Constructing simple biological networks for understanding complex high-throughput data in plants.

Running head: Constructing biological networks.

Tomás C. Moyano, Elena A. Vidal, Orlando Contreras-López and Rodrigo A. Gutiérrez*

FONDAP Center for Genome Regulation, Millennium Nucleus for Plant Functional Genomics, Departamento de Genética Molecular y Microbiología, Pontificia Universidad Catolica de Chile.

* Corresponding author: Rodrigo A. Gutiérrez, rgutierrez@bio.puc.cl

i. Summary:

Technological advances in the last decade have enabled biologists to produce increasing amounts of information for the transcriptome, proteome, interactome and other -omics data sets in many model organisms. A major challenge is integration and biological interpretation of these massive data sets in order to generate testable hypotheses about gene regulatory networks or molecular mechanisms that govern system behaviors. Constructing gene networks requires bioinformatics skills to adequately manage, integrate, analyze and productively use the data to generate biological insights. In this chapter, we provide detailed methods for users without prior knowledge of bioinformatics to construct gene networks and derive hypotheses that can be experimentally verified. Step-by-step instructions for acquiring, integrating, analyzing and visualizing genome-wide data are provided for two widely used open source platforms, R and Cytoscape platforms. The examples provided are based on Arabidopsis data, but the protocols presented should be readily applicable to any organism for which similar data can be obtained.

ii. Key Words

Gene networks, bioinformatics, interactions, networks generation, gene expression, correlation, Cytoscape.

1. Introduction

Systems-level analysis in biology aims to understand system structure and dynamic behaviors that emerge from molecular components and their functional relationships ***(1–3)***. A systems biology approach to study the physiology of plants or other living organism entails modeling the system as a whole rather than a selected set of parts. The accuracy of this approach, however, relies heavily on existing knowledge about the components and interactions of the system constituents, as well as on reliable methods to handle, integrate, analyze and visualize large data sets.

During the last decade, advances in experimental methods that generate large data sets accelerated the development of comprehensive resources in many model species. Development of next-generation sequencing (NGS) has been particularly important for data generation due to its broad applications, including genome sequencing, RNA sequencing (RNA-seq), chromatin immunoprecipitation coupled to sequencing (ChIP-seq), and analysis of epigenetic marks ***(4)***. Other important sources of biological data that provide important information about functional relationships are large-scale protein-protein interaction data sets determined by yeast two-hybrid, mass spectrometry, immunoprecipitation or fluorescence resonance energy transfer assays ***(5, 6)***. In addition, protein-DNA associations provide a starting point to construct regulatory networks. These associations are often predicted based on *cis*-regulatory elements and known transcription factor binding specificities ***(7, 8)*** and also on experimentally validated interactions based on one-hybrid or ChIP-seq assays ***(9)***. **Table 1** presents a list of selected databases that contain gene expression and interaction information for Arabidopsis and, in some cases, for other organisms.

A major challenge is the intelligible integration and interpretation of these massive data sets in order to generate testable hypotheses about regulatory networks that govern system behaviors (*e.g.*, molecular mechanisms underlying responses to environmental cues). Network theory applied to biological data has

proven extremely useful to integrate heterogeneous data types and for uncovering organizing principles in biological systems (reviewed in ***(10)***). A gene regulatory network (GRN) captures dependencies among molecular entities that are part of a system. GRNs are usually represented as network graphs were nodes represent molecular entities (*e.g.*, genes, proteins, metabolites) and edges represent functional relationships between them (*e.g.*, protein-protein interactions, protein-DNA interactions, microRNA:target interactions, coexpression). Integrating different types of large-scale data improves regulatory network reconstruction and allows for better understanding of the system structure and regulation ***(11, 12)***. GRN modeling has proven effective for understanding the structure of important biological processes in plants. The first qualitative network model of Arabidopsis was constructed by integrating diverse data types including metabolic and regulatory interactions for 6,176 genes and 1,459 metabolites ***(13)***. This network included 230,900 edges representing different functional relationships (*e.g.*, regulatory, metabolic, physical interaction) and was initially used to determine gene network modules controlled by carbon (C) and/or nitrogen (N) metabolites ***(13)***. In this study, network analysis prompted the hypothesis that auxin signaling was implicated in Arabidopsis root responses to C and/or N metabolites ***(13)***. This hypothesis was later confirmed experimentally ***(14–18)***. Albeit qualitative and incomplete, this network model proved extremely useful to generate concrete testable hypothesis in this and a series of follow-up studies ***(13–16, 19–23)***. For example, network analysis suggested a feedback regulatory loop between the circadian clock and N nutrition in Arabidopsis ***(24)***. Systems analysis showed that *CIRCADIAN CLOCK ASSOCIATED 1* (CCA1), one of the master regulators of the circadian clock, coordinates the organic N response of N-assimilatory genes by direct binding to the promoters of *BASIC REGION/LEUCINE ZIPPER TRANSCRIPTION FACTOR 1* (which in turn regulates *ASPARAGINE SYNTHETASE 1* expression), *GLUTAMINE SYNTHETASE 1.3*, and *GLUTAMATE DEHYDROGENASE 1* ***(24)***. In turn, N-metabolites can act as an input on the clock through modulation of *CCA1* gene expression ***(24)***.

GRN generation is highly dependent on computational analysis in order to adequately manage and employ data that are heterogeneous in nature, and that are presented in different formats. A number of online tools and resources have been developed to help biologists integrate and use available genome-wide data in plants as well as other organisms (*e.g.*, VirtualPlant ***(25)***, CORNET ***(26)***, STRING ***(27)***, GeneMania ***(28)***, ATTED-II ***(29)***). These online tools allow users with no bioinformatics background to generate GRNs to infer biological hypotheses. Albeit extremely useful, these GRNs are limited to data available in the corresponding databases and in most cases are not readily customizable. Moreover, in the case of resources for the plant community, with a few exceptions (e.g. VirtualPlant ***(25)*** and STRING ***(27)***) these tools are only available for Arabidopsis.

The goal of this chapter is to provide instructions on how to download, integrate, analyze and visualize genome-wide data in order to construct gene networks to users with limited bioinformatics skills. We present a simple pipeline that is straightforward to implement as long as the reader is familiar with the R environment ***(30)*** at a basic level. The examples provided use Arabidopsis data, but the protocol should be applicable to any organism for which similar data can be accessed.

2. Materials

a. R, a free software environment for statistical computing and graphics ***(30)***. R can be downloaded from: http://www.r-project.org/. In this chapter we use version 3.1.1.

b. Personal computer or server running R with access to the internet. Computer requirements vary depending on the data to be analyzed, but a minimum of 4 Gb of RAM and 10 Gb of free space are recommended to start.

c. Cytoscape ***(31)***, an open source software platform for visualizing complex networks. Cytoscape can be downloaded from http://www.cytoscape.org/ and requires JAVA™ JRE or JDK. In this chapter we use Cytoscape version 3.1.1 with the BiNGO 3.0.2 and clusterMaker2 0.9.3 plugins.

d. Gene expression data set obtained from microarray files or RNA-seq data. These data can be downloaded from public databases (*e.g.,* **Table 1**) or obtained in house. Microarray files used in this chapter were downloaded from the Gene Expression Omnibus (GEO) database and ArrayExpress using the URLs and experiment identifiers in **Table 2**.

3. Methods

In the following sections, we describe a pipeline for integrating transcriptomics and interaction data to generate gene networks (**Fig. 1**). This integrative network approach has been shown to be effective in identifying important genes in biological processes of interest in plants and other organisms ***(13–16, 20–23)***. As a case study, we will use gene expression data from microarray experiments of Arabidopsis roots treated with nitrate (**Table 2**) and interaction data obtained from public databases (see below) to identify potential key regulatory factors controlling nitrate responses in Arabidopsis roots.

3.1 Gene expression data acquisition from public databases

We will initiate our work to construct a gene network by generating a normalized gene expression data matrix from public data repositories (**Fig. 1**). For the purposes of this example, Affymetrix *.CEL files for the list of experiments used were downloaded from GEO and Array Express databases using the URL and the experiments indicated in **Table 2**. Please note that for this example we have selected data files from wild-type Arabidopsis plants and from root tissue, as described in Canales et al., 2014 ***(18)***. For your convenience, you can download a compressed file containing all the experiments from http://virtualplant.bio.puc.cl/share/pfg/data.tgz. The CEL files should be extracted and moved to an empty folder, which in this example will be named "example". If you are not sure how to extract files from the archive, please read the information in the README file provided within the same folder with the expression data (http://virtualplant.bio.puc.cl/share/pfg/README). An R script file with all the commands described below can be also downloaded from the same website (http://virtualplant.bio.puc.cl/share/pfg/R-script.R).

3.2. Constructing a normalized gene expression matrix for an arbitrary list of genes.

Raw data were normalized in R using Robust Multiarray Analysis (RMA) ***(32)*** from the *affy* library obtained from Bioconductor (www.bioconductor.org). If you want to work in R using a graphical environment, download and install RStudio (http://www.rstudio.com/). In the following instructions, lines that contain commands to execute in R will be indicated with a consecutive number and a "greater than" sign ">". Comments pertinent to each command line will be indicated with a hash symbol "#" immediately above the command line. To begin with our pipeline, run R and change the working directory to the "example" folder as indicated in command line *1* below.

```
# To select the working directory, replace the text inside the quotes
with the correct location for the folder with the example data sets. (see
Note 1):

1 > setwd("/Users/example")
```

```
# First, select the Bioconductor repository and download and install the affy package. Documentation on how to install packages can be obtained from the Bioconductor website (http://www.bioconductor.org). You need to do this only once and can skip to line 4 if you have the affy package already installed:

2 > source("http://bioconductor.org/biocLite.R")

3 > biocLite("affy")
```

Affymetrix *.CEL files are read and normalized using the following commands:

```
# To load the package, read the *.CEL files and normalize the data using the RMA method. Documentation for this package can be obtained from the Bioconductor website

(http://www.bioconductor.org/packages/release/bioc/html/affy.html)

4 > library(affy)

# To read all CEL files in the working directory:

5 > Data<-ReadAffy()

# To normalize the data (for details see http://www.bioconductor.org/packages/release/bioc/html/affy.html):

6 > eset<-rma(Data)

7 > norm.data<-exprs(eset)
```

The `norm.data` object contains the normalized expression for every probeset in the ATH1 microarrays used in this example. In order to convert the probeset IDs to Arabidopsis gene identifiers, the file ftp://ftp.arabidopsis.org/home/tair/Microarrays/Affymetrix/affy_ATH1_array_elements-2010-12-20.txt must be downloaded from the TAIR database and placed in the folder with the microarray data. In order to avoid ambiguous probeset associations (*i.e.*, probesets that have multiple matches to genes), we will only use probesets that match only one gene in the Arabidopsis genome.

```
8 > affy_names<-read.delim("affy_ATH1_array_elements-2010-12-20.txt",header=T)
```

```
# Select the columns that contain the probeset ID and corresponding AGI number (in this example, columns 1 and 5). Please, note that the positions used to index the matrix depend on the input format of the array elements' file. You can change these numbers to index the corresponding columns if you are using a different format:

9 > probe_agi<-as.matrix(affy_names[,c(1,5)])

# To associate the probeset with the corresponding AGI locus:

10 > normalized.names<-merge(probe_agi,norm.data,by.x=1,by.y=0)[,-1]

# To remove probesets that do not match the Arabidopsis genome:

11 > normalized.arabidopsis <-
     normalized.names[grep("AT",normalized.names[,1]),]

# To remove ambiguous probesets:

12 > normalized.arabidopsis.unambiguous<-
    normalized.arabidopsis[grep(pattern=";",normalized.arabidopsis[,1],
    invert=T),]

# In some cases, multiple probesets match the same gene, due to updates in the annotation of the genome. To remove duplicated genes in the matrix:

13 > normalized.agi.final<-
     normalized.arabidopsis.unambiguous[!duplicated(normalized.arabidopsi
     s.unambiguous[,1]),]

# To assign the AGI number as row name:

14 > rownames(normalized.agi.final)<-normalized.agi.final[,1]

15 > normalized.agi.final<-normalized.agi.final[,-1]
```

The resulting gene expression data set, `normalized.agi.final`, contains unique row identifiers (*i.e.*, AGI loci) and expression values obtained from different experiments on each column, for example:

```
GSM1054974_Col_0_KCl_2H_R1.CEL.gz GSM1054975_Col_0_KCl_2H_R2.CEL.gz
GSM1054976_Col_0_KCl_2H_R3.CEL.gz
```

```
ATMG00640        4.188101          4.109671          4.130230

ATMG00650        4.417411          4.542037          4.536882

ATMG00660        5.658079          5.717082          5.296106

ATMG00670        4.759369          4.849271          4.965505

ATMG00680        4.434071          4.395689          4.468155


# To export this data matrix from R to a tab-delimited file, use the
following command. The file will be written to the folder that you set up
as your working directory in R using the setwd() command in line 1 above:


16 > write.table (normalized.agi.final,"normalized.agi.final.txt",
sep="\t",col.names=NA,quote=F)
```

Users can continue analyzing the entire data set or a subset of it based on a list of genes of interest **(Fig. 1).** The latter is often recommended because it reduces computational requirements and calculation time and facilitates interpretation of results. In this example, we use a list of genes defined in a previous publication from our group (see Table S3 in Supplementary Material of Canales et al., 2014 ***(18)***). This file can be downloaded from http://virtualplant.bio.puc.cl/share/pfg/id.genes.txt. The file contains a list of genes that are regulated in response to nitrate treatments both in a nitrate reductase-null mutant and in wild-type plants ***(33)***. Since nitrate-reductase null mutants are unable to reduce nitrate, genes that respond similarly in wild-type and mutant plants are thought to respond to the nitrate signal and not to a signal produced after nitrate reduction or ammonia assimilation, and thus are direct nitrate responders.

To obtain expression values for the genes of interest from the expression data matrix prepared in the previous section, we will use R to intersect the Gene Expression Matrix (`normalized.agi.final.txt`) obtained in section 3.2 command line 16 with the list of genes of interest (`id.genes.txt`). This will create the object `data`, containing the expression values for the genes of interest.

```
# To read a gene list and gene expression matrix files. The id.genes.txt
file is a text file with one locus identifier per line. Any list of
interest can be generated in a text file.

17 > id.genes<-sort(unique(as.matrix(read.table("id.genes.txt"))))


# To read back into R the gene expression data matrix table we created in
command line 15, use the following instructions:

18 > normalized.agi.final <-read.table("normalized.agi.final.txt",
      header=T, row.names=1)

# Selects rows using the identifiers in the gene list. data.interest will
contain gene expression values for the genes of interest we uploaded from
the id.genes.txt file. The function na.exclude removes rows corresponding
to genes that were not found in the data set.

19 > data.interest<-na.exclude(normalized.agi.final [id.genes,])
```

3.3. Calculating correlation of gene expression

In this section, we will generate a matrix containing a list of all possible pairs of genes and their Pearson correlation coefficient, p-value and adjusted p-values using false discovery rate ***(34)*** **(Fig. 1)**. Correlation networks are informative to associate genes that are involved in the same biological pathway or that are part of protein complexes. The list of pairs generated will be later used for querying interaction data (see below).

The following function, *cor.pairwise,* uses two arguments. *data* (obtained in section 3.2 command line 19) defines the gene expression matrix file, and *outputfile* defines the name of the output file containing all the correlations between pairs of genes.

```
20.1 > cor.pairwise<-function(data,outputfile)
20.2  {  i=1
20.3     out<-NULL
20.4     while(i<dim(data)[1]) {
20.5     k=i+1
20.6     while(k<=dim(data)[1])
20.7    {
```

```
20.8      corre1<-cor.test(t(data[i,]),t(data[k,]), method="pearson")
20.9       corre<-cbind(corre1$estimate,corre1$p.value)
20.10      bl<- t(rownames(data)[c(i,k)])
20.11      corre<-cbind(bl,corre)
20.12      out<-rbind(out,corre)
20.13      k=k+1
20.14   }
20.15    print (rownames(data)[i])
20.16    i=i+1
20.17   }
20.18  pvals<-out[,4]
20.19  pval.adjust=p.adjust(pvals,method="fdr")
20.20  out<-cbind(out,pval.adjust)
20.21  colnames(out)<-c("id1","id2","cor","pval","adj.pval")
20.22  rownames(out)<-make.names(rownames(out),unique=T)
20.23  return(out)
20.24 }
# We use the function as follows. Please be patient because this function
      will take several minutes to run:
21 > output.cor<-cor.pairwise(data=(data.interest))
```

The object *output.cor* will include gene pair names, correlation, p-values and adjusted p-values.

```
22 > head(output.cor)
```

	id1	id2	cor	pval	adj.pval
cor	AT1G01190	AT1G02310	0.010	9.1E-01	9.2E-01
cor.1	AT1G01190	AT1G02340	-0.314	1.0E-04	2.4E-04
cor.2	AT1G01190	AT1G03080	-0.059	4.8E-01	5.4E-01
cor.3	AT1G01190	AT1G04770	0.461	4.2E-09	1.8E-08
cor.4	AT1G01190	AT1G05300	-0.451	1.0E-08	4.0E-08
cor.5	AT1G01190	AT1G05340	0.267	1.1E-03	2.1E-03

You can filter these correlations by defining a threshold for the adjusted p-values or correlation coefficients. In this example, we set a p-value ≤ 0.01 and correlation ≥ 0.75.

```
23 > output.cor_pval<-output.cor[output.cor[,5]<=0.01,]

24 > output.cor_pval_075<-
output.cor_pval[abs(as.numeric(output.cor_pval[,3]))>=0.75,]

25 > output.filtered<-output.cor_pval_075
```

3.4. Adding publicly available interaction information to the coexpression network

At this point, the `output.filtered` object contains the information necessary for building a simple correlation network **(Fig. 1)**. Although correlation networks are useful to associate genes that may be related at the functional level, we can add evidence to these putative functional associations by including additional interaction data. In order to enrich the network with other interaction data, the first step is to obtain corresponding data files from public sources. In plants, most interaction data available are for Arabidopsis, but there is increasing support for other plant species (*see* **Table 1** for examples of protein-protein, regulatory and microRNA:target interactions). To incorporate this information in the network, files should be tab-delimited. In this example, we used interaction data parsed and formatted from the Supplemental information of Srivastava et al., 2010, Barah et al., 2013 and Geisler-Lee et al., 2007 ***(35–37)*** and publicly available interaction databases: The Plant Interactome Database, ATPID, AtPIN, and PAIR databases (**Table 1**). Interaction data used in this example can be downloaded from http://virtualplant.bio.puc.cl/share/pfg/interaction.data.txt. Note that gene identifiers in all data sources must be in the same format. In this example, all gene identifiers are in upper case as shown below (see Note 2):

```
ida        idb        interaction_type source
AT1G05410  AT3G10140  protein-protein  AI_interactions
AT3G54850  AT5G19010  protein-protein  AI_interactions
AT3G07780  AT5G66720  protein-protein  AI_interactions
AT1G80040  AT5G66720  protein-protein  AI_interactions
AT1G09660  AT2G38610  protein-protein  AI_interactions
```

Using this simple format, we can readily integrate interaction information coming from different sources. The following example combines the data for different data sets (in this case, the correlation matrix obtained above and the interaction data table), using the function `merge` that intersects two matrices keeping rows with same pairs IDs and adding new columns with interaction information.

In order to intersect the interaction data table containing the protein-protein interaction information with the correlation data contained in the `output.filtered` object, we will use the `merge` function as follows:

```
# First, we load the interaction data table. To facilitate this example, we have prepared a text file with all the data sources we will be using from Table 1. You can download this file from http://virtualplant.bio.puc.cl/share/pfg/interaction.data.txt

27 > interaction.data<-
     as.matrix(read.table("interaction.data.txt",sep="\t",header=T,fill=T
     ))

# Then we combine the table with the recently created coexpression data matrix:

28 > output.filtered.interaction<-

      unique(rbind(merge(interaction.data,output.filtered,by.x=c(1,2),by.
      y=c(1,2)),merge(interaction.data,output.filtered,by.x=c(1,2),by.y=c
      (2,1))))
```

The same procedure (command lines 27,28) can be repeated with different interaction files, or a merged file of different interaction data parsed and formatted as above.

3.5 Determining putative TF-target pairs using expression data and the AGRIS database

In the specific case of the regulatory interaction database AGRIS (**Table 1**), there is only information about TF binding sites in the promoter of Arabidopsis genes (`BindingSite.tbl`). These data can be converted into a matrix containing each gene and all the described members of the TF gene families predicted to bind their promoters (see command line 38 below). This matrix can then be used as above to be intersected with the correlation matrix. Data from TF families and their members can be obtained from AtTFDB (`families_data.tbl`) in AGRIS.

```
29 > bstable<-as.matrix(read.table("BindingSite.tbl",sep="\t"))

30 > fam.tf<-as.matrix(read.table("families_data.tbl",sep="\t"))


# Once files are loaded, we can create a new object, agris.bs, selecting
only the columns of interest for our analysis. In this case we chose from
bstable the columns describing the binding site, promoter and TF family.

31 > agris.bs<-cbind(toupper(substr(bstable[,7] ,start=1,stop=9)),
bstable[,c(2,10,11)])

# For promoters that are bound more than once by the same family of TFs,
we use the command unique to merge the repeated binding sites present,
leaving one for each type.

32 > agris.prom.fam<-unique((agris.bs[,c(1,3)]))

# To create a table with only the minimal information from the
transcription factor and the family it belongs to:

33 > fam.tf.gen<-cbind(fam.tf[,1],toupper(fam.tf[,2]))

# To merge the binding site present in the promoters with the family of
transcription factors able to bind to the promoter sequence:

34 > pairs.fam.bs<-
as.matrix(merge(fam.tf.gen,agris.prom.fam,by.x=1,by.y=2))

# Then we parse the data for further use:
```

```
35 > agris.pairs<-
     as.matrix(cbind(pairs.fam.bs[,c(2,3)],"TF_TARGET","AGRIS",
     pairs.fam.bs[,1]))

36 > colnames(agris.pairs)<-
     c("TF","TARGET","interaction_type","source","family")

37 > agris.pairs<-unique(agris.pairs)

# To verify that the final Transcription Factor – Target pairs object has
the correct structure:

38 > head(agris.pairs)

          TF        TARGET    interaction_type source    family
          AT3G24650 AT4G21390 TF_TARGET        AGRIS     ABI3VP1
          AT3G24650 AT4G09070 TF_TARGET        AGRIS     ABI3VP1
          AT3G24650 AT1G53130 TF_TARGET        AGRIS     ABI3VP1
          AT3G24650 AT1G32200 TF_TARGET        AGRIS     ABI3VP1
          AT3G24650 AT2G27040 TF_TARGET        AGRIS     ABI3VP1
          AT3G24650 AT1G10960 TF_TARGET        AGRIS     ABI3VP1
```

Once the TF-TARGET table (`agris.pairs`) is created, we intersect this table with the correlation data (`output.filtered`).

```
40 > output.filtered.agris<-
      unique(rbind(merge(agris.pairs,output.filtered,by.x=c(1,2),by.y=c(1
     ,2)),
      merge(agris.pairs,output.filtered,by.x=c(1,2),by.y=c(2,1))))

# Create the final table adding header and all the information available:

41 > write.table(
     t(c("id1","id2","type","source","cor","p.val","p.val.adj","info1")),
     "out.info.txt",sep="\t",row.names=F,quote=F,col.names=F)

42 > write.table(output.filtered.agris[,c(1:4,6:8,5)],
     "out.info.txt",sep="\t",append=T, row.names=F, col.names=F,quote=F)


43 > write.table(output.filtered.interaction,"out.info.txt",
     sep="\t",append=T, row.names=F, col.names=F,quote=F)

# Parse and add to the table correlation data that passed the filters.
```

```
44 > cor.pairs<-cbind(output.filtered[,1:2],"correlation_pair",
     "own_analysis",output.filtered[,3:5])

45 > write.table(cor.pairs ,"out.info.txt",append=T, sep="\t", quote=F,
     row.names=F,col.names=F)
```

The final data file containing the correlations and interaction information from the analysis is stored in the file `out.info.txt`. This data file contains information for each gene pair, including correlation, adjusted p-values, interaction type and source, as shown below:

id1	id2	type	source	cor	p.val	p.val.adj	info1
AT4G22070	AT4G23700	TF_TARGET	AGRIS	0.771	0	0	WRKY
AT4G22070	AT5G64120	TF_TARGET	AGRIS	0.751	0	0	WRKY
AT5G10030	AT5G10210	TF_TARGET	AGRIS	0.755	0	0	bZIP
AT5G10030	AT5G10820	TF_TARGET	AGRIS	0.823	0	0	bZIP
AT5G10030	AT5G13110	TF_TARGET	AGRIS	0.824	0	0	bZIP
AT5G10030	AT5G13420	TF_TARGET	AGRIS	0.799	0	0	bZIP

3.6. Network visualization and analysis.

In order to analyze and visualize the network obtained in the previous section, we will use Cytoscape, an open source software platform ***(38, 39)***. After launching Cytoscape, the network data contained in *out.info.txt* can be directly imported into the program **(Fig. 1)**. This can be done by selecting the file in File > Import > Network > File. ‘Source Interaction’ is set to ‘Column 1’ and ‘Target Interaction’ to ‘Column 2’ to indicate the columns in *out.info.txt* that contain the gene ID information of the interacting pair (**Fig. 2**). In the example below, we set column 3 as ‘Interaction type’, since this column contains the interaction type in *out.info.txt*. To keep the information contained in the table shown in the ‘Preview’ window as an edge attribute (*e.g.*, correlation value, p-values or adjusted p-values), the corresponding column header should be clicked to activate it.

3.6.1. Including node attributes

Node attributes are useful for network visualization and analysis. Node attributes can include gene names, functional annotation (*e.g.*, DNA binding, transporter, catalytic activity) or gene family, for example. The format of a node attribute file consists of a column containing the identifier and at least one additional column with the attribute. In the example below, column 1 shows the gene identifier, column 2 the gene family and column 3 indicates that the gene is a transcription factor. The data in this example can be downloaded from http://virtualplant.bio.puc.cl/share/pfg/ (transcription_factor_family.agris.txt).

TF	TF_family	gen_type
AT1G01010	NAC	TF
AT1G01030	ABI3VP1	TF
AT1G01060	MYB-related	TF
AT1G01250	AP2-EREBP	TF
AT1G01260	bHLH	TF
AT1G01350	C2H2	TF

Node attributes can be imported by clicking File>Import>Table>File and selecting the file containing the node attributes (**Fig. 3**). In the Fig. 3, the first column shows the node that will receive the attribute, in this case, TF_family and gen_type.

3.6.2. Visualizing the network

For visualization, users can customize network layout and style by selecting node and edge positioning, shape, color, size, font among other features. To change network style, use the Control Panel, select the 'Style' tab and click on the desired characteristic to change (*e.g.*, label). In the emergent list, select 'Column' and select the attribute to be shown (*e.g.*, name). Under 'Mapping type' select 'Passthrough Mapping'. 'Passthrough mapping' passes the network attribute directly to visual attributes and is typically used to specify node or edge labels. For changing node shape, for example depending on whether the gene codes for a transcription factor, microRNA or enzyme, select 'Shape' and select 'Column' –

‘gen_type’. Then under ‘Mapping type’ select ‘Discrete Mapping’. ‘Discrete Mapping’ can map different types of molecules to different node shapes, for example, a triangle for transcription factors (**Fig. 4**). Also, the user can customize edge properties: for example, correlation values can be used to define line width, color or arrow shape, among others.

3.6.3. Network topology analysis

Networks generated following a protocol such as the one outlined in this chapter will generally have a large number of genes. A critical step in the analysis is to identify the small number of genes that may be biologically relevant for a given process of interest. Network topology analysis is a powerful way to prioritize nodes that can be important for gene network function. Cytoscape includes built-in tools that can give us basic network statistics, such as node degree, betweenness centrality, cluster coefficient, among others **(Fig. 1)**. To open the network analysis tool, click Tools > NetworkAnalyzer > Network Analysis > Analyze Network. To simplify this example, since our network connections are mainly based on correlation values, we choose the ‘Undirected’ option. The ‘Results Panel’ will summarize the network analysis results. Once calculated, network statistics can be used as attributes and added as visual cues using the ‘Control Panel’ as we described above. For example, we can visualize node degree by making node size proportional to this statistic. ‘Node degree’ indicates how many edges are attached to a node in the network. The most connected nodes or hubs are key for network structure and often regarded as key for biological network function. Since node degree is a numeric value, you can use the ‘Continuous Mapping’ Mapping Type to visualize this attribute (**Fig. 5**). The resulting network is shown in **Fig. 6.** Triangles represent genes that code for transcription factors, squares represent other genes, and node size represents the degree or number of connections to other nodes.

It is also possible to adjust node and edge positioning by changing the network layout. The organic layout algorithm is usually helpful when visualizing biological networks. In this layout, nodes are considered to be physical objects with mutually repulsive forces, and the connections between nodes are considered to

be springs attached to the pair of nodes. These springs produce repulsive or attractive forces between their end points. Resulting layouts often expose the inherent symmetric and clustered structure of a graph, they show a well-balanced distribution of nodes, and have few edge crossings. To apply this algorithm to the network, select Layout > yFiles > Organic. This layout is useful to distinguish highly connected regions of the graph from sparse ones. In addition, you can often identify highly connected nodes or hubs by visual inspection of the graph. The biggest triangle in Fig. 6 corresponds to the most connected TF in the network. We found that the most connected transcription factor is *TGA1* (AT5G65210), a gene that has recently been shown to be a key regulatory factor of the root nitrate response, controlling primary and lateral root growth ***(20)***. In order to analyze the subnetwork of genes connected with *TGA1*, we can select *TGA1* neighbors by clicking the *TGA1* node (in this case, the biggest green triangle) and selecting Select > Nodes > First Neighbors of Selected Nodes > Undirected. The selection contains all the genes connected with *TGA1* (AT5G65210). Interestingly, one of the *TGA1* neighbors is *NRT2.1*, a high-affinity nitrate transporter whose promoter has been shown to be bound by TGA1 and that acts downstream of this transcription factor to control root system architecture in response to nitrate ***(20)***. This example shows how a simple network analysis can be a powerful tool to identify key regulatory factors and their putative target effectors.

3.6.4. Performing cluster analysis of the network

To acquire a comprehensive visualization of node connectivity, we can perform cluster analysis of the network. Since more interconnected genes usually work together, these clusters can represent functional cellular modules **(Fig. 1)**. There are numerous network clustering algorithms to find highly connected regions. In Cytoscape Application Store web (http:// apps.cytoscape.org/), the user can find many plugins for downloading, installing and performing network analyses. For the purpose of this example, we will use clusterMaker2 plugin ***(40)***. clusterMaker2 offers different options to perform cluster analysis (refer to the clusterMaker

manual for details); in this example, we will use default options. We will use the Community Clustering (GLay) because it provides layout algorithms and structured and informative visualization optimized for efficient exploration and analysis of large networks ***(41)***. This analysis results in three mayor subnetworks containing most of the nodes and other smaller clusters (**Fig. 7**).

3.6.5. Functional analysis of network modules.

As we stated above, clusters of genes usually represent functional biological modules. In order to determine which processes are enriched in these clusters, we can perform a Gene Ontology (GO) enrichment analysis **(Fig. 1)**. There are numerous GO Term Enrichment tools that can determine whether the observed level of annotation for a group of genes is significant in the context of a background set. For example, the BiNGO plugin ***(42)***, available from the Cytoscape Application Store web, offers us different options to perform Gene Enrichment Analysis. To perform Enrichment Analysis, the user should first select a set of genes, for example, genes contained in subnetworks obtained after cluster analysis. Genes can be selected by either clicking on the desired nodes holding the Shift key or by defining the area that encloses a subnetwork using the mouse by holding the left click. Additionally, a list of genes of interest can be selected by clicking on "Select > Nodes > From ID List file…" and pasting the ID of the genes. After selecting the set of genes of interest, go to Apps > BiNGO. In the BiNGO Settings panel, the user can select the organism (in this case, *Arabidopsis thaliana*), and test parameters such as p-value (in our example $p < 0.01$), statistical test, or correction for multiple testing (see BiNGO documentation for details). BiNGO outputs include a table showing the overrepresented categories, number of genes and gene names or identifiers. BiNGO also produces a network of GO terms that can be visualized hierarchically by gene ontology level using Layout>yFiles Layout>Hierarchic. For our network, two of the three main subnetworks are enriched in genes involved in biological processes that have been previously described to be regulated directly by nitrate, such as response to nitrate, nitrate

transport, and processes related to nitrate reduction and assimilation ***(18)***. The third subnetwork contains an overrepresentation of genes involved in response to cytokinin (CK) stimulus. CK has been described as part of systemic N signaling regulating the expression of N uptake and assimilation as well as root architecture and might function as a root-to-shoot signal related to nitrate supply ***(43–47)***.Thus, our clustering approach is effective for generating useful hypothesis about the functional specialization of components of gene networks.

4. Final Remarks

The simple integrative bioinformatics approach presented here allowed us to identify the modular structure of the nitrate-responsive transcriptome of Arabidopsis roots and to highlight the role of *TGA1* as a key regulatory gene for the root nitrate response. Albeit a single example, the case study presented here illustrates how effective this method is for predicting important new regulatory factors involved in plant responses to a signal (*e.g.*, nitrate) and could be readily applied for similar studies in plant responses to other environmental cues. Please, note that data, methods and parameters used at each step are meant to be for demonstration purposes and by no means should be taken as the only way or as a general rule to carry out data analyses in all cases. Changes in methods and parameters can have a significant impact on your final results and should be carefully evaluated and decided upon depending on your scientific aims as well as experimental design.

5. Notes

1. Please note that in Windows operative systems you need to use the backslash symbol and start with your harddrive (e.g. “C:\”). Usually graphic user interface versions of R also have a “Set Working Directory” option where you can select your folder using Windows explorer.
2. All the gene identifiers used in the analysis must have the same format, since R scripts are case-sensitive. For example, in the case of Arabidopsis

AGI numbers, always use upper case letters. Also, for Arabidopsis identifiers, splicing variants are specified by appending a ".1, .2, .3,…" to the AGI number. These should be eliminated from the identifier.

3. Please note that the instructions provided in this Chapter were based on the software versions indicated in Materials. Although the same analysis can be done in different software versions, changes can occur in the specific instructions. Please refer to the corresponding software manual in case of problems.

Acknowledgments

Research in our group is funded by the International Early Career Scientist program from Howard Hughes Medical Institute, Fondo de Desarrollo de Areas Prioritarias (FONDAP) Center for Genome Regulation (15090007), Millennium Nucleus Center for Plant Functional Genomics (P10- 062-F), Fondo Nacional de Desarrollo Científico y Tecnológico (FONDECYT) 1141097 and 11121225. T.C.M. is funded by CONICYT doctoral fellowship 21110366.

6. References

1. Kitano, H (2002). Systems biology: a brief overview. *Science* **295**, 1662–4.

2. Ideker, T, Galitski, T and Hood, L (2001). A new approach to decoding life: systems biology. *Annu. Rev. Genomics Hum. Genet.* **2**, 343–72.

3. Gutiérrez, RA, Shasha, DE and Coruzzi, GM (2005). Systems biology for the virtual plant. *Plant Physiol.* **138**, 550–554.

4. Lister, R, Gregory, B and Ecker, J (2009). Next is now: new technologies for sequencing of genomes, transcriptomes, and beyond. *Curr. Opin. Plant Biol.* **12**, 19157957.

5. Bracha-Drori, K, Shichrur, K, Katz, A, Oliva, M, Angelovici, R, Yalovsky, S, *et al.* (2004). Detection of protein-protein interactions in plants using bimolecular fluorescence complementation. *Plant J.* **40**, 419–27.

6. Ciruela, F (2008). Fluorescence-based methods in the study of protein-protein interactions in living cells. *Curr. Opin. Biotechnol.* **19**, 338–43.

7. Davuluri, R V, Sun, H, Palaniswamy, SK, Matthews, N, Molina, C, Kurtz, M, *et al.* (2003). AGRIS: Arabidopsis gene regulatory information server, an information resource of Arabidopsis cis-regulatory elements and transcription factors. *BMC Bioinformatics* **4**, 25.

8. Yilmaz, A, Mejia-Guerra, MK, Kurz, K, Liang, X, Welch, L and Grotewold, E (2011). AGRIS: the Arabidopsis Gene Regulatory Information Server, an update. *Nucleic Acids Res.* **39**, D1118–22.

9. Furey, TS (2012). ChIP-seq and beyond: new and improved methodologies to detect and characterize protein-DNA interactions. *Nat. Rev. Genet.* **13**, 840–52.

10. Barabasi, A and Oltvai, Z (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **69**, 572–6.

11. Joyce, AR and Palsson, BØ (2006). The model organism as a system: integrating "omics" data sets. *Nat. Rev. Mol. Cell Biol.* **7**, 198–210.

12. Karlebach, G and Shamir, R (2008). Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.* **9**, 770–80.

13. Gutiérrez, R a, Lejay, L V, Dean, A, Chiaromonte, F, Shasha, DE and Coruzzi, GM (2007). Qualitative network models and genome-wide expression data define carbon/nitrogen-responsive molecular machines in Arabidopsis. *Genome Biol.* **8**, R7.

14. Krouk, G, Lacombe, B, Bielach, A, Perrine-Walker, F, Malinska, K, Mounier, E, *et al.* (2010). Nitrate-regulated auxin transport by NRT1.1 defines a mechanism for nutrient sensing in plants. *Dev. Cell* **18**, 927–37.

15. Vidal, E a, Araus, V, Lu, C, Parry, G, Green, PJ, Coruzzi, GM, *et al.* (2010). Nitrate-responsive miR393/AFB3 regulatory module controls root system architecture in Arabidopsis thaliana. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 4477–4482.

16. Vidal, EA, Moyano, TC, Riveras, E, Contreras-López, O and Gutiérrez, RA (2013). Systems approaches map regulatory networks downstream of the auxin receptor AFB3 in the nitrate response of Arabidopsis thaliana roots. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 12840–5.

17. Vidal, EA, Alvarez, JM and Gutiérrez, RA (2014). Nitrate regulation of AFB3 and NAC4 gene expression in Arabidopsis roots depends on NRT1.1 nitrate transport function. *Plant Signal. Behav.* **9**, e28501.

18. Canales, J, Moyano, TC, Villarroel, E and Gutiérrez, R a (2014). Systems analysis of transcriptome data provides new hypotheses about Arabidopsis root response to nitrate treatments. *Front. Plant Sci.* **5**, 22.

19. Gutiérrez, R a (2012). Systems biology for enhanced plant nitrogen nutrition. *Science* **336**, 1673–5.

20. Alvarez, JM, Riveras, E, Vidal, E a., Gras, DE, Contreras-López, O, Tamayo, KP, *et al.* (2014). Systems approach identifies TGA1 and TGA4 transcription

factors as important regulatory components of the nitrate response of Arabidopsis thaliana roots. *Plant J.*, n/a–n/a.

21. Gutiérrez, R a, Gifford, ML, Poultney, C, Wang, R, Shasha, DE, Coruzzi, GM, *et al.* (2007). Insights into the genomic nitrate response using genetics and the Sungear Software System. *J. Exp. Bot.* **58**, 2359–2367.

22. Nero, D, Krouk, G, Tranchina, D and Coruzzi, GM (2009). A system biology approach highlights a hormonal enhancer effect on regulation of genes in a nitrate responsive “biomodule”. *BMC Syst. Biol.* **3**, 59.

23. Ruffel, S, Krouk, G and Coruzzi, GM (2010). A systems view of responses to nutritional cues in Arabidopsis: toward a paradigm shift for predictive network modeling. *Plant Physiol.* **152**, 445–452.

24. Gutiérrez, R a, Stokes, TL, Thum, K, Xu, X, Obertello, M, Katari, MS, *et al.* (2008). Systems approach identifies an organic nitrogen-responsive gene network that is regulated by the master clock control gene CCA1. *Proc. Natl. Acad. Sci. U. S. A.* **105**, 4939–4944.

25. Katari, MS, Nowicki, SD, Aceituno, FF, Nero, D, Kelfer, J, Thompson, LP, *et al.* (2010). VirtualPlant: a software platform to support systems biology research. *Plant Physiol.* **152**, 500–15.

26. Bodt, S De, Hollunder, J, Nelissen, H, Meulemeester, N and Inze, D (2012). Methods interactions , regulatory interactions , gene associations and functional annotations, 707–720.

27. Franceschini, A, Szklarczyk, D, Frankild, S, Kuhn, M, Simonovic, M, Roth, A, *et al.* (2013). STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.* **41**, D808–15.

28. Zuberi, K, Franz, M, Rodriguez, H, Montojo, J, Lopes, CT, Bader, GD, *et al.* (2013). GeneMANIA prediction server 2013 update. *Nucleic Acids Res.* **41**, W115–22.

29. Obayashi, T, Okamura, Y, Ito, S, Tadaka, S, Aoki, Y, Shirota, M, *et al.* (2014). ATTED-II in 2014: evaluation of gene coexpression in agriculturally important plants. *Plant Cell Physiol.* **55**, e6.

30. R Core Team (2014). R: A Language and Environment for Statistical Computing.

31. Lopes, CT, Franz, M, Kazi, F, Donaldson, SL, Morris, Q and Bader, GD (2010). Cytoscape Web: an interactive web-based network browser. *Bioinformatics* **26**, 2347–8.

32. Irizarry, R a, Hobbs, B, Collin, F, Beazer-Barclay, YD, Antonellis, KJ, Scherf, U, *et al.* (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4**, 249–64.

33. Wang, R, Tischner, R, Gutiérrez, RA, Hoffman, M, Xing, X, Chen, M, *et al.* (2004). Genomic analysis of the nitrate response using a nitrate reductase-null mutant of Arabidopsis. *Plant Physiol.* **136**, 2512–22.

34. Benjamini, Y and Hochberg, Y (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B …* **1**, 289–300.

35. Srivastava, GP, Li, P, Liu, J and Xu, D (2010). Identification of transcription factor's targets using tissue-specific transcriptomic data in Arabidopsis thaliana. *BMC Syst. Biol.* **4 Suppl 2**, S2.

36. Barah, P, Jayavelu, ND, Mundy, J and Bones, AM (2013). Genome scale transcriptional response diversity among ten ecotypes of Arabidopsis thaliana during heat stress. *Front. Plant Sci.* **4**, 532.

37. Geisler-Lee, J, O'Toole, N, Ammar, R, Provart, NJ, Millar, a H and Geisler, M (2007). A predicted interactome for Arabidopsis. *Plant Physiol.* **145**, 317–29.

38. Shannon, P, Markiel, A, Ozier, O, Baliga, NS, Wang, JT, Ramage, D, *et al.* (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–504.

39. Cline, MS, Smoot, M, Cerami, E, Kuchinsky, A, Landys, N, Workman, C, *et al.* (2007). Integration of biological networks and gene expression data using Cytoscape. *Nat. Protoc.* **2**, 2366–82.

40. Morris, JH, Apeltsin, L, Newman, AM, Baumbach, J, Wittkop, T, Su, G, *et al.* (2011). clusterMaker: a multi-algorithm clustering plugin for Cytoscape. *BMC Bioinformatics* **12**, 436.

41. Su, G, Kuchinsky, A, Morris, JH, States, DJ and Meng, F (2010). GLay: community structure analysis of biological networks. *Bioinformatics* **26**, 3135–7.

42. Maere, S, Heymans, K and Kuiper, M (2005). BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics* **21**, 3448–9.

43. Ruffel, S, Krouk, G, Ristova, D, Shasha, D, Birnbaum, KD and Coruzzi, GM (2011). Nitrogen economics of root foraging: Transitive closure of the nitrate-

cytokinin relay and distinct systemic signaling for N supply vs. demand. *Proc. Natl. Acad. Sci. U. S. A.* **108**, 18524–18529.

44. Sakakibara, H, Takei, K and Hirose, N (2006). Interactions between nitrogen and cytokinin in the regulation of metabolism and development. *Trends Plant Sci.* **11**, 440–448.

45. Takei, K, Ueda, N, Aoki, K, Kuromori, T, Hirayama, T, Shinozaki, K, *et al.* (2004). AtIPT3 is a key determinant of nitrate-dependent cytokinin biosynthesis in Arabidopsis. *Plant Cell Physiol.* **45**, 1053–1062.

46. Kiba, T, Kudo, T, Kojima, M and Sakakibara, H (2011). Hormonal control of nitrogen acquisition: roles of auxin, abscisic acid, and cytokinin. *J. Exp. Bot.* **62**, 1399–409.

47. Krouk, G, Ruffel, S, Gutiérrez, R a, Gojon, A, Crawford, NM, Coruzzi, GM, *et al.* (2011). A framework integrating plant growth with hormones and nutrients. *Trends Plant Sci.* **16**, 178–182.

Figure Captions.

Figure 1. **Conceptual flowchart of data analysis used in this chapter.** White boxes represent input data. Dark gray boxes represent subsets of data obtained by filtering procedures (*see* main text). Dark gray and cursive text boxes represent data analysis steps. Black boxes mark analysis outputs. Black arrows correspond to direct steps. Dashed arrows show that multiple steps are needed to generate the output.

Figure 2. Importing a network file into Cytoscape. Screenshot of the "Import Network From Table" window in Cytoscape. Columns in the data file containing nodes (source and target) and interaction type are selected in the Interaction Definition section of the form. In this example, relevant information is contained in Column 1, 2 and 3. Note that all checked columns will also be loaded as edge attributes.

Figure 3. Import nodes attributes from table window. Screenshot of the "Import Columns From Table" window in Cytoscape. Columns in the data file containing node identifiers should match source or target as defined in Figure 2. Attributes are selected for each node identifier in the "New Table" section of the form. In this example, attributes are contained in Columns 2 and 3.

Figure 4. Control panel displaying different network style formatting options. Screenshot of a window for adjusting node fill color and shape based on node attributes. Different node shapes and colors can be assigned in order to improve the visualization. In this example, we select the triangle form and green color for transcription factor (TF) genes.

Figure 5. Control Panel Detail. Screenshot of the window for adjusting node size based on node degree. Different node sizes can be selected in order to improve the visualization and to facilitate the identification of nodes of interest.

Figure 6. Network displaying customized style. Screenshot of the network displayed in organic layout. It is possible to differentiate TF (triangles) from other genes (squares). Also, the most connected nodes (the biggest ones) can be

visualized properly. Nodes are grouped by the connections, making visible the underlying structure of the network.

Figure 7. Community cluster algorithm output. Screenshot of resulting clustering analysis identified three subnetworks grouping most of the nodes. The three upper subnetworks were selected to perform a gene ontology enrichment analysis.

Table Captions.

Table 1. Selected examples of databases with gene expression and interaction data. We show a list of selected databases containing gene expression and interaction information for Arabidopsis and other organisms.

Table 2. Gene expression data sets used in the case study. We show the list of experiments that were used to obtain the gene expression data analyzed in this case study. Please, note that only microarray files from wild-type plants and root tissue were used in this example.

Database name	URL	Type of data available	Plant organisms
ArrayExpress	http://www.ebi.ac.uk/arrayexpress/	Gene expression data	Various organisms
Gene Expression Omnibus (GEO)	http://www.ncbi.nlm.nih.gov/geo/	Gene expression data	Various organisms
Sequence Read Archive (SRA)	http://www.ncbi.nlm.nih.gov/sra	Gene expression data	Various organisms
miRbase	http://www.mirbase.org	microRNA-target	Various organisms
Arabidopsis thaliana protein interaction database (AtPID)	http://www.megabionet.org/atpid/webfile/	Protein-protein	*Arabidopsis thaliana*
Arabidopsis thaliana protein interaction network (AtPIN)	http://bioinfo.esalq.usp.br/atpin/atpin.pl	Protein-protein	*Arabidopsis thaliana*
Predicted Arabidopsis Interactome Resource (PAIR)	http://www.cls.zju.edu.cn/pair/	Protein-protein	*Arabidopsis thaliana*
Membrane-protein Interaction Network Database (MIND)	https://associomics.dpb.carnegiescience.edu/Associomics/MIND.html	Protein-protein	*Arabidopsis thaliana*
Plant protein-protein interaction database (PlaPID)	http://www.plapid.net/	Protein-protein	Various organisms
A predicted Rice Interactome Network (PRIN)	http://bis.zju.edu.cn/prin/	Protein-protein	*Oryza sativa*
Database of Interacting Proteins in *Oryza sativa* (DIPOS)	http://csb.shu.edu.cn/dipos/?id=5	Protein-protein	*Oryza sativa*
Plant Interactome Database	http://interactome.dfci.harvard.edu/A_thaliana/index.php?page=download	Protein-protein	*Arabidopsis thaliana*
Arabidopsis Gene Regulatory Information Server (AGRIS)	http://arabidopsis.med.ohio-state.edu/	TF-promoter	*Arabidopsis thaliana*
AthaMap	http://www.athamap.de/	TF-promoter	*Arabidopsis thaliana*
Transfac	http://www.gene-regulation.com/pub/databases.html#transfac	TF-promoter	Various organisms
Kyoto Encyclopedia of Genes and Genomes (KEGG)	http://www.genome.jp/kegg/	Reactions-Pathways	Various organisms
Plant metabolic pathway database (PMN / PlantCyc)	http://www.plantcyc.org/	Reactions-Pathways	Various organisms

Publication	Experiment tittle	Experiment ID	Download URL
Wang 2003	Treatment of Arabidopsis with low concentration of nitrate.	Exp479	http://data.iplantcollaborative.org/quickshare/f9317af35f1d91be/Exp479.zip
Wang 2004	WT vs NR null mutant high nitrate concentration treatment.	Exp480	http://data.iplantcollaborative.org/quickshare/94b553627c352c9/Exp480.zip
Wang 2007	Arabidopsis treated with nitrite and nitrate.	Exp481	http://data.iplantcollaborative.org/quickshare/d77f7ce802f42ebd/Exp481.zip
Gutierrez 2007	Transcription profiling by array of Arabidopsis grown in nutrient solutions with various concentrations of nitrate and sucrose.	MEXP-828	http://www.ebi.ac.uk/arrayexpress/files/E-MEXP-828/E-MEXP-828.raw.1.zip
Krouk 2010	High resolution NO3 response of Arabidopsis Roots	GSE20044	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE20044&format=file
Ju 2009	Expression data of 10-day-old wild-type and chl1-5 plants exposed to 25 mM nitrate for 0h or 0.5h	GSE9148	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE9148&format=file
Ruffel 2011	A systemic view of coordinated root responses to NO3- heterogeneous environment in Arabidopsis	GSE22966	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE22966&format=file
Patterson 2010	Comparison of root transcriptomes in Arabidopsis thaliana plants supplied with different forms of inorganic nitrogen	GSE29589	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE29589&format=file
Vidal 2013	Root nitrate response of Ws plants and afb3-1 mutant plants.	GSE35544	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE35544&format=file
Alvarez 2014	Root nitrate response of Col-0 plants and tga1/tga4 mutant plants	GSE43011	http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE43011&format=file