*GTscore 1.3*

Enhanced Pipeline for *GT-seq* Sequences

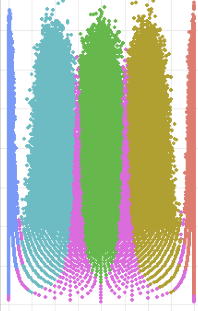
Garrett McKinney1

Carita Pascal

Lisa Seeb

School of Aquatic and Fishery Sciences

University of Washington

Seattle, WA

1Corresponding author: [gjmckinn@uw.edu](mailto:gjmckinn@uw.edu)

Table of Contents

[Introduction 4](#_Toc521659964)

[Flowchart of GTscore 5](#_Toc521659965)

[How Reads Are Counted 6](#_Toc521659966)

[Preparing Your Computer 7](#_Toc521659967)

[Step 1: Demultiplexing 7](#_Toc521659968)

[1.1 Input 7](#_Toc521659969)

[1.2 Output 8](#_Toc521659970)

[1.2.1 Fastq file for each sample 8](#_Toc521659971)

[1.2.2 Fastq file for all discarded sequences 8](#_Toc521659972)

[1.2.3 Text summary file of retained sequences by individual 8](#_Toc521659973)

[1.2.4 Text summary file of discarded reads giving the I7 + I5 sequences and number of read counts: 9](#_Toc521659974)

[Step 2: Amplicon Read Count 9](#_Toc521659975)

[2.1 Input 9](#_Toc521659976)

[2.2 Single SNP output 10](#_Toc521659977)

[2.2.1 Locus ID, ploidy and alleles for each single SNP: 10](#_Toc521659978)

[2.2.2 Counts for each SNP allele, rows are loci and columns are individuals: 11](#_Toc521659979)

[2.3 Haplotype output 11](#_Toc521659980)

[2.3.1 Locus ID, ploidy, and alleles for each locus: 11](#_Toc521659981)

[2.3.2 Counts each haplotype allele, rows are loci and columns are individuals 12](#_Toc521659982)

[2.4 Counts by individual of Total reads, Off-target Reads, Primer Only Reads, and Primer Probe Reads 12](#_Toc521659983)

[2.5 Counts by locus of Primer Reads and Primer Probe reads. 12](#_Toc521659984)

[Step 3: Genotyping and data output—GTScore.R 13](#_Toc521659985)

[3.1 Genotype output for single SNP 14](#_Toc521659986)

[3.2 Genotype output for haplotypes 14](#_Toc521659987)

[3.3 Exporting data 14](#_Toc521659988)

[3.3.1 Genepop format 14](#_Toc521659989)

[3.3.2 Rubias format 15](#_Toc521659990)

[Step 4: Data summaries—GTScore.R 15](#_Toc521659991)

[4.1. Locus data summaries 15](#_Toc521659992)

[4.1.1 Locus summary single SNPs 16](#_Toc521659993)

[4.1.2 Locus summary haplotypes 16](#_Toc521659994)

[4.1.3 Locus genotype rate plot 17](#_Toc521659995)

[4.1.4 Average Read Depth 17](#_Toc521659996)

[4.1.5 Average Read Depth vs Genotype Rate 18](#_Toc521659997)

[4.1.6 Plot of minor allele frequency 18](#_Toc521659998)

[4.1.7 Plot of major allele frequency 19](#_Toc521659999)

[4.2 Sample Data Summaries 20](#_Toc521660000)

[4.2.1 Summary of sample genotype rate 20](#_Toc521660001)

[4.2.2 Plot of sample genotype rate 20](#_Toc521660002)

[4.2.3 Plot of sample genotype rate relative to reads per sample 21](#_Toc521660003)

[4.3. Genotype Plots 22](#_Toc521660004)

[4.3.1 Allele ratio plots 22](#_Toc521660005)

[4.3.1.1 Singleton locus 22](#_Toc521660006)

[4.3.1.2 Duplicate locus 23](#_Toc521660007)

[4.3.1.3 Diverged duplicate locus 23](#_Toc521660008)

[4.3.2 Scatter plots 24](#_Toc521660009)

[4.3.2.1 Singleton locus 24](#_Toc521660010)

[4.3.2.2 Duplicate locus 24](#_Toc521660011)

[4.3.2.3 Diverged duplicate locus 25](#_Toc521660012)

[Step 5: Optional locus diagnostics—GTScore.R 26](#_Toc521660013)

[5.1 Sequence alignment to each locus (Perl) 26](#_Toc521660014)

[5.1.1 Output from matchReads.pl (sequences shortened for clarity) 26](#_Toc521660015)

[5.2 MSA alignment (GTscore.R) 27](#_Toc521660016)

[5.2.1 MSA alignment saved as a conditionally formatted excel file*.* 28](#_Toc521660017)

[5.3. Plotting Mismatches by Position (Perl and GTscore.R) 28](#_Toc521660018)

[5.3.1 Calculate sequence mismatches by position (Perl) 28](#_Toc521660019)

[5.3.2. Plot of sequence mismatches by position (GTscore.R) 29](#_Toc521660020)

***GTscore***

# Introduction

*GTscore* is an enhanced analysis pipeline for handling GT-seq data (Campbell *et al.* 2015). It has been designed to process amplicon sequencing data in fastq format and can directly handle output from *Stacks* (Catchen *et al.* 2013). It runs either in a Windows or Unix environment. Enhancements beyond the original GT-seq\_pipeline (see Campbell et al. 2015; <https://github.com/GTseq/GTseq-Pipeline>) include:

* Determines genotypes for both multi-SNP haplotypes and single-SNP genotypes
* Retains phase for multi-SNP haplotypes
* Handles genotypes for loci with varying ploidy levels
* Provides detailed summaries and plots and optional diagnostics

Multi-SNP haplotypes or so-called microhaplotypes are becoming increasing important in forensics (Kidd & Speed 2015); they also have been shown to substantially increase accuracy of mixture and individual assignments analyses in non-model organisms (e.g., McKinney *et al.* 2017a; Baetscher *et al.* 2018).

In addition, many eukaryotic genomes retain duplicates loci or paralogs. While these loci have historically been treated as a nuisance and commonly removed from NGS data by filtering (Dufresne 2016), recent analyses now allow highly accurate identification of these loci through pipelines such as *HDplot* (McKinney *et al.* 2017b). Duplicated loci are being increasingly incorporated into population genetic analyses (Waples *et al.* 2016; Limborg *et al.* 2017). Amplicon sequencing data derived from GT-seq pipelines is particularly powerful for determining genotypes of duplicated or higher ploidy loci because of the relatively high depths of coverage available (McKinney *et al.* 2018).

*GT-score* is capable of genotyping data both as single SNPs and as multi-SNP haplotypes; both analysis methods handle varying ploidy levels. There are main four steps to running *GT-score:*

1. Demultiplex raw sequence data
2. Count sequence reads for each locus
3. Genotype samples based on read counts
4. Produce data summaries and plots

Steps 1 and 2 are Perl scripts. Steps 3 and 4 are run through R and are partially based on the R scripts in *PolyGen* (McKinney *et al.* 2018).

An additional fifth diagnostic step is included to assist in identifying patterns of sequence variation for loci; diagnostics require both R and Perl scripts.

5. Optional diagnostics

# Flowchart of GTscore

Dashed line indicate optional commands. Green boxes are input. Blue boxes indicate single SNP pipeline; yellow boxes indicate haplotype pipeline.

**STEP 3. Genotyping**

**GTScore.R**

Genotypes single SNP

Genotypes Haplotypes

Text summary of discarded sequences

**STEP 1. Demultiplexing**

**PERL**

**STEP 2. Amplicon Read Counter**

**PERL**

**STEP 4. Data Summaries and Plots (run pipelines separately) GTScore.R**

**GTScore.R**

Haplotypes Analysis

Single SNPs Analysis

Fastq sequence file

Fastq file of all discarded sequence

Text summary of retained sequences by individual

Fastq file for each individual

Text barcode file

Allele Reads Haplotypes

Text file of discarded reads for each individual

Locus Data Summaries: genotype rate, average read depth, MAF, MajorAF, alleles per locus, freq. per allele

304

Text primer/probe file for each locus

Locus Table Haplotypes

Allele Reads Single SNPs

Locus Table single SNP

Text file of read counts by individual

Text file of read counts by locus

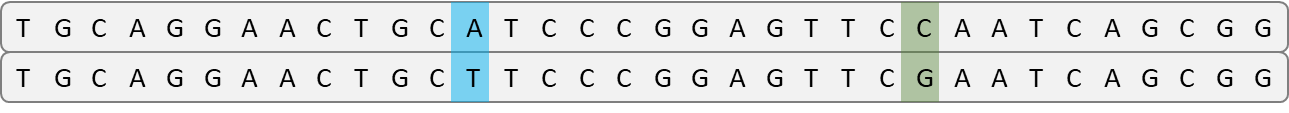
Plots for all summary metrics

Genotype Plots (single SNP only): Allele ratio, Scatter plots

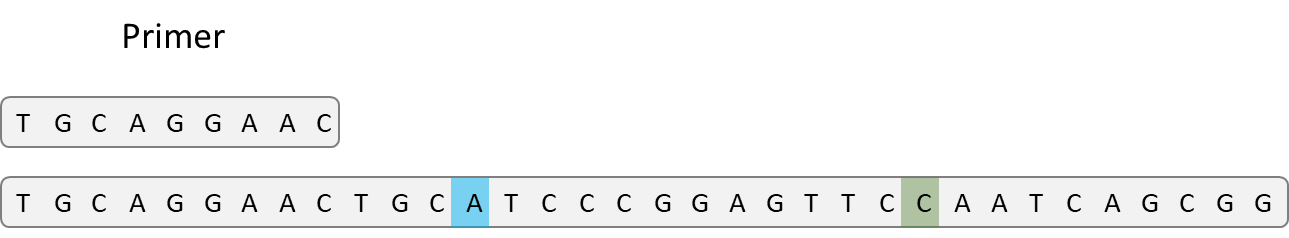
Export to Genepop or Rubias format; export pipelines separately

# How Reads Are Counted

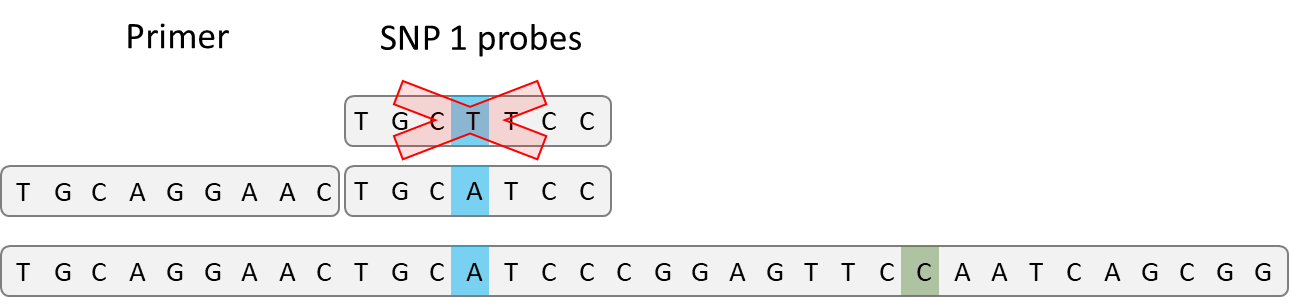
The steps below illustrate how reads are counted using bioinformatics primers and probes for an example locus with two SNPs (A/T and C/G).



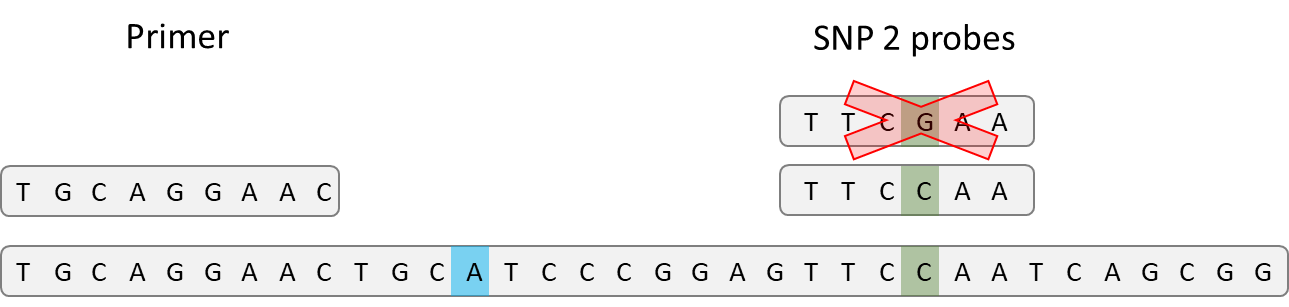
Step 1: align primers against beginning of sequence to see if there is a match



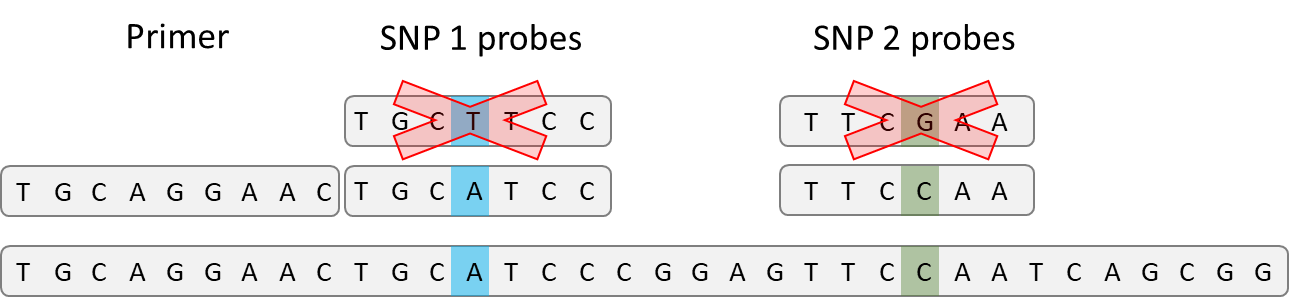
Step 2: if a primer matches, align the probes for the first SNP to the sequence to identify which allele is present. In this example, the A allele matches so the read is counted toward the A allele for the first SNP.



Step 3: if the locus contains multiple SNPs, align the probes for subsequent SNPs. This locus has two SNPs so only one more probe set will be aligned. In this example, the C allele is a match so the read is counted towards the C allele for the second SNP.



Step 4: combine the results for individual SNPs into haplotypes. Since the A allele had a match for the first SNP and the C allele had a match for the second SNP, the read is counted towards the AC haplotype.



During the read counting stage, output files are automatically generated for both single-SNP and haplotype analysis.

# Preparing Your Computer

1. Install Perl—a recent version is recommended.

The following Perl packages are required

* Algorithm::Combinatorics
* Excel-Writer-XLSX (for optional diagnostics)

2. Install R and R packages

The following R packages are required:

* library(stringr)
* library(ggplot2)
* library(tools)
* library(plyr)
* library(pbapply)
* library(msa) (for optional diagnostics)
  + Note: This is a bioCLite package.
  + To install: source ("https://bioconductor.org/biocLite.R") biocLite("msa")

# Step 1: Demultiplexing

The Demultiplexer splits the raw data into individuals based on unique plate and individual barcodes. The demultiplexer is written in perl (DemultiplexGTseq.pl) and can be run from a Unix or Windows Command.

## 1.1 Input

Input flags for this script are

*–b* tab delimited file of barcodes with sample names

*-s* fastq file of sequence data

>perl DemultiplexGTseq.pl -b *barcodeFile*.*txt* -s *sequenceFile.fastq*

An example of the barcode file format is given below. The first row is a header with labels as follows: first column contains the sample ID, the second column contains the i7 barcode (plate ID), and the third column contains the i5 barcode (individual ID).

|  |  |  |
| --- | --- | --- |
| Sample ID | I7\_barcode | I5\_barcode |
| Sample1 | AGCCTC | CCGTTT |
| Sample2 | AGCCTC | AACGTT |

The demultiplexer creates a file for every sample and will take from one to several hours depending on hardware configurations to demultiplex a single lane of Illumina HiSeq 4000 data (~400 million reads).

## 1.2 Output

Onscreen

The script processes a million reads at a time and prints out progress of *Total Reads Processed* and *Interval Time* for each million reads as follows:

Total Reads Processed: 1000000

Interval Time: 20

Total Reads Processed: 2000000

Interval Time: 20

Total Reads Processed: 3000000

Interval Time: 20

Total Reads Processed: 4000000

Interval Time: 19

Demultiplex.pl creates:

### 1.2.1 Fastq file for each sample

1.2.2 Fastq file for all discarded sequences*: sequenceFile\_discarded.fastq*

1.2.3 Text summary file of retained sequences by individual: *sequenceFile\_DemultiplexGTseq\_log.txt.* A listing of the top 100 discarded non-target barcodes follows the individual results.

|  |  |  |
| --- | --- | --- |
| Total | Reads: | 4000000 |
| Retained | Reads: | 2976976 |
| Discarded | Reads: | 1023024 |
|  |  |  |
| Sample | Barcode | Reads |
| KAROL05\_0037 | CGATGTAT+TCTTCTTC | 69760 |
| KGEOR05\_0712 | CGATGTAT+TCGCCATC | 64076 |
| KGONF06\_0023 | TTAGGCAT+TCTTCTTC | 62540 |
| KEEK02\_0008 | ATCACGAT+TCTTCTTC | 61820 |
| KAROL05\_0010 | CGATGTAT+ACTCTTTC | 60660 |
| KGEOR05\_0093 | CGATGTAT+TCGATTTC | 60136 |
| KAROL05\_0074 | CGATGTAT+AGCGCATC | 58488 |
| KGEOR05\_0092 | CGATGTAT+GTCATCTC | 57908 |
| KGEOR05\_0065 | CGATGTAT+TTATGATC | 57876 |
| . |  |  |
| . |  |  |
| . |  |  |
| Top 100 Discarded Barcodes | | |
| Barcode | Reads |  |
| CCCCCCCC+TCTTTCCC | 10036 |  |
| GCGCAACC+TCTTTCCC | 7429 |  |
| GCTCAACC+TCTTTCCC | 6324 |  |
| ATCACGAT+GTTCAGAG | 5458 |  |
| CCCCCCCC+CCCCCCCC | 5408 |  |
| CGATGTAT+GTTCAGAG | 5099 |  |

1.2.4 Text summary file of discarded reads giving the I7 + I5 sequences and number of read counts: *sequenceFile\_discardedSummary.txt*

|  |  |
| --- | --- |
| CCCCCCCC+TCTTTCCC | 10036 |
| GCGCAACC+TCTTTCCC | 7429 |
| GCTCAACC+TCTTTCCC | 6324 |
| ATCACGAT+GTTCAGAG | 5458 |
| CCCCCCCC+CCCCCCCC | 5408 |
| CGATGTAT+GTTCAGAG | 5099 |
| GAGCAACC+TCTTTCCC | 4162 |
| TTAGGCAT+GTTCAGAG | 3805 |

# Step 2: Amplicon Read Count

The read counter is written in perl (AmpliconReadCounter.pl).

The program:

* Identifies each unique sequence, then counts the number of times each unique sequence occurs within an individual
* Aligns each unique sequence with primer and probe; if the sequence doesn’t align, then it is excluded as an off-target sequence and reports by individual and by locus are given.
  + Note: By default, all primers are trimmed to the length of the shortest primer to increase speed. Optionally the full length primer can be used for the primer but this may significantly increase run timing depending on variation in primer lengths across loci.

## 2.1 Input

Input flags for this script are

*–p* a tab delimited file containing primer/probe information for each locus

*--files* a text file containing a list of .fastq sequence files to count reads from.

Optional flags

--*printDiscarded* outputs discarded reads for each individual

--*useFullPrimer* uses the full primer for counting reads rather than the trimmed primer

--*prefix* optional prefix for output file names

>perl AmpliconReadCounter.pl -p *primerProbeFile.txt --*files *sampleFiles.txt* [--printDiscarded] [--useFullPrimer] [--prefix]

In the primer/probe file, a separate line is included for each SNP at each locus; multi-SNP loci will include multiple lines. Example of Primer/probe input file; RAD18527 has 1 SNP, RAD18777 has 2 SNPs, and RAD18602 has 4 SNPs. RAD18048 is a tetraploid isolocus with 1 SNP.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Locus | Ploidy | SNPpos | Allele1 | Allele2 | Probe1 | Probe2 | Primer |
| RAD18527 | 2 | 71 | A | T | ATCCCACCATT | ATCCCTCCATT | AGCAGGTGTTCTGTTCTGTCC |
| RAD18602 | 2 | 55 | T | A | CTATGTAATGA | CTATGAAATGA | ACAGGGTGATGCCAAGCAG |
| RAD18602 | 2 | 20 | A | G | AGCAGAA[AG]GTTTG | AGCAGGA[AG]GTTTG | ACAGGGTGATGCCAAGCAG |
| RAD18602 | 2 | 22 | A | G | AGCAG[AG]AAGTTTG | AGCAG[AG]AGGTTTG | ACAGGGTGATGCCAAGCAG |
| RAD18602 | 2 | 45 | A | G | GCAATCATCTAACAATAACCTATG | GCAATCATCTAACAGTAACCTATG | ACAGGGTGATGCCAAGCAG |
| RAD18777 | 2 | 32 | T | A | AAAAATAAAAG | AAAAAAAAAAG | AGGACTTGGAGTGGAGTGTGG |
| RAD18777 | 2 | 51 | C | T | TTCTTCATCAC | TTCTTTATCAC | AGGACTTGGAGTGGAGTGTGG |
| RAD18048 | 4 | 56 | G | T | CCAGGGTCACG | CCAGGTTCACG | ACGACAGGCTGTTGTCCAG |

\*Note: Probe sequences can contain square bracketed bases to include known variations within the in-silico probe.

Example of sample input file:

KEEK05\_0031.fastq

KEEK05\_0032.fastq

KEEK05\_0033.fastq

KEEK05\_0034.fastq

KEEK05\_0035.fastq

Onscreen: The names of the samples are printed as the script progresses:

sample 1 finished: KEEK05\_0031 Processing time: 3 seconds

sample 2 finished: KEEK05\_0032 Processing time: 3 seconds

sample 3 finished: KEEK05\_0033 Processing time: 3 seconds

AmpliconReadCounter.pl outputs four primary files (two for singleSNPs and two for multi-SNP) plus two summary files.

## 2.2 Single SNP output

### 2.2.1 Locus ID, ploidy and alleles for each single SNP:

LocusTable\_singleSNPs.txt

|  |  |  |
| --- | --- | --- |
| Locus\_ID | ploidy | alleles |
| Ots\_100884-287\_1 | 2 | C,T |
| Ots\_101119-381\_1 | 2 | C,T |
| Ots\_101554-407\_1 | 2 | C,G |
| Ots\_101704-143\_1 | 2 | G,T |
| RAD18527\_71 | 2 | A,T |
| RAD18602\_20 | 2 | A,G |
| RAD18602\_22 | 2 | A,G |
| RAD18602\_45 | 2 | A,G |
| RAD18602\_55 | 2 | A,T |
| RAD18777\_32 | 2 | A,T |
| RAD18777\_51 | 2 | C,T |
| RAD18048\_56 | 4 | G,T |

### 2.2.2 Counts for each SNP allele, rows are loci and columns are individuals:

AlleleReads\_singleSNPs.txt

|  |  |  |  |
| --- | --- | --- | --- |
|  | KEEK02\_0001 | KEEK02\_0002 | KEEK02\_0003 |
| Ots\_100884-287 | 0,84 | 29,37 | 0,51 |
| Ots\_101119-381 | 22,0 | 16,0 | 17,0 |
| Ots\_101554-407 | 0,94 | 0,81 | 0,82 |
| Ots\_101704-143 | 35,0 | 54,0 | 63,0 |
| RAD18527\_71 | 3,53 | 27,30 | 42,0 |
| RAD18602\_20 | 0,26 | 0,28 | 0,33 |
| RAD18602\_22 | 6,20 | 26,2 | 20,13 |
| RAD18602\_45 | 6,21 | 27,2 | 20,12 |
| RAD18602\_55 | 0,27 | 0,29 | 9,23 |
| RAD18777\_32 | 26,0 | 15,0 | 10,0 |
| RAD18777\_51 | 0,45 | 0,40 | 0,24 |
| RAD18048\_56 | 194,1 | 161,72 | 131,0 |

## 2.3 Haplotype output

\*Note: Loci with a single-SNP are included in analysis and will have identical results to the single-SNP analysis above.

### 2.3.1 Locus ID, ploidy, and alleles for each locus:

*LocusTable\_haplotypes.txt*

All possible alleles, even if not observed, are tracked. If 2 SNPs, there are 4 possible alleles; 3 SNPs, 8 possible alleles; 4 SNPs 16 possible alleles, etc.

|  |  |  |
| --- | --- | --- |
| Locus\_ID | ploidy | alleles |
| Ots\_100884-287 | 2 | C,T |
| Ots\_101119-381 | 2 | C,T |
| Ots\_101554-407 | 2 | C,G |
| Ots\_101704-143 | 2 | G,T |
| RAD18527 | 2 | A,T |
| RAD18602 | 2 | AAAA,AAAT,AAGA,AAGT,AGAA,AGAT,AGGA,AGGT,GAAA,GAAT,GAGA,GAGT,GGAA,GGAT,GGGA,GGGT |
| RAD18777 | 2 | AC,AT,TC,TT |
| RAD18048 | 4 | G,T |

2.3.2 Counts each haplotype allele, rows are loci and columns are individuals:

*AlleleReads\_haplotypes.txt*

|  |  |  |  |
| --- | --- | --- | --- |
|  | KEEK02\_0001 | KEEK02\_0002 | KEEK02\_0003 |
| Ots\_100884-287 | 0,84 | 29,37 | 0,51 |
| Ots\_101119-381 | 22,0 | 16,0 | 17,0 |
| Ots\_101554-407 | 0,94 | 0,81 | 0,82 |
| Ots\_101704-143 | 35,0 | 54,0 | 63,0 |
| RAD18527 | 3,53 | 27,30 | 42,0 |
| RAD18602 | 0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,20 | 0,0,0,0,0,0,0,0,0,26,0,0,0,0,0,2 | 0,0,0,0,0,0,0,0,9,11,0,0,0,0,0,12 |
| RAD18777 | 0,25,0,0 | 0,14,0,0 | 0,10,0,0 |
| RAD18048 | 194,1 | 161,72 | 131,0 |

2.4 Counts by individual of Total reads, Off-target Reads, Primer Only Reads, and Primer Probe Reads:

*GTscore\_individualSummary.txt.* Only Primer Probe reads are used in downstream genotyping. Primer only and primer probe proportions relative to the total are also given.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sample | Total Reads | Off-target Reads | Primer Only Reads | Primer Probe Reads | Off-target Proportion | Primer Only Proportion | Primer Probe Proportion |
| KEEK02\_0001 | 10582 | 2878 | 4042 | 3662 | 0.27 | 0.38 | 0.35 |
| KEEK02\_0002 | 13760 | 3973 | 6164 | 3623 | 0.29 | 0.45 | 0.26 |
| KEEK02\_0003 | 12834 | 3857 | 6245 | 2732 | 0.30 | 0.49 | 0.21 |
| KEEK02\_0004 | 11399 | 3528 | 5603 | 2268 | 0.31 | 0.49 | 0.20 |
| KEEK02\_0005 | 12119 | 3328 | 4775 | 4016 | 0.27 | 0.39 | 0.33 |
| KEEK02\_0006 | 12407 | 3834 | 5792 | 2781 | 0.31 | 0.47 | 0.22 |

\*Note: If no reads are identified for a sample, the Total Reads entry will be a 0, all other entries will be NA

2.5 Counts by locus of Primer Reads and Primer Probe reads.

This is useful for identifying loci that may be amplifying other regions of the genome (e.g. Ots\_101554-407).

|  |  |  |  |
| --- | --- | --- | --- |
| Locus | Primer Reads | Primer Probe Reads | Primer Probe Proportion |
| Ots\_100884-287 | 481 | 469 | 0.98 |
| Ots\_101119-381 | 278 | 207 | 0.74 |
| Ots\_101554-407 | 6791 | 528 | 0.08 |
| Ots\_101704-143 | 479 | 399 | 0.83 |
| Ots\_101770-82 | 359 | 338 | 0.94 |
| Ots\_102213-210 | 146 | 111 | 0.76 |

# Step 3: Genotyping and data output—GTScore.R

Genotyping steps must be done separately using single SNP and multi-SNP haplotype pipelines. Portions of GTScore.R include scripts from polyGen.R (McKinney *et al.* 2018). The scripts first read in the locus tables and allele reads files for single-SNP and haplotype analyses and then calls polyGen.R. If desired, genotyping output can be exported as formatted files for population genetics programs (described below).

##################

#GENOTYPING

##################

#set working directory and load GTscore.R

genotypeDirectory="C:/GTscoreExampleDirectory "

setwd(genotypeDirectory)

source("GTscore.R")

#genotype samples

singleSNP\_locusTable<-read.delim("LocusTable\_singleSNPs.txt",header=TRUE)

singleSNP\_alleleReads<-read.delim("AlleleReads\_singleSNPs.txt",header=TRUE,row.names=1)

haplotype\_locusTable<-read.delim("LocusTable\_haplotypes.txt",header=TRUE)

haplotype\_alleleReads<-read.delim("AlleleReads\_haplotypes.txt",header=TRUE,row.names=1)

#generate singleSNP genotypes

polyGenResults\_singleSNP<-polyGen(singleSNP\_locusTable,singleSNP\_alleleReads)

#look at first five rows and columns

polyGenResults\_singleSNP[1:5,1:5]

#write to results to file

write.table(polyGenResults\_singleSNP,"polyGenResults\_singleSNP.txt",quote=FALSE,sep="\t")

#generate haplotype genotypes

polyGenResults\_haplotypes<-polyGen(haplotype\_locusTable,haplotype\_alleleReads)

#look at first five rows and columnspolyGenResults\_haplotypes[1:5,1:5]

#write results to file

write.table(polyGenResults\_haplotypes,"polyGenResults\_haplotypes.txt",quote=FALSE,sep="\t")

Onscreen: Percent completed and time remaining are displayed.



## 3.1 Genotype output for single SNP

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | KEEK02\_0001 | KEEK02\_0002 | KEEK02\_0003 | KEEK02\_0004 | KEEK02\_0005 |
| Ots\_100884-287\_1 | T,T | C,T | T,T | T,T | T,T |
| Ots\_101119-381\_1 | C,C | C,C | C,C | C,C | C,C |
| Ots\_101554-407\_1 | G,G | G,G | G,G | C,G | G,G |
| Ots\_101704-143\_1 | G,G | G,G | G,G | G,G | T,T |
| RAD18048\_56 | G,G,G,G | G,G,G,T | G,G,G,G | G,G,G,G | G,G,G,G |
| RAD18527\_71 | T,T | A,T | A,A | A,T | A,A |
| RAD18602\_20 | G,G | G,G | G,G | G,G | G,G |
| RAD18602\_22 | A,G | A,A | A,G | A,G | A,A |
| RAD18602\_45 | A,G | A,A | A,G | A,G | A,A |
| RAD18602\_55 | T,T | T,T | A,T | T,T | T,T |
| RAD18777\_32 | A,A | A,A | A,A | A,A | A,A |
| RAD18777\_51 | T,T | T,T | T,T | T,T | T,T |

\*Note: Unassigned genotypes are given a “0” entry

## 3.2 Genotype output for haplotypes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | KEEK02\_0001 | KEEK02\_0002 | KEEK02\_0003 | KEEK02\_0004 | KEEK02\_0005 |
| Ots\_100884-287 | T,T | C,T | T,T | T,T | T,T |
| Ots\_101119-381 | C,C | C,C | C,C | C,C | C,C |
| Ots\_101554-407 | G,G | G,G | G,G | C,G | G,G |
| Ots\_101704-143 | G,G | G,G | G,G | G,G | T,T |
| RAD18048 | G,G,G,G | G,G,G,T | G,G,G,G | G,G,G,G | G,G,G,G |
| RAD18527 | T,T | A,T | A,A | A,T | A,A |
| RAD18602 | GAAT,GGGT | GAAT,GAAT | GAAT,GGGT | GAAT,GGGT | GAAT,GAAT |
| RAD18777 | AT,AT | AT,AT | AT,AT | AT,AT | AT,AT |

\*Note: Unassigned genotypes are given a “0” entry

## 3.3 Exporting data

Genotyping results can be exported directly for use in Genepop (Rousset 2008) and Rubias (Moran & Anderson 2018) (https://cran.r-project.org/web/packages/rubias/index.html) using the exportGenepop and exportRubias commands. It is important to note that neither program supports paralog analysis, so by default paralogs are excluded from export in these formats. Paralogs can be optionally exported in the genepop output by specifying the command exportParalogs=TRUE.

### 3.3.1 Genepop format

#export single SNP results

exportGenepop(polyGenResults\_singleSNP,singleSNP\_locusTable, filename="polyGenResults\_singleSNP\_genepop.txt",exportParalogs=FALSE)

#export haplotype results

exportGenepop(polyGenResults\_haplotypes,haplotype\_locusTable,filename="polyGenResults\_haplotype\_genepop.txt",exportParalogs=FALSE)

### 3.3.2 Rubias format

Rubias requires four columns of metadata for each sample (see <https://github.com/eriqande/rubias#input-data>). Metadata for each sample can be optionally loaded and then included in the Rubias export, otherwise the appropriate columns will appear in the output file but all entries will be NA.

Example metadata format

|  |  |  |  |
| --- | --- | --- | --- |
| sample\_type | repunit | collection | indiv |
| reference | Eek | KEEK | KEEK02\_0001 |
| reference | Eek | KEEK | KEEK02\_0002 |
| reference | Eek | KEEK | KEEK02\_0003 |
| reference | George | KGEOR | KGEOR05\_0053 |
| reference | George | KGEOR | KGEOR05\_0054 |
| mixture | NA | Col1 | unk18\_0001 |
| mixture | NA | Col1 | unk18\_0002 |

#load sample metadata

sampleMetaData<-read.delim("sampleMetaData.txt",header=TRUE)

#export single SNP results

exportRubias(polyGenResults\_singleSNP,singleSNP\_locusTable,sampleMetaData,filename="polyGenResults\_singleSNP\_rubias.txt")

#export haplotype results

exportRubias(polyGenResults\_haplotypes,haplotype\_locusTable,sampleMetaData,filename="polyGenResults\_haplotypes\_rubias.txt")

# Step 4: Data summaries—GTScore.R

Data summaries for loci and samples are provided. Explanations of data summaries and example code are included below. Only plots from the single-SNP analysis are shown: the haplotype summary data can be plotted in the same manner but wasn’t shown for brevity.

## 4.1. Locus data summaries

The summarizeGTscore command generates summary data for each locus in table form. The summary data includes genotype rate, average read depth, minor (least frequent) allele frequency, major (most frequent) allele frequency, alleles per locus, and frequency per allele. Minor allele frequency is a common metric for filtering loci that are likely to be uninformative for population genetics; however, loci with haplotype alleles may have an allele with very low frequency but still have appreciable frequency at multiple other alleles. Because of this, the major allele frequency is included in output, as well as the observed frequencies for all alleles at a given locus.

### 4.1.1 Locus summary single SNPs

#summarize single SNP results

singleSNP\_summary<-summarizeGTscore(singleSNP\_alleleReads,singleSNP\_locusTable, polyGenResults\_singleSNP)

write.table(singleSNP\_summary,"singleSNP\_summary.txt",quote=FALSE,sep="\t",row.names=FALSE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Locus\_ID | AvgReadDepth | GenotypeRate | minAF | majAF | alleles | allFreqs |
| Ots\_100884-287\_1 | 40.01 | 0.99 | 0.12 | 0.88 | C,T | 0.12,0.88 |
| Ots\_101119-381\_1 | 17.86 | 0.97 | 0.00 | 1.00 | C,T | 1,0 |
| Ots\_101554-407\_1 | 48.36 | 1.00 | 0.09 | 0.91 | C,G | 0.09,0.91 |
| Ots\_101704-143\_1 | 34.36 | 0.99 | 0.03 | 0.97 | G,T | 0.97,0.03 |
| RAD18048\_56 | 197.03 | 0.99 | 0.03 | 0.97 | G,T | 0.97,0.03 |
| RAD18527\_71 | 75.55 | 0.99 | 0.34 | 0.66 | A,T | 0.66,0.34 |
| RAD18602\_20 | 26.61 | 0.99 | 0.01 | 0.99 | A,G | 0.01,0.99 |
| RAD18602\_22 | 26.61 | 0.98 | 0.27 | 0.73 | A,G | 0.73,0.27 |
| RAD18602\_45 | 26.30 | 0.98 | 0.27 | 0.73 | A,G | 0.73,0.27 |
| RAD18602\_55 | 26.74 | 0.99 | 0.05 | 0.95 | A,T | 0.05,0.95 |
| RAD18777\_32 | 39.23 | 0.99 | 0.00 | 1.00 | A,T | 1,0 |
| RAD18777\_51 | 69.46 | 0.99 | 0.11 | 0.89 | C,T | 0.11,0.89 |

### 4.1.2 Locus summary haplotypes

#summarize haplotype results

haplotype\_summary<-summarizeGTscore(haplotype\_alleleReads,haplotype\_locusTable,polyGenResults\_haplotypes)

write.table(haplotype\_summary,"haplotype\_summary.txt",quote=FALSE,sep="\t",row.names=FALSE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Locus\_ID | AvgReadDepth | GenotypeRate | minFreq | maxFreq | alleles | allFreqs |
| Ots\_100884-287 | 40.01 | 0.99 | 0.12 | 0.88 | C,T | 0.12,0.88 |
| Ots\_101119-381 | 17.86 | 0.97 | 0.00 | 1.00 | C,T | 1,0 |
| Ots\_101554-407 | 48.36 | 1.00 | 0.09 | 0.91 | C,G | 0.09,0.91 |
| Ots\_101704-143 | 34.36 | 0.99 | 0.03 | 0.97 | G,T | 0.97,0.03 |
| RAD18048 | 197.03 | 0.99 | 0.03 | 0.97 | G,T | 0.97,0.03 |
| RAD18527 | 75.55 | 0.99 | 0.34 | 0.66 | A,T | 0.66,0.34 |
| RAD18602 | 25.75 | 0.98 | 0.00 | 0.68 | AAAA,AAAT,AAGA,  AAGT,AGAA,AGAT,  AGGA,AGGT,GAAA,  GAAT,GAGA,GAGT,  GGAA,GGAT,GGGA,  GGGT | 0,0,0,  0,0,0,  0,0.01,0.05,  0.68,0,0,  0,0,0,  0.26 |
| RAD18777 | 38.30 | 0.99 | 0.00 | 0.88 | AC,AT,TC,TT | 0.12,0.88,0,0 |

Plots can be generated for each of the summary metrics using the example code below. Only plots from the single-SNP analysis are shown, the haplotype summary data can be plotted in the same manner but are not shown for brevity.

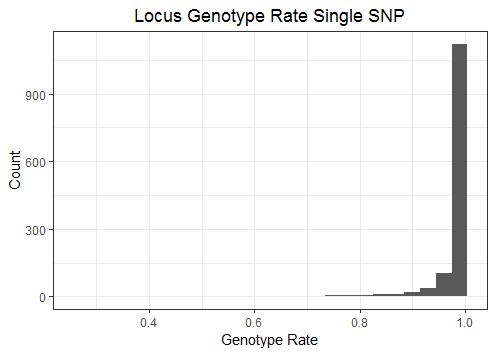
### 4.1.3 Locus genotype rate plot

#plot genotype rate for single SNPs

ggplot()+geom\_histogram(data=singleSNP\_summary,aes(x=GenotypeRate),binwidth=0.03) +xlim(0,1)+

labs(title="Locus Genotype Rate Single SNP", x="Genotype Rate", y="Count")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



### 4.1.4 Average Read Depth

#This is calculated by dividing total depth by number of samples with nonZero reads

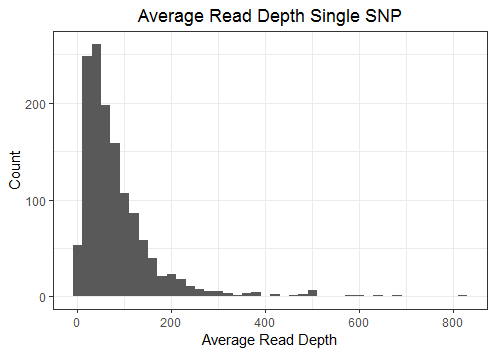
#possible diagnostic is difference in average depth between SNPs at a locus

#plot average read depth for single SNPs

ggplot()+geom\_histogram(data=singleSNP\_summary,aes(x=AvgReadDepth),binwidth=20)+

labs(title="Average Read Depth Single SNP", x="Average Read Depth", y="Count")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



### 4.1.5 Average Read Depth vs Genotype Rate

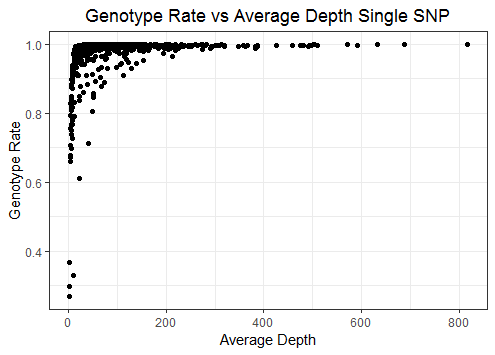
#plotting the average read depth vs genotype rate can show loci that have increased missing data as a result of low read depth.

#plot genotype rate relative to average depth for single SNPs

ggplot()+geom\_point(data=singleSNP\_summary,aes(x=AvgReadDepth,y=GenotypeRate))+ylim(0,1)+

labs(title="Genotype Rate vs Average Depth Single SNP", x="Average Depth", y="Genotype Rate")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



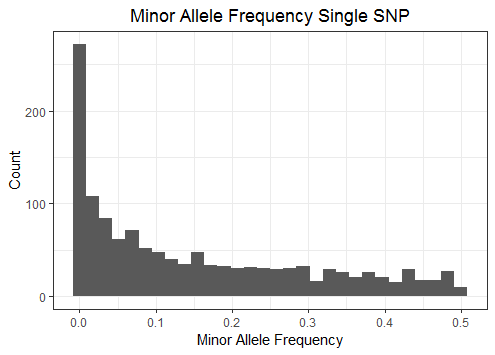
### 4.1.6 Plot of minor allele frequency

#plot distribution of minor allele frequency for single SNPs

ggplot()+geom\_histogram(data=singleSNP\_summary,aes(x=minAF))+

labs(title="Minor Allele Frequency Single SNP", x="Minor Allele Frequency", y="Count")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



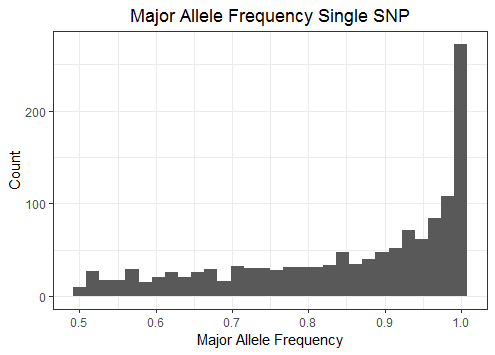
### 4.1.7 Plot of major allele frequency

#plot distribution of major allele frequency for single SNPs

ggplot()+geom\_histogram(data=singleSNP\_summary,aes(x=majAF))+

labs(title="Major Allele Frequency Single SNP", x="Major Allele Frequency", y="Count")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



## 4.2 Sample Data Summaries

### 4.2.1 Summary of sample genotype rate

Genotype rate per sample can be calculated for both single-SNP and haplotype results.

#calculate genotype rate per sample for single SNP data

sample\_genotypeRate\_singleSNP<-sampleGenoRate(polyGenResults\_singleSNP)

write.table(sample\_genotypeRate\_singleSNP,"sample\_genotypeRate\_singleSNP.txt",quote=FALSE,sep="\t")

#calculate genotype rate per sample for haplotypes

sample\_genotypeRate\_haplotypes<-sampleGenoRate(polyGenResults\_haplotypes)

write.table(sample\_genotypeRate\_haplotypes,"sample\_genotypeRate\_haplotypes.txt",quote=FALSE,sep="\t")

|  |  |
| --- | --- |
| sample | GenotypeRate |
| KEEK02\_0001 | 0.99 |
| KEEK02\_0002 | 0.99 |
| KEEK02\_0003 | 0.98 |
| KEEK02\_0004 | 0.98 |
| KEEK02\_0005 | 0.99 |
| KEEK02\_0006 | 0.98 |

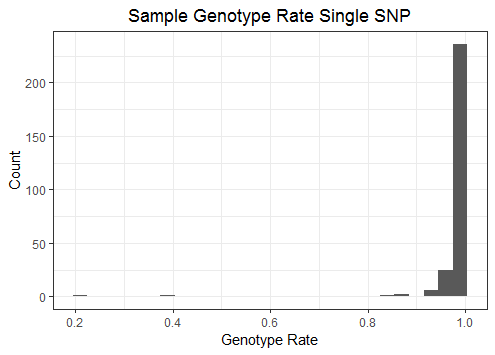
4.2.2 Plot of sample genotype rate

#plot genotype rate for single SNPs

ggplot()+geom\_histogram(data=sample\_genotypeRate\_singleSNP,aes(x=GenotypeRate),binwidth=0.03)+

labs(title="Sample Genotype Rate Single SNP", x="Genotype Rate", y="Count")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))

pol

### 4.2.3 Plot of sample genotype rate relative to reads per sample

Plotting the genotype rate per sample relative to reads per sample can reveal if samples have elevated missing data due to insufficient sequence. The number of reads in the plot below is taken from the Primer Probe Reads column of the GTscore\_individualSummary file that is output from AmpliconReadCounter.pl. These are the reads that aligned to primers and probes and were used for genotyping.

#load summary data containing total reads per sample

GTscore\_individualSummary<-read.delim("GTscore\_individualSummary.txt",header=TRUE,stringsAsFactors=FALSE)

#combine individual summary data with sample genotype rate

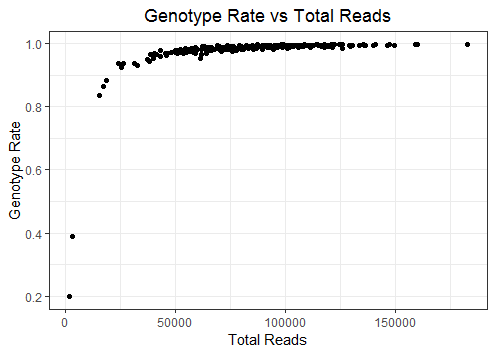
GTscore\_individualSummary<-merge(GTscore\_individualSummary,sample\_genotypeRate\_singleSNP,by.x="Sample",by.y="sample")

#plot genotype rate vs primer probe reads

ggplot()+geom\_point(data=GTscore\_individualSummary,aes(x=Primer.Probe.Reads,y=GenotypeRate))+

labs(title="Genotype Rate vs Total Reads", x="Primer Probe Reads", y="Genotype Rate")+

theme\_bw()+theme(plot.title=element\_text(hjust=0.5),plot.subtitle=element\_text(hjust=0.5))



## 4.3. Genotype Plots

Genotypes plots below are from plotGenotypes. This automatically generates plots for all loci with the option of two styles: allele ratio or taqman (scatterplot). Files are named by locus and the type of plot (ratio or scatter) is appended to the end of the filename to prevent over-writing if both types are saved to the same directory. Example plots are given below.

#Allele Ratio Plots

plotGenotypes(singleSNP\_locusTable, singleSNP\_alleleReads, polyGenResults\_singleSNP, type='ratio', savePlot="Y", saveDir="ratioPlots")

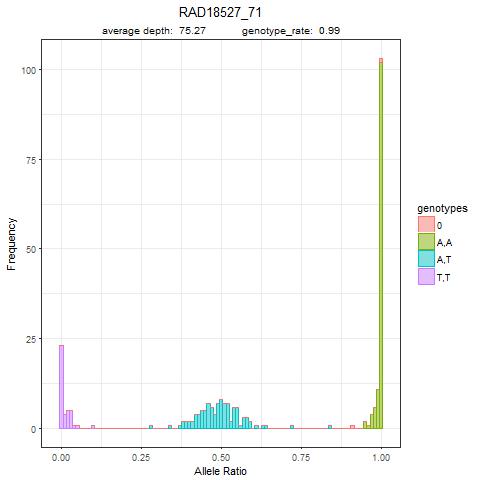
#Scatter Plots

plotGenotypes(singleSNP\_locusTable, singleSNP\_alleleReads, polyGenResults\_singleSNP, type='scatter', savePlot="Y", saveDir="scatterPlots")

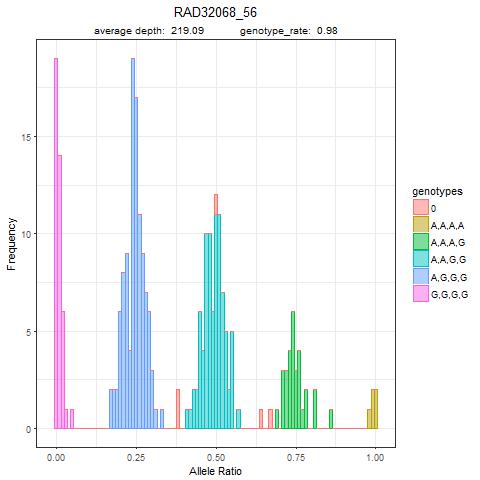
#NOTE: not produced for multi-SNP haplotypes

### 4.3.1 Allele ratio plots

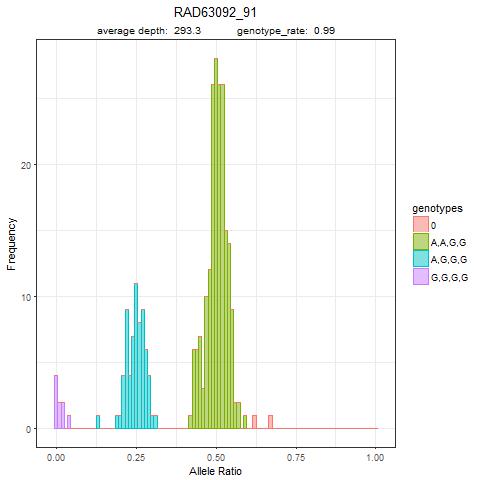
#### 4.3.1.1 Singleton locus



#### 4.3.1.2 Duplicate locus

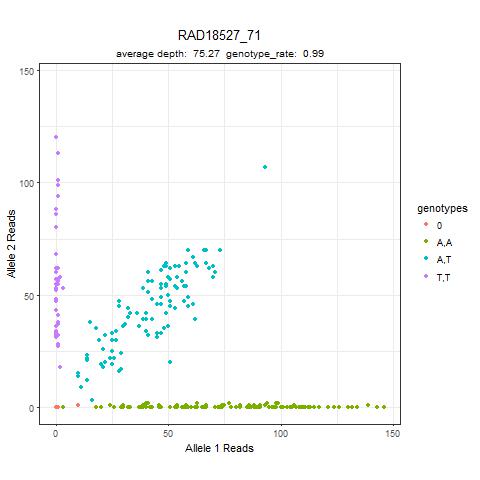


#### 4.3.1.3 Diverged duplicate locus

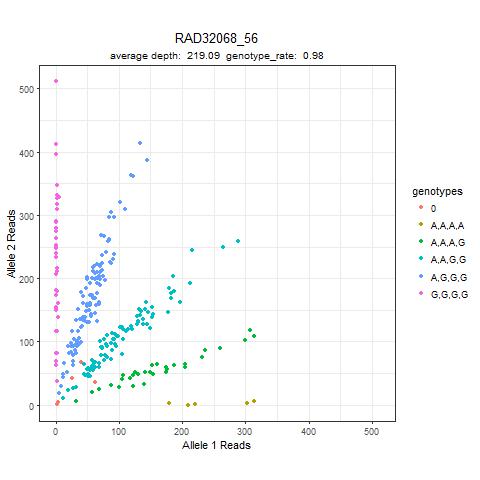


### 4.3.2 Scatter plots

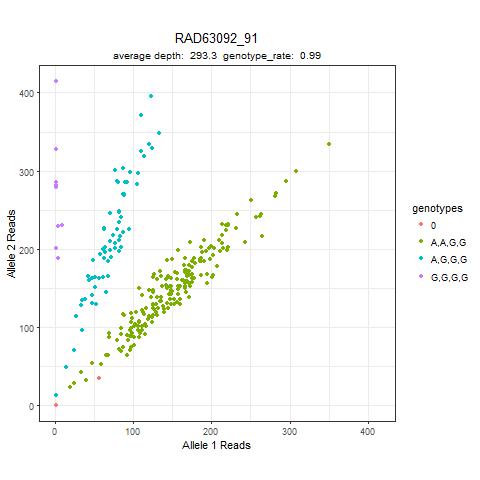
#### 4.3.2.1 Singleton locus



#### 4.3.2.2 Duplicate locus



#### 4.3.2.3 Diverged duplicate locus



# Step 5: Optional locus diagnostics—GTScore.R

##################

#Locus Diagnostics

##################

Diagnostic functions are provided to help in identifying patterns of sequence variation for loci. This can assist in identifying SNPs that were not accounted for in the initial probe design and in adjusting the *in-silico* probes to exclude off-target sequence from genotyping.

There are two approaches used by GTscore: First, plotting the number of nucleotide mismatches by position in the sequence data relative to the reference sequence for each locus, and second, performing sequence alignments using the program MSA (Bodenhofer *et al.* 2015). Plotting the mismatches by position is fast and easy to interpret except in the case of indels. The MSA alignments take longer to complete but facilitate the visualization of indels.

The first step for both methods is to get the sequences that aligned to each locus, this can be done based on primer alignments or primer-probe alignments using the script matchReads.pl.

## 5.1 Sequence alignment to each locus (Perl)

Input flags for this script are

*–p* a tab delimited file containing primer/probe information for each locus

*--files* a text file containing a list of .fastq sequence files to count reads from

--*matchType* the match type chosen for retaining sequences for each locus. Options are “primer” to retain sequences that match the primer for a locus or “primerProbe” to retain sequences that match both the primer and probe. primerProbe is the default if no option is specified.

Optional flags

--*prefix* optional prefix for output file names

>perl matchReads.pl -p *primerProbeFile.txt --*files *sampleFiles.txt* --matchType primerProbe [--prefix]

The default output file names are matchReads\_primerAligned.txt or matchReads\_primerProbeAligned.txt depending on which match type was chosen.

### 5.1.1 Output from matchReads.pl (sequences shortened for clarity)

Sample Locus Sequence Count

KAROL05\_0009 RAD18602\_20 ACAGGGTGATGCAA…TATGCCCCTGGGCCAGATCGGAAGAGCACACG 1

KAROL05\_0009 RAD18602\_20 ACAGGGTGATGCAA…TATGCCCCTGGGGCCAGATCGGAAGAGCACAC 1

KAROL05\_0009 RAD18602\_20 ACAGGGTGATGCAA…TATGCCCCTGGGCCAGATCGGAAGAGCACACG 20

KAROL05\_0010 RAD18602\_20 ACAGGGTGATGCCA…TATGCCCCTGGGCCAGATCGGAAGAGCACACG 1

KAROL05\_0010 RAD18602\_20 ACAGGGTGATGCCA…TATGCCCCTGGGCCAGATCGGAAGCACGTCTG 1

KAROL05\_0010 RAD18602\_20 ACAGGGTGATGCCA…TATGCCCCTGGGCCAGATCGGAAGAGCACACG 33

## 5.2 MSA alignment (GTscore.R)

MSA alignment of the matched reads can be done directly on the output from matchReads.pl using the alignMatchedSeqs command in GTscore. Probe sequence is incorporated into the alignment to visualize how probes may be affecting read counting, and reference sequence can optionally be included to facilitate comparison of the observed sequence with the target locus. The probe sequence is obtained from the primer-probe file originally used by AmpliconReadCounter.pl; the reference sequence file has a two column format where the first column is the locus name and the second column is the reference sequence for the locus amplicon. MSA will take a very long time to run if too many sequences are included for alignment so a minimum read threshold (minReads) is included as an option in the alignment command (suggest begin with minReads=20). In addition, a maximum of 100 unique sequences (ranked by number of reads) will be aligned. The maximum number of reads allowed for alignment can be changed using the maxAlignedSeqs option. Be sure the required perl package has been installed (Excel-Writer-XLSX).

#load input files for MSA alignment

#load optional reference sequence file  
referenceSeqs<-read.delim("ampliconRefSeqs.txt", header=TRUE, stringsAsFactors=FALSE)

Format of reference sequence file with SNPs represented by IUB codes. Longer sequences are truncated for clarity.

|  |  |
| --- | --- |
| Locus | refSeq |
| RAD18048 | ACGACAGGCTGTTGTCCAGGKTCACGCCAAGGTACTTTGC |
| RAD18527 | AGCAGGTGTTCTGTTCTGTCCCAGAGAGTTCAGAAGTAGAATATGTTTGTTATGTTGCTATCAATTATCCCWCCATT… |
| RAD18777 | AGGACTTGGAGTGGAGTGTGGTGTAAAAAWAAAAGCAAATTATTTCTTYATCACAGATAGACCTCCCCATCTTCAC… |

#load primer probe file  
primerProbes<-read.delim("PrimerProbeFile.txt", header=TRUE, stringsAsFactors=FALSE)

#load reads that contained primers  
primerMatchedReads<-read.delim("matchedReads\_primerAligned.txt", header=TRUE, stringsAsFactors=FALSE)  
#align reads that contained primers. Note: example includes optional referenceSeqs  
alignMatchedSeqs(primerProbes=primerProbes,matchedReads=primerMatchedReads,minReads=20, maxAlignedSeqs=100,type="primer",saveDir="MSA\_primerMatched")

#load reads that contained primers and probes  
primerProbeMatchedReads<-read.delim("matchedReads\_primerProbeAligned.txt", header=TRUE, stringsAsFactors=FALSE)  
#align reads that contained primers and probes. Note: example includes optional referenceSeqs  
alignMatchedSeqs(referenceSeqs, primerProbes=primerProbes, matchedReads=primerProbeMatchedReads,minReads=20, maxAlignedSeqs=100,type="primerProbe",saveDir="MSA\_primerProbeMatched")

### 5.2.1 MSA alignment saved as a conditionally formatted excel file*.*

Excel files are output for each locus within the directory specified by savDir. The first row is a consensus sequence based on the sequence alignments. This is followed by the probes, reference sequence if included, and then the sequence alignments. An example is included below.



## 5.3. Plotting Mismatches by Position (Perl and GTscore.R)

5.3.1 Calculate sequence mismatches by position (Perl)

Plotting the number of mismatches by position requires first comparing each matched sequence against the reference sequence and summing the number of mismatches for each nucleotide position. This is done with the perl script seqMismatchPositions.pl.

Input flags for this script are

*--amplicon* a fasta file containing the reference amplicon sequence for each locus

*--matchedSeqs* a tab delimited file containing sequences that matched each locus. This file is generated by matchReads.pl

--*matchType* the match type chosen for retaining sequences for each locus. Options are “primer” to retain sequences that match the primer for a locus or “primerProbe” to retain sequences that match both the primer and probe. primerProbe is the default if no option is specified.

Optional flags

--*prefix* optional prefix for output file names

>perl seqMismatchPositions.pl --amplicon *ampliconRefSeqs.fasta* *--*matchedSeqs *matchedReads\_primerAligned.txt* --matchType *primer* [--prefix]

The default output file names are mismatchPositions\_primer.txt or mismatchPositions\_primerProbe.txt depending on which match type was chosen. The output files will be used by the summarizeMismatches function in GTscore to generate plots of mismatches by position for each locus.

The output below was generated from reads that matched the primer and probe for the locus, the high number of mismatches in the middle of the sequence is due to the known SNP. Sequence and mismatches are truncated for clarity.

Locus RefSeq Mismatches

Ots\_RAD16850\_32 …CAGCCTTCTGCAGYTCAGCGTTGAT… …1,1,1,1,0,1,0,1,1,1,1,1,1,24450,0,1,1,1,1,1,1,0,1,1,1…

### 5.3.2. Plot of sequence mismatches by position (GTscore.R)

#load results from seqMismatchPositions.pl  
#primer matched sequences

mismatchPositionData\_primer<-read.delim("mismatchPositions\_primer.txt", header=TRUE, stringsAsFactors=FALSE)

#primer probe matched sequences

mismatchPositionData\_primerProbe<-read.delim("mismatchPositions\_primerProbe.txt", header=TRUE, stringsAsFactors=FALSE)

#generate plots of mismatches by position

#primer matched sequences

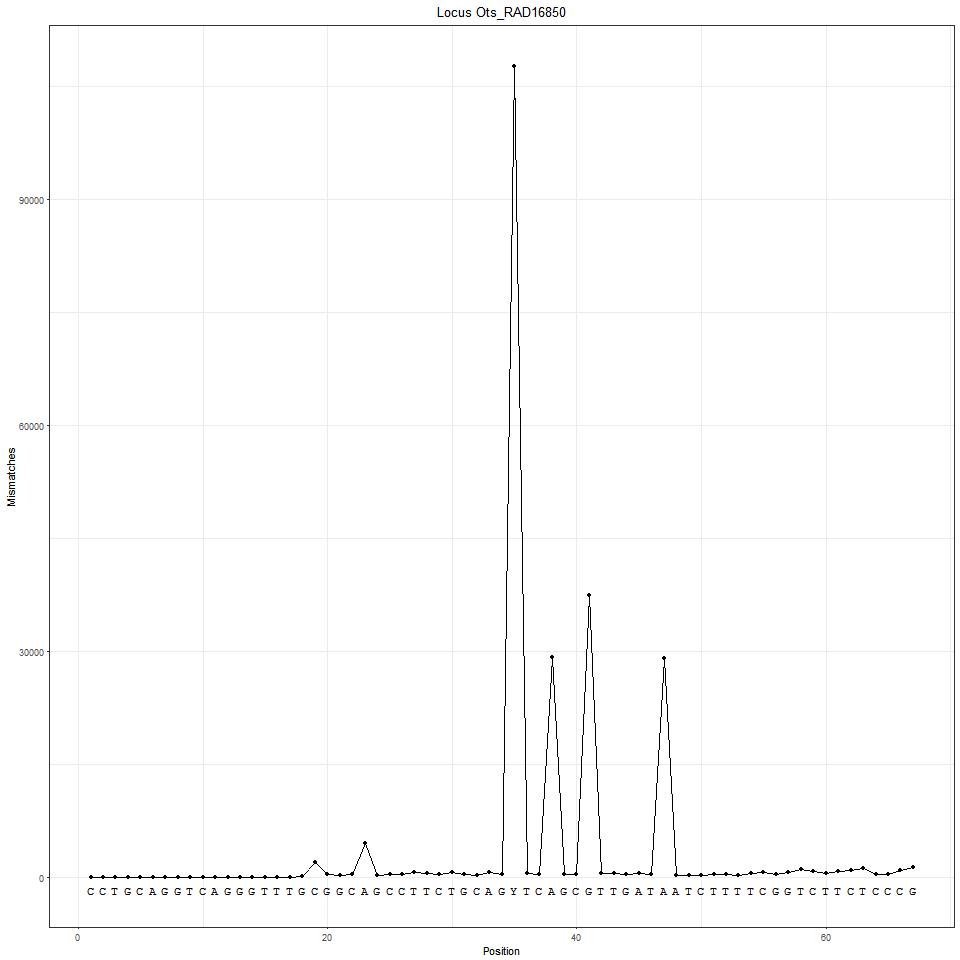
summarizeMismatches(mismatchPositionData\_primer,saveDir="mismatchPositionPlots\_primer")

#primer probe matched sequences

summarizeMismatches(mismatchPositionData\_primerProbe,saveDir="mismatchPositionPlots\_primerProbe")

### 

The data for this plot was generated using primer aligned reads for the locus without requiring the probe to match. The highest peak is at the known SNP, the other lower peaks may represent unknown SNPs or mismatches due to amplification of off-target sequence.



Acknowledgements:

We thank the Alaska Department of Fish and Game, especially Chris Habicht and Sara Gilk-Baumer, and members of the Seeb Laboratory for support and encouragement to complete the study. Funding was provided by the Alaska Department of Fish and Game Cooperative Agreement Number CT 170007906 and the Alaska Sustainable Salmon Fund Projects 44913 and 44914.

References:

Baetscher DS, Clemento AJ, Ng TC, Anderson EC, Garza JC (2018) Microhaplotypes provide increased power from short-read DNA sequences for relationship inference. *Mol Ecol Resour* **18**, 296-305.

Bodenhofer U, Bonatesta E, Horejs-Kainrath C, Hochreiter S (2015) msa: an R package for multiple sequence alignment. *Bioinformatics* **31**, 3997–3999.

Campbell NR, Harmon SA, Narum SR (2015) Genotyping-in-Thousands by sequencing (GT-seq): A cost effective SNP genotyping method based on custom amplicon sequencing. *Molecular Ecology Resources* **15**, 855-867.

Catchen J, Hohenlohe PA, Bassham S, Amores A, Cresko WA (2013) Stacks: an analysis tool set for population genomics. *Molecular Ecology* **22**, 3124-3140.

Dufresne F (2016) Don't throw the baby out with the bathwater: identifying and mapping paralogs in salmonids. *Molecular Ecology Resources* **16**, 7-9.

Kidd KK, Speed WC (2015) Criteria for selecting microhaplotypes: mixture detection and deconvolution. *Investigative Genetics* **6**, 1.

Limborg MT, Larson WA, Seeb LW, Seeb JE (2017) Screening of duplicated loci reveals hidden divergence patterns in a complex salmonid genome. *Molecular Ecology* **26**, 4509-4522.

McKinney GJ, Seeb JE, Seeb LW (2017a) Managing mixed-stock fisheries: genotyping multi-SNP haplotypes increases power for genetic stock identification. *Canadian Journal of Fisheries and Aquatic Sciences* **74**, 429-434.

McKinney GJ, Waples RK, Pascal CE, Seeb LW, Seeb JE (2018) Resolving allele dosage in duplicated loci using genotyping-by-sequencing data: A path forward for population genetic analysis. *Mol Ecol Resour* **18**, 570-579.

McKinney GJ, Waples RK, Seeb LW, Seeb JE (2017b) Paralogs are revealed by proportion of heterozygotes and deviations in read ratios in genotyping-by-sequencing data from natural populations. *Molecular Ecology Resources* **17**, 656-669.

Moran BM, Anderson EC (2018) Bayesian inference from the conditional genetic stock identification model. *Canadian Journal of Fisheries and Aquatic Sciences*.

Rousset F (2008) Genepop'007: A complete re-implementation of the Genepop software for Windows and Linux. *Mol Ecol Resour* **8**, 103-106.

Waples RK, Seeb LW, Seeb JE (2016) Linkage mapping with paralogs exposes regions of residual tetrasomic inheritance in chum salmon (*Oncorhynchus keta*). *Mol Ecol Resour* **16**, 17-28.