Chapter 13

Visualization of Results from Systems Genetics Studies in Chromosomal Context

Karen Y. Oróstica and Ricardo A. Verdugo

Abstract

This chapter describes methods currently available for visualizing results from systems genetics experiments. Here, we abstract from the statistical methods used for genetic mapping, which are dependent on the specific resource being used, i.e. F2, RILs, or outbred populations among others. We use a public dataset with results from a mouse eQTL experiment for three examples of visualization: genome-wide dot plots of marker-by-gene association, karyotype-like plots, and circos plots. Dot plots give a first overview of the results from eQTL mapping, allowing detecting genome-wide patterns of *cis*- and *trans*-genetic association to transcription level. Karyotype-like plots provide chromosomal context and allow integrating multiple tracks of information in a single plot. Circos plots can, in addition, display long-range interactions to provide an overview of genetic connectivity at the genome level. All examples are developed and explained using R code, an open-source language with powerful statistical and graphical capabilities. The principles reviewed here, however, can be applied with other software options, organisms, and to any type of molecular phenotype that can be assigned to a genomic position.

Key words Tools, Systems Genetics, xQTL, Visualization, Chromosomal context

1 Introduction

The systems genetics approach can be a powerful tool for better modeling complex heritable traits [1]. By this method, a forward genetics approach is used both to mapping loci with genetic variation affecting organismal-level phenotypes as well as molecular phenotypes that may be relevant mechanistic intermediaries [2]. Current technology allows quantitatively surveying hundreds or thousands of biological molecules such as DNA sequence variations, epigenetic marks, and levels of transcripts, proteins, and metabolites. The underlying hypothesis is that by measuring molecular phenotypes that are under tighter genetic regulation than organism-level

Electronic supplementary material: The online version of this chapter (doi: 10.1007/978-1-4939-6427-7_13) contains supplementary material, which is available to authorized users.

Klaus Schughart and Robert W. Williams (eds.), *Systems Genetics: Methods and Protocols*, Methods in Molecular Biology, vol. 1488, DOI 10.1007/978-1-4939-6427-7_13, © Springer Science+Business Media New York 2017

phenotypes, it may be possible to increase prediction accuracy for the target trait [3] as well as to produce testable hypothesis of causality that will suggest molecular mechanisms underlying phenotypes [4]. Different names have been given to QTLs associated to each of these types of molecular phenotypes. We can refer to them generically as xQTL. Because the most commonly profiled such phenotypes are gene expression levels measured by microarrays, we will develop examples for such expression QTL (eQTL).

Regardless of the genetic resource being used, the process can be partitioned in two mayor stages: (1) identification of loci associated to phenotypes, both molecular and physiological, and (2) modeling of networks of loci and phenotypes with the objective to discover association of the type *interaction* or *causal relation*. Here, we develop methods that are relevant for interpreting the results of the first part and that may suggest sensible models for the second part. Particularly, we will focus on displaying the results of a systems genetics experiment in a genomic layout. The aim is that by visualizing results in a chromosomal context, one may:

1. Identify errors in previous analysis steps.
2. Reveal relative proportion of cis- versus trans-genetic regulation.
3. Visualize hotspots of genetic variants regulating phenotypes.
4. Suggest genome-wide patterns of long-range genetic regulation or interactions.

2 Methods

2.1 Type of Data and Principles of Visualization

Any molecular phenotype with a genome position can be visualized in chromosomal context. The results from genetic mapping of these types of phenotypes can be summarized as a table containing loci as genomic regions or punctual positions and some sort of identification for the phenotypes. The data format will be similar to the one shown in Table 1.

In the examples that follow, we will use data from a study using Systems Genetics to genetically dissect hybrid sterility in male mice from a hybrid zone between two European subspecies of the house mouse, i.e. *M. musculus musculus* and *M. musculus domesticus* [5]. They investigated two phenotypes: relative testis weight (testis weight/body weight) and genome-wide testis gene expression pattern. Genetic mapping was done with the offspring of wild-caught mice by a GWAS approach using the Mouse Diversity Genotyping Arrays with 600K SNPs (Affymetrix, Santa Clara, CA). Abundance of 22K transcripts was profiled with the Whole Mouse Genome Microarray (Agilent, Santa Clara, CA). Genetic variation in gene expression was mapped both at the individual transcript level and for the first principal component of expression (PC1).

Table 1
BED format for data tracks

Probe.ID	Chr	Start	End	QTL.ID	QTL.Chr	QTL.Pos
A_52_P803348	19	18758768	18758827	JAX00339958	12	84287993
A_52_P486260	1	135806993	135806934	JAX00268096	1	137262920
A_52_P262997	11	115999665	115999606	JAX00322816	11	116044467
A_52_P333953	2	31658860	31658919	JAX00214519	19	21159325
A_52_P634829	7	111452887	111452828	JAX00232405	7	86614739
…						

Here we will use the following data obtained from the publication:

1. Significant phenotype-by-SNP associations.
2. Significant expression PC1-by-SNP associations.
3. Significant transcript-by-SNP associations.
4. Significant genetic interactions (SNP pairs) for testis expression PC1.

These data were obtained from the publication's material [5] or by personal communication. Probe-to-transcript mapping were obtained from the GEO website (http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL7202). All data are provided as electronic supplementary material accompanying this chapter.

2.2 Software Needed

The vizualization examples that we develop in this chapter are performed in the R programing language [6]. The software is open-source and is freely available from http://cran.r-project.org. Karyotype-like plots and circos plots demonstrated here require the chromPlot and OmicCircos R packages, respectively, which can be obtained from the Bioconductor repository (http://www.bioconductor.org). We assume basic knowledge of the R language and focus on the specifics for creating plots of genomic data. Readers unfamiliar with R are encouraged to consult online introductory material on installing, running, and using R (www.r-project.org). For brevity, we explain only the most important commands and concepts behind each visualization. Scripts with the full code and all necessary data files to replicate the graphs are provided in the accompanying website.

2.3 Genome-Wide Dot Plots

Dot plots give a quick overview of the distribution of molecular phenotypes and their respective genetic determinants of variation. The genomic position of the phenotype, transcripts in our example, is plotted according to its position (y-axis) and to the position of the QTL or most significant marker in a given locus (x-axis). This type of plot can be performed using R's built-in functionality, i.e. it does not require any extra R–package.

In order to display genomic positions across chromosomes on the same axis, we sort chromosomes by number, leaving sexual chromosomes last, and then we calculate cumulative positions. The units used for genomic position can vary depending on the application. Here we will use mega bases (one million basepairs) denoted by Mb, but one could use basepairs (bp) or centimorgans (cM) instead, for instance.

Dot plots are created with the plot() function in R as follows:

```
> plot(X, Y, axes=FALSE, frame.plot=TRUE,
pch=".")
```

where,

X	vector of positions of eQTL in Mb (start or mid position when segments)
Y	vector of positions for transcripts in Mb (molecular phenotype)
axes	we provide the logical value FALSE to the axes argument to omit axes
frame.plot	draw a square to delimit the plotting area
pch	sets the character used to represent data points. Use "." for dot plots.

The initial '>' symbol is not input but is used to indicate that the following text is R code. The X and Y are the names of R objects holding the vectors of cumulative positions in Mb for all SNPs and transcripts, respectively. This can be done with the cumsum() function in R. By default, plot()draws axes on the bottom and left of the plots, with automatic tickmarks indicating a scale of values in Mb. However, we omit this so that they can be drawn with chromosome names instead:

```
> axis(side=1, at=ticks, labels=labs, las=2)
> axis(side=2, at=ticks , labels=labs,
las=1)
```

Each line draws one axis,

side	axis placement, 1: bottom, 2: left
ticks	numeric vector with mid position in Mb of each chromosome
labs	character vector with chromosome names
las	argument to indicate text alignment of labels, 1: horizontal, 2: perpendicular

And lastly we can add vertical gray lines to delimit the borders of each chromosome:

```
> abline(v=seps, col="lightgray")
> abline(h=seps, col="lightgray")
```

With these two commands, we draw vertical and horizontal lines respectively, where,

seps numeric vector with the start position for each chromosome

col argument to set the line's color

The resulting plot for the eQTLs reported by [5] is shown in Fig. 1a. Several aspects of the dataset are revealed by this plot. For comparison, we produced the same plot for a different dataset from an eQTL mapping in the BXD panel of recombinant strains reported by [7] (Fig. 1b). See the 01.Dotplot_outcross.R and 02.Dotplot_BXD.R scripts in the electronic supplementary materials for the full list of commands that created these plots. The amount of transcript-by-SNP associations that are detected depends on several factors, such as power, significance level, and mapping strategy. Here we are comparing two very different experiments. One is a GWAS on 185 mice generated from 63 mating pairs involving 37 unrelated females and 35 unrelated males and the other is a single-marker QTL mapping on 37 individuals from different recombinant strains from the BXD cross. Expression was measured in two different tissues, testis and liver respectively, but significance in both studies was determined by sample permutation. Therefore, although results from both studies are not directly comparable, it is useful to display side-by-side for illustration purposes.

As seen in Fig. 1, the GWAS revealed widespread genetic variation affecting transcript abundance, evidenced by a much denser plot than for BXD . This may be due to the large genetic variation present in the hybrid zone were the mice were sampled [5]. We

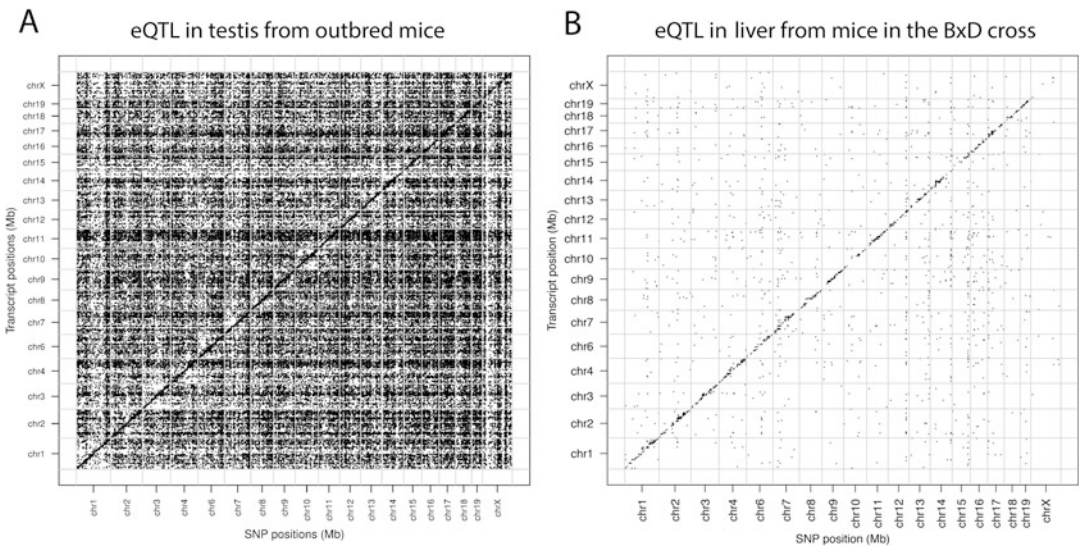

Fig. 1 Genome-wide scatter plots of eQTLs in system genetics studies. Genomic position of transcripts is shown by eQTL's SNP position in Mb. Only the most significant SNP per chromosome for each transcript is shown. (**a**) Results from eQTL mapping in the wild-derived male mice from [5]. (**b**) Results from eQTL mapping in the BxD recombinant inbred strains [7]

also evidence the presence of vertical and horizontal stripes of points in the GWAS, whereas BxD strains showed only vertical stripes. The latter are of more interest since they can represent regions of the genome with genetic variation affecting a large number of transcripts. Such loci may represent tans-eQTL shared among multiple transcripts and if colocalizing with QTL for the phenotype of interest, they may suggest a group of transcripts of interest to include in models to molecularly dissect such phenotype. By contrast, horizontal stripes may represent the effect of genome-wide differences between individuals, such as close relatedness or population structure from recent admixture. The study by [5] controlled for these effects when testing association for the organismal phenotype, relative testis weight, by including the kinship matrix as a random covariate. However, no such treatment was reported for molecular phenotypes, possibly because of computational restriction. The genome-wide dot plot reveals these effects and helps to orient further analyses by focusing on the regions and effects of most interest.

A second feature that is readily apparent from dot plots is the presence of a diagonal stripe of points, representing genetic loci that are in proximity to the gene they regulate. These are commonly referred to as cis-eQTL. By comparison, the GWAS experiment resulted in a lower proportion of cis- versus trans-eQTL than the BxD strains. cis-eQTL tend to have stronger effects and therefore are easier to detect, becoming dominant in experiments with low power of detection. Thus, detecting a diagonal line in genome-wide scatter plots provide good positive control; its absence should trigger checking for errors in data analyses or in the annotation of microarray probes for transcripts and SNPs.

Although a powerful tool, dot plots are limited in the number of data features they can show. Varying colors among data points can be used to add information about association significance (e.g. Fig. 3 in [7]) or some other property associated to transcripts and/or genetic loci. However, adding more layers of information can make the graph difficult to interpret. In addition, interaction effects among genetic loci or among molecular phenotypes cannot be represented in these plots. Therefore, other types of graphs are needed to inspect such properties of the data.

2.4 Karyotype-like Plots

Karyotypes are arrays of condensed metaphasic chromosomes used to detect large rearrangements, aneuploidies, polysomy, polyploidy, etc. One can use similar plots to display any type of genomic data along condensed chromosomes to provide location context. Location of data relative to important chromosomal structures such as centromere, telomeres, or heterochromatin is readily shown. The body of the chromosomes can be used to represent idiograms of G bands or some other type of data. Genomic data may also be represented on either side of the body. Depending on the type of data to be displayed, one could use histograms, points,

connected lines, or rectangular segments, among others. These types of plots have the advantage of being able to display large and diverse data in a way that results familiar to most biologists.

Karyotype-like plots have been used in the context of systems genetics. In the past, we have used these plots to reveal preferential cis- over trans-regulation from QTL on gene expression levels in mouse congenic strains [8]. Here, following from the previous section, we use data from the paper by [5] to illustrate this type of visualization. We use the R-package chromPlot to display the results of eQTL mapping in chromosomal context. The chromPlot package can be obtained from Bioconductor (http://www.bioconductor.org) or from our website (http://genomed.med.uchile.cl/software).

In order to create a karyotype-like plot, one must first load the chromPlot package by:

```
> library(chromPlot)
```

We then load data necessary for drawing idiograms , which include human and mouse in the chromPlot package. For a mouse genome:

```
> data(mm10_gap)
> data(mm10_cytoBandIdeo)
```

These commands load data frames (tables) with the above names containing the genomic locations for centromeres and cytogenetic bands respectively. Then we must load the location of the genomic features that we want to display as histograms. Here, we will use:

refseq	RefSeq genes in the mouse genome (assembly mm10 from UCSC)
annot	RefSeq genes with probes in the microarray
eqtl_phenos	RefSeq genes with at least one eQTL from [5]
QTLs	QTLs for relative testis weight (testis/body weight) from [5]
eQTLs	Expression eQTLs from [5]

All of these tables contain the columns called Chrom, Start, and End with the name of the chromosome, and the start and end nucleotides for each genomic element. The annot and eqtl_phenos tables are created as subsets of the refseq table. This allows creating stacked barplots for number of genes falling in three categories: with an eQTL, tested but not associated to any SNP, and not tested (because not present in the array). Thus, the pattern of genes falling in these categories can be inspected genome-wide. The Probe-to-RefSeq mapping was taken from the GPL7202 table deposited at the GEO repository by the authors of [5].

The karyotype-like plot shown in Fig. 2 is created by:

```
> chromPlot(gaps = mm10_gap, bands=mm10_cy-
toBandIdeo, annot1= refseq, annot2=annot,
annot3=eqtl_phenos, segment=eQTLs_track,
segment2=QTLs, chr = c(13, 17, "X"), scale.
```

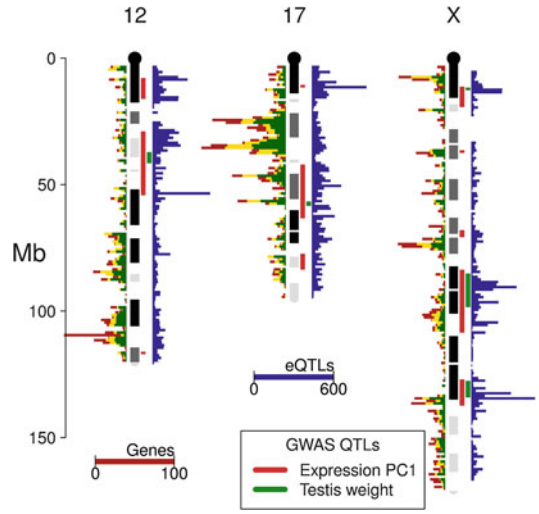

Fig. 2 Karyotype-like plot for three mouse chromosomes. Three types of data are shown. *Blue horizontal lines* on the body of chromosomes are typed SNP makers. The histogram on *left side* chromosomes represents number of genes in bins of 1 Mb. In *yellow* is the fraction of genes whose expression was tested by microarrays and *green* denotes genes whose expression was associated with at least one SNP in the GWAS from [5]. The color segments to the *right* of each chromosome represent GWAS regions for different phenotypes differentiated by color

```
title="Genes", segmentDesc="eQTL", segment-
2Desc=" GWAS QTLs", colSegments2="red",
cex=2, figCols=3)
```

This command tells the chromPlot function to use the mm10 coordinate system and creates histograms for genes to the left of each chromosome. The segment and segment2 arguments also create histograms if genomic features are smaller than the bin size (1 Mbp by default) or if too numerous to be plotted individually. Otherwise, they are plotted as colored lines to the right of chromosomes. In our example, eQTLs are plotted as histograms and testis weight QTLs as lines. Other arguments define text to be printed in the figure and change default colors or font size (cex). The figCols argument defines how many chromosomes are plotted per row (by default, chromPlot creates two rows).

The three chromosomes shown in Fig. 2 were selected because they contain a large number of eQTL and contain at least one QTL for relative testis weight, a fertility-related phenotype. A similar plot for all chromosomes is included in the electronic supplementary material (Figure S1.tif) and was created by the 03.Karyotype-like plots.R script. It becomes evident from Fig. 2 that the distribution of eQTLs along chromosomes is not uniform. Although, such conclusion would require a statistical test, this type of graph can suggest such pattern. One potential reason for this may be differences in gene density. By plotting a histogram of number of genes on the left side of chromosomes, we see some relation, especially on chromosome 17. However, this cannot

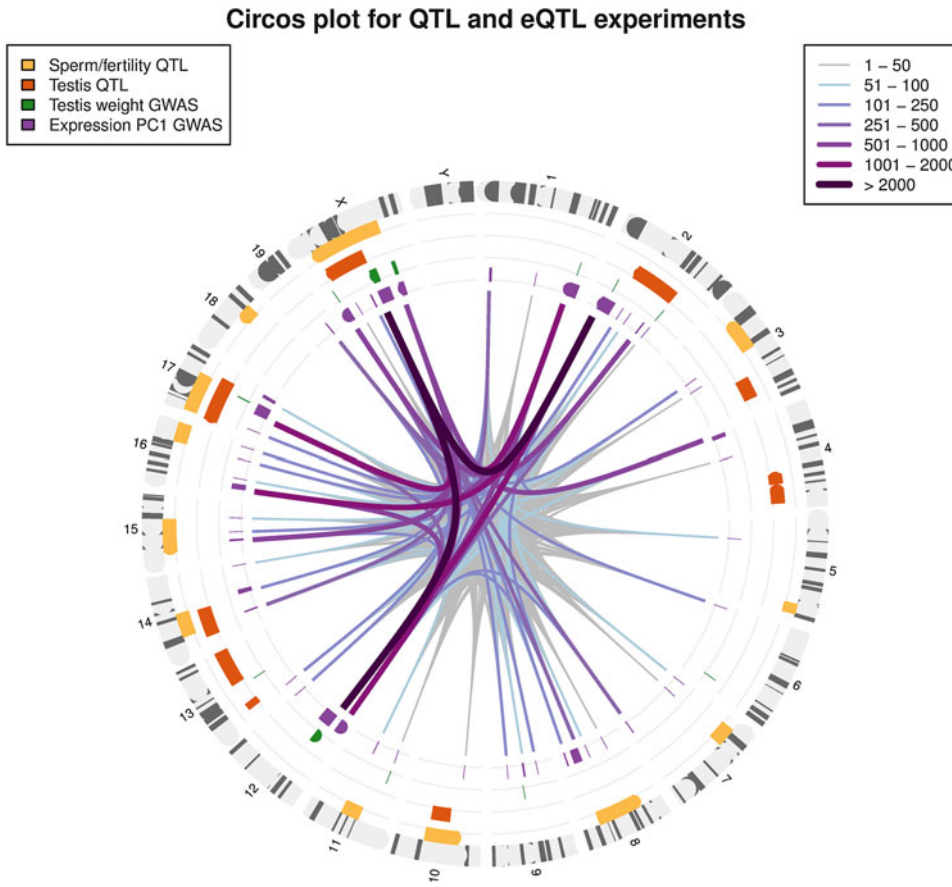

Fig. 3 Circos plot for QTL and eQTL experiments. *Yellow boxes* represent the QTLs for sterility phenotypes and *orange boxes* show QTL for relative testis weight as reported by [9, 10]. *Green boxes* indicate significant GWAS regions for relative testis weight and *purple boxes* denote significant GWAS regions for testis expression PC1 identified by [5]. The most inner track shows regions with significant genetic interactions reported in [5]. The color and line weight denote the number of significant pairwise interactions between SNPs for each region pair

explain completely the clustering in eQTLs. In fact, the largest number of eQTLs on chromosome 13 is present in a valley between two peaks of gene density. On chromosome X, the largest concentrations of eQTLs are around three QTL for testis weight. The region of chromosome X harboring the central QTL was the one explaining most overall variation in the testis transcriptome (PC1) as reported by [5]. This colocalization is highly unlikely by chance and is suggestive of presence of genetic factors on chromosome X that are driving low fertility, possibly by regulating the expression of a network of genes with eQTL in the region. Genes with cis-eQTL located in the testis QTL regions are obvious candidates for master regulators of such networks. Although beyond the scope of this chapter, formal testing of casual relations among genes should be applied in order to decompose the topology of the network and to identify regulator candidates [4].

2.5 Circos Plots

The visualization approaches revised so far are appropriate for inspecting global and local patterns of distribution of xQTL in the genome. However, they are not a good option for representing long-range interactions between genomic regions. By interaction, we mean any type of relationship of biological significance between elements that have different genomic position. For instance, gene-by-gene interactions may represent protein-protein interactions, a causal dependency of the expression of one gene on the expression of another gene, or a genotype-by-genotype effect from a statistical model on a quantitative phenotype. The type of interaction of interest will depend on the particular question under investigation. However, many times, visualizing such interactions genome-wide may reveal patterns that are suggestive of new hypotheses to be tested.

In Fig. 3, we have used a circos plot to summarize results from the eQTL experiment by Turner and Harr in [5]. Just as Karyotype-like plots, circos plots allow multiple tracks of information. In addition to the relative testis weight and PC1 QTLs, we included QTLs for several sperm fertility traits and relative testis weight taken from the literature as different tracks [9, 10]. Colocalization of phenotypic and expression QTLs suggest that both types of phenotypes are affected by the same genetic loci.

In addition to scanning for eQTLs, the authors tested for genetic interactions among SNPs in different chromosomes that significantly explained variation in gene expression. Interaction effects were evaluated for pairs of eQTL regions. The circos plot in Fig. 3 presents significant interactions as lines connecting loci. The color and width of the lines is proportional to the number of pairs of SNPs that interact in each region pair. By inspecting this plot, it becomes evident that loci don't act independently but that their effects are better modeled jointly and that gene expression may explain in part these interactions.

Here, we demonstrate how to create the circos plot in Fig. 3 by using the OmicCircos package in R. First, we load the OmicCircos functions:

```
> library(OmicCircos)
```

We then load the data tables as previously using the read.csv() function (not shown). The tracks in Fig. 3 consist of:

SpermFert QTLs for several sperm fertility traits from [9, 10]
testisWeight QTLs for relative testis weight from [9, 10]
TWgwas Regions spanning multiple SNPs associated at GWAS for testis weight in [5]
PC1gwas Regions spanning multiple SNPs associated at GWAS for Expression PC1 in [5]
RegionsPair Interaction effects between PC1gwas region pairs in [5]

The first four tracks have the BED format previously described. However, the RegionsPairs track is a table linking two QTL regions as shown in Table 2.

In order to draw the circos plot in Fig. 3, one must first create a system of coordinates on which each track of information will be plotted. This can be done as follows:

```
> plot(x=c(1, 800), y=c(1, 800), type = "n",
axes = FALSE, xlab = "", ylab = "", main =
"Circos plot for QTL and eQTL experiments");
```

The plot() function opens a graphics window and creates a new plotting area. Because, type="n", no plotting is actually performed and by axes=FALSE, the *x* and *y* axes are not drawn. However, this command has the effect of creating a plotting area between (1,1) at the bottom-left corner to (800,800) at the top-right corner. In this way, we obtain a square graphics window, which is necessary to draw a circle (instead of an ellipse). The range of values provided to *x* and *y* are arbitrary. However, the "radius" values later provided to circos() must be within this range (see bellow).

To add a track, one must decide at what distance from the center the data circle is to be plotted (the circle radius). This is set by the R argument of circos(). It is also important to tell circos() what system of genomic coordinates should be used, i.e. the exact UCSC name of the appropriate organism and genome assembly is needed.

First, let us plot the chromosomes:

```
> circos(xc=400, yc=400, R = 300, W = 15,
cir = "mm10", type = "chr", print.chr.lab =
TRUE, scale = FALSE);
```

The xc and yc parameters set the center of the circle used for the track in the *x* and *y* axes respectively. They are shown for completeness because 400 is the value by default and therefore could be omitted. This track will be drawn at a distance of 300 from the center of the plotting area ($R = 300$) and its width is set to 15

Table 2
Table liking pairs of QTL regions

seg1	po1	name1	seg2	po2	name2	freq	Colors
1	8010000	PC01	7	34970000	PC16	1	#BABABA
1	8010000	PC01	9	31940000	PC22	1	#BABABA
1	99030000	PC02	2	84120000	PC06	1	#BABABA
1	99030000	PC02	8	73660000	PC19	1	#BABABA
1	99030000	PC02	9	57230000	PC23	1	#BABABA
…							

The first six columns are required by the circos() function. The freq column contains the number of significant SNP pairs and the Colors column is used to plot a color gradient according to freq

(W= 15). We set to use the mm10 mouse genome by the cir parameter. This first track is special because it does not consist on actual data but it's used to plot chromosomes. This is done by setting type= "chr". We can ask to print chromosomes names by setting print.chr.lab=TRUE. Optionally, one can show a scale for the chromosome positions in Mb by setting scale=TRUE.

The following command is used to plot arcs for each QTL in the track:

```
> circos(R = 280, cir = "mm10", mapping =
SpermFert, type = "arc2", col = "#FCB14C",
print.chr.lab = FALSE, W = 10, scale =
FALSE, lwd = 10, B=FALSE);
```

The mapping parameter receives the data table in BED format. Since QTLs are large chromosomal segments, they can be plotted as arcs by setting type="arc2" as above. Arcs can be of two types: of variable ("arc") or fixed ("arc2") radius. The color is provided to the col argument as a quoted string with a color code supported by R. A gray background to a track can be set by setting B=TRUE. This can be useful to differentiate tracks more easily by alternating white and gray backgrounds. Commands similar to the one above must be used for each data track, adjusting the values of R and col so that arcs are placed at different distances from the center and in different colors. See also the 04.Circos plots.R script in the supplementary material for the full series of commands (Insert Table 3).

Finally, we can use circos() to plot lines connecting different genomic regions to represent the genetic interactions for PC1, reported by [5]. Data for this track was taken from the media-5 file in supplemental material of [5]. In the original data table, there was one line for each pair of interacting SNPs and the names of the two interacting regions is indicated. First, we summarize this by QTL region and determine the number of significant pairwise

Table 3
Colors by number of interacting SNPs used for connecting lines in Fig. 3

N° ranges	Ranges	Colors
1	1 < 50	"#BABABA"
2	51 < 100	"#A5C1DB"
3	101 < 250	"#828CC3"
4	251 < 500	"#8463AC"
5	501 < 1000	"#87459E"
6	1001 < 2000	"#81107C"
7	>2000	"#4A0247"

interactions (freq column in Table 2). We then define seven arbitrary ranges of SNP pair counts and assign a color to each range in order to give the impression of increasing intensity as counts increase [5]. Then, we must use the circos() function for to plot each range separately as in the following command line:

```
> circos(R = 210, cir = "mm10", W = 10, mapping = RegionsPair[(RegionsPair$freq >= 1
& RegionsPair$freq < 50),], type = "link",
col = RegionsPair[(RegionsPair$freq >= 1 &
RegionsPair$freq < 50),"Colors"], lwd=1);
```

For each range we use a different width and color by setting of lwd and col parameters. By setting type= "link", we tell circos to plot this track as connecting lines. The logical conditions that are placed within square brackets allow to subset the RegionsPairs track. In the above example, only the region pairs that meet the freq>=1 and freq<50 are provided to circos(). Note that in R, >= means "greater than or equal to". The legends are created with the R's builtin function legend(). See the 04.Circos plots.R script in the electronic supplementary material for the full list of commands used. All necessary data to produce these plots is also provided.

2.6 Summary of Results

In this chapter, we have demonstrated the use of three types of visualization for the inspection of results from Systems Genetics experiments in chromosomal context. Although simple in nature, a Dotplot revealed many properties of the data that were of significance for the following analyses. The appearance of vertical stripes locating trans-eQTL and horizontal stripes suggesting unaccounted population structure demonstrate that this global visualization is a must as one begins to inspect the results from Systems Genetics studies. Karyotype-like plots allowed incorporating new sources of information. Particularly, we were able to see colocalization of fertility and gene expression QTLs with peaks of eQTL frequency. This evidence suggested that loci harboring genetic variation affecting fertility traits may be mediating their effects through alteration in the expression of multiple genes. Whether those groups of genes form networks and whether those networks have a role in mediating QTL effects must be tested by alternative approaches. Finally, the Circos plot allowed inspecting the global pattern of gene-to-gene genetic interactions significantly affecting gene expression in testis. This visualization revealed that a few pairs of chromosomes account for most of the pairwise interactions affecting gene expression. In most cases, the interacting regions host phenotypic QTLs. However, some regions did not have QTLs identified from the GWAS nor from experimental crosses and the investigator may have missed them unless this type of analysis was performed.

3 Further Considerations and Limitations

We have presented three ways of visualizing genetic and genomic data using the R-programing language. This has the advantage of providing great flexibility for customizing the plot and has the potential to handle large amounts of data seemingly. However, this system does require some level of proficiency in a programing language. Although compared to others, this language is especially easy to learn, there is a learning curve for beginners. We hope that the examples and language used in this chapter make it easier for biologists and geneticists to get started and that we have provided the motivation to make the effort worthwhile.

Another limitation of the approach taken here is that the plots generated are not interactive, limiting the ability to explore the results at different levels of resolution or to direct users to further sources of information. There are some tools freely available that can provide this functionality. Of special mention, the Integrative Genomics Viewer can display genomic data in whole chromosomes or small genomic regions and allows simultaneous and diverse tracks of information in a similar manner than the tools demonstrated in this chapter [11]. By pointing, clicking, and dragging, the user can explore data at different resolutions and easily add public or private tracks of information. However, whole genome views as done here are no possible with this tool. The concepts provided in this chapter should prove useful when using any type of visualization tool, including IGV.

Acknowledgments

We acknowledge Dr. Bettina Harr at the Max-Planck-Institut fuer Evolutionsbiologie for making data available for this publication and for useful communications with the authors that helped improving the manuscript of this chapter. Development of the chromPlot package was funded by Grant FONDECYT 11121666.

References

1. Jansen RC, Nap JP (2001) Genetical genomics: the added value from segregation. Trends Genet 17(7):388–391
2. Civelek M, Lusis AJ (2014) Systems genetics approaches to understand complex traits. Nat Rev Genet 15(1):34–48. doi:10.1038/nrg3575
3. Jansen RC, Tesson BM, Fu J, Yang Y, McIntyre LM (2009) Defining gene and QTL networks. Curr Opin Plant Biol 12(2):241–246. doi:10.1016/j.pbi.2009.01.003
4. Schadt EE, Lamb J, Yang X, Zhu J, Edwards S, Guhathakurta D, Sieberts SK, Monks S, Reitman M, Zhang C, Lum PY, Leonardson A, Thieringer R, Metzger JM, Yang L, Castle J, Zhu H, Kash SF, Drake TA, Sachs A, Lusis AJ (2005) An integrative genomics approach to infer causal associations between gene expression and disease. Nat Genet 37(7):710–717. doi:10.1038/ng1589
5. Turner LM, Harr B (2014) Genome-wide mapping in a house mouse hybrid zone reveals

hybrid sterility loci and Dobzhansky-Muller interactions. eLife Sci 3:e02504. doi:10.7554/eLife.02504

6. Ihaka R, Gentleman R (1996) R: a language for data analysis and graphics. J Comput Graph Stat 5:299–314
7. Gatti DM, Zhao N, Chesler EJ, Bradford BU, Shabalin AA, Yordanova R, Lu L, Rusyn I (2010) Sex-specific gene expression in the BXD mouse liver. Physiol Genomics 42:456–468. doi:10.1152/physiolgenomics.00110.2009
8. Verdugo RA, Farber CR, Warden CH, Medrano JF (2010) Serious limitations of the QTL/microarray approach for QTL gene discovery. BMC Biol 8:96. doi:10.1186/1741-7007-8-96
9. Dzur-Gejdosova M, Simecek P, Gregorova S, Bhattacharyya T, Forejt J (2012) Dissecting the genetic architecture of F1 hybrid sterility in house mice. Evolution 66(11):3321–3335. doi:10.1111/j.1558-5646.2012.01684.x
10. White MA, Steffy B, Wiltshire T, Payseur BA (2011) Genetic dissection of a key reproductive barrier between nascent species of house mice. Genetics 189(1):289–U988. doi:10.1534/genetics.111.129171
11. Thorvaldsdottir H, Robinson JT, Mesirov JP (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform 14(2):178–192. doi:10.1093/bib/bbs017