

Contents

1	Introduction	9
1.1	Wheat Genetics	9
1.1.1	Polyploidy and Wheat	9
1.1.2	Genetic Linkage	10
1.1.3	Experimental lines	10
1.1.4	Introgressions	11
1.2	Wheat Genomics	11
1.2.1	Molecular biology	11
1.2.2	Primers and PCR	13
1.3	Sequencing	14
1.3.1	DNA sequencing with Illumina	14
1.3.2	RNA-Seq	15
1.4	Sequence analysis	17
1.4.1	RNA-Seq	18
1.5	Wheat specific resources resources	19
1.5.1	Genomic sequence	19
1.5.2	Gene models	19
1.5.3	Genetic markers	20
1.5.4	Genetic maps	20
1.5.5	Online resources	21
1.6	Aim and objectives	21
2	PolyMarker: A fast polyploid primer design pipeline.	23
2.1	Background.	23
2.1.1	PCR amplification.	23
2.1.2	KASP assays.	24
2.1.3	Genome specific primers.	24
2.1.4	Objective.	28
2.2	Pipeline.	29

<i>CONTENTS</i>	2
-----------------	---

2.2.1 PolyMarker public web service	41
2.3 Applications of PolyMarker	41
2.3.1 KASP assays for public sets of SNPs	41
2.3.2 SNPs in a mutant population	43
2.4 Modifications of PolyMarker	44
2.4.1 Deletions on a mutant population	45
2.4.2 Genotyping <i>Puccinia striiformis</i> f. sp. <i>tritici</i> isolates.	47
2.5 Discussion	49
3 Genetic map of <i>Yr15</i> with RNA-Seq.	52
3.1 Background.	52
3.1.1 Segregation on F_2 populations.	52
3.1.2 SNP calling	53
3.1.3 <i>In Silico</i> mapping.	54
3.2 Mapping population.	56
3.3 Sequencing and mapping.	58
3.4 SNP Calling	60
3.5 Bulk Frequency Ratios	63
3.6 <i>In silico</i> mapping	66
3.7 Assay selection	69
3.8 SNP Validation	74
3.9 Genetic map	76
3.10 Methods	79
3.10.1 Base-call and Quality Control of sequencing reads	79
3.10.2 Alignment reads to gene models	80
3.10.3 Bulk Frequency Ratios and SNP calling	80
3.10.4 <i>In Silico</i> mapping	83
3.10.5 Primer design and KASP assays	85
3.10.6 Genetic map	86
3.11 Discussion	86
4 expVIP	91
4.1 Background.	91
4.1.1 Alternative expression browsers.	91
4.1.2 Public wheat RNA-Seq experiments.	92
4.1.3 Expression quantification with Kallisto.	92
4.1.4 Relational databases.	94

4.1.5	SQL	95
4.1.6	Model-View-Controller	98
4.1.7	Data visualisation	99
4.1.8	Aims	99
4.2	General design	100
4.3	Database design	101
4.4	Data integration pipeline	105
4.5	Graphical interface	114
4.6	Discussion	122
5	General discussion and final remarks	129
5.1	Integration of different genetic maps	129
5.2	Integration of different genome references.	132
5.3	Integration with other services	133
5.4	Side projects	134
5.5	Final remarks	134
A	Supplemental tables	150
A.1	PolyMarker supplemental tables.	150
B	Quality control	158
B.1	Sequence read quality	159
B.2	Sequence GC content	161
C	expVIP tutorial	163

List of Figures

1.1	Hibridizations that lead to bread wheat <i>T. aestivum</i>	10
1.2	Types of experimental lines	12
1.3	General types of sequencing ant their relationship with this PhD. PolyInDel is a project still in progress, not discussed in this Thesis.	14
1.4	Topics of this Thesis	16
1.5	PhD timeline.	22
2.1	PCR Diagram	24
2.2	KASP Assays	25
2.3	Target of genome specific primer.	26
2.4	Effect of position of variation on primer specificity.	27
2.5	Global search of templates in the reference contigs.	27
2.6	Selected regions around the SNP on every chromosome. .	28
2.7	Steps and tools called by PolyMarker	30
2.8	PolyMarker input	31
2.9	Sequence of flanking regions around the SNP.	32
2.10	Local alignment on regions around the SNP detects indels.	33
2.11	Alignment with mask and primer candidates.	35
2.12	Examples outputs from the PolyMarker website.	42
2.13	KASP assays to validate homozygous deletions.	46
3.1	Alleles on F_2 population.	53
3.2	BFR formula	55
3.3	Layers of information to do <i>In Silico</i> mapping.	56
3.4	Avocet + <i>Yr15</i> F_2 mapping population.	57
3.5	Coverage and SNPs between progenitors	60
3.6	Gene models with putative SNPs	61
3.7	Effect of BFR threshold on the number of SNPs	65

3.8	Location of SNPs with BFR>6	66
3.9	<i>In silico</i> location of SNPs with BFR>6	68
3.10	Genetic location of genes with SNPs between AVS and Yr15.	70
3.11	Selection criteria for marker design	71
3.12	BFRs of selected SNPs across bulks.	72
3.13	KASP output from the wheat variety panel	74
3.14	Genetic maps for <i>Yr15</i>	78
3.15	Steps used to go from the <i>F</i> ₂ population to the genetic map.	79
3.16	Haplotype and phenotype of 113 doubled haploid lines	89
4.1	Overview of kallisto.	94
4.2	Example of a relationship between tables.	95
4.3	MVC interaction between components.	98
4.4	General design of expVIP	100
4.5	expVIP database design	102
4.6	expVIP load data	106
4.7	Steps to run and load Kallisto	114
4.8	expVIP GUI layout	116
4.9	Features on expVIP	116
4.10	expVIP export image	117
4.11	Features on expVIP (continued)	118
4.12	Features on expVIP for Multiple gene comparisons	120
4.13	expVIP resources time line	122
4.14	Units used for expression normalisation	123
4.15	Alignment vs mapping pipelines	125
4.16	Changes to improve gene comparasions	126
4.17	Heatmap of the number of unique sessions of expVIP in the world.	128
5.1	Relational database integrating all the datasets.	130
5.2	Tables to store information about genetic maps.	131
5.3	Tables to store information about genome assemblies and their annotation.	133

List of Tables

2.1	Count of KASP assays designed for the 40,267 SNP markers located in the genetic map from Wang et al. (2014). 4,228 assays did not align to the target chromosome. Not designed: Primer3 could not find viable primers flanking the SNP.	43
2.2	Count of KASP assays designed for the 616,525 SNP markers located to a CSS scaffold from the 819,556 SNPs from Winfield et al. (2016) Not designed: Primer3 could not find viable primers flanking the SNP.	44
2.3	Summary table of the validation of candidate SNPs by KASP marker assays. Candidate SNPs are classified by number of supporting variant reads or by allele frequency and validated by KASP assays. Table from King et al. (2015).	45
2.4	Validation of homozygous deletions on line Cadenza0423.	48
2.5	PolyMarker used to genotype PST. The X and Y represent the two possible alleles. X:X and Y:Y correspond to homozygous call of the corresponding allele. X:Y correspond to heterozygous calls. The '-' symbol correspond to failed assays.	50
3.1	Arrangement and number of sequenced base pairs per sample.	58
3.2	Number of genes with a coverage over 20x, 10x and at least one read (>0x).	59
3.3	Count of SNPs per 100 bp on genes with at least 20x coverage.	61

3.4	Number of genes with SNPs assigned to the wheat chromosome arm CSS scaffolds (Mayer et al., 2014) using the best hit from BLAT (Kent, 2002)	62
3.5	Total number of SNPs scored in parents, individual bulks and in silico merged bulks.	63
3.6	SNPs in chromosome group 1S vs total number of SNPs with a minimum BFR from 0 to 10. AVS: SNPs coming from Avocet S. <i>Yr15</i> : SNPs comming from Avocet + <i>Yr15</i>	64
3.7	SNP and genes with BFR > 6 mapping to each of the chromosomes from the CSS assemblies. The chromosome assignment on the "Genetically mapped" column correspond to the map published in Wang et al. (2014).	67
3.8	Number of genes (and SNPs) with a unique hit (> 99% sequence identity) to a single wheat survey sequence scaffold.	73
3.9	Primer details for the markers to validate.	75
3.10	Results of validation of primers on the progenitors (AVS and <i>Yr15</i> , varieties known to contain <i>Yr15</i> (Cortez, Ochre and, Boston)	77
4.1	Studies with RNA-Seq replicates, sequenced with Illumina, at the time when expVIP was under development.	93
4.2	Species	95
4.3	Studies	95
4.4	Join example	97
4.5	Results of query for metadata	104
4.6	Results of query for values	105
4.7	Factors file	107
4.8	Homoeology file	110
A.1	Validation of mutations on <i>M₄</i> on Cadenza	151
A.2	Validation of mutations on <i>M₄</i> on Kronos	155

Listings

2.1	<code>Bio::PolyplloidTools::ExonContainer.add_alignments</code>	32
2.2	<code>Bio::PolyplloidTools::SNP.add_exon</code>	33
2.3	Function that assigns a chromosome	33
2.4	<code>Bio::PolyplloidTools::SNP.aligned_sequences . . .</code>	34
2.5	<code>Bio::PolyplloidTools::SNP.mask_aligned_chromosomal_snp</code>	38
2.6	<code>Bio::DB::Primer3::Primer3Record.score</code>	39
2.7	<code>Bio::DB::Primer3::Primer3Record.initialize</code>	39
2.8	Score values to select semi-specific primers	47
2.9	Function that always returns PST130 as chromosome . .	49
3.1	Method to call for the consensus on progenitors from a pileup	81
3.2	<code>base_coverage</code> gets the number of bases called from a single pileup.	81
3.3	<code>base_ratios</code> gets the SNP-Index on a single pileup.	82
3.4	Section of the code that	83
3.5	<code>Bio::Blat::Report.each_best_hit</code>	84
3.6	Extension to <code>Bio::Blat::Report::Hit</code> for filtering of spurious alignments.	85
4.1	Join example query	97
4.2	Query experiments and factors	104
4.3	Query values for gene and experiment group	105
4.4	Gene set fasta file	108
4.5	Load factors	109
4.6	Load genes from Fasta	111
4.7	Load expression values from file	113
4.8	Export data example	121

Chapter 1

Introduction

1.1 Wheat Genetics

The section describes alleles an the concept of gene, both as a locus in the genome (Quantitative Trait Locus, QTL) and an specific transcript (central dogma of molecular biology). Finally, it discuses traditional Mendelian inheritance and the effect of polyploidy. Some of this is described in the Yr15 chapter, maybe it is not needed any more here.

1.1.1 Polyploidy and Wheat

A polyploid species contains more than one set of related genomes, that may come from a chromosomal duplication or from an hybridization with a related species. *Triticum aestivum* (bread wheat) has gone trough an specialisation event and two major hybridization events. Initially, an unknown species first evolved in two different species around 7 million years ago to form the A and B genomes, whose closest known relative are *Triticum urartu* and *Aegelopolis speltoides*. As both ancestral wheat were able to cross, at some point around 5.5 million years ago the D genome arose, *Aegelopolis tauschii*. Then, less than 800 thousand years ago the ancient species carrying the A and B genomes hybridized and formed a tetraploid wheat, *Triticum turgidum* (pasta wheat). A final event occurred less than 400 thousand years ago, when pasta wheat hybridized with the carrier of the D genome, leading to bread wheat (Figure 1.1.1, Marcussen et al. 2014).

Because bread wheat contains three independent copies of its genome, the expectation is to have three homoeologues for each gene.

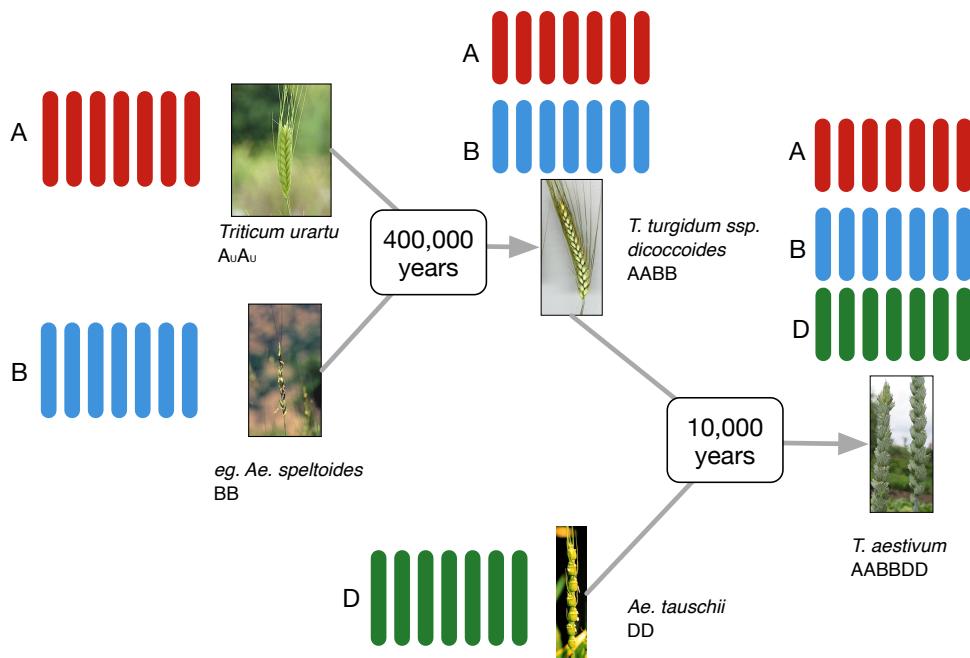


Figure 1.1: Hybridizations that lead to bread wheat *T. aestivum*.

Talk
about
par-
alogues.

1.1.2 Genetic Linkage

1.1.3 Experimental lines

In crops, genetic research usually consists on crossing individuals and study the traits of the next generation. The progeny of a cross is called population. Groups of seeds with the same genetic background are called lines. Plants with the same homozygous background are pure lines. To study particular locus experimental lines are developed, the following list describe the most common (Van Ooijen and Jansen, 2013b):

F₁ The first generation of the cross between two plants. If the progenitors are different homozygous, all the population is heterozygous.

F₂ populations come from a single heterozygous F₁ plant that is crossed to itself (self-crossed). The progeny is segregating (Figure 1.2a) with homozygous and heterozygous individuals. This type of population is used for the experiments in Chapter 3 and it is described in more detail in Section 3.1.1.

Back cross lines(BC) are used to fix a trait on a genetic background.

The process starts from a plant in the F_1 with the desired genotype or phenotype. This plant is crossed again to a plant from the line used as background (P_1). The progeny are called Back Cross 1 (BC1; Figure 1.2b). A plant from the BC1 with the desired genotype is selected and crossed to P_1 . The process can be repeated, and with each cross the region linked to the target locus is narrowed.

Near Isogenic Lines (NIL) After BC6, a line is considered Near Isogenic (Stam and Zeven, 1981). At this level of back crossing most of the genetic material is the same as P_1 , except for the region linked to the trait.

Recombinant Inbred Lines (RIL) are used to produce homozygous lines from an F_2 population. Each plant in the population is self-crossed. The lines that show an homozygous segregation are self-crossed again. After several iterations the line is considered homozygous. 1.2c).

Double Haploid Lines (DH) are an alternative technique to produce homozygous lines it. The individuals on the F_1 population are crossed to a different plant (ie maize crossed to produce wheat DH) to simulate pollination. On natural conditions, the gamets would be aborted. The embryos are rescued and treated with colchicine to induce a genome duplication. Since the duplication come from a single gamete, the resulting plant is homozygous (Figure 1.2d).

1.1.4 Introggressions

1.2 Wheat Genomics

1.2.1 Molecular biology

The genome is the genetic material of an organism, and can be studied using HTS to determine the sequence. However, only a small percentage of the genome is expressed in each cell in each condition, this differences in transcript production lead to differential gene expression between samples. The transcriptome is the complete set of RNA molecules present in a

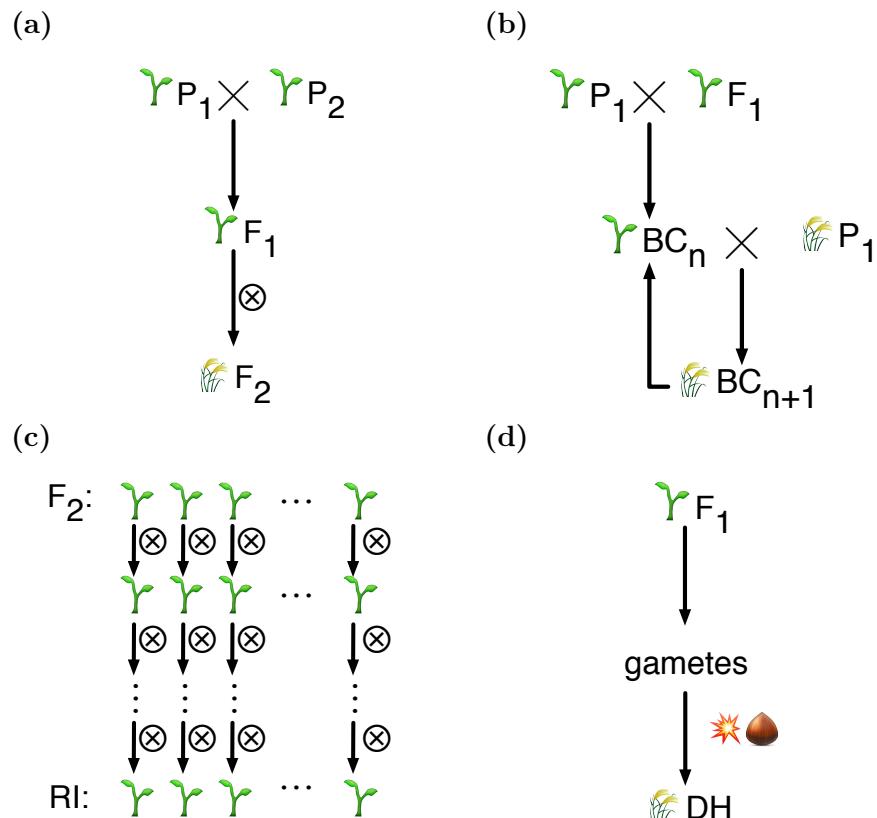


Figure 1.2: Types of experimental lines. represent populations and individual plants. (\times) represent a cross between lines. (\otimes) represent self-crosses. Ellipsis (...) represent repetition. represent a treatment to double the chromosomes from the gamete. (a) F_2 population. (b) Back Cross population. (c) Recombinant Inbred Lines. (d) Double haploid

biological system during specific conditions(Wang et al., 2009). The transcriptome of a sample and the differences in expression between samples can be studied using different methods. One of this is RNA-Seq, which is capable of capturing the dynamic range to quantify gene expression level(Mortazavi et al., 2008).

During transcription, a DNA sequence is transcribed into the primary RNA transcript (pre-mRNA) by the RNA polymerase. The RNA polymerase, and associated transcription factor, binds to promoter DNA and creates an RNA complementary copy. Depending on multiple factors, including gene regulation, this process can be repeated to create multiple copies of the sequence(Alberts et al., 2014). The pre-mRNA suffers various modifications like splicing and capping. In a gene, there are intragenic regions, called introns, that are removed from the pre-mRNA, while the remaining sequences, called exons, become part of the mature mRNA. Splicing is catalyzed by the spliceosome, which recognizes the sequences of the splice donor, branch and acceptor sites in the intron, cuts the intron and pastes the remaining exons. In some cases, different introns and exons can be included or excluded, creating alternative splicings of the RNA sequence(Alberts et al., 2014). Then, the 5' and 3' ends of the RNA sequence are modified to generate a mature messenger RNA (mRNA). A 7-methylguanosine cap is added to the 5' end. The 3' end is cleaved and polyadenylated, acquiring a poly(A) tail of approximately 200 adenines (Alberts et al., 2014). The result of this process is a mature mRNA, which forms part of the transcriptome.

1.2.2 Primers and PCR

To generate new copies of DNA, biological systems require a DNA polymerase to copy DNA into DNA or a reverse transcriptase to translate RNA to DNA. During DNA replication, the DNA polymerase binds to the template strand and to a short strand of RNA that serves as a primer for DNA synthesis and replication, and then adds complementary nucleotides. Eventually, this results in two complementary DNA strandsAlberts et al. (2014).

This process can be used to generate multiple copies of a DNA sequence using the Polymerase Chain Reaction (PCR). PCR uses an heat-stable DNA polymerase to amplify the template sequence. The DNA polymerase requires a primer sequence that is complementary to the

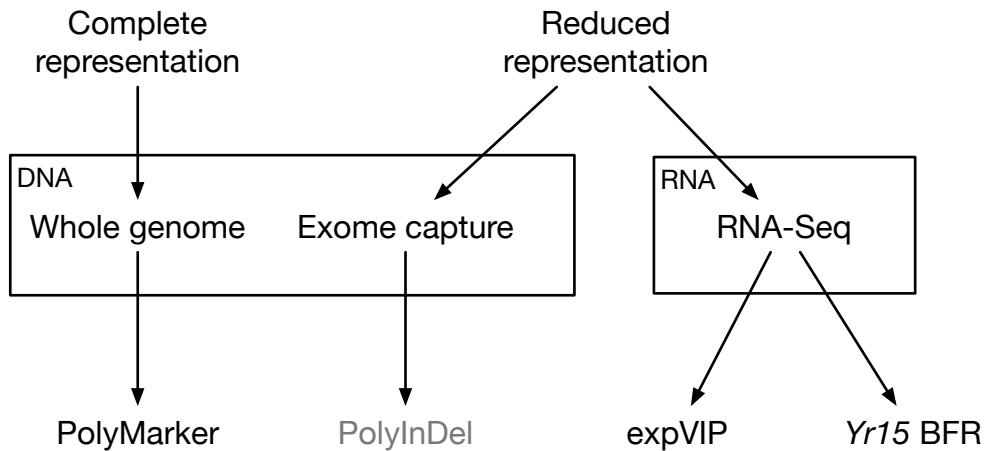


Figure 1.3: General types of sequencing ant their relationship with this PhD. PolyInDel is a project still in progress, not discussed in this Thesis.

template strand to begin the reaction. Usually, chemically synthesized oligonucleotides of approximately 20 bases are added as primers. This primer sequence can be specific to the target region, or random if multiple regions are being duplicated. Once the DNA strand has been elongated, the temperature is elevated and then lowered to separate the DNA strands. The process is repeated multiple times to increase the number of DNA sequences exponentiallyAlberts et al. (2014).

1.3 Sequencing

1.3.1 DNA sequencing with Illumina

In the Illumina sequencer, the DNA library is loaded into a flow cell, where the fragments are captured by their adapters. Each fragment is amplified into clonal clusters, resulting in multiple copies of the sequence bounded to the surface in close proximity. Then, the sequencer detects the fluorescent dye-labelled oligonucleotides that are added one by one to the bound sequences by taking multiple images. This process can be repeated for each pair of the DNA. copies of the same sequence are clustered together, it is possible to analyse the images and detect which base is added to each cluster, determining multiple sequences with high fidelity, and exporting them as a FASTQ file Lab (2015); Inc.

According to the objectives of the experiment and the quality and volume of the available DNA, the library can be prepared on fragments

of different sizes, the classification of the available sequencing for the fragments is the following Myllykangas et al. (2012); Metzker (2010); Shendure and Ji (2008); Hutchison (2007):

Single end When the fragments are short, it is possible to just sequence from the 5'-end the read.

Read Pairs When the sample consists fragments of up to 500bp, it is possible to read the 5' end up to the read length were the quality starts to drop, the molecule can be turned upside down, reverse complemented and sequence backwards. It is not required, but ideally, the fragments sequenced with read pairs should be selected to have an homogenous size. The reads are in opposite orientation relative to each other.

Overlapping Read Pairs are a variation to read pairs, where the size of the fragment is shorter than two times the read length. This allow an alignment between the two fragments to get an longer read with the limitations of the instrument.

Mate pairs are used to get reads separated at distances between 1kbp and 5kbp. To achieve this, the molecule is circularised and the point were the two ends of the fragment were joint a biotin marker is inserted. Then, the molecule is fragmented again and the fragments containing the biotin are sequenced in the same fashion that read pairs. The resulting reads have the same orientation.

1.3.2 RNA-Seq

Both DNA and RNA can be sequenced using available HTS technologies Originally, sequencers where designed with DNA in mind, so that analysing RNA requieres converting the transcriptome into a cDNA libraryLab (2015); Inc. To begin with, RNA is first isolated, purified and enriched for mature mRNA. For some sequencers, the resulting mRNAs need to be fragmented to improve sequence coverage, as current sequencers can only read sequences shorter than the transcripts. Once the quality of the RNA has been verified, it is converted into cDNA and adaptor sequences are added. First, the RNA strand is copied into first strand cDNA using the reverse transcriptase, as traditional polymerases

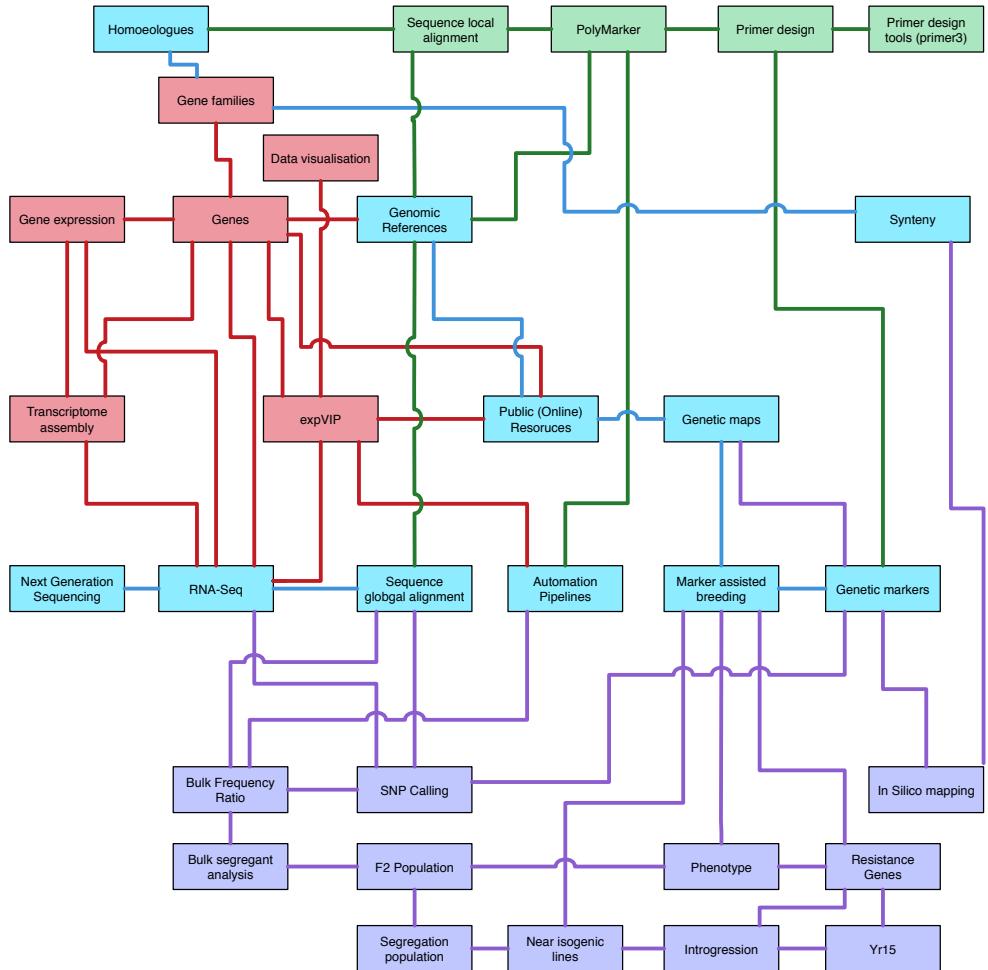


Figure 1.4: Topics of this Thesis

cannot convert from RNA to DNAAlberts et al. (2014). Reverse transcriptase, as other polymerase, requires a primer annealed to begin the polymerization, so random primer are used to avoid 3' bias and improve coverageLab (2015); Inc. To obtain the second strand, RNase H is used to cleave the original RNA. The remaining fragments serve as primers for DNA polymerase I, creating a double-stranded complementary DNA. The cDNA fragments then go through an end repair process, the addition of a single 'A' base, and then ligation of the adapters, which will be used by the sequencer to read the sequence. The products are then purified, to remove artefact and their quality is verifiedLab (2015); Inc. The remaining sequences are enriched using the polymerase chain reaction (PCR), which can generate thousands to millions of copies of a particular DNA sequenceLab (2015); Inc. The resulting cDNA library is sequenced using available high throughput sequencing technologies.

Figure
of This
process

1.4 Sequence analysis

This section discusses the criteria to decide analysis done after sequencing, when to do re-alignments or *de novo* assemblies, how to do SNP calling in diploid and polyploid organisms and the bulk frequency ratios.

DNA sequence alone is not alone to enough to understand the biology behind, a context is required. There are databases like Ensembl and NCBI that act as repositories of the known public sequences.

From the computational point of view, the problem can be viewed as a string matching. The Smith-WatermanSmith and Waterman (1981) and Needleman-WunschNeedleman and Wunsch (1970) algorithms are the gold standard interns of accuracy looking for similarity between sequences. However, the execution time for both of them is prohibitive to run in massive databases. The algorithm execution time is $O(mn)$, as it requires calculating a matrix of size mn where m is the target sequence and n is the query sequence. To scale this to a manageable problem algorithms like BLAST index the references and use heuristics to make the search more manageable, with some penalty in the accuracy. This alignments tools are useful for long stretches of DNA (like cDNA or contigs)Altschul et al. (1990).

TODO: List of global aligners -BLAST -BLAT -Exonerate -nucmer
-MAFFT -Clustal

When looking at a protein level, where the sequences may be only loosely similar, Hidden Markov Models (HMM) are used to search for protein families. This can be useful to annotate putative proteins and their functions. HMMs require a training dataset, where proteins are previously annotated and the reference is a model encoding the characteristics of a family, with associated probabilities. Hence, this technique is something between a sequences aligner and a classifierEddy (2004).

When analysing high-throughput sequencing, having millions of short sequences make unfeasible to try to align the data to every possible reference. However, one can take in advantage the fact that you know which organism you are looking for and, if available, use a genomic reference. For this, tools like MAQ, BWA, Bowtie, among others, provide indexed search. Once you have your reads aligned to a reference you can do more analysis, depending on the biological question being asked and

the type of sequencing carried on. Fortunately, most of the Short-Read sequence alignment produce similar outputs and the SAM format is becoming a de facto standard. This is allowing to make more modularised downstream analysis where you can test different aligners with different settings and pick the algorithm that better fits your experiment Liu and Schmidt (2012); Li and Durbin (2009); Li et al. (2009).

Ambiguity Codes

Make a table with the ambiguity codes and why they are useful.

1.4.1 RNA-Seq

One way to narrow down which genes are involved in certain trait or response to the environment is to focus on studying only the expressed genes. One of the techniques involving high-throughput sequencing is RNA-Seq. This technique captures the messenger RNA in the tissue being studied and sequenced. The premise is that you will find a gene more expressed if it is being used by the organism. Some proteins with a vital role for the cell are always expressed (i.e. RuBisCO for carbon fixation in plants GM (2000)). On the simplest of the experiments you would need two datasets to compare, one with the gene being looked expressed and one where it is not. The expression can come from different environmental conditions, development stage or different genotypes. Mortazavi et al. (2008)

Depending on how much *a priori* information of the analysed organism is available different bioinformatic approaches can be used.

Transcriptome alignment The reads are aligned to a database of known cDNA. Ideally, alternative splicing sequences are available, so a simple alignment should work (i.e. BWA, bowtie).

Genomic alignment The reads are aligned to the genome. The splice junctions, introns and exons need to be accounted, so simple alignment doesn't work. Regular alignments are used, but the reads may be trimmed at fixed sizes to allow discontinuous alignments using regular tools (i.e. Stampy, tophat/cufflinks)

De Novo transcriptome assembly If a reference of the organism is not available, it is possible to generate a draft transcriptome with

the RNA-Seq reads with traditional assemblers (velvet, abyss) or with specialised assembler tools like Trinity.

Once you have the alignments it is possible to evaluate the relative expression of the genes in the sample calculating the Reads per Kilobase per Million mapped reads (RPKM) or the Transcripts per Million (TPM). This normalises the expression by the amount of sequenced data and can be used to find which genes change in expression volume across different samples.

1.5 Wheat specific resources resources

During the course of this PhD, several resources were released in the wheat community.

1.5.1 Genomic sequence

Assemblies

-454 Liverpool -Chapman -IWGSC -TGACv1

1.5.2 Gene models

For this thesis, a set of gene models correspond to the coding sequence of a gene. The following sets of gene models are used in this thesis.

UniGenes. This is an effort of the National Center for Biotechnology Information (NCBI) to unify several ESTs deposited on their databases, per species. The set is generated automatically by aligning all the ESTs to each other and clustering them by identity. The longest EST is selected as the canonical representation of the gene (Pontius et al., 2002). Because this approach is not aware of the different genomes, and the set of ESTs may not include them, the algorithm collapses homoeologous genes. This needs to be taken in account when doing down stream analysis.

UCW gene models. This set of gene models come from tetraploid wheat come from a *de novo* assembly of *T. turgidum* (AABB) a *T. urartu*. In both cases RNA-Seq was assembled with several parameters and the resulting assemblies where merged. To separate homoeologues

genes that were assembled as a single gene, the reads were aligned to the assembly. Then, the phasing of the reads was used to separate the gene in the two homoeologues (Krasileva et al., 2013). This gene models are useful for tetraploid wheat. When using this reference with hexaploid wheat, care must be taken of reads coming from the D genome that will map to one of the alternative homoeologues.

Genome annotation IWGSC. To complement the CSS genome assembly, the genome was annotated using related grasses (*Brachypodium distachyon*, *Oryza sativa*, *Sorghum bicolor*, and *Hordeum vulgare*). The annotation was supported by RNA-Seq data from five samples from different tissues at three developmental stages (Mayer et al., 2014). This was the first genome-scale annotation effort including the three genomes independently. The main caveat of this annotation is that due the relatively small size of the scaffolds used as reference, several genes are splitted in two different gene models.

Genome annotation TGACv1. This is the companion annotation for the TGACv1 assembly. The annotation was produced with a novel approach, by merging four alternative assemblies from three RNA-Seq datasets. Also, long reads from PacBio were used to validate the gene models (ie. Full length, alternative splicing, canonical sequence) (Venturini et al., 2016). This gene models were made public during the last year of this PhD, hence they were not used for the analysis. However the tools described in Chapters 2 and 4 can work with the updated gene models.

1.5.3 Genetic markers

Markers -90k -820k -MASwheat/SRR -Chapman

1.5.4 Genetic maps

Genetic maps -Wang Wang et al. (2014) -Chapman/PopSeq (is the same population, improved) -latest additions! (Allen et al., 2016; Winfield et al., 2016)

1.5.5 Online resources

Portal -CeralsDB -MASWheat -Ensembl -Wheat-expression -Copo

The Collaborative Open Plant Omics (COPO; Davey et al. 2015) is an initiative to integrate data from different databases. The idea is to improve the interaction across repositories through an Application Programming Interface (API), a standard way to ensure the interoperability between systems. This approach keeps the responsibility of the system and data maintenance on the service providers. Service providers must ensure the compliance to standard of curation and integration, which would simplify the data analysis and visualisation, and ultimately promote the use of the integrated resources.

1.6 Aim and objectives

The main aim of this PHD is to make accessible the resources bioinformatic tools able to cope with the size and complexity of the wheat genome need to be developed. The tools can only be effective if they are designed taking in account the biological questions they will address. The tools developed in this PhD take advantage of the wheat resources as they were made public (Figure 1.5).

As this thesis is about the development tools that deal with polyploidy in wheat, the purpose is better reflective as objectives, rather than a constrained hypothesis.

1. Simplify the design of primers for marker assisted selection in polyploid wheat (Chapter 2).
2. Develop a pipeline that can be used find genetic markers for breeding, using bulk segregant analysis and reduced representation sequencing (Chapter 3).
3. Integrate and display expression experiments (Chapter 4)
4. Set a framework to integrate different resources (Chapter 5).

All the pipelines and data produced by this project is publicly available.

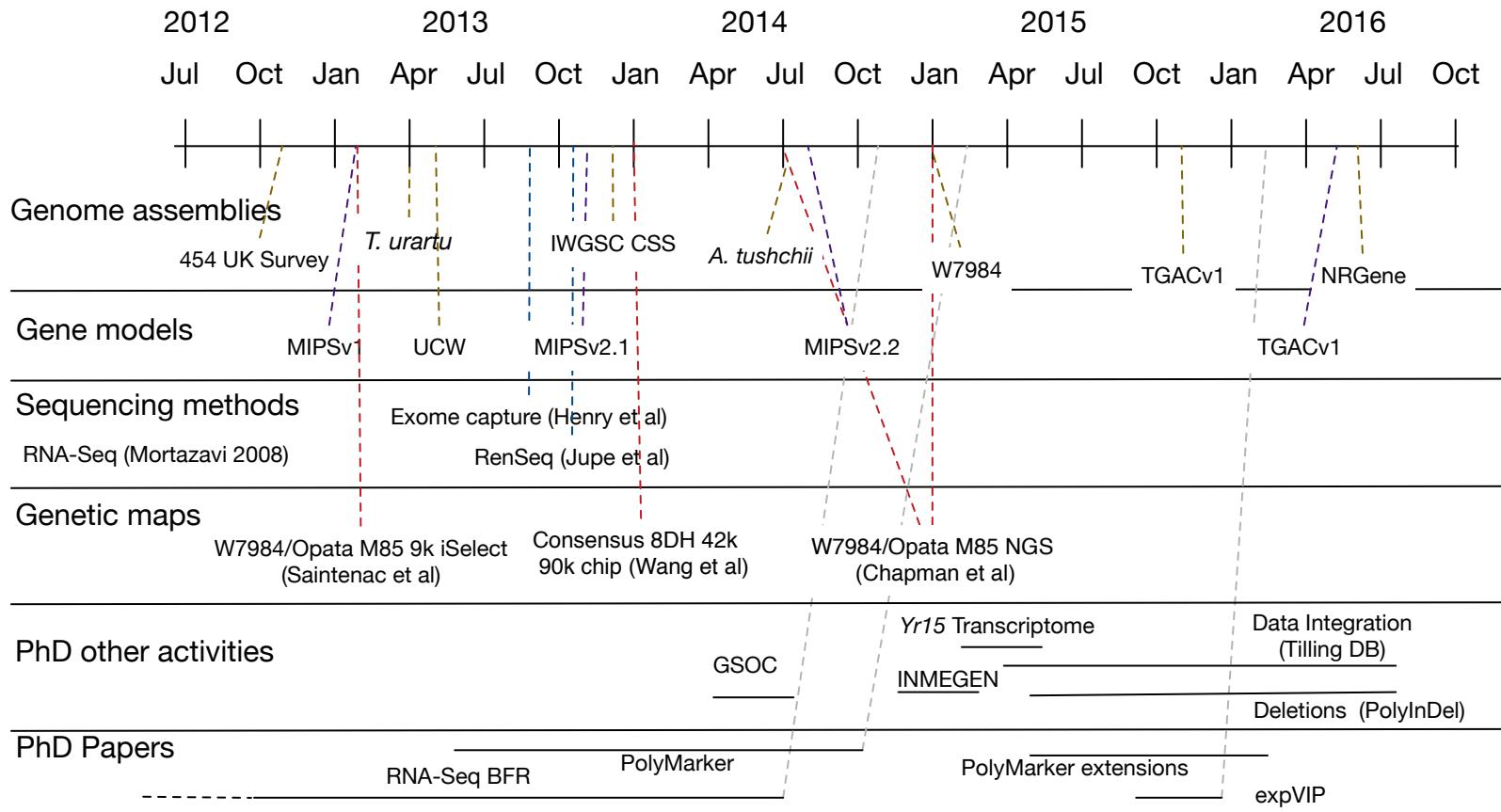


Figure 1.5: Timeline of the projects carried on during this PhD and the wheat resources that were released in the same period of time.

Chapter 2

PolyMarker: A fast polyploid primer design pipeline.

2.1 Background.

In modern breeding programs SNP markers are a prevalent technology to select seeds containing a particular locus, which is linked to a trait (ie. a marker linked to a resistance gene, see Chapter 3). SNP marker is a specific case of PCR amplification where two competing sequences from different alleles are amplified.

2.1.1 PCR amplification.

In general, PCR amplification is a technique that can be used to copy several times a fragment of DNA. To start the amplification a pair of sequences (left and right primers) on each side of the target sequence is required. The sequence between primers is copied thanks to the DNA polymerase, an enzyme that moves along the DNA strand making a copy (product). The process starts when the DNA molecule is melted in individual strands with an increase of temperature. Then, the temperature is dropped so the primers anneal to the DNA strands. At this point, the polymerase starts extending the strand from the 3'-end of the primer. The temperature is raised again to separate the new strand from the original DNA and lowered again to get the right primer to anneal to the new product. Then, in the extension step, the amplification occurs until the end of template sequence, were the 5'-end of the left primer was



Figure 2.1: PCR Diagram. PCR is used to amplify a region of the DNA (green bar) target to amplify (product; red line) is found by a pair of primers (blue lines). The 3' and 5' represent the orientation of the primers.

originally located. This process is repeated several times to increase the representation of the target DNA (Figure 2.1).

2.1.2 KASP assays.

A technology used for genotyping SNP markers is Kompetitive Allele Specific PCR (KASP) assay, the original target technology for Poly-Marker. The assay consists on triplets of primers, having a primer for each allele and a common primer that will amplify the alleles regardless of the allele. The allelic primers have at the 5'-end a tail, HEX (5' GAAGGTCGGAGTCAACGGATT 3') or FAM (5' GAAGGTGAC-CAAGTTCATGCT 3'), which is used to distinguish between them (Figure 2.2a). The KASP mix contains complementing oligos to the HEX and FAM tail, which contain a dye that is only visible when the corresponding allele has amplified. The intensity of each dye is used to measure relative amplification of each allele. In KASP assays, the distance between the left and right primers is kept as short as possible, to avoid having an extension step. As the primers are around 21-25bp, the minimum product size is between 42-50bp, with products rarely going over 75bp. Samples with the same genotype cluster: Samples on each axis correspond to homozygous individuals and samples clustered between the homozygous clusters are heterozygous (Figure 2.2b). If the experiment fails, because poor amplification or because all the samples have the same genotype, that means that there are no distinguishable clusters amongst the samples (Figure 2.2c; LGC Genomics 2014).

2.1.3 Genome specific primers.

One of the main challenges of working with polyploid species is the design of genome specific molecular markers. In hexaploid wheat, most

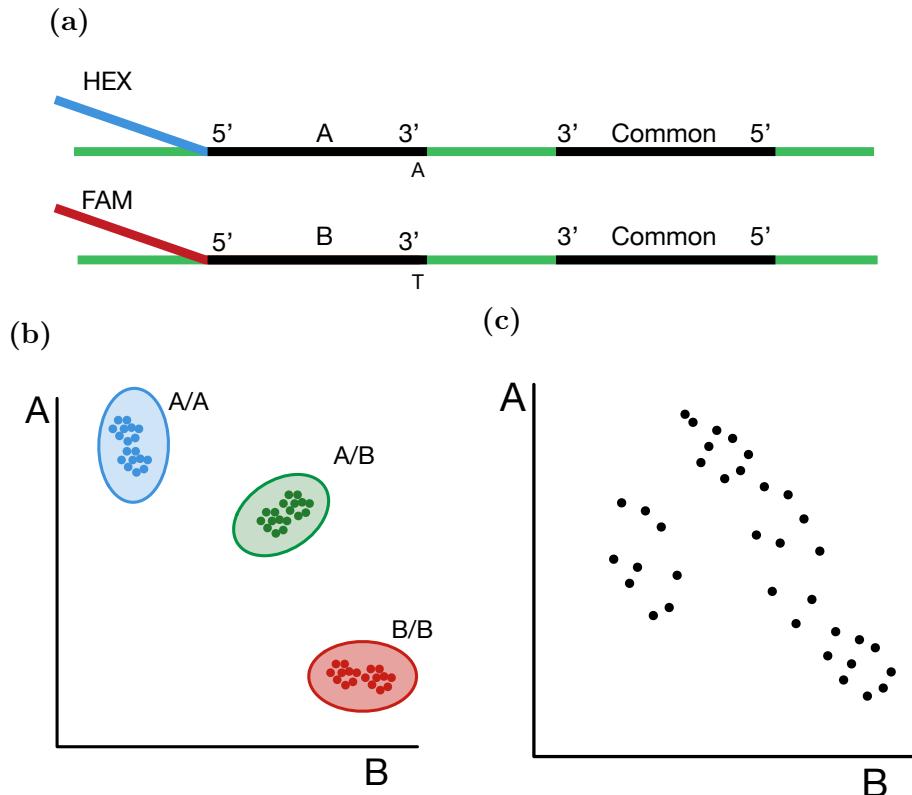


Figure 2.2: KASP Assays (a) A KASP assay consists on three primers. Primers A and B are specific for certain allele and the HEX and FAM tails are added at the 5'-end on each primer. The common primer amplifies both possible products. The SNP is an A/T, the only difference between alleles. (b) Ideal KASP results are obtained with samples containing only homozygous samples. The samples containing A allele clusters on the top-left (blue), the B allele cluster on the bottom-right (red) and the heterozygous cluster between the homozygous clusters (green). Each dot represent a sample and the axes are the relative intensity of amplification of each allele. (c) KASP results in a failed experiment, clear clusters between samples are missing.

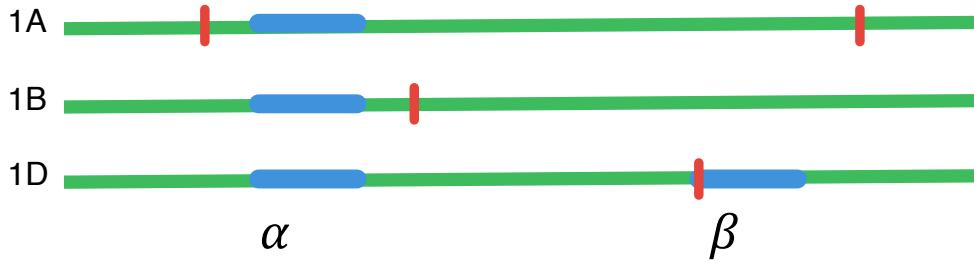


Figure 2.3: Target of genome specific primer. Primers selected randomly (blue lines) can bind to any of the three homoeologous regions if they fall on regions without variations between them (red vertical lines). The α primer doesn't contain any variation between chromosomes, hence it will bind to the chromosomes 1A, 1B and 1D. The β primer has a variation specific to the D genome, hence it will only amplify the 1D chromosome.

of the genes have at three homoeologues copies, one for each genome (See section 1.1.1). The similarity between homoeologues is around 98%, which represent around 1 mismatch for every 50 bp. This means that a primer in a conserved region of 21 bases targets any of the homoeologues if it does not have variations on it. In Figure 2.3, variations between genomes are represented with red lines, which are randomly distributed across homoeologues. The α is randomly generated using the sequence of chromosome 1D, however, because it doesn't have any variation specific to the D genome, products from it can amplify any of the genome. On the contrary, the β starts with a base that has a base that is unique to the D genome, hence the product is genome specific.

A variation between homoeologues in the primers is not enough to guarantee that the amplification is going to be genome specific. The polymerase is more specific to variations were the amplification starts, so variations in the 3'-end improve the specificity of primers (Huang and Brûlé-Babel, 2010). Hence, when designing genome-specific assays the specificity of the primers is scored according to the position of the variation as: strong, when the variation is on the 3'-end; OK, when the variation is on the 2nd position; no too strong when the variation is on the 3rd position; and not specific when the variation occurs after the 3rd position (Figure 2.4).

To ensure that all the constraints needed to produce pairs, the following steps need to be done:

1. First, a global alignment of the target sequence is used to find all the homoeologues and paralogues in the reference genome. This

```

Chromosome 1A  cgcatttgcgcgcgataccggcgctGtggaatatggcagcgaaggcgtg
Chromosome 1B  cgcatttacgcgcgcgataccggcgctTtggaatatggc---gaaggcgtg
Chromosoom 1D  c--atggcgatccggcgctGtggaatatggcagcgaaggcgtg

cgataccggcgctTtgg  Mismatch at 3' end = Strong specificity
cgataccggcgctTtgg  Mismatch at 2nd position = OK specificity
cgataccggcgctTtgg  Mismatch at 3rd position = specificity not too strong
cgataccggcgctTtgg  Mismatch at 4th position = does not provide specificity

```

Figure 2.4: Effect of position of variation on primer specificity. Several candidates to design a genome specific primer for chromosome 1B. The T highlighted in blue is a variation unique to the target chromosome. The closer the T providing specificity is to the 3' of the primer, the more specific it is.

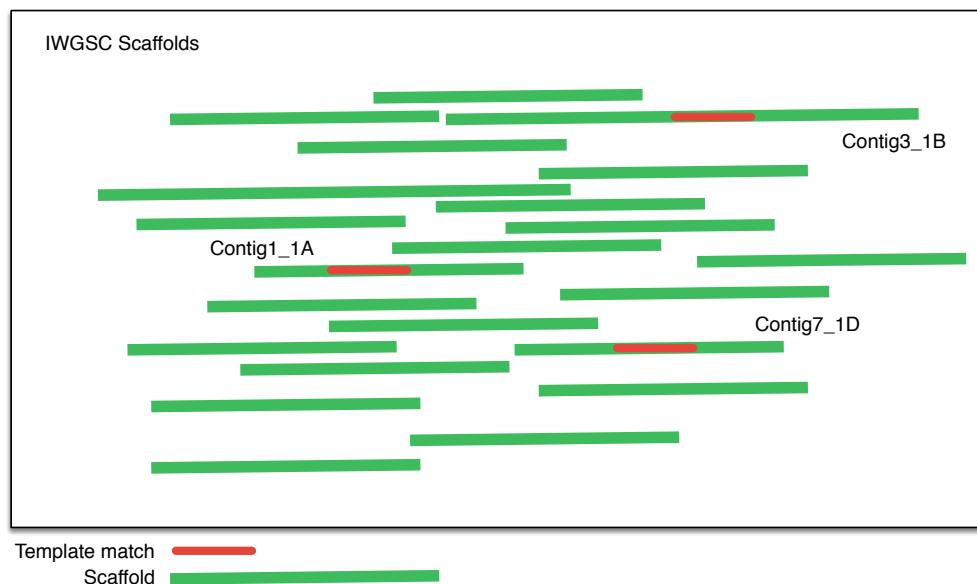


Figure 2.5: Global search of templates in the reference contigs.

is done with tools like `blast` (Altschul et al., 1990), `blat` (Kent, 2002) or `exonrate` (Slater and Birney, 2005). All of these tools take a reference sequence and make an index of the database to speed up the search of the queried sequence 2.5. Since some of the sources of SNPs come from transcriptome data and gene references, the original sequence may go over the intron-exon junction (see Section 1.4.1). The results are aligned to the target and may include sequences only from one exon, but not the adjacent intron, hence it is necessary to make a local alignment to ensure that corresponding bases are aligned correctly.

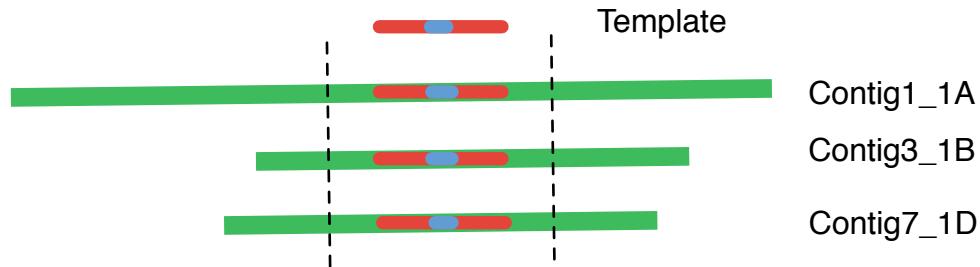


Figure 2.6: Selected regions around the SNP on every chromosome. The blue line represents the position of the SNP.

2. To put all the sequences in the appropriate context, a local alignment is done (Figure 2.6). This is done by extracting all the hits (matches) to the target reference and using a program like `mafft` (Katoh and Standley, 2013) or `clustal` (Higgins and Sharp, 1988). These tools are based on aligning all the possible sequences in pairs all the possible pairs. The distance between pairs is calculated to find which sequences are closer to each other, and then the process is repeated to refine the alignments until a consensus alignment is reached. This is useful on the context of genome-specific primer design because to correct the alignment on the presence of small insertions and deletions (indels).
3. Finally, the primers are validated to conform physiochemical properties that ensure the amplification. The melting temperature needs to be in the range where the DNA will separate, but not too high that the temperature will damage other elements in the reaction, such as the polymerase. Also, the primers must avoid sequences that self-bind, hairpins, or bind to the complementary primer. The validation of primers based on their intrinsic properties can be done with tools like `Primer3` (Rozen and Skaletsky, 2000).

If possible,
expand
and add
diagram

2.1.4 Objective.

Most of the steps required to design genome-specific primers require different bioinformatic tools and the rules to improve the efficiency of the primers are established. The objective of PolyMarker is to automate the primer design process, from a reference genome and a list of SNPs and produces genome-specific primers. The pipeline has been published in Ramirez-Gonzalez et al. (2015a).

2.2 Pipeline.

PolyMarker is an automated pipeline that takes as input a list of SNPs and a reference file and produces a list of primer triplets for SNP genotyping. The list of SNPs is first converted to a FASTA file with ambiguity codes (Cornish-Bowden, 1985) The template sequences are aligned with `exonrate` (Slater and Birney, 2005) to find the homoeologous and parologue regions to the target sequence. For my thesis, I implemented this using the IWGSC reference sequence (described in Chapter 1.5). Then, the alignment between homoeologues is refined using `MAFFT` (Katoh and Standley, 2013). A list of candidate variations is produced and used as input for `Primer3` (Rozen and Skaletsky, 2000). Finally, the output of `Primer3` is parsed to find the best primer pair that contains the targeted SNP and a base that is specific to the target genome (Figure 2.7). The pipeline is written as a Ruby script, using parsers and wrappers from BioRuby (Goto et al., 2010) and bio-samtools (Etherington et al., 2015; Ramirez-Gonzalez et al., 2012). The software is open source and released as a biogem (Bonnal et al., 2012), `bio-polyploid-tools`, the source code is available in: <https://github.com/TGAC/bioruby-polyploid-tools>.

The PolyMarker input consists on SNP list with: unique name for the marker, the target chromosome and the sequence for the marker. The alternative alleles are flanked by square brackets within the sequence. PolyMarker can take a list of several markers and design them in batch (Figure 2.8). A FASTA file is produced with all the template sequences, with the alternative alleles substituted by the IUAPC ambiguity codes (Cornish-Bowden, 1985). The flanking sequence surrounding the SNP is limited by default to 100bp to reduce the search time and avoid missing regions that diverge near the SNP, as when the variation is near an intron-exon junction.

The template sequences are aligned to the reference using `exonrate` (Slater and Birney 2005; Figure 2.5). The following parameters are used to optimise the output:

--verbose 0 --show --alignment no --show vulgar no. To override the default output.

--bestn 20. By default, it increases the number of best hits to 20. Intuitively, it would be expected to have 3 copies, one for each homoeologue. However, the CSS assembly has some duplication in

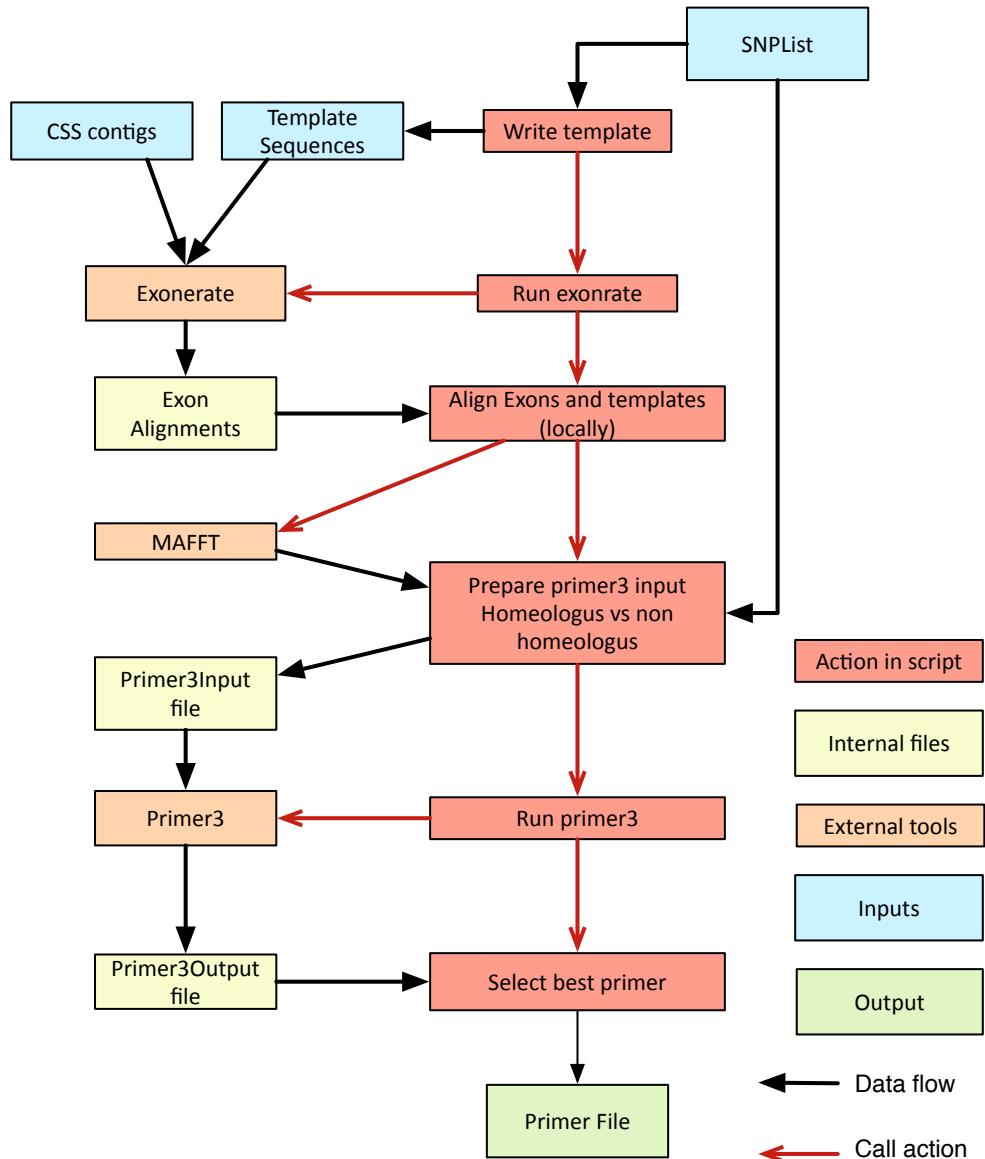


Figure 2.7: Steps and tools called by PolyMarker. The colours of the boxes represent: the step is an action inside the script(red); actions of the script(light red); temporary files(yellow); inputs(blue) and; outputs(green)

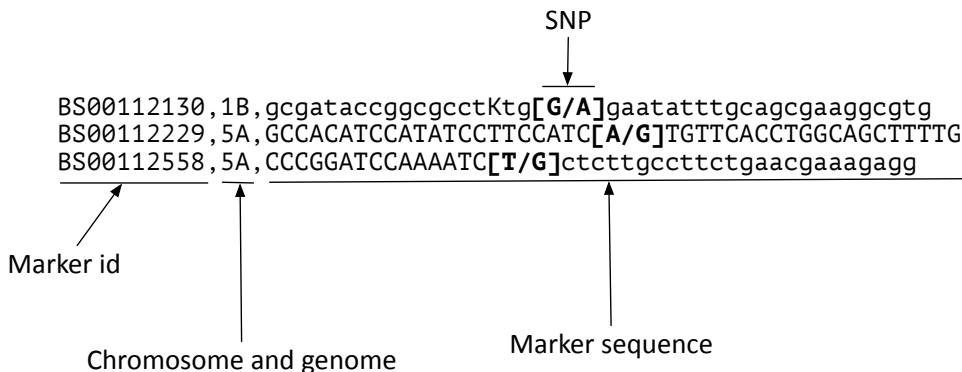


Figure 2.8: PolyMarker input. The alternative alleles are surrounded by brackets. The rest of the figures are based on BS00112130, renamed as SNP-1.

the scaffolds and it is possible to find paralogues elsewhere in the genome.

--model est2genome. To allow the search of sequences coming from transcripts, such as the SNPs described in Chapter 3 and in the SNP chip described by (Allen et al., 2011)

--ryo 'RESULT:\t%S\t%pi\t%ql\t%tl\t%g\t%\n'. To set the output in a tabular format that is easy to parse as follow: \%S the minimum information of the alignment, \%pi percentage of identity, \%ql query length, \%tl target length and, \%g orientation.

All the hits that contain the SNP and have a percentage of identity over 90% are extracted, this threshold allows to match homoeologs and paralogs. The coordinate of the SNP is calculated and 100bp on each flank are extracted by default, a reasonable product size for KASP assays. The flanking sequence may contain indels and the sequences do not align naturally (Figure 2.9). The following parameters can be adjusted to extend the functionality of PolyMarker: Minimum Identity to designs for organisms with homoeologous regions that are more divergent; flanking sequence for different types of primers (ie. for Sanger sequencing) and; **model** to adjust the search according to the source of the SNP (ie. if it is known that the SNP comes from DNA, **affine:local** would be a better option as **exonrate** will not pay attention to the intron-exon junctions).

Each SNP marker is represented on the **Bio::PolypliodTools::SNP** class, containing the flanking sequence, the position of the SNP, multiple alignments and primers. For each step there is a container that holds

Maybe explain the architecture before getting to the pipeline, or after

SNP-1 A	cgcattt G cgataccggcgcc K tgcgaaatattgcagcgaaggcgtg
SNP-1 B	cgcattt A cgataccggcgcc K tgcgaaatattgcagcgaaggcgtg
IWGSC-1A	cgcatttgcgcgcgcataccggcgcc T ggaaatattgcagcgaaggcgtg
IWGSC-1B	cgcatttacgcgcgcgcataccggcgcc T ggaaatattgcagcgaaggcgtg
IWGSC-1D	catttgcgcgcgcataccggcgcc T ggaaatattgcagcgaaggcgtg

Figure 2.9: Sequence of flanking regions around the SNP. The indels produce a slight shift on the sequence.

the SNP set and parses each output for all the called programs. The container for exonerate is `BIO::PolyplloidTools::ExonContainer`. The hits with the SNP are called exon henceforth, as the original design was for SNPs in gene models which may contain intron-exon junctions. The main job of the `ExonContainer` is to parse the `exonrate` output and add it to the corresponding SNP (Listing 2.1).

```
1 def add_alignments(opts=Hash.new)
2   opts = { :min_identity=>90 }.merge!(opts)
3   exonerate_filename = opts[:exonerate_file]
4   File.open(exonerate_filename) do |f|
5     f.each_line do | line |
6       record=Bio::DB::Exonerate::Alignment.parse_custom(
7         line)
8       if record and record.identity>=opts[:min_identity]
9         snp_array = @snp_map[record.query_id]
10        snp_array.each do |snp|
11          if snp.position.between?( (record.query_start + 1)
12            , record.query_end)
13            exon=record.exon_on_gene_position(snp.position)
14            snp.add_exon(exon, arm_selection.call(record.
15              target_id))
16          end
17        end
18      end
19    end
20  end
```

Each SNP contains a Hash to the best alignment to each chromosome, based on identity. When the `ExonContainer` adds an alignment, the SNP

```

SNP-1 A      cgcatttGcgcgYgcgataccggcgctKtgGaatatttcagcgaaggcgtg
SNP-1 B      cgcatttAcgcgYgcgataccggcgctKtgAgaatatttcagcgaaggcgtg
IWGSC-1A    cgcatttGcgcgCgataccggcgctGtgGaatatttcagcgaaggcgtg
IWGSC-1B    cgcatttAcgcgCgataccggcgctTtgGaatatttgc---gaaggcgtg
IWGSC-1D    c---atttGcgcgTgcgataccggcgctGtgGaatatttcagcgaaggcgtg

```

Figure 2.10: Local alignment on regions around the SNP detects indels.

verifies that is the best hit for a given chromosome, to avoid scaffolds with duplicated sequences (Listing 2.2).

Listing 2.2: Method in `Bio::PolyplloidTools::SNP` that adds an alignment

```

1 def add_exon(exon, arm)
2   @exon_list[arm] = exon unless @exon_list[arm]
3   @exon_list[arm] = exon if exon.record.score > @exon_list[
4     arm].record.score
4 end

```

As it is common to have different conventions over different references on how the chromosomes are named, PolyMarker can be easily extended to parse different naming conventions. To achieve this, when the `ExonContainer` is initialised a parsing function is set up. Then, when each alignment is added, the ID of the target sequence is parsed using the custom function (Listing 2.1, line 12). An example of parsing functions for a chromosome are in Listing 2.3.

Listing 2.3: Example function that assigns a chromosome from the two first letters of the scaffold

```

1 arm_selection_functions[:arm_selection_first_two] = lambda
2   do | contig_name |
3     ret = contig_name[0,2]
4     return ret
4 end

```

To ensure that the indels between homoeologues do not produce spurious mismatches a local alignment is produced with MAFFT (Figure 2.10). The arguments used are the recommended ones in the manual for small number of sequences:

--maxiterate 1000. The local alignment is defined up to 1000 times.

--localpair. Compares all the possible pairs of alignment to each other

--quiet. To reduce the size of the logs.

The class `Bio::PolyplloidTools::SNP` has the method `aligned_sequences` which executes MAFFT for the best hit on each chromosome to the marker. The first time it is invoked it stores the result as an attribute (Listing 2.4). This approach hides the execution of the local alignment as an attribute and it avoids executing it several times when calculating the variations between homoeologues.

Listing 2.4: Method in `Bio::PolyplloidTools::SNP` that calculates the local alignment

```

1 def aligned_sequences
2   return @aligned_sequences if @aligned_sequences
3   options = ['--maxiterate', '1000', '--localpair', '--quiet']
4   mafft = Bio::MAFFT.new( 'mafft' , options)
5   report = mafft.query_align(sequences_to_align)
6   @aligned_sequences = report.alignment
7   @aligned_sequences
8 end

```

PolyMarker searches across each base in the local alignment to identify the variations across homoeologues and the target marker. A mask is produced to highlight the bases with a variations, Figure 2.11, on the following categories:

Specific	Homoeologous polymorphism which is only present in the target genome (upper case).
Semi-specific	Homoeologous polymorphism is found in 2 of the 3 genomes, hence it discriminates against one of the off-target genomes or when not all the homoeologous sequences were found (lower case).
Non-specific	No variation is found across homoeologues (-).
Homoeologous	The target SNP is present across different chromosomes, so candidate SNP markers on this category are not expected to be reliably identifying the allele as these are not necessarily varietal polymorphisms. (:).

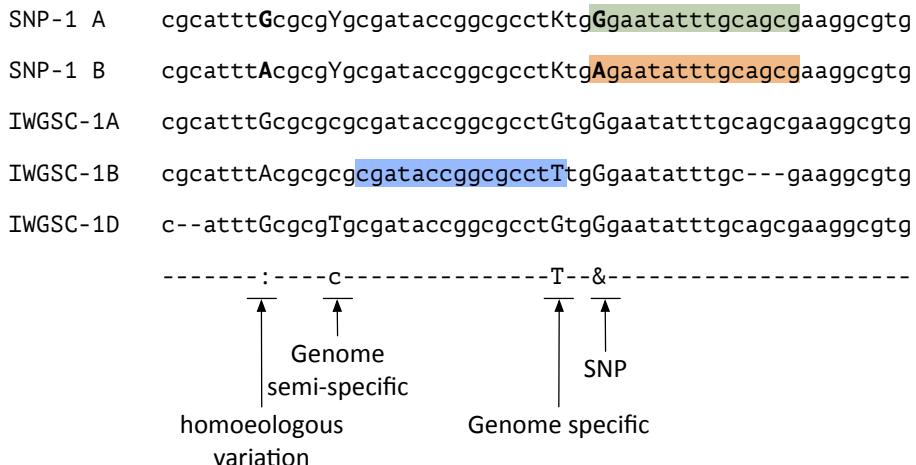


Figure 2.11: Alignment with mask and primer candidates. The green and light red boxes highlight the allele specific primers. The blue box highlights a genome specific primer.

Non-homoeologous The target SNP is not present across chromosomes, so it is most likely a varietal polymorphism which can be used to identify alternative alleles in the position.(&).

To generate the mask the following logic is followed:

1. The aligned sequence of the target chromosome is set up as the default mask (Listing 2.11, line 5).
2. Then each position in the mask is iterated base per base (Listing 2.11, line 7).
3. A count of how many bases are the same across the chromosome and how many from the same chromosome group (defined by the first letter in the parsed chromosome) and how many chromosomes have local alignment (excluding indels; lines 9-18).
4. A position is labelled as uninformative (-) when the position does not have any different bases, the sequence is only available from the target chromosome or there are unknown bases on that particular position (any chromosome has an N on the given position; line 19).
5. When no alignment is present at all, the mask is filled with * (line 20). This allows to identify the cases where only the initial marker sequence is available.

6. If the target chromosome has an unique variation, the base is converted to upper case (line 22). This implicitly leaves as a lower case the semi-specific variations. The `genomes_count` is a variable set at initialisation time and keeps track of the number of expected alignments from the target group. This allows to use the same code for any level of polyploidy.
7. At the position of the SNP, the special symbols are setup (lines 23-30)
 - (a) By default, the SNP position is labelled as & (line 24).
 - (b) All the observed bases, except the one in the target chromosome, are collected and converted to an IAUP ambiguity code (Cornish-Bowden, 1985) (lines 26-28). If the bases in the SNP are contained in the ambiguity code the SNP is marked as homoeologous (:; line 29)

When designing SNP markers the aim is to have the amplification as specific as possible. To improve the specificity of the assays, polymarker categorises all the possible primers as Specific; Semi-specific or; Non-specific. The candidate primer pairs are then evaluated with `Primer3` (Rozen and Skaletsky, 2000). `Primer3` receives a file with the preferences to design the markers, for PolyMarker the following preferences are set up:

PRIMER_PRODUCT_SIZE_RANGE=50-150. This is a reasonable size for KASP markers, as the technology does not have an extension step.

PRIMER_MAX_SIZE=25. KASP primers are usually between 21 and 25 bases.

PRIMER_LIB_AMBIGUITY_CODES_CONSENSUS=1. To ensure that bases with ambiguity code are matched between primer pairs.

PRIMER_LIBERAL_BASE=1. To allow the use of ambiguity codes in the sequence

PRIMER_NUM_RETURN=5. The maximum number of primer candidates.

To design a different kind of primers it is possible to have a different set of preferences by feeding a standard `Primer3` preferences file with the option `--primer_3_preferences FILE`.

The input file for `primer3` also include the template sequences with an ID. To keep track of what kind of marker each position will produce the ID field has the name of the primer and the specificity of the starting position of the common primer. The starting position of the primers is forced with the options `SEQUENCE_FORCE_LEFT_END` and `SEQUENCE_FORCE_RIGHT_END` on the specific and semi-specific positions. For the non specific positions only the `SEQUENCE_FORCE_LEFT_END` is given to make a full search of candidates.

Listing 2.5: Method in `Bio::PolyploidTools::SNP` that calculates the mask of the alignment

```

1 def mask_aligned_chromosomal_snp(chromosome)
2   names = exon_sequences.keys
3   parentals = parental_sequences.keys
4   local_pos_in_gene = aligned_snp_position
5   masked_snps = aligned_sequences[chromosome].downcase
6   i = 0
7   while i < masked_snps.size
8     different = cov = from_group = Count = 0
9     names.each do | chr |
10       if aligned_sequences[chr] and aligned_sequences[chr]
11         [i] != '-'
12       cov += 1
13       nCount += 1 if aligned_sequences[chr][i] == 'N' or
14         aligned_sequences[chr][i] == 'n'
15       from_group += 1 if chr[0] == chromosome_group
16       if chr != chromosome
17         different += 1 if masked_snps[i].upcase !=
18           aligned_sequences[chr][i].upcase
19       end
20     end
21     masked_snps[i] = '-' if different == 0 or if cov == 1
22       or nCount > 0
23     masked_snps[i] = '*' if cov == 0
24     expected_snps = names.size - 1
25     masked_snps[i] = masked_snps[i].upcase if different ==
26       expected_snps and from_group == genomes_count
27     if i == local_pos_in_gene
28       masked_snps[i] = '&'
29     bases = ''
30     names.each do | chr | { bases << aligned_sequences[
31       chr][i] if aligned_sequences[chr] and
32         aligned_sequences[chr][i] != '-' }
33     code_reference = 'n'
34     code_reference = Bio::NucleicAcid.to_IUAPC(bases)
35       unless bases == ''
36     masked_snps[i] = ':' if Bio::NucleicAcid.is_valid(
37       code_reference, original) and Bio::NucleicAcid.
38       is_valid(code_reference, snp)
39     end
40   i += 1
41 end
42 end
43 masked_snps
44 end

```

The class `Bio::DB::Primer3::Primer3Record` is used to keep the details of all the primers generated by `primer3` for each template. In order to prioritise which primer is selected as the best primer on for each SNP, each `Primer3Record` is scored according to their type and the product length (Listing 2.6). By default, more priority is given to the specific, semi-specific and non-specific primers, in that order. In case of having more than one primer pair with the same specificity, the one with the shortest product length is chosen (Listing 2.7).

Listing 2.6: Method that calculates the score of a primer
`Bio::DB::Primer3::Primer3Record`

```

1 def score
2   ret = 0
3   ret += @scores[type]
4   ret -= product_length
5   ret
6 end

```

Listing 2.7: Initialisation of the `Bio::DB::Primer3::Primer3Record` class, including the default score weights

```

1 def initialise
2   @properties = Hash.new
3   @scores = Hash.new
4   @scores[:chromosome_specific] = 1000
5   @scores[:chromosome_semispecific] = 100
6   @scores[:chromosome_nonspecific] = 0
7 end

```

Finally, the best primer for each marker is produced and a CSV file is produced with the following columns:

Marker The ID of the Marker

SNP The position of the SNP in the original sequence and the kind of SNP

RegionSize The size of the original sequence tested, up to the maximum size including the flanking sequence.

chromosome The target chromosome

total_contigs How many contigs mapped to the SNP. If it is more than the expected by the ploidy of the organism it can show paralogues or repetitive regions/

contig_regions The locations where the marker mapped. In the format Scaffold:start-end

SNP_type homoeologous or non-homoeologus. If it is homoeologous, the SNP is probably a variation between chromosomes.

A Primer for the first allele.

B Primer for the second allele.

common Common primer that gives the specificity to the assay.

primer_type specific, semi-specific or non-specific. Depending on the rules described previously.

orientation If it is forward, the allelic primers are in the same orientation as the original sequence. If it is reverse, the common primer is in the same orientation as the original sequence.

A_TM Melting temperature of the first allelic primer

B_TM Melting temperature of the secibd allelic primer

common_TM Melting temperature of the common primer

selected_from For internal purposes, points from which of the primers was used as template.

product_size The size of the PCR product produced by the primers.

PolyMarker also produces a text file with the local alignments that contain all the positions that can produce a genome-specific primer. The file has the same format as Figure 2.11, but without the highlights. The mask is useful in case that the original assay failed, or to explore the details of the other homoeologs and paralogs which are similar to the assay.

2.2.1 PolyMarker public web service

To make PolyMarker accessible to the community, a web server that allow the submission of SNPs was developed. The web interface consists on two virtual machines, one with a web-facing interface that stores the queries, and a dedicated node to submit jobs to an HPC cluster. The on-line interface further simplifies the design of KASP assays, a process that used to take between 15-45 minutes per marker is now automated. Since the release of the public service in July 2014 until August 2016, 1,739 requests to PolyMarker have been done, demonstrating the popularity of the PolyMarker tool.

Besides the previously described output, the web interface of PolyMarker provides a graphical representation of the multiple sequence alignment and the mask used to design the primer (Figure 2.12). The visualisation consists on a table containing the primers and the BioJS component MSAViewer (Yachdav et al., 2016), that highlights the designed primers. On an ideal case, you have an SNP that is in a non-homoeologous position with a genome-specific triplet (Figure 2.12a). However, sometimes the SNP is located in an homoeologous variation (Figure 2.12b), which can signal a miscalled SNP. In some extreme cases, a SNP is located in regions that have homoeologues and paralogs in several chromosomes (Figure 2.12c), it is useful to highlight such kind of SNPs that can produce spurious amplification from non-target chromosomes. The graphical representation is helpful to understand how the primers were designed.

2.3 Applications of PolyMarker

Besides the project described in Chapter 3, PolyMarker has been used to design KASP primers for the wheat community.

2.3.1 KASP assays for public sets of SNPs

PolyMarker was used to design KASP assays for the 81,587 markers from (Wang et al., 2014), available on the PolyMarker website and in CerealsDB (Wilkinson et al., 2012). Of those markers, 40,267 where designed using the target chromosome using the genetic map provided in Wang et al. (2014). Genes without a genetic position were aligned to

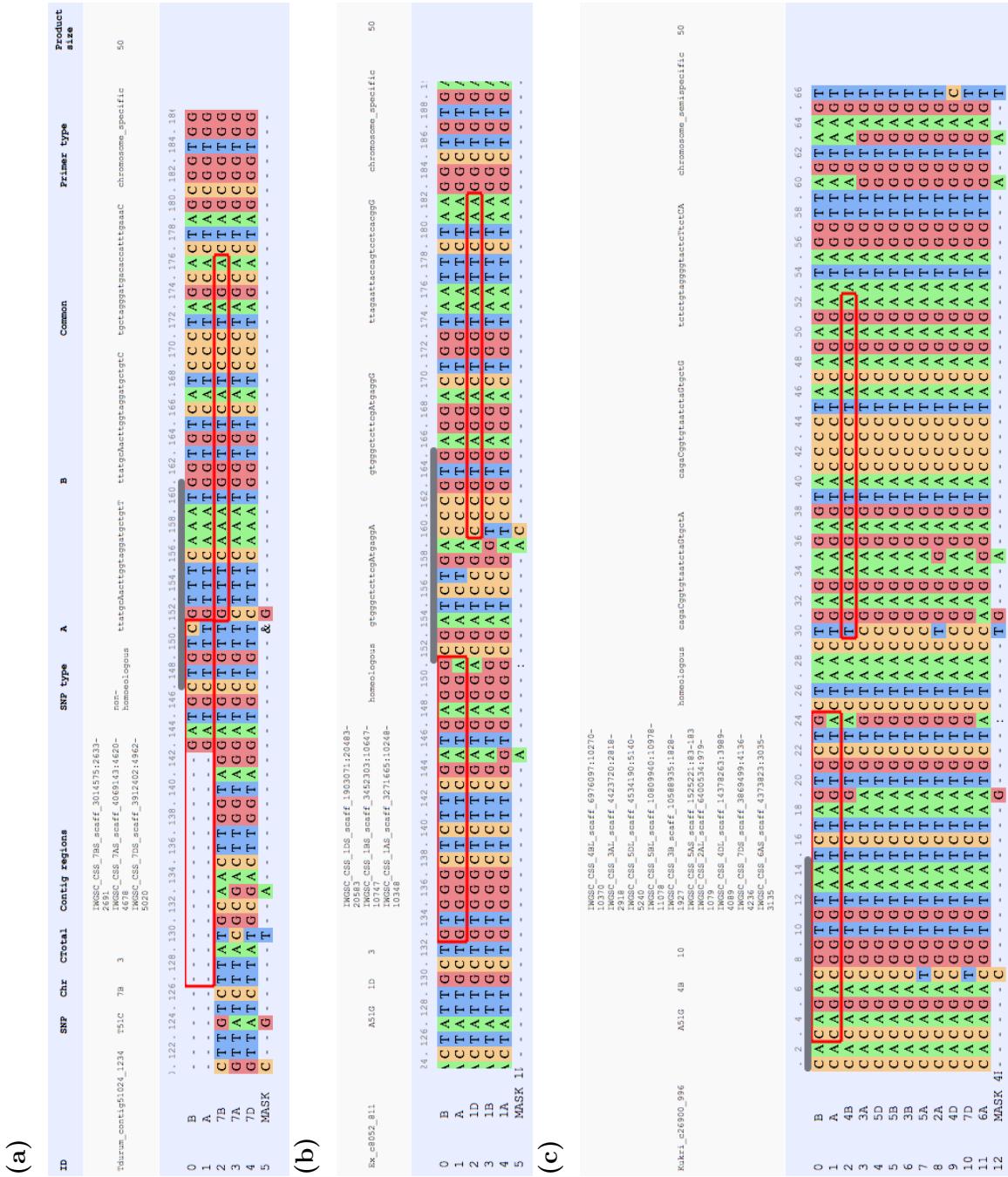


Figure 2.12: Examples outputs from the PolyMarker website. (a) The original marker sequence had the SNP near the beginning of the template, but PolyMarker used the genomic reference to complement the sequence. (b) An specific primer, but the SNP is located on the same position than an homologous variation. (c) A case where the marker sequence align to 10 different chromosomes. The SNP is also located on a position with variations between genomes.

Table 2.1: Count of KASP assays designed for the 40,267 SNP markers located in the genetic map from Wang et al. (2014). 4,228 assays did not align to the target chromosome. Not designed: Primer3 could not find viable primers flanking the SNP.

	Homoeologous variant	Varietal SNP	Percentage
Non-specific	1,765	5,857	21.15%
Semi-specific	7,942	6,907	41.20%
Specific	6,813	5,957	35.43%
Not designed	242	556	2.21%
Total	16,762	19,277	36,039

scaffolds sorted by chromosome from the International Wheat Genome Sequencing Consortium (Mayer et al., 2014) with BLAT (Kent, 2002) and the best hit was selected as the putative location. 97.5% of the assays where designed and 76% of them are semi-specific or specific, thereby improving their expected performance with respect to randomly designed primers (Table 2.1). The markers had been taken by the community, for example a subset of the designed assays was used to genotype a mapping population to find resistance to Fusarium head blight (Burt et al., 2015).

Also, PolyMarker was used to design KASP assays for the 820K SNP Axiom array described in Winfield et al. (2016). Briefly, the original set contains 819,556 SNPs called from exome capture on 43 bread wheat accessions and wheat relatives. Of those, 616,525 where mapped with `exonerate` (Slater and Birney, 2005) to the CSS scaffolds. Of those, 86.1% have an specific or semi-specific assay (Table 2.2. This set of primers is also available in CerealsDB and it provides a valuable resource to groups that want to genotype using a subset of SNPs in the array, without the need to run the Axiom array.

2.3.2 SNPs in a mutant population

PolyMarker was used to design primers to validate SNPs in a Targeted Induced Local Lesions in Genomes (TILLING) population, an approach to identify the function of genes by mutating them. Briefly, wheat lines are mutated with ethyl methanesulphonate that produce G>A or C>T mutations. The initial mutation is called M_1 and each plant is self crossed to fix the mutations. The second generation is called M_2 , and so on.

Table 2.2: Count of KASP assays designed for the 616,525 SNP markers located to a CSS scaffold from the 819,556 SNPs from Winfield et al. (2016) Not designed: Primer3 could not find viable primers flanking the SNP.

	Homoeologous variant	Varietal SNP	Percentage
Non-specific	20,189	56,516	12.44%
Semi-specific	167,018	132,145	48.52%
Specific	139,202	92,487	37.58%
Not designed	3,116	5,852	1.45%
Total	329,525	287,000	616,525

With each generation the originally heterozygous mutations get fixed and become homozygous. In the process, some mutations are lost. For this experiment, three M_5 lines were sequenced with exome capture. The purpose of the experiment was to assess the feasibility of exome capture for call for SNPs.

To validate the SNPs detected at different levels of coverage and allele frequencies 150 assays were designed. The assays were tested on the M_5 used for SNP calling and on the progenitors at M_2 , M_3 . Most of the SNP calls with more than 8 variant calls or an allele frequency over 0.8 were validated (Table 2.3). At the same time, only 27% of the SNPs with an allele frequency of 0.6 and 17% of the cases with seven or less variant reads were successful. (King et al., 2015). On this experiment PolyMarker was useful on validating and calibrating the minimum coverage to call SNPs reliably.

On a follow-up experiment consisting of 1,200 Cadenza (Hexaploid) and 1,535 Kronos (Tetraploid) wheat lines (Krasileva et al., submitted 2016) where also validated. Genome-specific primers 172 and 80 SNP assays on 19 and 8 M_4 Cadenza and Kronos lines respectively. Of those, 71(85.5%) Kronos and 147(88.8%) of the Cadenza primers where valid assays, consistent with the pilot study (Tables A.1 and A.2).

2.4 Modifications of PolyMarker

PolyMarker is not restricted to wheat or to KASP assays, the source code is flexible and can be extended for other types of analyses. On each of the following projects, PolyMarker has been adapted to design primers in

Table 2.3: Summary table of the validation of candidate SNPs by KASP marker assays. Candidate SNPs are classified by number of supporting variant reads or by allele frequency and validated by KASP assays. Table from King et al. (2015).

Criterion	Number/ Frequency	KASP assays	Validated SNPs	Validated (%)
Variant reads	4	25	1	4%
	5	17	3	18%
	6	14	2	14%
	7	14	3	21%
	8	10	5	50%
	9	12	9	75%
	>10	35	29	83%
Allele frequency	0.2	51	2	4%
	0.4	27	13	48%
	0.6	18	9	50%
	0.8	3	2	67%
	1	31	27	87%

species where KASP has not been used before, the primers are used for regular PCR amplification, or the use of KASP is not the conventional SNP calling.

2.4.1 Deletions on a mutant population

On some of the TILLING mutant lines long deletions were detected (Krasileva et al., submitted 2016). To validate the deletions it is possible to use KASP assays to produce primers that amplify homoeologues. PolyMarker was modified to search for variations across homoeologues to select a common primer that will amplify two genomes (Figure 2.13a, b; reverse primer). On lines without the targeted deletion, the amplification corresponds to an heterozygous assay with equal signal for both the A and the B allele. (Figure 2.13c). When a deletion is present the results of the assay resemble the results for a homozygous individual, with the intensity of the assay towards the the conserved homoeologue (Figure 2.13d).

To be able to select primers that will amplify two homoeologues, the default scoring values (Listing 2.7) are changed. The altered scoring gives priority in the following order semi-specific, non-specific and specific. The

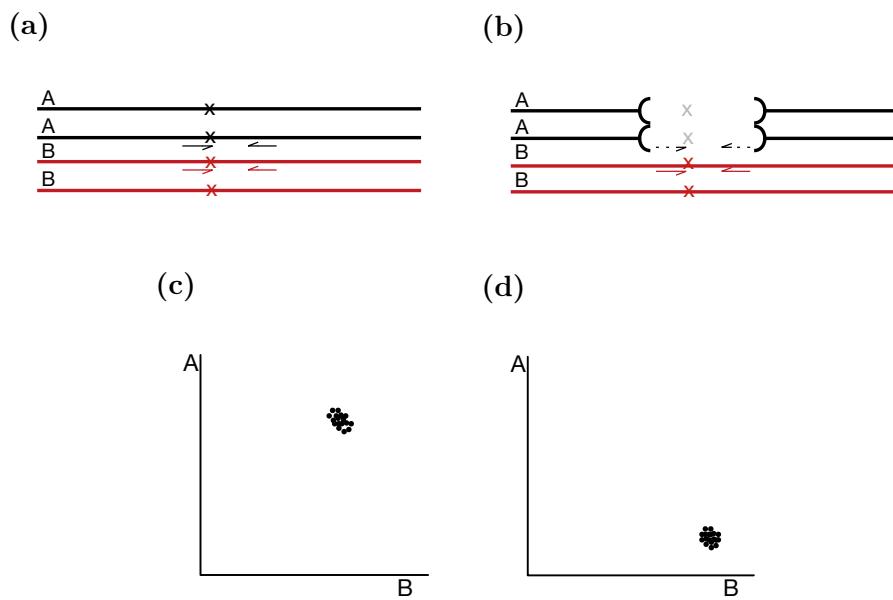


Figure 2.13: KASP assays to validate homozygous deletions. (a) Primer positions for wildtype. Red and black indicate the A and B genome respectively. Primers are indicated by arrows with the target homoeologous SNP marked by an "X"(b) Primer positions on homozygous deletion on M_4 (c) Heterozygous amplification on wildtype DNA (no deletion), including both homoeologues. (d) Homozygous amplification on deletion line, only the non-deleted homoeologue is amplified.

rest of the pipeline is unaltered, showing that the modular design allows to add new functionality without breaking the pipeline.

Listing 2.8: Score values to select semi-specific primers

```

1 kasp_container.scores[:chromosome_specific] = 0
2 kasp_container.scores[:chromosome_semispecific] = 1000
3 kasp_container.scores[:chromosome_nonspecific] = 100

```

A set of KASP assays for the the deletions and mutations located on the same chromosome where designed to validate 11 homozygous deletions on *M₄* plants. In all cases the segregation of the mutations was as expected, except for a predicted heterozygous mutation that was called as homozygous. Also, all the KASP assays that contained a deletion were called homozygous, as expected. To ensure that the calls did not come from a single cluster, 4 wildtype plants were genotyped and the markers for deletions where called as heterozygous. An example of a validated deletion and the surrounding mutations, with the calls for each individual is shown on Table 2.4.

2.4.2 Genotyping *Puccinia striiformis* f. sp. *tritici* isolates.

In Hubbard et al. (2015), *Puccinia striiformis* f. sp. *tritici* (PST) isolates were sequenced and assigned to clusters, according to their genotype. The clusters are useful to monitor the changes in the pathogen population, which can be used to predict if certain wheat lines will be resistant to the isolates in the field. PST is a dikaryon, an organism with two nuclei, each one containing a single haploid of chromosomes. For PolyMarker it can be treated as diploid, so the `--genomes_count 1` argument was used. PolyMarker was used to design primers for PST, using the assembly PST-130 Cantu et al. (2011a). As the assembly is fragmented, an *ad hoc* function was used to always get the name of the assembly (Listing 2.9). Out of 15 assays, 11 can be used to identify to which cluster of isolates a sample is likely to belong, Table 2.5. Until this study, previous method to genotype of PST was SSR markers.

Table 2.4: Validation of homozygous deletions on line Cadenza0423.

Marker	Deletion	chr	cM	1	2	3	4	5	6	7	8	9	10	11	12	C	C	C	Result
5BL_2297308_Cadenza0423_12664_C12664T	-	5B	4.551	X	X	-	X	X	X	X	X	X	-	X	Y	Y	Y	HOM Mutation	
5BL_10812849_Cadenza0423_5664_G6664T	-	5B	38.769	X	X	-	X	X	X	X	X	X	-	X	Y	Y	Y	HOM Mutation	
5BL_10825062_Cadenza0423_7917_G7917A	-	5B	38.769	X	X	-	X	X	X	X	X	X	-	X	Y	Y	Y	HOM Mutation	
IWGSC_CSS_5BL_scaff_10847976_27068-27231	+	5B	38.769	X	X	-	X	X	X	X	X	X	-	X	H	H	H	Hom Deletion	
IWGSC_CSS_5BL_scaff_10847976_28118-28674	+	5B	38.769	X	X	-	X	X	X	X	X	X	-	X	H	H	H	Hom Deletion	
5BL_1083722_Cadenza0423_4616_G4616A	-	5B	39.905	X	X	-	X	X	X	X	X	X	-	X	Y	Y	Y	HOM Mutation	
5BL_10891320_Cadenza0423_18847_C18847T	-	5B	45.594	Y	Y	-	Y	H	X	X	Y	H	-	H	Y	Y	Y	HET Mutation	

Listing 2.9: Function that always returns PST130 as chromosome

```
1 arm_selection_functions[:pst130] = lambda do |contig_name|
2   return "PST130"
3 end
```

2.5 Discussion

PolyMarker is a tool that was born as part of the validation of the SNPs found in Chapter 3. Originally, the primer design was done manually, a slow, error-prone and, repetitive process. The steps require the use of several bioinformatics tools, but once I understood the primer design steps I decided to automate the process. Since designing genome-specific primers is a common task in wheat research and breeding, the community showed interest on the tool and I decided to refine it and make it open source. PolyMarker has been used successfully in several projects and it even allowed the novel use of KASP assays to validate long deletions in polyploids.

As a common source of SNPs are gene models, designing primers directly from the sequence flanking the SNP may run over the intron-exon junctions, producing primers that will not amplify on genomic DNA. To be able to use DNA I had to identify on which hit the SNP was located, the internal coordinate and a mapping coordinate to the original sequence. With this dual coordinate system I was able to design primers in the genome space, even when the origin was a transcript.

In order to be able to represent more than one base at the same tame, the IUAPC ambiguity codes (Cornish-Bowden, 1985) were useful in the development of PolyMarker. With an ambiguity code, the template for the search can contain the SNP. Also, the codes were useful when representing all the observed bases on each coordinate in the local alignment.

The ideas behind PolyMarker had been taken by other projects like the scripts described in Ma et al. (2015) and the corresponding web interface, GSP (Wang et al., 2016). Briefly, GSP does uses **blast** to search in a local database to find all the homoeologous regions and provides a diagram with the bases that are genome specific. It then allows the user to select a primer pair according to the constrains for their individual

Look at
<http://www.ncbi.nlm.nih.gov/p>
to get
some
way to
compare
SSR vs
SNP
markers

Table 2.5: PolyMarker used to genotype PST. The X and Y represent the two possible alleles. X:X and Y:Y correspond to homozygous call of the corresponding allele. X:Y correspond to heterozygous calls. The '-' symbol correspond to failed assays.

experiment, like product size. The advantage over PolyMarker is that it allows to pick arbitrary primers, at the cost of having a step for manual selection of the pair. Recently, LGC also developed a program (MAG-ICBOX) that require a SNP sequence, does the alignment and selects primers with a genome specific anchor. As PolyMarker, it produces a local alignment with the genome-specific bases (Curry et al., 2016). On personal communications in conferences I had found out that LGC, the company behind KASP, uses PolyMarker to design primers. Also Bayer has an in house implementation of the algorithm.

As the code is open source, anyone can see the implementation details and extend the code for different types of primers. A successful modification to PolyMarker was to be able to design primers to detect homozygous deletions with KASP assays, despite the fact that neither KASP or PolyMarker were designed for deletions. The modularity of the code permits to swap components with relatively little effort.

The current web interface of PolyMarker is limited to KASP assays, however the command line version is more flexible and has been used to design primers for PCR amplicons, capillary sequencing and on other organisms. However, to install the command requires a linux machine and some knowledge on the command line.

Overall, PolyMarker provides an useful resource to the wheat community, as the primer design process is now streamlined. As new references of wheat come available, PolyMarker should be updated to work with pseudomolecules and the web interface updated accordingly. The source code of PolyMarker is open source and available on <https://github.com/TGAC/bioruby-polyplloid-tools>.

Chapter 3

Genetic map of *Yr15* with RNA-Seq.

3.1 Background.

Wheat breeding programs aim to improve the wheat lines available for production. One of the traits desired in an elite line is the resistance to pathogens, such as *Puccinia striiformis* f. sp. *tritici*, the fungi responsible of yellow rust. A source of resistance genes is are introgressions from other species, such as *Triticum dicoccides*. In the University of Sydney a collection of Near Isogenic Lines (NILs) with introgressions to several Yellow Rust resistance genes on a susceptible background were developed (Wellings and McIntosh, 1998). On this chapter the NIL for the *Yr15* locus is used to produce a mapping population to improve diagnostic markers.

3.1.1 Segregation on F_2 populations.

Line selection can be done with molecular markers that can be used to test if certain allele is present in a line, without the need to do a phenotype. To find which regions are linked to a trait the use of F_2 mapping populations is a common practice. The population is produced by crossing two homozygous parents (P_1 and P_2) with different alleles, A/A (dominant) and a/a (recessive). When the trait is dominant and has a mendelian segregation, the F_1 population show the dominant trait, as it has a copy of each allele (A/a). The F_1 is then self-crossed to and the population segregates with a ration 1:2:1, dominant:heterozygous:recessive

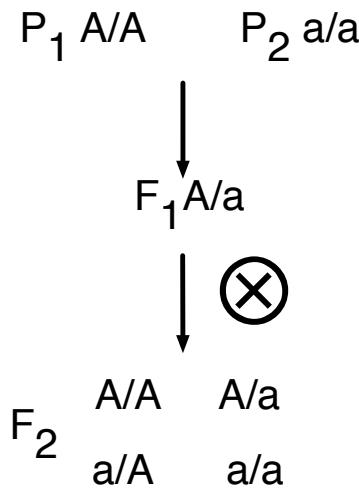


Figure 3.1: The cross of two homozygous parents, P_1 and P_2 , with a dominant and a recessive allele of a gene produce an heterozygous F_1 . The F_1 crossed with itself produce a segregating F_2 population with a 1:2:1 ratio (A/A:A/a:a/a). The upper and lower cases represent dominant and recessive alleles

respectively. This generates a population with a phenotype ratio of 3:1 (dominant:recessive), since the effect of the recessive allele is masked by the dominant gene (Van Ooijen and Jansen 2013a; Figure 3.1).

3.1.2 SNP calling

Bulk Segregant Analysis (BSA) consists on pooling the DNA of individuals with contrasting phenotypes (Michelmore et al., 1991) on a segregating population. The bulks show as heterozygous except for the region that is linked to the trait of interest. This approach can be used to identify SNPs using High Throughput Sequencing, such as: exome capture (Hodges et al., 2007), RNA-Seq (Pickrell et al., 2010), whole genome resequencing (Schneeberger et al., 2009), among others.

To Call for SNPs from RNA-Seq a reference transcriptome is used as target to align the reads. The Bulk Frequency Ratio (BFR) methodology can work on organisms that have more than one pseudo genome with not all the genes, homoeologues or paralogues, characterised independently; it works with a single reference collapsing similar regions. The UniGenes database, from NCBI, contains the genes of each species with all the variations of each gene automatically collapsed and represented with the longest cDNA (Pontius et al., 2002). The UCW genes described in Krasileva et al. (2013) contains 94,177 models from tetraploid and hexaploid wheat, assembled and phased to separate different homoeologues. Both gene sets complement each other, however, the UCW gene models should provide an improved alignment, since the different ho-

moeologues aren't merged in a single model, a possible side effect of the UniGene pipeline.

Homoeologous variants, as exemplified by the G>T variant at position 181; K in consensus (Figure 3.2), will produce the same ambiguity code for both parental consensus sequences and can therefore be excluded. Real allelic SNPs between the parental genotypes, exemplified by the G>A variant at position 184; R in consensus, are distinguished by the presence in one, but not the other parental consensus sequence. The allelic SNPs are then examined further with the alignments of the bulks to identified the SNPs that are enriched on the resistant plants. The SNP index is the proportion of times an alternative allele is observed over the coverage at certain, in the example the the susceptible bulk has an SNP index of $1/8 = 0.125$ and $6/8 = 0.75$ for the resistant bulk (Takagi et al., 2013b). traditional The BFR are then calculated by dividing the SNP Index of sample containing the target phenotype (resistance) over the sample without the trait (susceptible), on the example is $0.75/0.125 = 6$. A high BFR suggests that the SNP is linked to the target trait (Trick et al., 2012). The implementation of the BFR analysis is detailed in Section 3.10.3 and the results on the F_2 population are discussed in Section 3.5.

3.1.3 *In Silico* mapping.

There are several layers of information that can be used to add a context to the SNPs. When the SNPs are called from genes like the UniGenes (Pontius et al., 2002) or the UCW gene models (Krasileva et al., 2013), the location of the genes can be assigned by aligning them to a genomic reference, even if it is fragmented. A source to get the order of the scaffolds are genetic maps previosly published, such as the genetic map described in Wang et al. (2014), which has the sequence of the markers available. The markers and the genes can be aligned to the scaffolds with a high percentage of identity (over 98%), to avoid them being assigned to an homoeologue or parologue region in a different chromosome. The use of genetic maps to sort genomic sequence is frequently used to produce pseudo-chromosomes on genome wide projects, usually with ad-hoc tools (Tang et al., 2015). Since the CSS assembly is quite fragmented the genetic maps don't have enough resolution to produce a pseudomolecule, however it is enough to sort the scaffolds in bins when several markers

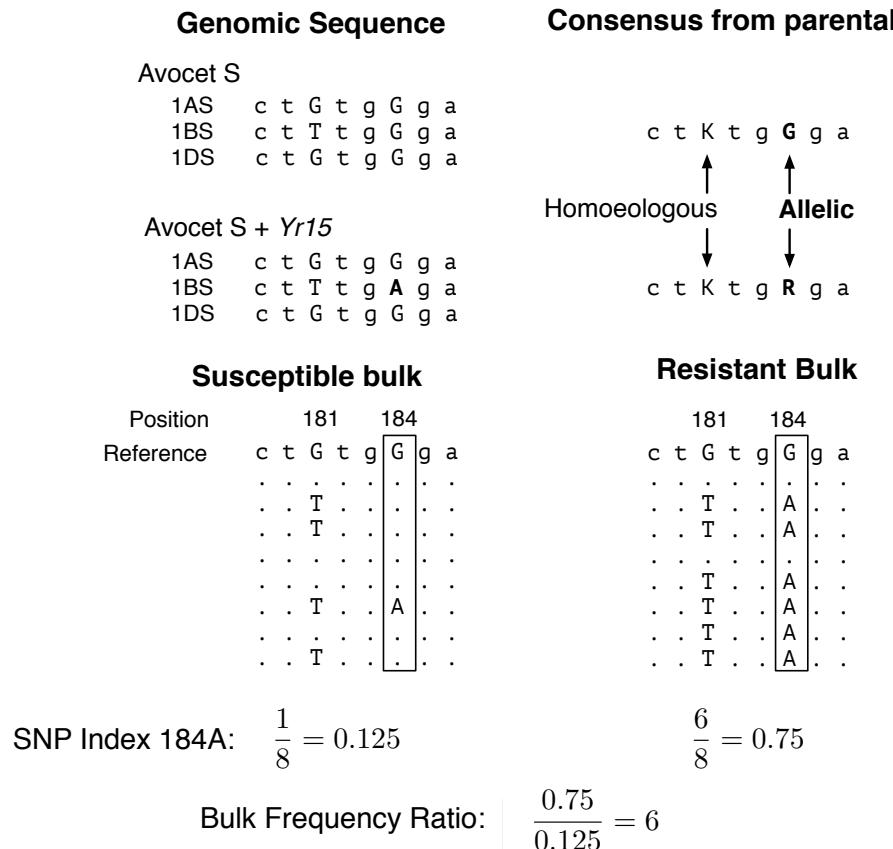


Figure 3.2: BFR formula. Illustration of a non-informative homoeologous SNP (G181T) present in both parental lines, and an informative allelic SNP (G184A), only present in the resistant progenitor Avocet S + Yr15. The consensus sequences from the parental genotypes include this information in the form of ambiguity codes (K and R, respectively). In the bulks, the individual reads align across the reference sequence, with matches indicated by dots, and polymorphisms at positions 181 and 184 indicated by the corresponding nucleotide variants at those positions. The SNP index is calculated as the frequency of the informative allelic SNP in each bulk. The Bulk Frequency Ratio is the quotient of the resistant and susceptible bulk SNP Indexes. Figure previously published in Ramirez-Gonzalez et al. (2015b).

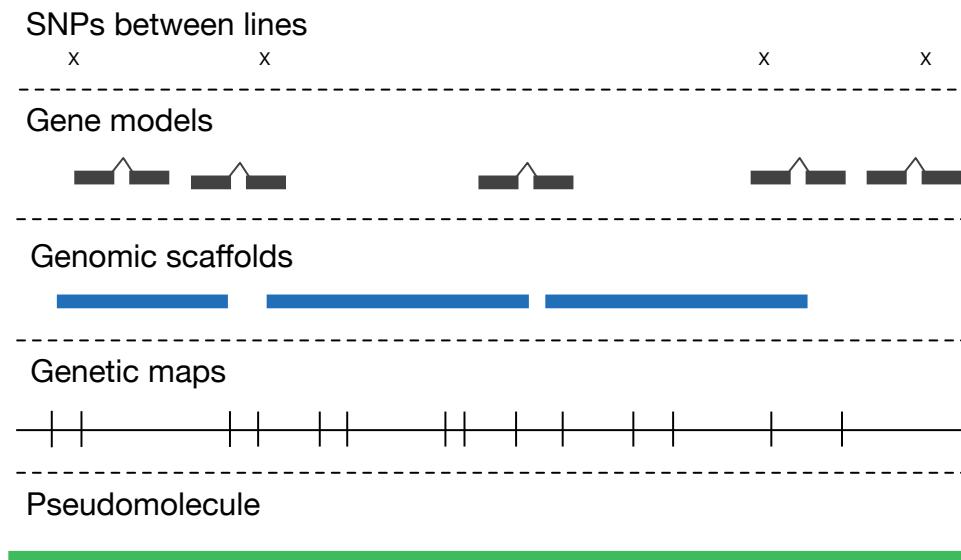


Figure 3.3: Layers of information to do *In Silico* mapping. SNPs are called from gene models. The genes and markers from genetic maps are aligned to scaffolds. The order of the markers in a genetic map can be used to sort the scaffolds.

map to the same location. In this way, it is possible to use the scaffolds as a proxy to map the genes to their genetic position (Figure 3.3). The results of mapping the genes with SNPs to the CSS assembly and the genetic map are described in Section 3.6. For a longer description of resources available for wheat see Section 1.5.

Finally, the best candidate SNPs were selected to produce a genetic map which lead to a triplet of markers diagnostic to the target locus.

The steps described in this chapter were first published in Ramirez-Gonzalez et al. (2015b) and the results of this chapter are published in Ramirez-Gonzalez et al. (2015c).

To do:
section talking about genetic map.

To do:
Microsatellites vs SNP markers.

3.2 Mapping population.

The population was developed by crossing the resistant line Avocet + *Yr15* (*Yr15*) (Wellings and McIntosh, 1998), Figure 3.4a, to the susceptible line Avocet S (AVS), Figure 3.4b. *Yr15* is a NIL of a 6th generation Back-cross (BC) and the AVS background is highly susceptible to yellow rust, hence the resistance is conferred by the *Yr15* locus. *F*₂ seeds from three independent *F*₁ plants were sown and tissue was collected, before the fungal inoculation to avoid the effect of the response on the

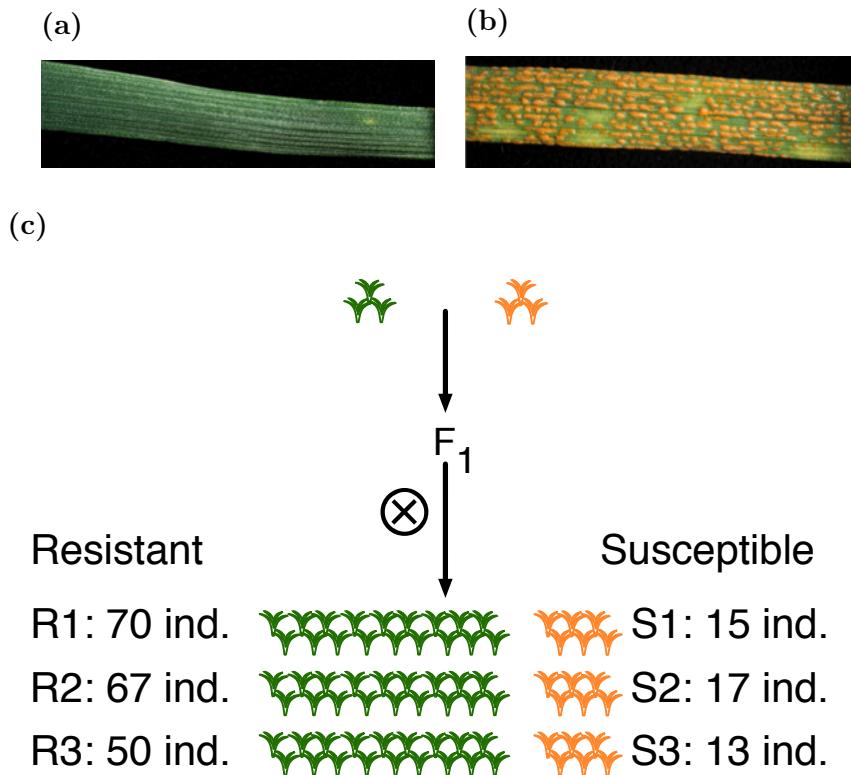


Figure 3.4: Avocet + *Yr15* F_2 mapping population. Response of (a) Avocet + *Yr15* and (b) Avocet when inoculated with *Puccinia striiformis* f. sp. *tritici* at the three leaf stage. (c) The phenotype of the F_2 population was used to produce 6 bulks, 3 resistant and 2 susceptible. The RNA was pooled in bulks accordingly. Adapted from (Ramirez-Gonzalez et al., 2015c)

gene expression. The plants were challenged at the three leaf stage as it is known that *Yr15* confers resistance in seedlings (Gerechter-Amitai et al., 1989). The expected segregation on an F_2 population is 3:1 (resistant:susceptible), since *Yr15* is a dominant gene. From the 232 plants in the F_2 population that germinated, 187 were resistant and 45 were susceptible, which deviates slightly from the expected ratio ($\chi^2 = 0.049$). Segregation distortion has been shown for the same *Yr15* donor (Randhawa et al., 2009), however the decreased number of susceptible plants can be explained by escapes in the virulence assays (i.e. plants scored as resistant without the *Yr15* locus). For this study we extracted DNA from individual plants in the F_2 population and we bulked RNA on 6 different bulks: 3 resistant and, 3 susceptible (Figure 3.4c).

Table 3.1: Arrangement and number of sequenced base pairs per sample.

Library	name	Bar code	Lane	Reads ($\times 10^8$ bp)
LIB1715	Bulk R1	ATCACG	1	0.77
LIB1716	Bulk R2	TAGCTT	1	1.20
LIB1717	Bulk R3	ACTTGA	2	0.96
LIB1718	Bulk S1	GGCTAC	2	1.64
LIB1719	Bulk S2	CGTACG	2	1.49
LIB1720	Bulk S3	GTGGCC	1	1.88
LIB1721	AvocetS	N/A	3	4.13
LIB1722	AvocetS + <i>Yr15</i>	N/A	4	3.99

3.3 Sequencing and mapping.

RNA-Seq was used to avoid sequencing the non-coding regions and reduce the search space. The sequencing of the bulks and the parents were done on a single Illumina Hi-Seq2000 each. The bulks were multiplexed and sequenced on a third of a lane each, as shown on Table 3.1. To ensure that the quality of the sequencing was good, **fastqc-0.10** (Babraham Bioinformatics, 2012) was run with its default parameters in each one of the fastq files. The GC content was around 52% in all the samples (Appendix B.2), which is expected as the sample should be of coding regions, and for wheat the reported GC content in genes is around 55%. The quality of the reads is fairly consistent, in general dropping after the base 80 across the samples (Appendix B.1).

When the analysis was started, the draft genome and the corresponding annotation where not released yet, hence gene models were used. All the samples were aligned to the Unigenes v60 (56,954 genes) and the gene models from UCW (Krasileva et al., 2013) using **BWA 0.5.9** (Li and Durbin, 2009). The alignment provided showed that a few genes were overly expressed, however we still have 22,107 and 36,808 genes, on the Unigenes and the UCW gene set respectively, with a coverage greater than 20x in the progenitor with *Yr15*. Both gene sets performed similarly in terms of the percentage of genes with reads and percentage of aligned reads. For AVS and *Yr15*, the percentage of genes with a coverage of at least 20x is 45% and 39% respectively across both references (Figure 3.5a). Since each individual bulk has a lower coverage, the susceptible and resistant reads were merged *in silico* as: (i) susceptible bulks 1 with 2 (S1 + S2) and resistant bulks 1 with 2 (R1 + R2) and (ii) all the

Table 3.2: Number of genes with a coverage over 20x, 10x and at least one read (>0x).

Coverage	Reference	R1	R2	Bulks			Bulk mixes			Progenitors	
				R3	S1	S2	S3 R1+R2	S1+S2	R1+R2+R3	S1+S2+S3 Yr15	AVS
20x	UCW	16,434 17%	27,871 30%	27,223 29%	32,287 34%	28,669 30%	34,898 37%	33,968 36%	41,019 44%	40,985 44%	47,507 50%
	UniGene v60	9,643 17%	16,182 28%	15,222 27%	19,549 34%	17,397 31%	20,567 36%	20,219 36%	25,270 44%	24,598 43%	29,052 51%
10x	UCW	27,371 29%	38,282 41%	37,777 40%	42,658 45%	38,999 41%	44,610 47%	43,266 46%	49,473 53%	49,182 52%	54,781 58%
	UniGene v60	16,201 28%	22,948 40%	22,130 39%	26,200 46%	24,130 42%	26,914 47%	26,318 46%	30,579 54%	29,857 52%	33,557 59%
>0x	UCW	68,302 73%	72,484 77%	72,957 79%	74,694 79%	73,290 80%	75,201 80%	74,397 79%	77,093 82%	76,715 81%	78,796 84%
	UniGene v60	40,717 71%	42,489 75%	42,595 75%	43,625 77%	43,059 76%	43,748 77%	43,393 76%	44,655 78%	44,364 78%	45,392 80%

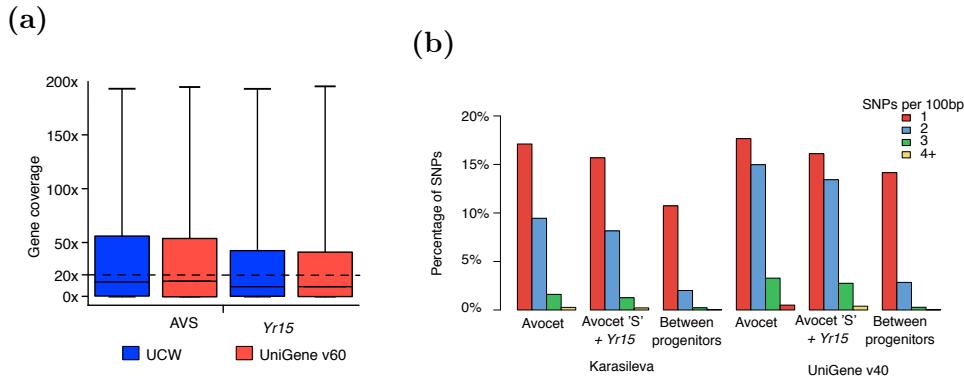


Figure 3.5: A coverage and SNPs between progenitors. (a) Box plot distribution of the gene coverage of the parent reads (AVS and *Yr15*) across the UCW (blue) and the UniGene (red) gene models. The dashed line represents the 209 minimum coverage required for SNP calling. The full line represents the average coverage across all gene models. (b) Percentage of genes exhibiting SNPs across references. The number of SNPs between the parent reads and the corresponding references was calculated (per 100 bp, rounded). The ‘between-parents’ category corresponds to putative SNPs when comparing the consensus sequence between AVS and *Yr15*. Adapted from Ramirez-Gonzalez et al. (2015c)

susceptible ($S_1 + S_2 + S_3$) and resistant bulks ($R_1 + R_2 + R_3$). The merged samples increased the percentage of genes with coverage over 20x to 44% and 50% in the resistant and susceptible bulks (Table 3.2), which is close to the coverage from the progenitors.

3.4 SNP Calling

The SNP calling was done on positions with a coverage of at least 20x on the progenitor lines against the gene reference. The AVS progenitor had roughly 3% more genes with polymorphisms than *Yr15*, consistent with the difference in coverage, suggesting that with a higher coverage we could recover more SNPs from *Yr15*. The UniGenes have a higher number of SNPs because the UCW gene models have a higher number of monomorphic genes when compared to the UniGenes. (Figure 3.5b; Table 3.3). The difference in the number of relative monomorphic SNPs between references can be explained by the fact that the UniGenes have homoeologues can be represented as a single sequence, as opposed to the UCW set which are homoeologue-specific, improving the mapping to the correct homoeologue in the genes from the UCW set over the UniGenes.

Table 3.3: Count of SNPs per 100 bp on genes with at least 20x coverage.

SNPs per 100bp	UCW			UniGene v60		
	AVS	AVS+ <i>Yr15</i>	Between progenitors	AVS	AVS+ <i>Yr15</i>	Between progenitors
0	67,389 71.6%	70,338 74.7%	81,921 87.0%	36,210 63.6%	38,339 67.3%	47,097 82.7%
1	16,111 17.1%	14,770 15.7%	10,107 10.7%	10,058 17.7%	9,175 16.1%	8,061 14.2%
2	8,904 9.5%	7,676 8.2%	1,893 2.0%	8,529 15.0%	7,648 13.4%	1,621 2.9%
3	1,517 1.6%	1,192 1.3%	215 0.2%	1,870 3.3%	1,568 2.8%	59 0.3%
4+	253 0.3%	198 0.2%	38 0.0%	287 0.5%	224 0.4%	16 0.0%

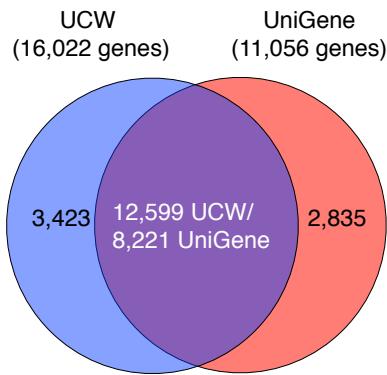


Figure 3.6: Gene models with putative SNPs in common between the UCW and UniGenes reference. The intersection represents the genes that are common in both sets. Adapted from Ramirez-Gonzalez et al. (2015c)

Both gene sets were done from varieties different to AVS and are likely to be incomplete, hence we set a low threshold of at least 20% of the observed nucleotides on any position to call an SNP. To represent cases where more than one consensus base is called we use International Union of Pure and Applied Chemistry (IUPAC) codes (Cornish-Bowden (1985); Section 1.4; Figure 3.2). To focus the analysis on informative SNPs, the common varietal SNPs and variations between homoeologues were removed by finding the cases when the consensus call on both progenitors is the same. The SNPs that are unique to a single parental were examined in detail. There are 66,426 putative SNPs across 16,022 (17%) UCW genes and 52,262 SNPs on 11,056 UniGenes (19.4%; Figure 3.6).

The high number of genes with SNPs was unexpected as a BC6 NIL used for an *F*₂ mapping population expects to have < 1% of the genetic background segregating. The both sets of gene models were aligned with BLAT (Kent, 2002) to the Chinese Spring Chromosome arm sur-

Table 3.4: Number of genes with SNPs assigned to the wheat chromosome arm CSS scaffolds (Mayer et al., 2014) using the best hit from BLAT (Kent, 2002)

Wheat Chromosome Arm	UCW (94,177)	UniGene v60 (56,954)	Total (151,131)
1AL	3,251 (3.45%)	1,404 (2.47%)	4,655 (3.08%)
1AS	1,366 (1.45%)	560 (0.98%)	1,926 (1.27%)
1BL	2,610 (2.77%)	1,280 (2.25%)	3,890 (2.57%)
1BS	1,487 (1.58%)	693 (1.22%)	2,180 (1.44%)
1DL	997 (1.06%)	1,057 (1.86%)	2,054 (1.36%)
1DS	753 (0.80%)	687 (1.21%)	1,440 (0.95%)
2AL	3,491 (3.71%)	1,460 (2.56%)	4,951 (3.28%)
2AS	2,305 (2.45%)	974 (1.71%)	3,279 (2.17%)
2BL	3,658 (3.88%)	1,546 (2.71%)	5,204 (3.44%)
2BS	2,790 (2.96%)	1,139 (2.00%)	3,929 (2.60%)
2DL	1,098 (1.17%)	1,069 (1.88%)	2,167 (1.43%)
2DS	796 (0.85%)	833 (1.46%)	1,629 (1.08%)
3AL	2,135 (2.27%)	978 (1.72%)	3,113 (2.06%)
3AS	1,543 (1.64%)	718 (1.26%)	2,261 (1.50%)
3B	6,559 (6.96%)	2,839 (4.98%)	9,398 (6.22%)
3DL	915 (0.97%)	938 (1.65%)	1,853 (1.23%)
3DS	412 (0.44%)	450 (0.79%)	862 (0.57%)
4AL	3,393 (3.60%)	1,335 (2.34%)	4,728 (3.13%)
4AS	2,011 (2.14%)	817 (1.43%)	2,828 (1.87%)
4BL	2,119 (2.25%)	898 (1.58%)	3,017 (2.00%)
4BS	1,946 (2.07%)	892 (1.57%)	2,838 (1.88%)
4DL	1,069 (1.14%)	945 (1.66%)	2,014 (1.33%)
4DS	800 (0.85%)	699 (1.23%)	1,499 (0.99%)
5AL	2,640 (2.80%)	1,132 (1.99%)	3,772 (2.50%)
5AS	963 (1.02%)	407 (0.71%)	1,370 (0.91%)
5BL	5,324 (5.65%)	1,943 (3.41%)	7,267 (4.81%)
5BS	1,360 (1.44%)	591 (1.04%)	1,951 (1.29%)
5DL	2,067 (2.19%)	1,688 (2.96%)	3,755 (2.48%)
5DS	620 (0.66%)	614 (1.08%)	1,234 (0.82%)
6AL	2,397 (2.55%)	896 (1.57%)	3,293 (2.18%)
6AS	2,285 (2.43%)	936 (1.64%)	3,221 (2.13%)
6BL	1,564 (1.66%)	820 (1.44%)	2,384 (1.58%)
6BS	1,308 (1.39%)	731 (1.28%)	2,039 (1.35%)
6DL	1,399 (1.49%)	1,050 (1.84%)	2,449 (1.62%)
6DS	870 (0.92%)	680 (1.19%)	1,550 (1.03%)
7AL	1,918 (2.04%)	849 (1.49%)	2,767 (1.83%)
7AS	1,717 (1.82%)	764 (1.34%)	2,481 (1.64%)
7BL	1,592 (1.69%)	776 (1.36%)	2,368 (1.57%)
7BS	1,239 (1.32%)	713 (1.25%)	1,952 (1.29%)
7DL	2,040 (2.17%)	1,301 (2.28%)	3,341 (2.21%)
7DS	1,224 (1.30%)	1,016 (1.78%)	2,240 (1.48%)
Assigned	80,031 (84.98%)	41,118 (72.20%)	121,149 (80.16%)

Table 3.5: Total number of SNPs scored in parents, individual bulks and in silico merged bulks.

Gene set	$\frac{R1}{S1}$	$\frac{R2}{S2}$	$\frac{R3}{S3}$	$\frac{R1+R2}{S1+S2}$	$\frac{R1+R2+R3}{S1+S2+S3}$	SNPs in parents
UCW	16,269 24.49%	29,703 44.72%	31,891 48.01%	44,224 66.58%	64,522 97.13%	66,426
UniGene v60	15,261 29.20%	25,143 48.11%	24,548 46.97%	35,698 68.31%	49,738 95.17%	52,262

vey sequence (CSS; Mayer et al. 2014); the alignment resulted on 80,031 (85.0%) UCW gene models and 41,118 (72.2%) UniGenes assigned to a chromosome arm (Table 3.4). The SNPs found in the mapped genes are evenly distributed across all the chromosomes (Figure 3.10a), suggesting that the Avocet S (JIC, UK) used as parent in the F_2 is different to the Avocet S used for the *Yr15* NIL development (University of Sydney, Australia).

To confirm that the Avocet S seed stocks from JIC are distinct to the stocks in Sydney DNA from both stocks was procured and compared with the iSelect 90k wheat SNP chip. Between two independent Avocet S seeds from JIC only 58 out of 71,972 (0.08%) valid assays were polymorphic. Nonetheless, ther are over 5,000 ($> 7.5\%$) assays with polymorphisms between JIC-Avocet S and Avocet S from Sydney. The different was not expected originally, but considering that the Avocet S seeds are coming from different stocks and the fact that in both countries commercial varieties with the same name had been released, it is not surprising.

3.5 Bulk Frequency Ratios

The objective was find the SNPs enriched on each bulk and hence linked to the phenotype, variations from *Yr15* to resistance and from AVS to susceptibility in the segregating population. Across individual bulks, it was possible to score between 15,261 (24.5%) to 31,891(48.0%) SNPs across both reference sets. On the *in silico* mixes over 95% of SNPs where scored (Table 3.5), suggesting that the coverage of individual bulks is not enough to score all the SNPs. The scoring was done with the Bulk Frequency Ratio (Trick et al. 2012;Figure 3.2; Section 3.10.3), which has a value that increases as the *Yr15* allele is observed more times relatively to the AVS allele.

Table 3.6: SNPs in chromosome group 1S vs total number of SNPs with a minimum BFR from 0 to 10. AVS: SNPs coming from Avocet + Yr15.

	Min BFR	Gene Set	R1/S1 Yr15	R1/S1 AVS	R2/S2 Yr15	R2/S2 AVS	R3/S3 Yr15	R3/S3 AVS	S1+2/ R1+2 Yr15	S1+2/ R1+R2+R3 Yr15	S1+S2+S3/ S1+S2+S3/ AVS
0	UCW	308/8,049 (3.83%)	305/8,220 (3.71%)	505/14,121 (3.58%)	556/15,582 (3.57%)	532/14,875 (3.58%)	623/17,016 (3.66%)	670/18,760 (3.57%)	885/25,464 ^a (3.48%)	860/24,026 (3.58%)	1,505/40,496 (3.72%)
		UniGene v60 (3.92%)	307/7,823 (4.02%)	299/7,438 (3.45%)	428/12,409 (3.31%)	421/12,734 (3.54%)	427/12,050 (3.31%)	415/12,498 (3.34%)	536/15,672 (3.42%)	595/20,026 (2.97%)	901/30,380 (2.97%)
1	UCW	214/4,415 (4.85%)	194/4,108 (4.72%)	325/7,603 (4.27%)	314/7,374 (4.26%)	365/7,920 (4.61%)	415/8,850 (4.69%)	426/10,122 (4.21%)	494/12,185 (4.05%)	539/13,037 (4.13%)	842/19,466 (4.33%)
		UniGene v60 (4.63%)	207/4,474 (4.63%)	194/3,630 (5.34%)	269/6,649 (4.05%)	269/6,193 (4.34%)	279/6,511 (4.29%)	272/6,436 (4.23%)	329/8,704 (3.78%)	369/9,343 (3.95%)	541/14,226 (4.11%)
2	UCW	92/651 (14.13%)	75/671 (11.18%)	142/1,377 (10.86%)	111/1,101 (12.65%)	147/1,162 (10.56%)	149/1,411 (12.61%)	167/1,324 (12.61%)	163/1,478 (11.03%)	194/1,370 (14.16%)	207/1,765 (11.73%)
		UniGene v60 (13.56%)	77/588 (11.01%)	58/527 (9.93%)	101/1,017 (11.25%)	81/720 (10.77%)	105/775 (13.55%)	84/867 (9.69%)	122/991 (12.31%)	116/973 (11.92%)	145/1,030 (14.08%)
3	UCW	78/299 (26.09%)	45/295 (25.59%)	118/646 (17.43%)	70/409 (18.37%)	123/577 (24.54%)	85/494 (15.24%)	145/673 (17.41%)	98/563 (17.41%)	168/768 (21.56%)	122/665 (18.35%)
		UniGene v60 (32.81%)	63/192 (16.35%)	28/160 (17.50%)	109/484 (22.52%)	44/217 (20.28%)	105/416 (25.24%)	44/246 (17.89%)	134/539 (24.86%)	149/640 (23.28%)	78/489 (19.81%)
4	UCW	75/232 (32.33%)	28/160 (17.50%)	19/108 (21.28%)	95/416 (22.52%)	33/138 (18.71%)	96/354 (28.47%)	105/416 (16.76%)	127/477 (24.13%)	149/640 (21.56%)	64/323 (15.95%)
		UniGene v60 (32.81%)	63/192 (16.35%)	17/104 (13.98%)	83/390 (21.28%)	29/155 (18.71%)	82/288 (24.54%)	29/173 (16.76%)	104/431 (24.13%)	40/214 (18.69%)	127/519 (24.47%)
5	UCW	69/202 (34.16%)	65/179 (35.58%)	19/108 (17.50%)	95/416 (22.55%)	33/138 (17.73%)	96/354 (13.73%)	105/416 (30.70%)	127/477 (17.86%)	134/539 (25.71%)	53/277 (15.75%)
		UniGene v60 (37.75%)	58/163 (15.71%)	11/70 (17.50%)	76/337 (22.55%)	14/102 (23.91%)	70/228 (23.91%)	20/112 (26.62%)	100/389 (26.62%)	23/146 (15.75%)	140/580 (25.16%)
6	UCW	65/179 (36.31%)	12/85 (14.58%)	86/380 (14.12%)	86/380 (22.63%)	22/98 (22.45%)	87/299 (29.10%)	11/94 (16.08%)	127/477 (16.08%)	28/175 (16.62%)	42/222 (18.92%)
		UniGene v60 (37.75%)	57/151 (14.58%)	7/48 (10.81%)	73/300 (24.33%)	6/71 (8.45%)	65/191 (34.03%)	13/84 (15.48%)	104/431 (27.37%)	40/214 (16.39%)	29/266 (10.90%)
7	UCW	58/161 (36.02%)	11/73 (4/37)	77/340 (25.10%)	13/74 (22.65%)	73/248 (17.57%)	7/69 (29.44%)	116/393 (10.14%)	122/429 (29.52%)	21/130 (18.02%)	29/165 (24.14%)
		UniGene v60 (42.42%)	56/132 (3.33%)	68/273 (25.10%)	5/58 (24.91%)	60/171 (8.62%)	60/171 (35.09%)	9/64 (14.06%)	98/358 (28.14%)	20/122 (17.48%)	115/439 (26.20%)
8	UCW	58/149 (38.93%)	10/62 (16.13%)	68/310 (21.94%)	12/59 (20.34%)	66/214 (22.65%)	6/56 (30.84%)	104/359 (10.71%)	116/393 (28.97%)	126/514 (18.02%)	22/143 (15.38%)
		UniGene v60 (43.65%)	55/126 (9.09%)	64/255 (25.41%)	5/50 (10.87%)	55/150 (10.00%)	55/150 (36.67%)	9/55 (16.36%)	91/334 (29.07%)	14/89 (15.73%)	15/108 (13.89%)
9	UCW	54/135 (40.00%)	8/53 (15.09%)	63/289 (21.80%)	8/51 (15.69%)	61/182 (33.52%)	5/49 (10.20%)	100/331 (30.21%)	104/359 (30.21%)	15/91 (16.48%)	13/106 (13.45%)
		UniGene v60 (45.30%)	53/117 (1.33%)	62/244 (25.41%)	5/46 (10.87%)	50/136 (36.76%)	9/48 (18.75%)	88/291 (30.24%)	88/291 (30.24%)	13/83 (14.89%)	15/108 (12.12%)
10	UCW	52/126 (41.27%)	8/50 (16.00%)	62/279 (22.22%)	8/50 (16.00%)	56/165 (33.94%)	4/45 (8.89%)	96/309 (31.07%)	14/82 (31.07%)	91/355 (25.63%)	13/100 (13.00%)
		UniGene v60 (47.62%)	50/105 (3.57%)	60/226 (26.55%)	5/39 (12.82%)	43/119 (36.13%)	7/45 (15.56%)	85/272 (31.25%)	13/82 (15.85%)	92/318 (28.93%)	12/97 (12.37%)

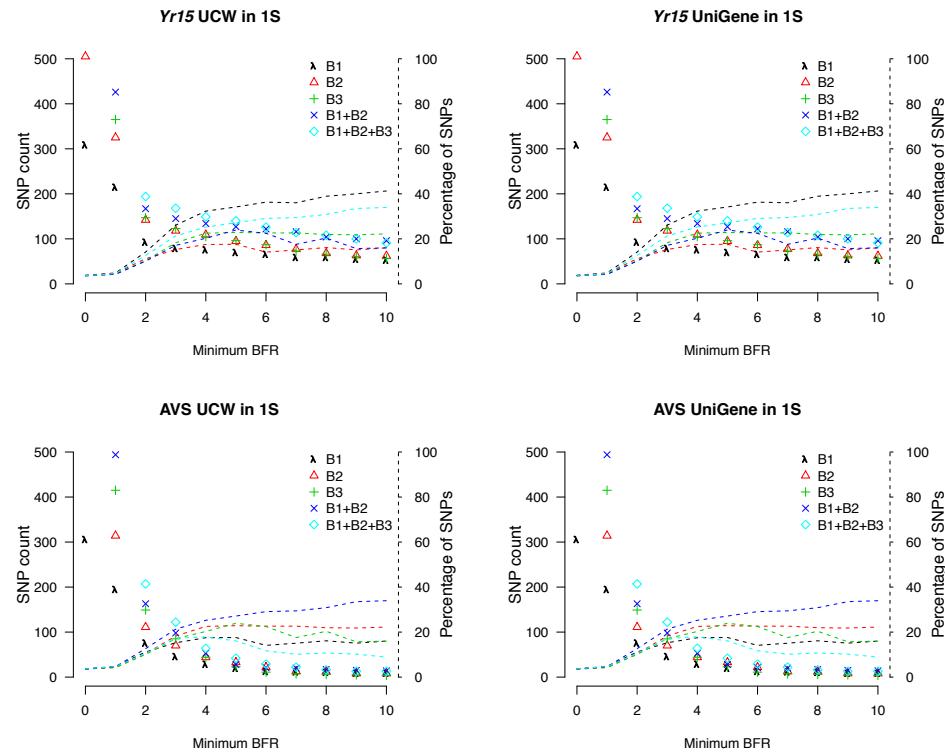


Figure 3.7: Effect of BFR threshold on the number of SNPs across the short arm of chromosome group 1. Figure previously published in Ramirez-Gonzalez et al. (2015c).

When increasing the minimum BFR threshold, enrichment of SNPs was observed in the short arm of the group 1 chromosomes (1S). Without taking in account the BFR, 3.6% of the SNPs are located in the 1S group, similar to the number of SNPs located in other groups 3.4. However, when increasing the threshold (between $BFR > 5$ and $BFR > 7$) the relative number of SNPs in group 1S increases. After $BFR > 7$ the gains in relative enrichment only improves marginally, but the number of called SNPs is reduced (Table 3.6; Figure 3.7). For that reason, SNPs with a $BFR > 6$ were selected for further validation. The method described by Trick et al. (2012) was extended to include cases where there is a complete lack of coverage in one of the samples ($BFR = \infty$), which is an ideal case where the linkage between the SNP and the phenotype is perfect. A total of 1,582 SNPs across 1,173 genes had a $BFR > 6$.

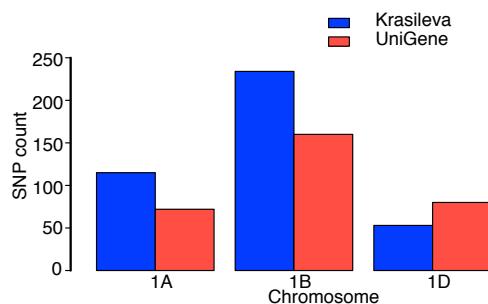


Figure 3.8: Location of SNPs with $BFR > 6$ according to the best alignment of the UniGene (red) and UCW (blue) gene models to the flow-sorted group 1 chromosomes from the Chinese Spring Survey sequence (CSS) (Mayer et al., 2014). Figure adapted from Ramirez-Gonzalez et al. (2015c).

3.6 *In silico* mapping

From the mapped SNPs with a $BFR > 6$, 872 of 1470 ($\sim 60\%$) were assigned to the chromosomes in group 1 of hexaploid wheat, being the only group with more than 4% of the SNPs assigned to it (Table 3.7). From the group 1, the B genome contained the higher proportion of SNPs mapped (54%), having 255 (54%) and 214 (46%) assigned to the long and short arms respectively (Figure 3.8). This results are expected since previous studies have located *Yr15* near the centromere in the short arm of chromosome 1B and, the *Yr15* introgression contains regions from the long and short arm from *T. dicoccoides* (Murphy et al., 2009; Peng et al., 2000; Sun et al., 1997).

The CSS assembly was used as a common reference between the reference genes and the SNPs 40,266 SNP markers published at the time when this analysis was done (Wang et al., 2014) to locate the SNPs with a $BFR > 6$ (including $BFR = \infty$) in a genomic position (Figures 3.9, 3.10). From the 1,582 SNPs across 1,173 genes, only 678 SNPs (43%, 474 genes) were successfully located in the genetic map. Since the CSS assembly is quite fragmented, the low percentage of located SNPs can be because not all candidate SNPs had a corresponding scaffold that has at least one of the 40,266 markers in the genetic map. Even if the number of located SNPs was not enough to give a position for over 50% of the SNPs from the parental line, the resolution of the genetic position SNPs that were assigned improved over just having the chromosome arm information from the CSS assembly. The mapping position further confirmed an enrichment of SNPs near the centromere of chromosome 1B with 325 out of 678 SNPs. Furthermore, 311 of those where located within an interval of 30cM (Figures 3.10b, 3.9a).

Studies in diploid organisms using QTL-Seq (Takagi et al., 2013a) or other NGS-enable genetic approaches (James et al., 2013) have shown

Table 3.7: SNP and genes with BFR > 6 mapping to each of the chromosomes from the CSS assemblies. The chromosome assignment on the "Genetically mapped" column correspond to the map published in Wang et al. (2014)).

Reference Chromosome	CSS assemblies						Genetically mapped									
	UCW gene models		UniGene v60		UCW gene models		UniGene v60		SNPs		Genes					
	SNPs	Genes	SNPs	Genes	SNPs	Genes	SNPs	Genes	SNPs	Genes	SNPs	Genes				
1AL 113	13.15%	79	12.29%	78	10.79%	50	9.43%	14	1.63%	8	1.24%	7	0.97%	4	0.75%	
1AS 26	3.03%	21	3.27%	20	2.77%	17	3.21%	42	4.89%	32	4.98%	38	5.26%	28	5.28%	
1BL 157	18.28%	110	17.11%	98	13.55%	64	12.08%	60	6.98%	35	5.44%	36	4.98%	23	4.34%	
1BS 120	13.97%	74	11.51%	94	13.00%	44	8.30%	127	14.78%	80	12.44%	102	14.11%	46	8.68%	
1DL 30	3.49%	21	3.27%	58	8.02%	47	8.87%	2	0.23%	2	0.31%	4	0.55%	4	0.75%	
1DS 40	4.66%	25	3.89%	38	5.26%	24	4.53%	12	1.40%	6	0.93%	8	1.11%	5	0.94%	
2AL 22	2.56%	20	3.11%	14	1.94%	12	2.26%	9	1.05%	8	1.24%	7	0.97%	5	0.94%	
2AS 11	1.28%	11	1.71%	10	1.38%	7	1.32%	9	1.05%	9	1.40%	2	0.28%	2	0.38%	
2BL 17	1.98%	15	2.33%	18	2.49%	17	3.21%	7	0.18%	5	0.78%	4	0.55%	4	0.75%	
2BS 11	1.28%	10	1.56%	12	1.66%	7	1.32%	13	1.51%	12	1.87%	7	0.97%	7	1.32%	
2DL 2	0.23%	2	0.31%	15	2.07%	10	1.89%	1	0.12%	1	0.16%	3	0.41%	2	0.38%	
2DS 0	0.00%	0	0.00%	5	0.69%	3	0.57%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
3AL 7	0.81%	7	1.09%	2	0.28%	2	0.38%	2	0.23%	2	0.31%	1	0.14%	1	0.19%	
3AS 1	0.12%	1	0.16%	4	0.55%	4	0.75%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
3B 31	3.61%	26	4.04%	28	3.87%	24	4.53%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
3BL 0	0.00%	0	0.00%	0	0.00%	0	0.00%	9	1.05%	7	1.09%	4	0.55%	4	0.75%	
3BS 0	0.00%	0	0.00%	0	0.00%	0	0.00%	2	0.23%	2	0.31%	5	0.69%	5	0.94%	
3DL 7	0.81%	6	0.93%	2	0.28%	2	0.38%	1	0.12%	1	0.16%	0	0.00%	0	0.00%	
3DS 1	0.12%	1	0.16%	2	0.28%	2	0.38%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
4AL 18	2.10%	15	2.33%	6	0.83%	6	1.13%	14	1.63%	11	1.71%	5	0.69%	4	0.75%	
4AS 5	0.58%	5	0.78%	6	0.83%	5	0.94%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
4BL 11	1.28%	10	1.56%	6	0.83%	6	1.13%	3	0.35%	3	0.47%	4	0.55%	4	0.75%	
4BS 6	0.70%	5	0.78%	13	1.80%	10	1.89%	4	0.47%	3	0.47%	4	0.55%	3	0.57%	
4DL 4	0.47%	4	0.62%	5	0.69%	5	0.94%	0	0.00%	0	0.00%	1	0.14%	1	0.19%	
4DS 2	0.23%	2	0.31%	5	0.69%	4	0.75%	0	0.00%	0	0.00%	1	0.14%	1	0.19%	
5AL 7	0.81%	5	0.78%	3	0.41%	3	0.57%	3	0.35%	2	0.31%	1	0.14%	1	0.19%	
5AS 1	0.12%	1	0.16%	2	0.28%	2	0.38%	1	0.12%	1	0.16%	1	0.14%	1	0.19%	
5BL 31	3.61%	28	4.35%	14	1.94%	14	2.64%	12	1.40%	12	1.87%	6	0.83%	6	1.13%	
5BS 7	0.81%	5	0.78%	6	0.83%	5	0.94%	2	0.23%	2	0.31%	1	0.14%	1	0.19%	
5DL 8	0.93%	7	1.09%	15	2.07%	14	2.64%	2	0.23%	2	0.31%	6	0.83%	6	1.13%	
5DS 4	0.47%	3	0.47%	6	0.83%	5	0.94%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
6AL 22	2.56%	17	2.64%	9	1.24%	7	1.32%	6	0.70%	5	0.78%	3	0.41%	3	0.57%	
6AS 8	0.93%	8	1.24%	11	1.52%	10	1.89%	5	0.58%	5	0.78%	4	0.55%	4	0.75%	
6BL 7	0.81%	6	0.93%	3	0.41%	2	0.38%	4	0.47%	3	0.47%	1	0.14%	1	0.19%	
6BS 7	0.81%	5	0.78%	2	0.28%	2	0.38%	5	0.58%	4	0.62%	0	0.00%	0	0.00%	
6DL 11	1.28%	10	1.56%	7	0.97%	7	1.32%	3	0.35%	3	0.47%	1	0.14%	1	0.19%	
6DS 5	0.58%	3	0.47%	2	0.28%	2	0.38%	4	0.47%	2	0.31%	1	0.14%	1	0.19%	
7AL 9	1.05%	8	1.24%	7	0.97%	6	1.13%	6	0.70%	5	0.78%	4	0.55%	4	0.75%	
7AS 5	0.58%	5	0.78%	8	1.11%	7	1.32%	0	0.00%	0	0.00%	0	0.00%	0	0.00%	
7BL 10	1.16%	10	1.56%	4	0.55%	4	0.75%	5	0.58%	5	0.78%	3	0.41%	3	0.57%	
7BS 3	0.35%	3	0.47%	4	0.55%	4	0.75%	4	0.47%	4	0.62%	1	0.14%	1	0.19%	
7DL 15	1.75%	10	1.56%	12	1.66%	12	2.26%	5	0.58%	2	0.31%	2	0.28%	2	0.38%	
7DS 8	0.93%	4	0.62%	6	0.83%	6	1.13%	1	0.12%	1	0.16%	1	0.14%	1	0.19%	
Unmapped	49	5.70%	35	5.44%	63	8.71%	46	8.68%	460	53.55%	358	55.68%	444	61.41%	341	64.34%
Mapped	810	94.30%	608	94.56%	660	91.29%	484	91.32%	399	46.45%	285	44.32%	279	38.59%	189	35.66%

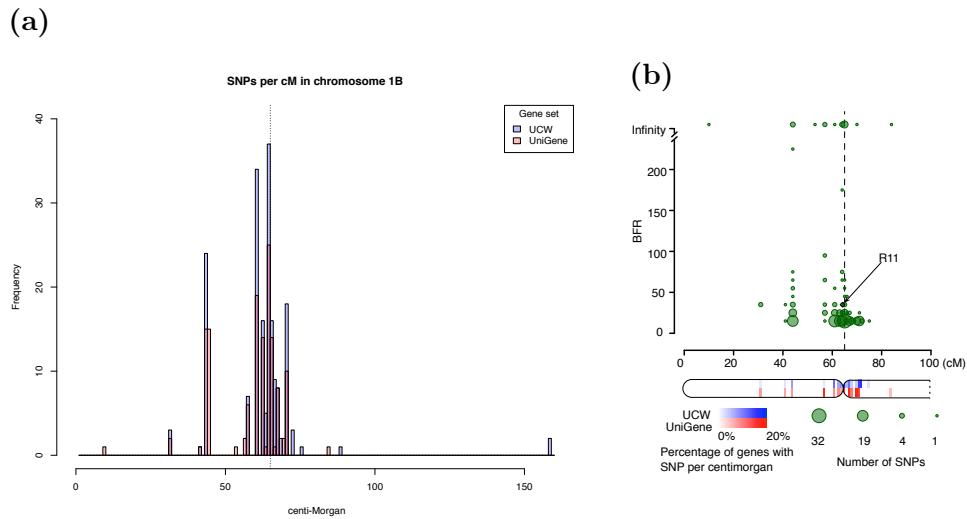


Figure 3.9: *In silico* location of SNPs with $BFR > 6$. (a) Number of SNPs with $BFR > 6$ per cM in chromosome 1B. (b) BFRs of mapped genes with SNPs on chromosome 1B. The area of the circle represents the number of SNPs clustered by location (windows size: 10 cM) and BFR (window size: 5cM). R11 is the only marker near the *Yr15* locus that had a corresponding position in the genetic map. The percentage of genes with SNPs per cM is also illustrated based on UCW (blue) and UniGene (red) gene models. The centromere is imputed by the centre of a window of 10 cM where the short arm switches to the long in the genetic map. BFRs correspond to those from the mixed *in silico* bulk S1 + S2 + S3/R1 + R2 + R3. Adapted from (Ramirez-Gonzalez et al., 2015c).

smooth curves with a defined peak in the region linked to the studied trait. In practice, we only observe clusters of SNPs with enriched BFRs near the centromere of chromosome 1B (Figures 3.9a, 3.10b).

The location of the clusters with an enrichment of SNPs near the centromere is not expected on a random selection of genes, as the gene density increases with the distance to the centromere (Akhunov et al., 2003). This suggests that the experiment was successful on finding SNPs linked to *Yr15*. There are several factor that prevent a clear peak; like the biases induced by the differential expression, the fragmented reference sequence with scaffolds that are not long enough to go across genetic positions. Since there are several SNPs with a high BFR and the genetic map is not enough to locate a single region linked to *Yr15*, multiple criteria was needed to prioritise SNPs that were more likely to yield on successful genetic markers.

3.7 Assay selection

Three independent criteria were use to prioritize the SNPs for marker development and validation:

High BFR. SNPs with a $BFR > 6$ in at least two independent bulk replicates or in either of the *in silico* mixes were selected to ensure consistency and recover SNPs with a low coverage on a particular bulk.

Group 1S. SNPs that are in CSS scaffolds in the short arm of chromosome group 1 were selected. This is to be consistent with the *in silico* genetic map and with previous studies (Murphy et al., 2009; Peng et al., 2000; Sun et al., 1997).

***Yr15* parent.** The SNPs should originate from the *Yr15* parent to ensure that the SNP is coming from the *T. dicoccoides* introgression and not from a SNP in the AVS genetic background, who would be less useful in breeding programs with a different background.

Only SNPs meeting the three criteria were selected for further analysis.

With the multiple criteria the number of genes with a putative SNP went down from $> 27,000$ to just 175; 77 and 98 from the UniGene and

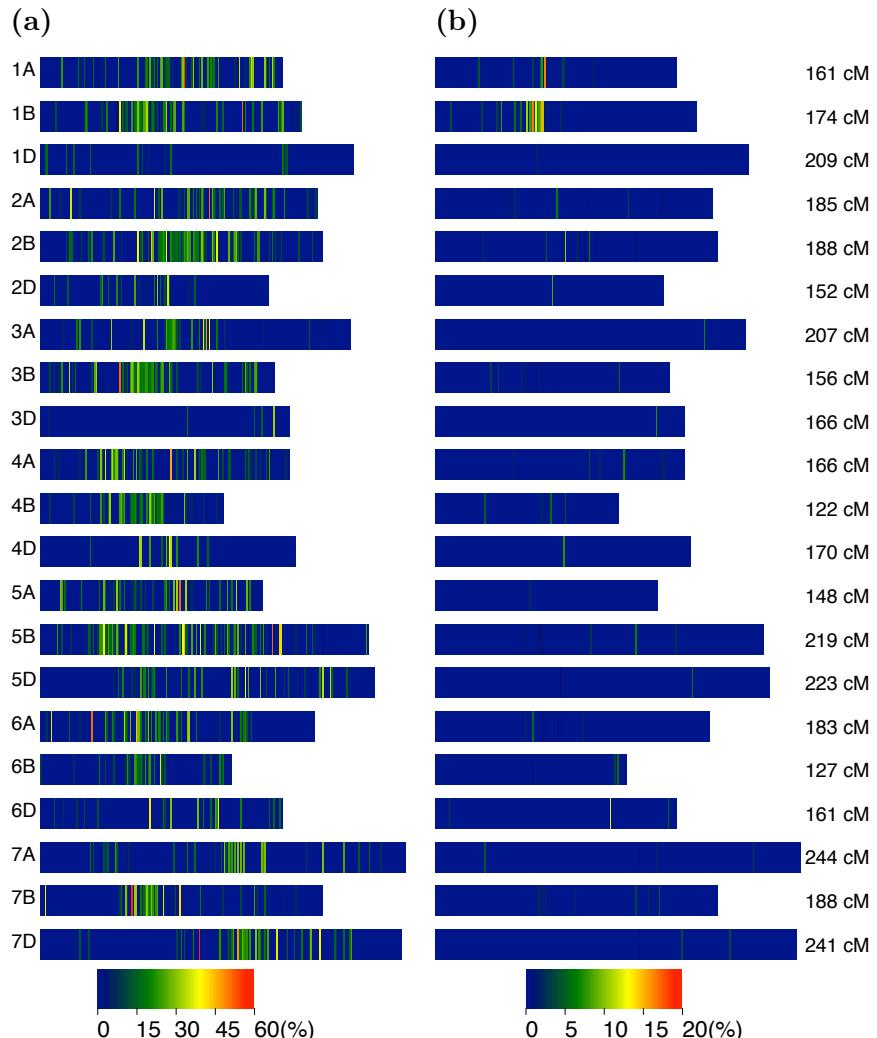


Figure 3.10: Genetic location of genes with SNPs between AVS and Yr15. The colour scale indicates the percentage of genes with SNPs per centi-Morgan (cM) across the 21 wheat chromosomes. The location of the genes was determined by the best alignment to the CSS scaffolds, and the location of these was determined by their position on a genetic map (Wang et al., 2014) (a). All the SNPs between progenitors. Note the lack of enrichment across any individual chromosome. (b) SNPs with $BFR > 6$. Note the enrichment in Chromosome 1B

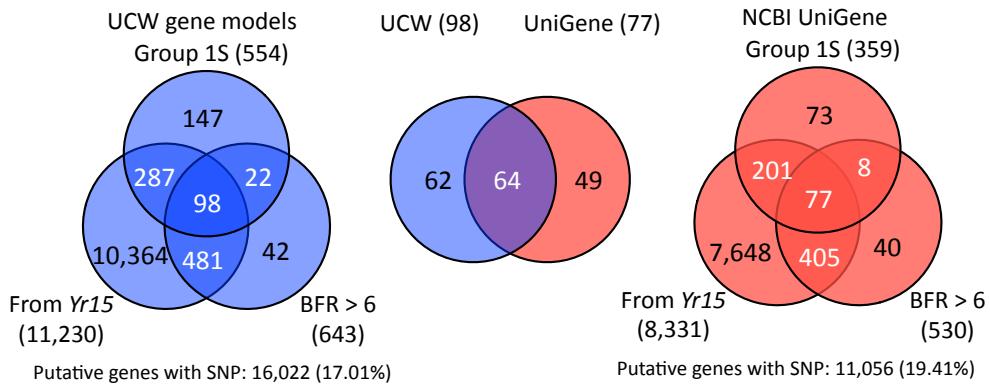


Figure 3.11: Selection criteria for marker design. Venn diagrams based on the three selection criteria (SNP in the short arm of chromosome group 1; SNP has a $BFR > 6$; and SNP is from the *Yr15* parent) for the UCW (blue) and UniGene (red) gene models. The centre diagram shows the intersection between common genes matching all three criteria across both data sets. Note that the numbers are not directly additive as in cases, multiple models from one reference set will relate to a single gene model in the other values. Published in (Ramirez-Gonzalez et al., 2015c)

UCW gene sets respectively. The selected genes from both references were aligned between references, as they come from independent sources an overlap in the selection between them is expected and, as expected, around half of the genes between gene sets overlap (Figure 3.11). The 50 SNPs with the highest BFRs, out of the 175 genes, were selected for validation, 15 of them were redundant between references, resulting on 35 SNPs to validate.

The separate bulks and the *in silico* mixes were evaluated in detail to understand the behaviour and value of having multiple bulks. The initial expectation was that as the number of SNPs with $BFR = \infty$ should drop in the mixes, as the improved coverage should reduce the instances where the absence of an allele is because of the lack of coverage on a particular sample. However, the opposite happened, the additional coverage in the *in silico* mixes recovered SNPs in genes with a low expression at the time of the sampling (Figure 3.12). Some SNPs were present across all the samples, however the value of the BFR changed depending on the sample(marker R5). On some cases a SNP are missing in an individual bulk, but present in the rest of them and in the mixes (marker R8). The main reason affecting the scoring is the coverage in the sample for each particular gene, hence an strategy with a consistent coverage would be preferred for this kind of analysis. Previous studies have shown that a

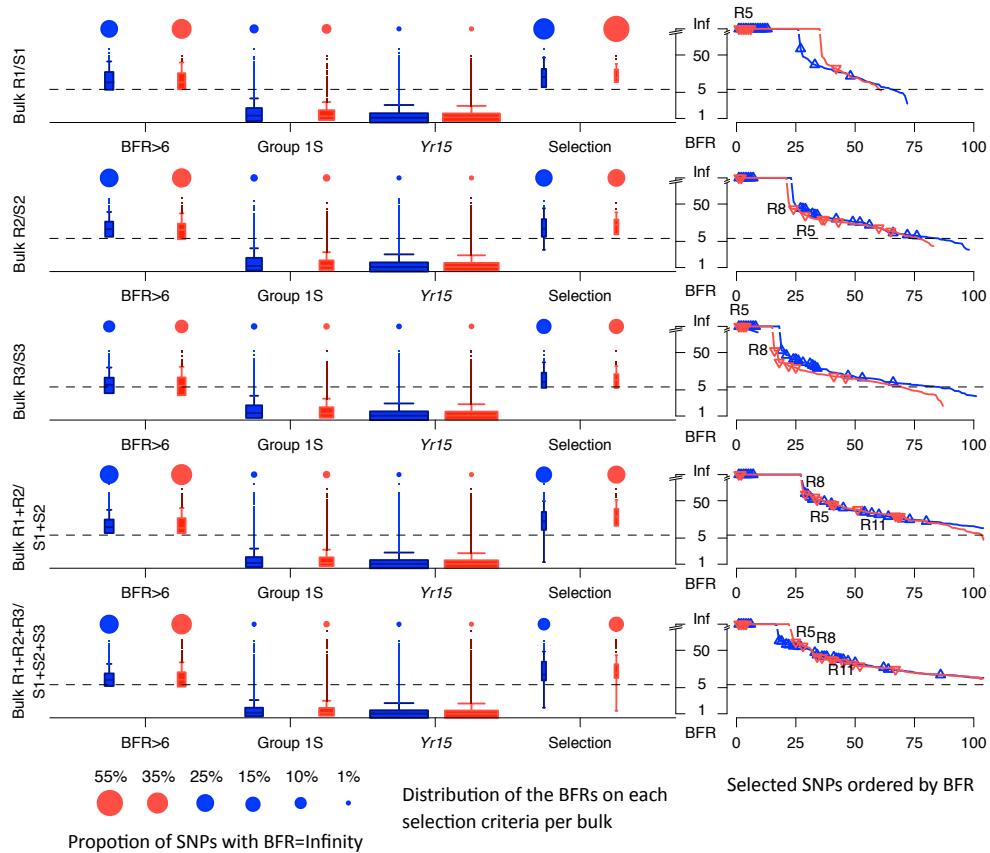


Figure 3.12: BFRs of selected SNPs across the individual bulks and in silico mixes (UCW, red; UniGene, blue). The dotted line represents the BFR threshold of 6 (logarithmic scale). Left: Distribution of the BFRs for each selection criteria and the selected SNPs for validation. The circles on the top of each plot represent the percentage of SNPs with $BFR = \infty$. The Selection may include SNPs with $BFR < 6$ when the same SNP has a higher score on the complementing reference (ie. $BFR > 6$ on UCW, but $BFR < 6$ on Uni-Gene). Right: The BFR values of selected SNPs were sorted in descending order across the different bulks and according to their origin. Validated SNPs are indicated by open triangles, and SNPs corresponding to markers R5, R8 and R11 are labelled across different bulks and mixes. Note that some SNPs are below the threshold in a specific bulk as they meet the BFR criteria across others.

Table 3.8: Number of genes (and SNPs) with a unique hit (> 99% sequence identity) to a single wheat survey sequence scaffold.

Chromosome 1		All SNPs		BFR>6		% BFR>6	
		SNP	Genes	SNP	Genes	SNPs	Genes
UCW	Unique	5,283	1,245	311	214	5.89%	17.19%
	Total	8,086	1,954	486	330	6.01%	16.89%
	Percentage	65.34%	63.72%	63.99%	64.85%		
UniGene	Unique	3,687	745	213	139	5.78%	18.66%
	Total	6,422	1,318	386	246	6.01%	18.66%
	Percentage	57.41%	56.53%	49.17%	56.07%		
UCW + UniGene	Unique	8,970	1,990	524	353	5.84%	17.74%
	Total	14,508	3,272	872	576	6.01%	17.60%
	Percentage	61.83%	60.82%	60.09%	61.28%		

All SNPs		All SNPs		BFR>6		% BFR>6	
		SNP	Genes	SNP	Genes	SNPs	Genes
UCW	Unique	39,247	9,585	481	368	1.23%	3.84%
	Total	66,426	16,022	859	643	1.29%	4.01%
	Percentage	59.08%	59.82%	56.00%	57.23%		
UniGene	Unique	27,292	5,698	344	252	1.26%	4.42%
	Total	52,262	11,056	723	530	1.38%	4.79%
	Percentage	52.22%	51.54%	47.58%	47.55%		
UCW + UniGene	Unique	66,539	15,283	825	620	1.24%	4.06%
	Total	118,688	27,078	1,582	1,173	1.33%	4.33%
	Percentage	56.06%	56.44%	52.15%	52.86%		

coverage of $< 5x$ is enough to call for SNPs in model organisms with a high-quality reference (Schneeberger and Weigel, 2011). However, the results on this study are in line with other studies using populations for SNP calling (Abe et al., 2012; Takagi et al., 2013a). The non-uniform distribution of the coverage in RNA-Seq experiments affects the number of reads that can be used to call for SNPs, specially on genes with a low expression level (Mortazavi et al., 2008).

Around 60% of the gene models, across both references, had a unique hit with > 99% sequence identity to a single CSS scaffold (Table 3.8). This is likely because there is no unique homoeologue in the gene models, leading to reads, from two different homoeologues, mapping to the same region. To reduce the number of spurious SNPs we used IUAPC ambiguity codes (Section 1.4, Cornish-Bowden (1985)) when two different alleles were observed. This had as side effect that in order to keep only high confidence SNPs we required a higher coverage ($> 20x$). On the original study introducing the BFR in tetraploid wheat, the authors show that increasing the coverage, from $8x$ to $16x$, reduces the putative SNPs

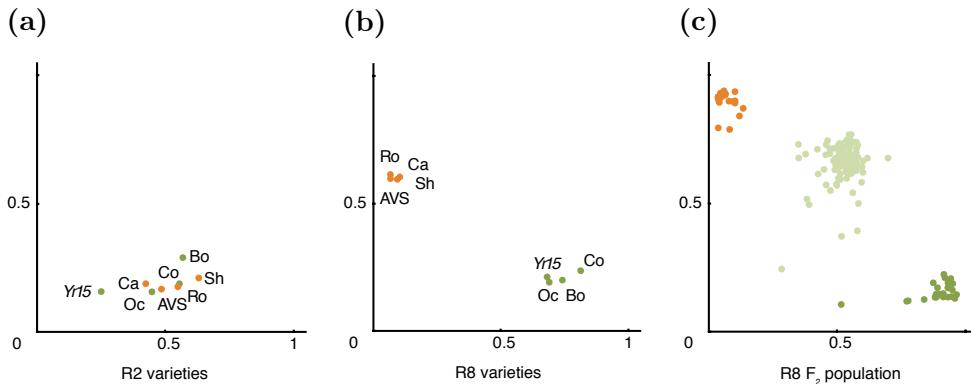


Figure 3.13: KASP output from the wheat variety panel with (Ochre, Boston, Cortez) and without (Robigus, Cadenza and Shamrock) *Yr15*. Marker R2 (a) is monomorphic while R8 (b) is polymorphic between varieties known to carry the gene. Marker R8 results for the F2 population (c) showing three distinct clusters. The central cluster (light green) is comprised of heterozygous individuals, whereas clusters near the axes are homozygous for either AVS (VIC; orange) or *Yr15* (FAM; dark green).

by 60%, but the validated SNPs increase from 57% to 83% (Trick et al., 2012). Hence, a compromise between increasing the minimum coverage at the cost of reducing the SNP candidates has to be reached in line with the objectives and available resources for a particular study.

3.8 SNP Validation

KASP assays were designed to validate and generate a genetic map of the *Yr15* locus for the 35 selected SNPs. To automate the design of genome-specific primers for polyploid organisms PolyMarker was developed (Chapter 2). Out of the 35 assays to design, 17 were designed as specific, 9 as semi-specific to chromosome 1BS, and 9 were not specific because there was no information for the homoeologous on the CSS scaffolds. PolyMarker also identified putative homoeologous variants (between genomes, as opposed to between varieties) that were in the list of candidate SNPs, but were not identified previously (Figure 2.11; Table 3.9).

To validate if the 35 SNPs were polymorphic across the parents and, diagnostic to *Yr15* we tested them in the progenitors plus six commercial varieties, three containing *Yr15* (Ochre, Boston and, Cortez) and three without it (Shamrock, Robigus and, Cadenza). Two of the lines without *Yr15* have *T. dicoccoides* in their pedigree (Shamrock and Robigus), as

Table 3.9: Primer details for the markers to validate.

it is the donor species of *Yr15* (McIntosh et al., 1995). This test panel allows to test if the SNPs are only diagnostic to *T. diccoccoides* instead of *Yr15*. On the test panel, 28 (80%) SNPs were polymorphic across the parents and three of them were diagnostic to *yr15* (R5, R8, R33). From the five homoeologous SNPs, three of them were monomorphic and two polymorphic, suggesting that PolyMarker is effective on detecting which assays are less likely to work (Table 3.10; Figure 3.13a,b). The segregation of the SNPs in the full *F*₂ population (Section 3.2, Figure 3.13c) and a genetic map was produced (Section 3.9).

3.9 Genetic map

Initially, the 28 polymorphic markers were used to genotype a subset of 66 plants from the *F*₂ population. From those, 23 (82%) were linked to *Yr15* and several markers fall in a small interval around *Yr15* (Figure 3.14a; Table 3.10), confirming that the multiple-criteria strategy (Section 3.7) for selecting candidate SNPs was effective. Then, the complete *F*₂ population was assessed with:

- the seven markers that were most linked to *Yr15*, including two of the diagnostic markers from the variety panel (R5 and R8),
- The flanking SSR microsatellite markers used by UK breeders for germoplasm selection (Xbarc8 and Xgwm413).
- A marker based on barley-wheat synteny (R43) which met the selection criteria, but wasn't on the original set of 50 markers with high BFR.

The *F*₂ population consisted on 232 plants with phenotypic information, of those 196 where genotyped reliably (no more than one data point missing). Using the eight SNP markers and 2 SSRs, the *Yr15* locus was mapped to an interval of 0.77cM, with R8/xgwm413 0.26cM distal, and R5/R11 0.77cM proximal from *Yr15* (Figure 3.14b,c).

The sub-cM resolution is expected on an *F*₂ population of 196 individuals, as 392 gametes provide a resolution of 0.26cM. Despite the fact that none of the selected markers have perfect linkage to *Yr15*, the produce genetic map is an improvement in the resolution of the map for the locus and it enables the shift to SNP markers from microsatellites, which

Table 3.10: Results of validation of primers on the progenitors (AVS and *Yr15*, varieties known to contain *Yr15* (Cortez, Ochre and, Boston) and, varieties without *Yr15* (Robigus, Cadenza and, Shamrock). Shamrock and Robigus have *T. dicoccoides* introgressions. The bold markers are diagnostic in the panel (R5, R8, R88) or in the genetic map (R11).

Assay ID	Gene set	Gene model name	SNP	<i>Yr15+</i>		<i>Yr15-</i>		comment
				Yr15 Ochre	Yr15 Shamrock	Yr15 Robigus	Yr15 Cadenza	
R1	UCW	UCW_Tk-k55_contig_8830;tt-k21_contig_10204	C341G	A	H	A	A	-
R2	UniGene v60	gnl UGITa#S13126619	C491T	B	B	B	B	Yes
R3	UCW	contig95240	C220G	H	B	B	B	No
R4	UCW	contig105384	C1227T	A	B	B	B	Yes
R5	UniGene v60	gnl UGITa#S58861868	A214G	A	A	B	B	Yes
R6	UCW	KukriC706_1	T2979C	A	H	B	B	Yes
R7	UniGene v60	gnl UGITa#S37932863	C281T	H	A	A	B	No
R8	UniGene v60	gnl UGITa#S58863387	T241C	B	B	A	A	Yes
R9	UniGene v60	gnl UGITa#S588892239	C303T	H	B	B	B	No
R10	UCW	UCW_Tk-e63_contig_79829	C207T	H	A	B	B	Yes
R11	UCW	UCW_Tk-k45_contig_39011	C726T	A	A	H	-	Yes
R12	UCW	contig53088	G587A	-	H	H	B	Yes
R14	UniGene v60	gnl UGITa#S44692929	C549T	A	A	A	A	Yes
R15	UCW	UCW_Tk-k51_contig_2344;tt-k55_contig_2091	T686G	A	A	A	A	No
R16	UniGene v60	gnl UGITa#S17898149	G227A	A	B	A	B	Yes
R17	UCW	CL3339Contig1	T509C	H	H	H	H	No
R19	UCW	UCW_Tk-e21_contig_8407;tt-k61_contig_5972	C1405T	A	B	B	B	Yes
R20	UCW	UCW_Tk-k21_contig_8407;tt-k61_contig_5972	T1102C	A	B	B	B	Yes
R21	UCW	UCW_Tk-k31_contig_53804;tt-k41_contig_31582	G1810T	H	B	B	B	Yes
R22	UCW	UCW_Tk-k31_contig_14966	T408C	A	A	A	A	Yes
R23	UCW	UCW_Tk-k31_contig_12731;tt-k55_contig_13077;tt-k61_contig_18734	C507T	A	H	H	-	Yes
R24	UCW	UCW_Tk-k55_contig_8830;tt-k21_contig_10204	T3005G	H	H	B	B	Yes
R25	UCW	UCW_Tk-k63_contig_79829	G184A	A	A	A	A	No
R26	UCW	UCW_Tk-k21_contig_3794	C702T	H	A	B	B	Yes
R28	UCW	KukriC701_1	T1053C	A	A	B	B	Yes
R29	UCW	UCW_Tk-k55_contig_8640;tt-k41_contig_8875	G783A	H	A	B	B	Yes
R30	UCW	UCW_Tk-k55_contig_8830;tt-k21_contig_10204	T2184A	A	B	A	B	Yes
R31	UCW	UCW_Tk-k45_contig_22098	G683T	A	B	B	A	Yes
R32	UCW	UCW_Tk-k21_contig_33188;tt-k25_contig_30647	C596A	H	A	A	A	No
R33	UniGene v60	gnl UGITa#S58861868	G486T	A	A	B	B	Yes
R34	UCW	UCW_Tk-k31_contig_34099	G1713A	H	A	B	B	No
R35	UniGene v60	gnl UGITa#S58900202	T889C	A	B	B	B	Yes
R36	UCW	UCW_Tk-k55_contig_8830;tt-k21_contig_10204	T2349C	H	H	-	H	Yes
R37	UCW	UCW_Tk-k31_contig_34099	C846T	B	B	B	B	No
R38	UniGene v60	gnl UGITa#S588840501	T179G	B	B	B	B	No
R40	UCW	UCW_Tk-k31_contig_34099	C846T	A	H	B	B	Yes
R43	UniGene v60	gnl UGITa#S588843705	G268A	A	B	-	B	Yes

Linked *Yr15*
polymorphic
sites
based on barley
synteny

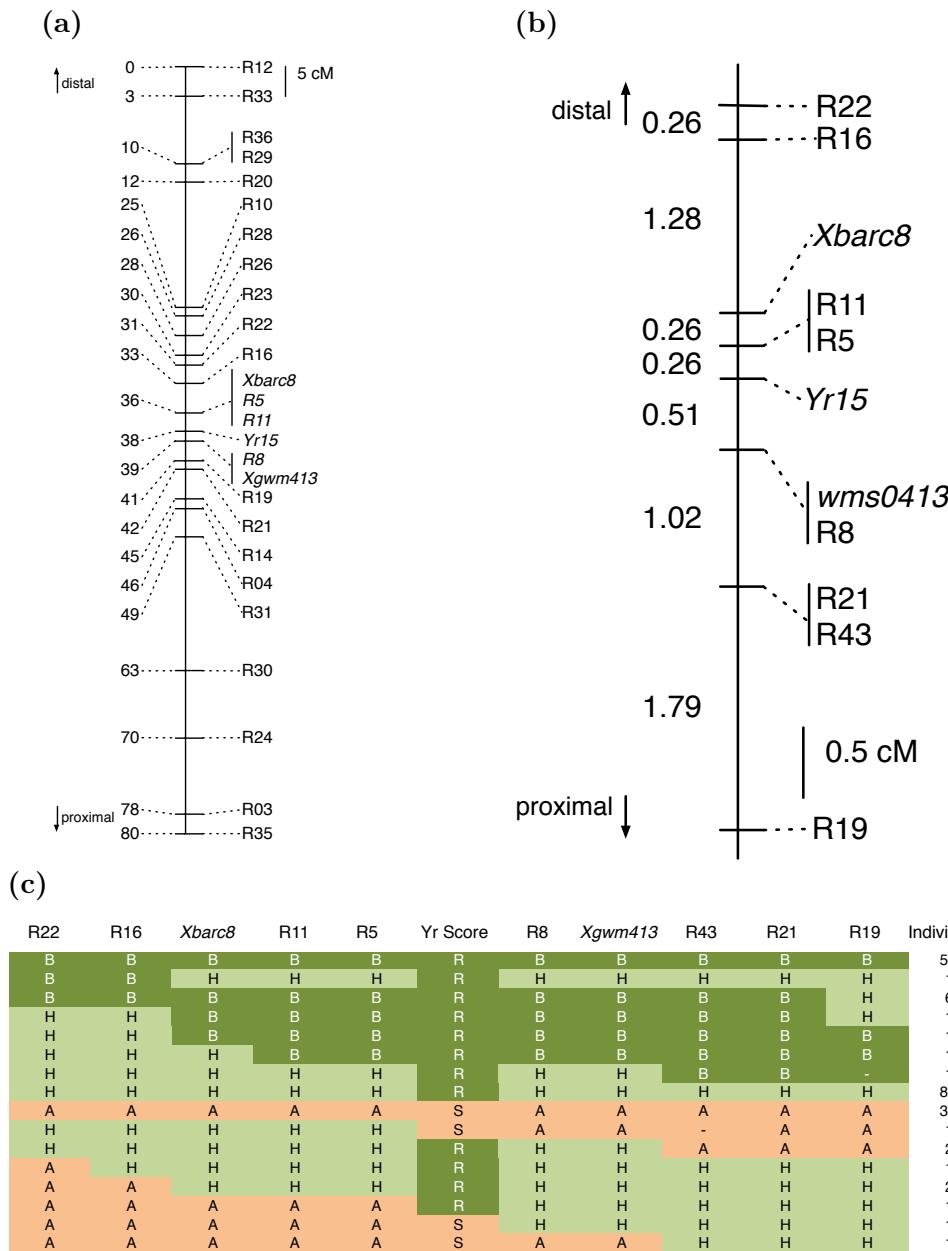


Figure 3.14: Genetic maps for *Yr15*. (a) Genetic map of the test panel from 50 individuals. (b) Genetic map from 196 individuals from the full population only with the 8 markers previously identified as closer to the *Yr15* locus. (c) Graphical genotype of the 196 *F₂* individuals used to develop the genetic map. The alleles are abbreviated according to their origin: A: AVS; B: *Yr15* and H: Heterozygous. Missing calls are indicated by a hyphen.

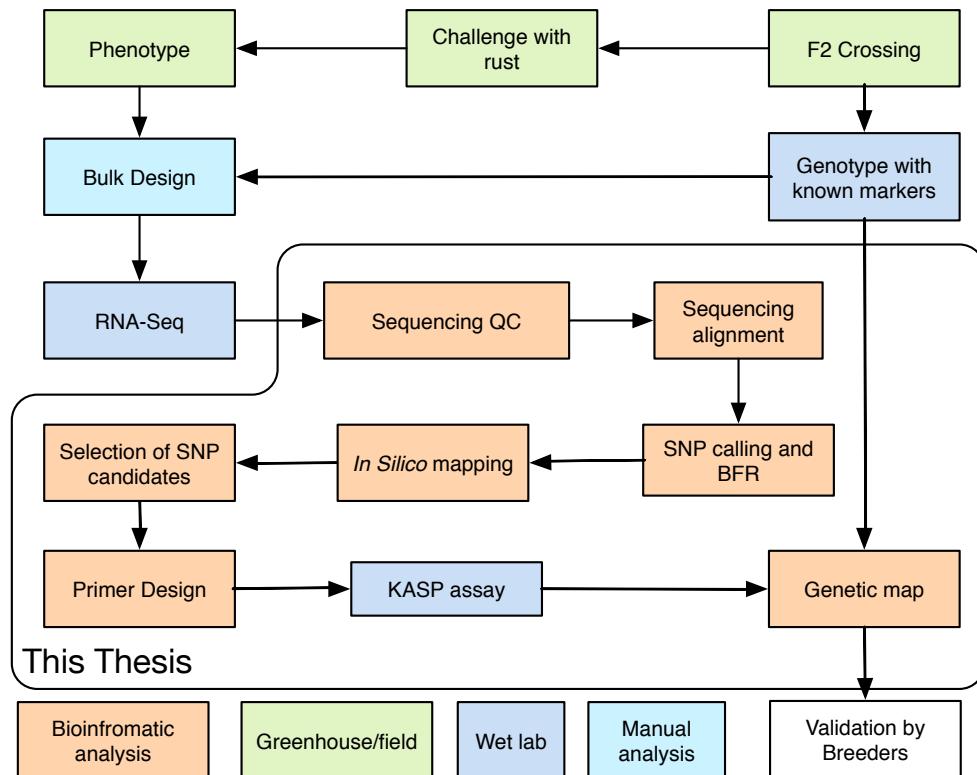


Figure 3.15: Steps used to go from the F_2 population to the genetic map.

has become the preferred marker system in MAS pipelines in breeding programmes.

3.10 Methods

The data analysis for this PhD required the use of some standard tools and custom developed code. All the code produced for this project is available and updated on the a github repository: <https://github.com/TGAC/bioruby-polyploid-tools>. For clarity, the snippets of code on this section had been simplified by removing the exception handling, type checks and caching mechanism.

3.10.1 Base-call and Quality Control of sequencing reads

The raw output from the Illumina HiSeq 2000 was processed with Casava v1.8 (Illumina, 2011). Lanes 1 and 2, containing multiplexed bulks (Table 3.1) was demultiplexed with a tolerance of 1 mismatch in the barcode. Lanes 3 and 4 contained the parental sequences without a barcode. The

FastQ files were left compressed and in chunks of 40,000, as the default for the BCL conversion pipeline from Casava to allow parallel processing in a cluster environment. The quality of the sequencing lanes was assessed with FastQC v0.10.1 (Babraham Bioinformatics, 2012).

3.10.2 Alignment reads to gene models

The RNA-Seq reads were aligned with BWA 0.5.9 (Li and Durbin, 2009) to the wheat UniGene database v60 (Pontius et al., 2002) and to the UCW gene models (Krasileva et al., 2013), including the *T. turgidum* and complementary ORFs (MAS Wheat, 2013). The alignments were sorted and stored as single BAM files to have random access (Li et al., 2009).

3.10.3 Bulk Frequency Ratios and SNP calling

To avoid the creation of several temporary files with the coverage information on all the bases I developed a Ruby pipeline based on the `bio-samtools` library (Ramirez-Gonzalez et al., 2012), and some of the improvements to work with pileups were published as a followup on the library (Etherington et al., 2015). To call for the consensus, the function `Bio::DB::Sam::mpileup` is called to generate the pileup of each gene. As the pileups are used several times during the analysis, a function that caches the current pileup is implemented. The consensus is called by counting how many times each base appears, and if the number of bases is higher than `minimum_ratio_for_iuap_consensus` the base is added to the set of possible bases (Cornish-Bowden, 1985) If there is no coverage at a certain position, the reference base is used, and set as lowercase. If the set of called bases is not empty, the ambiguity code for the observed bases is called, and set as upper case (Listing 3.1). The minimum ratio was done on 0.2 (20%), that allows for calling for a consensus even when more than one homoeologue is mapping to the same reference.

Snippet with submission of the alignments. However I haven't got access to the old cluster files.

Listing 3.1: Method to call for the consensus on progenitors from a pileup

```

1 def consensus_iuap(minimum_ratio_for_iuap_consensus)
2   minimum_ratio_for_iup_consensus
3   @consensus_iuap = self.ref_base.downcase
4   bases = self.bases
5   tmp = String.new
6   bases.each do |k,v|
7     if v/self.coverage > minimum_ratio_for_iup_consensus
8       tmp << k[0].to_s
9     end
10    if tmp.length > 0
11      @consensus_iuap = Bio::NucleicAcid.to_IUAPC(tmp)
12    end
13  end
14  @consensus_iuap.upcase
15 end

```

Then, to calculate the BFRs as shown on Figure 3.2 extra extensions for the `Bio::DB::Pileup` were added to get the actual number of bases in the pile (to exclude short insertions and deletions; Listing 3.2), and to calculate the SNP-Index (Listing 3.3).

Listing 3.2: `base_coverage` gets the number of bases called from a single pileup.

```

1 def base_coverage
2   total = 0
3   @bases.each do |k,v|
4     total += v
5   end
6   total
7 end

```

Listing 3.3: `base_ratios` gets the SNP-Index on a single pileup.

```

1 def base_ratios
2   return @base_ratios if @base_ratios
3   bases = self.bases
4   @base_ratios = Hash.new
5   bases.each do |k,v|
6     @base_ratios[k] = v.to_f / self.base_coverage.to_f
7   end
8   @base_ratios
9 end

```

To calculate BFRs the class `Bio::BFRTTools::Container` was implemented to contain all the `BIO::DB::Sam` objects corresponding to the progenitors and the bulks. The class `Bio::BFRTTools::BFRRegion` was implemented to contain the ratios and consensus sequences of each region. The method `bfr` uses the calculated SNP-Indices on every position, from the point of view of both progenitors (lines 15-16: Listing 3.4, and in the case of lack of coverage the value is set to 0 or `Infinity` (lines 8-13), depending on the progenitor where the base is not called at all. Using this design were the values of each region are calculated at once increases reduces the number of times the pileup needs to be generated for each sample and, allows to have in a single place in memory all the elements to calculate the BFRs without having to write any temporary files on disc. Also, the fact that the calculation of each region is independent to other regions, it is possible to use a computing cluster to distribute the analysis on several nodes.

The code produces a table with the SNP-Indices and BFRs for all the SNPs found in the progenitors. The program was used to calculate the BFRs on the independent conditions (Bulk 1: S1–R1, Bulk 2: S2–R2 and Bulk 3: S3–R3); the *in silico* mixes of bulks 1 and 2; and bulks 1, 2 and 3.

Listing 3.4: Section of the code that

```

1 for i in (0..self.size-1)
2   ratios_1 = @ratios_bulk_1[i]
3   ratios_2 = @ratios_bulk_2[i]
4   BASES.each do |base|
5     if ratios_1[base] == 0 and ratios_2[base] == 0
6       bfr1 = 0
7       bfr2 = 0
8     elsif ratios_1[base] == 0
9       bfr1 = 0
10      bfr2 = Float::INFINITY
11    elsif ratios_2[base] == 0
12      bfr1 = Float::INFINITY
13      bfr2 = 0
14    else
15      bfr1 = ratios_1[base] / ratios_2[base]
16      bfr2 = ratios_2[base] / ratios_1[base]
17    end
18    @BFRs[:first][base] << bfr1
19    @BFRs[:second][base] << bfr2
20  end
21 end

```

3.10.4 *In Silico* mapping

To find the chromosomal position of the SNPs with a high BFR the sequence of the markers with a genetic position from Wang et al. (2014) were aligned with BLAT (Kent, 2002) to the CSS scaffolds (Mayer et al., 2014). To find the best hit for each query was kept using a Ruby script. Briefly, the class `Bio::Blat::Report` from BioRuby (Goto et al., 2010) was extended to include an iterator only for the best alignment of each query: First, the whole file is iterated (line 5); the alignment with the best score is stored in a hash (lines 7-9) and; the hash is iterated (line 11). The script found 46,977 scaffolds that contained at least one marker from the map.

Listing 3.5: Extension to `Bio::Blat::Report` that selects the best alignment from a psl file from BLAT

```

1 def self.each_best_hit(text = '')
2   emptyHit = Bio::Blat::Report::Hit.new
3   emptyHit.score = 0
4   best_aln = Hash.new(emptyHit)
5   self.each_hit(text) do |hit|
6     current_score = hit.score
7     if current_score > best_aln[current_name].score
8       best_aln[current_name] = hit
9     end
10  end
11  best_aln.each_value { |val| yield val }
12 end

```

Then, the UniGenes and the UCW gene models were also aligned with BLAT to the scaffolds that were located in the genetic map. The class `Bio::Blat::Report::Hit` was extended to calculate how many bases are covered in the alignment and the percentage of covered bases in both, the target and query sequences (Listing 3.6). Only the genes that align over 60% of covered bases with an identity of at least 90% were considered. This removes spurious mappings from repetitive regions while retaining a location to an homoeologue in case that the correct scaffold is not in the genetic map. The genes were also align to the full CSS reference, to be able to locate the genes to a chromosome arm, even when it is not possible to assign a position in the genetic map and, to the cDNA of *Hordeum vulgare* (Mayer et al., 2011) as deposited in Ensembl! Plants, release 16 (Kersey et al., 2012). The genetic position of the contigs was used to calculate the density of SNPs between AVS and *Yr15* in the genetic bins for Figure 3.10. This information was used to select the SNPs with high BFR to validate.

Include code on how the coordinates where extracted, with the patch to the Ensembl package

Listing 3.6: Extension to `Bio::Blat::Report::Hit` for filtering of spurious alignments.

```

1 class Bio::Blat::Report::Hit
2   def covered
3     match + mismatch
4   end
5   def query_percentage_covered
6     covered * 100.0 / query_len.to_f
7   end
8   def target_percentage_covered
9     covered * 100.0 / target_len.to_f
10  end
11 end

```

3.10.5 Primer design and KASP assays

The primer design for KASP were designed with PolyMarker as described in Chapter 2. The only difference with the default settings is that instead of using a template sequence, the sequence for each allele is calculated from the consensus of the alignments. As described in as described in Ramirez-Gonzalez et al. (2015c), the primers

were ordered from Sigma-Aldrich (Gillingham, UK), with primers carrying standard FAM or HEX compatible tails (FAM tail: 5' GAAGGTGACCAAGTTCATGCT 3'; HEX tail: 5' GAAGGTCGGAG TCAACGGATT 3') with the target SNP at the 3' end. Primer mix was set up as recommended by LGC [46 μL dH₂O, 30 μL common primer (100 nM) and 12 μL of each tailed primer (100 nM)] (LGC Genomics, 2014) Assays were tested in 384-well format and set up as 4-μL reactions [2-μL template (10–20 ng of DNA), 1.944 μL of V4 29 Kaspar mix and 0.056 μL primer mix]. PCR cycling was performed on a Eppendorf Mastercycler pro 384 using the following protocol: hotstart at 95 °C for 15 min, followed by ten touchdown cycles (95 °C for 20 s; touchdown 65 °C, ?1 °C per cycle, 25 s) then followed by 30 cycles of amplification (95 °C 10 s; 57 °C 60 s). As KASP amplicons are smaller than 120 bp, an extension step is unnecessary in the PCR protocol. 384-well optically clear

Fix
meth-
ods

plates (Cat. No. E10423000; Starlab Milton Keynes, UK) were read on a Tecan Safire plate reader. Fluorescence was detected at ambient temperature. If the signature genotyping clusters had not formed after the initial amplification, additional amplification cycles (usually 5–10) were conducted, and the samples were read again. Data analysis was performed manually using Klustercaller software (version 2.22.0.5; LGC Hoddesdon, UK).

3.10.6 Genetic map

As described in Ramirez-Gonzalez et al. (2015c):

JoinMap version 3 (van Ooijen and Voorrips, 2002) was used for linkage analysis and genetic map construction, using default settings. The linkage to *Yr15* was determined using a divergent log-of-odds (LOD) threshold of 3.0, and genetic distances were computed based on recombination frequency.

3.11 Discussion

Resequencing the \sim 17Gbp genome of hexaploid wheat is costly and approaches to reduce the required sequenced volume to effectively call for SNPs had been evolving since the conception of this project. The RNA and DNA extraction and the sequencing for this project was carried on before the beginning of my PhD (before October 2012). At that point exome capture was already established for genotyping humans (Ng et al., 2009), however the first exome capture on wheat was just recently published, with probes coming from unassembled 454 reads (Winfield et al., 2012), and a probe designed from transcripts (Henry et al., 2014) was not published after the analysis of this section was completed and validated. An even more targeted capture for resistance genes (RenSeq) was published while this study was executed (Jupe et al., 2013). On the other hand RNA-Seq was already tested for Bulk Segregant Analysis on tetraploid wheat (Trick et al., 2012). Hence, the decision of reducing the sequenced space with RNA-Seq was appropriate at the time (Figure 1.5). Unfortunately, one of the shortcomings of RNA-Seq used to call for

SNPs is that the coverage is not uniform and the genes that have low expression don't have enough coverage to call for SNPs (Section 3.3). If a similar study is to be started today, a better alternative would be to use exome capture in general from a segregating population for any trait, or RenSeq if the target gene is a resistance gene.

The quality and completeness of the reference genome or gene models directly affects the mapping NGS reads. This is particularly true on polyploid organisms: if one of the homoeologues is absent, the reads are likely to map to the wrong genome if the parameters of the aligner are relaxed or; not map at all if the required identity is high. When the bioinformatic analysis of this project started, the only available wheat genomic reference was a whole genome shotgun 454 sequencing, unassembled (Brenchley et al., 2012); the Chinese Spring Chromosome arm survey sequence (CSS) assembly was being finished (Mayer et al., 2014); the longer scaffolds from Chapman et al. (2015) were not public yet and; the efforts to make a whole genome shotgun assembly were being planned independently by the International Wheat Genome Sequencing consortium (Pozniak, 2016) and TGAC (Clark, 2016). Because a contiguous assembly with the corresponding annotation wasn't available at the time of the analysis and the fact that the data available was from a transcriptome, the use of gene models as a reference for the alignment was a suitable approach.

In terms of available gene sets when the analysis started, the canonical reference was the UniGenes from the NCBI (Pontius et al., 2002). The UniGenes are produced with an automated pipeline that clusters all the ESTs deposited in the NCBI by identity and selects the longest transcript, which can merge homoeologous transcripts as a single reference. Shortly after I started the bioinformatic analysis, two additional gene models were available, the draft annotation for the CSS assembly (MIPSv1) in January 2013 and the UCW gene models (Krasileva et al., 2013) in May 2013. I selected the UCW gene models, as they were more mature and were phased to distinguish between genomes and already published, over the the MIPSv1 genes, still being refined from an initial approach lifting proteins from related organisms and a few RNA-Seq experiments. The MIPS gene models were improved by removing duplications in the assembly in a later stage and the nomenclature before the release of the assembly (Mayer et al., 2014), but at that point the results of this project

where already submitted for publication (Figure 1.5; Ramirez-Gonzalez et al. 2015c).

To locate the gene models in the chromosome arms and see if there was an enrichment on the called SNPs the use of a high resolution consensus map is needed, as the genome assemblies available during the analysis are fragmented. Timely, a genetic map with > 42,000 markers was published (Wang et al., 2014). I was able to use it to locate several CSS scaffolds before the assembly was published, as I collaborated in the project. The located scaffolds were used as proxy to sort just under half of the reference genes in their chromosomal position (Section 3.6). Despite the resolution not being enough to find a single point of enrichment, it was enough to confirm that the SNPs were in the expected location, including one of the SNP candidates flanking the *Yr15* locus (SNP R11, Figure 3.9b). If the analysis was to be done today, the genetic map from Chapman et al. (2015) along with their longer scaffolds, or the scaffolds from TGACv1 or the NRGene should provide a better resolution. Even without having all the CSS scaffolds sorted, the fact that they come from individual chromosome arms they enabled the assignment of the genes to a chromosome.

Should I talk briefly about barley?.
I would need to add a section before

The original expectation was to have a NIL for the BSA, however the number of SNPs called in the progenitors suggested that the background, Avocet S, was not the same. This happened because despite both susceptible lines being called the same and having the same response to the pathogen, they are different lines from different countries (Section 3.4). This highlights the importance of genotyping the material used when developing mapping populations, specially if the source of the seeds come from different seed banks.

Despite this shortcomings, the use of the BFRs to score the putative SNPs was effective as most of the SNPs with a high score mapped in chromosome 1B, as expected from previous studies ($BFR > 6$, Section 3.7). Using the extra criteria of only selecting SNPs from the resistant progenitor and in the expected chromosome arm I was able to produce a high resolution genetic map (Section 3.9). The genetic map was of the expected resolution for the size of the population (0.26cM on 196 individuals). Since the mapping population contained only one critical recombinant between *Yr15* and the flanking markers, the population couldn't yield to a better map. To improve the map, a cross from the

SNP haplotype			Reaction to <i>P. striiformis</i>		
R11	R5	R8	Resistant	Intermediate	Susceptible
C	A	T	-	6	16
T	A	T	-	11	-
T	G	C	79	1	-

Figure 3.16: Haplotype analysis and phenotypic evaluation of the 113 doubled haploid lines used in the study. The TGC haplotype corresponds to that originally identified in the *Yr15* parent and which was diagnostic across 112 of the 113 lines studied.

two critical recombinants could be used to repeat a similar analysis, but sequencing with either exome capture or RenSeq.

As described in Ramirez-Gonzalez et al. (2015c):

The markers R11, R5 and R8 were tested across 122 doubled haploid (DH) lines. These DH lines were derived from crosses between five different UK varieties/breeding lines to *Yr15* derivatives known to carry the resistance gene. The expected *Yr15* haplotype corresponded to T, G and C alleles at markers R11, R5 and R8, respectively (TGC haplotype). The DH lines were tested at seedling stage for reaction to *P. striiformis*, with 84 showing complete resistance and 34 presenting an intermediate or completely susceptible reaction. The resistant lines all carried the complete *Yr15* haplotype (TGC, Figure 3.16) across the three SNP markers with the exception of five lines which had a single missing data point, but were otherwise consistent. This compared favourably with the most diagnostic in-house SNP markers available within the breeding programmes. Using the three in-house markers, 79 resistant lines carried the expected haplotype, but five completely resistant DH lines were scored as false negative due to the presence of the non-*Yr15* haplotype. Within the intermediate and susceptible DH lines, all but one had a non-*Yr15* haplotype (CAT or TAT) across R11, R5 and R8 (Figure 3.16). This single DH line was scored as a false positive as it carried the TGC *Yr15* haplotype, but was found to have an intermediate (chlorotic) reaction to *P. striiformis*. This

Talk about
Why is
R33 di-
agnostic
on the
varieties,
but
maps
away?.

line was also the only one scored as a false positive using the three in-house markers.

The fact that the developed markers perform better than the markers developed by breeders show the value of this particular experiment and further confirms that BSA combined with NGS is an effective way to develop novel markers.

In this chapter the integration of different levels of data helped to improve the selection of the candidate SNPs. The main criteria for selecting SNPs was theBFR score. However, thanks to the genetic map from Wang et al. (2014) and the CSS scaffolds (Mayer et al., 2014)) enabled to confirm that the high scores were in the expected region. As the reference genome for wheat improves, the location of SNPs linked to a trait will be easier. With a continuous reference between two markers flanking a locus and an improved annotation, a more focused set of gene candidates will be possible.

This is a very long quote, but I'm finding hard to shorten it.

Mention other people using a similar strategy since this was published.

Chapter 4

expVIP: a customisable RNA-seq data analysis and visualisation platform.

4.1 Background.

4.1.1 Alternative expression browsers.

To the best of my knowledge, the only alternative expression browser developed for specifically for wheat is WheatExp (Pearce et al., 2015b). This expression browser contains information of 6 studies from diploid, tetraploid and hexaploid wheat. The studies were selected to be complementary among them; " a broad study of five different tissues across multiple timepoints (Choulet et al., 2014), a study of seedling photomorphogenesis (Fox et al., 2014), a study of drought and heat stress in wheat seedlings (Liu et al., 2015), a study of wheat grain layers at a single time-point (Pearce et al., 2015a), a senescing leaf timecourse (Pearce et al., 2014) and a timecourse of different grain tissue layers during development (Pfeifer et al., 2014)". The expression quantification is produced by aligning to the gene transcripts using BWA (Li and Durbin, 2010), and the counts are extracted with HTSeq (Anders et al., 2015). The Expression Atlas from EBI is a public resource that collects expression experiments of several species (Petryszak et al., 2016). The samples are processed automatically from the reads deposited ArrayExpress (Kolesnikov et al., 2015), containing data from microarrays and RNA-Seq. The studies included are manually curated and annotated with the relevant ontologies.

As of September 2016, 202 differential and 82 base experiments had been included. The visualisation is designed to explore each gene individually, showing all the related ontologies and details on the expression by factor, or by pairs of factor on a heatmap. The quantification is calculated with HTSeq (Anders et al., 2015), from alignments produced with Tophat (Ling et al., 2013).

introduce
BioGPS.
It is not
wheat
related,
but is
feature
rich.

4.1.2 Public wheat RNA-Seq experiments.

Table of experiments to include

4.1.3 Expression quantification with Kallisto.

Differential expression experiments try to elucidate which genes change under different conditions. To do that, a quantification of the levels of expressions is needed. When using RNA-Seq, the expression analysis usually consists on: aligning the reads to the genome or transcriptome reference and quantify the expression according to how many reads map to a region. However, this process usually takes around 6 hours per sample. Aligners such as `bwa` or `bowtie` produce a detailed alignment of each read, which is useful find polymorphisms (see Chapter 3) or to find novel alternative splices (Trapnell et al., 2012).

For expression analysis only the count of how many reads with a transcript is required, calculating the best local alignment and the output of each read is unnecessary. `Kallisto` is a tool that generates an index based on overlapping k-mers (sequences of size k), which are connected sequentially to represent each transcript on a transcriptome de Bruijn Graph (TDBG). For alternative splicings of the same gene, were some sequence overlap between transcripts, the connections produce two different sets of connections between k-mers. The k-mers on each read are then used to find the compatible transcripts across the TDBG and those are counted. Finally, the program estimates the abundance of the transcript in the sample (Figure 4.1; Bray et al. 2016).

Table 4.1: Studies with RNA-Seq replicates, sequenced with Illumina, at the time when expVIP was under development.

CHAPTER 4. EXPVIP

93

Study Id	Summary of study	Brief SRA description	Manuscript
DRP000768	Phosphate starvation in roots and shoots	Transcriptome profiles of wheat variety Chinese Spring (CS) in response to Pi starvation (-P) for 10 days.	Characterisation of the wheat (<i>Triticum aestivum L.</i>) transcriptome by <i>de novo</i> assembly for the discovery of phosphate starvation-responsive genes: gene expression in Pi-stressed wheat (Ono et al., 2013).
ERP003465	fusarium head blight infected spikelets	Near isogenic wheat lines, differing in the presence of the <i>Fusarium graminearum</i> FHB-resistance QTL Fhb1 and Qfls.fifa-5A, under disease pressure (30 and 50 hai) as well as with mock-inoculation	Quantitative trait loci-dependent analysis of a gene co-expression network associated with Fusarium head blight resistance in bread wheat (<i>Triticum aestivum L.</i>).. (Kugler et al., 2013).
ERP004505	grain tissue-specific developmental time-course	Analysis of the cell type specific expression of homeologous genes in the developing wheat grain	Genome interplay in the grain transcriptome of hexaploid bread wheat (Pfeifer et al., 2014).
SRP004884	flag leaf downregulation of GPC	Wild type bread wheat plants and GPC RNAi plants 12 days after anthesis	Effect of the down-regulation of the high Grain Protein Content (GPC) genes on the wheat transcriptome during monocarpic senescence (Cantu et al., 2011b).
SRP013449	grain tissue-specific developmental time-course	Transcriptomes of the aleurone and starchy endosperm tissues of the wheat seed (<i>Triticum aestivum</i>) at time points critical to the development of the aleurone layer of 6, 9 and 14 days post anthesis.	Gene expression in the developing aleurone and starchy endosperm of wheat (Gillies et al., 2012).
SRP017303	stripe rust infected	Pool of stripe rust infected wheat leaves seedlings	Genome analyses of the wheat yellow (stripe) rust pathogen <i>Puccinia striiformis f. sp. tritici</i> reveal polymorphic and haustorial expressed secreted proteins as candidate effectors (Cantu et al., 2013).
SRP022869	<i>Septoria tritici</i> infected seedlings	Molecular mechanisms underlying the interplay between fungal pathogenicity and host responses at specific growth phases and the factors triggering disease transition.	Transcriptional Reprogramming of Wheat and the Hemibiotrophic Pathogen <i>Septoria tritici</i> during Two Phases of the Compatible Interaction (Yang et al., 2013).
SRP028357	shoots and leaves of nulli tetra group 1 and group 5	RNAseq of nulli-tetrasomic wheat lines (shoots and leaves)	Patterns of homeologous gene expression shown by RNA sequencing in hexaploid bread wheat (Leach et al., 2014).
SRP029372	grain tissue-specific developmental time-course	Gene expression profiling of morphological stage of developing wheat grain	Evaluation of Assembly Strategies Using RNA-Seq Data Associated with Grain Development of Wheat (<i>Triticum aestivum L.</i> ; Li et al. 2013).
SRP038912	comparison of stamen, pistil and pistilliody expression	Transcriptional profiling of pistillody stamen, pistil and stamen in wheat line HTS-1	Pistillody mutant reveals key insights into stamen and pistil development in wheat (<i>Triticum aestivum L.</i> ; Yang et al. 2015).
SRP041017	stripe rust and powdery mildew time-course of infection in seedlings	Transcriptome Divergence and Overlap for Wheat in Response to Stripe rust and Powdery Mildew Pathogen Stress	Large-scale transcriptome comparison reveals distinct gene activations in wheat responding to stripe rust and powdery mildew. (Zhang et al., 2014).
SRP041022	developmental time-course of synthetic hexaploid	RNAseq of three tissues of nascent allohexaploid wheat and its following generations, their progenitors, and Chinese Spring	mRNA and Small RNA Transcriptomes Reveal Insights into Dynamic Homeolog Regulation of Allopolyploid Heterosis in Nascent Hexaploid Wheat (Li et al., 2014).
ERP008767	grain tissue-specific expression at 12 days post anthesis	Inner pericarp, outer pericarp and endosperm layers from developing grain of bread wheat cv. Holdfast at 12 days post-anthesis.	Heterologous expression and transcript analysis of gibberellin biosynthetic genes of grasses reveals novel functionality in the <i>GA3ox</i> family (Pearce et al., 2015a).
SRP045409	drought and heat stress time-course in seedlings	RNAseq of 1-week old wheat seedling leaves subjected to drought stress, heat stress and their combination before (0h) and after stress (1h or 6h)	Temporal transcriptome profiling reveals expression partitioning of homeologous genes contributing to heat and drought acclimation in wheat (<i>Triticum aestivum L.</i> ; Liu et al. 2015).
INRA-RNAseq (ERP004714)	developmental timecourse of Chinese Spring	Whole transcriptome sequencing of wheat 3B chromosomes	Structural and functional partitioning of bread wheat chromosome 3B (Choulet et al., 2014).
SRP056412	grain developmental timecourse with dormancy QTL	This study was to identify candidate genes underlying the 4AL QTL for grain dormancy in wheat. RNA was sequenced from pooled NILs segregating for the QTL	Transcriptomic analysis of wheat near-isogenic lines identifies <i>PM19-A1</i> and <i>A2</i> as candidates for a major dormancy QTL (Barro et al., 2015)

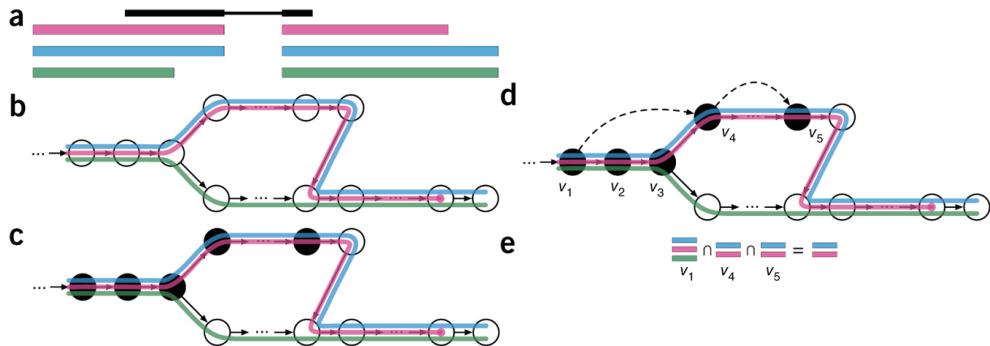


Figure 4.1: "Overview of kallisto. The input consists of a reference transcriptome and reads from an RNA-seq experiment. (a) An example of a read (in black) and three overlapping transcripts with exonic regions as shown. (b) An index is constructed by creating the transcriptome de Bruijn Graph (T-DBG) where nodes (v_1, v_2, v_3, \dots) are k-mers, each transcript corresponds to a colored path as shown and the path cover of the transcriptome induces a k-compatibility class for each k-mer. (c) Conceptually, the k-mers of a read are hashed (black nodes) to find the k-compatibility class of a read. (d) Skipping (black dashed lines) uses the information stored in the T-DBG to skip k-mers that are redundant because they have the same k-compatibility class. (e) The k-compatibility class of the read is determined by taking the intersection of the k-compatibility classes of its constituent k-mers" (Bray et al., 2016).

4.1.4 Relational databases.

A Relational database is a set of structured tables that have relationships between each other. The tables correspond to the data that is essential for the represented concept (domain). For example, in a table representing several species, the common name and the scientific name belong to the same domain (ie name: Bread wheat; scientific name *Triticum aestivum*). Tables in the same relational database form relationships between each other. Continuing with the example, a species can have several scientific studies related to them. The domain of a study can be formed by the accession, a title, a corresponding manuscript and the species it is concerned with. The two tables can be connected by the usage of a common ID signifying the species. It is therefore common practice to create an extra column containing unique integer values, to act as connecting keys between two or more tables. Such a column is defined as the primary key of the table; although, strictly speaking, any set of columns whose combination of values is unique for each row can act as primary key. In our example an extra `id` column is added allowing to create a stable relationship between the `Species` and the `Studies` tables.

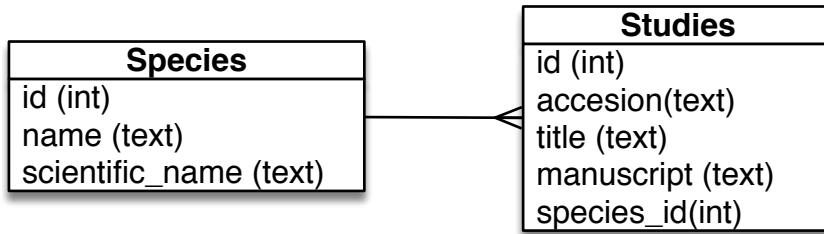


Figure 4.2: Example of a relationship between tables. The tables Species and Studies are related. Each study has one species and each species can have several studies.

Table 4.2: Example content for the table species

id	name	scientific_name
1	Bread wheat	Triticum aestivum
2	Yellow rust	Puccinia striiformis
3	wheat and rust	T.aestivum,S.tritici

(Figure 4.2; Codd 1970). The tables 4.2 and 4.3 have the content of their corresponding domains.

4.1.5 SQL.

Standard Query Language (SQL) is a common language to retrieve information from relational databases. SQL has operations to select columns and rows, join tables, group repeated values and order the results. Those simple operations are enough to retrieve information between tables (Oracle, 2014). The following list shows a brief description of some commands that can be used to build a query.

SELECT <EXPRESSIONS> . A list of columns or an expression that will be displayed, separated by commas (,). To display all the columns, the

Table 4.3: Example content for the table studies

id	accession	manuscript	species_id
1	DRP000768	10.1186/1471-2164-14-77	1
2	ERP003465	10.1186/1471-2164-14-728	1
3	ERP004505	10.1126/science.1250091	1
4	SRP004884	10.1186/1471-2164-12-492	1
5	SRP013449	10.1111/j.1467-7652.2012.00705.x	1
6	SRP017303	10.1186/1471-2164-14-270	2
7	SRP022869	10.1371/journal.pone.0081606	3

* character all the columns in the table. The order of the columns will be the same as the order given in this part of the command

FROM <TABLE> . follows the column names to add a list of tables to select from.

JOIN <TABLE> ON <EXPRESSION> . is used to join the table from the left side of the statement with the <EXPRESSION> given after the ON clause.

WHERE <EXPRESSION> filters the rows by the <EXPRESSION>

ORDER BY <COLUMNS> . The rows will be sorted by the natural order of the given <COLUMNS>.

GROUP BY <COLUMNS> . The rows are merged by the columns stated. This can be used to get an unique set of values and apply a function to all the rows that have the same value, as a count.

Expressions can be values, operators or functions like:

COLUMN The value of a column.

<EXPRESSION> = <EXPRESSION> TRUE when the left and right <EXPRESSION> are equal. FALSE otherwise

<EXPRESSION> > <EXPRESSION> TRUE when the left <EXPRESSION> is greater than the right <EXPRESSION> are equal. FALSE otherwise

<EXPRESSION> < <EXPRESSION> TRUE when the left <EXPRESSION> is less than the right <EXPRESSION> are equal. FALSE otherwise

COUNT(*) The count of rows that have the same values, as selected in the GROUP BY clause.

A simple query to join the **species** and **studies** tables and displaying only the species name, scientific name and accession of the study is shown in Listing 4.1. The results of the query are in Table 4.4.

Listing 4.1: Join example query

```

1 SELECT
2   species.name,
3   species.scientific_name,
4   studies.accession,
5 FROM species
6 JOIN studies ON species.id = studies.species_id;

```

Table 4.4: Join of the `species` and `studies` table.

name	scientific_name	accession
Bread wheat	Triticum aestivum	DRP000768
Bread wheat	Triticum aestivum	ERP003465
Bread wheat	Triticum aestivum	ERP004505
Bread wheat	Triticum aestivum	SRP004884
Bread wheat	Triticum aestivum	SRP013449
Yellow rust wheat and rust	Puccinia striiformis T.aestivum,S.tritici	SRP017303 SRP022869

The relationships between tables can be of the following types:

one-to-one. When rows on a table can be related to a row in a second table. On the diagrams they are represented by a straight line.

many-to-many. Rows on a table can have many corresponding rows in a second table, represented with lines with whiskers on both sides of the line.

one-to-many. Rows on a table can be related to many rows on the second table, represented with whiskers only on one side of the line.

An important feature of a database is the ability to store the data consistently. A transaction is a set of related operations that need to be performed at the same time; as such, it needs to follow the principles of Atomicity, Consistency, Isolation and Durability (ACID) (Haerder and Reuter, 1983).

Atomicity. All the operations or none have to be performed. If any of them fails or an error happens while the transaction is executed, the data has to be restored to the original status.

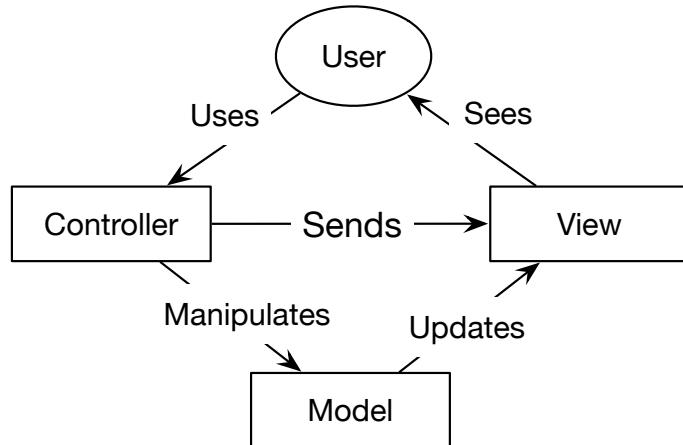


Figure 4.3: MVC interaction between components.

Consistency. The changes in the database have to be valid before and after the transaction.

Isolation. If more than one transaction is being executed at the same time, the result must be the same as if the transactions were executed one after the other.

Durability. The result of the transaction is stored even if the server is restarted.

Several Relational Database Management System (RDBMS) implement SQL, with various levels of compliance to the standard and different licenses. A popular RDBMS is MySQL. From the beginning MySQL aimed to be a lightweight and easy to install open source product (Oracle, 2014). This characteristics made it popular on the web and it is currently the RDBMS behind ensembl! (Flicek et al., 2012).

Add explanation of inserts.

4.1.6 Model-View-Controller

The Model View Controller (MVC) is a metaphor to isolate the user interactions from the underlying data. The models hold the data on logical their domains. The views contain the layout on how the models are displayed to the user. The controllers receive the requests from the users and modify the models hold the data as mapped to their logical domains accordingly and send a view back for display. The MVC metaphor allows the development of independent parts of the system and helps to structure the underlying representation of the domains. (Figure 4.3; Krasner and Pope 1988).

Ruby on Rails (RoR) is a framework to develop web applications heavily influenced by the MVC metaphor. It is based on the Rails language and provides several tools designed to facilitate development, such as automated tasks designed to create models with their corresponding views and controllers. On the top of that, it provides the tools to manage the connection and queries to the RDBMS, allowing the developer to focus on functionality (Rails Guide, 2016).

4.1.7 Data visualisation

In the last couple of decades the amount of information available in any given field has been growing exponentially, in part thanks to the internet. A standing challenge is therefore to produce tools that help interpretation. An effective way to communicate large amount of data is through visualisation, but it has to have the following properties to be usable (Myatt and Johnson, 2011):

Time to learn. The user needs to take little time to learn how to use the system to extract the information they need. Also, all the features should be easy to find.

Performance. The tool needs to be quick to access and transform the visualisation as the user requests.

Accuracy. The tool has to perform as the user expects, so if the tool is prone to make users commit mistakes, it is not accurate.

Memorability. Once the user learns to use the system, is it easy to remember how to use it? Systems that change their interface often, or between windows are not as easy to remember.

Satisfaction. The user responds positively to the use of the system and the time is spent actually exploring the data, rather than trying to make the system work.

4.1.8 Aims

The aims of expVIP are to:

1. Integrate RNA-Seq experiments from several sources in a single database (Section 4.3).

2. Automate the calculation of the expression values and load them in to the database (Section 4.4).
3. Produce a usable visualisation for said expression values (Section 4.5).
4. Make the system available to the community (Section 4.5).

The software developed in this chapter is published in Borrill, Ramirez-Gonzalez, and Uauy (2016).

4.2 General design

One of the main objectives of expVIP is to make the public expression datasets easily accessible and explorable for the target community (currently wheat, but not limited to it). A web interface is an effective way to reach a global audience. A web service requires to have a server to run the application, and a browser to connect to the server and display the application to the user (ie Internet Explorer, Chrome, Firefox). The web server technology used for expVIP is RoR, as it abstracts the MVC metaphor and it is designed to speed development (Rails Guide, 2016). In order to display the expression data to the users, expVIP relies on a BioJS component (Yachdav et al. 2015, Section 4.5) developed for expVIP. All the data is stored in a MySQL database (Section 4.3) and it is accessed through models developed under RoR (Figure 4.4).

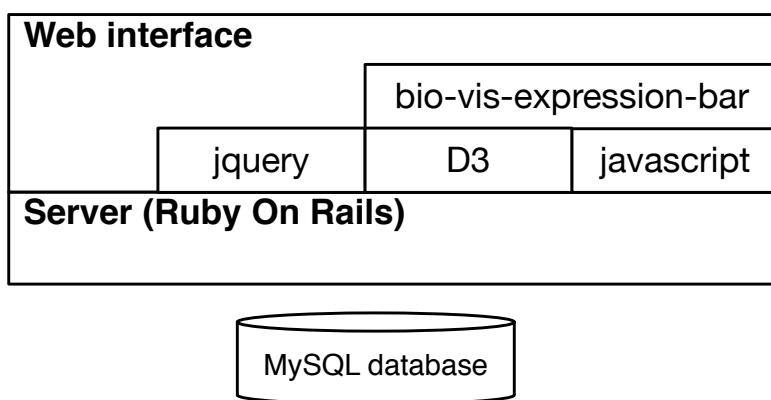


Figure 4.4: General design of expVIP

4.3 Database design

To address the different types of conditions over different experiments, expVIP is designed around a relational database. The design comprises two core groups of tables and two auxiliary tables that take care of different species and homoeologues (Figure 4.5).

Metadata The tables in this group contain the information of each one of the studies.

Studies holds general information for a study, which contain several experiments. The table also contains the reference to the paper where the data is published and the accession number for the study.

Experiment group keeps together all the individual experiments that come from the same study and that were taken on the same condition (ie. replicates).

Factors holds all the possible factors used to group the experiments. Each experiment group has many factors and each factor group has many experiment groups. As the experiment does not have a fixed number of columns representing each factor, it is possible to have any arbitrary number factors to group.

Experiment holds the information of each individual experiment, with the corresponding accession.

Expression values. The tables on this block contain the information for each genes and their expression values.

Types of value keeps a list of different units that are stored. On the original design TPM and raw counts are set up, but as the units are not hard coded it is possible to use FPKM, RPKM or any other unit.

Gene Set contains the name of a reference gene set for the analysis. On the original version of expVIP, the gene models from the IWGSC as deposited in Ensembl release 26 were used (Mayer et al., 2014). However, this table allows to use several reference gene models on the same database.

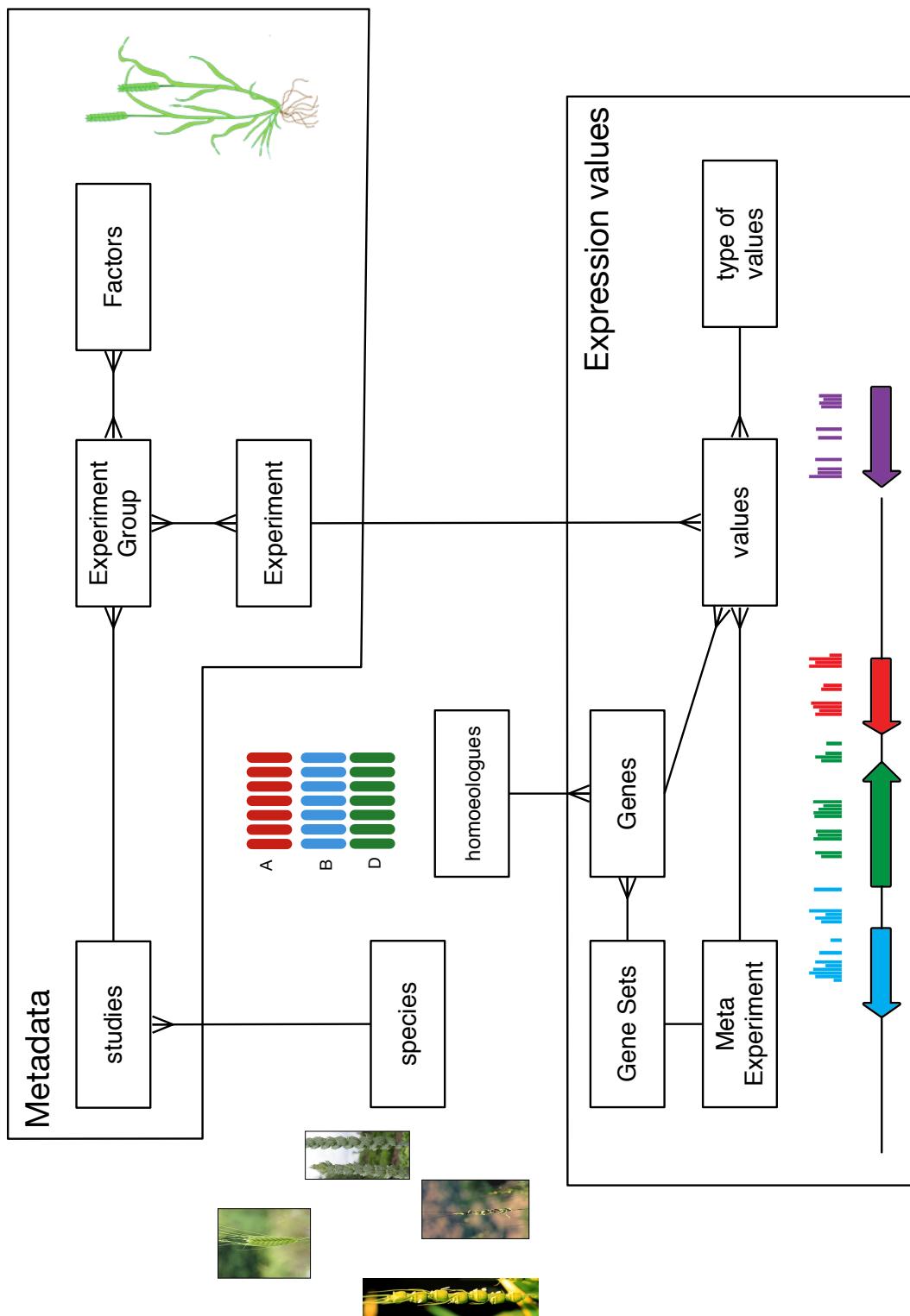


Figure 4.5: Database design. The block on the top stores the meta-data about the experiments and the studies. The bottom block consist on the tables related to the expression values. Species and homoeologues are outside the main blocks as they are not core to the groups. The whiskers in the connections show the cardinality of the relationships.

Genes are related to a `gene set`, so even if two genes coming from two different datasets share the same name it is possible to distinguish them and avoid ID collisions. This situation is unlikely to occur when using published references, but might arise when joining several *de novo* gene model datasets.

Meta Experiment allows to have the same data analysed with different tools. By default expVIP uses Kallisto (Bray et al., 2016). However other tools, or different versions of the same tool, can be used to repeat the analysis.

Values have a domain that includes the `meta experiment`, `gene` and, `type of value`.

Homoeologues contains the relationship between genes. This allows to get the expression values of several related genes.

Species contains the target species of a study. It is not linked to the gene models to allow the direct comparison between related species using the same gene models (ie, *T. aestivum* vs *T. turgidum*).

In the cases where a relationship between tables is not unique, such as `experiment_groups` having many `factors` and the `factors` having many `experiment_groups`, storing of relationships is done with an auxiliary table (ie. `ExperimentGroups_Factors`, not explicitly shown in Figure 4.5, but implicit by the lines with whiskers).

Once all the data is stored, the tables can be queried together to make clear the relationship between specific rows. One of the core tasks of expVIP is to get all the factors that define each experiment, in order to be able to merge similar studies. To retrieve the `experiments` and `factors` of an `experiment group`, the auxiliary tables `ExperimentGroups_Factors` and `experiment_groups_experiments` are used in the query. (Listing 4.2 and Table 4.5).

Listing 4.2: Query experiments and factors
Query experiments and factors from accession 'DRR003148'

```

1 SELECT
2   experiments.accession,
3   factors.factor,
4   factors.description,
5   experiment_groups.name as expriment_group
6 FROM factors
7 JOIN ExperimentGroups_Factors
8   ON factors.id = ExperimentGroups_Factors.factor_id
9 JOIN experiment_groups
10  ON experiment_groups.id = ExperimentGroups_Factors.
11    experiment_group_id
11 JOIN experiment_groups_experiments
12  ON experiment_groups_experiments.experiment_group_id =
13    experiment_groups.id
13 JOIN experiments
14  ON experiments.id = experiment_groups_experiments.
15    experiment_id
15 WHERE accession = 'DRR003148'
```

Table 4.5: Results of querying the metadata for accession 'DRR003148' (Listing 4.2)

accession	factor	description	expriment group
DRR003148	Age	24 days	Group1
DRR003148	High level age	vegetative	Group1
DRR003148	High level stress-disease	no stress	Group1
DRR003148	High level tissue	roots	Group1
DRR003148	High level variety	Chinese Spring	Group1
DRR003148	Stress-disease	none	Group1
DRR003148	Tissue	roots	Group1
DRR003148	Variety	Chinese Spring	Group1

Likewise, to get the `expression_values` for a `gene` with the corresponding unit (`type_of_values`) and `experiment` a simple query joining the four tables is used. The Listing 4.3 retrieves the `expression_values` for the `gene` 'Traes_5BS_0AFC3F795.1', and the result is on Listing 4.6

Listing 4.3: Query values from 'Group1' and gene 'Traes_5BS_0AFC3F795.1'

```

1 SELECT
2   genes.name as gene,
3   expression_values.value,
4   experiments.accession,
5   type_of_values.name as unit
6 FROM expression_values
7 JOIN genes
8   ON expression_values.gene_id = genes.id
9 JOIN type_of_values
10  ON type_of_values.id = expression_values.
11    type_of_value_id
11 JOIN experiments
12  ON experiments.id = expression_values.experiment_id
13 WHERE
14   genes.name = 'Traes_5BS_0AFC3F795.1'

```

Table 4.6: Results of query to get the values for gene 'Traes_5BS_0AFC3F795.1' (Listing 4.3), only 'Group1' is displayed from the output.

gene	value	accession	experiment	unit
group				
Traes_5BS_0AFC3F795.1	136.995	DRR003148	Group1	count
Traes_5BS_0AFC3F795.1	120.683	DRR003149	Group1	count
Traes_5BS_0AFC3F795.1	140.94	DRR003150	Group1	count
Traes_5BS_0AFC3F795.1	24.2277	DRR003148	Group1	tpm
Traes_5BS_0AFC3F795.1	23.9739	DRR003149	Group1	tpm
Traes_5BS_0AFC3F795.1	24.9835	DRR003150	Group1	tpm

With those two queries is enough to retrieve all the information required to do sub-groupings.

The database is implemented using the RDBMS MySQL 5.5.

4.4 Data integration pipeline

To prepare the database, expVIP requires to have all the metadata for the experiments to integrate. ExpVIP contains tasks to load all the metadata and a wrapper for Kallisto that can be run from expVIP. Alternatively, the expression values can be calculated with another tool and loaded as

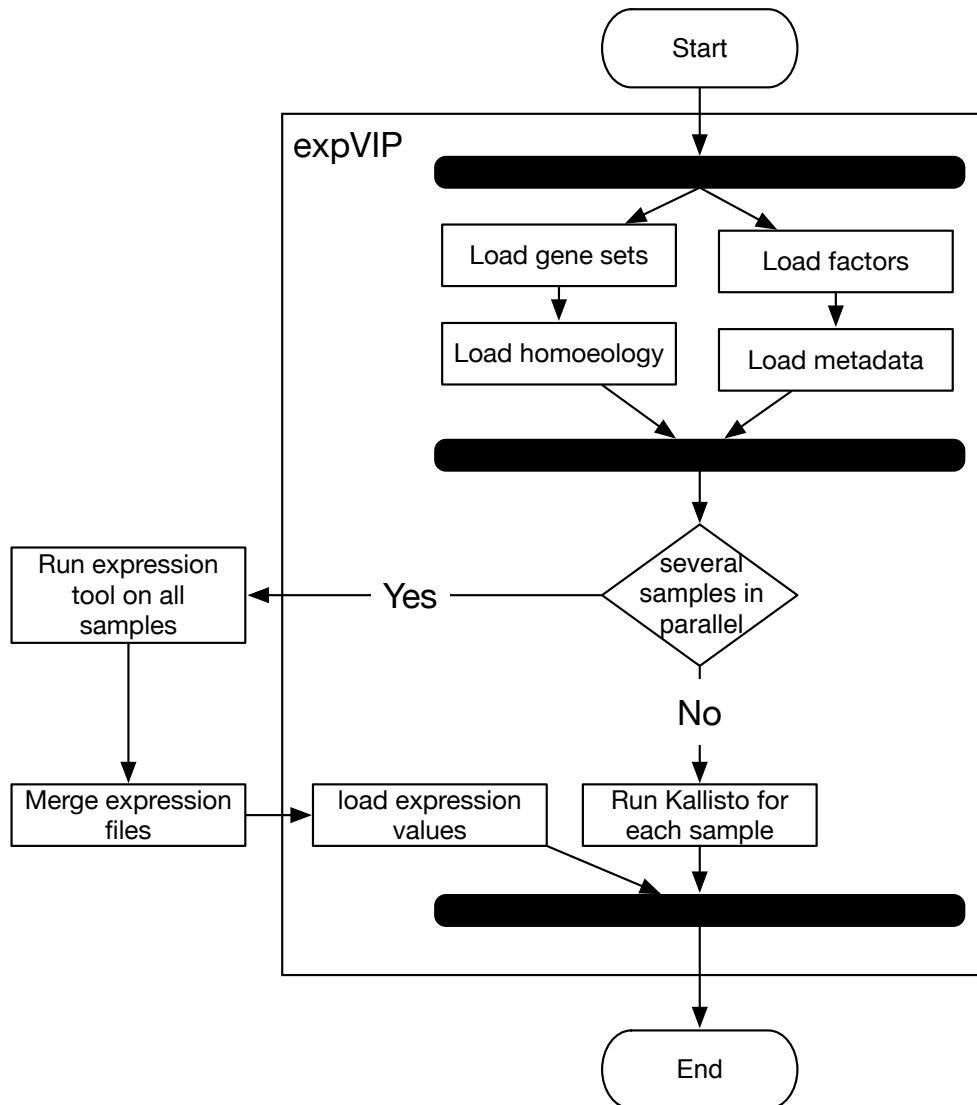


Figure 4.6: The pipeline for loading the data into expVIP. The black lines represent a border of tasks that are not required to be executed in a particular order.

a single file, this approach is preferred for a large set of samples (Figure 4.6). Details on how to load the files in the database are in the expVIP tutorial (Appendix C).

The required files for the metadata are:

Factors. The file contains all the possible factors that can be used to group all the experiments. The file must contain the following columns (Table 4.7):

factor The category were the factor belongs. In the case of the initial dataset used in expVIP, the grouping factors are: Age,

Table 4.7: Factors file. The table must be saved as a text file, with columns separated by tabs

factor	order	name	short
Age	1	7 days	7d
Age	2	seedling stage	see
Age	3	14 days	14d
Age	4	three leaf stage	3_lea
Age	5	24 days	24d
High level age	1	seedling	see
High level age	2	vegetative	veg
High level age	3	reproductive	repr
High level stress-disease	1	none	none
High level stress-disease	2	disease	dis
High level stress-disease	3	abiotic	abio
High level stress-disease	4	transgenic	trans
High level tissue	1	spike	spike
High level tissue	2	grain	grain
...			

stress-disease, tissue and a corresponding 'High level' for each factor. The metadata file must contain a column corresponding to each one of this factors.

order The default order in which to display each factor. This ensures that the age of the plants is sorted chronologically.

name Long description of each level for the category. The values in this column must match the values in the metadata file (see below).

short Is a short name, used when the space to display the full description of the factor is not enough.

metadata The metadata file is the file that contains the information related to each study and the corresponding experiments. Each study contains several experiment groups (replicates), which in turn contain every individual experiment. The factors must be shared across experimental groups.

secondary_study_accession The accession number for experiments carried as part of a single study. This is usually the high level BioProject or SRA number.

run_accession The accession of the individual run.

scientific_name of the species.

experiment_title A description for the individual RNA-seq sample.

study_title A description of the general study.

Manuscript The DOI of the study.

Group_for_averaging A description of the experiment. This must be the same all the replicates in the same study.

Group_number_for_averaging A short name for replicated experiments.

Total reads (optional)

Mapped reads (optional)

Besides the main fields, each factor has a corresponding column Variety, Tissue, Age, Stress-disease, High level variety, High level tissue, High level age and High level stress-disease

Gene set The gene set is provided as a single **fasta** file. The file may contain alternative transcripts from the same gene. To identify this, the **fasta** header may include the optional fields **gene** and **transcript**. In the absence of this, the only stored value is the name derived from the header, going from the >character to the first space on the line (Listing 4.4).

Listing 4.4: A fasta entry on of the gene set.

```

1 >Traes_5BL_3FC5BA305.1 cdna:novel scaffold:IWGSC2:
   IWGSC_CSS_5BL_scaff_1082268:5:199:-1 gene:
   Traes_5BL_3FC5BA305 transcript:Traes_5BL_3FC5BA305
   .1
2  TGCTGCTGCTAGGCTTGAAGAGGTTGCTGGCAAGCTCCAGTCTGCTC
3  GGCAGCTCATTCAAGAGGGCTGTGAGGAGTGCCCCAAGAACGAGGT
4  GTTTGGTCGAGGCATGCCGGTGGCTAGCCCAGATGAGTCAGAGGC
5  AGTAATTGCCAGGGGTGTGAAGGCAATTCCAAGTCTGTGAAGCTGT
6  GGCTGCA

```

homoeologues A file containing the homoeologues for the A, B and D genomes. Currently these are the only supported default names.

The file also includes a column with the gene name and to which Group (ie 1, 2, 3 ... 7) and Genome (ie A, B or D) it belongs (Table 4.8).

expVIP includes several tasks to load the different files. For example, to load the factors the `load_data:factor` starts a transaction (Listing 4.5; line 2) to ensure that all the data is loaded, and if for some reason the load fails, the database is restored to the previous status. In the transaction, the file is read row by row using the `csv` library (line 3). The function `find_or_create_by` is a function that RoR provides on models to create an entry in the table, or update it if already exists. Each row is used to create or update a `Factor` (lines 374-376). A similar strategy is used for all the files that are regular tables.

Listing 4.5: Task that loads factors

```

1 task :factor, [:filename] => :environment do |t, args|
2   ActiveRecord::Base.transaction do
3     CSV.foreach(args[:filename], :headers => true, :
4       col_sep => "\t") do |row|
5       factor = Factor.find_or_create_by(:factor=>row["factor"], :description=>row["name"], :name=>row
6         ["short"])
7       factor.order = row["order"].to_i
8       factor.save!
9     end
10   end
11 end

```

The gene sets are loaded slightly differently, as the input is a `fasta` file, as opposed to tabular file. The reader for the `FastaFormat` from BioRuby (Goto et al., 2010) is used to read the file (Listing 4.6; line 4). Since expVIP only records the name of the genes, only the id of the fasta sequence is extracted (lines 6-79). The name is stored in the name and cDNA columns. The parser for entries from ensembl, such the one in Listing 4.4 include code to load the cDNA and transcript fields correctly.

Table 4.8: Example tabular file containing the homoeology across the three genomes.

Gene	A	B	D	Group	Genome
Traes_5BS_0AFC3F795	Traes_5BS_0AFC3F795	Traes_5BS_0AFC3F795	Traes_5DS_0AFC3F795	C204EBAA9	5
Traes_5DS_C204EBAA9	Traes_5DS_C204EBAA9	Traes_5DS_C204EBAA9	Traes_5DS_0AFC3F795	C204EBAA9	5
Traes_7DL_82360D4EE1	Traes_7DL_82360D4EE1	Traes_7DL_82360D4EE1	Traes_7DL_82360D4EE1		7
Traes_2AL_1368BE0AD		Traes_2AL_1368BE0AD	Traes_2AL_1368BE0AD	CD459994C1	2
...					A

Listing 4.6: Task that load genes from a fasta file

```

1 task :de_novo_genes, [:gene_set,:filename] => :environment
2   do |t, args|
3     ActiveRecord::Base.transaction do
4       gene_set = GeneSet.find_or_create_by(:name=>args[:gene_set])
5       Bio::FlatFile.open(Bio::FastaFormat, args[:filename])
6         do |ff|
7           ff.each do |entry|
8             arr = entry.definition.split(/\s+/)
9             name = arr[0]
10            g = Gene.new
11            g.gene_set = gene_set
12            g.name = name
13            g.cdna = name
14            g.save!
15          end
16        end
17      end
18    end
19  end
20
```

There are two options to load the expression values from the database: a matrix with all the expression values and running **Kallisto** from expVIP.

The task in Listing 4.7 loads the expression values from a tabular file with the genes as rows and the values as columns. The exception handling and messages are removed. The task requires the following arguments:

meta_experiment. A name for the analysis. This can be the name of the tool used for the expression quantification combined with the name of the reference, as a single text variable.

gene_set. The reference used for the analysis.

value_type. The unit of the file (ie. TPM, count)

filename. The file that is going to be loaded in the database.

The steps to load the values are:

1. A transaction is initiated at the beginning of the task, to ensure that if any step fails and the execution is aborted the database will stay in a consistent state (Listing 4.7; line 2).

2. The connection is assigned to the variable `conn`, to be able to execute queries directly to the database (line 3).
3. The `meta_experimet`, `gene_set` and `value_type` are loaded and stored to get the corresponding IDs in the insertion (line 4).
4. All the `Genes` and `Experiments` are loaded in their corresponding hash table, to be able to get the IDs when the actual values are inserted (lines 7-11).
5. The file is read with the CSV library from Ruby, keeping the headers to be able to assign the correct experiment (line 14).
6. The first column is named `target_id` and contains the gene name. The ID of the gene in the database is retrieved from the previously loaded hash (lines 15-16)
7. Each column is iterated and the values needed to execute the insertion to the database are concatenated.
8. Whenever the number of queued insertions reaches 1,000, the command to perform the insertions is executed (line 25).
9. As the number of genes is not usually a multiple of 1,000, when the process finished reading the file an extra insertion is executed to empty the queue (line 30).

The decision to execute the insertions in batches of 1,000 objects is to reduce the number of processes running in the database, while keeping low the memory usage of the application. This approach is faster than using the functions for insertions RoR on multiple values. For trivial operations, the functions from the framework are used, as they are easier to maintain (compare insertion in line 4 to the block of code from line 18 to 26).

Listing 4.7: Task to load the expression values from a tabular file.

```

1 task :values, [:meta_experiment, :gene_set, :value_type, :
2   filename ] => :environment do |t, args|
3   ActiveRecord::Base::transaction do
4     conn = ActiveRecord::Base.connection
5     meta_exp = MetaExperiment.find_or_create_by(:name=>args[:meta_experiment])
6     gene_set = GeneSet.find_by(:name=>args[:gene_set])
7     value_type = TypeOfValue.find_or_create_by(:name=>args[:value_type])
8     experiments = Hash.new
9     meta_exp.gene_set = gene_set
10    genes = Hash.new
11    Gene.find_by_sql("SELECT * FROM genes where gene_set_id
12      ='#{gene_set.id}'").each{|g| {genes[g.name] = g.id}}
13    Experiment.find_each{|e| experiments[e.accession] = e.id}
14    count = 0
15    inserts = Array.new
16    CSV.foreach(args[:filename], :headers => true, :col_sep =>
17      "\t") do |row|
18      gene_name = row["target_id"]
19      gene = genes[gene_name]
20      row.delete("target_id")
21      row.to_hash.each_pair do |name, val|
22        val = val.to_f
23        str = "(#{experiments[name]},#{gene},#{meta_exp.id},#{{
24          value_type.id},#{val},NOW(),NOW())"
25        inserts.push str
26      end
27      count += 1
28      if count % 1000 == 0
29        sql = "INSERT INTO expression_values (`experiment_id
30          ,`gene_id`, `meta_experiment_id`, `
31          type_of_value_id`, `value`, `created_at`, `
32          updated_at`) VALUES #{inserts.join(", ")}"
33        conn.execute sql
34        inserts = Array.new
35      end
36    end
37    sql = "INSERT INTO expression_values (`experiment_id`, `
38      gene_id`, `meta_experiment_id`, `type_of_value_id`, `
39      value`, `created_at`, `updated_at`) VALUES #{inserts.
40      join(", ")}"
41    conn.execute sql
42  end
43end

```

Alternatively, expVIP can execute **Kallisto** on all the samples loaded in the database. For this purpose, expVIP stores the raw reads in FastQ format, organized in directories named with the same accessions as in the metadata (ie a directory named DRR003148 contains the reads for the metadata displayed in table 4.5). expVIP takes all the accessions for the experiments in the database and searches for the corresponding folder. If the folder exists and if it contains the **fastq** files, then it is deemed as valid. If the folder already has the **kallisto** output, the next folder is evaluated, otherwise **kallisto** is executed with its default settings and the results are loaded into the database.

This process is repeated for all the accessions (Figure 4.7). This pipeline allows to populate the database partially, in case that not all the experiments are ready from the beginning.

New experiments can be added to the existing metadata file or to a new file; the expVIP loading procedure then can be run again to update the list of experiments and the corresponding expression values. This design allows to keep the database updated as more experiments become available. The fact that the loading is done in transactions ensures that the database is kept consistent, regardless of potential errors in the input files.

4.5 Graphical interface

With the expression across experiments integrated in a single database, the next challenge is to make the data accessible to a wide audience. RoR has tasks to automate the construction of controllers and views from the MVC metaphor, helping on the retrieval and formatting of the raw data. However, the main objective was to visualise the data from

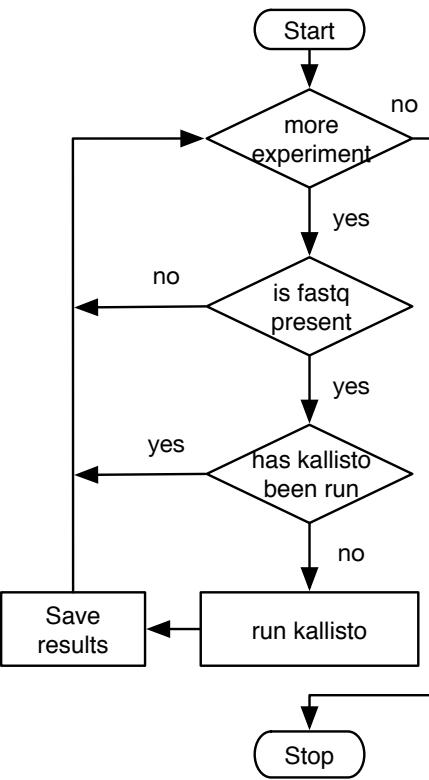


Figure 4.7: Steps to run and load Kallisto

all the experiments as intuitively as possible. To make the visualisation dynamic in a browser, the use of **JavaScript** is necessary, as it is the only widely adopted programming language used for web content. Among the tools built on the top of **JavaScript**, **D3** is a framework designed explicitly to do dynamic visualisations (Bostock et al., 2011).

Usability was a top priority on the design of the visualisation component for expVIP. The time needed to learn, performance and accuracy were taken into account when designing expVIP. As memorability and satisfaction are subjective, they were not directly tested for during development.

Time to learn. The controls are located in two blocks, one for global controls (ie. unit, save plot) and one to modify the factors, close to the factor they affect (Figure 4.8).

Performance. All the data for the genes being displayed is loaded from the database in a single transaction. The data is available in the cache of web browser and whenever the user changes anything in the visualisation, new values are calculated locally.

Accuracy. As knowing what the users will do is not obvious, the visualisation was given to a panel of potential users (other members of the Uauy lab) for comments. In previous versions the legends were a confusing and the location of the buttons was too distant from the aspect of visualisation that they controlled. After reviewing the feedback, the accuracy was improved. Also, the position of the controls don't change, regardless of the representation of the data (bar plot or heatmap).

The elements in the graphical are shown in Figures 4.9, 4.11 and 4.12, with a description of each element of the GUI listed below.

1. **Search box:** at any point it is possible to type a new gene name (based on Ensembl Plants nomenclature) and generate a new set of expression data.
2. **Compare box:** it is possible to add a second gene and press the Compare button to generate two expression graphs drawn at the same scale.

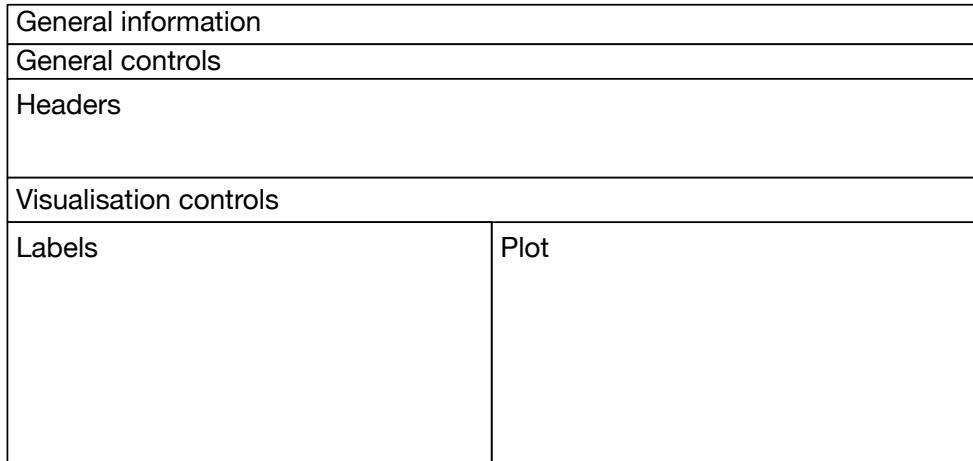


Figure 4.8: expVIP Graphical User Interface components components. The top bar shows a short description of what is displayed. The General and Visualisation controls contain the buttons and menus that enable the interaction with the figure. The Headers, Labels and Plot are the actual components of the visualisation

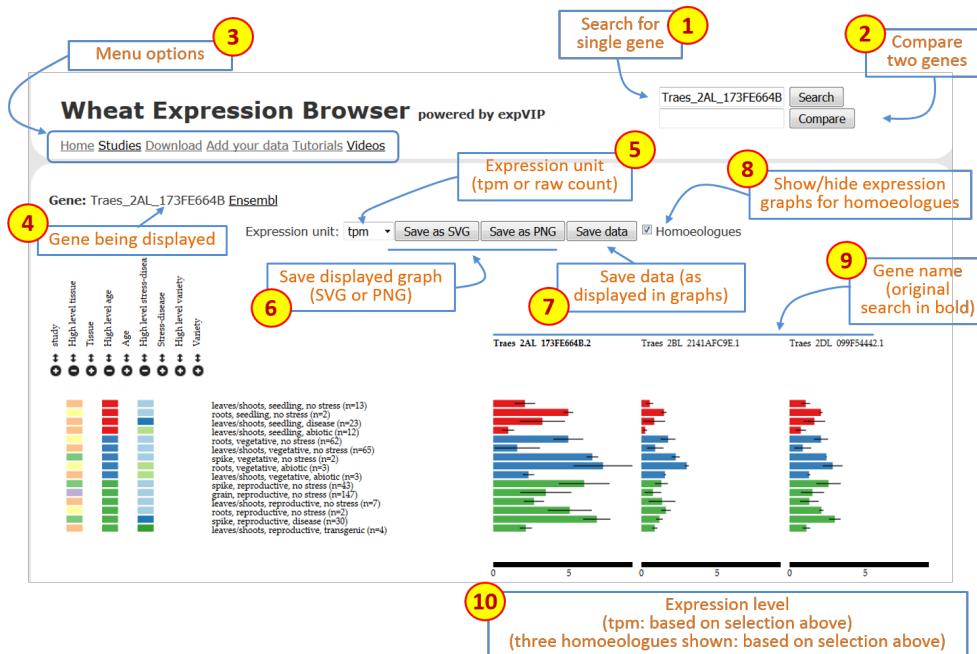


Figure 4.9: Features on expVIP

(a)

Headers	
Labels	Plot

(b)

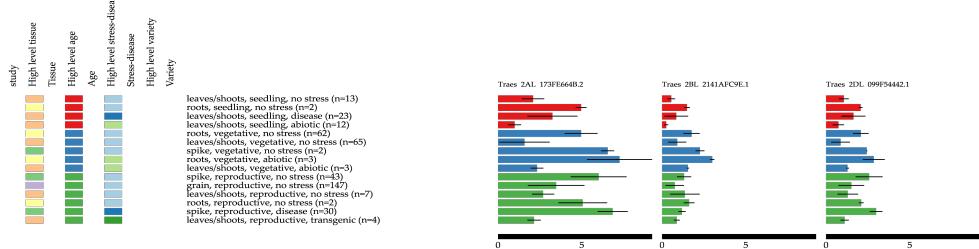


Figure 4.10: expVIP export image. (4.10a) GUI components components exported in image

(4.10b) The exported components from Figure 4.9) .

3. **Menu options:** has several links on the details of the study and tutorial. The menu can be edited to customize instances of expVIP.
4. **Gene:** shows the currently displayed gene, with a link to Ensembl to the corresponding description.
5. **Expression unit:** selects the expression unit to visualise. This can be either Transcripts per Million of mapped reads (TPM) or estimated counts. If other units are loaded in the database, they will appear in this field.
6. **Save graph:** these two buttons allow users to save the current graphs in either SVG (to work on Adobe Illustrator) or as PNG files. The export process selects the graphical elements from the visualization and binds them together on a single SVG file. The reasons to follow this process are: to ensure that the plot reflects the user selection; remove the elements that do not have a meaning in a static context and; allow the user to format the image with publication quality (Figure 4.10).

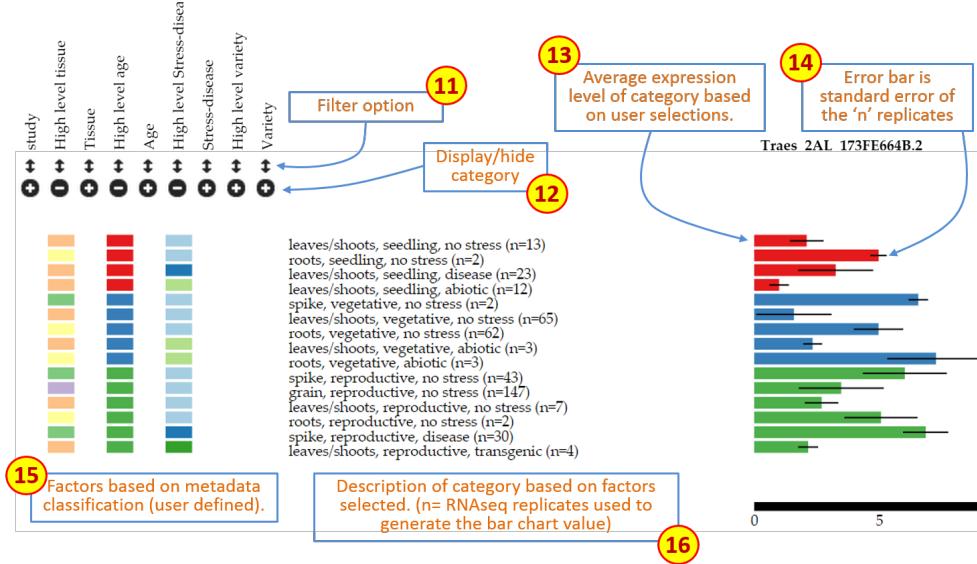


Figure 4.11: Features on expVIP (continued)

7. **Save data:** downloads a csv file with the data with the current selection and order of factors as displayed on the screen. The data will include the standard errors and the number of samples that make up each value. An example of how the output looks is in Listing 4.8.
8. **Homoeologues:** this button displays the graphs of known homoeologues of the original primary gene. This gene is highlighted in bold and the homoeologous graphs will be displayed according to A, B, D genome ordering. The scale of all the homoeologues is the same to facilitate the comparison between them.
9. **Gene names:** each displayed gene is labelled on this block. If the list of genes is too long, the gene names are rotated for readability. In case that a gene name is too long to fit, the font is scaled to the largest size that will fit in the designated area.
10. **Expression level:** the expression level adjusts according to the expression of each set of gene homoeologues. The scale remains consistent across homoeologues to allow comparison. The values are based on the unit selected in the expression unit box (see point 5 above).
11. **Filter:** opens a pop-up window revealing the levels of the selected category. By default, all the levels are selected, but the users can

decide to exclude from the visualisation experiments containing any level. The order of the levels can be modified by dragging the levels on the the pop-up window.

12. **Display/hide category:** Each category can be displayed or hidden by pressing the $+/ -$ button. As categories are added or removed the expression graphs show the new values for the new groups. Data is not removed when changing the displayed categories, instead the values are distributed according to the new groups (the number of samples remains the same). The colours below the category correspond to each level, and the plot is coloured according to the sorting category.
13. **Expression bars:** These bars represent the expression level for the n grouped samples according to the chosen categories (11 and 12 above). When hovering over the bar with the mouse a small tooltip will appear, containing the expression level (tpm or counts) and the standard error (sem) used for the error bars (see 14)
14. **Error bars:** Standard error of the means for the n expression values on which the bar graph is based.
15. **Factors:** The colour of the rectangles represents the displayed categories, according to the selection criteria (11 and 12 above). To clarify the meaning of the colour, hovering above the rectangle displays a tooltip with the long name of the examined level.
16. **Description:** Textual description of the grouped factors, according to the selection criteria (11 and 12 above). The number of grouped samples is also displayed.
17. **Expression unit:** For heatmaps, the default unit $\log_2(\text{tpm})$ as the logarithmic scale provides a better context for comparisons across several genes.
18. **Heatmap:** To compare several genes, the values are represented as a heatmap. The sorting and filtering is done with the same controls as for single genes. Up to 50 genes are displayed in the heatmap, as more genes will degrade the performance of the database, and the visual clutter makes the plot hard to interpret. This view allows to

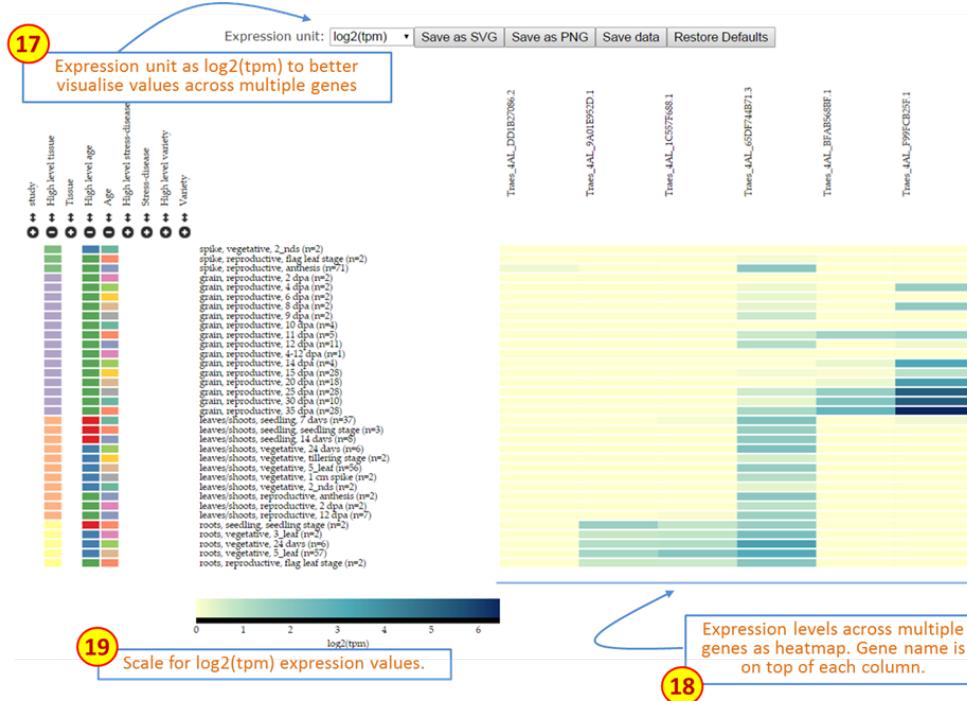


Figure 4.12: Features on expVIP for Multiple gene comparisons

visualise several candidate genes for a trait expressed under certain conditions and quickly asses which one is a good candidate for further research.

19. **Scale:** The scale is calculated according to the highest displayed value in the current heatmap. Since logarithmic values below 1 result in negative values and anything with a TPM under 2 is considered as very low expressed, every value lower than 1 is plotted as 0.

For a comprehensive user manual see Appendix C.

If I have time, I'll add the section about the virtual machine, as I would also need to add something in the background on virtualisation, it can potentially be a sink of time

Listing 4.8: Export data example, corresponding to the data plot in Figure 4.9

```
1 High level age seedling, vegetative, reproductive,
2 High level stress-disease no stress, disease, abiotic, transgenic,
3 High level tissue spike, grain, leaves/shoots, roots,
4 High level variety Chinese Spring, other, Nullitetra Chinese Spring,
5 tpm SEM tpm SEM tpm SEM
6 Traes_2AL_173FE664B.2 Traes_2AL_173FE664B.2 Traes_2BL_2141AFC9E.1 Traes_2BL_2141AFC9E.1 Traes_2DL_099F54442.1 Traes_2DL_099F54442.1
7 roots, vegetative, no stress(n=62) 4.96 0.99 1.77 0.49 2.08 0.47
8 roots, vegetative, abiotic(n=3) 7.26 1.95 3.00 0.13 2.85 0.66
9 leaves/shoots, vegetative, no stress(n=65) 1.59 1.50 0.91 0.55 0.87 0.56
10 leaves/shoots, vegetative, abiotic(n=3) 2.33 0.38 1.55 0.07 1.29 0.09
11 spike, reproductive, disease(n=30) 6.85 0.90 1.19 0.22 2.99 0.38
12 spike, reproductive, no stress(n=43) 6.01 1.67 1.32 0.43 2.58 0.81
13 grain, reproductive, no stress(n=147) 3.47 1.69 0.76 0.55 1.51 0.77
14 leaves/shoots, reproductive, transgenic(n=4) 2.15 0.40 0.88 0.18 1.11 0.25
15 leaves/shoots, reproductive, no stress(n=7) 2.69 0.67 1.37 0.89 1.30 0.62
16 leaves/shoots, seedling, disease(n=23) 3.25 1.50 0.85 0.71 1.64 0.72
17 leaves/shoots, seedling, no stress(n=13) 2.10 0.67 0.55 0.23 1.07 0.29
18 leaves/shoots, seedling, abiotic(n=12) 0.99 0.39 0.25 0.12 0.74 0.34
19 roots, seedling, no stress(n=2) 4.96 0.32 1.49 0.17 2.07 0.11
20 roots, reproductive, no stress(n=2) 5.06 1.46 1.62 0.32 2.10 0.14
21 spike, vegetative, no stress(n=2) 6.55 0.39 2.27 0.27 2.43 0.03
```

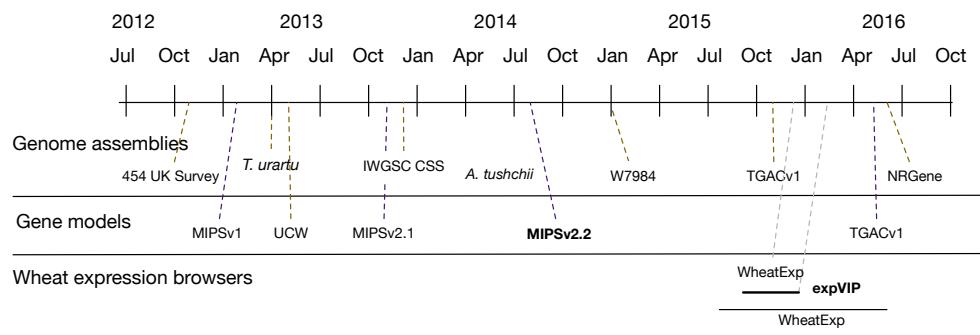


Figure 4.13: expVIP resources time line. In bold the line of the time for development of expVIP and the annotation used. The Expression Atlas has a line starting on the initial study deposited on it till the last update, as of September 2016.

4.6 Discussion.

In model organisms there are several on-line resources that aggregate the raw data and meta analysis of several studies. For example, the Expression Atlas EBI includes over 2,000 studies for *Arabidopsis thaliana* inclusive of other technologies besides RNA-Seq (ie Affymetrix expression arrays). However, when I started the development of expVIP the Expression Atlas only included a couple of baseline experiments and WheatExp had not been published yet (Figure 4.13).

WheatExp was developed roughly at the same time of expVIP and contains 6 studies (against 16 on expVIP; Pearce et al. 2015b). Four of the studies in WheatExp are on hexaploid wheat and are included in expVIP as well. In WheatExp, the expression for each gene is displayed on the context of the original experiment without a direct comparison between studies. However, because the studies are kept independently the levels of the categories are a direct reflection of the original studies. WheatExp allows to search genes by sequence, a feature not implemented in expVIP.

The Expression Atlas from EBI, as of September 2016, contains 10 studies, 4 baseline and 6 for differential expression (Petryszak et al., 2016). ExpVIP contains studies released before EBI started to upload expression experiments for wheat. Even if some of the most recent studies in the Expression Atlas are not included in expVIP yet, we are in the process of updating our tool to include studies published after the initial release. In terms of visualisation, the EBI included a heatmap to compare to different factors at the same time for given gene (ie. tissue vs stress);

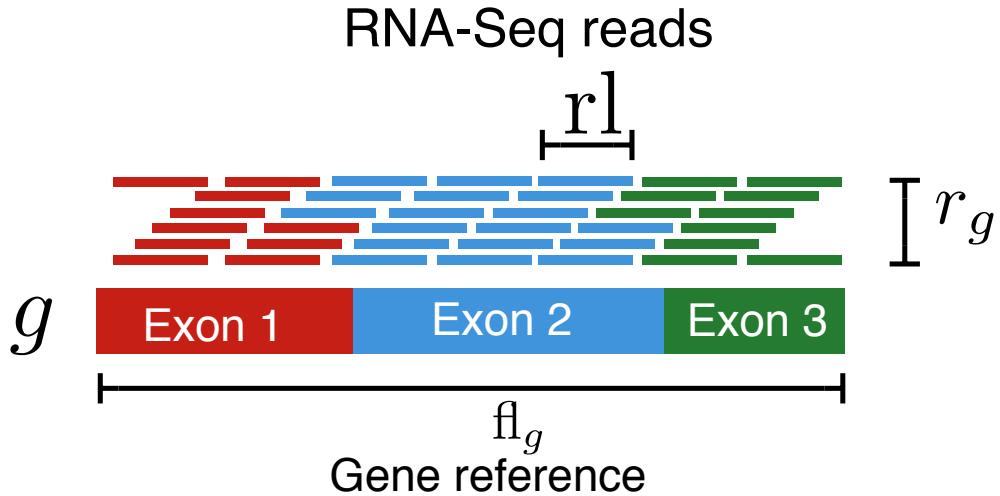


Figure 4.14: Units used for expression normalisation

the same information can be displayed on expVIP by sorting by two factors.

Most of the RNA-Seq studies report their results in terms of RPKM. This normalisation, which is computed for done for each feature g in the reference G , requires the count of the number of reads r_g , the feature length fl_g and the total number of mapped reads R (Figure 4.14; equations 4.1 and 4.2 Mortazavi et al. 2008). As the denominator of 4.2 is based on the number of mapped reads rather than the number of nucleotides covered, RPKM does not allow to compare correctly results obtained between different samples, or even results from the same sample when the average read length changes due to variations in the experimental protocol Wagner et al. 2012.

$$R = \sum_{g \in G} r_g \quad (4.1)$$

$$\text{RPKM}_g = \frac{r_g \times 10^9}{fl_g \times R} \quad (4.2)$$

The TPM is an alternative to RPKM that approximates the total number of transcripts T as a normalisation factor (equation 4.3). Besides the previously described parameters, the TPM also includes the read length rl , which is dependent on the study. This formula assumes that each read corresponds to a full observed transcript (equation 4.4; Wagner et al. 2012).

$$T = \sum_{g \in G} \frac{r_g \times \text{rl}}{\text{fl}_g} \quad (4.3)$$

$$\text{TPM}_g = \frac{r_g \times \text{rl} \times 10^6}{\text{fl}_g \times T} \quad (4.4)$$

One of the aims in developing TPM was to be able to compare samples from different studies; as it is more stable across different experiments, we decided to use it as the main unit of comparison in expVIP. After we took the decision of which unit to use, we found a couple of tools, Kallisto (Bray et al., 2016) and Sailfish (Patro et al., 2014), that could calculate the TPMs directly from mapping the reads to a reference, without producing a precise alignment. The main advantage of only doing mapping, without aligning is a significant reduction in both the computational resources and the time needed to analyse a sample.

The traditional pipeline to quantify expression from RNA-Seq consists on the following steps:

1. Index the reference. Only done once, as the same index can be used for all the samples.
2. Align the reads to the reference.
3. Sort the alignment and remove duplicates.
4. Quantify the expression.

This is the prevailing pipeline for expression analysis. In my experience, on a computing cluster it takes between 6 to 8 hours to process each wheat sample on a computing cluster, using multiprocessing and up to 24 Gb of RAM (Figure 4.15a). This pipeline is usually implemented by aligning the reads with **BWA** (Li and Durbin, 2009) or **tophat** (Trapnell et al., 2012) and the quantification is performed with tools like **HTSeq** (Anders et al., 2015) or **cufflinks** (Trapnell et al., 2012).

An advantage of using a mapper is that the quantification pipeline is shorter:

1. Index the reference. Only done once, as the same index can be used for all the samples.
2. Map the reads to the reference and quantify the expression in a single program.

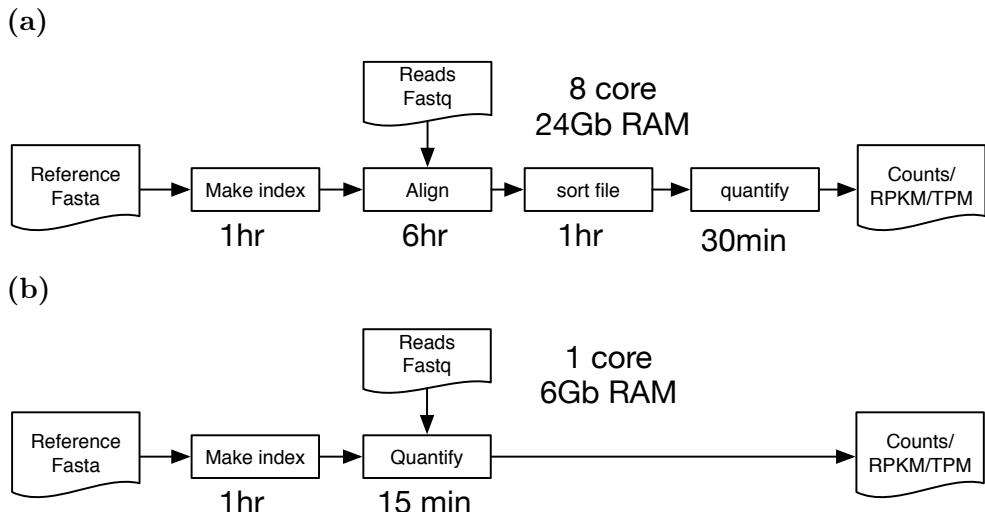


Figure 4.15: Alignment vs mapping pipelines for RNA quantification. (a) Alignment pipeline. (b) Mapping pipeline.

Since mapping does not require the precise alignment of every single base on the read and the output is only the quantification for each gene, as opposed to the alignment of each read, the programs implementing mapping take around 15 minutes to run. RNA-Seq mapping algorithms are implemented by **Kallisto** (Bray et al., 2016) and **Sailfish** (Patro et al., 2014).

I decided to integrate **Kallisto** into expVIP because the algorithm is able to walk through different splicing junctions via a T-DBG (see Section 4.1.3; Bray et al. 2016), as opposed to Sailfish, which is based on counting k-mers only (Patro et al., 2014). Since homoeologues with a high level of identity may form a bubble in the graph, in a similar way to small indels or SNPs (Leggett et al., 2013), **Kallisto** should be able to assign the reads to the correct k-compatibility class for their corresponding homoeologue.

Some features that could improve expVIP in the long term and that already present in other expression browsers would include searching by sequence, by gene ontology and a heatmap of a particular gene displaying two different factors on each axis. A feature that expVIP could leverage from other genomic resources is the retrieval of genes flanking a gene of interest, or between genetic markers. The upcoming assemblies , from the IWGSC (Clark, 2016) and TGACv1 (Pozniak, 2016) with longer scaffolds in conjunction with the high resolution genetic maps from Wang

Make
 a dia-
 grame
 with the
 graph.
 Unfor-
 tunately,
 I don't
 have in
 hand
 my dia-
 grams
 for cor-
 tex an
 Richard
 did the
 dia-
 grams
 for
 Legget
 et al.

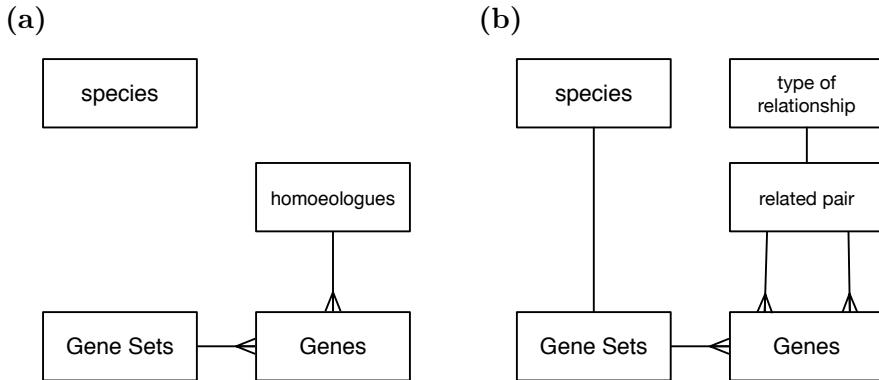


Figure 4.16: Changes to improve gene comparisons. (a) Current implementation of homoeology. (a) Proposed implementation to extend the types of relationships between genes.

et al. (2014) and Chapman et al. (2015) can enable such kind of queries (resources described in Section 1.5).

For the TGACv1 assembly, an improved wheat annotation was made available after the initial release of expVIP (Figure 4.13). The IWGSC will provide an updated annotation for their genome as well. In the near future, expVIP will be updated to include those annotations and some development will be needed to allow the comparison between annotations, at least while the community adopts a canonical reference.

The same mechanisms to compare expression between different references can be used to compare the expression between different organisms. However, the current implementation uses the homology table with one column for each genome group in hexaploid wheat (A, B, and D; Figure 4.16a). To be able to allow the inclusion of different polyploids with different genome names, to compare homologues and paralogues effectively and to add any arbitrary relationship between genes the homoeologues table needs to be updated. Instead of representing triplets, the table should contain binary relationships; each gene pair will have a type to be able to distinguish between relationships. Furthermore, each gene set should be linked to a species. With that explicit relationship, equivalent genes from the same species, but coming from different gene models can be identified. Likewise, genes known to be conserved across relatives (ie Barley vs Wheat genes) can be compared through these modifications to the database (Figure 4.16b).

To the best of my knowledge, none of the expression browsers available for wheat, or other polyploid species, allow the direct comparison

between homoeologues. However, the effect of different related genes is a topic of active research in polyploids. Making the relationship of the expression between related genes easily accessible can provide some initial evidence of having an uniform expression across homoeologues or of a triplet with a dominant gene. Likewise, when the update to the table that keeps arbitrary relationships between genes is completed, it will be possible to find if related genes have a conserved expression pattern.

Since its conception, I wanted expVIP to be open source and available to the community. As part of the project, I released the visualisation component as a biojs component (`bio-vis-expression-bar`; <http://biojs.io/d/bio-vis-expression-bar>; Yachdav et al. 2015). In a collaboration with the Earlham Institute and eLife, another PhD student is working to integrate the visualisation plugin as a live figure. If I had kept the code closed, this potential use of the component would have never happened.

The webserver is also open and hosted on github (<https://github.com/homonecloco/expvip-web>), with a tutorial on how to load the data on a personal server (Appendix C). As the quantification tool used by expVIP has a relatively low memory requirement, I was able to prepare and preconfigure two virtual machines, one without any data loaded and one with all the data used in the original article <http://www.wheat-expression.com>. This allows the comparison of private projects in the context of previous studies. On the Norwich Research Park, at least two groups have shown interest on using expVIP to make their data available to the community on other species. We had also been contacted by both the team behind the annotation of TGACv1 and the IWGSC to include the upcoming annotations to the public instance of expVIP.

The public server has received visitors from several research institutes around the globe (Figure 4.17). Since April, when the google analytics tracker was setup in the website, over 1,500 users have visited the site. Most of the users are based in the UK (34%), and within the nation the majority of the traffic originates from Norwich, Cambridge, Harpenden, London and Dundee. Those cities have research institutes that work on wheat, so it is very likely that the users are real. Most of the international users are coming from the US, China, Australia, India, Germany and Canada. In those countries, most of the visitors also come from cities where wheat institutes are based. Furthermore, around half of the

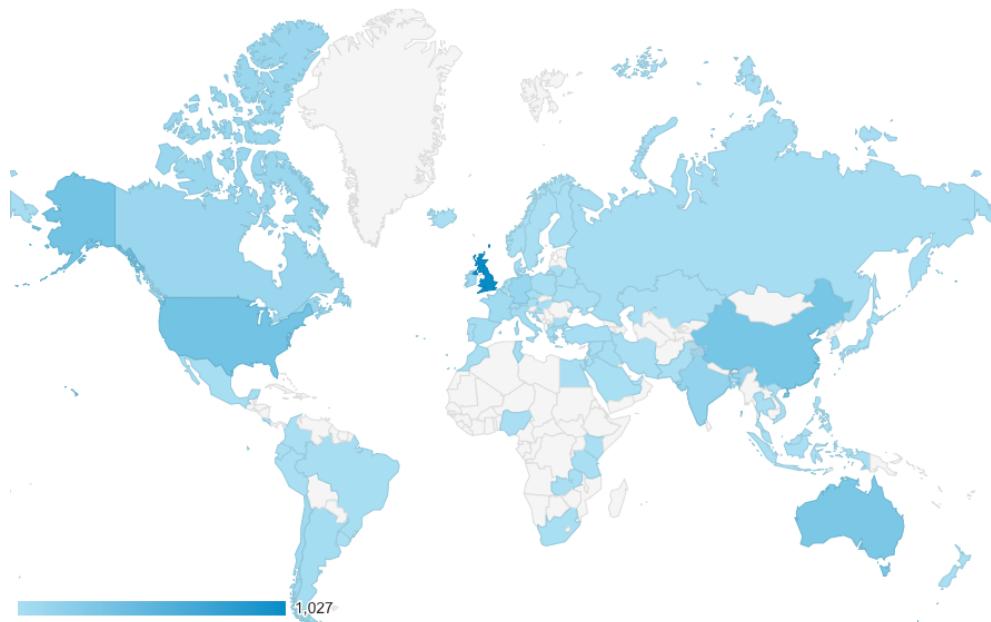


Figure 4.17: Heatmap of the number of unique sessions of expVIP in the world.

sessions stay on the website to access either the the individual genes or the heatmap. Around 25% of the sessions consult their genes using the heatmap, suggesting that they have a list of candidate genes for a trait of interest to select by comparing their expression.

Overall, expVIP has met its aims (Section 4.1.8). I designed a relational database capable of storing several RNA-Seq experiments with their corresponding metadata. The expression analysis has been automated, to facilitate the process of running Kallisto, the selected quantification tool. The stored data can be visualised to compare the expression across several conditions, and the visualisation tool allows the arbitrary grouping and selection of studies. Finally, the open source code and the virtual machines facilitate the adoption of expVIP on other communities, not necessarily working on wheat.

Chapter 5

General discussion and final remarks

Knowledge from computer science can be applied to produce software for specific needs, and which can also be useful for the community. One of the limitations though is that most approaches are developed for diploid systems and are sometimes not compatible with polyploid species, such as wheat. Polyploidy adds an extra level of complexity (due to homoeologs) and in the case of wheat the large genome size also hinders certain approaches to genome analyses. Therefore bespoke tools are required to deal with this barriers.

In this thesis we have taken advantage of developments in the technology and in resources to generate a series of solutions to many of the problems associated with polyploidy. These methods and approaches are not restricted to wheat, but can be applied and implemented to other polyploid systems.

I have discussed individual elements of project at the end of each chapter. However, looking forward my interest is to integrate this data into a single scheme. I outline this in Figure 5.1 and discuss it below.

5.1 Integration of different genetic maps

Genetic maps are a common starting point to start the search for a locus linked to a trait. For example, in Chapter 3 previous genetics map had already identified the short arm of chromosome 1B as the locus for *Yr15* (Murphy et al., 2009). Furthermore, I was able to confirm an enrichment of SNPs linked in the expected region thanks to the mapping of the

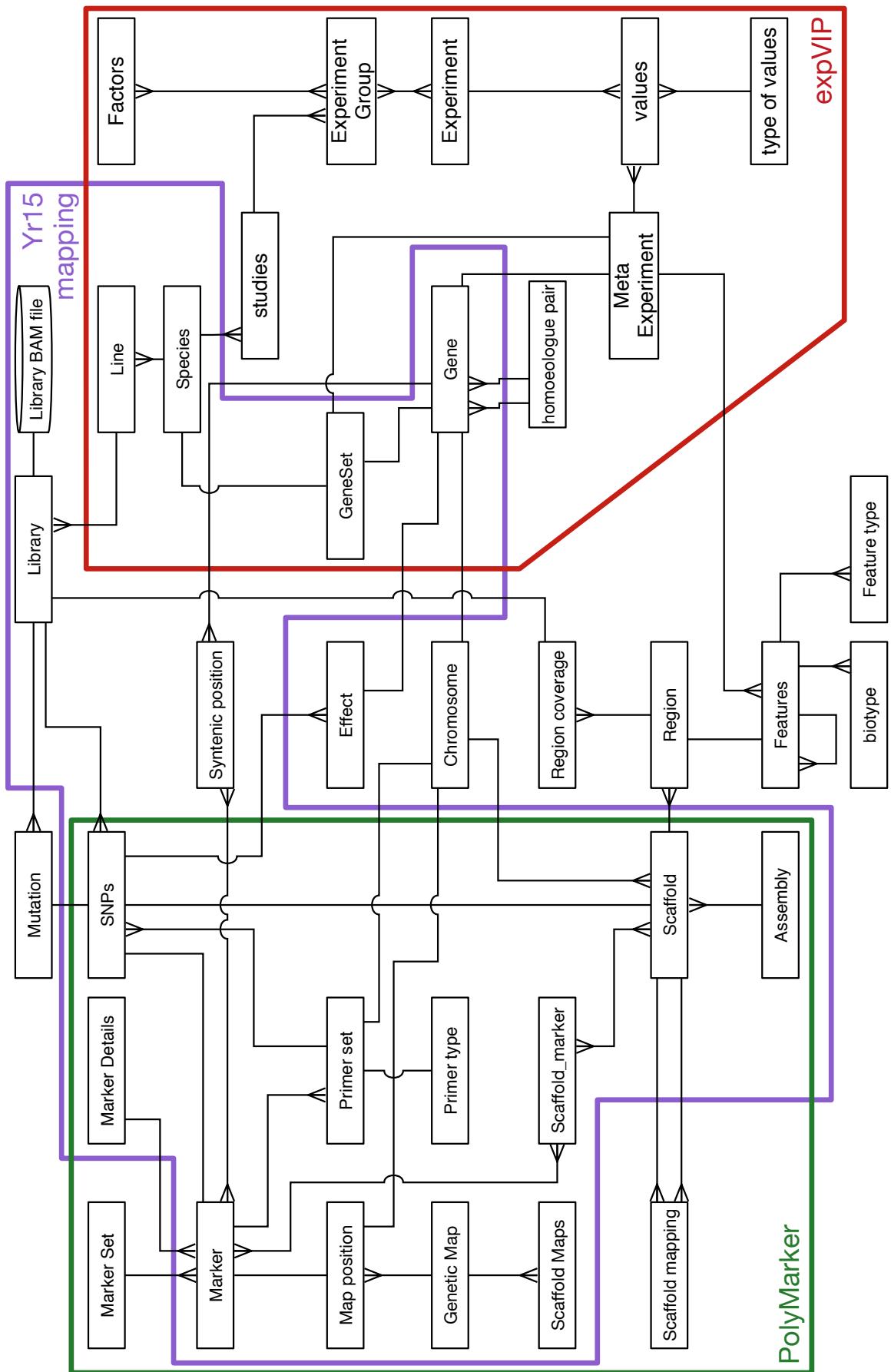


Figure 5.1: Relational database integrating all the datasets. The boxes represent the tables that contain information related to each chapter.

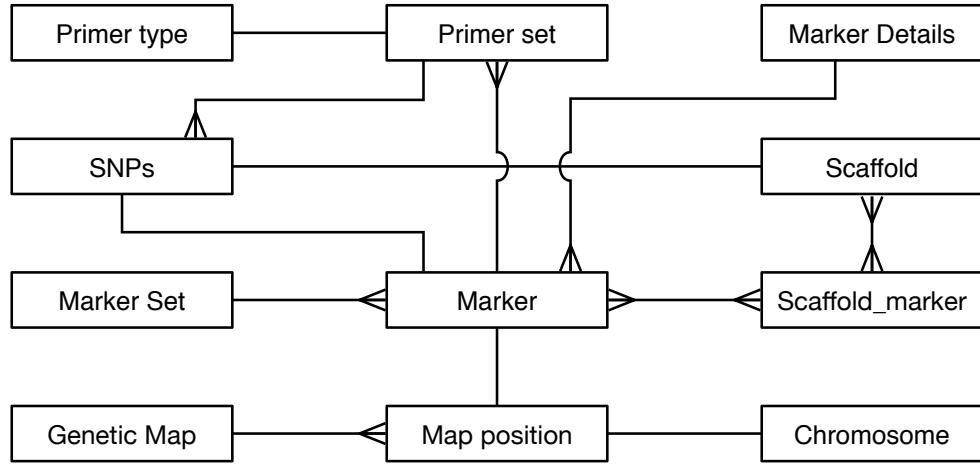


Figure 5.2: Tables to store information about genetic maps.

markers included in the genetic map from Wang et al. (2014) to the CSS scaffolds (Mayer et al., 2014). Since my initial analysis, other genetic maps with a higher resolution have been published (Chapman et al., 2015; Allen et al., 2016; Winfield et al., 2016). The relationship between the tables used in the genetic map is shown in Figure 5.2

Genetic maps are produced from the genotype of several markers (marker sets). Those markers can be developed explicitly for the genetic map or from an already published marker set, usually in the form of an SNP array. A database containing several genetic maps should be able to distinguish the origin of the used markers. However, a genetic map may contain markers from several marker sets. For those reasons, a genetic map is not connected directly to a marker set, but the relationship is maintained through the markers and their position in the genetic map.

To be able to use the genetic marker in a genomic context, the sequence of the markers must be mapped to a reference. The assembly may be in a single pseudomolecule molecule or in separated scaffolds. If the assembly is fragmented, the map can be used to anchor the scaffolds to a genetic position. If the assembly consists of long scaffolds the genetic map and the positions can be used to find if the lines used for the map have a rearrangement event. If a rearrangement is present, the collinearity between the genomic and genetic reference is not conserved. Having the markers and scaffolds in the database simplify this kind of analysis, as all the needed data is readily available.

Furthermore, PolyMarker (Chapter 2) has been used to design KASP assays for most of the primers in the 90k (Wang et al., 2014) and 820k

Winfield et al. (2016) SNP arrays. The primers for the assays can be integrated in the database. This allow a use case were known flanking markers are queried and the database can return a list of possible markers with the primers already designed to be validated on a mapping population.

Do we have any paper on the lab where this approach is used?

5.2 Integration of different genome references.

The efforts to produce a wheat reference genome had been focused on the Chinese Spring (CS) variety. CS is only cultivated as a research line, as it is susceptible to several pathogens and its yield is inferior to modern varieties. The reason for CS to be the selected cultivar to be sequenced as reference is historic: it has long been a variety for research. CS was originally chosen because it was able to cross with rye. It has also been used to produce lines with chromosomal aberrations, useful to find if any particular chromosome is responsible for certain traits (Sears and Miller, 1985).

New assemblies are required to address the shortcomings of the use of CS as a genome reference and to include the diversity from other lines (Allen et al., 2016; Winfield et al., 2016). The assemblies may include their own annotation, and that should be reflected in the database too.

As of september 2016 there are four sets of genomic sequence used by the wheat community:

1. A 454 whole genome shotgun sequence that is unassembled, and each read is treated as a scaffold (Brenchley et al., 2012).
2. The genome assembly and annotation from the CSS done by the IWGSC (Mayer et al., 2014).
3. A whole genome shotgun sequence from a synthetic wheat, without a corresponding annotation (Chapman et al., 2015).
4. The whole genome shotgun sequencing and annotation from CS (TGACv1; Clark 2016).

All those references can be aligned to each other to find corresponding region. The corresponding regions can be stored in the scaffold mapping table. Furthermore, the scaffolds can be mapped to relative species,

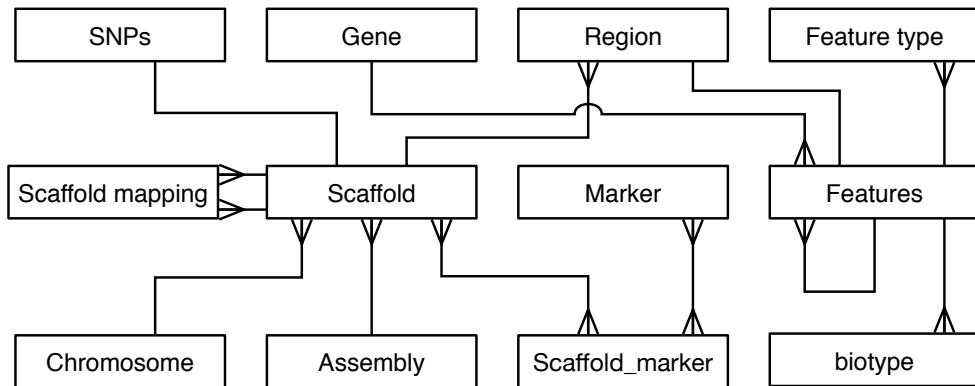


Figure 5.3: Tables to store information about genome assemblies and their annotation.

such as *Brachypodium distachyon*, *Oryza sativa*, *Sorghum bicolor*, and *Hordeum vulgare* to find synthenic blocks.

As each genome assemblies are usually annotated with their genes and other features. To include the annotation, each scaffold can contain several features. As some features consist on sub-features, like genes conformed by several exons, a recursive relationship is included in the features tables. With the support of the scaffold mapping, the different annotations can be projected on different references. Gene models, like the ones described in (Krasileva et al., 2013), and genetic markers can be aligned to any of the reference genomes.

With all those relationships available, the *In silico* mapping in Chapter 3 could be produced on several references at once. Also, the relationship between different gene models could be simplified, as the corresponding features will share coordinates. Likewise, the co-expression of genes located in the same region could be an useful feature for expVIP (Chapter 4).

Figure 5.3 include the tables used to store assemblies, the relationships between them and their annotation.

5.3 Integration with other services

Currently, the publicly available wheat resources are scattered as supplemental materials on their corresponding publication or available as ad hoc systems focused on a particular field. For example ensembl has two different views for every organism: from the genomic point of view and from the expression data. The Collaborative Open Plant Omics (COPO;

Davey et al. 2015) is trying to integrate different sources and types of data by connecting the data providers. This approach requires the co-operation of the service providers, which have their own view of what is important. I believe that in order to effectively integrate the resources it is necessary to understand how the users are likely to interact with the data.

5.4 Side projects

PolyInDel / .

5.5 Final remarks

In order to produce bioinformatic software that is powerful and usable it is required an understanding of both: the biological processes to solve and; the computational methods and software development practices.

Bibliography

- Abe, A., Kosugi, S., Yoshida, K., et al. Genome sequencing reveals agro-nomically important loci in rice using MutMap. *Nature biotechnology*, 30(2):174–8, February 2012. ISSN 1546-1696. doi: 10.1038/nbt.2095.
- Akhunov, E. D., Goodyear, A. W., Geng, S., et al. The organization and rate of evolution of wheat genomes are correlated with recombination rates along chromosome arms. *Genome research*, 13(5):753–63, May 2003. ISSN 1088-9051. doi: 10.1101/gr.808603.
- Alberts, B., Johnson, A., Lewis, J., et al. Molecular biology of the cell (6th edn). 2014.
- Allen, A. M., Barker, G. L. a., Berry, S. T., et al. Transcript-specific, single-nucleotide polymorphism discovery and linkage analysis in hexaploid bread wheat (*Triticum aestivum* L.). *Plant biotechnology journal*, 9(9):1086–99, December 2011. ISSN 1467-7652. doi: 10.1111/j.1467-7652.2011.00628.x.
- Allen, A. M., Winfield, M. O., Burridge, A. J., et al. Characterisation of a Wheat Breeders’ Array suitable for high throughput SNP genotyping of global accessions of hexaploid bread wheat (*Triticum aestivum*). *Plant Biotechnology Journal*, sep 2016. ISSN 14677644. doi: 10.1111/pbi.12635.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology*, 215(3): 403–10, October 1990. ISSN 0022-2836. doi: 10.1016/S0022-2836(05)80360-2.
- Anders, S., Pyl, P. T., and Huber, W. Genome analysis HTSeq — a Python framework to work with high-throughput sequencing data. 31 (2):166–169, 2015. doi: 10.1093/bioinformatics/btu638.

- Babraham Bioinformatics. FastQC A Quality Control tool for High Throughput Sequence Data, 2012. URL <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.
- Barrero, J. M., Cavanagh, C., Verbyla, K. L., et al. Transcriptomic analysis of wheat near-isogenic lines identifies PM19-A1 and A2 as candidates for a major dormancy QTL. *Genome Biol*, 16:93, 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0665-6.
- Bonnal, R. J. P., Aerts, J., Githinji, G., et al. Biogem: An effective tool-based approach for scaling up open source software development in bioinformatics. *Bioinformatics*, 28(7):1035–1037, April 2012. ISSN 13674803. doi: 10.1093/bioinformatics/bts080.
- Borrill, P., Ramirez-Gonzalez, R., and Uauy, C. expVIP: a customizable RNA-seq data analysis and visualization platform. *Plant Physiology*, 170(4):2172–2186, feb 2016. doi: 10.1104/pp.15.01667.
- Bostock, M., Ogievetsky, V., and Heer, J. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.
- Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5): 525–527, apr 2016. ISSN 1087-0156. doi: 10.1038/nbt.3519.
- Brenchley, R., Spannagl, M., Pfeifer, M., et al. Analysis of the bread wheat genome using whole-genome shotgun sequencing. *Nature*, 491 (7426):705–710, Nov 2012. ISSN 0028-0836. doi: 10.1038/nature11650.
- Burt, C., Steed, A., Gosman, N., et al. Mapping a type 1 fhb resistance on chromosome 4as of triticum macha and deployment in combination with two type 2 resistances. *Theoretical and Applied Genetics*, 128(9): 1725–1738, 2015. ISSN 1432-2242. doi: 10.1007/s00122-015-2542-9.
- Cantu, D., Segovia, V., MacLean, D., et al. Genome analyses of the wheat yellow (stripe) rust pathogen *Puccinia striiformis* f. sp. *tritici* reveal polymorphic and haustorial expressed secreted proteins as candidate effectors. *BMC Genomics*, 14:270, 2013. ISSN 1471-2164. doi: 10.1186/1471-2164-14-270\ n1471-2164-14-270[pii].
- Cantu, D., Govindarajulu, M., Kozik, A., et al. Next generation sequencing provides rapid access to the genome of *Puccinia striiformis* f. sp.

- tritici*, the causal agent of wheat stripe rust. *PLoS ONE*, 6(8):1–8, 08 2011a. doi: 10.1371/journal.pone.0024230.
- Cantu, D., Pearce, S. P., Distelfeld, A., et al. Effect of the down-regulation of the high Grain Protein Content (GPC) genes on the wheat transcriptome during monocarpic senescence. *BMC Genomics*, 12(1):492, 2011b. ISSN 1471-2164. doi: 10.1186/1471-2164-12-492.
- Chapman, J. a., Mascher, M., Buluç, A., et al. A whole-genome shotgun approach for assembling and anchoring the hexaploid bread wheat genome. *Genome Biology*, 16(1):1–17, 2015. ISSN 1465-6906. doi: 10.1186/s13059-015-0582-8.
- Choulet, F., Alberti, A., Theil, S., et al. Structural and functional partitioning of bread wheat chromosome 3B. *Science*, 345(6194):1249721–1249721, jul 2014. ISSN 0036-8075. doi: 10.1126/science.1249721.
- Clark, M. Unlocking bread wheat genomes diversity with new sequencing and assembly approaches, 1 2016. URL <https://pag.confex.com/pag/xxiv/webprogram/Paper20198.html>.
- Codd, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, jun 1970. ISSN 00010782. doi: 10.1145/362384.362685.
- Cornish-Bowden, A. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic acids research*, 13(9):3021–30, May 1985. ISSN 0305-1048.
- Curry, J., Kietzmann, S. S., Markus, and Kirby, S. Magic box – automated primer assay design for haploid, diploid and polyploid species, 1 2016. URL <https://pag.confex.com/pag/xxiv/webprogram/Paper20323.html>.
- Davey, R., Etuk, A., Shaw, F., et al. Copo: Aiding the journey from data to publication in the plant sciences, 2015. URL <https://doi.org/10.7490/f1000research.1110111.1>.
- Eddy, S. R. What is a hidden Markov model? *Nature biotechnology*, 22(10):1315–6, October 2004. ISSN 1087-0156. doi: 10.1038/nbt1004-1315.

- Etherington, G. J., Ramirez-Gonzalez, R. H., and MacLean, D. bio-samtools 2: a package for analysis and visualization of sequence and alignment data with SAMtools in Ruby. *Bioinformatics*, pages 1–2, 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv178.
- Flicek, P., Amode, M. R., Barrell, D., et al. Ensembl 2012. *Nucleic acids research*, 40(Database issue):D84–90, January 2012. ISSN 1362-4962. doi: 10.1093/nar/gkr991.
- Fox, S. E., Geniza, M., Hanumappa, M., et al. De Novo Transcriptome Assembly and Analyses of Gene Expression during Photomorphogenesis in Diploid Wheat Triticum monococcum. 9(5), 2014. doi: 10.1371/journal.pone.0096855.
- Gerechter-Amitai, Z. K., van Silfhout, C. H., Grama, A., and Kleitman, F. Yr15 — a new gene for resistance to puccinia striiformis in triticum dicoccoides sel. g-25. *Euphytica*, 43(1):187–190, 1989. ISSN 1573-5060. doi: 10.1007/BF00037912.
- Gillies, S. A., Futardo, A., and Henry, R. J. Gene expression in the developing aleurone and starchy endosperm of wheat. *Plant biotechnology journal*, 10(6):668–79, aug 2012. ISSN 1467-7652. doi: 10.1111/j.1467-7652.2012.00705.x.
- GM, C. Chloroplasts and Other Plastids. In *The Cell: A Molecular Approach*. Sinauer Associates, 2000. URL http://www.ncbi.nlm.nih.gov/books/NBK9905/?redirect-on-error=__HOME__.
- Goto, N., Prins, P., Nakao, M., et al. BioRuby: bioinformatics software for the Ruby programming language. *Bioinformatics (Oxford, England)*, 26(20):2617–9, October 2010. ISSN 1367-4811. doi: 10.1093/bioinformatics/btq475.
- Haerder, T. and Reuter, A. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317, dec 1983. ISSN 03600300. doi: 10.1145/289.291.
- Henry, I. M., Nagalakshmi, U., Lieberman, M. C., et al. Efficient Genome-Wide Detection and Cataloging of EMS-Induced Mutations Using Exome Capture and Next-Generation Sequencing. *The Plant*

- cell*, 26(4):1382–1397, April 2014. ISSN 1532-298X. doi: 10.1105/tpc.113.121590.
- Higgins, D. G. and Sharp, P. M. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–244, dec 1988. ISSN 03781119. doi: 10.1016/0378-1119(88)90330-7.
- Hodges, E., Xuan, Z., Balija, V., et al. Genome-wide in situ exon capture for selective resequencing. *Nat Genet*, 39(12):1522–1527, Dec 2007. ISSN 1061-4036. doi: 10.1038/ng.2007.42.
- Huang, X.-Q. and Brûlé-Babel, A. Development of genome-specific primers for homoeologous genes in allopolyploid species: the waxy and starch synthase ii genes in allohexaploid wheat (*triticum aestivum l.*) as examples. *BMC Research Notes*, 3(1):1–11, 2010. ISSN 1756-0500. doi: 10.1186/1756-0500-3-140.
- Hubbard, A., Lewis, C. M., Yoshida, K., et al. Field pathogenomics reveals the emergence of a diverse wheat yellow rust population. *Genome Biology*, 16(1):1–15, 2015. ISSN 1465-6906. doi: 10.1186/s13059-015-0590-8.
- Hutchison, C. a. DNA sequencing: bench to bedside and beyond. *Nucleic acids research*, 35(18):6227–37, January 2007. ISSN 1362-4962. doi: 10.1093/nar/gkm688.
- Illumina. *CASAVA v1.8.2 User guide*. Revc edition, 2011. URL http://support.illumina.com/sequencing/sequencing_software/casava/documentation.ilmn.
- Inc, I. *TruSeq RNA Sample Preparation v2 Guide*. Illumina Inc.
- James, G. V., Patel, V., Nordström, K. J., et al. User guide for mapping-by-sequencing in Arabidopsis. *Genome biology*, 14(6):R61, June 2013. ISSN 1465-6914. doi: 10.1186/gb-2013-14-6-r61.
- Jupe, F., Witek, K., Verweij, W., et al. Resistance gene enrichment sequencing (RenSeq) enables reannotation of the NB-LRR gene family from sequenced plant genomes and rapid mapping of resistance loci in segregating populations. *The Plant journal : for cell and molecular biology*, 76(3):530–544, November 2013. ISSN 1365-313X. doi: 10.1111/tpj.12307.

- Katoh, K. and Standley, D. M. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–80, April 2013. ISSN 1537-1719. doi: 10.1093/molbev/mst010.
- Kent, W. J. BLAT—The BLAST-Like Alignment Tool. *Genome Research*, 12(4):656–664, March 2002. ISSN 1088-9051. doi: 10.1101/gr.229202.
- Kersey, P. J., Staines, D. M., Lawson, D., et al. Ensembl Genomes: an integrative resource for genome-scale data from non-vertebrate species. *Nucleic acids research*, 40(Database issue):D91–7, January 2012. ISSN 1362-4962. doi: 10.1093/nar/gkr895.
- King, R., Bird, N., Ramirez-Gonzalez, R., et al. Mutation scanning in wheat by exon capture and next-generation sequencing. *PLoS ONE*, 10(9):1–18, 09 2015. doi: 10.1371/journal.pone.0137549.
- Kolesnikov, N., Hastings, E., Keays, M., et al. ArrayExpress update — simplifying data submissions. 43(October 2014):1113–1116, 2015. doi: 10.1093/nar/gku1057.
- Krasileva, K., Vasquez-Gross, H., Howell1, T., et al. Uncovering hidden variation in young polyploid wheat genomes. submitted 2016.
- Krasileva, K. V., Buffalo, V., Bailey, P., et al. Separating homeologs by phasing in the tetraploid wheat transcriptome. *Genome biology*, 14(6):R66, June 2013. ISSN 1465-6914. doi: 10.1186/gb-2013-14-6-r66.
- Krasner, G. E. and Pope, S. T. A Cookbook for Using the Model- View- Controller User Interface Paradigm in Smalltalk-80. *Joop Journal Of Object Oriented Programming*, 1:26–49, 1988. ISSN 0896-8438. doi: 10.1.1.47.366.
- Kugler, K. G., Siegwart, G., Nussbaumer, T., et al. Quantitative trait loci-dependent analysis of a gene co-expression network associated with Fusarium head blight resistance in bread wheat (*Triticum aestivum* L.). *BMC Genomics*, 14(1):728, 2013. ISSN 1471-2164. doi: 10.1186/1471-2164-14-728.
- Lab, C. RNA-seqlopedia, 2015. URL <http://rnaseq.uoregon.edu/>.

- Leach, L. J., Belfield, E. J., Jiang, C., et al. Patterns of homoeologous gene expression shown by RNA sequencing in hexaploid bread wheat. *BMC genomics*, 15(1):276, 2014. ISSN 1471-2164. doi: 10.1186/1471-2164-15-276.
- Leggett, R. M., Ramirez-Gonzalez, R. H., Verweij, W., et al. Identifying and Classifying Trait Linked Polymorphisms in Non-Reference Species by Walking Coloured de Bruijn Graphs. *PLoS ONE*, 8(3):e60058, March 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0060058.
- LGC Genomics. KASP Genotyping, 2014. URL <http://www.lgcgroup.com/LGCGroup/media/PDFs/Products/Genotyping/kasp-explanation-fact-sheet.pdf>.
- Li, A., Liu, D., Wu, J., et al. mRNA and Small RNA Transcriptomes Reveal Insights into Dynamic Homoeolog Regulation of Allopolyploid Heterosis in Nascent Hexaploid Wheat. *The Plant Cell*, 26(5):1878–1900, may 2014. ISSN 1040-4651. doi: 10.1105/tpc.114.124388.
- Li, H. and Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14):1754–60, July 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp324.
- Li, H. and Durbin, R. Fast and accurate long-read alignment with burrows-wheeler transform. *Bioinformatics*, 26(5):589–595, jan 2010. doi: 10.1093/bioinformatics/btp698.
- Li, H., Handsaker, B., Wysoker, A., et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16):2078–9, August 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp352.
- Li, H.-Z., Gao, X., Li, X.-Y., et al. Evaluation of Assembly Strategies Using RNA-Seq Data Associated with Grain Development of Wheat (*Triticum aestivum* L.). *PLoS ONE*, 8(12):e83530, dec 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0083530.
- Ling, H.-Q., Zhao, S., Liu, D., et al. Draft genome of the wheat A-genome progenitor *Triticum urartu*. *Nature*, pages 3–6, March 2013. ISSN 0028-0836. doi: 10.1038/nature11997.

- Liu, Y. and Schmidt, B. Long read alignment based on maximal exact match seeds. *Bioinformatics (Oxford, England)*, 28(18):i318–i324, September 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts414.
- Liu, Z., Xin, M., Qin, J., et al. Temporal transcriptome profiling reveals expression partitioning of homeologous genes contributing to heat and drought acclimation in wheat (*Triticum aestivum* L.). *BMC Plant Biology*, 15(1):152, dec 2015. ISSN 1471-2229. doi: 10.1186/s12870-015-0511-8.
- Ma, J., Stiller, J., Zheng, Z., et al. A high-throughput pipeline for detecting locus-specific polymorphism in hexaploid wheat (*triticum aestivum* L.). *Plant Methods*, 11(1), aug 2015. doi: 10.1186/s13007-015-0082-6.
- Marcussen, T., Sandve, S. R., Heier, L., et al. Ancient hybridizations among the ancestral genomes of bread wheat. *Science*, 345(6194):1250092–1250092, jul 2014. ISSN 0036-8075. doi: 10.1126/science.1250092.
- MAS Wheat. Mas wheat transcriptome. supplemental file 17., 2013. URL <http://maswheat.ucdavis.edu/Transcriptome/index.htm>.
- Mayer, K. F. X., Rogers, J., Dole el, J., et al. A chromosome-based draft sequence of the hexaploid bread wheat (*Triticum aestivum*) genome. *Science*, 345(6194):1251788–1251788, July 2014. ISSN 0036-8075. doi: 10.1126/science.1251788.
- Mayer, K. F. X., Martis, M., Hedley, P. E., et al. Unlocking the barley genome by chromosomal and comparative genomics. *The Plant cell*, 23(4):1249–63, April 2011. ISSN 1532-298X. doi: 10.1105/tpc.110.082537.
- McIntosh, R., Wellings, C.R., and Park, R.F. *Wheat Rusts: an Atlas of Resistance Genes*. CSIRO Publishing, East Melbourne, Victoria, Australia, 1995.
- Metzker, M. L. Sequencing technologies - the next generation. *Nature reviews. Genetics*, 11(1):31–46, January 2010. ISSN 1471-0064. doi: 10.1038/nrg2626.
- Michelmore, R. W., Paran, I., and Kesseli, R. V. Identification of markers linked to disease-resistance genes by bulked segregant analysis: a

- rapid method to detect markers in specific genomic regions by using segregating populations. *Proceedings of the National Academy of Sciences*, 88(21):9828–9832, November 1991. ISSN 0027-8424. doi: 10.1073/pnas.88.21.9828.
- Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature methods*, 5(7):621–8, July 2008. ISSN 1548-7105. doi: 10.1038/nmeth.1226.
- Murphy, L. R., Santra, D., Kidwell, K., et al. Linkage Maps of Wheat Stripe Rust Resistance Genes and for Use in Marker-Assisted Selection. *Crop Science*, 49(5):1786, 2009. ISSN 1435-0653. doi: 10.2135/cropsci2008.10.0621.
- Myatt, G. J. and Johnson, W. P. *Making Sense of Data III: A Practical Guide to Designing Interactive Data Visualizations*. Wiley, 2011. ISBN 0470536497. URL <http://www.amazon.com/Making-Sense-Data-III-Visualizations/dp/0470536497%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0470536497>.
- Myllykangas, S., Buenrostro, J., and Ji, H. P. *Bioinformatics for High Throughput Sequencing*. Springer New York, New York, NY, 2012. ISBN 978-1-4614-0781-2. doi: 10.1007/978-1-4614-0782-9. URL <http://www.springerlink.com/index/10.1007/978-1-4614-0782-9>.
- Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970. ISSN 00222836. doi: 10.1016/0022-2836(70)90057-4.
- Ng, S. B., Turner, E. H., Robertson, P. D., et al. Targeted capture and massively parallel sequencing of 12 human exomes. *Nature*, 461(7261): 272–276, Sep 2009. ISSN 0028-0836. doi: 10.1038/nature08250.
- Oono, Y., Kobayashi, F., Kawahara, Y., et al. Characterisation of the wheat (*Triticum aestivum* L.) transcriptome by de novo assembly for

- the discovery of phosphate starvation-responsive genes: gene expression in Pi-stressed wheat. *BMC genomics*, 14(1):77, 2013. ISSN 1471-2164. doi: 10.1186/1471-2164-14-77.
- Oracle. *MySQL 5.7 Reference Manual*, 2014. URL <http://downloads.mysql.com/docs/refman-5.7-en.a4.pdf>.
- Patro, R., Mount, S. M., and Kingsford, C. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature biotechnology*, 32(5):462–464, 2014. ISSN 1546-1696. doi: 10.1038/nbt.2862.
- Pearce, S., Tabbata, F., Cantu, D., et al. Regulation of Zn and Fe transporters by the GPC1 gene during early wheat monocarpic senescence. pages 1–23, 2014. doi: 10.1186/s12870-014-0368-2.
- Pearce, S., Huttly, A. K., Prosser, I. M., et al. Heterologous expression and transcript analysis of gibberellin biosynthetic genes of grasses reveals novel functionality in the GA3ox family. *BMC plant biology*, 15:130, 2015a. ISSN 1471-2229. doi: 10.1186/s12870-015-0520-7.
- Pearce, S., Vazquez-Gross, H., Herin, S. Y., et al. WheatExp: an RNA-seq expression database for polyploid wheat. *BMC plant biology*, 15(1):299, 2015b. ISSN 1471-2229. doi: 10.1186/s12870-015-0692-1.
- Peng, J., Fahima, T., Röder, M., and Huang, Q. High-density molecular map of chromosome region harboring stripe-rust resistance genes YrH52 and Yr15 derived from wild emmer wheat, *Triticum dicoccoides*. *Genetica*, pages 199–210, 2000.
- Petryszak, R., Keays, M., Tang, Y. A., et al. Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants. *Nucleic Acids Research*, 44(D1):D746–D752, jan 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1045.
- Pfeifer, M., Kugler, K. G., Sandve, S. R., et al. Genome interplay in the grain transcriptome of hexaploid bread wheat. *Science*, 345(6194):1250091–1250091, jul 2014. doi: 10.1126/science.1250091.
- Pickrell, J. K., Marioni, J. C., Pai, A. A., et al. Understanding mechanisms underlying human gene expression variation with rna sequenc-

- ing. *Nature*, 464(7289):768–772, Apr 2010. ISSN 0028-0836. doi: 10.1038/nature08872.
- Pontius, J., Wagner, L., and Schuler, G. UniGene: A Unified View of the Transcriptome. In *The NCBI Handbook [Internet]*, chapter 21. National Center for Biotechnology Information (US), October 2002. URL <http://www.ncbi.nlm.nih.gov/books/NBK21083/>.
- Pozniak, C. J. IWGSC whole genome shotgun sequencing of chinese spring: Towards a reference sequence of wheat, 1 2016. URL <https://pag.confex.com/pag/xxiv/webprogram/Paper22378.html>.
- Rails Guide. Getting started with rails, 2016. URL http://guides.rubyonrails.org/getting_started.html.
- Ramirez-Gonzalez, R. H., Uauy, C., and Caccamo, M. PolyMarker: A fast polyploid primer design pipeline. *Bioinformatics*, pages 2–3, 2015a. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv069.
- Ramirez-Gonzalez, R. H., Bonnal, R., Caccamo, M., and Maclean, D. Bio-samtools: Ruby bindings for SAMtools, a library for accessing BAM files containing high-throughput sequence alignments. *Source code for biology and medicine*, 7(1):6, January 2012. ISSN 1751-0473. doi: 10.1186/1751-0473-7-6.
- Ramirez-Gonzalez, R. H., Segovia, V., Bird, N., Caccamo, M., and Uauy, C. *Next Generation Sequencing Enabled Genetics in Hexaploid Wheat*, pages 201–209. Springer Japan, Tokyo, 2015b. ISBN 978-4-431-55675-6. doi: 10.1007/978-4-431-55675-6_22. URL http://dx.doi.org/10.1007/978-4-431-55675-6_22.
- Ramirez-Gonzalez, R. H., Segovia, V., Bird, N., et al. Rna-seq bulked segregant analysis enables the identification of high-resolution genetic markers for breeding in hexaploid wheat. *Plant Biotechnology Journal*, 13(5):613–624, 2015c. ISSN 1467-7652. doi: 10.1111/pbi.12281.
- Randhawa, H. S., Mutti, J. S., Kidwell, K., et al. Rapid and targeted introgression of genes into popular wheat cultivars using marker-assisted background selection. *PLoS one*, 4(6):e5752, January 2009. ISSN 1932-6203. doi: 10.1371/journal.pone.0005752.

- Rozen, S. and Skaletsky, H. Primer3 on the WWW for general users and for biologist programmers. *Methods in molecular biology* (Clifton, N.J.), 132:365–86, January 2000. ISSN 1064-3745.
- Schneeberger, K. and Weigel, D. Fast-forward genetics enabled by new sequencing technologies. *Trends in Plant Science*, 16(5):282–288, 2011.
- Schneeberger, K., Ossowski, S., Lanz, C., et al. Shoremap: simultaneous mapping and mutation identification by deep sequencing. *Nat Meth*, 6 (8):550–551, Aug 2009. ISSN 1548-7091. doi: 10.1038/nmeth0809-550.
- Sears, E. R. and Miller, T. E. The History of Chinese Spreing. *Cereal Research Communications*, 13(2/3):261–263, 1985.
- Shendure, J. and Ji, H. Next-generation DNA sequencing. *Nature biotechnology*, 26(10):1135–45, October 2008. ISSN 1546-1696. doi: 10.1038/nbt1486.
- Slater, G. S. C. and Birney, E. Automated generation of heuristics for biological sequence comparison. *BMC bioinformatics*, 6:31, January 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-31.
- Smith, T. and Waterman, M. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, March 1981. ISSN 00222836. doi: 10.1016/0022-2836(81)90087-5.
- Stam, P. and Zeven, A. C. The theoretical proportion of the donor genome in near-isogenic lines of self-fertilizers bred by backcrossing. *Euphytica*, 30(2):227–238, 1981. ISSN 00142336. doi: 10.1007/BF00033982.
- Sun, G. L., Fahima, T., Korol, A. B., et al. Identification of molecular markers linked to the Yr15 stripe rust resistance gene of wheat originated in wild emmer wheat, *Triticum dicoccoides*. *TAG Theoretical and Applied Genetics*, 95(4):622–628, September 1997. ISSN 0040-5752. doi: 10.1007/s001220050604.
- Takagi, H., Abe, A., Yoshida, K., et al. QTL-seq: rapid mapping of quantitative trait loci in rice by whole genome resequencing of DNA from two bulked populations. *The Plant journal : for cell and molecular biology*, 74(1):174–83, May 2013a. ISSN 1365-313X. doi: 10.1111/tpj.12105.

- Takagi, H., Uemura, A., Yaegashi, H., et al. MutMap-Gap: whole-genome resequencing of mutant F2 progeny bulk combined with de novo assembly of gap regions identifies the rice blast resistance gene Pii. *The New phytologist*, 200(1):276–83, October 2013b. ISSN 1469-8137. doi: 10.1111/nph.12369.
- Tang, H., Zhang, X., Miao, C., et al. Allmaps: robust scaffold ordering based on multiple maps. *Genome Biology*, 16(1):1–15, 2015. ISSN 1465-6906. doi: 10.1186/s13059-014-0573-1.
- Trapnell, C., Roberts, A., Goff, L., et al. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols*, 7(3):562–78, March 2012. ISSN 1750-2799. doi: 10.1038/nprot.2012.016.
- Trick, M., Adamski, N., Mugford, S. G., et al. Combining SNP discovery from next-generation sequencing data with bulked segregant analysis (BSA) to fine-map genes in polyploid wheat. *BMC plant biology*, 12(1):14, January 2012. ISSN 1471-2229. doi: 10.1186/1471-2229-12-14.
- Van Ooijen, J. and Jansen, J. *Estimation of recombination frequencies; Genetic Mapping in Experimental Populations*, pages 73–133. Cambridge University Press, Cambridge, 2013a. ISBN 978-1-107-0132-16.
- Van Ooijen, J. and Jansen, J. *Introduction; Genetic Mapping in Experimental Populations*, pages 15–31. Cambridge University Press, Cambridge, 2013b. ISBN 978-1-107-0132-16.
- van Ooijen, J. and Voorrips, R. Joinmap: Version 3.0: Software for the calculation of genetic linkage maps, 2002.
- Venturini, L., S, C., Mapleson, D. L., G, K. G., and Swarbreck, D. Integrating multiple transcript assemblies for improved gene structure annotation with mikado, 2016.
- Wagner, G. P., Kin, K., and Lynch, V. J. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory in Biosciences*, 131(4):281–285, 2012. ISSN 14317613. doi: 10.1007/s12064-012-0162-3.

- Wang, S., Wong, D., Forrest, K., et al. Characterization of polyploid wheat genomic diversity using a high-density 90 000 single nucleotide polymorphism array. *Plant biotechnology journal*, 12(6):787–796, March 2014. ISSN 1467-7652. doi: 10.1111/pbi.12183.
- Wang, Y., Tiwari, V. K., Rawat, N., et al. GSP: a web-based platform for designing genome-specific primers in polyploids. *Bioinformatics*, page btw134, mar 2016. doi: 10.1093/bioinformatics/btw134.
- Wang, Z., Gerstein, M., and Snyder, M. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.
- Wellings, C. and McIntosh, R. A. Host-pathogen studies of wheat stripe rust in australia. In Slinkard, A., editor, *Proceedings 9th International Wheat Genetics Symposium*, pages 336–338. University of Saskatchewan, Saskatoon, SK, Canada, 1998.
- Wilkinson, P. a., Winfield, M. O., Barker, G. L. a., et al. CerealsDB 2.0: an integrated resource for plant breeders and scientists. *BMC bioinformatics*, 13(1):219, January 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-219.
- Winfield, M. O., Wilkinson, P. A., Allen, A. M., et al. Targeted re-sequencing of the allohexaploid wheat exome. *Plant Biotechnology Journal*, 10(6):733–742, 2012. ISSN 1467-7652. doi: 10.1111/j.1467-7652.2012.00713.x.
- Winfield, M. O., Allen, A. M., Burridge, A. J., et al. High-density snp genotyping array for hexaploid wheat and its secondary and tertiary gene pool. *Plant Biotechnology Journal*, 14(5):1195–1206, 2016. ISSN 1467-7652. doi: 10.1111/pbi.12485.
- Yachdav, G., Goldberg, T., Wilzbach, S., et al. Anatomy of BioJS, an open source community for the life sciences. *eLife*, 4(JULY 2015):1–7, 2015. ISSN 2050084X. doi: 10.7554/eLife.07009.001.
- Yachdav, G., Wilzbach, S., Rauscher, B., et al. MSAViewer: interactive JavaScript visualization of multiple sequence alignments. *Bioinformatics*, jul 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw474.
- Yang, F., Li, W., and Jørgensen, H. J. L. Transcriptional Reprogramming of Wheat and the Hemibiotrophic Pathogen *Septoria tritici* during Two

Phases of the Compatible Interaction. *PLoS ONE*, 8(11):e81606, nov 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0081606.

Yang, Z., Peng, Z., Wei, S., et al. Pistillody mutant reveals key insights into stamen and pistil development in wheat (*Triticum aestivum L.*). *BMC Genomics*, 16(1):1–11, 2015. ISSN 1471-2164. doi: 10.1186/s12864-015-1453-0.

Zhang, H., Yang, Y., Wang, C., et al. Large-scale transcriptome comparison reveals distinct gene activations in wheat responding to stripe rust and powdery mildew. *BMC Genomics*, 15(1):898, 2014. ISSN 1471-2164. doi: 10.1186/1471-2164-15-898.

Appendix A

Supplemental tables

A.1 PolyMarker supplemental tables.

Table A.1: Validation of mutations on M_4 on Cadenza

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Cadenza)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_3B_scaff_10445294	Cadenza1772	6019	C	T	het	het	caggatAgtGggactgtcaaAG	caggatAgtGggactgtcaaAA	ggagacGGctGtggacatT
IWGSC_CSS_3DL_scaff_6955403	Cadenza1772	2418	C	T	het*	hom	ttagCggattgtcgggatG	ttagCggattgtcggtatA	tgtcCatgaaTcttgtccacG
IWGSC_CSS_4AL_scaff_7106846	Cadenza1772	11277	G	A	hom	hom	tgggatccatgcctacactG	tgggatccatgcctacactA	gatgttGgatttgcgtA
IWGSC_CSS_4AS_scaff_5991335	Cadenza1772	15710	G	A	hom	hom	ctggccctgcgtctcaC	ctggccctgcgtctcaT	gtggaaGttcagaaggaccaG
IWGSC_CSS_4BS_scaff_4956646	Cadenza1772	252	G	A	het*	hom	gcagggttgacttcccgGA	gcagggttgacttcccgGA	tGaggtaGcTaagAAagC
IWGSC_CSS_4DS_scaff_1715962	Cadenza1772	1225	G	A	hom	hom	cagctgtggTatctcaactG	cagctgtggTatctcaactG	CcCtGaaACACcGtttggAT
IWGSC_CSS_5AL_scaff_2763407	Cadenza1772	2119	G	A	hom	hom	gcgacGaacctcgagatctG	gcgacGaacctcgagatctA	gaTggcaAtcgtCgtgcA
IWGSC_CSS_5AS_scaff_1548786	Cadenza1772	12625	C	T	het	het	AtaggcacattgtctgactgA	AtaggcacattgtctgactgA	ggattgggttgtcagcG
IWGSC_CSS_5BL_scaff_10849226	Cadenza1772	2289	C	T	het*	hom	cctgacatcattgttcacgatC	cctgacatcattgttcacgatT	cactccgagggtgtccatgA
IWGSC_CSS_5BS_scaff_2270737	Cadenza1772	2262	G	A	hom	—	attcCTgttgttggCaaatgA	attcCTgttgttggCaaatgA	taaGcacaAAccctccagctG
IWGSC_CSS_1AL_scaff_3022915	Cadenza1661	891	C	T	hom	hom	ccacagtggacttcttggA	ccacagtggacttcttggA	atgtcgttGtcGtgcC
IWGSC_CSS_1AS_scaff_3297240	Cadenza1661	1970	C	T	het	het	cateccccGtttccctC	catcccggatgtttccctC	gctccggatgaaagacG
IWGSC_CSS_1BL_scaff_3828996	Cadenza1661	1340	G	A	hom	hom	acggcgatgttagttAAC	acggcgatgttagttAAC	acgacttggTggcataactC
IWGSC_CSS_1DS_scaff_1884529	Cadenza1661	10575	G	A	hom	hom	aCagatacaAtttcatgcaggC	aCagatacaAtttcatgcaggT	acctgggTTgtccataactC
IWGSC_CSS_2AL_scaff_6318370	Cadenza1661	19142	C	T	het	—	cgtggcCgaatCtcGacG	cgtggcCgaatCtcGacA	ttcttggtggagccggG
IWGSC_CSS_2AS_scaff_5213460	Cadenza1661	1358	G	A	hom	hom	gtcacgaaCcggctcagG	gtcacgaaCcggctcagA	aggaaaagagggaaaaaGcG
IWGSC_CSS_2BS_scaff_5179331	Cadenza1661	5604	G	A	het	het	actctgtcgaaactgtatACA	actctgtcgaaactgtatACA	gcaGagaatgttcttcaACT
IWGSC_CSS_2DS_scaff_5341235	Cadenza1661	4673	G	A	het	het	ggtgaggatctcgaggatG	ggtgaggatctcgaggatA	gcccggatgtacgaggTT
IWGSC_CSS_3AL_scaff_4250995	Cadenza1661	7046	G	A	hom	hom	cCaagaaaacgggtgttccaG	cCaagaaaacgggtgttccaA	ctgcagctgtcccatcatcgT
IWGSC_CSS_3B_scaff_10404421	Cadenza1661	4303	G	A	het	het	ccttcgtcgacCaggacctG	ccttcgtcgacCaggacctA	GCcagactCacAtgcctC
IWGSC_CSS_5DL_scaff_2390496	Cadenza1538	2125	C	T	hom	het	gcagtttatcttcgtatgttG	gcagtttatcttcgtatgttG	ttctgagaaTgtatgtcGatG
IWGSC_CSS_6AL_scaff_5753680	Cadenza1538	3920	C	T	hom	hom	tgctccaaattttggaccaaTaaC	tgctccaaattttggaccaaTaaT	aaatgcaagggttaagttttgT
IWGSC_CSS_6AS_scaff_4425792	Cadenza1538	4307	G	A	hom	het	agatgttgcggCcaG	agatgttgcggCcaA	gctgaagaacacgcgtacaaT
IWGSC_CSS_6BS_scaff_3003630	Cadenza1538	6933	C	T	het	het	ggcgtatgtgtgtcgagC	ggcgtatgtgtgtcgagT	tTgaCttctgggtggcA
IWGSC_CSS_6DL_scaff_3246988	Cadenza1538	9186	G	A	het	het	gctaagaaggatgttggagaaATT	gctaagaaggatgttggagaaATT	aatttctgaagaggttgttatG
IWGSC_CSS_7AL_scaff_4480114	Cadenza1538	3446	C	T	het	—	gatatactccacacggcG	gatatactccacacggcA	tgagccacttgcagttT
IWGSC_CSS_7AS_scaff_4193541	Cadenza1538	8359	C	T	hom	het	agcaatttttgcataattagC	agcaatttttgcataattagT	tcatctGtcttaactctactgtG
IWGSC_CSS_7BL_scaff_6721572	Cadenza1538	9223	C	T	het	het	gctCaggggagaagacaagaaG	gctCaggggagaagacaagaaA	tgctatgaagaattccgacctC
IWGSC_CSS_7BS_scaff_3152545	Cadenza1538	3960	G	A	hom	—	ttagccaaaatcacctgcCgC	ttagccaaaatcacctgcCgT	gCtgcggccatcatcggttaT
IWGSC_CSS_7DS_scaff_3963838	Cadenza1538	2913	G	A	het	het	tCgttgcaggCttTtgtG	tCgttgcaggCttTtgtG	agaGttTAcaagCTactgtcAcA
IWGSC_CSS_1AL_scaff_3903380	Cadenza1469	6193	G	A	hom	hom	ctttcAgagatgaacggG	ctttcAgagatgaacggA	tcGtGagatGtggttGTtA
IWGSC_CSS_1AS_scaff_3287728	Cadenza1469	3817	C	T	het*	hom	ccgaccaAttactaaccG	ccgaccaAttactaaccA	accctttcccAgacatgA
IWGSC_CSS_1BL_scaff_3815304	Cadenza1469	513	G	A	hom	hom	aacatttgccTaCaaaacGC	aacatttgccTaCaaaacGT	acacagcaagtataatgCAAGC
IWGSC_CSS_1DL_scaff_2266648	Cadenza1469	5926	C	T	het	het	caacatgagacacaacacccT	caacatgagacacaacacccT	gtcaacgcgtgaggatttC
IWGSC_CSS_1DS_scaff_1906671	Cadenza1469	3697	C	T	hom	hom	tggTGtagacacttggcGA	tggTGtagacacttggcGA	catggcaccaccAcctG
IWGSC_CSS_2AL_scaff_6337088	Cadenza1469	7334	G	A	het*	hom	acaatgccAaggtgacaggTT	acaatgccAaggtgacaggTT	gggagtgttgggtCagaacaT

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Cadenza)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_2BL_scaff_7972799	Cadenza1469	8995	C	T	het	hom	gTgCtcctcGgcacccTT	gTgCtcctcGgcacccTT	gatccGGcaaaactacTG
IWGSC_CSS_2DL_scaff_9832343	Cadenza1469	3262	G	A	het	het	TtgtctaAcagcacCGcagG	TtgtctaAcagcacCGcagA	agatctcggtcagcttTC
IWGSC_CSS_2DS_scaff_5327939	Cadenza1469	3889	G	A	het	het	tttTgccttatgtgactcttagtaC	ttttTgccttatgtgactcttagtaT	gaggccatcacagatagcG
IWGSC_CSS_3B_scaff_10395219	Cadenza1469	1292	G	A	hom	—	agggtcttgctgtctG	agggtcttgctgtctG	ccttctggggcccttataC
IWGSC_CSS_3B_scaff_10592217	Cadenza0580	2994	C	T	het	—	acacgagtatcaagccctC	acacgagtatcaagccctT	tgatacttttgTggCggAG
IWGSC_CSS_3DS_scaff_2596771	Cadenza0580	1037	G	A	het	het	tggttatgCAcaggataatCagG	tggttatgCAcaggataatCagA	tggcaaATgtgtgtttaggT
IWGSC_CSS_4AL_scaff_7093953	Cadenza0580	9881	C	T	hom	hom	GacaggaaGCCgttaAC	GacaggaaGCCgttaAC	ctccAGcaggcatgggAT
IWGSC_CSS_4BL_scaff_7037448	Cadenza0580	1837	C	T	hom	hom	CgtggaaaaAGtcgaagaacttaAC	CgtggaaaaAGtcgaagaacttaAT	cagttcttcTtCaGagcagataT
IWGSC_CSS_4BS_scaff_4929479	Cadenza0580	10668	G	A	hom	—	tggattttccgcactttC	tggattttccgcactttT	gtaaaacaaggcattcaagatcA
IWGSC_CSS_4DL_scaff_14359838	Cadenza0580	1408	G	A	hom	—	gCtcAttcaggatTGTcTatT	gCtcAttcaggatTGTcTatTA	tgaCagaacagtgttacatC
IWGSC_CSS_4DS_scaff_2276484	Cadenza0580	8034	G	A	hom	hom	gccgtgggtatggAgaG	gccgtgggtatggAgaA	cgtccaggattactgatactgcA
IWGSC_CSS_5AL_scaff_2756579	Cadenza0580	5278	G	A	het	het	tgaatggattttgcgtccgttC	tgaatggatttgcgtccgttT	ggAAtCCTATgCAGAAgAAAATG
IWGSC_CSS_5BL_scaff_10787208	Cadenza0580	10627	G	A	het	—	gcctctcacatgggagaC	gcctctcacatgggagaT	acgatgtcAggtggGcgT
IWGSC_CSS_5BS_scaff_2282179	Cadenza0580	5267	G	A	het	—	tgtatggctacgtgtC	tgtatggctacgtgtC	tcggcccttggaaAtCC
IWGSC_CSS_5DL_scaff_4498073	Cadenza0423	4937	C	T	hom	hom	gcaccctctgttgtcatC	gcaccctctgttgtcatT	tgagcagaAAGcagccG
IWGSC_CSS_5DS_scaff_2738970	Cadenza0423	2319	C	T	het	—	cgtgagggtgggtatggC	cgtgagggtgggtatggT	tggaaactgttacactgcagtTC
IWGSC_CSS_6AL_scaff_5757109	Cadenza0423	2788	G	A	hom	hom	caggaGcctggcaaataaaAG	caggaGcctggcaaataaaGA	cttcGcgtctttagttcG
IWGSC_CSS_6AS_scaff_4387871	Cadenza0423	2543	G	A	hom	hom	gcatgtaaacaggcggaaaAG	gcatgtaaacaggcggaaaAG	ctcatgtcttgcattaaagtT
IWGSC_CSS_6BL_scaff_4271391	Cadenza0423	4660	C	T	hom	hom	tacgtcatgttgtgttgtcaC	tacgtcatgttgtgttgtcaT	gtttaaagtgtcatgttgtgttgtcaA
IWGSC_CSS_6DS_scaff_1880206	Cadenza0423	9159	G	A	het	het	ctgCgaaggctccacaaG	ctgCgaaggctccacaaA	ggatgagaagtgtgcattgtcC
IWGSC_CSS_7AS_scaff_4227506	Cadenza0423	952	G	A	het	—	ccatgtttccaatgttagagC	ccatgtttccaatgttagagT	tgcctactgtgtatgcT
IWGSC_CSS_7BL_scaff_6681782	Cadenza0423	1486	C	T	hom	hom	agtaagCGtgacagacaatggG	agtaagCGtgacagacaatggA	AtgtctTtgGtggaaagtacatCA
IWGSC_CSS_7BS_scaff_3160328	Cadenza0423	7801	C	T	het	het	tgttaaatGatacagCtcgcgC	tgttaaatGatacagCtcgcgT	tggaaatgggCgttggTTT
IWGSC_CSS_7DS_scaff_407428	Cadenza0423	2051	G	A	het	het	gtcGCgcacatctgcacaG	gtcGCgcacatctgcacaA	actcatAGgtcagccccA
IWGSC_CSS_3AL_scaff_442479	Cadenza0364	3198	C	T	het	het	gagtcTTaagtgttaagattggC	gagtcTTaagtgttaagattggT	GCaGaTaaCACaggatcacG
IWGSC_CSS_3AL_scaff_4447942	Cadenza0364	11917	G	A	het	het	gtcataaaaggattgtctgtgaaG	gtcataaaaggattgtctgtgaaA	ctcGgatgtgggaggaaAG
IWGSC_CSS_3AS_scaff_1557483	Cadenza0364	2547	C	T	het	het	aaagtccatcatgttaccatcaaG	aaagtccatcatgttaccatcaaA	cgaaaatccaaacgcctcatA
IWGSC_CSS_3AS_scaff_2648747	Cadenza0364	2688	G	A	het	het	tggAaggcAcaaggggccC	tggAaggcAcaaggggccT	GccgcgtatggagactcG
IWGSC_CSS_3AS_scaff_3304956	Cadenza0364	1017	G	A	het	het	gtccccctgcacacagcttG	gtccccctgcacacagcttA	cctgctggactacaacttcaA
IWGSC_CSS_3AS_scaff_3321091	Cadenza0364	4585	C	T	het	het	caagaatGATgtctgttgtGA	caagaatGATgtctgttgtGA	acatgctgaatcgccgaatC
IWGSC_CSS_3AS_scaff_3371333	Cadenza0364	538	G	A	het	het	ggggaaaCgAGcaggcG	ggggaaaCgAGcaggcG	ccgtgccttcacccT
IWGSC_CSS_3AS_scaff_3371815	Cadenza0364	1061	C	T	het	het	atccccacggcacaagAG	atccccacggcacaagAG	aAtggcccttggattcC
IWGSC_CSS_3AS_scaff_3440912	Cadenza0364	4498	G	A	het	het	ccgtaaaactttctgtgtcttgC	ccgtaaaactttctgtgtcttgT	atActgacaaaactatcatgtgtc
IWGSC_CSS_3B_scaff_10343586	Cadenza0364	2242	G	A	het	—	ggttcTgTcctcttcactG	ggttcTgTcctcttcactA	tgtgttigaaccgcgaagcA
IWGSC_CSS_3AL_scaff_442479	Cadenza0364	3198	C	T	het	het	gagtcTTaagtgttaagattggC	gagtcTTaagtgttaagattggT	GCaGaTaaCACaggatcacG
IWGSC_CSS_3AL_scaff_4447942	Cadenza0364	11917	G	A	het	het	gtcafaaaggattgtctgtgaaG	gtcafaaaggattgtctgtgaaA	ctcGgatgtgggaggaaAG
IWGSC_CSS_3AS_scaff_1557483	Cadenza0364	2547	C	T	het	het	aaagtccatcatgttaccatcaaG	aaagtccatcatgttaccatcaaA	cgaaaatccaaacgcctcatA
IWGSC_CSS_3AS_scaff_2648747	Cadenza0364	2688	G	A	het	het	tggAaggcAcaaggggccC	tggAaggcAcaaggggccT	GccgcgtatggagactcG
IWGSC_CSS_3AS_scaff_3304956	Cadenza0364	1017	G	A	het	het	gtccccctgcacacagcttG	gtccccctgcacacagcttA	cctgctggactacaacttcaA
IWGSC_CSS_3AS_scaff_3321091	Cadenza0364	4585	C	T	het	het	caagaatGATgtctgttgtGA	caagaatGATgtctgttgtGA	acatgctgaatcgccgaatC

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Cadenza)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_3AS_scaff_3371333	Cadenza0364	538	G	A	het	het	gggaaaCgAgAcgagcgG	gggaaaCgAgAcgagcgA	ccgtgccttcacccT
IWGSC_CSS_3AS_scaff_3371815	Cadenza0364	1061	C	T	het	het	atccccacggcacagagG	atccccacggcacagagA	aAtggcccttggtgattcC
IWGSC_CSS_3AS_scaff_3440912	Cadenza0364	4498	G	A	het	het	ccgtaaaactttctgtgcctgC	ccgtaaaactttctgtgcctgT	atActgacaaaactacatgtatgc
IWGSC_CSS_3B_scaff_10343586	Cadenza0364	2242	G	A	het	—	ggttcTgTcctcttccactG	ggttcTgTcctcttccactA	tgtgttgaaccgcaga
IWGSC_CSS_5DL_scaff_242342	Cadenza0281	2433	C	T	hom	hom	catggCgacggtGtcctG	catggCgacggtGtcctA	aAccctatTTtggCTACTtCT
IWGSC_CSS_5DL_scaff_4538822	Cadenza0281	1208	G	A	hom	—	acgtcagaacaacccgttgaC	acgtcagaacaacccgttgaT	ttaaatgggtggccac
IWGSC_CSS_6AL_scaff_5813297	Cadenza0281	4532	C	T	hom	—	ggggagggggacgtctcgG	ggggagggggacgtctcgA	ttctctgccaacgattccG
IWGSC_CSS_6AS_scaff_4378990	Cadenza0281	6748	C	T	hom	hom	cccaggttctgtcttttC	cccaggttctgtcttttC	caagtatacggaaatgaaggTgT
IWGSC_CSS_6BL_scaff_4360781	Cadenza0281	5426	C	T	het	het	aCtactcaaattggcttGgtgtAA	aCtactcaaattggcttGgtgtAA	tcaagtccaaatgtCaaagatT
IWGSC_CSS_7AL_scaff_4488310	Cadenza0281	3808	G	A	hom	hom	gttctctttagtagcagccG	gttctctttagtagcagccA	ggcgctttctggcactA
IWGSC_CSS_7BL_scaff_6696509	Cadenza0281	9232	G	A	het	het	gctctaggGgtggcaaAagG	gctctaggGgtggcaaAagA	ggcttGAGtcGcagtG
IWGSC_CSS_7BS_scaff_3143575	Cadenza0281	1866	C	T	het	het	agatgttggagggccgttC	agatgttggagggccgttT	gttggAttgtggcaagtT
IWGSC_CSS_7DL_scaff_3346250	Cadenza0281	1663	G	A	het	het	acgtcagcaacatcttaAC	acgtcagcaacatcttaA	Tttccaccaggccaa
IWGSC_CSS_7DS_scaff_3933917	Cadenza0281	1243	C	T	het	het	tgCtgagCttTcacttG	tgCtgagCttTcacttG	agagttttttccatGG
IWGSC_CSS_3B_scaff_10626860	Cadenza0148	7847	G	A	het	het	gcagctctgggaggagG	gcagctctgggaggagA	gttaatgtacCTcctagctcG
IWGSC_CSS_3DL_scaff_6915683	Cadenza0148	6904	C	T	het	het	cgtcaaCctgtggcaattG	cgtcaaCctgtggcaattA	tcatgctataatgtCataaggT
IWGSC_CSS_4AS_scaff_5929057	Cadenza0148	4238	G	A	hom	hom	gcgcaacgttagCacctacC	gcgcaacgttagCacctacT	ttatctgtgaagtgcacaggTCA
IWGSC_CSS_4AS_scaff_5950625	Cadenza0148	10590	C	T	het	het	agaTattCaaaTcggtggAttggC	agaTattCaaaTcggtggAttggT	cctgCtcccccacgtcC
IWGSC_CSS_4AS_scaff_5967119	Cadenza0148	11626	C	T	hom	hom	cgtGacaccccggactG	cgtGacaccccggactA	gacgacgacactgcacgaC
IWGSC_CSS_4DL_scaff_14455742	Cadenza0148	1946	C	T	hom	hom	gCctgaggagatcgcG	gCctgaggagatcgcT	aaccgGtAACTGtGgGcA
IWGSC_CSS_4DS_scaff_2318993	Cadenza0148	4000	C	T	hom	hom	tccaggttgacacaggatgtatggG	tccaggttgacacaggatgtatggA	tgagaTtcgtttccatcAttG
IWGSC_CSS_5AL_scaff_2750707	Cadenza0148	4603	G	A	het	het	ccttggtctgtagccatttcaagTaG	ccttggtctgtagccatttcaagTaA	ccaggaTgcAgtcaatattcaaG
IWGSC_CSS_5BL_scaff_10794137	Cadenza0148	9235	C	T	hom	hom	gaagctgttctgcgttG	gaagctgttctgcgttA	agtatcccccataatagcgtG
IWGSC_CSS_5BS_scaff_1646558	Cadenza0148	2916	C	T	het	het	gccGtacactcacatCccttG	gccGtacactcacatCccttA	gcaaTgtccacttAtcatccT
IWGSC_CSS_1AL_scaff_3883106	Cadenza0110	27536	C	T	het	het	accttccatactgtgcgtG	accttccatactgtgcgtA	gtagaagacaacatggtaagC
IWGSC_CSS_1BL_scaff_3812829	Cadenza0110	10770	G	A	het*	hom	cccccaactccatcccgG	cccccaactccatcccgA	gGatgttgcgttgcgttgcA
IWGSC_CSS_1DL_scaff_2266648	Cadenza0110	6156	G	A	het	het	actgcgtgttatgtggacC	actgcgtgttatgtggacT	ccccatcaactgcacacaAC
IWGSC_CSS_1DS_scaff_1889435	Cadenza0110	8826	C	T	hom	hom	aaccatgaattactcgacagG	aaccatgaattactcgacagA	gccctgtgaagaattgtatcaaacaG
IWGSC_CSS_2AS_scaff_5268634	Cadenza0110	4636	G	A	het	het	gatccatgtgattggcatgtttG	gatccatgtgattggcatgtttA	TgctgtTggatatgcgttacT
IWGSC_CSS_2BL_scaff_7965110	Cadenza0110	15801	C	T	hom	hom	cattgaagcAtacacAattgcAtaC	cattgaagcAtacacAattgcAtaT	gccagagtatccagataaggTttA
IWGSC_CSS_2DL_scaff_9852812	Cadenza0110	13788	G	A	hom	hom	atttttgtatgttcaatcttcgC	atttttgtatgttcaatcttcgT	gaacgtTcattctgtacttgcT
IWGSC_CSS_2DS_scaff_5371379	Cadenza0110	2166	C	T	hom	hom	agacacaaaactagtGatgcG	agacacaaaactagtGatgcG	gctgctgagaatgttTgttatttG
IWGSC_CSS_3AL_scaff_4384278	Cadenza0110	1276	C	T	het	het	agcTgaactccccTgtA	agcTgaactccccTgtA	agggacctCgGtggatgaA
IWGSC_CSS_3AS_scaff_3340122	Cadenza0110	1467	C	T	hom	hom	attctAgtgttgcggacatG	attctAgtgttgcggacatA	gagaagactagaaatgttgcA
IWGSC_CSS_5DL_scaff_4554222	Cadenza2103	6528	C	T	het*	hom	gctgcctacaaaagaaaaacaaattG	gctgcctacaaaagaaaaacaaattA	aTccaaactatCGaTtttgtcataC
IWGSC_CSS_6AL_scaff_5833640	Cadenza2103	7346	C	T	hom	hom	aagaaaaaggccacaaatgtttctC	aagaaaaaggccacaaatgtttctT	aCTctgTcagtgttcccaG
IWGSC_CSS_6AS_scaff_4429974	Cadenza2103	3867	G	A	hom	hom	GagatgaAtttattgagcatgtggC	GagatgaAtttattgagcatgtggT	ggttccggcgtcataagT
IWGSC_CSS_6DL_scaff_3307626	Cadenza2103	4970	C	T	hom	hom	tgcagatgttgcgttgcgttA	tgcagatgttgcgttgcgttA	ctaggaagggtgatTTtgcgttG
IWGSC_CSS_6DS_scaff_2059604	Cadenza2103	5224	G	A	het	—	gctcaatgcatgcTgagtgG	gctcaatgcatgcTgagtgA	tgtcaatgttattttctgcgttG
IWGSC_CSS_7AL_scaff_4552322	Cadenza2103	1412	C	T	het	het	gcaaaaggcTgatactccaacaG	gcaaaaggcTgatactccaacaA	ggcAAGccAgtataaaaagtaaGC

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Cadenza)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_7BS_scaff_3147455	Cadenza2103	4607	G	A	het	—	gcaccttaggatgttagTtatgC	gcaccttaggatgttagTtatgT	gcatgttaggttatttgactgtTA
IWGSC_CSS_7DL_scaff_3382467	Cadenza2103	3473	C	T	hom	—	GGTtcgCaGTTCATAAActcatC	GGTtcgCaGTTCATAAActcatT	attgaatacaactgatacGaaGactC
IWGSC_CSS_3B_scaff_10457010	Cadenza0277	10599	G	A	het	het	aacctggcccgacaacaC	aacctggcccgacaacaT	actggctgcacgagaggG
IWGSC_CSS_3B_scaff_10593852	Cadenza0277	10124	C	T	het	het	tgacaggggacgtatacaG	tgacaggggacgtatacaA	gtctcaaCTtACAttAccatcagC
IWGSC_CSS_3DS_scaff_2583390	Cadenza0277	663	G	A	hom	hom	actgcactatacatActtCtgC	actgcactatacatActtCtgT	tcCacctggacagcaagtG
IWGSC_CSS_4AL_scaff_7093953	Cadenza0277	10004	C	T	hom	hom	ccttgtatccaatggATtgTtttgG	ccttgtatccaatggATtgTtttgA	tcccacaataaaaggaaagagC
IWGSC_CSS_4AL_scaff_7176064	Cadenza0277	6220	C	T	het	het	gtgcgtgtTCgcctG	gtgcgtgtTCgcctG	atgttcggggatgggG
IWGSC_CSS_4DL_scaff_14122349	Cadenza0277	1010	C	T	hom	hom	gtgcgtgtCttgtGA	gtgcgtgtCttgtGA	ggaacaggccaaaggagG
IWGSC_CSS_5AL_scaff_2736916	Cadenza0277	4296	G	A	het	het	aagaactATgAaaGtaacacacgaC	aagaactATgAaaGtaacacacgaT	ttcGTTtTaAgGcAttCtcG
IWGSC_CSS_5BL_scaff_10883744	Cadenza0277	2080	C	T	hom	hom	gcctcttTgttTagectcaG	gcctcttTgttTagectcaA	cgacaagggtcgatTgcA
IWGSC_CSS_1AL_scaff_3932013	Cadenza0548	11765	C	T	hom	hom	accgccaaCccaagacaG	accgceaaCccaagacaA	ccattaaGccgTgcAacG
IWGSC_CSS_1BS_scaff_3417505	Cadenza0548	373	C	T	het	het	gtgtgaggAGgtgGaG	gtgtgaggAGgtgGaA	tggtcgCcaggatgttgA
IWGSC_CSS_2AS_scaff_5305619	Cadenza0548	2786	C	T	hom	hom	atacagatgcctAAgtggTtC	atacagatgcctAAgtggTtT	ggaagacaAtGtcaggtaC
IWGSC_CSS_2AS_scaff_5306489	Cadenza0548	46953	T	G	het	wt	aggttgcattgtccatagaaGT	aggttgcattgtccatagaaGT	aggctaTAactctgtACAgT
IWGSC_CSS_2BL_scaff_7984123	Cadenza0548	11660	G	A	het	het	catttgtggcatagtaatcgtacaG	catttgtggcatagtaatcgtacaA	aatacattgaggaaatcaaagccC
IWGSC_CSS_2DL_scaff_9907477	Cadenza0548	1363	C	T	hom	hom	tgccctcccttgcagaaC	tgccctcccttgcagaaT	ggcaaaacctgttgccatC
IWGSC_CSS_2DS_scaff_5330886	Cadenza0548	5449	G	A	hom	hom	gcatgtccatttatactgaaCgtG	gcatgtccatttatactgaaCgtA	catgtcttcttcgtgac
IWGSC_CSS_3AL_scaff_4449951	Cadenza0548	633	C	T	het	het	tccaaacctaacagtctaactaG	tccaaacctaacagtctaactaA	gtctgcagTGCaatgtgC
IWGSC_CSS_3B_scaff_10479889	Cadenza0097	3339	C	T	hom	—	ttgTttctGgagaagatgcCG	ttgTttctGgagaagatgcA	ggtgcattcaAcGgcA
IWGSC_CSS_3B_scaff_10562262	Cadenza0097	7819	C	T	het	het	agaggggtgtctatccatAttgG	agaggggtgtctatccatAttgA	aggcgttgcacaggcttc
IWGSC_CSS_4AL_scaff_7040796	Cadenza0097	10772	G	A	hom	hom	acacaacattgcacaggAG	acacaacattgcacaggA	CAatCgtattgttcTtc
IWGSC_CSS_4AL_scaff_7063488	Cadenza0097	6360	C	T	het	het	gcctctcacCttAatttgaagctgC	gcctctcacCttAatttgaagctgT	aggcagtgggatgtgaaggTT
IWGSC_CSS_4AL_scaff_7091701	Cadenza0097	5050	G	A	het	het	catgagcatctggggagaaatG	catgagcatctggggagaaatA	agcaaggaaAtaatgaacggaaA
IWGSC_CSS_4DS_scaff_1845841	Cadenza0097	7110	G	A	hom	hom	aatgTAgctccccatCgG	aatgTAgctccccatCgA	actgaaacTgcatacgTtatggA
IWGSC_CSS_5AL_scaff_2767581	Cadenza0097	3737	G	A	het	het	gagagggtctcaactAtcgC	gagagggtctcaactAtcgT	cgTcatcacaatattgtcgG
IWGSC_CSS_5BL_scaff_10784643	Cadenza0097	1568	C	T	hom	hom	agaaaTAcatggatggatggA	agaaaTAcatggatggatggA	catctCCttccaCgGaaaG
IWGSC_CSS_1AL_scaff_3952258	Cadenza2092	8107	C	T	het	—	tgtagagaaaaattgcacgttgG	tgtagagaaaaattgcacgttgA	tgccacattgcacatfagagaG
IWGSC_CSS_1BL_scaff_3858008	Cadenza2092	10278	G	A	hom	hom	tttgagcaggcaggatcgC	tttgagcaggcaggatcgT	actcacggcttatatCActattC
IWGSC_CSS_1DL_scaff_2265172	Cadenza2092	9094	C	T	hom	hom	tgcaTGTcattttgtttatcgC	tgcaTGTcattttgtttatcgT	agtgtccaaatccGttcatC
IWGSC_CSS_2AL_scaff_6435867	Cadenza2092	16201	G	A	hom	hom	tttctgTacccttaacgtcaattgaC	tttctgTacccttaacgtcaattgaT	gtgaggatgtggatggaaacC
IWGSC_CSS_2AL_scaff_6439430	Cadenza2092	25101	C	T	het	—	caagaaaggCagCtCagC	caagaaaggCagCtCagT	tcGttAcTtttcActggtaA
IWGSC_CSS_2DL_scaff_9760848	Cadenza2092	4733	C	T	het	het	gcaccatgggtctcggtA	gcaccatgggtctcggtA	tcagtcatGCTCtgTCTG
IWGSC_CSS_3AL_scaff_4407012	Cadenza2092	2785	C	T	hom	hom	acatatAgtttctcatcaccatC	acatatAgtttctcatcaccatT	acctcttcatgttaataggttgT
IWGSC_CSS_3AS_scaff_3441108	Cadenza2092	541	G	A	het	het	GtgatgaccttgcacGgaG	GtgatgaccttgcacGgaA	aggcaTgacaaCgcgaA
IWGSC_CSS_3B_scaff_10449827	Cadenza1551	4779	G	A	hom	hom	ggcaaggtaagaaacGgtC	ggcaaggtaagaaacGgtT	aCagaGtgggttagaggcaG
IWGSC_CSS_3B_scaff_10550638	Cadenza1551	3250	C	T	het	het	ctccctcacattgttgcggC	ctccctcacattgttgcggT	gcaacAtTttgataactgcaagG
IWGSC_CSS_3DL_scaff_6945816	Cadenza1551	589	C	T	hom	hom	agcatctacccgtcaaCaataC	agcatctacccgtcaaCaataT	TgtgccCTctgaAtatTTcaTG
IWGSC_CSS_3DL_scaff_6954177	Cadenza1551	3508	C	T	het	het	tgttagcatcacatTAacttctG	tgttagcatcacatTAacttctA	gcttggataaacCttacgacA
IWGSC_CSS_4AS_scaff_5938272	Cadenza1551	19080	G	A	hom	hom	agAcCccgAtgcctatG	agAcCccgAtgcctatG	GggAgatAcaggtaaaActcTtcG
IWGSC_CSS_4AS_scaff_5977594	Cadenza1551	11092	C	T	het	het	gccttgattcggaaacaacaaaC	gccttgattcggaaacaacaaaT	gcgtctctcagtcctgcA

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Cadenza)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_5AL_scaff_2671035	Cadenza1551	5859	C	T	het	het	cggtgatattTtttagacttcgacgC	cggtgatattTtttagacttcgacgT	ggcagttcagcGaccatT
IWGSC_CSS_5BL_scaff_10889480	Cadenza1551	2530	G	A	hom	hom	gagcttaactcgccagatggaG	gagcttaactcgccagatggaA	tccatgCAacGccttggT
IWGSC_CSS_3B_scaff_10528396	Cadenza2088	8059	G	A	hom	—	cttttccgtccgtaaagcaataG	cttttccgtccgtaaagcaataA	gtgcactgttcaggcctgA
IWGSC_CSS_3B_scaff_10637573	Cadenza2088	16815	G	A	het	het	agcaagcttaccGgtctgC	agcaagcttaccGgtctgT	cgagcAactacgagcagctT
IWGSC_CSS_4AL_scaff_7086469	Cadenza2088	6697	G	A	het	het	gccgtctacttcaacgcG	gccgtctacttcaacgcA	ccaGaggctgtTGcatTTT
IWGSC_CSS_4AL_scaff_7126302	Cadenza2088	3627	G	A	hom	hom	gttccaaaaacaaggatggctAatttgC	gttccaaaaacaaggatggctAatttgT	cacaaggatatgaagCTttctagA
IWGSC_CSS_4BL_scaff_7041808	Cadenza2088	10234	G	A	hom	hom	tcaatggatgagggtgtttC	tcaatggatgagggtgtttT	ccatagcagcatcagccacA
IWGSC_CSS_5AL_scaff_2794167	Cadenza2088	13162	G	A	het	—	agtattcaggacaaggcatCttCaG	agtattcaggacaaggcatCttCaA	caatgaaacctctcgaaagaGaG
IWGSC_CSS_5BL_scaff_10889232	Cadenza2088	3885	G	A	het	het	cTcaaccacaatggcaAatC	cTcaaccacaatggcaAatT	tccttcataatcatcaatgttgG
IWGSC_CSS_5BS_scaff_2267405	Cadenza2088	11113	C	T	hom	hom	ctttgatgtccatggctctTG	ctttgatgtccatggctctTA	tgatttggTCgtttAgatttGA
IWGSC_CSS_3B_scaff_10475354	Cadenza1409	2203	G	A	hom	hom	agCgaacaagagGtcaaacG	agCgaacaagagGtcaaacA	ctgaaacacaCtagaCAattAccG
IWGSC_CSS_3B_scaff_10674115	Cadenza1409	4555	C	T	het	het	gtttcagtgcgtccatcaG	gtttcagtgcgtccatcaA	cttcacacCCGagataatGtattG
IWGSC_CSS_4AL_scaff_7153568	Cadenza1409	13073	C	T	hom	hom	tccgaccGAtcaaccttG	tccgaccGAtcaaccttG	gaccggaaacttcggcC
IWGSC_CSS_4DL_scaff_14314966	Cadenza1409	2010	G	A	het	hom	gttggccccctctCAGgG	gttggccccctctCAGgA	cggcgTcacaAgttgCcT
IWGSC_CSS_4DS_scaff_2324074	Cadenza1409	7606	G	A	het	het	tGcatgaaaatgttGcaGaG	tGcatgaaaatgttGcaGaA	gggtaAgttcAaaactGaagtgaaG
IWGSC_CSS_5AS_scaff_1517889	Cadenza1409	3561	G	A	het	het	tctcgacatctcccggttaC	tctcgacatctcccggttaT	gtgcctggAACATTGCTTATT
IWGSC_CSS_5AS_scaff_1523866	Cadenza1409	8054	G	A	hom	—	ggtgatctaccgccaGgaC	ggtgatctaccgccaGgaT	tcctgcagCcTetcctcA
IWGSC_CSS_5BL_scaff_10917655	Cadenza1409	19073	G	A	hom	hom	caaatacgatgaaaaaaggatgC	caaatacgatgaaaaaaggatgT	cgcttcatactacaAaatatgtcT
IWGSC_CSS_1AL_scaff_3886649	Cadenza1599	5204	C	T	het	het	tgtatcccaaccacatGcC	tgtatcccaaccacatGcT	ggactgactgcgaccatatttaG
IWGSC_CSS_1BL_scaff_3810267	Cadenza1599	6634	C	T	hom	hom	ccCaggaaatggacaccC	ccCaggaaatggacaccT	cgcaggcgaagatgtgaTtG
IWGSC_CSS_1DL_scaff_2291677	Cadenza1599	12856	C	T	hom	hom	GgttagacaatgcgcgaG	GgttagacaatgcgcgaA	cctctcttcaacGCcG
IWGSC_CSS_2AL_scaff_6354492	Cadenza1599	7566	G	A	het	het	gGagaatgcACAgAAcTtctgG	gGagaatgcACAgAAcTtctgA	ttccgaagaaccacaTccTG
IWGSC_CSS_2AS_scaff_5282937	Cadenza1599	9736	G	A	het	het	gctgttagattttatagctgtatG	gctgttagattttatagctgtatG	cacCagaattttCactgattTC
IWGSC_CSS_2BL_scaff_7952427	Cadenza1599	19249	G	A	hom	hom	cgTccctCcttagcacgaC	cgTccctCcttagcacgaT	aTcactccattagcgcgAG
IWGSC_CSS_2DL_scaff_9897981	Cadenza1599	5627	C	T	het	het	cttggtgtCTgattgttactC	cttggtgtCTgattgttactT	gTttgtCTctctgtatCTtgtG
IWGSC_CSS_3AL_scaff_4446105	Cadenza1599	1765	G	A	hom	—	aaatgtttctaCcgctagtG	aaatgtttctaCcgctagtA	tttAgaggcaatagctTatatgcT

Table A.2: Validation of mutations on M_4 on Kronos

IWGSC contig	Line	Pos	WT	Mut	Predicted	M_4	Primer 1 (Kronos)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_1AS_scaff_3284790	Kronos3085	7449	G	A	Het	Het	ccacacccTggcctcgC	ccacacccTggcctcgT	gtgatttgcacggggAG
IWGSC_CSS_1BL_scaff_3897513	Kronos3085	1515	C	T	Het	Het	gtttccactCGgttcTcgC	gtttccactCGgttcTcgT	acAaggactgttcagaGaC
IWGSC_CSS_2AL_scaff_6434745	Kronos3085	3424	C	T	Het	Het	cctcGgtttgcaatttctatgC	cctcGgtttgcaatttctatgT	gGCaaTggcataacaacagataA
IWGSC_CSS_3AS_scaff_3408995	Kronos3085	732	C	T	Het	Het	aggccatttgcattccgC	aggccatttgcattccgT	ggTgttaTccagAaccTgAG
IWGSC_CSS_3B_scaff_10708748	Kronos3085	2675	G	A	Het	Het	gttgcattgcgttacccacG	gttgcattgcgttacccacA	gtaacaatctgatgttgcgtacC
IWGSC_CSS_4AL_scaff_7132733	Kronos3085	1799	C	T	Hom	Hom	caccctgtggatgcacccT	caccctgtggatgcacccT	aCcGcctaGaaagaagactTC

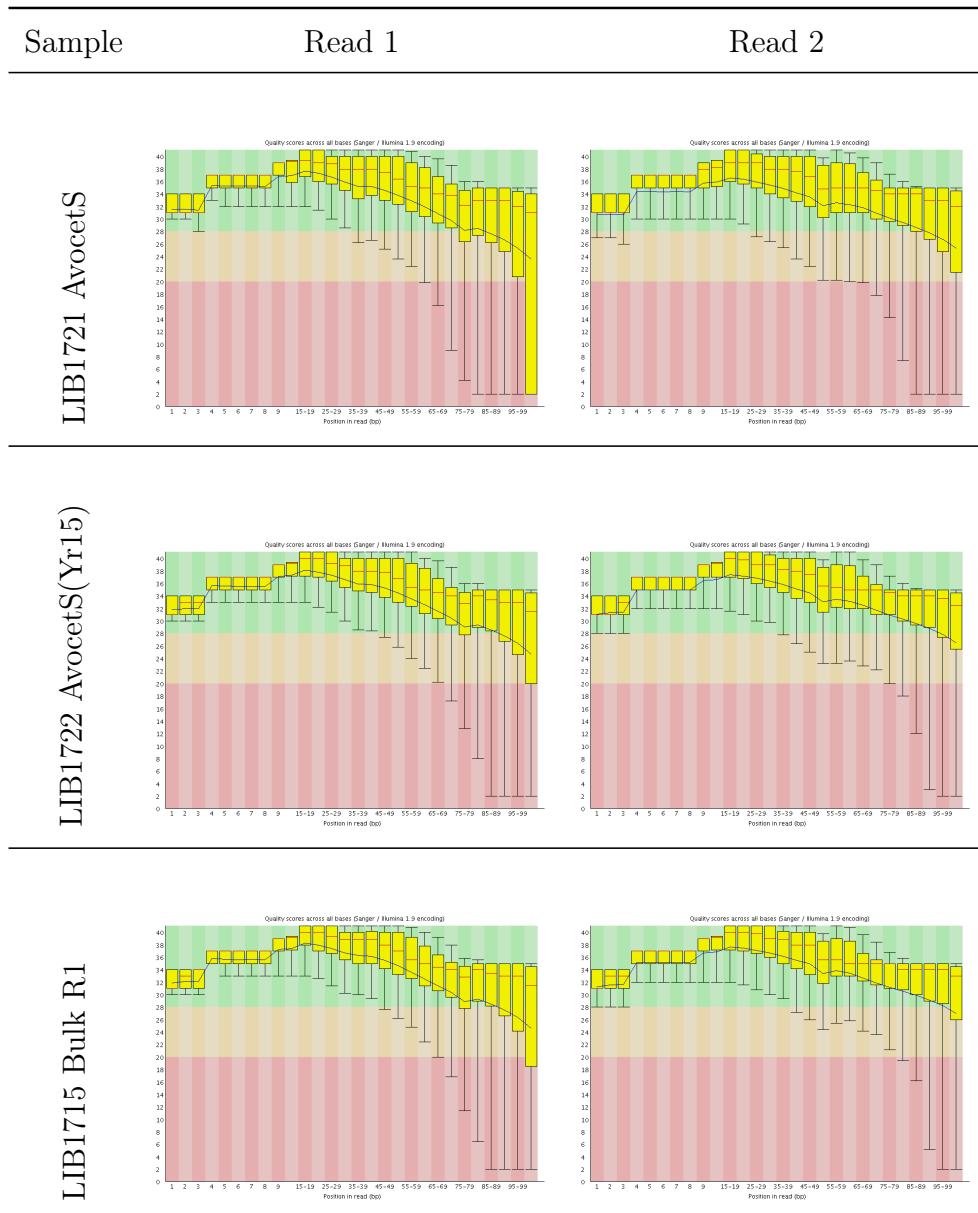
IWGSC contig	Line	Pos	WT	Mut	Predicted	<i>M</i> ₄	Primer 1 (Kronos)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_5AS_scaff_1534693	Kronos3085	4605	C	T	Het	Het	cagttctggccctcAtC	cagttctggccctcAtT	gtACtccacgAgtcaTgAG
IWGSC_CSS_6AS_scaff_4361911	Kronos3085	8857	G	A	Het	Het	tcacgaaagacgacttcaacctcC	tcacgaaagacgacttcaacctcT	catgagggtgcgtcatctccatA
IWGSC_CSS_6BS_scaff_3008326	Kronos3085	1528	G	A	Het	Het	ccatgttgtactgggttgC	ccatgttgtactgggttgT	ggaagcatggCaatgtcA
IWGSC_CSS_7AS_scaff_4214385	Kronos3085	27835	C	T	Hom	Hom	cgtacccctgtggaaG	cgtacccctgtggaaG	cttggcgtcgatgtataagacT
IWGSC_CSS_1AL_scaff_3929964	Kronos3191	1336	C	T	Het	Het	tttcggccataacctgacatC	tttcggccataacctgacatT	attgcctccaggcttgcAG
IWGSC_CSS_1BL_scaff_3899789	Kronos3191	7925	C	T	Het	Het	acttcacTggcagcagC	acttcacTggcagcagT	caaagtgtgtcccatGtA
IWGSC_CSS_2AL_scaff_6426728	Kronos3191	1481	G	A	Hom	Hom	gaaActgccgcgtCgC	gaaActgccgcgtCgT	ccaGcaGcgtgtgagaaA
IWGSC_CSS_2BL_scaff_7960273	Kronos3191	690	C	T	Hom	Hom	gcccattatccttaggcG	gcccattatccttaggcT	acatgcattgtgtgatgactG
IWGSC_CSS_3AS_scaff_3286603	Kronos3191	2975	G	A	Het*	Hom	cctgtgtttttgtgtggG	cctgtgtttttgtgtggA	gaaggaaacgtgTcaTgcaG
IWGSC_CSS_5AL_scaff_2694249	Kronos3191	2399	C	T	Het	Het	gccttcaggatagagccGC	gccttcaggatagagccGT	cggcacatcgacattctcG
IWGSC_CSS_5BL_scaff_10923577	Kronos3191	3713	C	T	Het	Het	gtggattgcgtgagcttG	gtggattgcgtgagcttG	tgggtggccttgggAC
IWGSC_CSS_6AL_scaff_5823017	Kronos3191	13225	C	T	Hom	Hom	cccttgcgccttggAG	cccttgcgccttggAA	ttcgagaaggccatcgA
IWGSC_CSS_6BS_scaff_2955394	Kronos3191	1622	C	T	Het*	Hom	gtggagatgggttgcgtcaaG	gtggagatgggttgcgtcaaA	gatactcgTcaatgggtG
IWGSC_CSS_7BL_scaff_6739382	Kronos3191	12261	G	A	Hom	Hom	gagacaaggatgttgcgtcC	gagacaaggatgttgcgtcT	CgggtgactCTatttccG
IWGSC_CSS_1AS_scaff_3276389	Kronos3288	9720	C	T	Hom	Hom	aCcaGcaggaccAatgtctC	aCcaGcaggaccAatgtctT	atgatgcaaccctcgccAT
IWGSC_CSS_2AL_scaff_6367515	Kronos3288	6976	G	A	Het	Het	caggtcgagTgtctccG	caggtcgagTgtctccG	gggggtatCtggaggG
IWGSC_CSS_2AL_scaff_6422019	Kronos3288	4523	G	A	Het	Het	cgttagtccctgcatacG	cgttagtccctgcatacA	acgcAcgtcaaccgtA
IWGSC_CSS_3AL_scaff_4284850	Kronos3288	7901	C	T	Hom	Hom	tggcttggacacatcG	tggcttggacacatcA	tgtcAgcatcgacagccA
IWGSC_CSS_4AS_scaff_5962359	Kronos3288	13049	G	A	Het	Hom	ccatcaagaaggatcagggtcG	ccatcaagaaggatcagggtcA	accatgcggccatgtcA
IWGSC_CSS_6AL_scaff_5777873	Kronos3288	6853	G	A	Het	Het	gagtgcaccccttcgttT	gagtgcaccccttcgttT	ggagaacacgtactcgG
IWGSC_CSS_6AS_scaff_4392100	Kronos3288	3434	C	T	Het	Het	atggaaagcagggtgaccG	atggaaagcagggtgaccA	ggAaggcggaaatggaaacAA
IWGSC_CSS_7BL_scaff_6744240	Kronos3288	9772	G	A	Het	Het	agctgttctcttacttcaaG	agctgttctcttacttcaaA	caggctgttcttgcgtcC
IWGSC_CSS_1AL_scaff_3887185	Kronos3413	9708	C	T	Hom	Hom	gcacgccttattcgaggtaaA	gcacgccttattcgaggtaaA	AgaaacagcagcgcA
IWGSC_CSS_2BS_scaff_3381362	Kronos3413	5160	C	T	Het*	Hom	caacttgcgtgttgtG	caacttgcgtgttgtA	tgAgaattctgacGaaaaAG
IWGSC_CSS_3AS_scaff_3296605	Kronos3413	6154	G	A	Het	Het	ctggtcacgggtctagC	ctggtcacgggtctagT	cacgactgagacatggGA
IWGSC_CSS_3B_scaff_10693516	Kronos3413	12632	C	T	Het	Het	ctaggcttggacaaaaggC	ctaggcttggacaaaaggT	agcttgcattatcggtcAT
IWGSC_CSS_5AS_scaff_1547699	Kronos3413	2686	G	A	Het	Het	gTCacaccccttccaaatcgC	gTCacaccccttccaaatcgT	gacggctttaatgtgtcAT
IWGSC_CSS_5BL_scaff_10856077	Kronos3413	5853	G	A	Het	Het	agagcttccccatgtc	agagcttccccatgtcT	acgCacatttAatagctgaaG
IWGSC_CSS_6AL_scaff_5750718	Kronos3413	11046	G	A	Hom	Hom	cacgcTTcccgacttataG	cacgcTTcccgacttataA	AgacgatgtatcgaggattcA
IWGSC_CSS_7AL_scaff_4433177	Kronos3413	3511	C	T	Het	Het	GaTgcctcGtcaggctgG	GaTgcctcGtcaggctgA	cactactggacaaagcttG
IWGSC_CSS_7BL_scaff_6742567	Kronos3413	667	C	T	Het	Het	gttgcgttgcgtggcagaC	gttgcgttgcgtggcagaT	cattttgcacgtgtgtcTG
IWGSC_CSS_1AL_scaff_3976389	Kronos3935	10941	C	T	Hom	Hom	ggtgaggagatcgCgtG	ggtgaggagatcgCgtA	cagtcattatcgaggggtcA
IWGSC_CSS_1BL_scaff_3873362	Kronos3935	1392	G	A	Het	Het	cagatctgaaggctaGacatG	cagatctgaaggctaGacatA	actaccagaatcgcacaaaaAC
IWGSC_CSS_2BL_scaff_7882382	Kronos3935	2721	C	T	Het	Het	gcaagctaaggatgtacgtG	gcaagctaaggatgtacgtT	gccacagttaggagaaagactT
IWGSC_CSS_3AL_scaff_4242376	Kronos3935	2410	C	T	Het	Het	agaacccaaaaccGacttaG	agaacccaaaaccGacttaA	gtagGgtCcattCtaaagcttG
IWGSC_CSS_3B_scaff_10485067	Kronos3935	3349	C	T	Hom	Hom	gttggaccaacttcaactG	gttggaccaacttcaactA	gcaatttcccttaTccgcagT
IWGSC_CSS_4AS_scaff_5984153	Kronos3935	6006	G	A	Het	Het	agCaggctggccaaagtT	agCaggctggccaaagtA	cgaatGtatgaGtaggcgcT
IWGSC_CSS_4BL_scaff_7019402	Kronos3935	9081	C	T	Het	Het	tgcaatcatgtgtgagctG	tgcaatcatgtgtgagctG	agcatgatcccttagaaCatac
IWGSC_CSS_5BL_scaff_10842786	Kronos3935	3304	G	A	Het	Het	tggttcccGaaggctgaaC	tggttcccGaaggctgaaT	cgcatacttgaacaTGagcAC
IWGSC_CSS_6BS_scaff_3045205	Kronos3935	2293	G	A	Het	Het	aaggaccaaggccaaactctcG	aaggaccaaggccaaactctcA	agtgtaccaaggccaaatgtcG

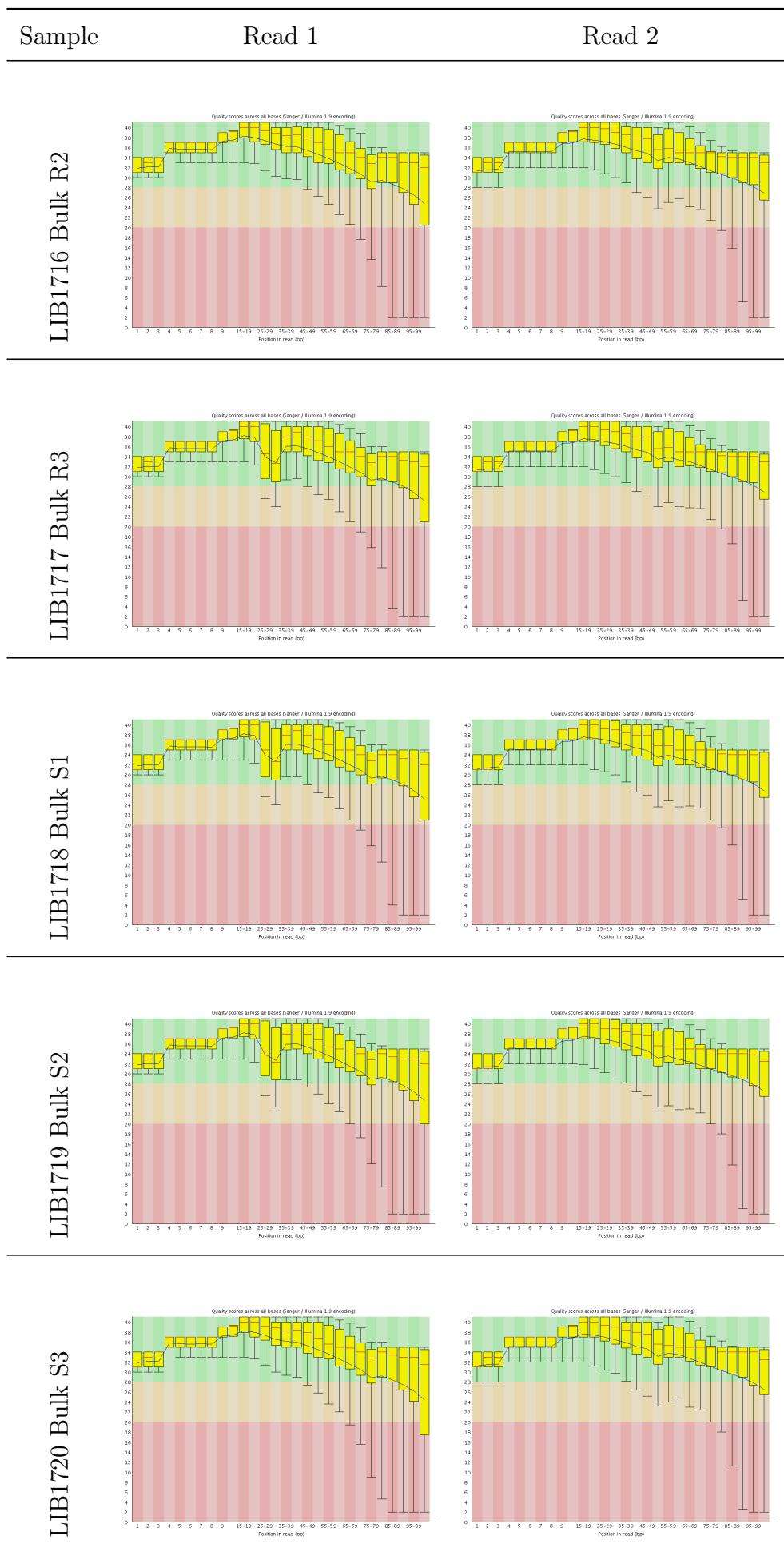
IWGSC contig	Line	Pos	WT	Mut	Predicted	<i>M</i> ₄	Primer 1 (Kronos)	Primer 2 (mutant)	Common Primer
IWGSC_CSS_7AL_scaff_4555249	Kronos3935	4487	C	T	Het	Het	cAgtgctcgagatggcgC	cAgtgctcgagatggcgT	cCttgcacccctcgatT
IWGSC_CSS_1BL_scaff_3918498	Kronos4240	6096	G	A	Het	Het	ttgcattccccaaagaagaG	ttgcattccccaaagaagaA	tggcgaactgtaatgtG
IWGSC_CSS_2BS_scaff_5131713	Kronos4240	5900	G	A	Het	Het	cctttatcgaggaaagagacacC	cctttatcgaggaaagagacacT	caccattgttaggttccttTttC
IWGSC_CSS_5AL_scaff_2769540	Kronos4240	9626	C	T	Het	Het	tgCagtgtggaaacggaG	tgCagtgtggaaacggaA	catgagtGagatcttcgtcT
IWGSC_CSS_5BL_scaff_10871091	Kronos4240	7062	G	A	Het	Het	gccaaggAaccataacctgC	gccaaggAaccataacctgT	GgacttggcAaccggA
IWGSC_CSS_6AL_scaff_5800333	Kronos4240	2360	G	A	Het	Het	cgacaggattgtgagCgC	cgacaggattgtgagCgT	tcaagatgtcaagattcatcT
IWGSC_CSS_7BL_scaff_6716931	Kronos4240	2613	G	A	Het	Het	gGtgGgtattTgcttggtaG	gGtgGgtattTgcttggtaA	tgGtggactcgacaGtGtA
IWGSC_CSS_2BL_scaff_8029221	Kronos4346	2860	G	A	Het	Het	tgcttccgtctgtctC	tgcttccgtctgtctC	atTtgcattCgAtcggcC
IWGSC_CSS_3B_scaff_10460714	Kronos4346	14359	C	T	Hom	Hom	ctacccatcgacatG	ctacccatcgacatA	agcaccccgatcttgcacG
IWGSC_CSS_4AS_scaff_5989735	Kronos4346	6404	G	A	Hom	Hom	acgcattctaaccatcagcC	acgcattctaaccatcagcT	actcaagataccacCcgcacG
IWGSC_CSS_5BL_scaff_7648030	Kronos4346	6893	C	T	Het	Het	tacccttctactggcagG	tacccttctactggcagA	ttttcagggAACACAGGTtAcA
IWGSC_CSS_6AL_scaff_5755840	Kronos4346	778	C	T	Het	Het	atcgagtaactgtcacCgC	atcgagtaactgtcacCgT	acctgcattgtcaCatccC
IWGSC_CSS_6BS_scaff_2972151	Kronos4346	7876	G	A	Hom	Hom	gcagaatgtcActgtttG	gcagaatgtcActgtttA	gcttggactggcattttG
IWGSC_CSS_7AL_scaff_4542983	Kronos4346	18700	G	A	Het	Het	gcaggcgtAccggatacT	gcaggcgtAccggatacT	catctgccGgttaaacatcG
IWGSC_CSS_7BS_scaff_3098098	Kronos4346	5183	C	T	Het	Het	gCgatatggtaacttgcataG	gCgatatggtaacttgcataA	tttacattgttataGTtgtCcgG
IWGSC_CSS_1AS_scaff_3259804	Kronos4485	219	C	T	Het	Het	gtcggcacaaccccttgC	gtcggcacaaccccttgT	gcttcttaaggaggcgcA
IWGSC_CSS_2AL_scaff_6315418	Kronos4485	10490	G	A	Hom	Hom	gcccctctcaaCttctcagC	gcccctctcaaCttctcagT	ttcagacgtCgaggatttccC
IWGSC_CSS_2BS_scaff_5181092	Kronos4485	3742	G	A	Het	Het	TggccagcacactgcAG	TggccagcacactgcAG	tggacgttagTgtatggAaAT
IWGSC_CSS_3B_scaff_10425015	Kronos4485	2372	C	T	Het	Het	gctactaaggatggCtcGG	gctactaaggatggCtcGA	cttcacatccctggggTtC
IWGSC_CSS_3B_scaff_10775915	Kronos4485	4701	C	T	Het	Het	ccaaggcgtcagagagG	ccaaggcgtcagagagA	agacctcacatGtcctcC
IWGSC_CSS_5AL_scaff_2754304	Kronos4485	2301	G	A	Het	Het	taaccTGccatcgccccG	taaccTGccatcgccccA	cattGccaggcaTgacT
IWGSC_CSS_5BL_scaff_10919959	Kronos4485	1867	C	T	Hom	Hom	gatgcccttggagaagG	gatgcccttggagaagA	tcttgtccgaaacatgtcA
IWGSC_CSS_7AS_scaff_4245431	Kronos4485	3402	G	A	Hom	Hom	aaggcgctgtgttc	aaggcgctgtgttc	agtaagtggAAcgtcaagatcaT
IWGSC_CSS_7BL_scaff_6667357	Kronos4485	641	C	T	Het	Het	gatcAgctgctcattcgagG	gatcAgctgctcattcgagA	ttccctgtcaattgtgccC

Appendix B

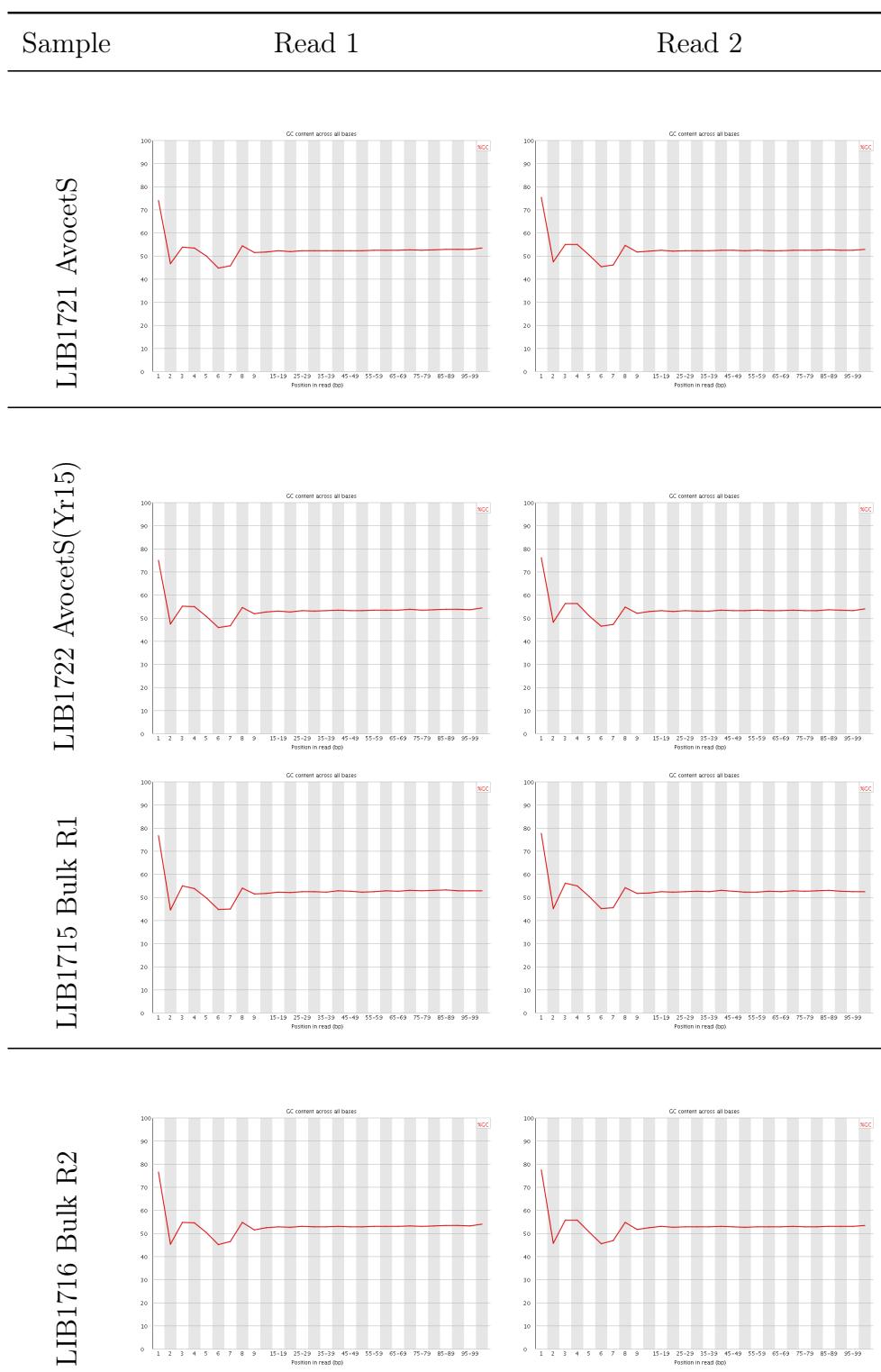
Quality Control

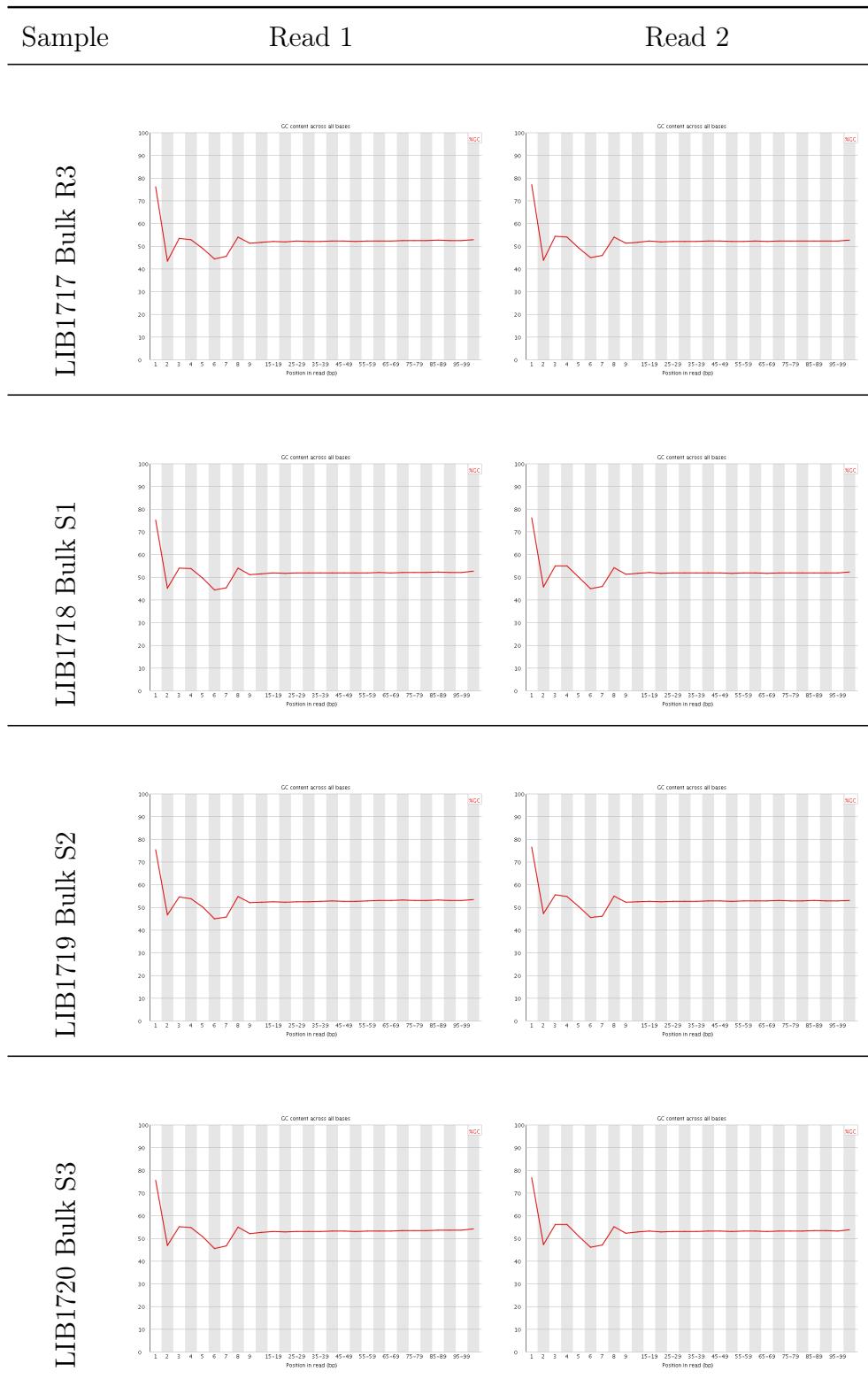
B.1 Sequence read quality





B.2 Sequence GC content





Appendix C

expVIP tutorial

This repository Search Pull requests Issues Gist

homonecloco / expvip-web

Code Issues 5 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

AllConcat

Ricardo H. Ramírez-Gonzalez edited this page a minute ago · 1 revision

Welcome to the expvip-web wiki! This wiki contains tutorials on how to setup the database and run it locally.

Updating the virtual machine Before loading the metadata, double click on `update_expvip.sh` in the desktop to get the latest version of expVIP.

1. [Loading Virtual Machine](#). Instructions on how to setup Virtual Box to run expVIP
2. [Loading Metadata](#). Detailed scripts in the virtual machine to prepare expVIP for your samples.
3. [Loading data](#). Description on how to prepare and load the data to expVIP.
4. [Running Kallisto](#). Instructions on how to run Kallisto and load the results in the database in a single step
5. [Running Kallisto in batch](#). Instructions on how to run Kallisto and load the results in the database in a single step from multiple samples
6. [Starting up the web server](#). Instructions on how to start the local web server for expVIP
7. [Exporting Data](#). How to extract data from expVIP database.
8. [Graphical Interface Tutorial](#). How to get the most of the expVIP graphical interface, exemplified with the [Wheat Genome Browser](#).

Pages 11
Home
AllConcat
ExportData
List of tutorial videos
LoadingData
LoadingMetadata
LoadingVM
RunKallisto
RunKallistoBatch
StartWebServer
Tutorial expVIP Graphical Interface (Wheat Expression Browser example)

Loading Virtual Machine

The expVIP Virtual Machine (VM) allows you to analyse your own RNA-Seq expression experiments locally.

Requirements

The virtual machine requires:

- [VirtualBox](#), version 5 or newer.
- 6GB of RAM
- A 64-bit operating system running on an x86_64 architecture. (Intel, AMD)
- 10GB of free space.

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/homonecloco/expvip-web>

[Clone in Desktop](#)

Default data

The default values loaded in the virtual machine are available in this [link](#). These correspond to the wheat data from Borrill, Ramírez-Gonzalez and Uauy, 2015 (*submitted*).

You can get a virtual machine with expVIP installed with either the wheat data preloaded or an empty database for your analysis [here](#).

Available VMs

1. **expVIPNoData.ova** This VM is ready to use, but it has no data on it. You can load a custom set of RNA-seq reads, transcriptome reference and metadata.
2. **expVIPwithWheatData.ova** This has all the data loaded from www.wheat-expression.com. You can add your own data and compare it with the values of publicly available experiments.

Setup shared folders

To load your custom RNA-seq experiments, you have to setup a shared folder with your input files. This shared folder will contain the data and information required by the VM to implement expVIP and it provides the "connection" between your computer and the VM. This shared folder should include:

1. RNA-seq reads: as `fastq` or `fq.gz` files.
2. Transcriptome reference: currently only the cdna fasta file from ensembl is supported.
3. Metadata: this includes two separate files; one factor file and one metadata file ([explained here](#)).

Some important information:

- The shared folder must contain one sub-folder per each set of RNA-seq reads. So for example if you wish to analyse data from three samples, you will need three sub-folders (one each with the individual sample RNA-seq reads)
- Each RNA-seq sub-folder must be named with the same accession number that you use in your metadata (see [here](#)).
- If you wish to add your own wheat data to that previously provided in www.wheat-expression.com you will need to include sub-folders with your RNA-seq reads and then modify the metadata files: `default_metadata.txt` and `FactorOrder.tsv` which are provided in the `expVIPwithWheatData.ova` or can be downloaded [here](#). Additional factors and metadata can be added at the end of these files following similar nomenclature as that already present in the files.

Loading the virtual machine

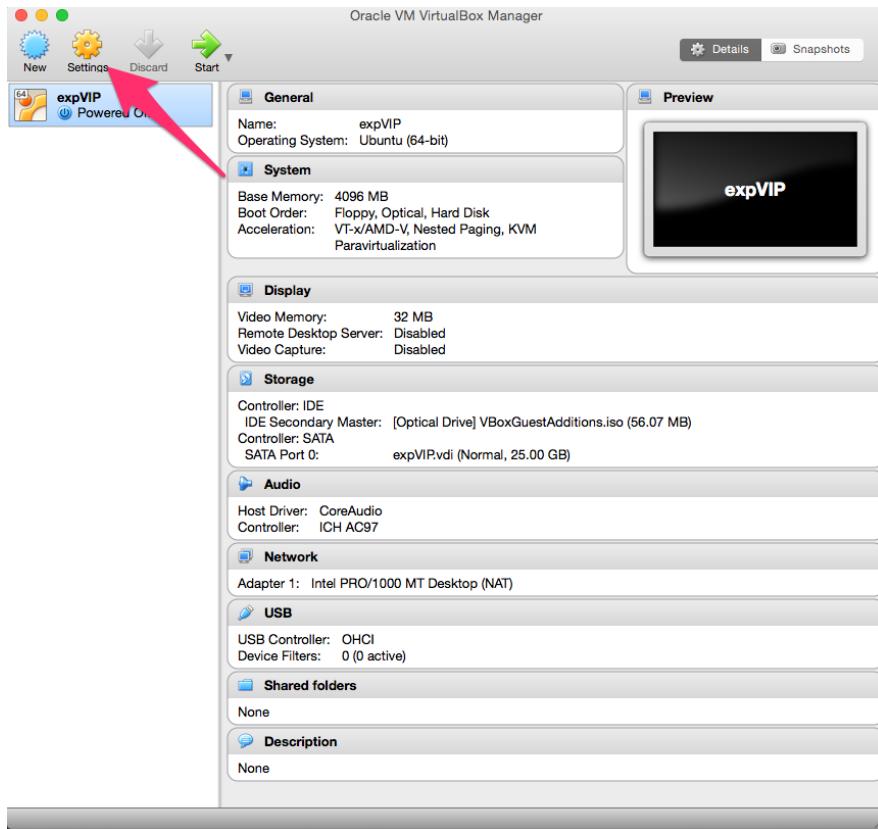
Download the `.ova` virtual machine and double click it. Virtual Box will open it. Accept the default options.

If the virtual machine is not loaded, go to the menu `File` and click `Import appliance`. Open the `.ova` you want to use

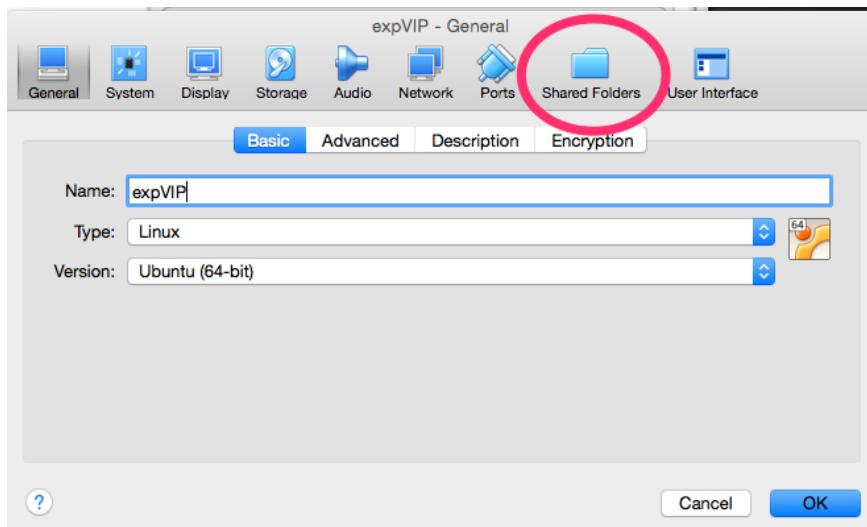
Available VMs:

- `expVIP.ova` expVIP is installed with an empty database. This VM requires to setup your own samples.
- `expVIPwithWheatData.ova` expVIP is installed with the wheat expression data, transcriptome reference and metadata. This VM allows the inclusion of additional samples to integrate with the previously analysed wheat data.

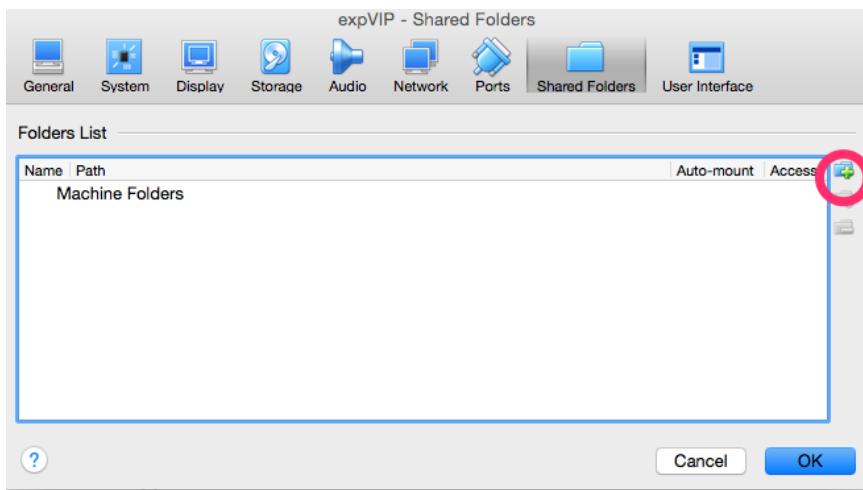
1. On the Oracle VM VirtualBox Manager select expVIP and click on the settings button



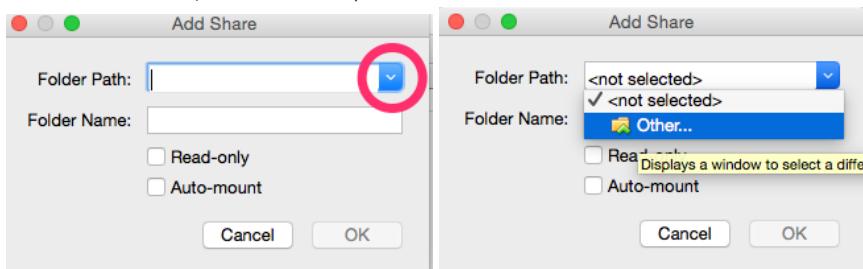
2. Click in Shared folders



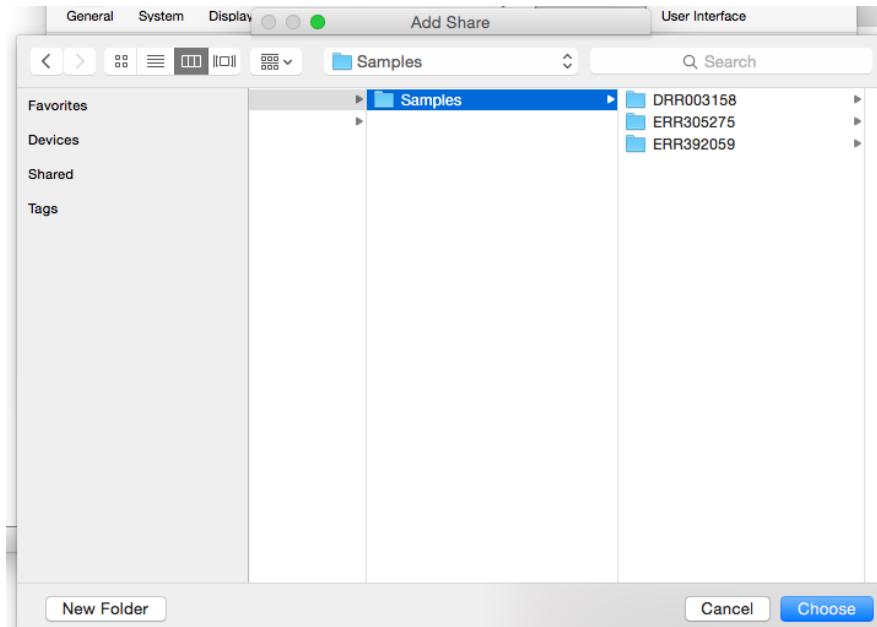
3. Add a new folder



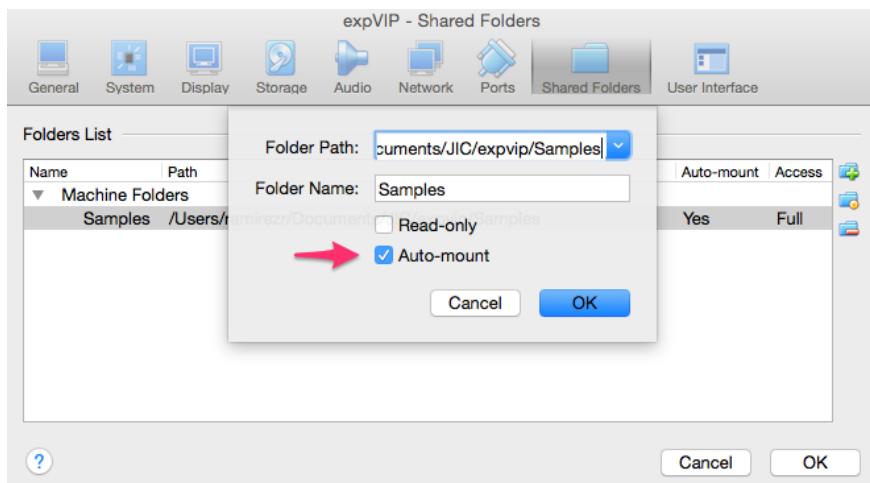
4. Search the Folder path with the experiments and the files with the metadata



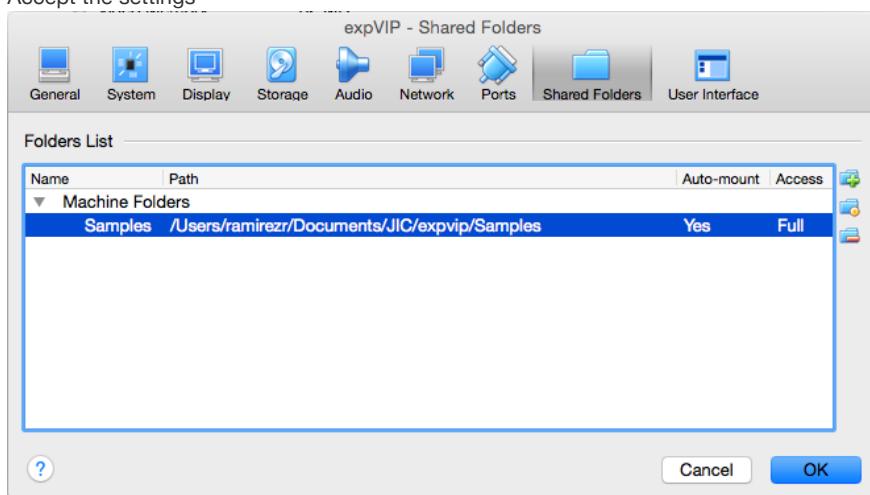
5. Select the folder



6. Make sure that the Auto-mount option is selected.

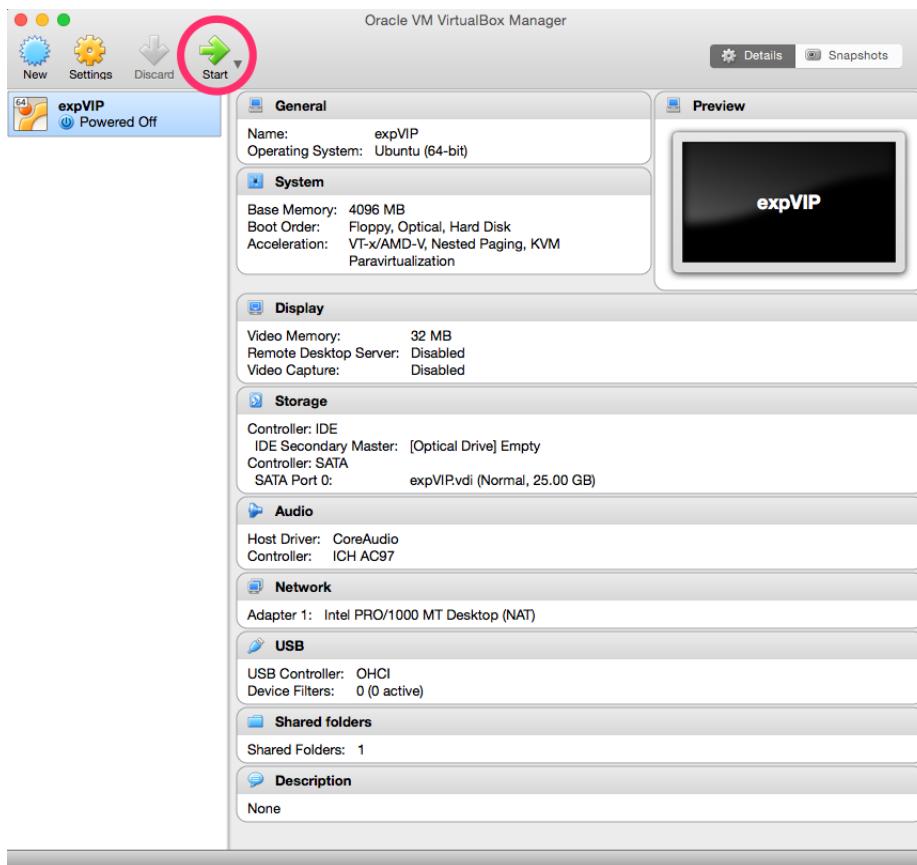


7. Accept the settings



Starting the virtual machine

Select expVIP from the VM list and press start.



NOTES

[kallisto](#) is included as part of the virtual machine and is free for non-commercial use. However, it requires a license for commercial use. The distribution of kallisto, with the corresponding license is included in `~/software/` in the VM.

Loading expVIP metadata

This tutorial covers the shell scripts that can be used to load the metadata with the graphical interface, with screenshots, and the rake task, to be run in the command line, if you are more comfortable in the terminal. The assumption is that expVIP is located in `~/expvip-web/`

Loading factor file

The first thing to do is to setup the available factors. The `factor` file is a text file, where each field is separated by tabs. A header is necessary on each column. The headers are the following:

- **factor**: The name of the factor to group. These must match those used in the metadata file (see below).
- **order**: Default display order in the graphical interface.
- **name**: The long name of the grouped factor. These must match those used in the metadata file (see below).
- **short**: Short name of the grouped factor. This is used in the graphical interface when many factors are displayed.

factor file example:

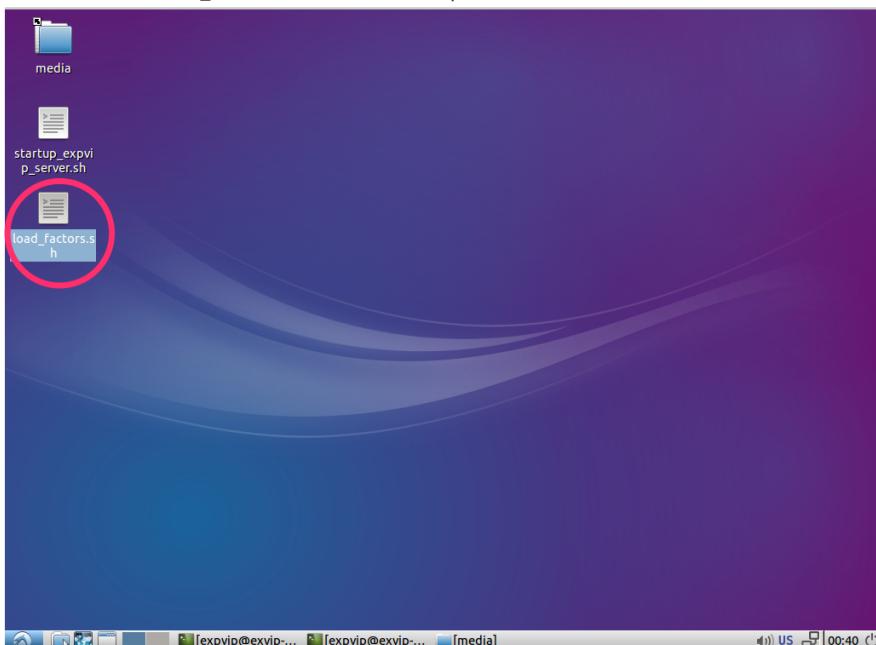
```

factor order name short
Age 1 7 days 7d
Age 2 seedling stage see
Age 3 14 days 14d
Age 4 three leaf stage 3_lea
Age 5 24 days 24d
Age 6 tillering stage till
Age 7 fifth leaf stage 5_lea
Age 8 1 cm spike 1_sp
Age 9 two nodes detectable 2_no
Age 10 flag leaf stage f_lea
Age 11 anthesis anth
Age 12 2 dpa 2dpa
Age 13 4 dpa 4dpa
High level age 1 seedling see
High level age 2 vegetative veg
High level age 3 reproductive repr
High level stress-disease 1 none none
High level stress-disease 2 disease dis
High level stress-disease 3 abiotic abio
High level stress-disease 4 transgenic trans
High level tissue 1 spike spike
High level tissue 2 grain grain
...

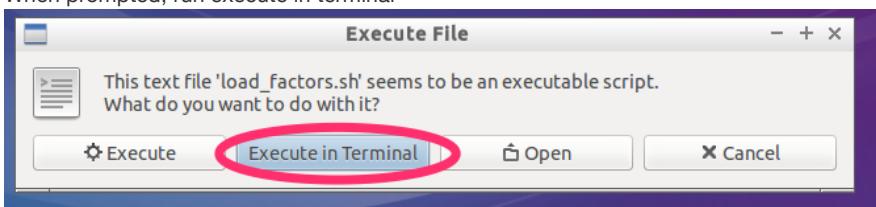
```

Wizard to load factors

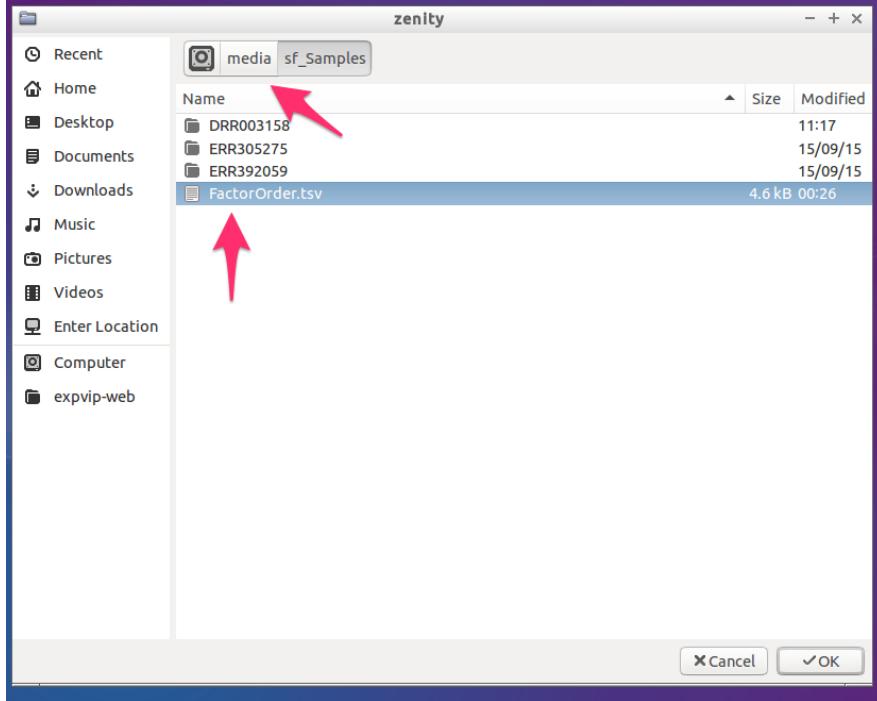
1. Double click in `load_factors.sh` in the desktop



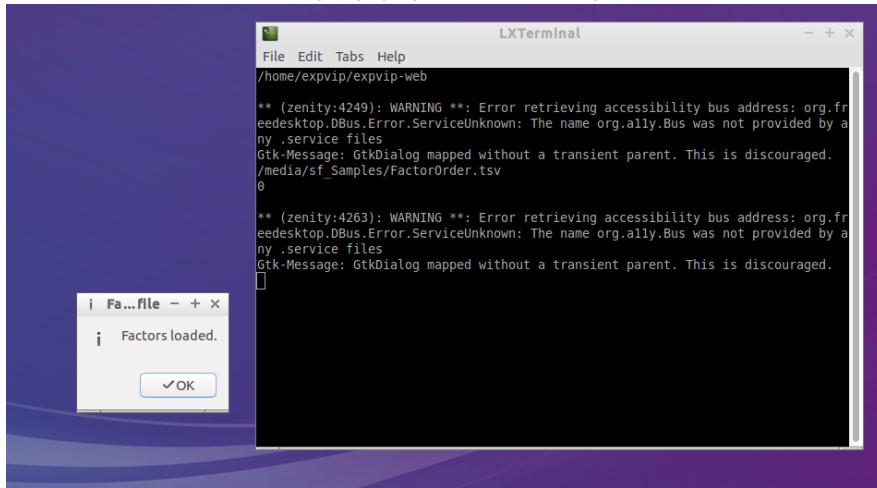
2. When prompted, run execute in terminal



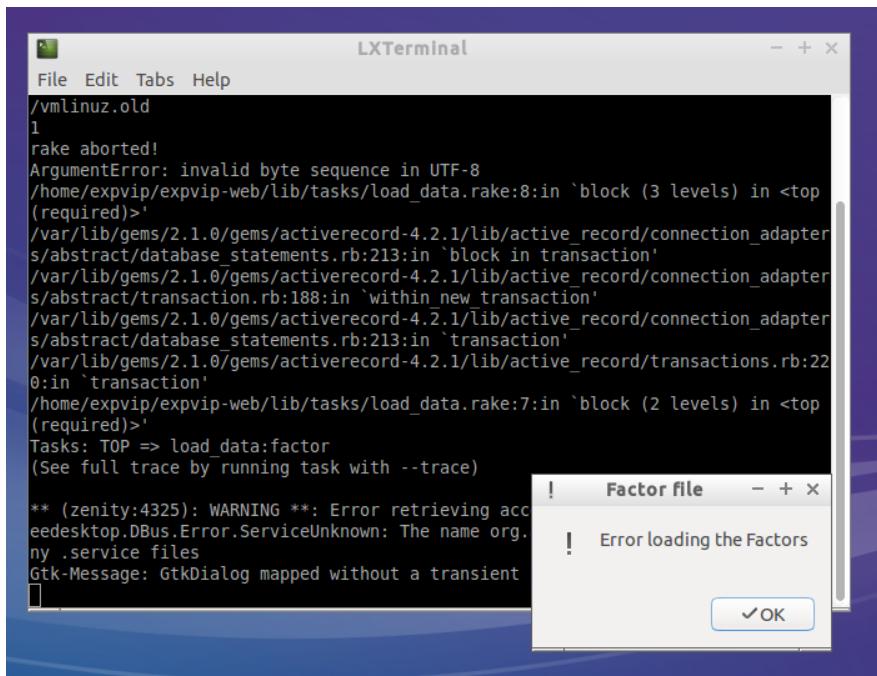
3. By default, the script goes to `/media`, which is the folder containing the shared folders that we have setup in the **LoadingVM** step.



4. If the factors are loaded correctly, a pop up window will notify about it



5. If there was an error loading the factors, a message will notify about it. The error log may give a hint of what went wrong, but if you can't figure out send a screenshot of the terminal to the developers.



Rake task

To load the factors, you can run directly the rake task from `~/exvpip-web/`.

```
rake load_data:factor[FILE_WITH_FACTORS];
```

Loading metadata

The second step is to load the experiment metadata. Currently, a tab separated file is the input and it **must** contain the following columns with the header named exactly as stated:

- **secondary_study_accession:** The accession number for experiments carried as part of a single study. This is usually the high level BioProject or SRA number.
- **run_accession:** The accession of the individual run.
- **scientific_name:** of the species.
- **experiment_title:** A description for the individual RNA-seq sample.
- **study_title:** A description of the general study.
- **Variety**
- **Tissue**
- **Age**
- **Stress-disease**
- **Manuscript:** The DOI of the study.
- **Group_for_averaging** A description of the experiment. This must be the same all the replicates in the same study.
- **Group_number_for_averaging:** A short name for replicated experiments.
- **Total reads:** (optional)
- **Mapped reads:** (optional)
- **High level variety:** A higher level grouping to get summarized data of the factors.
- **High level tissue**
- **High level age**

- High level stress-disease

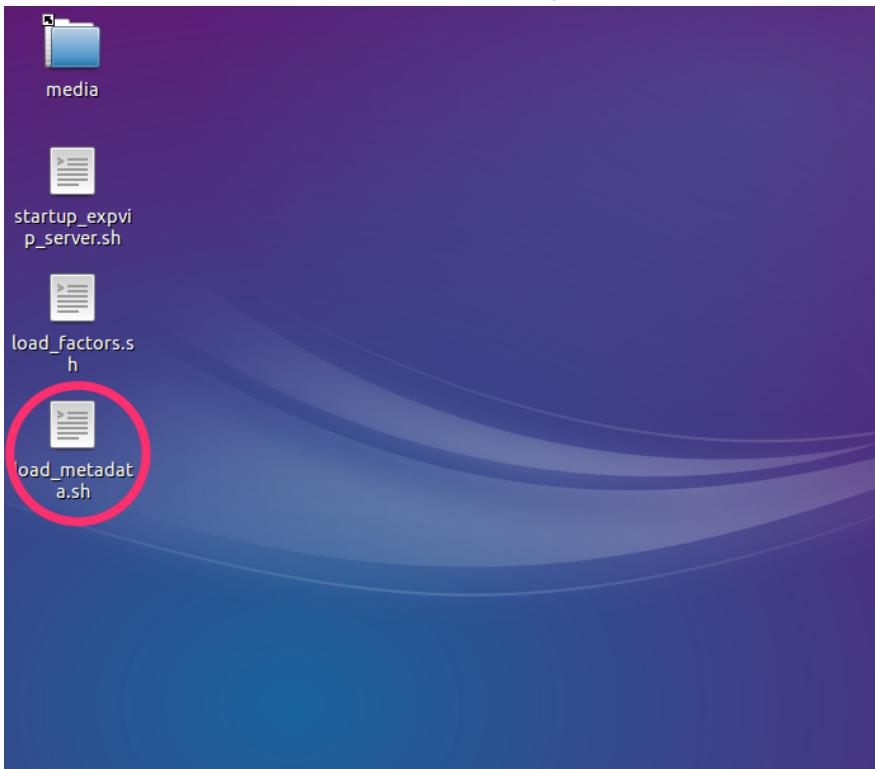
Important points

- Variety , Tissue , Age , Stress-disease , and their corresponding High level factors must be exactly the same as in the columns factor and name from the factor file (see above).
- The graphical interface will group samples based on these factors. Therefore these can be defined based on the user needs. For example the factor High level tissue will include tissue types such as grain , roots , spike and leaves/shoots . Within each of these tissue types, a more detailed description can be included under the Tissue heading. For example: starchy endosperm , seed coat , transfer cells , etc. RNA-seq samples which share factor names in common will be displayed as groups in the visual interface.
- If Mapped reads and Total reads are missing, you need to run kallisto mapping from the rake task.

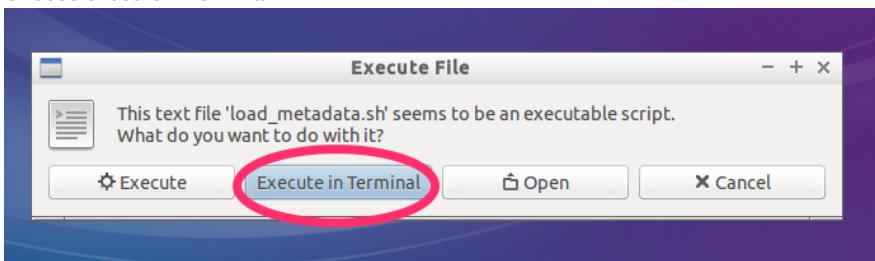
Using the graphical interface

The process is similar to loading the factors. However, the metadata file is selected.

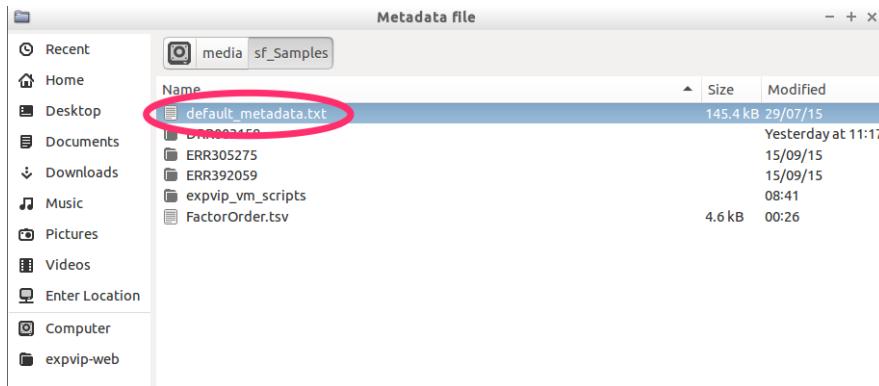
1. Double click on the `load_metadata.sh` icon in the desktop



2. Choose execute in terminal



3. Select the metadata file



Rake task

```
rake load_data:metadata[FILE_WITH_THE_METADATA]
```

Loading the gene sets

Before loading the actual expression data or running kallisto, it is necessary to load the gene models. Currently, only the fasta file with the cdna from ensembl is supported. The fasta header should contain the following fields, besides the gene name (first string in the header).

- **cdna**
- **chromosome** or **scaffold** are converted to position
- **gene**
- **transcript**
- **description** a free text, in quotes. Any other field with quotes may fail in the load.

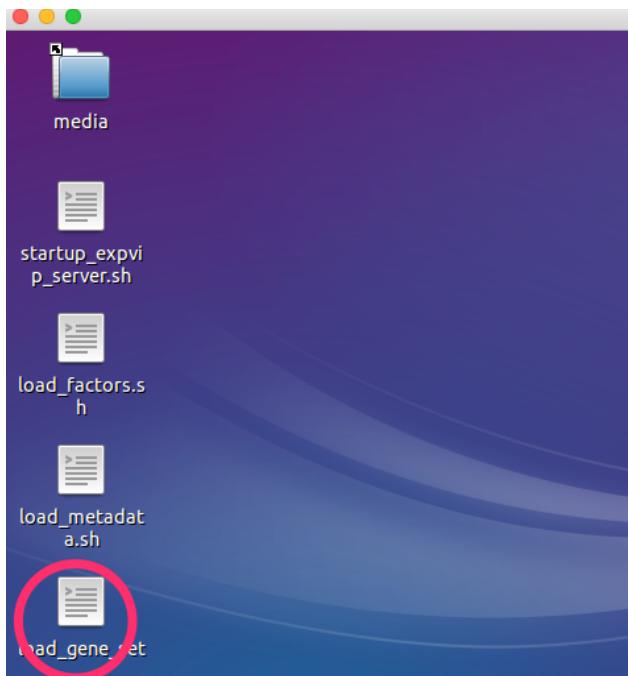
Besides the fasta file, it is necessary to give a name to the gene set. For this tutorial, the gene_set will be IWGSC2.26

Example fasta file

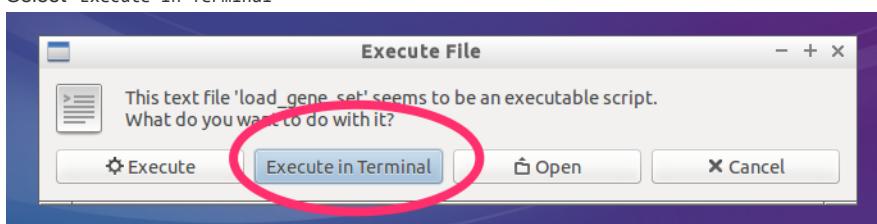
```
>Traes_5BL_3FC5BA305.1 cdna:novel scaffold:IWGSC2:IWGSC_CSS_5BL_scaff_1082268:5:199:-1 ge
TGCTGCTGCTAGGCTTGAAGAGGTTGCTGGCAAGCCTCCAGTCAGTCTGCTCGGCAGCTCATTC
GAGGGGCTGTGAGGAGTGCCCCAAGAACGAGGATGTTGGTTGAGGCATGCCGGTTGG
TAGCCAGATGAGTCAGTAATTGCCAGGGGTGTGAAGGCAATTCCAACCTCTGT
GAAGCTGTGGCTGCA
>Traes_6BL_9BB648D51.1 cdna:novel scaffold:IWGSC2:IWGSC_CSS_6BL_scaff_430516:302:1741:-1
TCCCTATCTGTTCTTGCGACTCCCTGATCCAATCGATCCATCAGGGCTCGACTAACT
TCTTCAGCGCTCTTCAAGCGGGAGATCTACCAGCTGGCGGAGGGGGCTAGGTGCA
GGCGTGCAGCCAAAGTCCGCACCCGGCTCTAGGTTCTGCTAATCTTCTCCACCTGTGA
TACGCGCTCCGGGCTAGGAGCACTCGTTGCCGGCTGCCCTCGTCGGAATGGCGGATG
```

Graphical interface

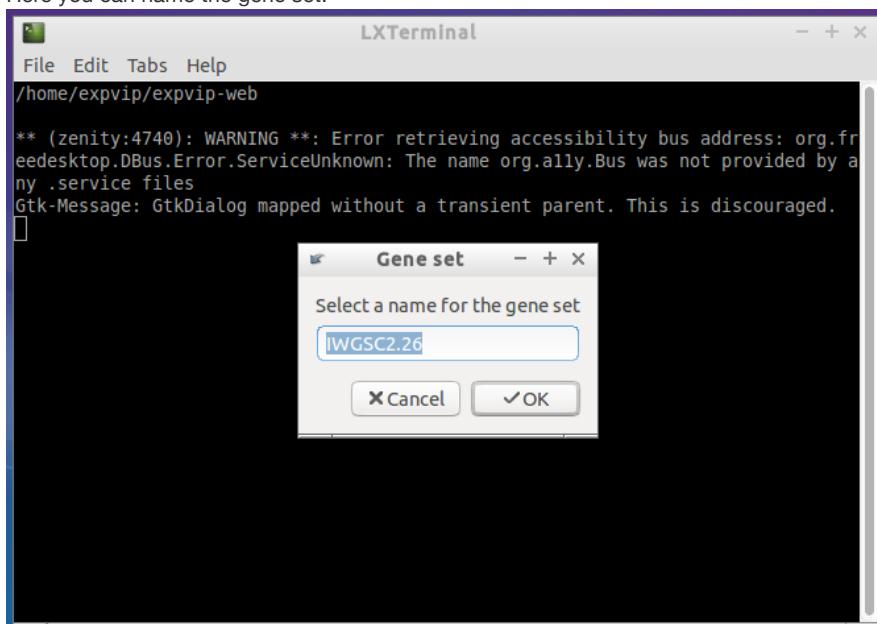
1. Double click on the `load_gene_set` script



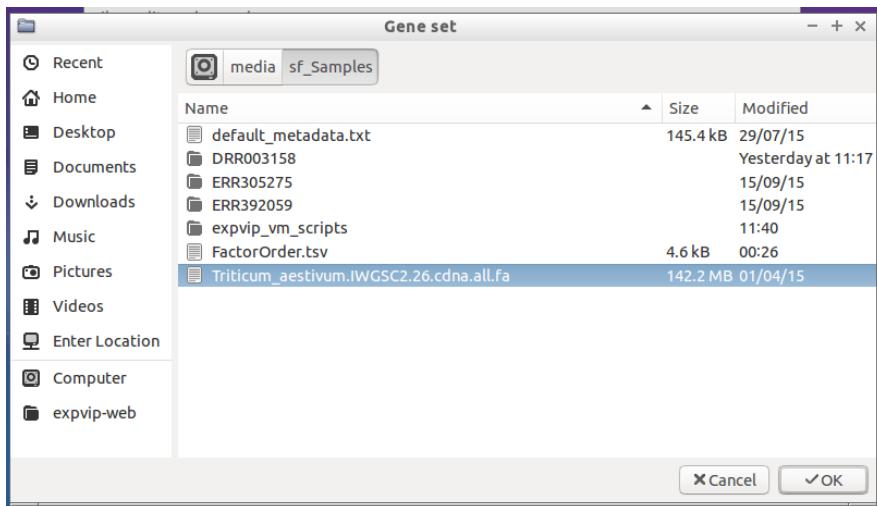
2. Select Execute in Terminal



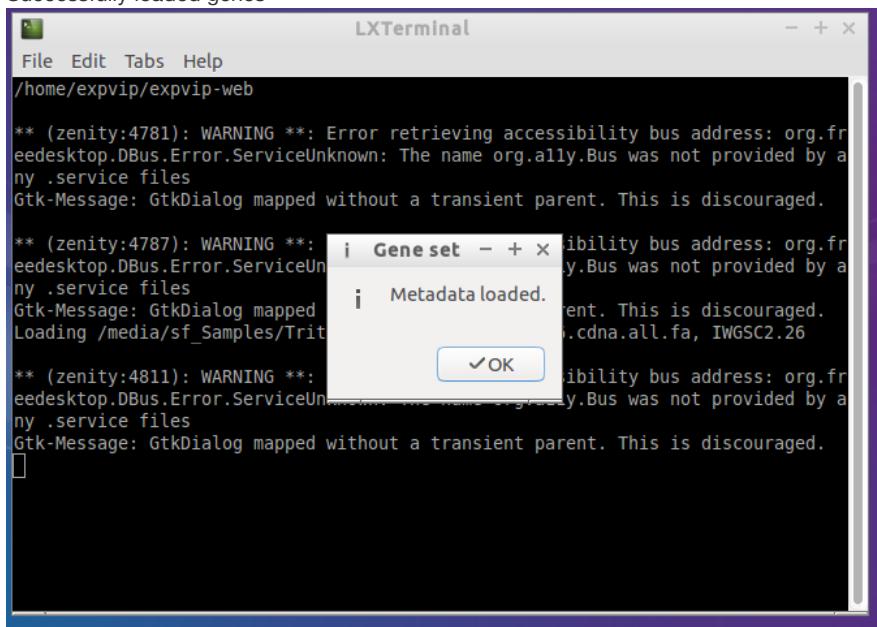
3. Here you can name the gene set.



4. And select the reference file. This may take a few minutes to load.



5. Successfully loaded genes



Rake task

```
rake load_data:ensembl_genes[IWGSC2.26,/Triticum_aestivum.IWGSC2.26.cdna.all.fa]
```

Loading the homoeologues

In order to show the homoeologues, a file with the homoeologies must be loaded. The file is tab separated with the following format:

Gene	A	B	D	Group	Genome
Traes_5BS_0AFC3F795		Traes_5BS_0AFC3F795		Traes_5DS_C204EBAA9	5 B
Traes_5DS_C204EBAA9		Traes_5BS_0AFC3F795		Traes_5DS_C204EBAA9	5 D
Traes_7DL_82360D4EE1				Traes_7DL_82360D4EE1	7 D
Traes_2AL_1368BE0AD	Traes_2AL_1368BE0AD			Traes_2AL_CD459994C1	2 A
...					

Note that the gene names are not the same as the transcript names, they correspond to the gene name.

Generating the file with the homoeologues from Ensembl compara

The file can be generated with ensembl compara, using the following query:

```
SELECT
    homology_member.homology_id, cigar_line, perc_cov, perc_id, perc_pos,
    gene_member.stable_id as genes,
    gene_member.genome_db_id

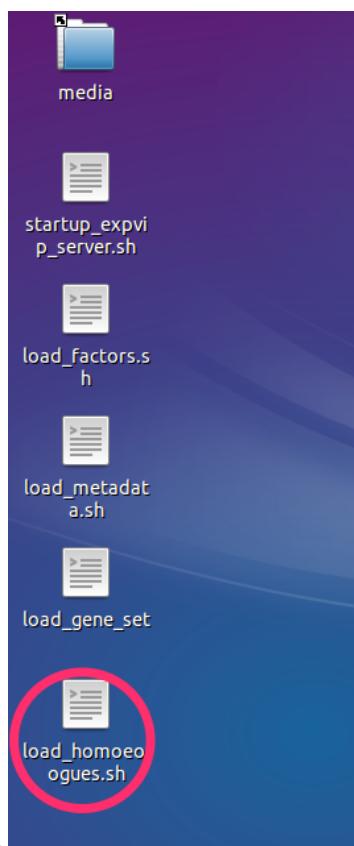
FROM
    homology_member
INNER JOIN homology USING (homology_id)
INNER JOIN method_link_species_set USING (method_link_species_set_id)
INNER JOIN gene_member USING (gene_member_id)
WHERE method_link_species_set.name="T.aes homoeologues";
```

Then, to format the result of the query (saved as `compara_homology.txt`), you can use the provided script

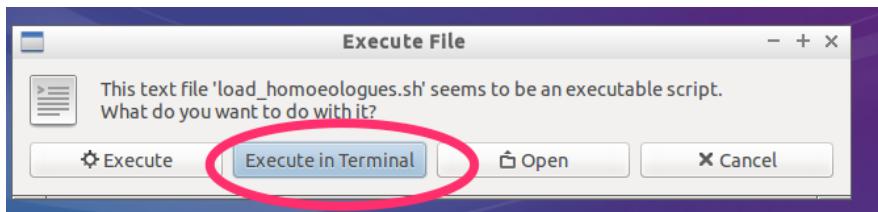
```
ruby bin/homologyTable.rb compara_homology.txt homology.txt homology_counts.txt
```

You can get your homoeologies elsewhere, as long as you keep the file format.

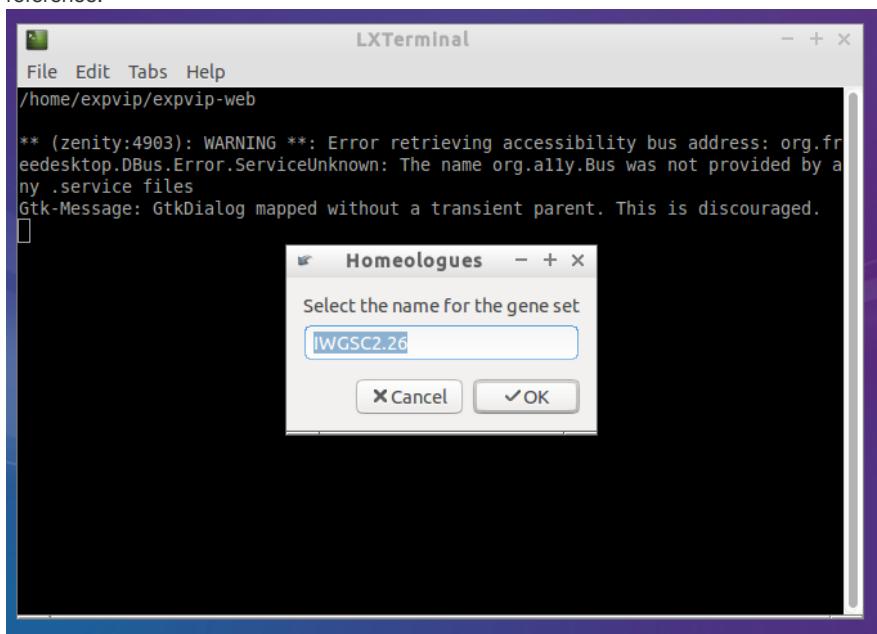
At this point, the homoloegues are called A,B and D. This is going to change on a future release to allow any chromosome group naming.



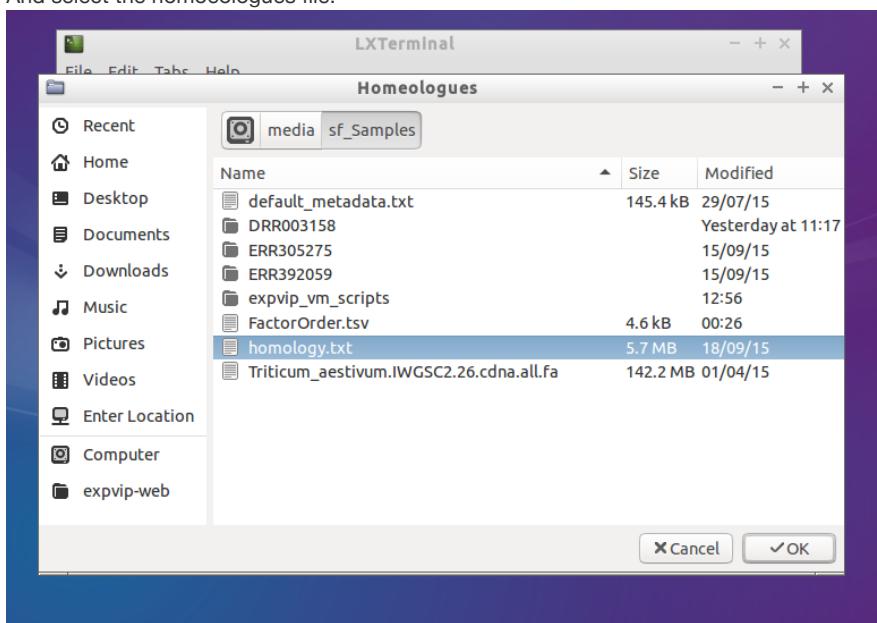
1. Double click on the `load_homoeologues.sh` script
2. Select Execute in Terminal



3. Here you can name the gene set. It must be the same name you added for the gene reference.



4. And select the homoeologues file.



5. Successfully loaded. At the end of the log you can see which how many homologies where loaded.

```

File Edit Tabs Help
ny .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

** (zenity:5266): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.
Loading /media/sf_Samples/home/allybus/IWGSC2.26/.service
{:gene_set=>"IWGSC2.26", :file=>"/home/allybus/IWGSC2.26/homology.txt"}
Loaded 103274 genes in memory
Loaded 10000 Homologies (Trae...
Loaded 20000 Homologies (Trae...
Loaded 30000 Homologies (Trae...
Loaded 40000 Homologies (Trae...
Loaded 50000 Homologies (Trae...
Loaded 60000 Homologies (Traes_5DL_7B922D9F7)
Loaded 70000 Homologies (Traes_2BS_7A9FD08C2)
Loaded 79349 Homologies

** (zenity:5279): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

```

Rake task

```
rake load_data:homology[IWGSC2.26,/homology.txt]
```

Data loading

Once the database has been created, expVIP currently supports two methods to load expression data onto the database:

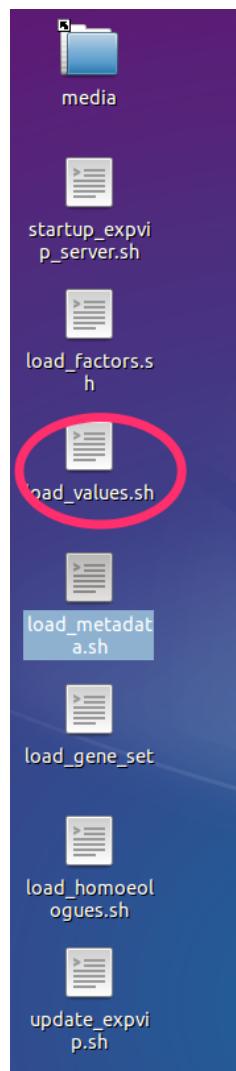
1. Load the precomputed expression values into the database, or
2. Run kallisto to generate the expression data. These are then loaded directly into the database.

Single big table

The fastest way to load the data to expVIP is to produce a table with all the values for each expression unit (tpm, counts). The table must contain a column `target_id` that has the gene name, as the first field in the fasta file used for the mapping. The rest of the columns must contain a header with the accession of the experiment. Each row represents a value. All the values in the table must be from the same time.

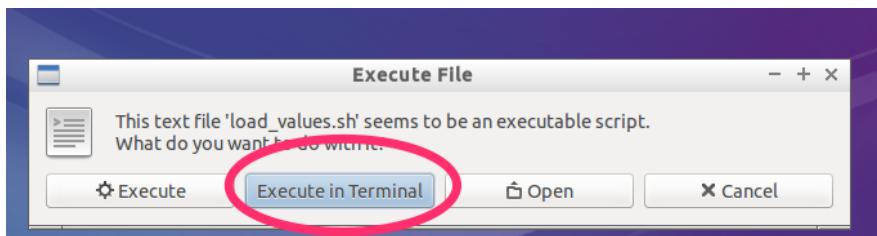
For the case of wheat in which we have already generated the kallisto mapping of 418 RNA-seq studies, the expression table can be downloaded directly from [here](#). The `.txt` files are called `final_output_counts.txt` and `final_output_tpm.txt` for the corresponding expression unit.

Loading from the script

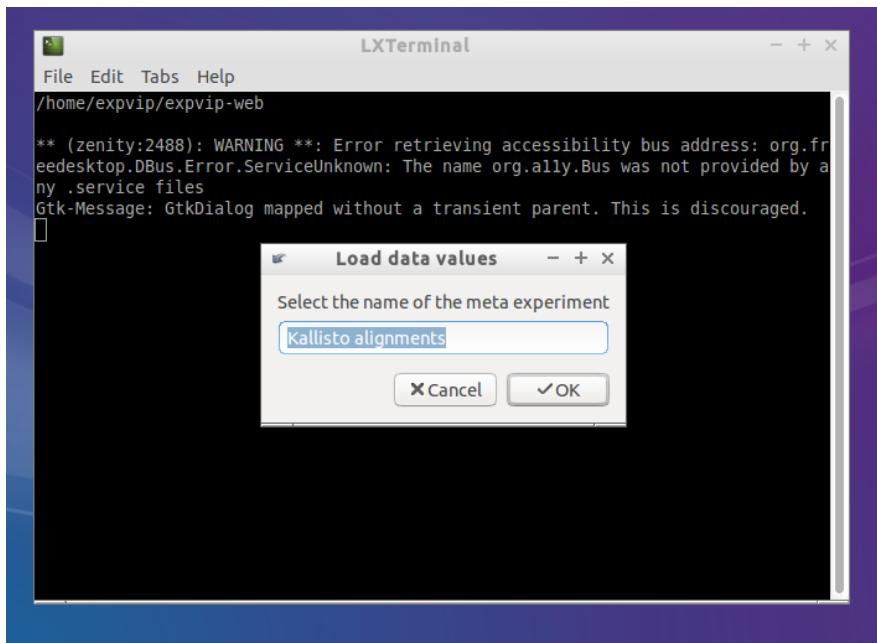


1. Double click on the `load_values.sh`

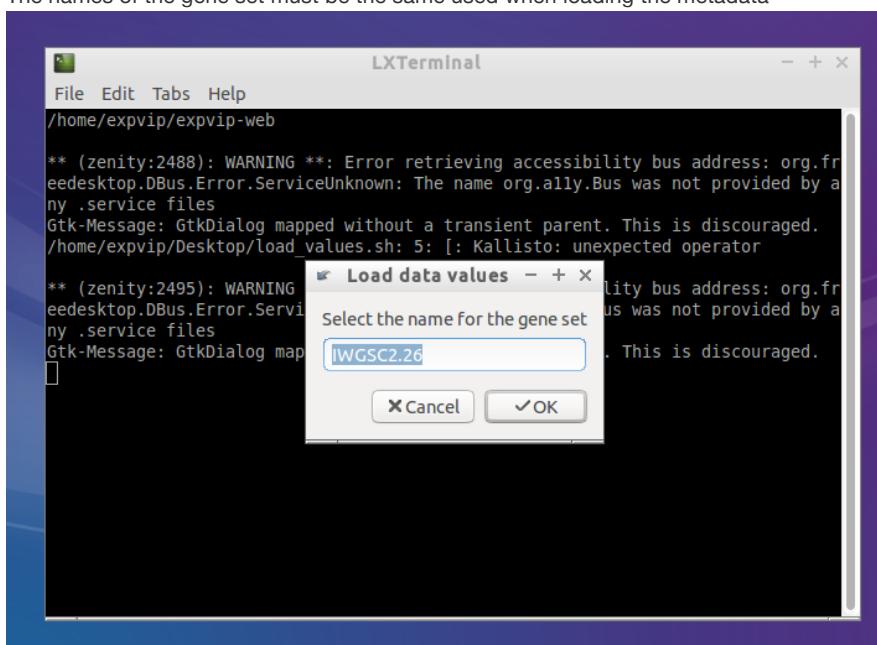
2. Click one Execute in Terminal



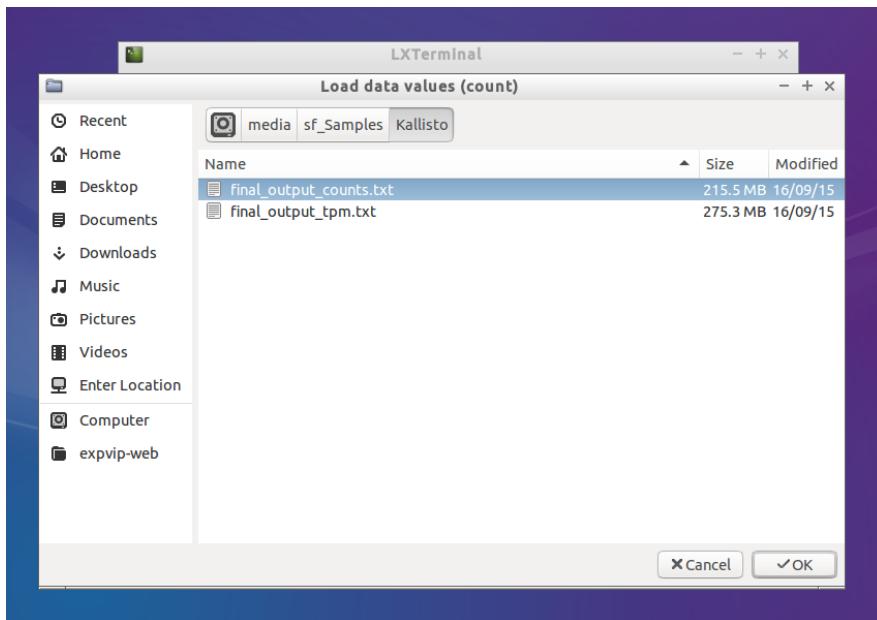
3. Select a name for the set of alignments. expVIP can keep several runs of alignments in the database. The ability to select between them will be added in a future release.



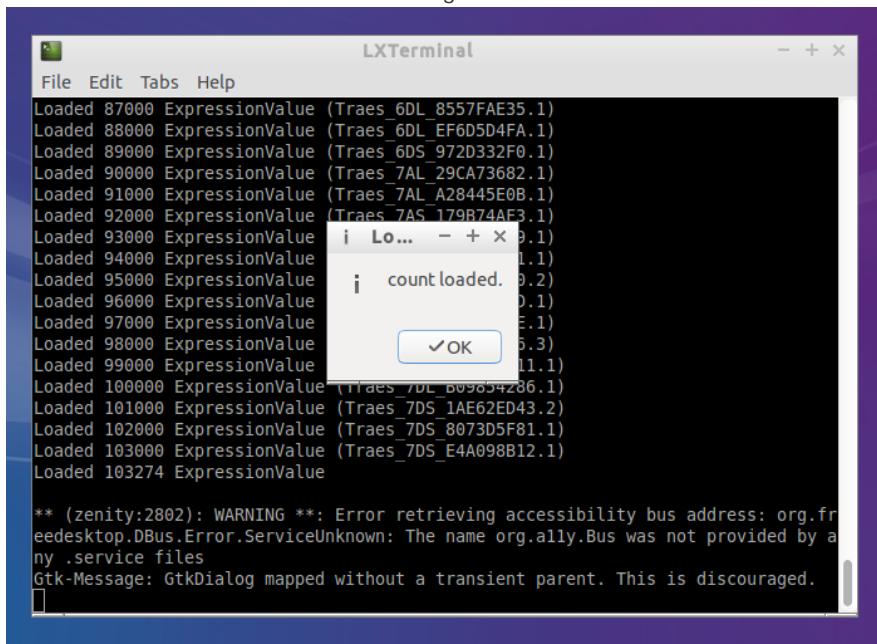
4. The names of the gene set must be the same used when loading the metadata



5. Select the file with the big table. The process takes some time, so be patient.



6. An alert comes when the data finished loading.



If the accession numbers are not the same as in the metadata the process will fail.

Rake Task

In order to load the data, the task `load_data:values` is provided. For example, to load the tpm, the following command is used.

```
rake "load_data:values[First run,IWGSC2.26,tpm,edited_final_output_tpm.txt]"
```

Running Kallisto

You can load the data directly to the database provided that you generated the `Kallisto`

index on your reference:

```
kallisto index --index=Triticum_aestivum.IWGSC2.26.cdna.all.fa.kallisto.k31 Triticum_aest
```

You can modify the index options as you find it suitable for your experiment.

To run Kallisto on single sample, the following task is available:

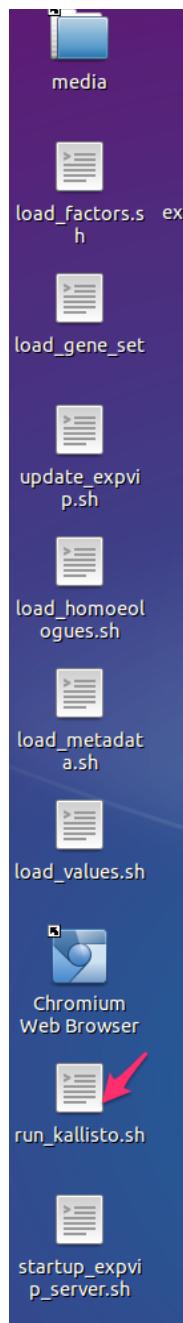
```
rake kallisto:runAndStorePaired[Index, folder/with/samples/ACCESSION, experiment_title, IWGS
```

The task requires that the reads are in a folder named exactly as the `secondary\study\accession*` column in the metadata file. If the accession doesn't exist, the task will fail. `experiment_title` is a name to group alignments.

Run Kallisto on a single sample

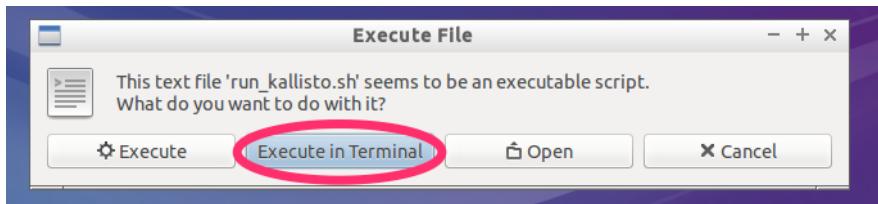
expVIP can run `Kallisto` and load the `tpm` and `counts` to the database. The only requirement is to run `kallisto index` on the transcriptome reference.

Graphical interface

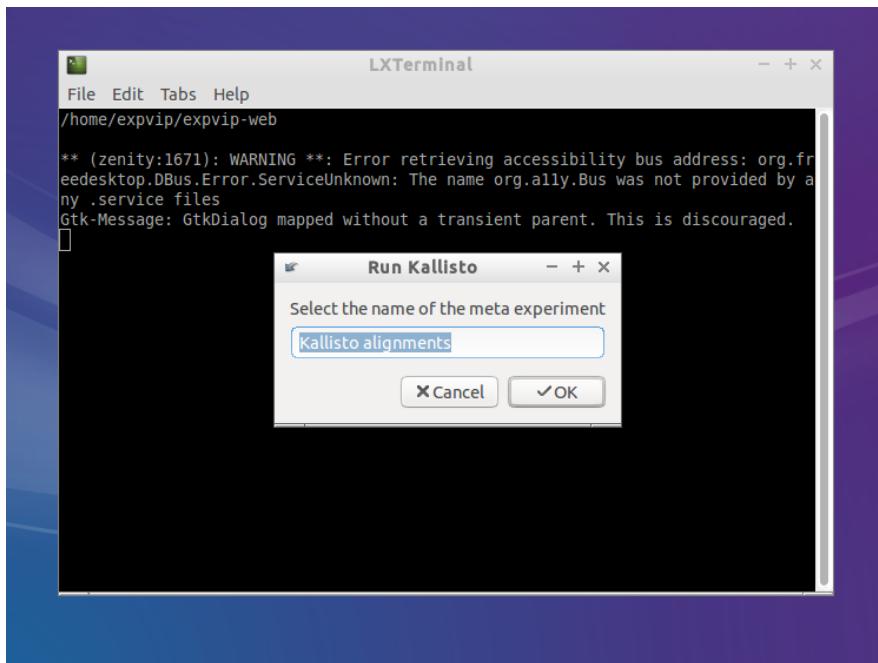


1. Double click on run_kallisto.sh

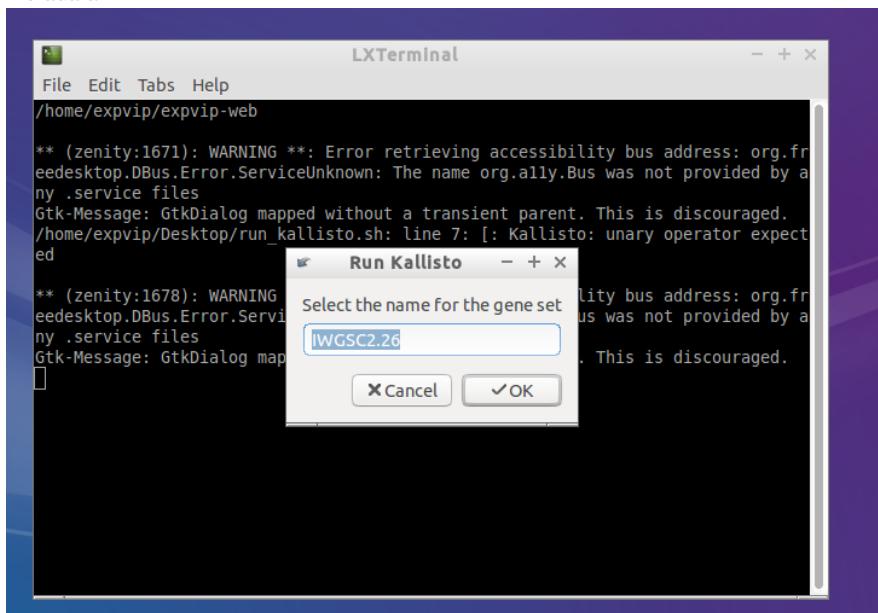
2. Click on Execute on terminal



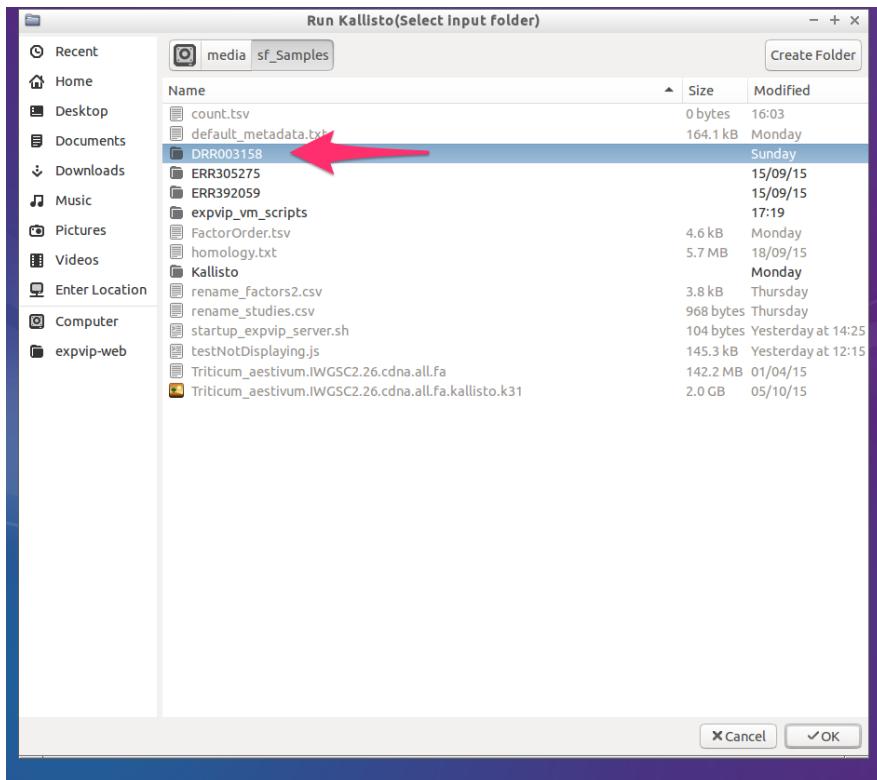
3. Give a name to the set of mappings to be grouped. All mappings done with the same reference and preference should have the same name.



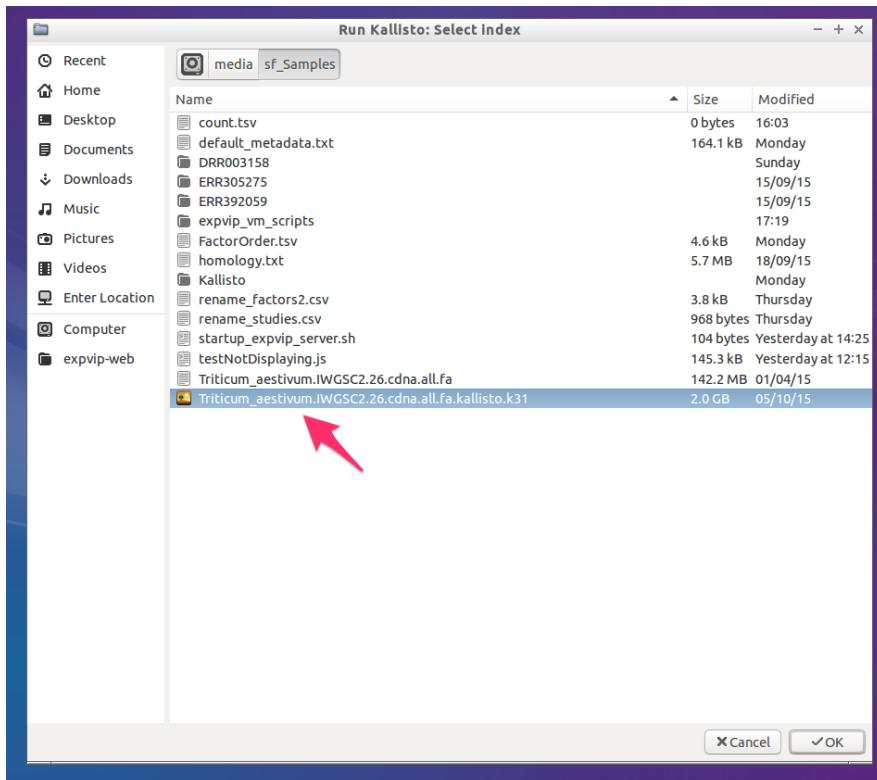
- Get the name of the reference. This name must be the same used when loading the metadata



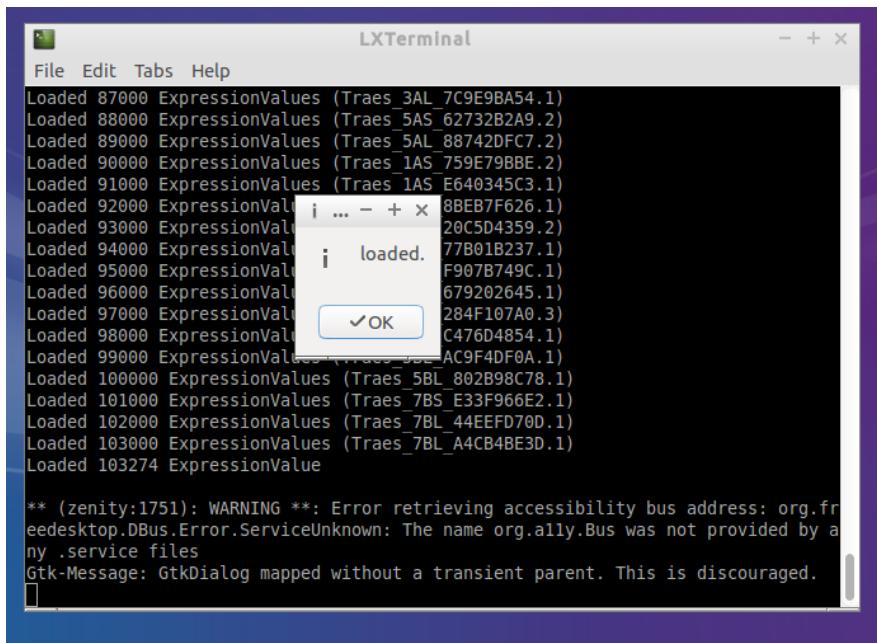
- Select a folder with the reads. The reads must be paired reads. The folder name must be the same as the accession used on the metadata.



6. Select the kallisto index



7. Wait for Kallisto to run and load the data



You can repeat this with all the samples or you can use the [batch load](#).

Rake task

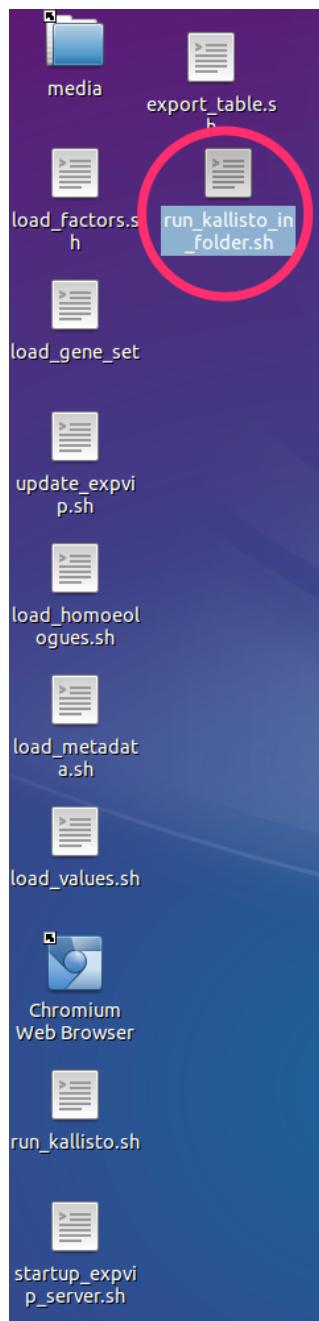
```
kallisto:runAndStorePaired[kallistoIndex,input_folder,metaExperimentName,geneSetName]
```

Where `metaExperimentName` is the name of the group of alignments under the same conditions and `geneSetName` is the name of the reference.

Run Kallisto on a multiple samples

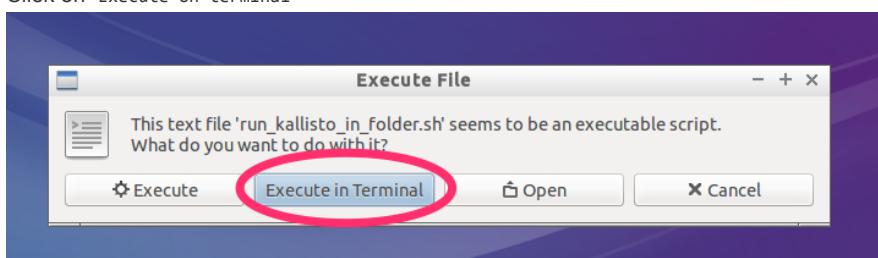
expVIP can run `Kallisto` and load the `tpm` and `counts` to the database from multiple samples. The only requirement is to run `kallisto index` on the transcriptome reference.

Graphical interface

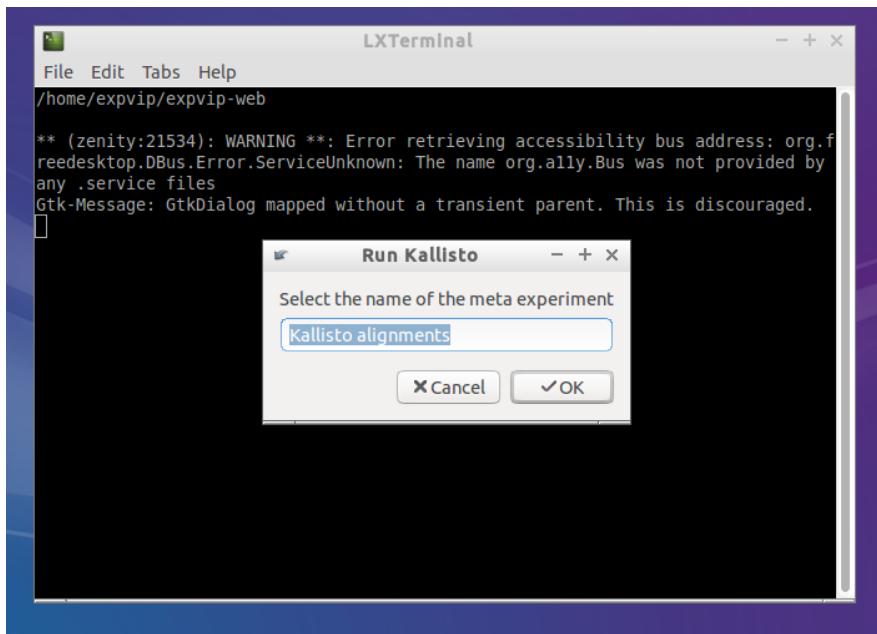


1. Double click on run_kallisto.sh

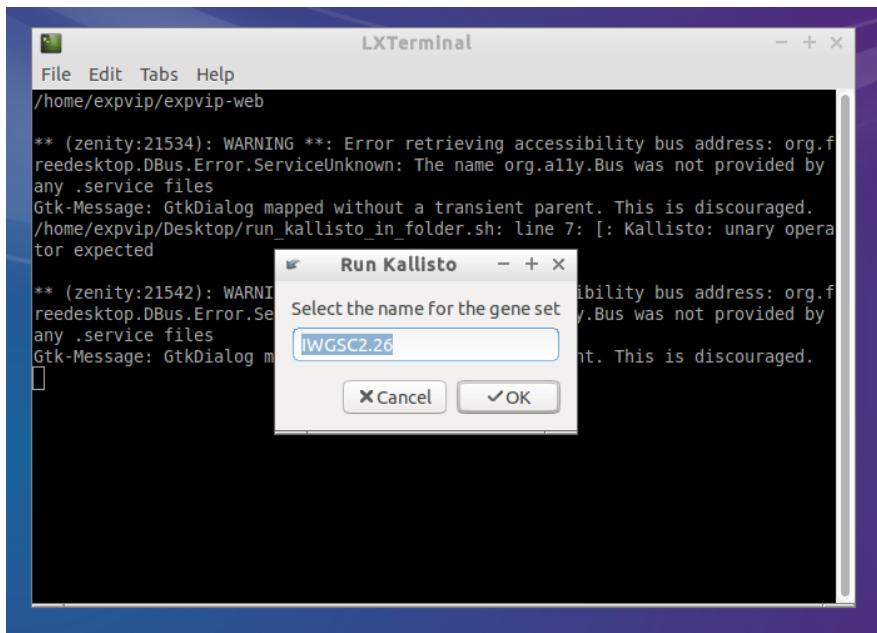
2. Click on Execute on terminal



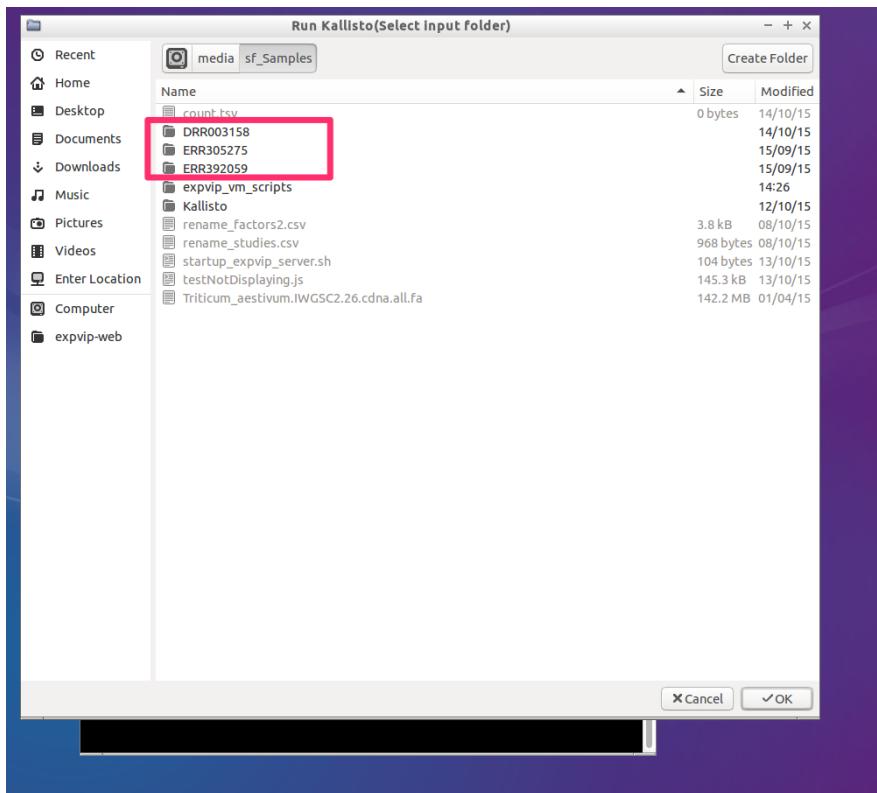
3. Give a name to the set of mappings to be grouped. All mappings done with the same reference and preference should have the same name.



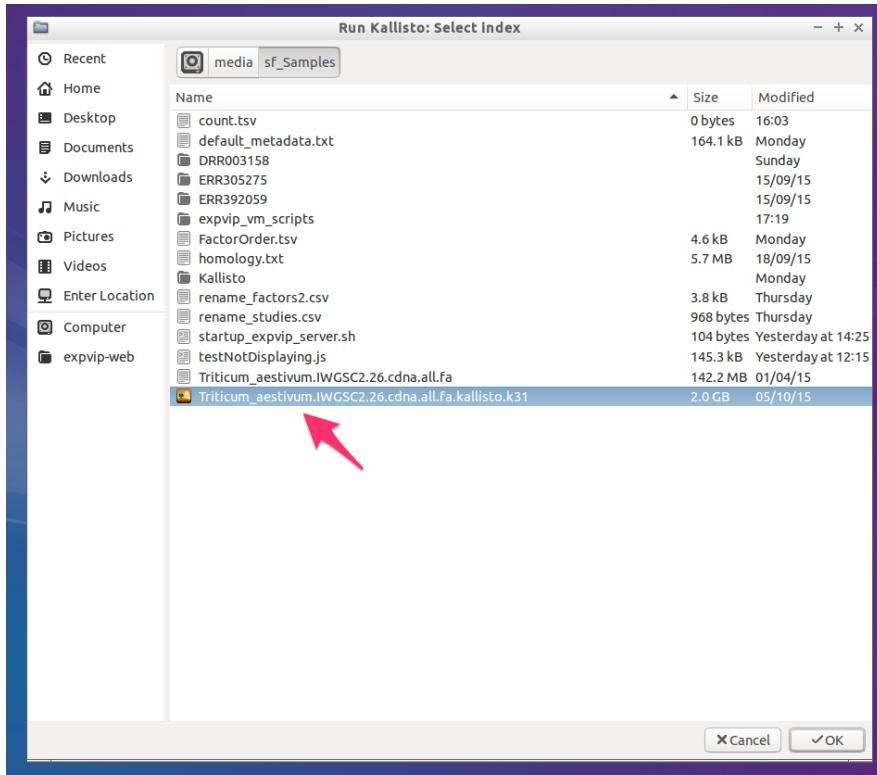
- Get the name of the reference. This name must be the same used when loading the metadata



- Select a folder with the folders containing the reads. The reads must be paired reads. The subfolder names must be the same as the accession used on the metadata. If a subfolder has an experiment that has been loaded already, it is not loaded.



6. Select the kallisto index



7. Wait for Kallisto to run and load the data

```

Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.
/home/expvip/Desktop/run_kallisto_in_folder.sh: line 7: [: Kallisto: unary operator expected

** (zenity:4398): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

** (zenity:4402): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

** (zenity:4411): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

** (zenity:4436): WARNING **: Error retrieving accessibility bus address: org.freedesktop.DBus.Error.ServiceUnknown: The name org.ally.Bus was not provided by any .service files
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.

```

A small modal dialog box is overlaid on the terminal window, displaying the text "i ... - + x" and "loaded." with a blue "OK" button.

Repeat this with all the samples.

Rake task

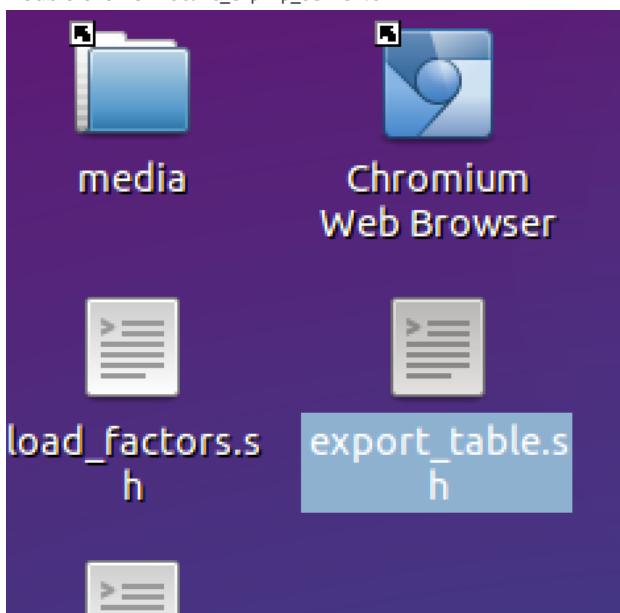
```
kallisto:runAndStorePairedFolder[kallistoIndex,input_folder,metaExperimentName,geneSetNameName]
```

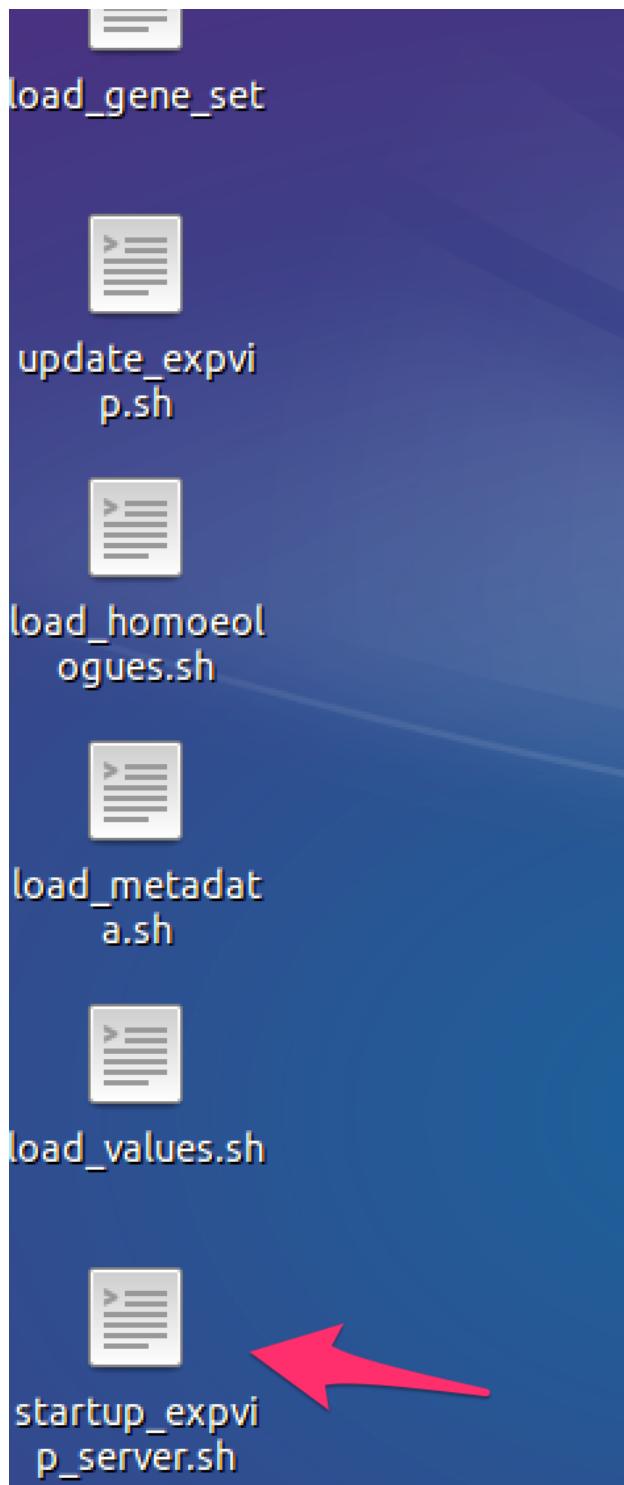
Where `metaExperimentName` is the name of the group of alignments under the same conditions and ``geneSetName`` is the name of the reference.

Starting expVIP web server

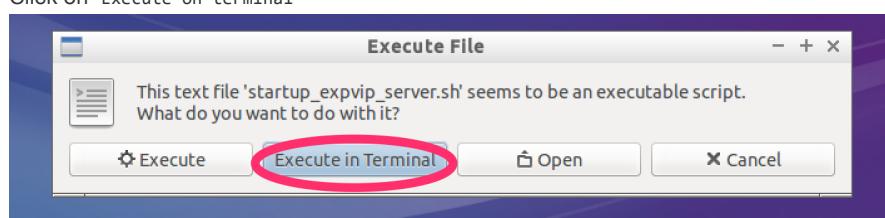
Once the data is loaded, you can visualize the the expression in the expVIP virtual machine.

1. Double click on `start_exvip_server.sh`



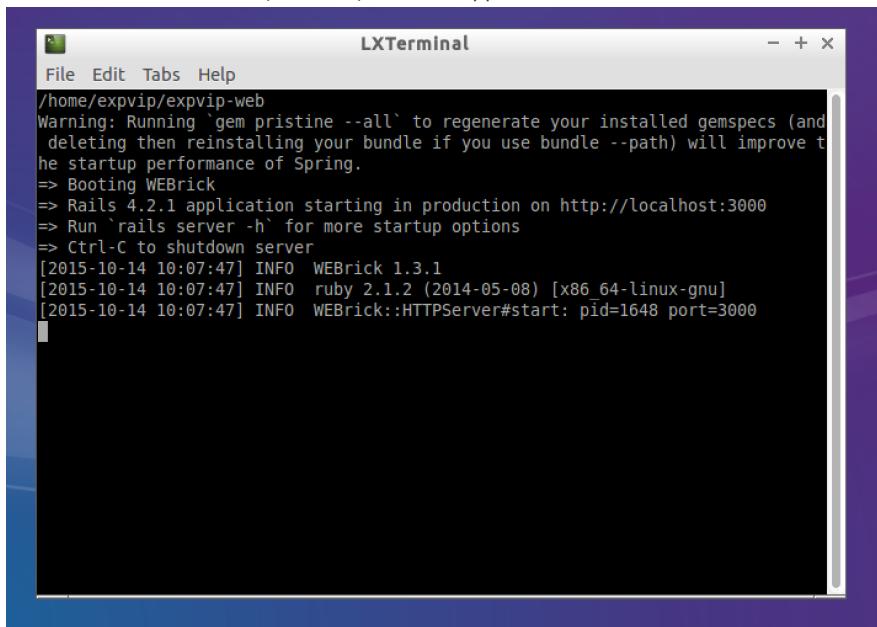


2. Click on Execute on terminal



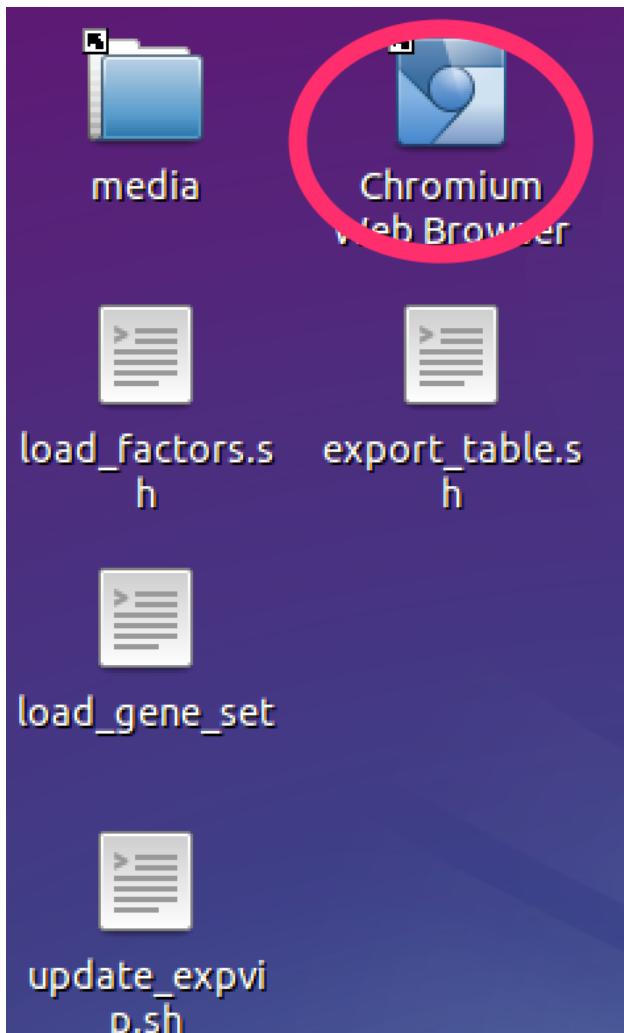
3. Wait for the webserver to start. You know it is ready when the line

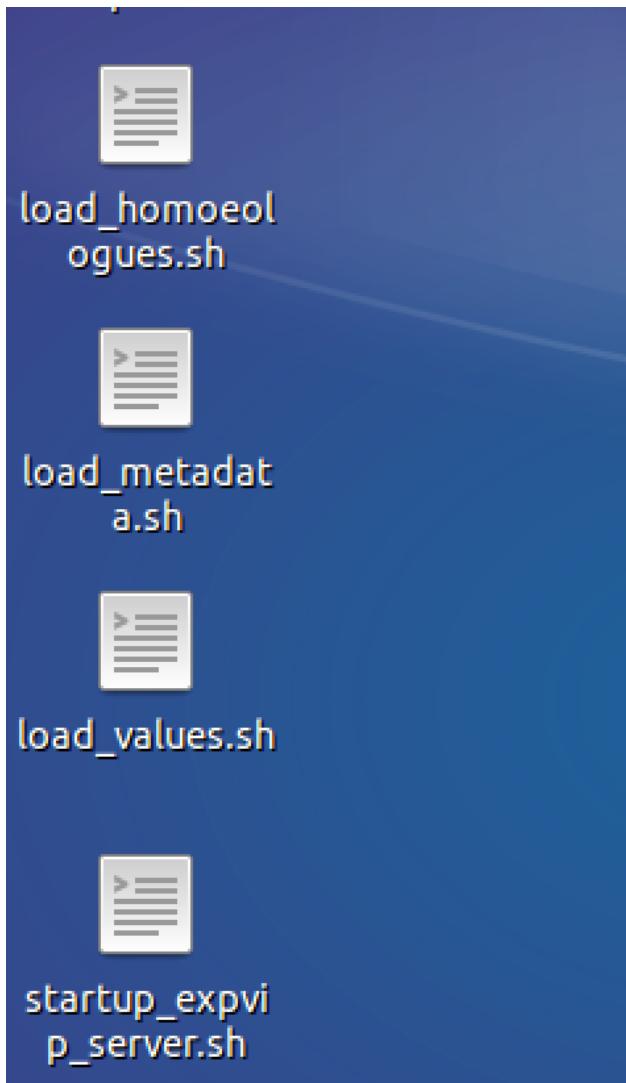
WEBrickHTTPServer#start: pid=xxxx port=3000 appears in the console



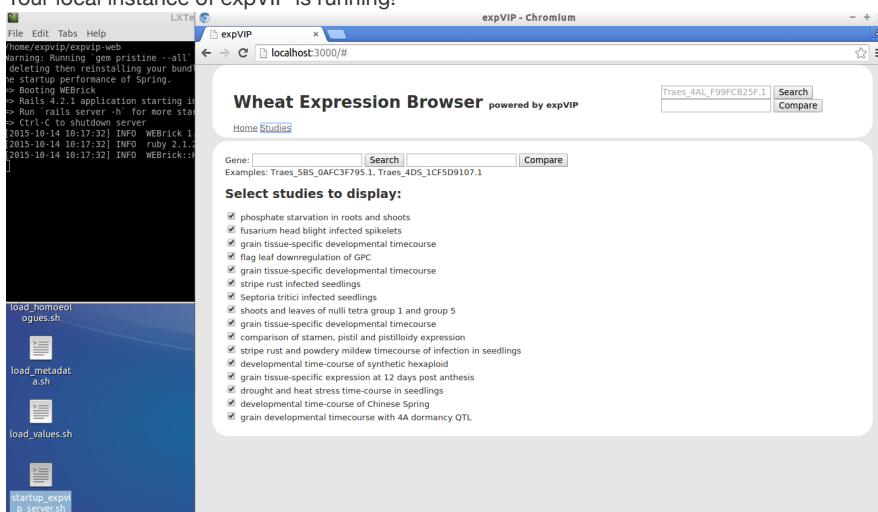
```
LXTerminal
File Edit Tabs Help
/home/expvip/expvip-web
Warning: Running `gem pristine --all` to regenerate your installed gemspecs (and
deleting them reinstalling your bundle if you use bundle --path) will improve t
he startup performance of Spring.
=> Booting WEBrick
=> Rails 4.2.1 application starting in production on http://localhost:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2015-10-14 10:07:47] INFO  WEBrick 1.3.1
[2015-10-14 10:07:47] INFO  ruby 2.1.2 (2014-05-08) [x86_64-linux-gnu]
[2015-10-14 10:07:47] INFO  WEBrick::HTTPServer#start: pid=1648 port=3000
```

4. Double click in Chromium Web Broser.





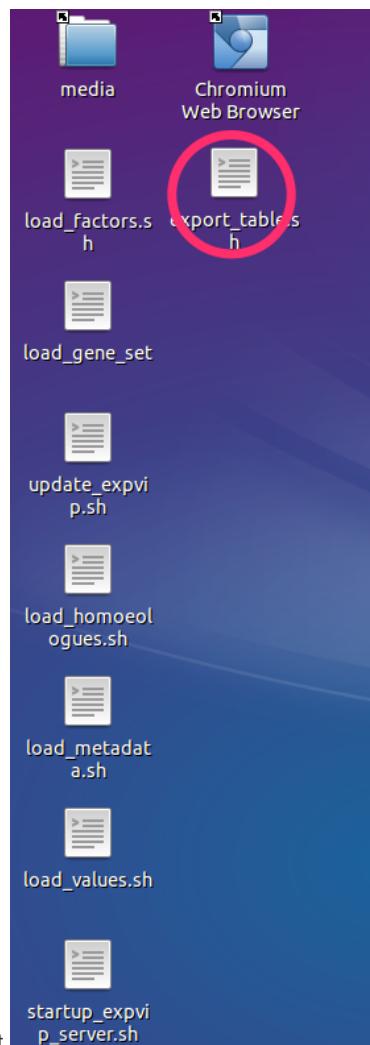
5. Your local instance of expVIP is running!



Export data

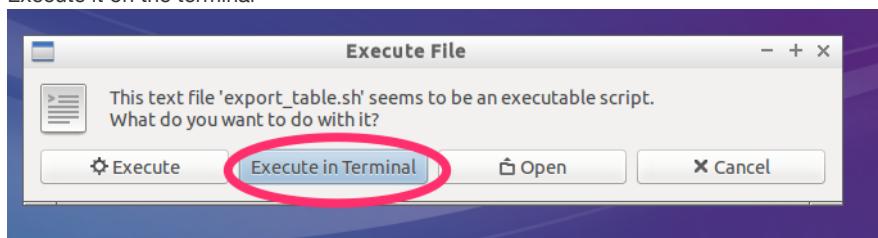
The data loaded in expVIP can be exported to be run in DESeq2, or any other software to do differential expression analysis.

Wizard to export data

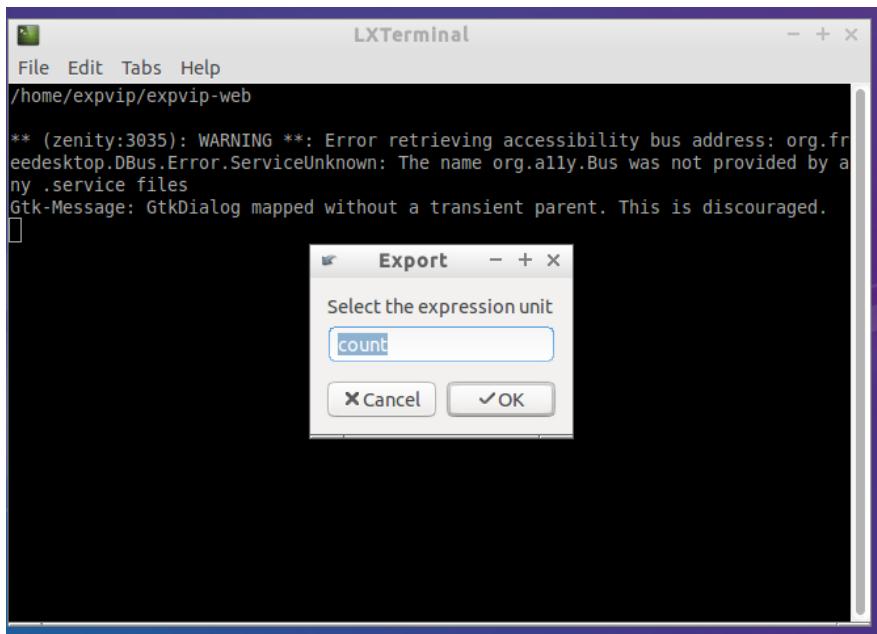


1. Double click on the the `export_tables.sh` script

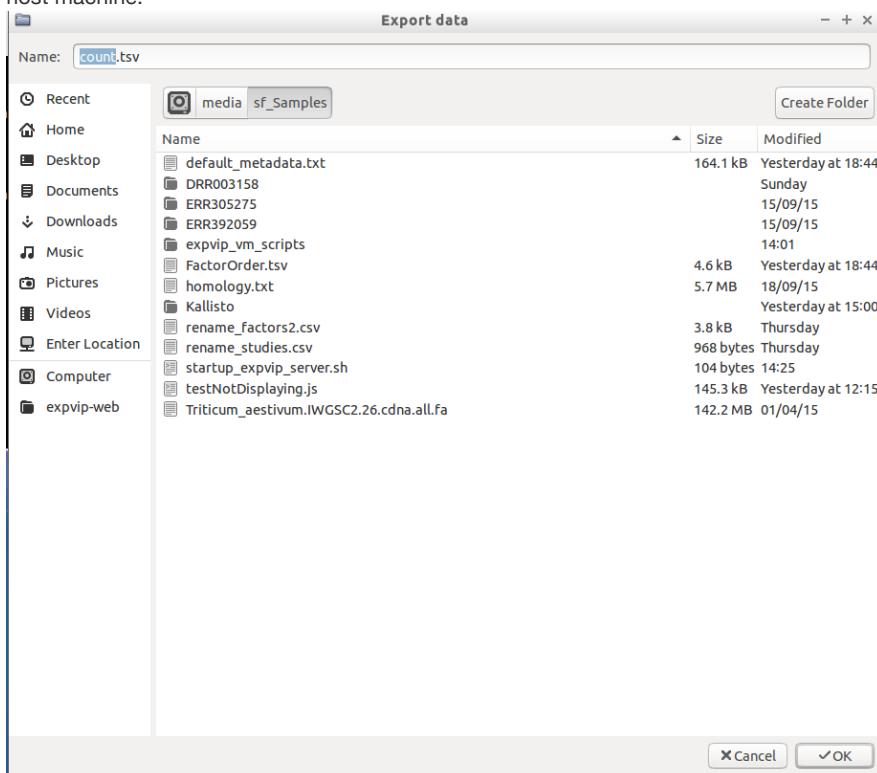
2. Execute it on the terminal



3. Select the value you want to export. If you have used Kallisto , the available options are count and tpm . If you imported the data manually, it will be whatever units you inserted



4. Select a location for the output file. It is suggested to export it in the shared folder with the host machine.



Rake task

```
rake "export:values[tpm, tpm.csv]"
```

Abundance files

To use Sleuth, the abundance files from the `kallisto` runs can be grabbed from the folder `kallisto` inside the folder with reads.

The abundance files for the runs in the VM and in wheat-expression.com can be found in the [here](#).

Troubleshooting

If you get an error like this:

```
ActiveRecord::StatementInvalid: Mysql2::Error: Error writing file '/tmp/MYg0xdqm' (Errcod
```

You can try increasing the size of the virtual machine disk or install expVIP in a dedicated workstation. The best thing to do is to download the precalculated tables from the expVIP website and add the columns with your experiment at the end of the table.

This tutorial is based on the [Wheat Expression Browser](#). However, the principles are the same for any transcriptome study which is powered by the expVIP graphical interface.

Home Page

The home page allows the user to insert a gene name to search and to define which studies are to be included in the visualisation interface. By default all studies are selected, but users can select/deselect a study by simply clicking on the specific button.

You can also compare expression between two genes by introducing both gene names in the boxes and pressing the `Compare` button.

Alternatively you can compare expression across multiple genes (up to 50) to generate a heatmap. You can add a list of genes separate by commas or one gene per line in the `Multiple genes` box.

All gene names are based on the transcriptome reference used for expVIP: for the case of the Wheat Expression Browser we used the IWGSC transcriptome available through [Ensembl Plants](#) release 26.

Visualisation interface

Single gene or two-gene comparison

Once the gene expression loads the page includes several features. These are shown below and explained point by point:

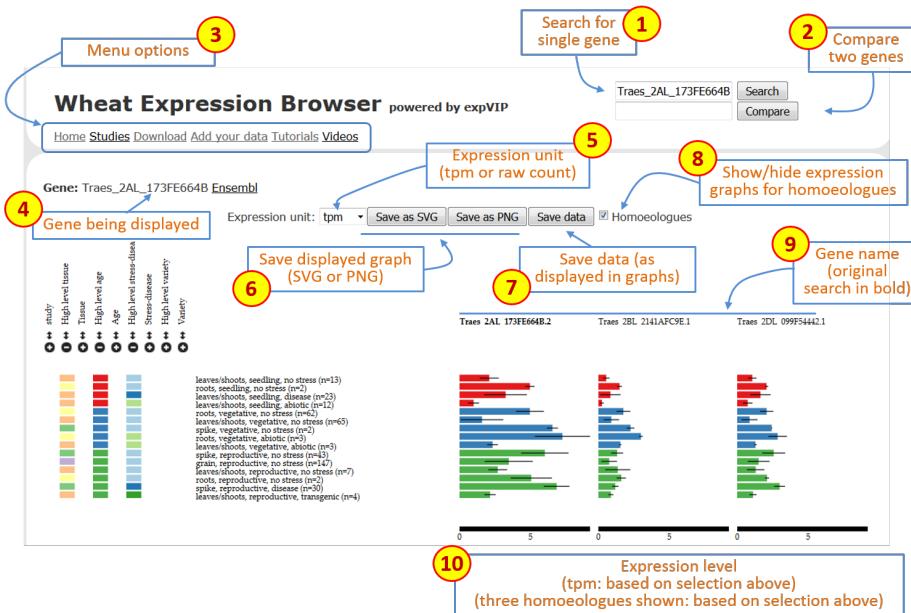


Figure 1: Overall description of features on Wheat Expression Browser

1. **Search box** : at any point you can type or copy a new gene name (based on Ensembl Plants nomenclature) and generate a new set of expression data.
2. **Compare box** : you can type a second gene name and press the **Compare** button to generate two expression graphs drawn at the same scale.
3. **Menu options** : this includes a series of links to different options:
 - **Home** : return to home screen.
 - **Studies** : opens up a popup screen with a summary and short description of each study and a link to manuscript.
 - **Download** : link to download all the wheat expression database including **tpm** and counts and associated metadata.
 - **Add your data** : link to GitHub to download virtual machine.
 - **Tutorials** : link to Wheat Expression Browser Tutorial.
 - **Videos** : link to Wheat Expression Browser Video Tutorial.
4. **Gene** : shows the gene which is currently being displayed with link to Ensembl Plants gene page.
5. **Expression unit** : allows user to select the expression unit used to visualise the expression data. This can be either “transcript per million (**tpm**)” or “estimated counts (**counts**)”. We have not provided RPKM given the inconsistencies generated across samples when using this measure. A detailed discussion can be found in [Wagner et al \(2012\)](#). It is important to mention that **tpm** is preferred over RPKM since it allows an easier comparison for abundances between samples. However it is important to stress that while **tpm** serves as a relative measure to compare genes across experiments, a proper normalisation and statistical analysis with differential gene expression programs must be performed. **expVIP** generates outputs which allow easy implementation of **sleuth** , **DESeq** and **EdgeR** .
6. **Save graph** : these two buttons allow users to save the current graphs in either **SVG** (to work on Adobe Illustrator) or as **PNG** files. The graphical file will render based on the current selection and order of factors as displayed on the screen.

7. **Save data** : this allows the user to download a `csv` file with the data based on the current selection and order of factors as displayed on the screen. The data will include the standard errors and the number of samples that make up each value.
8. **Homoeologues** : by clicking on this button, the Wheat Expression Browser will display the expression graphs of known homoeologues of the original primary gene. This gene name will remain in bold and the homoeologous graphs will be displayed according to A, B, D genome ordering. When homoeologues are displayed the same expression scale is used across graphs and the sorting and filtering of factors is simultaneous to allow easier comparison.
9. **Gene names** : gene name for corresponding graph. When homoeologues are shown the original gene used for the search is shown in bold.
10. **Expression level** : the expression level adjusts according to the expression of each set of gene homoeologues. The scale remains consistent across homoeologues to allow easier comparison. The values are based on the unit selected in the expression unit box (see point 5 above).

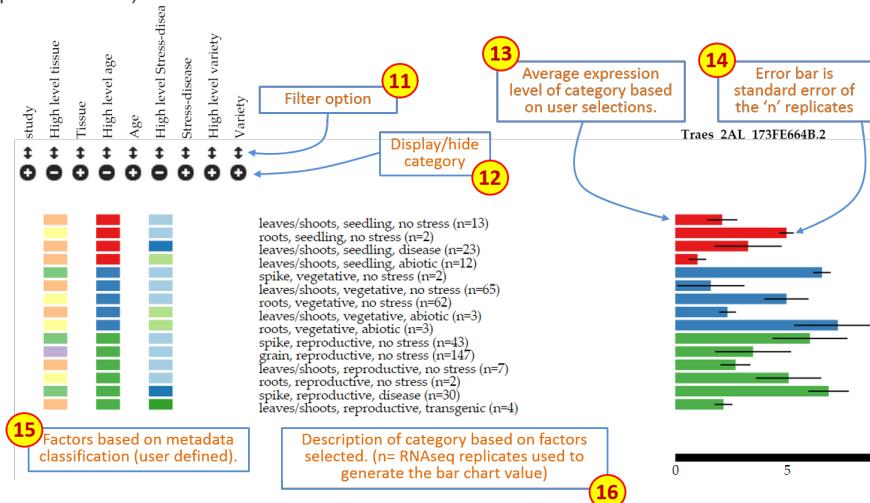


Figure 2: Overall description of features on Wheat Expression Browser (continued)

11. **Filter** : This feature open a pop-up window which reveals all the levels within the particular category. All levels are pre-selected, but users can choose to display specific levels by selecting or deselecting them accordingly. If a level is deselected, then the data associated with this factor is removed from the graph. Within the pop-up window levels can also be re-arranged according to the user's preference by dragging the level to the specific position within the pop-up window (see Features section below).
12. **Display/hide category** : Each individual category can be displayed or hidden by pressing the +/- button. When a category is displayed, the expression graphs will re-arrange according to the new category which has been introduced. If a category is hidden, then the graphs will also adjust accordingly. Data is not removed when doing this, rather it is grouped within the categories selected such that the total samples displayed remains the same. The colours within the category correspond to unique values or levels (up to 24 different colours) and are also used in the bar graphs corresponding to the expression data.
13. **Expression bars** : These bars represent the expression level of the "n" samples which are grouped according to the factors chosen based on the selection criteria (11 and 12 above). When hovering over the bar with the mouse a small tooltip will indicate the expression level (`tpm` or `counts`) and the standard error (`sem`) used for the error bars (see 14).
14. **Error bars** : Standard error of the means for the "n" expression values on which the bar graph is based.

15. **Factors** : Coloured rectangles represent the categories which are displayed according to the factors chosen based on the selection criteria (11 and 12 above). When hovering above the rectangles a tooltip will appear to show the long name of the level being examined.

16. **Description** : Text description of the factors chosen based on the selection criteria (11 and 12 above) and the number of RNAseq samples (n) which meet this specific criterion.

Multiple gene comparisons

17. **Expression unit** : For heatmaps, log2(tpm) is suggested as the expression unit as this provides better resolution to compare multiple genes across several categories.

18. **Heatmap** : Expression data is represented as a heatmap. As for single genes, categories can be sorted and filtered using the same tools. Gene names appear on the top of each column. Currently, up to 50 genes can be visualised in one heatmap. In Figure 3, for example, the two right-most genes are expressed solely in grains, with one being expressed to higher levels as suggested by the dark blue colour.

19. **Scale** : Colour scale for the expression values in the heatmap. The values adjust according to the highest tpm value being displayed within the current heatmap visualisation. Since tpm values below 2 are considered as very low expressed genes and log2 values of tpm<1 result in negative expression values, we forced tpm values below 1 to have a log2 value of zero (i.e. $\log_2(<1)=0$).

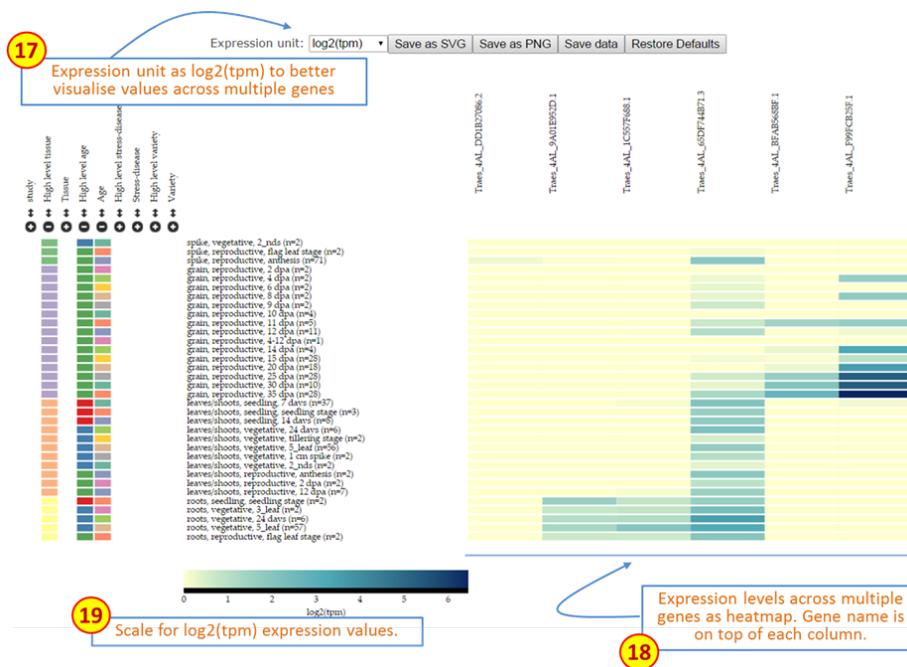


Figure 3: Description of features on Wheat Expression Browser using Multiple gene comparisons.

Features

Sorting

Factors can be sorted within each category in two ways.

1. The first is by simply clicking the mouse on top of the coloured rectangles underneath the

heading. For example in Figure 2 samples are sorted on High level age from seedling (red), vegetative (blue) to reproductive (green). If the user clicks on any of the coloured rectangles in the High level tissue category, then the graph is automatically reorganised based on this factor. In this case it includes four categories as defined by the user in the metadata and the bar graphs on the right hand side change colour according to the latest factor used for sorting. The previous factor used (in this case high level age) remains as a secondary sorting factor (Figure 3).

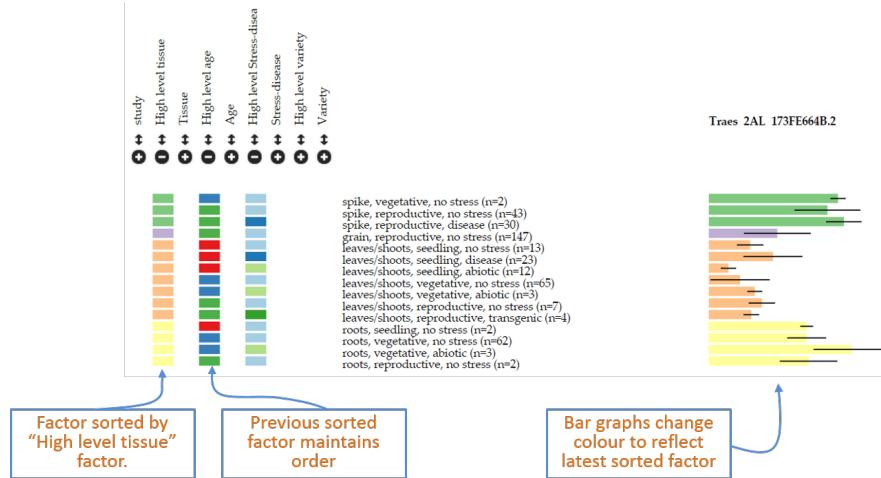


Figure 4: Example of new sorting of data based on clicking of rectangles within “high level tissue”.

2. Alternatively, the user can define the exact order of factors within the browser interface. To do so the filter option (point 11 above) can be used. By clicking on the double arrow button the user opens a pop-up window which shows the levels within the factor. In this example by pressing the double-arrow underneath high level tissue a pop-up with four levels appears based on the order as determined in the user defined metadata (spike , grain , leaves/shoots , roots). To rearrange this, the user can simply click, hold and drag the level to the desired position. This will automatically re-arrange the data based on the new order and the corresponding graph and legends will follow suit. The bottom panel of Figure 5 shows a new order of roots , leaves/shoots , spike and grain .

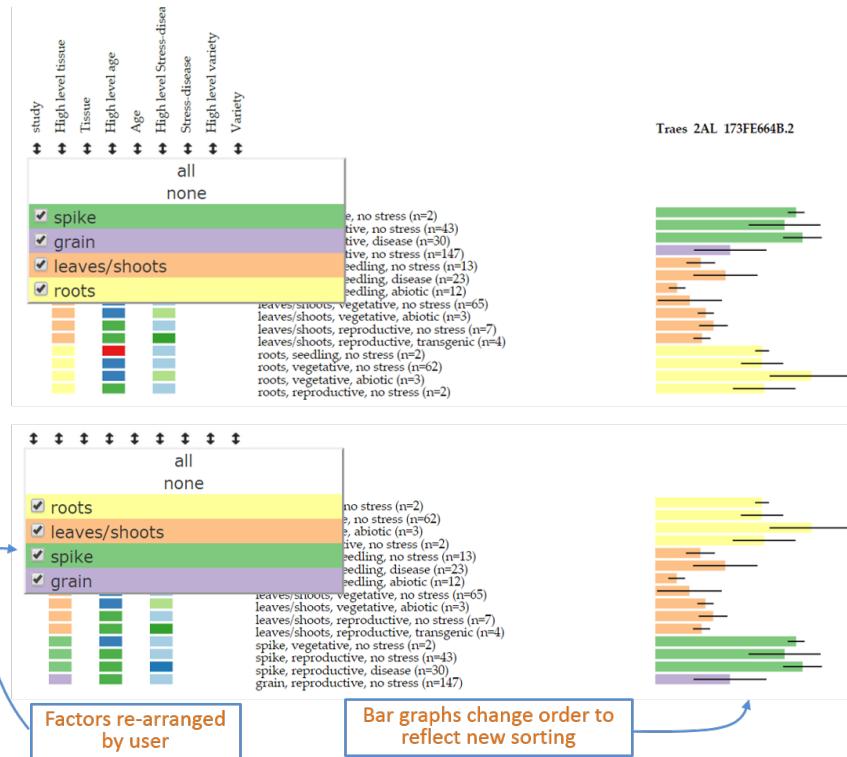


Figure 5: Example of sorting of data based on new user defined order within the filter pop-up window.

Filtering

In cases it may be required to remove certain samples from the visualisation. Note that displaying or hiding a category (point 12 above) does not remove the underlying data from the visualisation: this just simply groups the data within the selected category. Therefore to remove samples from the visualisation the user can open the filter pop-up as described for the **Sorting** option. Individual levels within the category can then be removed by using the “check-box” on the left hand side of the level name. By de-selecting a given level (in the example for Figure 6 we have deselected `leaves/shoots` and `spike`), samples defined as such will be removed from the analysis and will not be shown in the bar graphs. In Figure 6 now only two levels remain (`roots` and `grains`) and hence the bar graphs only show these two levels. Notice that the numbers of samples which comprise each bar graph are the same as those on Figure 5. The pop-up window also includes an `all` and `none` option to rapidly select/deselect individual samples. The filtering option can be used on any factor: for example to remove a complete study from the analysis the easiest way is to select the `study` filtering pop-up on the far left and deselect the study in question.

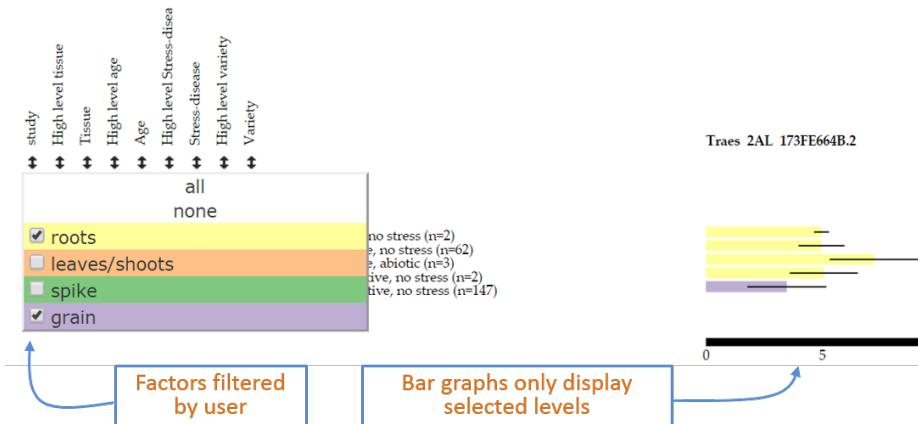


Figure 6: Example of filtering data based on user defined selection within the filter pop-up window.

