

1 How to use mQTL.NMR package

1.1 Introduction

mQTL.NMR provides a complete QTL analysis pipeline for 1H-NMR metabolomic data. Distinctive features include normalisation using various approaches, peak alignment using RSPA approach, dimensionality reduction and mQTL mapping for animal crosses and humans cohorts. We provide below an application example showing the use of mQTL.NMR package on real-world datasets.

1.2 Data format

Raw data files should be formatted according to the targeted type of analysis (mQTL or mGWA analysis). However, this step can be skipped if the data files are already formatted by the user. In order to format the datafiles into the supported format by the mQTL.NMR package (comma-delimited format *csvs*), the `format_mQTL()` function should be used with the input files in the following format:

Metabolomic raw data					Genomic raw data				
	S1	...	S167	S168		gnf01.004.225	...	D19Mit53	D19Mit35
ppm1=0.400000	0.00995	...	0.00743	0.00427	Chrom	1	...	19	19
ppm2=0.400550	0.01032	...	0.00739	0.00422	cM	0	...	43	53
ppm3=0.401100	0.01066	...	0.00762	0.00418	Mb	7261408	...	45205220	55440868
:	:	...	:	:	S1	h	...	a	h
ppm13553=9.999150	0.01089	...	0.00843	0.00407	S2	a	...	ND	b
ppm13454=9.999700	0.01106	...	0.00862	0.00398	:	:	...	:	:
					S167	ND	...	h	ND
					S168	b	...	b	a

Additional data				
	S1	...	S167	S168
sex	1	...	1	1
pgm	1	...	1	1

- *Si*: ID of the *i*th sample
- *ppmi*: *i*th metabolomic variable (chemical shift)
- *Chrom*: chromosome
- *cM*: genomic position in CentiMorgan
- *Mb*: genomic position in nucleotide
- *a*: homozygote AA; *b*: homozygote BB; *h*: heterozygote AB
- *pgm*: parental grandmother (0/1)

Figure 1: Format datafiles by using `format_mQTL()`

It is preferable that the user create a new directory to load the demonstration dataset. Data files can be formatted as follows:

```
> # Download raw data files
> library(mQTL.NMR)
> load_datafiles()
> format_mQTL(phenofile, genofile, physiodat, cleandat, cleangen)
```

```
[1] "Start formatting the datafile /tmp/Rtmpw2Pnjm/Rinst15273b65c867/mQTL.NMR/extdata/phenofile"
```

An example of *csvs* resulted files are shown below:

Metabolomic raw data							Genomic raw data				
ppm_0.4	...	ppm_9.999150	ppm_9.999700	sex	pgm	id	id	gnf01.004.225	...	D19Mit53	D19Mit35
0.00995	...	0.00743	0.00427	1	1	S1		1	...	19	19
0.01032	...	0.00739	0.00422	1	1	S2		0	...	43	53
0.01066	...	0.00762	0.00418	1	1	S3	S1	H	...	A	H
:	...	:	:	:	:	:	S2	A	...	NA	B
0.01089	...	0.00843	0.00407	1	1	S167	:	:	...	:	:
0.01106	...	0.00862	0.00398	1	1	S168	S167	NA	...	H	NA
							S168	B	...	B	A

Figure 2: Datafiles formatted in csvs

1.3 Normalization of metabolomic data

The mQTL.NMR package implements several approaches to perform the normalization of metabolomic data. We show below an example of the use of normalisation approach 'constant sum' (CS)

```
> # Constant Sum normalization
> nmeth<-'CS'
> normalise_mQTL(cleandat,CSnorm,nmeth)
[1] "Start constant sum normalisation:"
```

1.4 Alignment of metabolomic data

The mQTL.NMR package uses the Recursive Segment-wise Peak Alignment (RSPA) approach to align positions of NMR spectral peaks with respect to a reference spectrum picked up either automatically or manually.

```
> align_mQTL(CSnorm,aligdat)
[1] "...Automatic selection of a reference spectrum..."
[1] ""
[1] "...Returning aligned spectra..."
```

1.5 Dimensionality reduction

The mQTL.NMR package offers mainly two approaches in order to reduce the number of traits considered for genetic mapping analysis: statistical recoupling of variables ("SRV") and binning ("bin") approaches. An example of using SRV approach is given below:

```
> met="rectangle" # choose the statistical summarizing measure ("max","sum","trapez",...)
> pre_mQTL(aligdat, reducedF, RedMet="SRV",met, corrT=0.9)
[1] "Processing a 168 x 1454 matrix with minsize of 10 and a correlation Threshold of 0.9
[1] "length of events: 66"
```

```
[1] "Xcluster" "168"      "66"      "66"
[1] "balise length:" "3"
[1] "Found 64 clusters"
[1] "dataset: data file /tmp/Rtmpw2Pnjm/Rinst15273b65c867/mQTL.NMR/extdata/reducedF.txt ger
```

1.6 Quantitative trait locus mapping

The mQTL.NMR package enables association studies for different species, from animal crosses to human cohorts. Below an example of mQTL mapping on mouse crosse:

```
> results<- list() # a list to stock the mQTL mapping results
> nperm<- 0 # number of permutations if required
> results<-process_mQTL(reducedF, cleangen, nperm)

--Read the following data:
      168 individuals
      30 markers
      67 phenotypes
--Cross type: f2
[1] "TotM is 44 and np is 64"
[1] 6.000000 1.265767
[1] 12.000000 3.069176
[1] 18.000000 3.069176
[1] 24.000000 3.069176
[1] 30.000000 3.069176
[1] 36.000000 3.069176
[1] 42.000000 3.069176
[1] 48.000000 3.069176
[1] 54.000000 3.069176
[1] 60.000000 19.76617
[1] "SRV ehk in this area was 19.77 for 3.73905 ppm at c9.loc65"
```

1.7 Visualization of results

mQTL mapping results can be shown in several plots as follows (Figure 1):

```
> post_mQTL(results)

[1] "Post processing"
pdf
  2
```

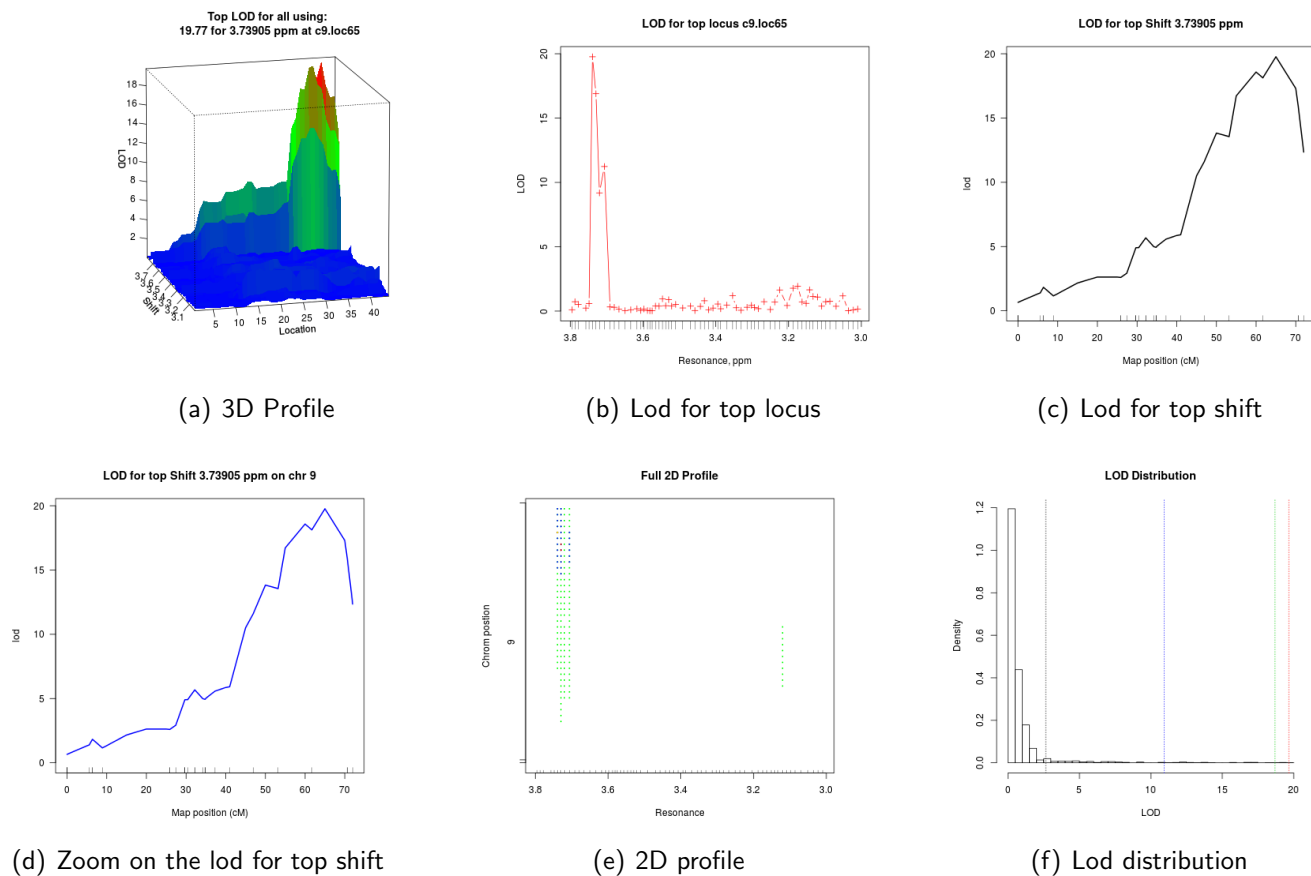


Figure 3: figures generated by post_mQTL()

1.8 Summary of results

A summary of the results can be obtained in a table as follows:

```
> summary_mQTL(results,rectangle_SRV,T=8)
```

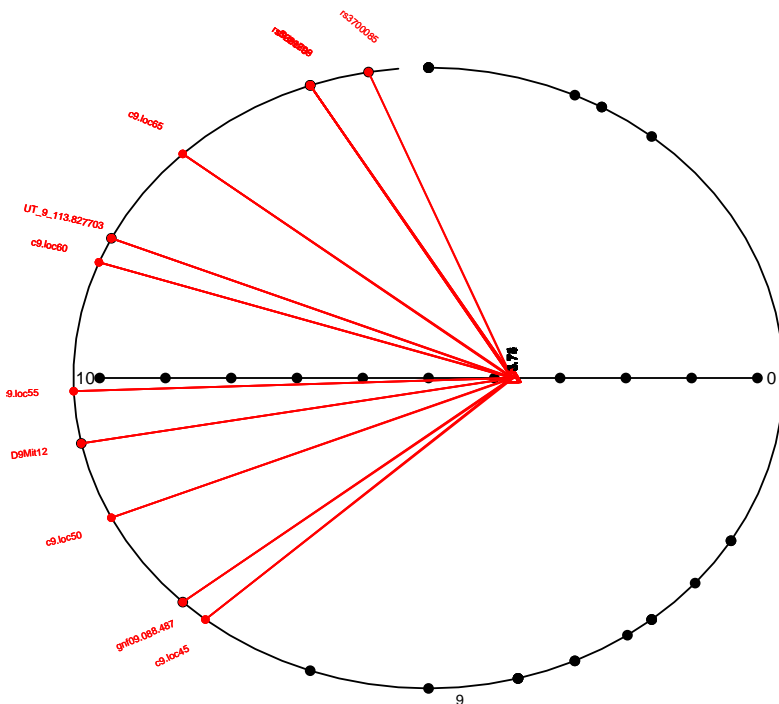
```
[1] "Summarize single runs /tmp/Rtmpw2Pnjm/Rinst15273b65c867/mQTL.NMR/extdata/rectangle_SRV"
```

	Min	Max	ppm	chr	pos	lod
c9.loc55	3.70110	3.71155	3.706325	9	55.00000	13.08101
UT_9_113.827703	3.71540	3.72420	3.719800	9	61.70003	9.21992
c9.loc60	3.72475	3.73410	3.729425	9	60.00000	19.38129
c9.loc65	3.73465	3.74345	3.739050	9	65.00000	19.76617

1.9 Circular genome-metabolome plot

mQTL.NMR package provides a functionality to visualize the obtained mQTL mapping results. The user can start first by visualizing the plot of the whole results of mQTL mapping. For optimal visualization, the user can then plot the results for one or several chromosomes.

```
> circle_mQTL(results, T=8,spacing=0)
```



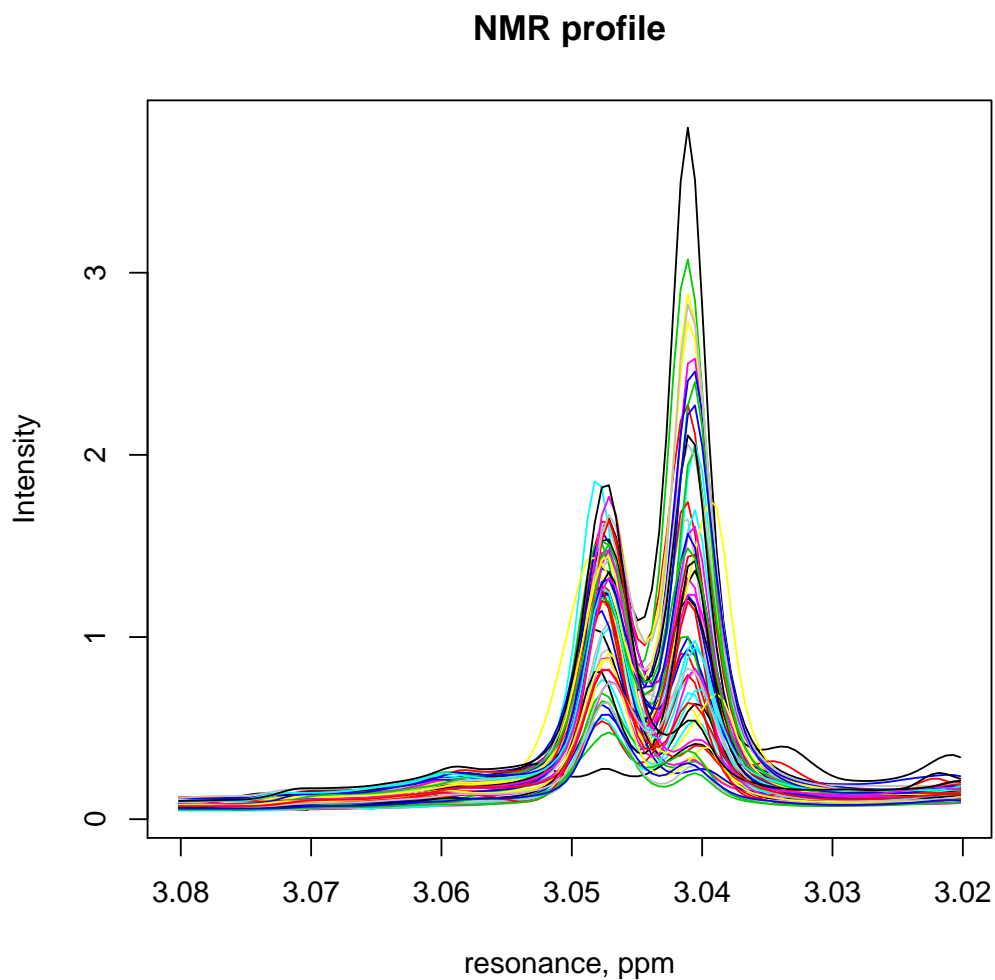
1.10 Structural assignment and metabolite identification

mQTL.NMR package provides several visualization tools to help the user in peak annotation and metabolite identification. In particular, users can visualize identified SRV clusters on the chemical shift axis and automatically generates a plot showing the top ranked SRV cluster in the association study in order to identify the corresponding metabolite.

1.10.1 Plot a region of NMR profiles

Displays a region of the NMR profiles by specifying a starting and ending point

```
> simple.plot(file=aligdat,lo=3.02,hi=3.08,k=1:60,title="NMR profile")
```

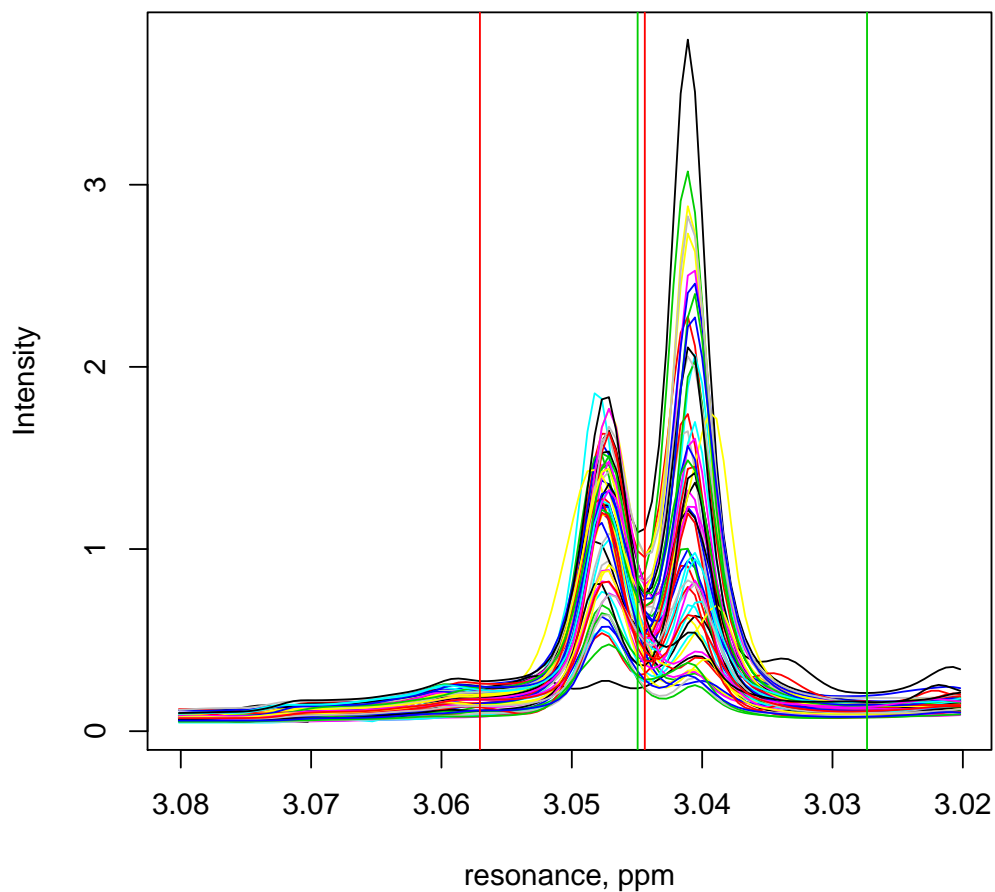


1.10.2 Visualization of SRV clusters

Displays the NMR profile with the clusters identified by SRV defined by horizontal lines on the profile (green=min cluster border, red=max cluster border)

```
> SRV.plot(file1=aligdat,file2=rectangle_SRV,lo=3.02,hi=3.08,k=1:60,title="Clusters plot")
```

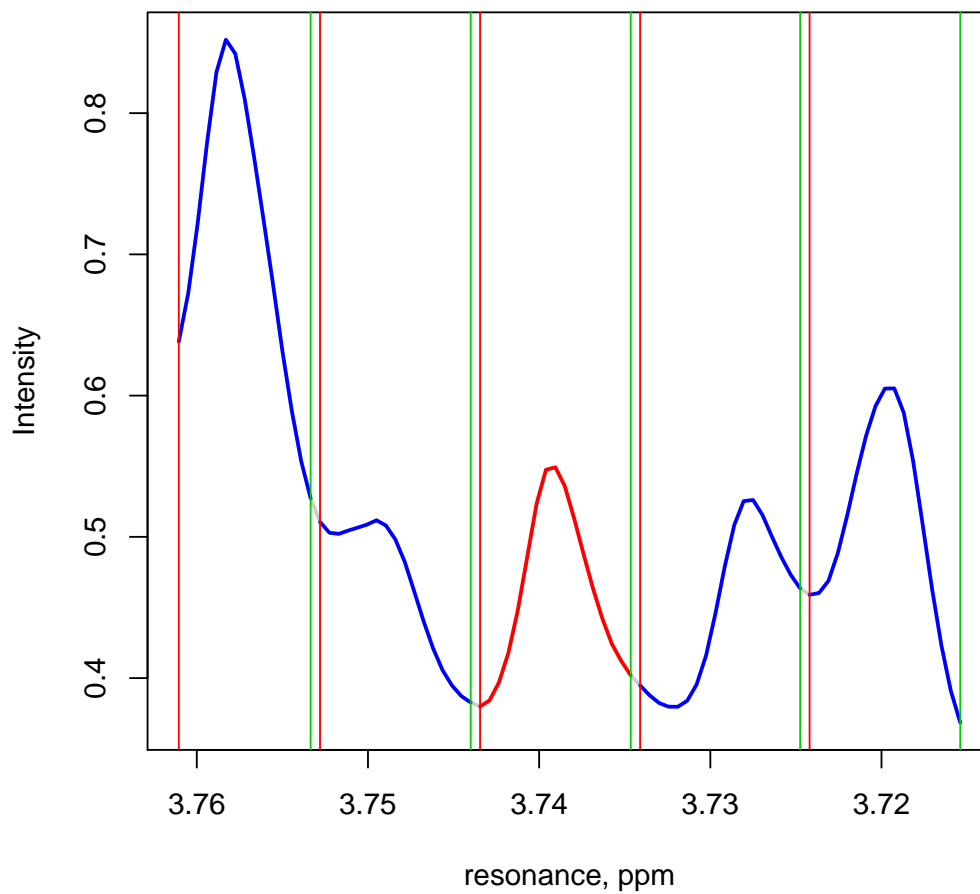
Cluster plot



1.10.3 Top SRV clusters

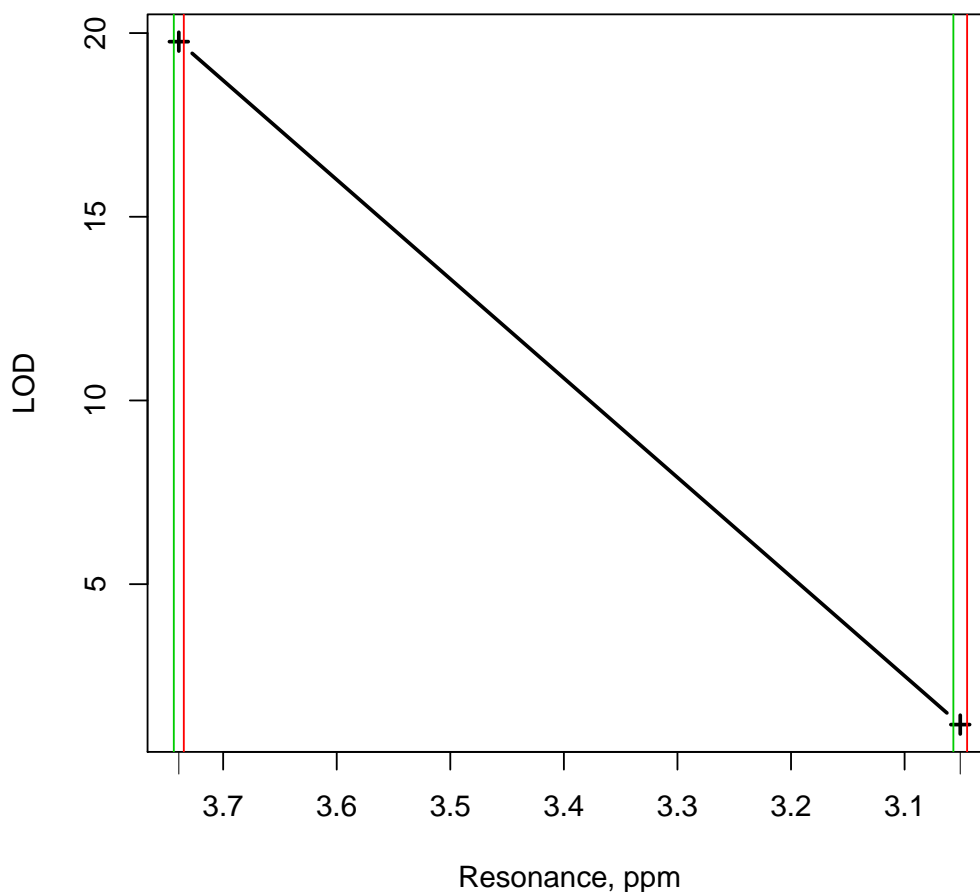
Generates the plot of SRV cluster corresponding to the top association score with the two right and left clusters to be used for metabolite identification

```
> Top_SRV.plot(file1=aligdat,file2=rectangle_SRV,results=results,met=met,intMeth="mean")
```

Top cluster at ppm 3.73905 +/- 2 clusters

```
> SRV_lod.plot(results,rectangle_SRV,T=1)
```


LOD for locus c9.loc65 on all SRV clusters



The manual page on `mQTL.NMR` has a number of additional examples that show how applicable functions can be used.

2 Session Information

The version number of R and packages loaded for generating the vignette were:

```
R version 3.3.0 (2016-05-03)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.4 LTS
```

locale:

[1] LC_CTYPE=en_US.UTF-8	LC_NUMERIC=C	LC_TIME=en_US.UTF-8
[4] LC_COLLATE=C	LC_MONETARY=en_US.UTF-8	LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8	LC_NAME=C	LC_ADDRESS=C

```
[10] LC_TELEPHONE=C          LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
other attached packages:
```

```
[1] mQTL.NMR_1.6.0
```

```
loaded via a namespace (and not attached):
```

```
[1] MASS_7.3-45      BiocStyle_2.0.0 parallel_3.3.0  tools_3.3.0     qtl_1.39-5
```

```
[6] outliers_0.14    GenABEL_1.8-0
```