Master Project
Lab book

Cyril Matthey-Doret

July 9, 2017

Chapter 1

Introduction

The goal of this project is to identify the locus or loci responsible for sex determination in the parasitoid wasp *Lysiphlebus fabarum*. The species is known to have CSD, which means there are one or more loci that need to be heterozygous in order to triggger female development. I will use the offspring of asexual (thelytokous) females to identify these loci. The offspring consists of haploid males, diploid males and diploid females. Using DNA sequencing data, I will exclude haploid males as they do not provide information on homozygosity/heterozygosity and compare only diploid males versus females to identify the CSD locus/loci. *Lysiphlebus fabarum* wasp specimens (generation F4) issued from thelytokous mothers (generation F3) are used. These individuals come from crossing experiments in a strongly inbred line. Here, we use restriction site associated DNA sequencing(RAD-seq) with a custom pipeline to locate the locus/loci. Samples were single-end sequenced using a ddRAD protocole and digested with ecoRI and mseI. There are 2 separate libraries with two different illumina adaptors (detailed in next section). Statistics for the raw RAD-seq data of both libraries of F4 individuals are shown in table **??**. I also reused sequencing data for the mothers (F3) from Casper, however I don't have the raw reads statistics as they were already processed.

Data summary:		
library	lib7	lib7b
raw reads	163,506,603	133,574,055
containing adaptors	23.25%	45.84%
fragment size	302bp	302bp
mean quality score	34.88	35.05
>= Q30 bases	92.62%	92.13%

Table 1.1: Summary statistics of raw reads from the 2 libraries of F4 individuals RAD-seq data.

This lab book will document my progression for mapping the CSD locus, which can be broken down into 4 main steps: preprocessing of sequencing data, STACKs analysis for RAD-seq data, association mapping and orthology search. These steps are likely to change and throughout the project and it is likely that more will be added. The general process is described visually in figure **??**.

Figure 1.1: **Pipeline described in the current lab book**. Diamonds represent data, rectangles represent operations/programs. Operations/programs in blue are included in the makefile, thos in red are not.

Chapter 2

Processing reads

RAD-seq data was split into 2 separate libraries: 7 and 7b. Together, the libraries contain 173 F4 individuals from 11 different F3 mothers. There were 96 samples in library 7 (one of which was contaminated) and 77 in library 7b. In total we have 172 valid samples across 11 families. Additionally, I inherited from 28 individuals from another library (lib6) which are offspring from families A and B.

2.1 Quality control

fastqc was used for quality control, separately on each file, and on all files together in the library.

2.2 Demultiplexing

The process-radtags program from stacks was used for demultiplexing and removal of Illumina adaptors. The operation was performed separately for libraries 7 and 7b:

```
$ process_radtags −p raw/ −o processed/ −b \
/barcodes −e ecoRI −−filter_illumina −E phred33 \
 −r −c −q −−adapter_1 adapter −−adapter_mm 2
```

lib7	Truseq adapter, index 6 [1]
lib7b	TruSeq adapter, index 12 [2]

2.3 Trimming adaptors

This step is performed by process radtags at the same time as demultiplexing. I tried different values for adapter mismatches, between 0 and 3 (i.e. reads containing sequences distant from the adapter by n mismatches are removed). This did not cause any major difference and therefore I will perform all downnstream analyses with an allowance of 2 mismatches in the adapter.

Note: Demultiplexing did not yield any read for CF4F10.

---

[1]GATCGGAAGAGCACACGTCTGAACTCCAGTCACGCCAATATCTCGTATGCCGTCTTCTGCTTG
[2]GATCGGAAGAGCACACGTCTGAACTCCAGTCACCTTGTAATCTCGTATGCCGTCTTCTGCTTG

Chapter 3

STACKS pipeline

The pipeline described in this chapter will map the processed reads to the reference genome and build a catalogue of loci. It will eventually implement additional features such as calculating population statistics. At each step of the pipeline, I will try different combinations of parameters and choose the one yielding the best results.

3.1 Mapping

Since the reference genome of *Lysiphlebus fabarum* was recently released, I will first map the sequencing reads to the reference, using BWA, but I may also want to use bowtie to compare the results, eventually. At the moment, a draft reference genome is available, table **??** displays some summary statistics of it.

3.1.1 BWA

BWA provides 3 different algorithms: MEM, backtrack (aln) and SW. MEM is normally the better for Illumina reads longer than 70bp, therefore I will be using this one here. Backtrack is preferred for short reads and SW with frequent gaps. There are also 2 algorithms for building the index: 'is' and 'bwtsw'. 'is' is used with reference ¡2GB, 'bwtsw' with larger references.

General commands:

```
$ bwa index −p <out_index_name> −a <algorithm>
$ bwa mem <index> <sample.fq> > <out.sai>
> $sample−$prefix.sam
```

When using bwa-aln, there the command running the alignment (after indexing) is:

```
$ bwa aln −n <mismatches> $index <sample.fq> > <sample.sai>
$ bwa samse −n <max_dupl> $index $sample.sai $data_dir/$sample.fq.gz
```

The first command (bwa index) constructs an index from the reference genome, whereas the second one (bwa mem/aln) actually runs the aligner. The third command (bwa samse) allows to transform the .sai into .sam files.

Statistics	Values
Assembly length (Mbp)	140.7
Largest contig (Mbp)	2.2
Mean contig size (kbp)	82.9
Median contig size (kbp)	30.4
N50 (kbp)	216.1
Number of contigs	1698

Table 3.1: Assembly statistics for the current reference genome of *Lysiphlebus fabarum*

Here is the list of different mapping parameters I may try with bwa-mem:

- -k : minimum seed length (will miss matches shorter than value)[19]

- -w : band width (gaps longer than value will not be found)[100]

- -d : maximum distance between query and reference positions before stopping seed extension. [100]

- -r : triggers reseeding for a MEM longer than min_seed_len*float. Larger values yield fewer seeds $->$ faster alignment but lower accuracy [1.5].

In the case we use the backtrack (aln) algorithm instead of MEM, the only parameter worth tuning is -n, the number of mismatches allowed.

Note: I did not include parameters that are not relevant to sensitivity (e.g. threads) or parameters that involve scoring (changing these naively would probably have a negative impact). Full list of parameters is on the official bwa website

Note2: About multiple hits and BWA-mem:

(https://github.com/lh3/bwa)

2. Why does a read appear multiple times in the output SAM?

BWA-SW and BWA-MEM perform local alignments. If there is a translocation, a gene fusion or a long deletion, a read bridging the break point may have two hits, occupying two lines in the SAM output. With the default setting of BWA-MEM, one and only one line is primary and is soft clipped; other lines are tagged with 0x800 SAM flag (supplementary alignment) and are hard clipped.

3.1.2 Bowtie2

General commands:

```
$ bowtie2−build reference_genome.fa L_fabarum
$ bowtie2 −x <ref_index> −U <unpaired_reads_files> −S <out_SAM_file>
```

The first command (bowtie2-build) constructs a set of index (extension: .bt2) from the reference genome, whereas the second one actually runs the aligner.

Here is the list of different mapping parameters I may try:

- --trim5 ¡n¿ : trim n bases from the 5' end (left) of each read before alignment

- --trim3 ¡n¿ : trim n bases from the 3' end (right) of each read before alignment

- -D : maximum number of seed extension that can fail in a row before stopping (increasing makes bt2 slower)

- -R : maximum number of re-seeding when attempting to align read with repetitive seeds (increasing makes bt2 slower)

- -N : number of mismatches permitted per seed (increasing reduces false negative, but makes bt2 slower)

- -L : length of seeds (decreasing makes bt2 slower but more sensitive)

- -i : interval between seeds (increasing makes bt2 slower but more sensitive)

Recommandations from the bowtie2 website to make alignment more sensitive: a) make seeds closer (reduce i) b) make seeds shorted (reduce L) c) allow more mismatches per seed

End-to-end versus local alignment (--local): end to end takes all bases in the reads into account, while local allows to trim reads to exclude the ends from the alignment. Seeds are substring of the reads which bowtie2 tries to align to narrow down the valid regions for aligning a read. There is a list of preset values for these parameters on the official bowtie2 website. Preset parameters differ between local and end-to-end modes. There are other parameters, such as score weights for gaps mismatches and allowance for 'N' ambiguous characters but changing these naively could probably have negative effect on the alignments. Procedure: try out different preset values and select the one yielding the best results. Then, eventually tweak the parameters slightly from the preset values. These simulations can be run on a subset of individuals to speed up the process.

note to self: at the end of a run, bowtie2 prints a summary to stderr such as:

20000 reads; of these:
20000 (100.00%) were unpaired; of these:
1247 (6.24%) aligned 0 times
18739 (93.69%) aligned exactly 1 time
14 (0.07%) aligned >1 times
93.77% overall alignment rate

If I use Bowtie2, I will redirect the stderr and parse it into a csv file and generate plot to visually estimate the best parameter values.

3.1.3 Mapping results

I used the aln algorith of bwa, since the mem algorithm did not report multiple alignments and has very few documentation available. I tried different mismatches values with aln (Figure **??**), ranging from 0 to 8, and I chose to use 4 since the increase in single was quite low above this value. When the mismatch parameter is set to 4, running bwa-aln on a subset of 12 samples yielded 57% of single hit reads, 26% of multiple hits and 17% of unmapped reads.

Figure 3.1: Results of the BWA mapping with different parameter values for the number of mismatches allowed during mapping. The plot shows the proportion of reads that are mapped uniquely to the reference genome.

3.2 pstacks

pstacks is a component of the STACKS suite that takes stacks of reads aligned to a reference genome as an input (typically in the SAM format) and idenify SNPs
general command:

```
$ pstacks −f <input_path> −i <sample_ID_int> −o <out_dir> \
 −m <min_depth> −p <num_threads> −t <file_type>
```

Parameters I may want to change are:

- -m : minimum depth of coverage required to call a stack [3]

- --max_clippped : alignments with more than X soft-clipped bases are discarded [15%]

- --min_mapq : minimum required quality [10]

min·cov	nloci	mean·cov	sd·cov
1	5020.1	31.8	58.6
2	3659.1	42.5	65.0
3	2747.2	49.9	68.2
4	1505.3	55.6	70.1
5	781.9	60.1	71.3
6	597.9	63.9	72.3

Table 3.2: Summary statistics of stacks obtained with different parameter values for minimum coverage in pstacks.

Note: there are 3 models; SNP, bounded SNP and fixed. SNP is the default model, bounded SNPs allows to give prior expectations about the error rate, which can allow better estimations of heterozygosity and the fixed model identifies all fixed sites and masks all others.

3.2.1   pstacks results

Pstacks was run on aligned reads (BWA, 4 mismatches allowed). I tried different values for the minimum coverage required to call a stack (-m parameter), ranging from 1 to 6 (Figure **??**). Below is the value for minimum coverage, along with the mean number of loci and alleles that was produced per sample. This table was produced including all non-empty samples (198 individuals) and the variables are averaged (arithmetic mean) over all those samples.

I will use 4 reads minimum coverage as this is already high and using lower values does not improve the output in anyway.

Note: Locus are regions formed by one or more stacks. Alleles are different stacks at the same locus.

3.3   cstacks

cstacks is a component of the STACKS suite that builds a catalog of loci with different alleles from a set of processed samples.

general command:

```
$ cstacks −s <sample_prefix> −o <out_dir> −b <catalogue_ID>\
−p <num_threads> −n <num_mm> −M <pop_map>
```

Parameters I may want to change are:

- -n : Number of mismatches allowed between sample loci when building catalogue [1]

- -g : base catalog on alignment position instead of sequence identity

Note: There are also advanced options such as gapped assembly parameters and loci matching multiple catalogue entries, but these are probably not relevant here.

3.3.1   cstacks results

I changed the number of mismatches allowed between samples between 1 and 4. Table **??** shows the results with a subset of 11 samples ($[A - L\&\&[\wedge B]]01$) because B01 was empty (few, low quality reads).

I tried running cstacks on all (non-empty) samples, and also modifying the script so that it will first compute the mean number of tags across all samples, and exclude those with less than 10% of this value when building the catalogue. From the 202 original samples, 4 were empty and excluding low quality ones removed 13 additional ones.

Summary results with n=1:

- All non-empty samples (198): 25618 alleles in catalogue, over 7062 loci.

mismatch	mean˙loci	mean˙alleles
1	2696	7799
2	3129	8464
3	3198	8704
4	3263	8914

Table 3.3: Summary statistics of loci obtained with different parameter values in cstacks.

- Excluding samples with <10% of mean tags (185): 25585 alleles in catalogue, over 7046 loci

note: following recommandations from Paris et al. 2017, Lost in parameter space: a roadmap for STACKS. *Methods in Ecology and Evolution*, I set the value of n to 3 (M-1=3)

3.4   sstacks

sstacks is used to generate one file per individual, in each file, the matching loci point to the cstacks catalogue. There is no crucial parameter to change in this program.

3.5   populations

The "populations" component of STACKS is used to compute population genetics statistics on a set of individuals. I use it here to compute FST statistics (fixation index) along the genome. It offers several features to compute different statistics, including a bootstrapping feature and a "kernel smoothing" flag, allowing to take neighbouring region into account with a decreasing weight as a function of their distance from the focus nucleotide. I will use both of these features to compute FST.

The main features to change in FST calculation are :

- -r: minimum percentage of individuals in a population required to process a locus for that population.

- -p: minimum number of populations a locus must be present in to be procesed.

- -m: minimum stack depth required for individuals at a locus.

Example populations call for FST calculation:

```
$ populations −P <stacks_files> −M <popmap> −b 1
−k −r  0.75  −f  p_value
```

3.5.1   populations results

I ran populations with the following parameters:

- -r: 0.75 - 0.85

- -p: 2

- -m: 5

Therefore, only loci with at least 5 reads of coverage were included, each loci also needs to be in at least 75% to 85% of all individuals and in both populations (males and females).

This table summarizes the first statistics I extracted from the VCF files (using vcftools) with the different values for r:

r parameter	obs.hom.	exp.hom.	n.sites	inbreed.coef.	mean depth	obs./exp. hom.
75	957.89	895.09	1171.82	0.23	121.09	1.07

plotting the inbreeding coefficient per individual with r=75 yields:

When increasing the minimum depth above 15, populations crashed. Disabling bootsrapping and kernel smoothing fixed the issue. Guess: May be caused by a contig smaller than the sliding window size during kernel smoothing. This could be the case if the sliding window has a min number of loci, in which case increasing minimum depth would cause the window to enlarge.

I tested for correlations between Mean depth of loci and homozygosity in all individuals and per family, no significant correlation was found. This means there should not be significant allele excludion caused by too stringent min depth (i.e. stacks removed because of low coverage, resulting in homozygous loci).

3.6 STACKS parameters summary

process radtags	Mis:2
bwa	Mis:4
Pstacks	MinDep:3
Cstacks	LocMis:1
Sstacks	-
populations	IndProp:0.75, MinDep:5(25)*, MaxHet:0.9

* Stringent MinDepth value used to exclude haploid males. The smaller value is used for downstream analyses to keep more loci.
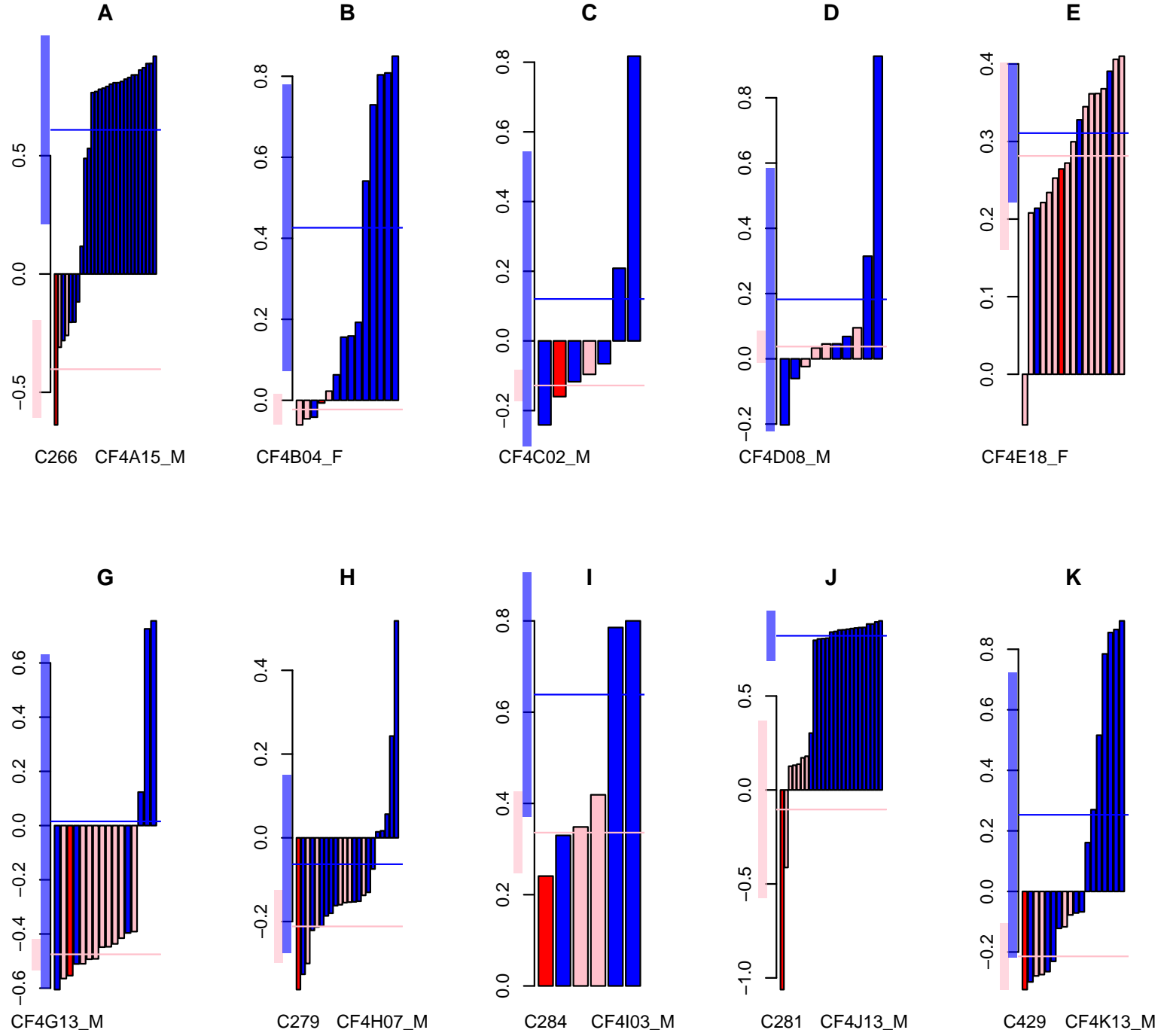
Figure 3.2: Inbreeding coefficient (F). Each plot is a family, each bar is an individual. Blue bars represent males and pink ones represent females and red ones are mothers. color bars on the y axis span the mean +- standard deviation of males and females, respectively. In theory, mothers should have the lowest inbreeding coefficient of their family (highest heterozygosity)

Chapter 4

Excluding haploids

I used the F statistics (coefficient of inbreeding) computed by vcftools to measure homozygosity levels in each individual. The F statistics used were generated with more stringent parameters (min depth: 25) to call ploidy more confidently. It is calculated as $F = \frac{O-E}{N-E}$ where O is the observed number of homozygous loci, E is the number expected by chance and N is the total number of loci.

First, I calculated the Mean μ and standard deviation σ of the inbreeding coefficient F among daughters in each family. I then classified males with:

$$F > \mu + 2\sigma$$

as haploid. This is stringent threshold should exclude all haploid males, but it can also potentially exclude diploid males from the analysis.

Update: 08.06.2017

I tried 12 different thresholds for excluding haploids. The thresholds were computed according to these rules:

- 1: All thresholds follow the formula $\mu + \tau(N\sigma)$ with $1 \leq N \leq 4$

- 2: $1 \leq \mu \leq 4$

- 3: τ is a function for transforming the inbreeding coefficient values. Either $\tau(F) = F^2$ or $\tau(F) = \sqrt{F}$

As shown above, in this report I will use the threshold named "m2" which corresponds to $F > \mu + 2\sigma$. Figure ?? shows a plot showing the separation of haploid and diploids using this threshold.
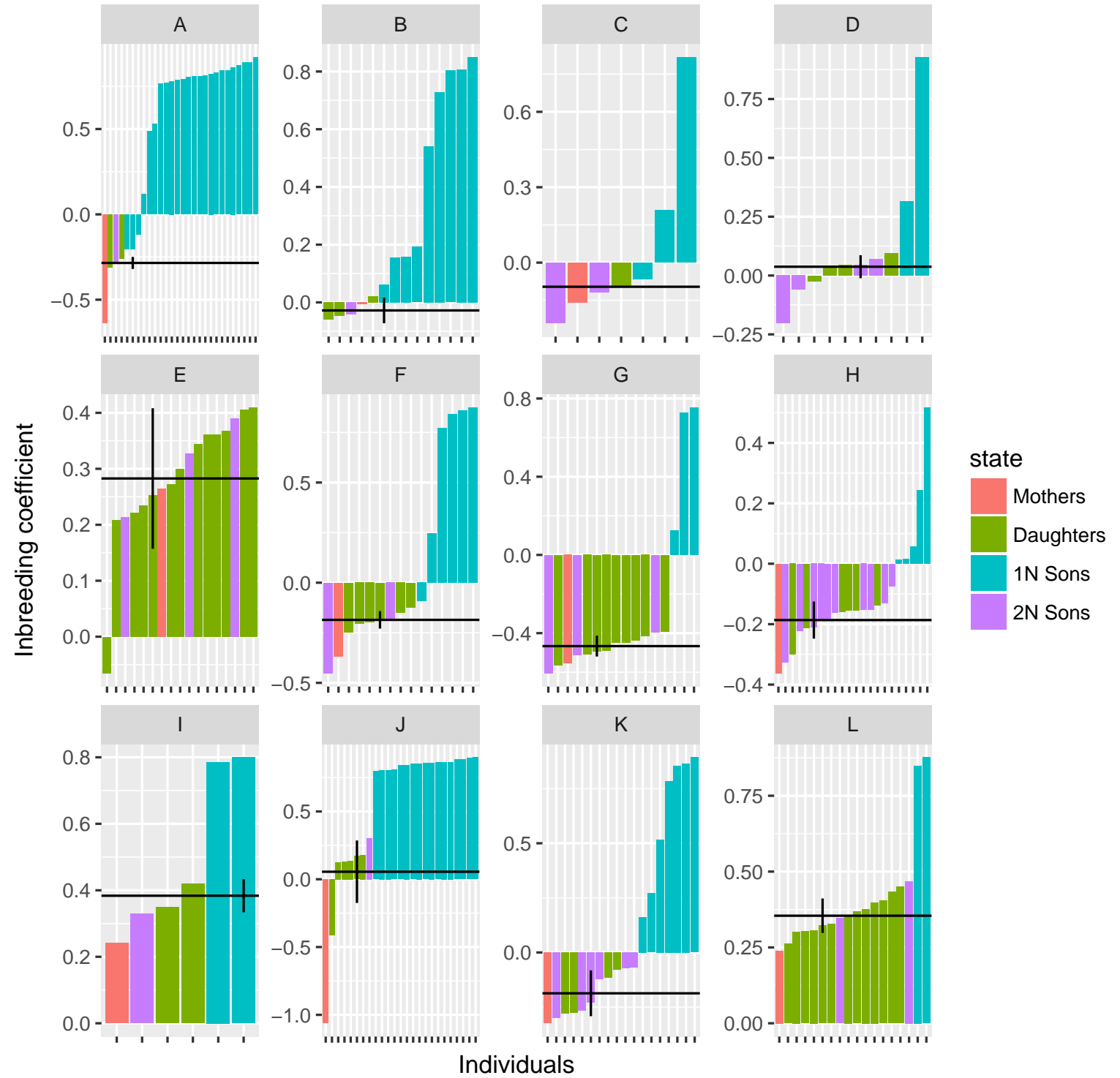
Figure 4.1: Inbreeding coefficient (F). Different colors represent the different types of individuals, as shown in the legend. Each plot is a family, each bar is an individual. Horizontal black bars show the mean inbreeding coefficient of daughters within the family $+/-$ the standard deviation (vertical black line). In theory, mothers should have the lowest inbreeding coefficient of their family (highest heterozygosity)

After separating haploids from diploids, I performed exploratory analyses prior to association mapping. The goal of these is to assess the homozygosity rate of SNPs in diploid males and females and how many seem to fit the CSD pattern. As shown in Figure **??** and **??**, there is no single SNPs thatis heterozygous in all females and homozygous in all diploid males. That can imply that the species has ml-CSD and there is no single locus that always fit the CSD pattern, or the restriction enzyme used in the RAD-seq protocol (ecoR1) did not cut in the CSD locus directly. It is very likely a combination of both explanations. It is also worth mentioning that, besides the number of SNPs, it makes little difference whether I use permissive (Figure **??**) or stringent (Figure **??**) parameters. The only notable change is that including more SNPs results in higher homozygosity for both sexes.

In this analysis, I excluded all consensus loci (where all individuals had the same allele) and did not consider SNPs with more than 2 genotypes in a single individuals in the heterozygosity calculation (counted as missing values), as they are likely due to contaminations.
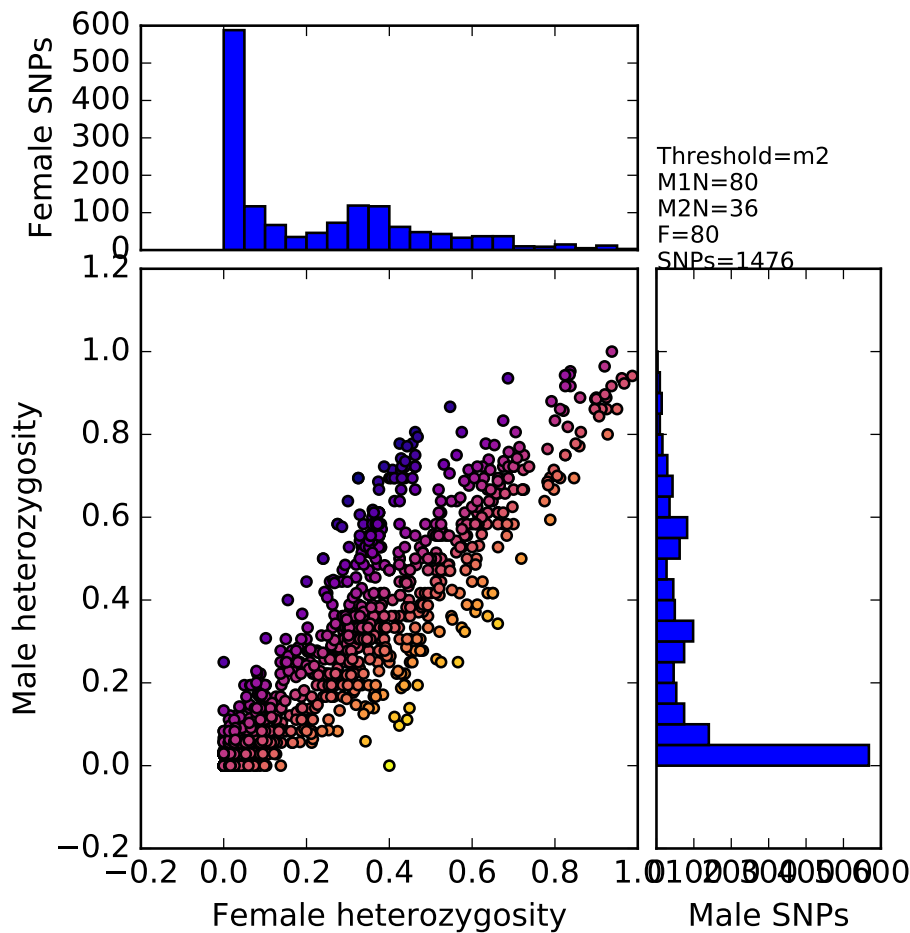

Figure 4.2: Homozygosity of female and diploid male SNPs where depth ≥ 5 (permissive parameters). Each point on the scatterplot is a SNP, and its coordinate are the proportion of females (x) and males (y) in which it is homozygous. Histograms allow to visualize the distribution of homozygosity for SNPs of each sex. The color code shows how SNPs fit the CSD pattern with lighter points being closer to it (i.e. more homozygous in males and heterozygous in females). Summary statistics on the top right show the threshold used, the number of haploid males excluded (M1N) as well as the number of diploid males (M2N), females (F) and SNPs included.
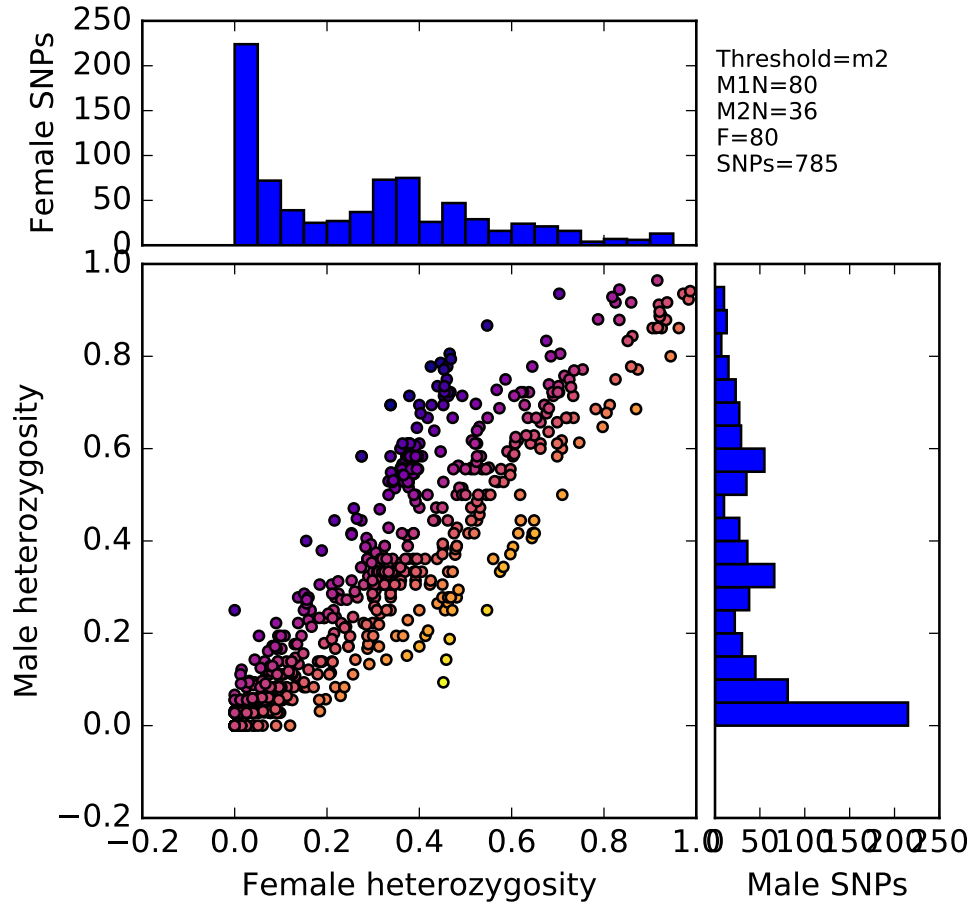
Figure 4.3: Homozygosity of female and diploid male SNPs ≥ 25 (stringent parameters). Each point on the scatterplot is a SNP, and its coordinate are the proportion of females (x) and males (y) in which it is homozygous. Histograms allow to visualize the distribution of homozygosity for SNPs of each sex. The color code shows how SNPs fit the CSD pattern with lighter points being closer to it (i.e. more homozygous in males and heterozygous in females). Summary statistics on the top right show the threshold used, the number of haploid males excluded (M1N) as well as the number of diploid males (M2N), females (F) and SNPs included.

14

Chapter 5

Cleaning genomic data

date : 23.06.2017

The first attempt at separating individuals based on inbreeding coefficient revealed several issues inherent to the data:

- The inbreeding coefficient is not clearly bimodal in all families (1 haploid mode and 1 diploid mode)

- Some loci have more than 2 haplotypes, although we are working with diploids.

- There are no loci that clearly stand out as perfectly 'CSD like' among all individuals, this should be looked at family-wise.

These issues will be solved one by one in this chapter.

5.1 Improving ploidy separation metric

The inbreeding coefficient was computed using all males/females. Also, the raw heterozygosity may be a more accurate metric than inbreeding coefficient. Both issue require the analysis to be done in a per-family fashion, either by re-running the populations program for each family, or implementing family as an additional population layer.
Date: 27.06.2017
Solution: I could switch to observed homozygosity, but I think it is also fine to keep this metric (I can always change it later). I did however, re-run the populations program separately for each family. I did not add a population layer to the popmap, because it would not have allowed for different loci blacklists between families when solving the third issue. Visualizing the distribution of inbreeding coefficients across individuals reveals a pretty good (?) separation of individuals (Figure **??**).

5.2 Loci with 3 or more haplotypes

There are two possible explanation for the existence of these loci. It could be either due to a contamination, in which case few individuals would present many loci with 3 or more haplotype. On the other hand, if few loci have this issue in several individuals, it could be due to paralog merging. This can be solved by identifying which case is the most likely, and either excluding concerned individuals (first case), or loci (second case).

24.06.2017: Asked on STACKS google group about the haplotypes.tsv file. It is apparently normal to have more than 2 genotypes and depends on the parameters that were set in pstacks/ustacks. The answer I got, is that I should only work with the VCF file, which calls only 2 genotypes/site/individual. This imply figures **??** and **??** are not valid, as I misinterpreted the file content. I will re-generate these figures using the VCF file directly.
Date: 27.06.2017
Solution: I used the -012 argument in VCFtools to produce a genotype matrix from the populations output VCF file. This matrix encodes the genotype of each individual at each SNP with an integer representing the
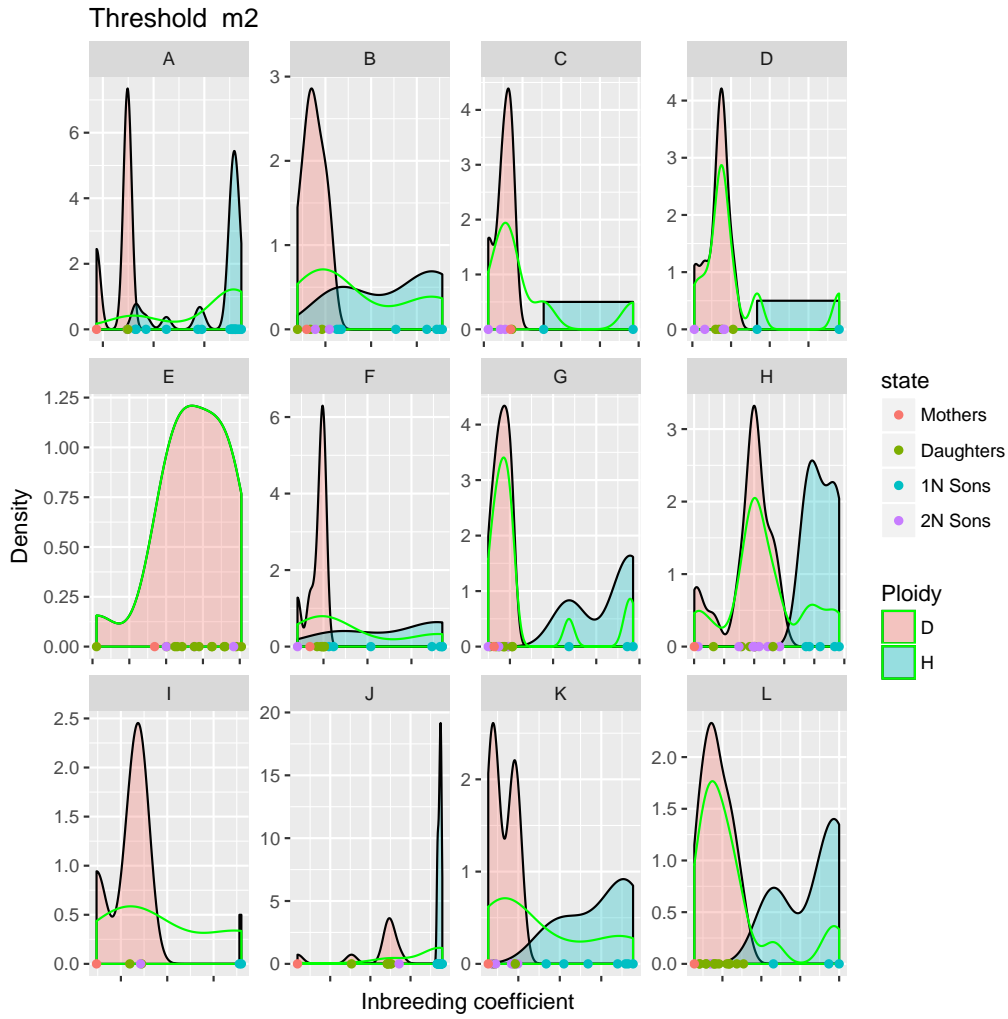
Figure 5.1: Distriution of individual inbreeding coefficient. The green curve represents the overall distribution of inbreeding coefficients within the family. The blue and pink areas represent the distribution of inbreeding coefficients for haploids and diploids, respectively, split using the m2 threshold defined at the beginning of chapter 4. Ideally, the green curve would follow a bimodal distribution clearly separating individuals by homozygosity.

number of non-reference alleles present. These matrices were generated separately for each family and used to regenerate the figures **??** and **??**.

The issue when visualizing the SNPs heterozygosity per family (Figure **??**) is that the low number of individuals allow for few different heterozygosity values for any given SNPs, yielding a high number of 'perfectly' CSD-like candidates. Solving the next issue or grouping SNPs by contig/locus may help reduce this number, but ultimately only association mapping will allow to find the best candidate region.

5.3   No clear signal in whole population

This issue could be expected from the biological system because it is believed that there is ml-CSD in this species; there should be no single locus that will fit the CSD pattern across all individuals. This mean we need to identify different loci independently in each family. There are several things which can be done to work around this issue.

First, and most importantly, the populations program should be taking family information into account, but this was done already when solving the first issue (Improving ploidy separation metric). Second, the data should be cleaned by blacklisting loci that are homozygous in mothers in each family, as there is no way these
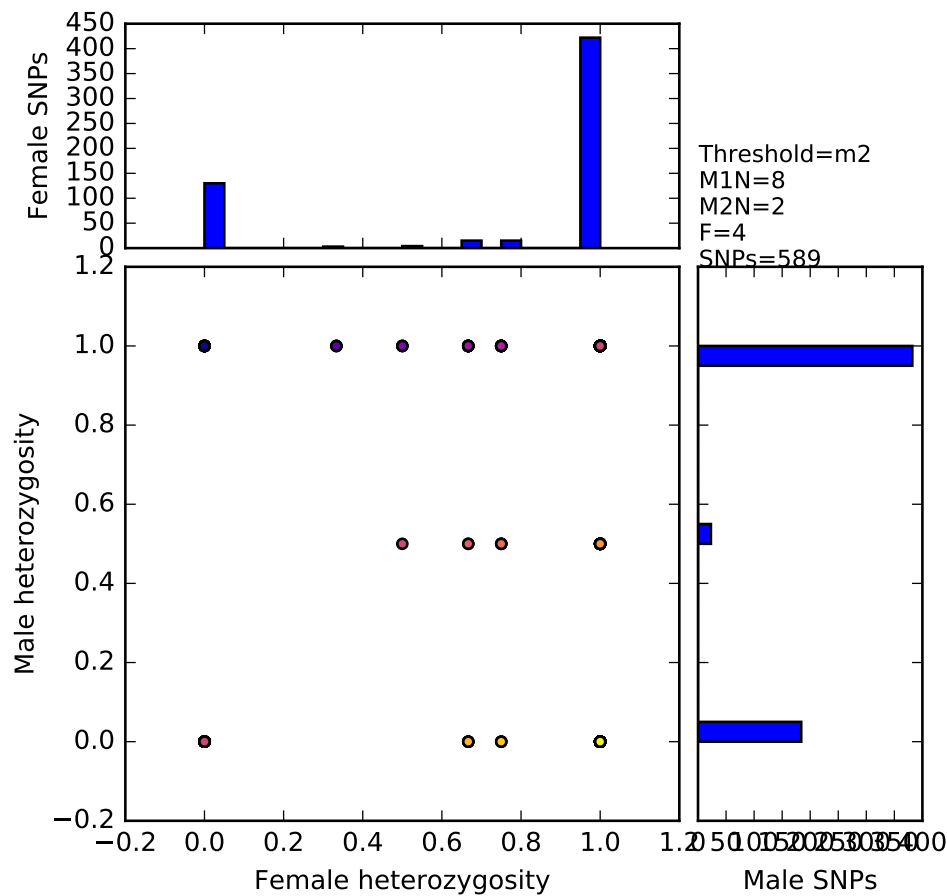
Figure 5.2: Example: SNPs heterozygosity across individuals of family B. The yellowness represents the proximity to CSD-like pattern (i.e. homozygous in all males and heterozygous in all females. Note: populations parameters set to : min depth (D)=5 and prop pop (r)=75

will be heterozygous in their respective daughters and be CSD candidates.

Date: 29.06.2017

Solution: I re-ran the explo assoc script used to generate figure **??** and excluded all SNPs that are homozygous in the mother (Figure **??**). It removed a significant number of SNPs from the set, but there are still way too many that fit the CSD pattern.

family	N SNPs
A	291
B	106
C	75
E	166
F	178
G	429
H	253
I	31
J	243
K	92
L	129

Table 5.1: Number of homozygous SNPs found in the mother of each family. Note family D is excluded since there was no genomic data available for this mother.

family	N SNPs
A	277
B	231
C	251
D	418
E	47
F	376
G	398
H	138
I	52
J	165
K	323
L	56

Table 5.2: Number of homozygous SNPs that fit the CSD pattern in each family.
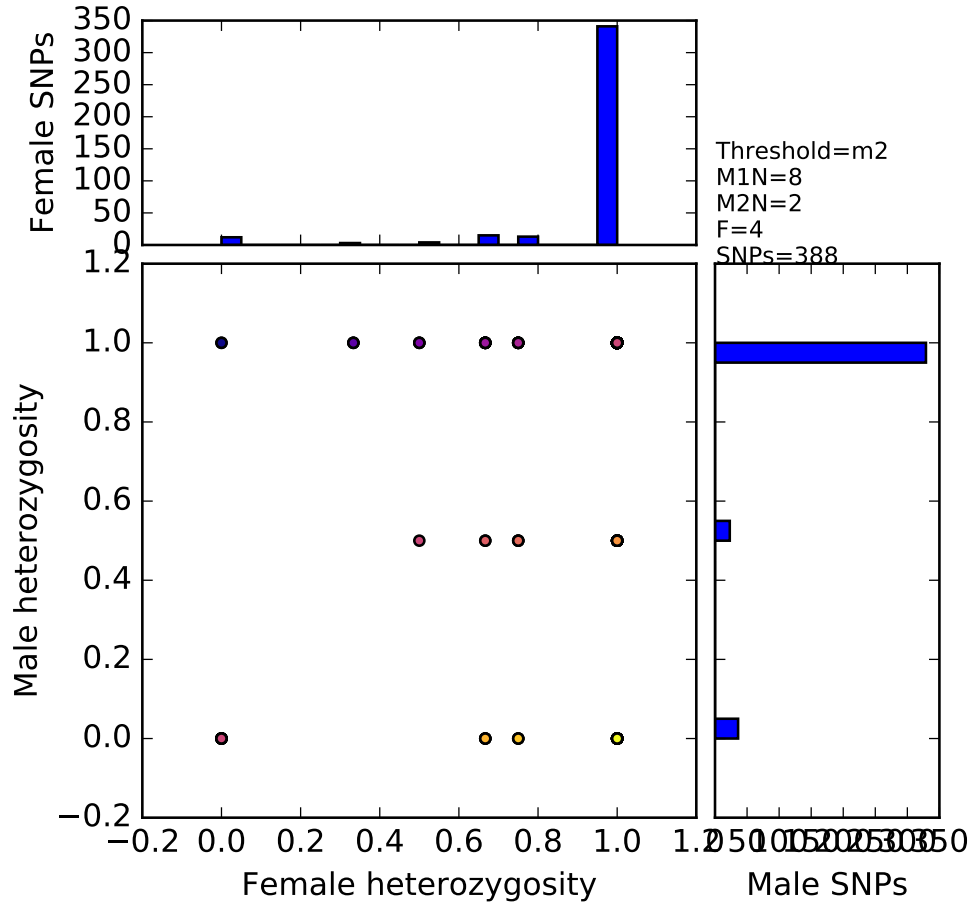
Figure 5.3: Example: SNPs heterozygosity across individuals of family B after removing SNPs that are ho-mozygous in the mother. The yellowness represents the proximity to CSD-like pattern (i.e. homozygous in all males and heterozygous in all females. Note: populations parameters set to : min depth (D)=5 and prop pop (r)=75

5.3.1 Transmission bias

Here I check how frequently SNPs that are homozygous in a mother are heterozygous in its offspring and vice versa. (to be continued)

Chapter 6

Association mapping

Date: 05.07.2017

Now that the data is cleaner, I will start with the association mapping. Casper provided me the genome with the contigs ordered by chromosomes (but not oriented?) he obtained with his linkage map. I will use this as the new reference genome and run the pipeline all the way from the mapping with it.

Date : 08.07.2017

I ran the whole pipeline on the ordered genome. Here are all statistics associated with all steps of the analysis previously described.

Statistics	Values
Assembly length (Mbp)	140.7
Largest scaffold (Mbp)	17.2
Mean scaffold size (kbp)	100
Median scaffold size (kbp)	25.7
N50 (kbp)	9518.5
Number of scaffolds	1408

Table 6.1: Assembly statistics for the ordered reference genome of *Lysiphlebus fabarum*

	tot	single	multi	miss
Number	170435458	94668507	44531856	31235095
proportion	1	0.555	0.261	0.183

Table 6.2: Mapping: Proportion and number of reads aligned on the ordered reference genome using BWA's aln algorithm with 4 mismatches allowed.

min˙cov	nloci	mean˙cov	sd˙cov
3	2694.0	50.0	69.4

Table 6.3: Pstacks: Summary statistics of stacks obtained with minimum coverage set to 3 in pstacks, using the ordered genome.

mismatch	mean˙loci	mean˙alleles
3	9072	34311

Table 6.4: Cstacks: Summary statistics of loci obtained with a mismatch value set to 3 in cstacks using the ordered genome.

6.1   TO DO

- change the n parameter in cstacks to M+-1 (probably 3 is best)

- save plots with n=1 just in case (maybe?)

- try splitting by ploidy with observed homozygosity instead of inbreeding coefficient

- prepare presentation for the 12th

- subset very confident individuals (top x% high hom and high het) and find set of loci always CSD-like in these.

- use these loci to classify ambiguous individuals

- Visualize average heterozygosity (or allele frequency) at each SNPs in offpring versus in mother to assess transmission bias (hom SNPs in mothers more often het in offspring or reversse).

- Validate haploid definition using haplotypes deduced from linkage map.

- convert VCF file into PED format using vcftools

- import data into genABEL and perform association mapping