $count1 =
$geno_per_pos{$sel_geno}{'DPR'}[0]*$AR+$geno_per_pos{$sel_geno}{'DPR'}[1]*(1-$AR);
      my $count2 = $geno_per_pos{$sel_geno}{'DPR'}[0]*(1-
$AR)+$geno_per_pos{$sel_geno}{'DPR'}[1]*$AR;
      $counts_per_context{$sel_geno}[0]+=$count1; # counts P1 allele count
      $counts_per_context{$sel_geno}[1]+=$count2; # counts P2 allele count
#       print "counts: $counts_per_context{$sel_geno}[0] $counts_per_context{$sel_geno}[1] \n";
          }
    }
return;
}
##################################################
# end of sum_up_per_context
##################################################

##################################################
# Rest of MAIN part - only global variables defined in MAIN before sub sum_up_per_context because
of use strict
##################################################

#################################################
# intialise the genotype combinations count for the first time
#################################################
foreach my $A (@genotypes)
{
    foreach my $CR (@genotypes)
    {
         foreach my $BSAm (@genotypes)
          {
             $genotype3{$A}{$CR}{$BSAm} = 0;
           }
    }
```

```
}
# end of initialise genotype combinations count
#####################################################################################

# initalises the read counts for contexts;
&sum_up_init;

#####################################################################################
# start of read all VCF records and count combinations
# write counts of reads for genotypes for 0/0 - 1/1 or 1/1 - 0/0 to one file
# write counts of reads for genotypes for 0/0 - 0/1 or 0/1 - 0/0 to second file
# doesn't write three or four allelic for now but they are counted in the combinations file
#####################################################################################

print STDERR "reading VCF started\n";

while ((my $newline = <STDIN> ))
  { # Begin VCF

        if ( substr( $newline, 0, 2 ) eq "##" )
        {
                $double_hashtag_read=1;
         # do nothing but indicate ## read; go to end of while for next line until no longer in ## lines
        }

        elsif ( substr( $newline, 0, 6 ) eq "#CHROM" )
          { #Begin start check #CHROM
            chomp($newline);

         # fields in header separated by tab (\t)
            my @field = split /\t/, $newline;
            $hd_chrom[0]=$field[0]; #CHROM
            $hd_chrom[1]=$field[1]; #POS
            $hd_chrom[2]=$field[2]; #ID
            $hd_chrom[3]=$field[3]; #REF
            $hd_chrom[4]=$field[4]; #ALT
            $hd_chrom[5]=$field[5]; #QUAL
            $hd_chrom[6]=$field[6]; #FILTER
            $hd_chrom[7]=$field[7]; #INFO
            $hd_chrom[8]=$field[8]; #FORMAT

# prints header of list of sample names  (directory/filenames of bam-files used in mpileup)
        print NAME "Sample_name\n";

# makes short names (A, CR, BSAm, A01,A02...H12) of samples - depending on name conventions
used
# adapt for new experiment

         # $#field is length of array @field (zero-based)
          for my $i (9..$#field)
                {
           my $j =$i-9;
```

```
                        $name[$j]=$field[$i];

                # prints sample names one by one
                print NAME "$name[$j]\n";

                # prints first row of geno.out and context.out with short sample names (now A, CR, BSAm,
A01..H12)
# lines below if you want to retain the xy. prefix
#                  $name[$j] =~ s/(^split\/)(.*)(.bam)/$2/;
#                  $name[$j] =~ s/(...)(AZ_)(CR_183)/$1$3/;
#                  $name[$j] =~ s/(...)r182-4_(.*)/$1$2/;
#                  $name[$j] =~ s/(...)(A)(tlas)(_183)/$1$2$4/;
#                  $name[$j] =~ s/(...)(B_)(A)tlas_A-1(_)(r)(182)/$1$3$4$6/;
#                  $name[$j] =~ s/(...)(CR_)(B-1_)(r)(182)/$1$2$5/;

                $name[$j] =~ s/(^split\/)(.*)(.bam)/$2/;
                $name[$j] =~ s/^(...)//;
                $name[$j] =~ s/(AZ_)(CR_183)/$2/;
                $name[$j] =~ s/r182-4_(.*)/$1/;
                $name[$j] =~ s/(A)(tlas)(_183)/$1$3/;
                $name[$j] =~ s/(B_)(A)tlas_A-1(_)(r)(182)/$2$3$5/;
                $name[$j] =~ s/(CR_)(B-1_)(r)(182)/$1$4/;

                      } # END for all geno names on #CHROM line

          close NAME;
          print GENO "contig pos ", join(" ",@name),"\n";
          print CONTXT "contig pos ", join(" ",@name),"\n";
#          print COMBI "contig A CR BSA:./. 0/0 0/1 1/1 0/2 1/2 2/2 0/3 1/3 2/3 3/3 0/4 1/4 2/4 3/4 4/4
number of variants\n";
          print COMBI "contig A CR BSA:./. 0/0 0/1 1/1 0/2 1/2 2/2 0/3 1/3 2/3 3/3 number of
variants\n";

          $chrom_read=1;
               } # end elseif of check on #CHROM
            elsif (($chrom_read==0) or ($double_hashtag_read==0))
               { # if not newline starts with ## or #CHROM
                             print STDERR "not a proper VCF as no #CHROM header line found after ## lines";
                             die;
               }

            else # carry out reading a dataline if ## line and a #CHROM was encountered before

#####################################################
# reads a dataline with genotype/sample data, adds to genotype combinations, filters, prints filter-
passed variants and sums up contexts
#####################################################
            {
                        chomp ($newline);
                        my @field = split /\t/, $newline;   # @field is an array with the tab-delimited fields
in the data line per position
                        $contig = $field[0];
                $pos = $field[1];
                my $locus =$contig."-".$pos;

#check whether a new contig starts
# need for the contig value of previous line to be present here
# initialise outside while with my $ previous_contig="";
# after checking whether contig = previous_contig, put present contig in previous_contig

      if ((($contig ne $previous_contig) and ($previous_contig ne "")) or (eof(STDIN)))
      {
      $new_contig = 1; # new contig starts

      #  print final counts on genotype combinations per contig

      foreach my $A (@genotypes)
      {
         foreach my $CR (@genotypes)
         {
            print COMBI "$previous_contig $A $CR ";
                  foreach my $BSAm (@genotypes)
               {
#              if ($genotype3{$A}{$CR}{$BSAm} > 0) # print line only when > 0 cases found for this
combination
#                     prints all combinations
                {
# prints 15 columns with counts for all 15 combinations for one combinations of A and CR
# number of lines for each contig: 15 x 15 = 225 lines ....we may go back to only up to 4 alleles after
the first split file is handled (then only 10 x 10 lines ...)

                 print COMBI "$genotype3{$A}{$CR}{$BSAm} ";
                }
           }
                      print COMBI "\n";
         }
      }


      if (eof(STDIN))
           {print COMBI "FINISHED FILE\n";}

      # print sum_up
      &sum_up_print;
      &sum_up_init;

      #reset flag new_contig
      $new_contig=0;

      #re-initialise genotype combinations count for new contig
       foreach my $A (@genotypes)
       {
           foreach my $CR (@genotypes)
           {
```

```
                    foreach my $BSAm (@genotypes)
                    {
                            $genotype3{$A}{$CR}{$BSAm} = 0;
                    }
                }
          }
        } # end of print output for a finished contig or finished file

        $previous_contig = $contig;


##############################################
# start filter and output per line here:
###############################################
# genotype data are in column 10 to end (with indices 9 to end -1 in the array @field)
##############################################

    my @data = @field;
    my @subfields; # array with the subfields for each column with genotype data
    my @format = split (":",$data[8]); # the string in data[8] is split on : to get GT, PL etcetera in the
elements

    for my $i (9..$#data)   # split data columns into subfields (still for the current line = current
position)
           {
             my $j=$i-9; # i is column number (-1 as data starts with [0]), each : separated field is put in
different element of data_field_value
             my @data_fields=split(":",$data[$i]); # data_fields should have same number of elements
ans @format_subfield
             my $geno_id_number = $j;

             for my $format_field (0 .. $#format)     # @format_field; created for each format_field (for
GT, PL etcetera), stores the different values in the field GT or PL and so on
                {
                 my @data_subfield_value = split(",",$data_fields[$format_field]); # the data fields are
split into the individual comma separated values
                 for my $sv (0 .. $#data_subfield_value)
                     {
                      $geno_per_pos{$geno_id_number}{$format[$format_field]}[$sv] =
$data_subfield_value[$sv];
                     }
                }  # end of looping through format fields (GT, PL, DP, DP4 etcetera)
             } # end of for each genotype column (9 to end) in this line/position

# GT has only one field, therefore index [0]

           $A_GT = $geno_per_pos{'1'}{'GT'}[0];
           $CR_GT = $geno_per_pos{'0'}{'GT'}[0];
           $BSAm_GT = $geno_per_pos{'2'}{'GT'}[0];

       my $BSAm_DPRn = $#{$geno_per_pos{'2'}{'DPR'}}+1; # number of alleles found for BSAmerged
($# is index of last element, number is index of last element plus 1)
```

```
        $genotype3{$A_GT}{$CR_GT}{$BSAm_GT}++;

# end of adding to 1000 genotype combinations for P1, P2 and BSAmerged (per data line)

# START OF FILTERING PROCESS
# examples:
#       if ((($A_GT eq '0/0') and ($CR_GT eq '0/1') and ($BSAm_GT eq '0/1')) or (($A_GT eq '0/1')
and ($CR_GT eq '0/0') and ($BSAm_GT eq '0/1')))
#       if (($BSAm_GT eq '0/1') or ($BSAm_GT eq '1/2') or (($BSAm_GT eq '1/1') and ($A_GT eq
'1/1') and ($CR_GT ne '1/1'))
#       if (($BSAm_GT eq '1/1') and ($A_GT eq '1/1') and ($CR_GT ne '1/1'))

#       if ($BSAm_GT ~~ ['0/1','0/2','1/2','1/3','2/3'])
#       if ($BSAm_GT ~~ ['0/2','1/2','0/3','1/3','2/3'])

#     if (($A_GT eq '0/1') and ($CR_GT ~~ ['0/0','1/1']) and ($BSAm_DPRn le 2))


# DEFINE YOUR FILTER HERE:
     if (((($A_GT eq '0/0') and ($CR_GT eq '1/1')) or (( $A_GT eq '1/1') and ($CR_GT eq '0/0'))) or
(eof(STDIN)))

# NOW: only print data lines for which segregation is expected to be easy: only 2 alleles (one of
which is reference for all genotypes)
        {

          $filter_passed=1;

       &sum_up_per_context;

# subroutine adds up for n consecutive variants that pass the filter and prints for each n variants
added up; resets to 0 for new context

#       $previous_pos=$pos; # needed for sum_up to print the last line of a contig well

          print GENO "$contig $pos";

          for my $sel_geno (0 .. 98)
          {
#              for my $sv (0 .. $#{ $geno_per_pos{$sel_geno}{'DPR'} }) # $# is length to possibly also
print more than 2 alleles
        for my $sv (0 .. 1) # dit drukt alleen allelen 0 en 1 af (die splitsen uit in de gebruikte selectie)
              {
               print GENO " ", $geno_per_pos{$sel_geno}{'DPR'}[$sv];
              }
          }
          print GENO "\n";
       $filter_passed=0; # reset to 0
        } # end of if for selection filter and printing selected values (printed per contig/pos position
to reduce memory use)

        $previous_pos=$pos;
####################################################
# end of reading/filtering/printing a single data line with genotype/sample data for all samples
#####################################################

                } # End of else for reading and filtering a single data line with genotype/sample data for all
samples

    } # end while VCF

close GENO;
close CONTXT;


# End of script
```

Attachment 2 : 87 saponin biosynthesis pathway related genes

Attachment 2: Overview of the multigene families of oxidosqualene cyclases (OSCs), cytochromes P450 (P450s) and family 1 UDP-glycosyltransferases (UGTs) involved in triterpenoid saponin biosynthesis. The name, genebank ID for the protein, genebank ID for nucleotide sequence, plant spieces and the reference in which paper it has been published are given.

Name	GenBank ID protein	GenBank ID nucleotide	Plant species	Reference
Oxidosqualene cyclases (OSCs)				
Accurate (and putative) β-amyrin synthases				
AaBAS	ACA13386	EU330197	A. annua	Kirby et al. (2008)
AsOXA1	AAX14716	AY836006	A. sedifolius	Cammareri et al. (2008)
AsbAS1	CAC84558	AJ311789	A. strigosa	Haralampidis et al. (2001)
BgbAS	BAF80443	AB289585	B. gymnorhiza	Basyuni et al. (2007)
BPY	BAB83088	AB055512	B. platyphylla	Zhang et al. (2003)
EtAS	BAE43642	AB206469	E. tirucalli	Kajikawa et al. (2005)
GgbAS1	BAA89815	AB037203	G. glabra	Hayashi et al. (2001)
GmAMS1	AAM23264	AY095999	G. max	Chung et al. (2007)
GsAS1	ACO24697	FJ790411	G. straminea	Liu et al. (2009)
cOSC1	BAE53429	AB181244	L. japonicus	Sawai et al. (2006a)
MtAMY1 = β-AS	AAO33578	AF478453	M. truncatula	Iturbe-Ormaetxe et al. (2003)
β-AS = MtAMY1	CAD23247	AJ430607	M. truncatula	Suzuki et al. (2002)
NsβAS1	ACH88048	FJ013228	N. sativa	Scholz et al. (2009)
PNY	BAA33461	AB009030	P. ginseng	Kushiro et al. (1998)
PNY2	BAA33722	AB014057	P. ginseng	Kushiro et al. (1998)
PSY	BAA97558	AB034802	P. sativum	Morita et al. (2000)
SlTTS1	ADU52574	HQ266579	S. lycopersicum	Wang et al. (2011)
SvBS	ABK76265	DQ915167	S. vaccaria	Meesapyodsuk et al. (2007)
Accurate lupeol synthases				
BgLUS	BAF80444	AB289586	B. gymnorhiza	Basyuni et al. (2007)
BPW	BAB83087	AB055511	B. platyphylla	Zhang et al. (2003)
GgLUS1	BAD08587	AB116228	G. glabra	Hayashi et al. (2004)
cOSC3	BAE53430	AB181245	L. japonicus	Sawai et al. (2006)
OEW	BAA86930	AB025343	O. europaea	Shibuya et al. (1999)
RcLUS	ABB76766	DQ268869	R. communis	Guhling et al. (2006)
TRW	BAA86932	AB025345	T. officinale	Shibuya et al. (1999)
Accurate dammarenediol synthases				
CaDDS	AAS01523	AY520818	C. asiatica	Kim et al. (2009)
PNA = DDS	BAF33291	AB265170	P. ginseng	Tansakul et al. (2006)
DDS = PNA	ACZ71036	GU183405	P. ginseng	Han et al. (2006)
Accurate (and putative) cycloartenol synthases				

AsCS1	CAC84559	AJ311790	A. strigosa	Haralampidis et al. (2001)
CAS1/At2g07050	NP_178722	NM_126681	A. thaliana	Corey et al. (1993)
BPX	BAB83085	AB055509	B. platyphylla	Zhang et al. (2003)
BPX2	BAB83086	AB055510	B. platyphylla	Zhang et al. (2003) [
CaCYS	AAS01524	AY520819	C. asiatica	Kim et al. (2005)
CPX	BAD34644	AB116237	C. pepo	Shibuya et al. (2004)
CsOSC1/CSI	BAB83253	AB058507	C. speciosus	Kawano et al. (2002) [
GgCAS1	BAA76902	AB025968	G. glabra	Hayashi et al. (2000)
KcCAS	BAF73930	AB292609	K. candel	Basyuni et al. (2007)
LcCAS1	BAA85266	AB033334	L. aegyptiaca	Hayashi et al. (2001b)
cOSC5	BAE53431	AB181246	L. japonicus	Sawai et al. (2006a)
PNX	BAA33460	AB009029	P. ginseng	Kushiro et al. (1998)
PsCAS	BAA23533	D89619	P. sativum	Morita et al. (1997)
RsCAS	BAF73929	AB292608	R. stylosa	Basyuni et al. (2007)
RcCAS	ABB76767	DQ268870	R. communis	Guhling et al. (2006)
	Accurate (and putative) lanosterol synthases			
LSS1/At3g45130	NP_190099	NM_114382	A. thaliana	Kolesnikova et al. (2006); Suzuki et al. (2006)
CPR	BAD34646	AB116239	C. pepo	Shibuya et al. (2004)
LcOSC2	BAA85267	AB033335	L. aegyptiaca	Hayashi et al. (2001)
cOSC6	BAE95409	AB244670	L. japonicus	Sawai et al. (2006)
cOSC7/LAS	BAE95410	AB244671	L. japonicus	Sawai et al. (2006)
PNZ	BAA33462	AB009031	P. ginseng	Suzuki et al. (2006)
TRV	BAA86933	AB025346	T. officinale	Shibuya et al. (1999)
	Moderate accurate OSCs			
CAMS1/At1g78955	NP_683508	NM_148667	A. thaliana	Kolesnikova et al. (2007)
AtBAS1/At1g78950	NP_178016	NM_106544	A. thaliana	Shibuya et al. (2009)
PEN1/At4g15340	NP_567462	NM_117622	A. thaliana	Husselstein-Muller et al. (2001); Xiang et al. (2006); Kolesnikova et al. (2007)
BARS1/At4g15370	NP_193272	NM_117625	A. thaliana	Lodeiro et al. (2007)
PEN3/At5g36150	NP_198464	NM_123006	A. thaliana	Morlacchi et al. (2009)
THAS1/At5g48010	NP_199612	NM_124175	A. thaliana	Fazio et al. (2004); Field and Osbourn (2008)
MRN1/At5g42600	NP_199074	NM_123624	A. thaliana	Xiong et al. (2006)
CPQ	BAD34645	AB116238	C. pepo	Shibuya et al. (2004)
KdLUS	ADK35126	HM623871	K.daigremontiana	Wang et al. (2010)
KdCAS	ADK35127	HM623872	K.daigremontiana	Wang et al. (2010)
LcIMS1	BAB68529	AB058643	L. aegyptiaca	Hayashi et al. (2001b)
StrBOS	BAH23676	AB455264	S. rebaudiana	Shibuya et al. (2008)

	Multifunctional OSCs			

LUP1/At1g78970	NP_178018	NM_106546	A. thaliana	Herrera et al. (1998);
				Segura et al. (2000);
				Husselstein-Muller et al. (2001)

AtLUP2/At1g78960	NP_178017	NM_106545	A. thaliana	Husselstein-Muller et al. (2001); Kushiro et al. (2000)
LUP5/At1g66960	NP_176868	NM_105367	A. thaliana	Ebizuka et al. (2003)
PEN6/At1g78500	NP_177971	NM_106497	A. thaliana	Ebizuka et al. (2003); Shibuya et al. (2007)
CsOSC2/CSV	BAB83254	AB058508	C. speciosus	Kawano et al. (2002)
LjAMY2	AAO33580	AF478455	L. japonicus	Iturbe-Ormaetxe et al. (2003)
KcMS	BAF35580	AB257507	K. candel	Basyuni et al. (2006)
KdTAS	ADK35123	HM623868	K.daigremontiana	Wang et al. (2010)
KdGLS	ADK35124	HM623869	K.daigremontiana	Wang et al. (2010)
KdFRS	ADK35125	HM623870	K.daigremontiana	Wang et al. (2010)
OEA	BAF63702	AB291240	O. europaea	Saimaru et al. (2007)
PSM	BAA97559	AB034803	P. sativum	Morita et al. (2000)
RsM2	BAF80442	AB263204	R. stylosa	Basyuni et al. (2007)
RsM1	BAF80441	AB263203	R. stylosa	Basyuni et al. (2007)
SlTTS2	ADU52575	HQ266580	S. lycopersicum	Wang et al. (2011)
	Cytochromes P450			
CYP51H10	ABG88961	DQ680849	A. strigosa	Qi et al. (2006)
CYP93E1	BAE94181	AB231332	G. max	Shibuya et al. (2006)
CYP93E3	BAG68930	AB437320	G. uralensis	Seki et al. (2008)
CYP88D6	BAG68929	AB433179	G. uralensis	Seki et al. (2008)
	Family 1 UDP-glycosyltransferases (UGTs)			
UGT73P2	BAI99584	AB473730	G. max	Shibuya et al. (2010)
UGT91H4	BAI99585	AB473731	G. max	Shibuya et al. (2010)
UGT71G1	AAW56092	AY747627	M. truncatula	Achnine et al. (2005)
UGT73K1	AAW56091	AY747626	M. truncatula	Achnine et al. (2005)
UGT73F3	ACT34898	FJ477891	M. truncatula	Naoumkina et al. (2010)
UGT74M1	ABK76266	DQ915168	S. vaccaria	Meesapyodsuk et al. (2007)