

Reading Raw MUGA and MegaMUGA Data

Daniel M. Gatti

03 October 2013

1 Extracting Genotypes and Intensities

This vignette explains how to read in the raw MUGA and MegaMUGA data as it is received from GeneSeek (Lincoln, NE). When genotyping Diversity Outbred (DO) mice, GeneSeek provides six files:

1. *_DNAReport.csv: Quality metrics per sample containing allele call rates and frequencies.
2. *_FinalReport.txt: The main data file containing allele calls and intensities.
3. *_LocusSummary.csv: Quality metrics for each marker.
4. *_LocusXDNA.csv: Combined sample and marker quality report.
5. Sample_map.txt: A list of sample IDs and plate locations.
6. SNP_Map.txt: A list of the SNPs assayed with genomic locations.

The two files required by DOQTL are *_FinalReport.txt and Sample_map.txt. They are expected to be in the same directory. Each data set from GeneSeek should also be in a separate directory.

Below, we load in example data and write it out to directories similar to what GeneSeek produces.

```
> library(DOQTL)

R/QTLRel is loaded

> wd = tempdir()
> data.dirs = paste(wd, "/DataSet", 1:2, sep = "")
> dir.create(data.dirs[1])
> dir.create(data.dirs[2])
> library(MUGAExampleData)
> data(FinalReport1)
> data(Samples1)
> write(FinalReport1, file = paste(data.dirs[1], "DataSet1_FinalReport.txt",
+                               sep = "/"))
> write(Samples1, file = paste(data.dirs[1], "Sample_Map.txt", sep = "/"))
> data(FinalReport2)
> data(Samples2)
> write(FinalReport2, file = paste(data.dirs[2], "DataSet2_FinalReport.txt",
+                               sep = "/"))
> write(Samples2, file = paste(data.dirs[2], "Sample_Map.txt", sep = "/"))
```

The *_FinalReport.txt files are tab delimited files that contain 11 columns.

1. SNP Name: marker ID
2. Sample ID: sample ID
3. Allele1 - Forward: allele call for one DNA strand
4. Allele2 - Forward: allele call for the other DNA strand
5. X: normalized X intensity
6. Y: normalized X intensity
7. GC Score: uncertain, ranges from 0 to 1, with a skew toward 1
8. Theta: X and Y intensity transformred to θ
9. X Raw: raw X intensity
10. Y Raw: raw Y intensity
11. R: X and Y intensity transformred to ρ

```
> read.delim(paste(data.dirs[1], "DataSet1_FinalReport.txt",
+               sep = "/"), nrows = 6 ,skip = 9)
```

	SNP.Name	Sample.ID	Allele1...Forward	Allele2...Forward	X	Y
1	backupJAX00000484	F01	A	G	0.548	0.546
2	backupJAX00003592	F01	G	G	0.007	1.152
3	backupJAX00004293	F01	T	C	0.557	0.396
4	backupJAX00005508	F01	G	G	0.019	0.814
5	backupJAX00012065	F01	T	G	1.055	0.816
6	backupJAX00012423	F01	G	G	0.001	0.633

	GC.Score	Theta	X.Raw	Y.Raw	R
1	0.6971	0.499	7415	5846	1.094
2	0.8788	0.996	1020	11655	1.158
3	0.7054	0.394	7497	4401	0.953
4	0.8860	0.985	1127	8385	0.833
5	0.7188	0.419	13503	8518	1.871
6	0.9637	0.999	888	6639	0.634

Each sample is listed sequentially. Note that the markers are not in genomic order in this file.

The Sample_Map.txt files contain a listing of the sample IDs and plate locations for each sample.

```
> read.delim(paste(data.dirs[1], "Sample_Map.txt", sep = "/"), nrows = 6)
```

Index	Name	ID	Gender	Plate	Well	Group	Parent1	Parent2	Replicate
1	1	F01 9376-F01	Unknown	P02723	A01	NA	NA	NA	NA
2	2	F02 9376-F02	Unknown	P02723	A02	NA	NA	NA	NA
3	3	F03 9376-F03	Unknown	P02723	A03	NA	NA	NA	NA
4	4	F04 9376-F04	Unknown	P02723	A04	NA	NA	NA	NA
5	5	F05 9376-F05	Unknown	P02723	A05	NA	NA	NA	NA

```

6      6  F06 9376-F06 Unknown P02723  A06   NA     NA     NA     NA
      SentrinxPosition
1 5532807102_R01C01
2 5532807102_R03C01
3 5532807102_R05C01
4 5532807102_R07C01
5 5532807102_R09C01
6 5532807102_R11C01

```

In practice, you will have one or more directories with genotyping results from GeneSeek. The genotype, X and Y intensity data can be extracted from these directories using the function `extract.raw.data()`. Place the path to the data directories in the `in path` argument, the output path in `out path` and specify whether the array is `muga` or `megamuga` in the `array` argument.

```
> extract.raw.data(in.path = data.dirs, out.path = wd, array = "muga")
```

This will create `x.txt`, `y.txt`, `geno.txt` and `call.rate.batch.txt` files in the output directory.

```
> list.files(path = wd, pattern = "txt$")
```

```
[1] "call.rate.batch.txt" "geno.txt"           "x.txt"
[4] "y.txt"
```

Optionally, you may filter out samples with low allele call rates. Samples with call rates below 90% often produce poor genome reconstructions. The function removes samples with call rates below the threshold (default = 0.9), writes out the `x.filt.txt`, `y.filt.txt` and `geno.filt.txt` files and returns the samples that were removed.

```
> removed = filter.samples(path = wd)
```

```
> removed
```

```

      sample call.rate
37      M12 0.4146931
38      M13 0.3863000
115     F69 0.8506494

                                batch
37 C:\\Users\\dgatti\\AppData\\Local\\Temp\\Rtmpw1Dejg\\DataSet1
38 C:\\Users\\dgatti\\AppData\\Local\\Temp\\Rtmpw1Dejg\\DataSet1
115 C:\\Users\\dgatti\\AppData\\Local\\Temp\\Rtmpw1Dejg\\DataSet2

```

Three samples had call rates below 0.9.

Finally, you may perform batch normalization on the intensity files. Currently, this simply subtracts the median intensity from each batch. Future improvements may be made to these methods. You must provide the SNP locations in the `snps` argument. We obtain these from the JAX FTP site.

```
> load(url("ftp://ftp.jax.org/MUGA/muga_snps.Rdata"))
> batch.normalize(path = wd, snps = muga_snps)
```

This will write out the files `x.filt.norm.txt` and `y.filt.norm.txt`. You may then use these as input into DOQTL's genome reconstruction pipeline.