Estimate eQTL networks using qpgraph

***Inma Tur*[1,3]*, Alberto Roverato*[2] *and Robert Castelo*[1]**

May 2, 2019

1. Universitat Pompeu Fabra, Barcelona, Spain.
2. Università di Bologna, Bologna, Italy.
3. Now at Kernel Analytics, Barcelona, Spain.

1 Introduction

In this vignette we introduce the functionality of the *qpgraph* package to estimate eQTL networks from genetical genomics data. To meet the space and time constraints in building this vignette within the *qpgraph* package, we are going to simulate genetical genomics data instead of using a real data set. For this purpose, we will use the functionality described in another vignette from this package, entitled "Simulating molecular regulatory networks using qpgraph". If you use the approach and functions described in this vignette in your own research, please cite the following article:

> Tur, I., Roverato, A. and Castelo, R. Mapping eQTL networks with mixed graphical Markov models. *Genetics*, 198(4):1377-1393, 2014.

2 Simulating an eQTL network and data from it

We are going to simulate an eQTL network in the following steps:

1. Load the necessary packages.

```
> library(GenomeInfoDb)
> library(qtl)
> library(qpgraph)
```

2. Simulate a genetic map using the R/CRAN package *qtl*, consisting of nine chromosomes, being 100 cM long with 10 markers equally spaced along each of them, no telomeric markers and no X sexual chromosome.

```
> map <- sim.map(len=rep(100, times=9),
+                n.mar=rep(10, times=9),
+                anchor.tel=FALSE,
+                eq.spacing=TRUE,
+                include.x=FALSE)
```

3. Create a first empty eQTL network as an empty *eQTLcross* object using the previously simulated genetic map.

4. Simulate an eQTL network consisting of 50 genes, where half of them have one *cis*-acting (local) eQTL, there are 5 eQTL *trans*-acting (distant) on 5 genes each and each gene is connected to 2 other genes (default). Each eQTL has an additive effect of $a = 2$ and each gene-gene association has a marginal correlation $\rho = 0.5$. We seed the random number generator to enable reproducing the same eQTL network employed in this vignette. A dot plot of the simualted eQTL associations is displayed in Figure 1.

```
> set.seed(12345)
> sim.eqtl <- reQTLcross(eQTLcrossParam(map=map, genes=50, cis=0.5, trans=rep(5, 5)),
+                        a=2, rho=0.5)
```

```
> plot(sim.eqtl, main="Simulated eQTL network")
```

5. Simulate genotyping and expression data for 100 individuals from this eQTL network. We seed again the random number generator to enable random sampling the same data.

```
> set.seed(12345)
> cross <- sim.cross(map, sim.eqtl, n.ind=100)
> cross
  This is an object of class "cross".
  It is too complex to print, so we provide just this summary.
    Backcross

    No. individuals:    100

    No. phenotypes:     50
    Percent phenotyped: 100

    No. chromosomes:    9
        Autosomes:      1 2 3 4 5 6 7 8 9

    Total markers:      90
    No. markers:        10 10 10 10 10 10 10 10 10
    Percent genotyped:  100
    Genotypes (%):      AA:50.7  AB:49.3
```

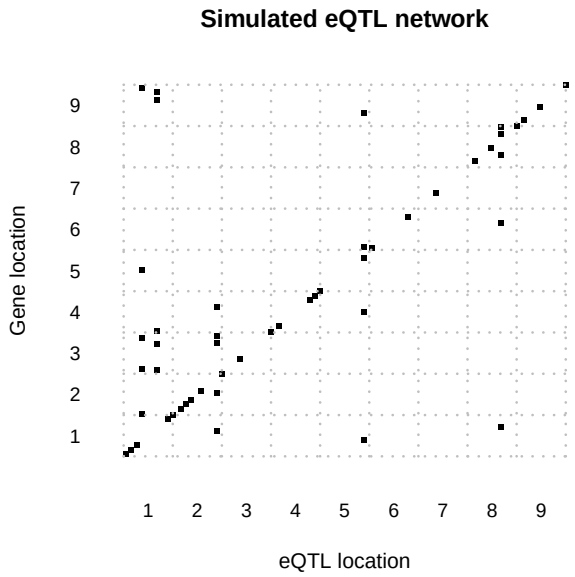

Figure 1: Dot plot of eQTL associations in a simulated eQTL network

3 Estimating an eQTL network from genetical genomics data

Here we briefly illustrate how to estimate an eQTL network from genetical genomics data stored as a R/CRAN *qtl* *cross* object. This object is the one we have simulated before.

To use this functionality we need to provide an annotation for the genes we have in our data. This is retrieved from the simulated eQTL network object.

```
> annot <- data.frame(chr=as.character(sim.eqtl$genes[, "chr"]),
+                     start=sim.eqtl$genes[, "location"],
+                     end=sim.eqtl$genes[, "location"],
+                     strand=rep("+", nrow(sim.eqtl$genes)),
+                     row.names=rownames(sim.eqtl$genes),
+                     stringsAsFactors=FALSE)
```

For later visualization purposes, we also need a physical map, which we calculate assuming a constant Kb/cM rate of 5. We scale the gene annotations and chromosome lengths also using this Kb/cM rate. We create a *Seqinfo* object storing the chromosome lengths of this simulated genome.

```
> pMap <- lapply(map, function(x) x * 5)
> class(pMap) <- "map"
> annot$start <- floor(annot$start * 5)
> annot$end <- floor(annot$end * 5)
> genome <- Seqinfo(seqnames=names(map), seqlengths=rep(100 * 5, nchr(pMap)),
+                   NA, "simulatedGenome")
```

Estimate eQTL networks using *qpgraph*

The entire estimation procedure can be performed in the following steps.

1. Create a paramter object of class *eQTLnetworkEstimationParam*.

```
> param <- eQTLnetworkEstimationParam(cross, physicalMap=pMap,
+                                     geneAnnotation=annot, genome=genome)
```

2. Calculate all marginal associations between markers and genes.

```
> eqtlnet.q0 <- eQTLnetworkEstimate(param, ~ marker + gene, verbose=FALSE)
> eqtlnet.q0

eQTLnetwork object:
  Genome: simulatedGenome
  Input size: 90 markers 50 genes
  Model formula: ~marker + gene
```

3. Obtain a first estimate of the eQTL network by selecting associations at FDR < 0.05.

```
> eqtlnet.q0.fdr <- eQTLnetworkEstimate(param, estimate=eqtlnet.q0,
+                                       p.value=0.05, method="fdr")
> eqtlnet.q0.fdr

eQTLnetwork object:
  Genome: simulatedGenome
  Input size: 90 markers 50 genes
  Model formula: ~marker + gene (q = 0,)
  G^(0,): 140 vertices and 1710 edges corresponding to
          822 eQTL and 888 gene-gene associations meeting
          a fdr-adjusted p-value < 0.05
          and involving 50 genes and 81 eQTLs
```

Display a comparison of the dot plot of the simulated eQTL associations with the ones estimated by marginal associations at FDR < 0.05. The result is shown in Figure 2.

```
> par(mfrow=c(1, 2))
> plot(sim.eqtl, main="Simulated eQTL network")
> plot(eqtlnet.q0.fdr, main="Esiimated eQTL network")
```

4. Calculate non-rejection rate values with $q = 3$ between markers and genes.

```
> eqtlnet.q0.fdr.nrr <- eQTLnetworkEstimate(param, ~ marker + gene | gene(q=3),
+                                           estimate=eqtlnet.q0.fdr, verbose=FALSE)
> eqtlnet.q0.fdr.nrr

eQTLnetwork object:
  Genome: simulatedGenome
  Input size: 90 markers 50 genes
  Model formula: ~marker + gene | gene (q = 0,3)
  G^(0,3): 140 vertices and 1710 edges corresponding to
           822 eQTL and 888 gene-gene associations meeting
           a fdr-adjusted p-value < 0.05
           and involving 50 genes and 81 eQTLs
```

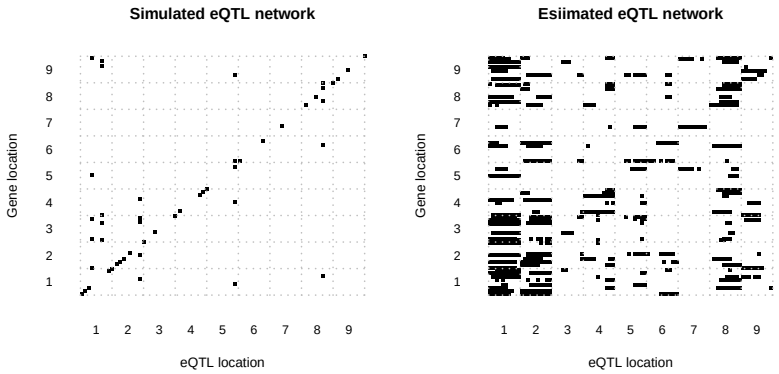

Figure 2: **Dot plots of eQTL associations in a simulated eQTL network (left) and in an estimated eQTL network (right) selecting marginal associations at FDR $< 5\%$**

5. Obtain a second estimate of the eQTL network by selecting associations at FDR < 0.05 and with non-rejection rate value $\epsilon < 0.1$.

```
> eqtlnet.q0.fdr.nrr <- eQTLnetworkEstimate(param, estimate=eqtlnet.q0.fdr.nrr,
+                                           epsilon=0.1)
> eqtlnet.q0.fdr.nrr

eQTLnetwork object:
  Genome: simulatedGenome
  Input size: 90 markers 50 genes
  Model formula: ~marker + gene | gene (q = 0,3)
  G^(0,3): 140 vertices and 350 edges corresponding to
           223 eQTL and 127 gene-gene associations meeting
           a fdr-adjusted p-value < 0.05,
           a non-rejection rate epsilon < 0.10
           and involving 50 genes and 75 eQTLs
```

Display a comparison of the dot plot of the simulated eQTL associations with the ones estimated by marginal associations at FDR < 0.05 and non-rejection rates meeting a cutoff $\epsilon < 0.1$. The result is shown in Figure 3.

```
> par(mfrow=c(1, 2))
> plot(sim.eqtl, main="Simulated eQTL network")
> plot(eqtlnet.q0.fdr.nrr, main="Esiimated eQTL network")
```

Examine the median number of eQTLs per gene.

```
> eqtls <- alleQTL(eqtlnet.q0.fdr.nrr)
> median(sapply(split(eqtls$QTL, eqtls$gene), length))

[1] 5
```

6. Note that while we have simulated at most one eQTL per gene, we have currently estimated a median of 6 eQTLs per gene. This leads to the horizontal patterns in the dot plot where multiple markers in the same chromosome target the same gene and are

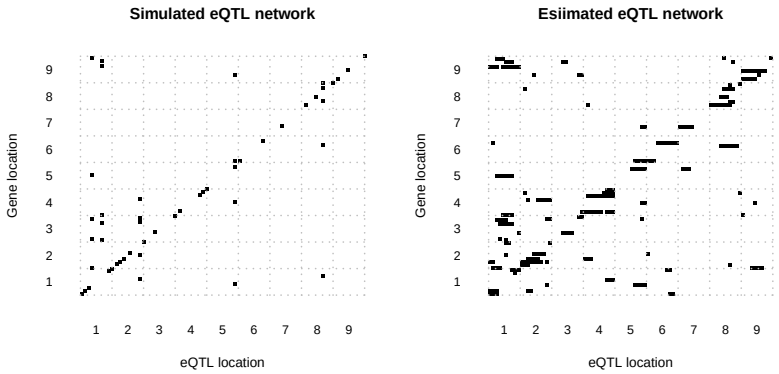

Figure 3: **Dot plots of eQTL associations in a simulated eQTL network (left) and in an estimated eQTL network (right) selecting marginal associations at FDR $< 5\%$ and non-rejection rate meeting a cutoff $\epsilon < 0.1$**

the result of independently mapping eQTLs that are tagging the same causal one. To remove these redundant eQTL associations we perform a forward selection procedure at a nominal significance level $\alpha < 0.05$, as follows:

```
> eqtlnet.q0.fdr.nrr.sel <- eQTLnetworkEstimate(param, estimate=eqtlnet.q0.fdr.nrr,
+                                               alpha=0.05)
> eqtlnet.q0.fdr.nrr.sel

eQTLnetwork object:
  Genome: simulatedGenome
  Input size: 90 markers 50 genes
  Model formula: ~marker + gene | gene (q = 0,3)
  G^(0,3,*): 140 vertices and 209 edges corresponding to
             82 eQTL and 127 gene-gene associations meeting
             a fdr-adjusted p-value < 0.05,
             a non-rejection rate epsilon < 0.10,
             a forward eQTL selection significance level alpha < 0.05
             and involving 50 genes and 51 eQTLs
```

In Figure 4 we can see a comparison between the dot plots of the simulated eQTL network and the final estimate obtained by first selecting marginal associations at FDR < 0.05, discarding those that did not meet a NRR cutoff $\epsilon < 0.1$ and further performing a forward selection procedure at a significance level $\alpha < 0.05$ among eQTLs within the same chromosomes targeting a common gene. Observe that in this final eQTL network estimate many of the redundant eQTL associations have been effectively discarded.

```
> par(mfrow=c(1, 2))
> plot(sim.eqtl, main="Simulated eQTL network")
> plot(eqtlnet.q0.fdr.nrr.sel, main="Esiimated eQTL network")
```

Finally, the *qpgraph* package provides functionality to ease the visualization of the eQTL network, going beyond the dot plot to display not only eQTL associations, but also the gene-gene associations where one of the two genes has at least one eQTL. This functionality is based on the concept of hive plot (Krzywinski et al., 2012) and has been adapted from the

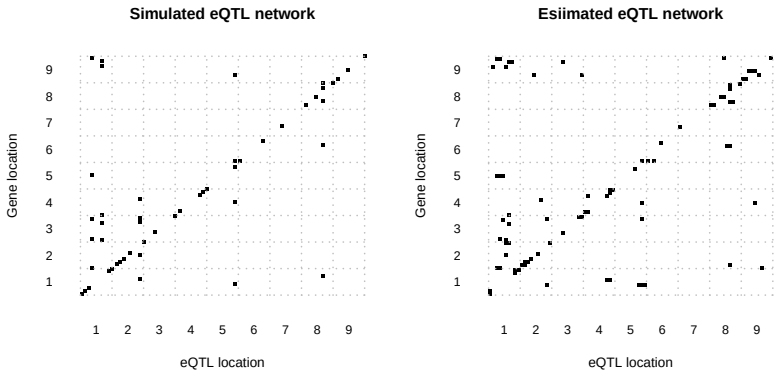

Figure 4: **Dot plots of eQTL associations in a simulated eQTL network (left) and in an estimated eQTL network (right) selecting marginal associations at FDR $< 5\%$ and non-rejection rate with $\epsilon < 0.1$**

code provided by the *HiveR* package (Hanson, 2014) to display eQTL networks. It uses the *grid* package for plotting purposes and the code below illusrates how to produce the hive plots in Figure 5, which shows a hive plot per chromosome of the final estimated eQTL network. The fact that in hive plots vertex (node) positions are fixed eases the task of comparing them. In our context, this facilitates the comparison of the genetic control of gene expression across chromosomes.

```
> library(grid)
> library(graph)
> grid.newpage()
> pushViewport(viewport(layout=grid.layout(3, 3)))
> for (i in 1:3) {
+   for (j in 1:3) {
+     chr <- (i-1) * 3 + j
+     pushViewport(viewport(layout.pos.col=j, layout.pos.row=i))
+     plot(eqtlnet.q0.fdr.nrr.sel, type="hive", chr=chr)
+     grid.text(paste0("chr", as.roman(chr)), x=unit(0.05, "npc"),
+               y=unit(0.9, "npc"), just="left")
+     grid.text("genes", x=unit(0.08, "npc"), y=unit(0.1, "npc"), just="left", gp=gpar(cex=0.9))
+     grid.text("all chr", x=unit(0.92, "npc"), y=unit(0.2, "npc"), just="right", gp=gpar(cex=0.9))
+     grid.text("genes", x=unit(0.92, "npc"), y=unit(0.1, "npc"), just="right", gp=gpar(cex=0.9))
+     grid.text("markers", x=unit(0.5, "npc"), y=unit(0.95, "npc"), just="centre", gp=gpar(cex=0.9))
+     popViewport(2)
+   }
+ }
```

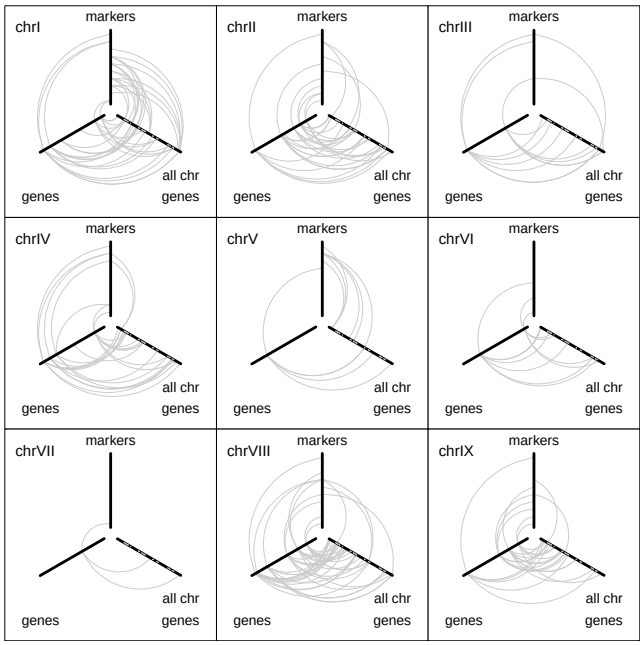

Figure 5: Hive plots of an eQTL network estimated from simulated data, involving only connected components with at least one eQTL association
For each chromosome, the hive plot shows three axes, where markers and genes are ordered from the center according to their genomic location. Vertical and left axes represent the chromosome in the corresponding plot, while the right axis represents the entire genome alternating black and gray along consecutive chromosomes. Edges between genes axes correspond to gene-gene associations.

4 Session information

```
> toLatex(sessionInfo())
```

- R version 3.6.0 (2019-04-26), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 18.04.2 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.9-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, stats4, utils
- Other packages: BiocGenerics 0.30.0, GenomeInfoDb 1.20.0, IRanges 2.18.0, S4Vectors 0.22.0, graph 1.62.0, qpgraph 2.18.0, qtl 1.44-9

- Loaded via a namespace (and not attached): AnnotationDbi 1.46.0, Biobase 2.44.0, BiocManager 1.30.4, BiocParallel 1.18.0, BiocStyle 2.12.0, Biostrings 2.52.0, DBI 1.0.0, DelayedArray 0.10.0, GenomeInfoDbData 1.2.1, GenomicAlignments 1.20.0, GenomicFeatures 1.36.0, GenomicRanges 1.36.0, Matrix 1.2-17, R6 2.4.0, RCurl 1.95-4.12, RSQLite 2.1.1, Rcpp 1.0.1, Rgraphviz 2.28.0, Rsamtools 2.0.0, SummarizedExperiment 1.14.0, XML 3.98-1.19, XVector 0.24.0, annotate 1.62.0, assertthat 0.2.1, biomaRt 2.40.0, bit 1.1-14, bit64 0.9-7, bitops 1.0-6, blob 1.1.1, compiler 3.6.0, crayon 1.3.4, digest 0.6.18, evaluate 0.13, hms 0.4.2, htmltools 0.3.6, httr 1.4.0, knitr 1.22, lattice 0.20-38, magrittr 1.5, matrixStats 0.54.0, memoise 1.1.0, mvtnorm 1.0-10, pkgconfig 2.0.2, prettyunits 1.0.2, progress 1.2.0, rlang 0.3.4, rmarkdown 1.12, rtracklayer 1.44.0, stringi 1.4.3, stringr 1.4.0, tools 3.6.0, xfun 0.6, xtable 1.8-4, yaml 2.2.0, zlibbioc 1.30.0

References

Hanson, B. A. (2014). *HiveR: 2D and 3D Hive plots for R*. R/CRAN pkg. ver. 0.2-27.

Krzywinski, M., Birol, I., Jones, S. J., and Marra, M. A. (2012). Hive plots–rational approach to visualizing networks. *Brief Bioinform*, 13(5):627–644.