Title

RNA sequencing analyses for deciphering potato molecular responses

Authors:

Živa Ramšak[1], Marko Petek[1], Špela Baebler[1]

Affiliations:

[1] Department of Biotechnology and Systems Biology, National Institute of Biology, Večna pot 111,

1000 Ljubljana, Slovenia

*** Corresponding author:** Živa Ramšak, e-mail: ziva.ramsak@nib.si

Running head

Potato transcriptomics

Abstract

Understanding the molecular mechanisms of potato development and responses to environmental stressors is of utmost importance for achieving stable crop yields. RNA sequencing (RNA-Seq) provides an insight into responses of all of the organism genes to the environmental and developmental cues and thus provides insights into underlying modes of action. In this chapter, we guide a researcher through some of the most important steps in the analysis of transcriptomics data. The initial topic of experimental design is followed by a more wet-lab oriented section on RNA-Seq sample preparation. Next, we present intermediate steps of data retrieval, quality control, mapping and differential expression of the dataset and a section on how to expose your data to the public (i.e. public repositories) and making it findable, accessible, interoperable and reusable (FAIR). In the last four sections, we describe specific tools or web applications, which ease exploration of generated results in the context of their gene function and network-based visualizations, specifically GoMapMan, GSEA, DiNAR and Biomine Explorer. All sections are accompanied by potato dataset examples and include general hints and tricks, as well as potato specificities that one should be aware of.

Key words: transcriptomics, RNA-Seq, RNA, potato, gene mapping, bioinformatics, enrichment analysis, visualization, networks

1. Introduction

Understanding the molecular mechanisms of plant development and responses to environmental stressors is of utmost importance for achieving stable crop yields. High-throughput omics approaches enable analyses of the whole biological system. Due to rapid development and low price of the next-generation sequencing approaches, RNA sequencing (RNA-Seq) based transcriptome analysis enables simultaneous quantification of all the components of the system (i.e. all the genes that are transcribed at a given point in time) in a large number of samples, capturing spatial and temporal regulation as well as single cell-level analysis. The generated high-resolution datasets enable exploration by systems biology approaches to decipher the underlying mechanisms ***(1)***.

In comparison to the plant model organism *Arabidopsis thaliana*, analysis of the potato transcriptome was lagging behind. One of the reasons lies in the complexity of the potato genome, as the cultivated cultivars are highly heterozygous autotetraploids ***(2)***. Although some findings on the molecular mechanisms can be transferred from model to crop species using orthologous information ***(3)***, this is not always the case. It is therefore necessary to perform such studies in potato as well. So far, genome models exist only for the double monoploid clone from *Solanum tuberosum* group Phureja ***(2)***, while for ssp. *tuberosum* the complete genome of only the diploid homozygous genotype Solyntus is available ***(4)***. Recently, a cultivar pan-transcriptome for three ssp. *tuberosum* genotypes (Désirée, PW363 and Rywal) was published ***(5)*** (see Chapter 8).

In this chapter, we provide protocols for RNA-Seq gene expression analysis, from guidelines for experimental design and sample preparation to detailed protocols for quality control, mapping and differential expression analysis. In support of FAIR data management, we provide guidelines for preparing metadata and data deposition to public repositories. Finally, we present tools and methodologies that allow meaningful data interpretation. Content wise, this chapter is well complemented with Merkator4 and network analyses chapters (see Chapter 9 and Chapter 12). Also, if you are interested in the analysis of small RNAs, see Chapter 11.

2. Experimental design of transcriptome analysis

The first step in any transcriptome analysis is the experimental design. At this point, data management usually starts by creating sample metadata (phenodata; see Section 3). There are several aspects to be considered ***(6)***:

a) **Number of samples**: You will want to analyse at least 3 samples per experimental group (treatment/time point/genotype; see Note 1), whereas the optimal cost-benefit number is four.

b) **Sequencing platform**: The most commonly used platform for differential gene expression studies currently is Illumina, which is also used in our example. There are several Illumina sequencers on the market (e.g. MiSeq, NextSeq, HiSeq, NovaSeq) which differ in troughput, sequencing chemistry and available read length options. For eukaryote RNA-Seq studies NextSeq will have enough troughput but HiSeq or NovaSeq sequencers might be preferred in terms of costs.

c) **Sequencing library prep:** In the library prep you want to retain the information on which is the origin strand in the cDNA synthesis step. Such libraries are called stranded libraries and are preferred for differential gene expression studies, because you can take this into account in the mapping step. For eukaryotes, polyA selection is usually apllied to enrich for mRNA. This is generally recommended, unless you are studying the expression of unpolyadenilated non-coding RNA. In that case you can use rRNA removal/depletion procedures. RNA-Seq libraries can also be normalized to reduce the number of highly expressed genes, e.g. using DSN normalization procedure, but this is not recommended for differential gene expression studies, as it affects quantification.

d) **Read length**: The choice of read length for your samples will depend on the application and the final library size. Gene expression analyses benefit from longer reads (above 100 bp), which are less likely to map to multiple genomic loci. For the same reason, choosing paired-end reads over single-end is also desired (see Note 2).

e) **Single or paired-end**: In the single-end (SE) analysis the transcripts are read from a single end, the sequencing is cheaper and usually sufficient for gene expression analysis. Nevertheless, with the genomes which are less perfectly characterized, such as potato, there is an advantage of using paired-end (PE) sequencing (reading the transcript from both ends) as more reads will map to unique genomic loci. Also, you get twice as much data for less than twice the price.

f) **Sequencing depth**: At least 20x coverage is needed for gene expression analysis, hence for potato 20 million reads is usually sufficient (see Note 3).

2.1. Sample preparation

As there are several protocols for RNA preparation available, we here provide just some guidelines that are important for the preparation of potato RNA, suitable for RNA-Seq.

1. **RNA stabilization**: When sampling, you want to preserve RNA by flash freezing the tissue in liquid nitrogen or using RNA stabilization agent, such as RNAlater RNA Stabilization Solution (Thermo Fisher Scientific).
2. **RNA isolation**: Use any RNA isolation method, yielding RNA of sufficient quality and quantity (see point 5). Methods that are customized for plant material might be of advantage. Generally, column-based methods (e.g. RNeasy Plant Mini kit, Qiagen) yield cleaner RNA.
3. **DNase treatment:** All RNA isolation methods have some DNA carryover. DNAse treatment can be included in the RNA isolation protocol or can be done separately as a separate reaction (e.g. using RNAse-Free DNase Set, Qiagen, 0.5 µl DNase per µg RNA).
4. **Purification:** For RNA-Seq, usually additional RNA purification is needed. Here we recommend Zymo RNA Clean & Concentrator-5 kit (Zymo Research) that includes DNase treatment and RNA purification in one step. Optionally, you may want to add RNase inhibitor (Thermo Fisher Scientific) to prevent RNA degradation during shipping to sequencing service.
5. **Quality control (QC):** Typically, the sequencing provider will prescribe the amount, concentration and quality of RNA. Preparation of standard libraries from plant material requires minimally 1 µg of total RNA, with a RIN value above 6.5. In the case of low-input

libraries, the required amount of total RNA can be as low as 20 ng (see Note 4). There are different approaches to assess RNA quantity and quality:

a. Nanodrop (Thermo Fisher Scientific) gives you nucleic acid concentration (prone to errors due to contaminations) and purity estimate expressed as ratios of absorbance at specific wavelengths. While 260/280 ratio below 2 might be an indication of the presence of protein or other contaminants that absorb strongly at or near 280 nm, low 260/230 ratio (<2) indicates EDTA, carbohydrates or TRIzol contamination.
b. A more reliable method for RNA concentration is Qubit Fluorometer (Thermo Fisher Scientific) which relies on dye-specific RNA quantification.
c. The ultimate RNA quality control method is electrophoresis, where a clear separation of rRNA subunits is an indication of intact RNA. While agarose gel electrophoresis might be sufficient, capillary electrophoresis with Bioanalyzer (Agilent) gives you more information with a lower amount of RNA used. In plant leaf RNA samples, besides 18S and 25S rRNA subunits, additional chloroplast rRNA peaks are visible (Figure 1). Therefore, the RNA integrity number (RIN) of perfect-quality RNA is underestimated and in many cases not determined. Still, good quality RNA can be recognised by a stable baseline and high 25S/18S rRNA ratio (Figure 1).

2.2. RNA-Seq: From quality control to differential gene expression

Upon receiving the RNA-Seq data from your sequencing provider, or downloading it from publically available databases (Subsection 2.2.1), you should perform some quality assurance steps to ensure your reads are of appropriate quality (Subsections 2.2.2, 2.2.3). Next you perform mapping of the reads to the potato genome or transcriptome (Subsection 2.2.4). With mapped reads you can proceed to differential expression analysis (Subsection 2.2.5; Figure 2).

2.2.1. Downloading the Data

Experimental data can be downloaded as per your sequencing provider instructions, or from publically available databases (see Section 3). The provider will send you information on how to access it (e.g. FTP access). Instructions for downloads from public databases are specific, but as an example, we will showcase downloading from the National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA).

1. Find your dataset of interest on SRA: https://www.ncbi.nlm.nih.gov/sra/ (e.g. 'potato').
2. Select a sample of interest, in this example, we will select 'RNA-seq of HB1501 3dpi' in the results list (as it is for presentation purposes only, it should not matter).
3. Clicking on the sample link will bring you to the SRA description page of the sample. Proceed by clicking on the 'Run' identifier link, in our case 'SRR10999774' (Figure 3a).
4. This leads to the Run Browser of SRA, automatically to the 'Metadata' tab. Selecting the 'Data access tab' allows you to download the data files as submitted originally under the 'Original format' section. You then simply download them within your web browser (Figure 3b; see Note 5).
5. The standard file format for sequence reads exchange is generally the FastQ format, a text-based format which stores the information on both a biological sequence (usually nucleotides) and its corresponding quality scores ***(7)***. For each sequence, there will generally be 4 lines in the FastQ file (Figure 4).

2.2.2. File integrity verification

It is good practice to check that the files you have downloaded match with the original ones in the so-called MD5 hash, a compact digital fingerprint of a file. For that, you will need to request the MD5 fingerprints from the provider. On your end, you can then either use a Windows-based MD5 calculator (e.g. WinMD5Free; https://www.winmd5.com/) or use the Linux command 'md5sum'.

As example on our downloaded files, we generated the MD5 sums using the WinMD5Free (v1.20):

```
HB1501-3dpi-2_1.fq.gz   c8e97f7fdc8d334bafd873549f148a6b

HB1501-3dpi-2_2.fq.gz   c5ed105223fe489dd2bade47956ee8ce
```

And the same on a Linux system:

```
md5sum HB1501-3dpi-2_1.fq.gz  c8e97f7fdc8d334bafd873549f148a6b

md5sum HB1501-3dpi-2_2.fq.gz  c5ed105223fe489dd2bade47956ee8ce
```

As you can see, the MD5 sum must be the same for the same file; if not, it can indicate download problems (that the file has changed during transfer), and the files should be re-downloaded (see Notes 6 and 7).

2.2.3. Quality control of the read data

In the next step, you examine the contents of your FastQ files in order to assess the quality of the sequence reads it contains. Several tools for quality control exist, e.g. FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/), MultiQC (https://multiqc.info/) or Fastx toolkit (http://hannonlab.cshl.edu/fastx_toolkit/). For the purpose of this subsection, we will use FastQC (version v0.11.9), a simple to use Java-based tool with graphical user interface.

1. Download the latest FastQC version from https://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc and follow the 'Installation and setup instructions' from their webpage for your particular operating system.
2. Run FastQC in the interactive graphical application mode (run_fastqc.bat), which will allow you to dynamically load FastQ files and view their results (see Notes 8 and 9).
3. Select 'File' > 'Open', select the FastQ files you wish to analyse (in this example, files 'HB1501-3dpi-2_1.fq.gz' and 'HB1501-3dpi-2_2.fq.gz') and wait for the analysis to finish.
4. Once finished, you can save the reports using the 'File' > 'Save Report...' option for each file separately.

FastQC reports on several calculated statistics, which are covered fully in its manual (https://dnacore.missouri.edu/PDF/FastQC_Manual.pdf). Here, we will only point out some most pressing sections to check. General questions to ask yourself are: Do the quality value boxplots look all right (Figure 5)? Does it have several GC peaks, contaminations, adaptors (see Note 10)?

1. **Basic Statistics**: This table (Figure 5a) will give you basic information on your file, particularly on the file encoding, which is done using FastQ (also called Phred) quality scores ***(8, 9)***; see Note 11).
2. **Overrepresented sequences / Adapter Content**: This module will list all sequences which make up more than 0.1% of the total (Figure 5b), thus indicating either a highly biologically significant sequence, or that the library is contaminated or not as diverse as expected. Furthermore, for each overrepresented sequence, the program looks in a database of common contaminants and reports the best found hit (e.g. Illumina or SOLID adapters; see Note 12).
3. **Per base sequence quality**: This section shows you the range of quality values across all bases at each position in the FastQ file. In both files of our example (Figure 5c/d), median per-base quality across the whole read length is higher than Phred 30 (meaning a 99.9% or higher base call accuracy; see Note 13).

2.2.4. Mapping the reads to the potato genome

In order to map the reads of your datasets to the potato genome, you will need the reference genome (i.e. sequences of the potato chromosomes) and a corresponding General Feature Format file (GFF), a tab-delimited text file that describes the annotated genes, transcripts or proteins. These features are described by their unique position on the reference genome (e.g. chromosome number, start and end of the feature and on which strand), how the feature was generated (e.g. *in silico* predictions), its type (e.g. gene, transcript, CDS, exon, miRNA). Depending on the mapper used, you can either use GFF3 files or General Transfer Format files (GTF; which are actually GFF2 files), which contain information on the same features, but are structured differently.

In 2011 the first genome of the potato *Solanum tuberosum* group Phureja DM1-3 516 R44 was published by The Potato Genome Sequencing Consortium (PGSC) and with it also the PGSC gene models [v3.4; 39031 genes; ***(2)***]. Much later, in 2018, the Buell group re-arranged scaffolds not matching any of the 12 chromosomes and predicted an additional 400 PGSC gene models (v4.04; ***(10)***. On the other hand, also in 2011, The International Tomato Genome Consortium (ITAG) predicted their own set of gene models (v1.0; 35004 genes; ***(11)***). As there were noticeable known overlaps between predicted genes, efforts were taken in order to merge them into a single genome model set (PGSC-ITAG-merged; 49322 genes; ***(5)***.

For the purpose of this chapter, the files needed for the analysis have been uploaded to GitHub repository "NIB-SI/PTDA":

- FASTA file containing the chromosome sequences: StPGSC4.04n_2018-01-18.fasta.gz (https://github.com/NIB-SI/PTDA/tree/master/genome-file; upon download, combine them into a single file)
- GTF file containing the merged PGSC-ITAG gene models: StPGSC4.04n_PGSC-ITAG-merged_representative-transcript_genes_2020-01-20.gtf.gz (https://github.com/NIB-SI/PTDA/blob/master/2.2.4_StPGSC4.04n_PGSC-ITAG-merged_representative-transcript_genes_2019-04-23.gtf.gz)

For mapping, there are a plethora of read aligners, the selection of the best depends on the goal and the application, the dataset itself and the organism. For this purpose you can find plenty solutions ***(12, 13)***, but in this protocol, we will here give an example of STAR ***(14)*** usage in a Linux environment. To run, STAR requires the FASTA file (chromosomes or transcripts) and the GTF file mentioned above (if you do not have a GTF file, you can convert a GFF3 file using e.g. Cufflinks).

1. Unarchive the chromosome file and the GTF file:
   ```
   gunzip StPGSC4.04n_2018-01-18.fasta.gz
   gunzip StPGSC4.04n_PGSC-ITAG-merged_representative-
   transcript_genes_2020-01-20.gtf.gz
   ```
2. Download the latest STAR from GitHub repository 'alexdobin/STAR' by following the instructions given (see Note 14) and unarchive it into your working directory.
   ```
   wget https://github.com/alexdobin/STAR/archive/2.7.3a.tar.gz
   tar -xzf 2.7.3a.tar.gz
   ```
3. The next step requires the generation of STAR genome index. The first command ('ulimit') only increases the number of open files allowed by the system, the rest is STAR specific.

Modify the options '--runThreadN' and '--limitGenomeGenerateRAM' appropriately, as they are specific to your local hardware capabilities (the former meaning the number of threads and the latter the amount of RAM available at the machine).

```
ulimit -n 10240
mkdir PGSCITAGmerged star_untrimmed_PGSCITAGmerged
# generate genome indexes
STAR-2.7.3a/bin/Linux_x86_64/STAR --runThreadN 28 --runMode
genomeGenerate --genomeDir PGSCITAGmerged --genomeFastaFiles
StPGSC4.04n_2018-01-18.fasta --sjdbGTFfile StPGSC4.04n_PGSC-ITAG-
merged_representative-transcript_genes_2019-04-23.gtf --sjdbOverhang
100 --limitGenomeGenerateRAM 100000000000
```

4. When the STAR genome index is successfully built, we can proceed by mapping RNA-Seq reads. Input files are provided using option '--readFilesIn'. For paired-end sequencing, the first and second read files have to be delimited by space. The command has to be run for each sample separately (see Note 15). We recommend to set '--outFilterMultimapNmax' which allows a read to map to max 10 loci on the genome. This is to facilitate mapping of reads to genes (or gene regions) with multiple copies in the genome (see Note 16).

```
STAR --runThreadN 28 --genomeDir PGSCITAGmerged --genomeLoad
LoadAndKeep --readFilesIn HB1501-3dpi-2_1.fq.gz HB1501-3dpi-2_2.fq.gz
--readFilesCommand gunzip -c --limitBAMsortRAM 50000000000 --
outFileNamePrefix ./star_untrimmed_PGSCITAGmerged/HB1501-3dpi-2. --
outReadsUnmapped Fastx --outSAMtype BAM SortedByCoordinate --
outFilterMultimapNmax 10 --outFilterMismatchNoverReadLmax 0.05 --
quantMode GeneCounts --quantTranscriptomeBan Singleend
```

The mapping step using the above STAR command will result in several generated files:

- **BAM file format**: Binary version of the Sequence Alignment/Map (SAM) file, containing sorted and indexed reads aligned to your genome. This format is appropriate for use in visualisation software, e.g. Integrative Genomics Viewer (IGV; ***(15)***).
- **MATE file(s) for unmapped reads**: FastQ file(s) for the unmapped reads; one or two files, depending on whether your sequencing is single-read or paired-end (see Note 17). In our example, we used paired-end data.
- **Log files**: Log files of the run details (Log and Log.progress) and the statistics for the mapping (https://github.com/NIB-SI/PTDA/blob/master/2.2.4_HB1501-3dpi-2.Log.out.zip), such as the number of reads that mapped uniquely or not are generated.
- **Reads per gene file**: File containing the read counts for genes specified in the used potato GTF file (Table 1; https://github.com/NIB-SI/PTDA/blob/master/2.2.4_HB1501-3dpi-2.ReadsPerGene.out.tab.zip).

Once you have the 'Reads per gene' files for all your samples, you can extract the unstranded RNA-Seq column (second) from each, merge them into a single counts table (see Note 18; raw_counts.txt in the input folder of https://github.com/NIB-SI/PTDA/blob/master/2.2.5_differential-expression.zip) and carry on with determining which genes are differentially expressed given different onditions.

2.2.5. Differential Expression in R

The main goal of this step in the pipeline is finding the effects of various conditions or treatments on gene expression. In this section, we will be using a part of data from Lukan et al., 2020 ***(16)*** (GSE142002), comparing control and PVY- inoculated potato cv. Rywal leaf tissue sections (3 samples per group). RNA-seq was performed on the Illumina HiSeq platform using 150-bp paired-end reads. R software (v3.6.0; ***(17)*** will be used in this section in the Windows environment. The sample file used can be downloaded from GitHub NIB-SI/Potato_transcriptome_data_analysis repository. The pipeline is as follows (deg-script.r in https://github.com/NIB-SI/PTDA/blob/master/2.2.5_differential-expression.zip):

1. **Prepare the environment**: Upon installation of R in Windows, load the program and first install required packages. Then load the packages into the working environment and specify the working directory.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
 install.packages("BiocManager")
BiocManager::install("limma"); BiocManager::install("edgeR")
install.packages("stringr")
library("limma")
library("edgeR")
library("stringr")
setwd("C:/PATH/TO/ANALYSIS/FOLDER/CONTAINING/THE/INPUT/SUBFOLDER")
```

Import the raw read counts: We will be using the file 'raw_counts.txt' (in the input folder of https://github.com/NIB-SI/PTDA/blob/master/2.2.5_differential-expression.zip) that was generated at the end of Subsection 2.2.4 (see Notes 18 and 19).

2. The script assumes it is placed in the subfolder /input.

```
counts <- read.table("input/raw_counts.txt", header=T, sep="\t",
row.names="GeneID", stringsAsFactors=FALSE)
```

3. **Define sample groups for later comparisons**: In this step, you define the sample groups based on the order in which they appear in your 'raw_counts.txt' file. So based on the print-out of the 'colnames(counts)' command, we define 3 replicates of control, 'Rywal mock' (ryw_mock), 3 replicates of 'Rywal section A, treated' (ryw_treat_secA) and 3 replicates of 'Rywal, section B treated' (ryw_treat_secB), samples. The order of sample names in this command is important and should be rigorously followed.

```
group <- factor(c("ryw_mock", "ryw_mock", "ryw_mock",
"ryw_treat_secA", "ryw_treat_secA", "ryw_treat_secA",
"ryw_treat_secB", "ryw_treat_secB", "ryw_treat_secB"))
```

Alternatively, sample groups can be read from ‘phenodata.txt’ tab-delimited file. In this case, the read.table function is used to store the table as an R object. The all function checks if the ‘counts’ and ‘phenodata’ R objects have the same sample order. If it returns TRUE, sample groups can be defined as follows. In case the all function returns FALSE, correct the phenodata sample order.

```
phenodata <- read.table("../analytes.txt", row.names=1, header =
TRUE, sep = "\t")
all(rownames(phenodata)==colnames(counts))
group <- as.factor(phenodata$group)
```

4. **Specify library sizes for each sample**: One of the two most common options here are to either (a) take the number of original sequenced reads as they were in the FastQ files prior to mapping them; (b) or take the sum of the total mapped reads (function default). In our case, we will use the latter option.

```
reads.raw <- DGEList(counts=counts, group=group,
lib.size=colSums(counts))
```

5. **Filter low expressed reads and normalise the data**: Read filtering depends on further use. Whereas Gene Set Enrichment Analysis (GSEA, Section 5) requires normalised counts of the full set (48922 genes) for enrichment, for differential expression it is good to filter the low-expressed genes based on raw read counts prior to setting up a linear model ***(18, 19)***. In our example, we only want to keep the genes that have over 50 reads in at least 4 samples (~44% for our case of 9 samples; see Note 20). The results of our filtering are shown in Figure 6.

```
keep.exprs <- rowSums(reads.raw$counts>50)>=4
table(keep.exprs)
reads.norm <- reads.raw[keep.exprs, keep.lib.sizes=TRUE]
```

Using the ‘calcNormFactors’ function a weighted TMM normalization is performed by default. The output of the function are scaling factors for samples that are used to calculate sample's effective library sizes. The ‘voom’ function performs a transformation and outputs log2-counts per million (log2CPM) values. The function adds 0.5 to all normalized counts to avoid calculating the logarithm of zero. The 'model.matrix' function creates a boolean design matrix that assigns samples to groups.

```
reads.norm <- calcNormFactors(reads.norm)
design <- model.matrix(~0+group)
colnames(design) <- c("mock", "treat_secA", "treat_secB")
v <- voom(reads.norm,design)
write.table(cbind(rownames(v$E), v$E),
file="output/results_normalised-matrix_full_GSEA.txt", sep="\t",
quote=FALSE, row.names=FALSE)
```

6. **Differential expression with Empirical Bayes statistics**: Starting with the voom object 'v' from the previous section we can now fit a linear model ('lmFit) for each gene given a set of samples. We can also define the contrast matrix ('makeContrasts') that defines contrasts or comparisons, in our case, we compare the treated leaf section A or B with the mock leaf section samples. Then, given a linear model, we compute estimated coefficients and standard errors of a given contrast set using the 'contrasts.fit' function. Lastly, we compute the moderated *t*-statistic, moderated F-statistic and log odds of differential expression by empirical Bayes moderation of the standard errors towards a common value ('eBayes').

```
fit <- lmFit(v, design)
contrastMatrix <- makeContrasts("treat_secA-mock", "treat_secB-mock",
levels=design)
fit2 <- contrasts.fit(fit, contrastMatrix)
fit2 <- eBayes(fit2)
```

7. **Export the files**: Once the previous steps finish, all that is left is to export the data, which the following commands perform. The final exported table combines the log2 fold-change values (logFC) with corresponding false discovery rate (FDR) corrected p-values for the tests, and adds raw count values for all the samples used in statistical model (results_comparisons-with-

raw-counts.xlsx in the output folder of https://github.com/NIB-SI/PTDA/blob/master/2.2.5_differential-expression.zip).

```
secAvsMock <- topTable(fit2, coef= 1, number=1000000, sort.by="none")
secBvsMock <- topTable(fit2, coef= 2, number=1000000, sort.by="none")
results <- cbind(secAvsMock$logFC, secBvsMock$logFC,
secAvsMock$adj.P.Val, secBvsMock$adj.P.Val)
colnames(results)
colnames(results) <- c(paste(colnames(fit2$contrasts),"_logFC",
sep=""),paste(colnames(fit2$contrasts),"_padj", sep=""))
colnames(results)
rownames(results) <- rownames(secAvsMock)
# add raw expression data and write a file for MapMan
output.results.raw <- merge(x=results, y=reads.norm$counts,
by.x="row.names", by.y="row.names", all.x = TRUE, all.y= FALSE, sort=
FALSE)
head(output.results.raw)
colnames(output.results.raw)[1] <- 'geneID'
write.table(output.results.raw, file="output/results_comparisons-
with-raw-counts.txt", sep="\t", quote=TRUE, row.names=FALSE)
```

Adding functional descriptions to genes (e.g. from GoMapMan): The table from the previous step can be imported into a spreadsheet viewer and further modified (

8.). Alternatively, we can add the functional annotations in R still.

3. Making your data FAIR

To support the reuse of the scholarly data, a concept of findable, accessible, interoperable and reusable (FAIR) principles for data management have been defined ***(20)***. Data management according to FAIR principles is expected to lead to improved knowledge discovery and innovation, with increased subsequent data and knowledge integration and reuse by the community after the data publication process.

FAIR data management should be the first step after you got an idea for the experiment, and before setting up the actual experiment. Starting early also helps with experimental design, as you have to create the **sample metadata file** (also called phenodata), an EXCEL file or a tab-delimited text file that describes your samples. It can be required in analysis pipelines, but also for publication and data deposition to repositories (Section 3.1). All samples you use in your experiment must have unique sample IDs (see Note 21). Apart from the sample ID, which is always in the first column, phenodata file should contain a longer and more descriptive name (easily understandable by biologists), and all relevant sample descriptions, such as species, subspecies and more detailed information on the genotype used (Rywal, NahG-Rywal, ...), treatment (mock, PVY, …), time of the treatment (day time and season in the year might affect your sample), time after treatment (day 1, 2, 3, ...), position of the sample on the plant (upper leaf, ...) and any further information you consider relevant for analyses and reproducibility (Figure 7; see Note 22).

While the key findings of a study are reported in a scientific paper, for the data to be FAIR, it must be shared. For transcriptomics, the concept developed already 20 years ago with microarray technology. At the time, the metadata standard MIAME (Minimal Information about Microarray Experiment) was developed and first repositories were established ***(21)***. Currently, the most important data repositories for gene expression studies are The Gene Expression Omnibus (GEO, ***(22)*** and Array Express ***(23)***. While those only host metadata and normalized count tables, the raw data (RNA-Seq reads) is automatically transferred to the Sequence Read Archive (SRA, ***(24)*** and the European Nucleotide Archive (ENA, ***(25)***.

3.1. Submitting transcriptomics data to GEO

Three types of data should be provided to deposit RNA-Seq based expression studies into the GEO repository: metadata, processed data files and raw data files. The latest instruction and templates for submitting your expression data are available from GEO website (https://www.ncbi.nlm.nih.gov/geo/info/seq.html. We are summarizing the current procedure below:

1. **Prepare the metadata**: Raw or processed data without properly organized metadata are useless. The metadata includes information on the experiment, samples, wet-lab protocols and the data processing pipeline. It is prepared in a spreadsheet form so you can easily copy-paste the necessary data (https://github.com/NIB-SI/PTDA/blob/master/3.1_metadata_spreadsheet.xls). As with phenodata, a unique identifier "Sample name" connects sample metadata with the experimental data (Figure 7).
2. **Prepare processed data file(s):** In the case of an RNA-Seq experiment, this is usually a matrix with raw and normalized gene counts for all the samples (Subsection 2.2.5). Note that sample identifiers/names should be the same as in the metadata spreadsheet and gene identifiers/names should be linked to the reference genome used for mapping reads.
3. **Prepare raw data files:** Those are usually submitted in the FastQ format, containing sequence reads and quality scores (Subsection 2.2.1). Through GEO, the raw data files are deposited to the Sequence Read Archive (SRA) database. Their filenames and checksums are listed in the metadata spreadsheet. In the case of paired-end experiments pairs of raw files for the same sample are listed in the spreadsheet (Figure 7).
4. **Upload Data to GEO:** All three types of data should be uploaded to the GEO FTP server.
5. **Share your accession number**. After the submission is processed, a unique GEO accession number is assigned to the study that you should reference in your publication (see Note 23).

4. GoMapMan for exploring gene functions

After data analysis, the biggest challenge lies in meaningful data interpretation. Here, ontologies are of big help, as they identify the processes that are underlying a specific experimental setup. One of the most known ontologies is the Gene Ontology (GO; http://geneontology.org; ***(26)***, where gene products are assigned to classes (GO terms) with defined relationships between them, covering molecular function, cellular component and biological process. GO terms are arranged as a directed acyclic graph. A single gene product is usually annotated with a multitude of GO terms, which is beneficial for a rich annotation, but the high redundancy can pose difficulties for omics data visualization ***(27)***.

Another hierarchically organized gene ontology is the Kyoto Encyclopedia of Genes and Genomes (KEGG) ontology ***(28)***.

MapMan ontology was developed specifically for plants ***(29)***. It is a simple hierarchical tree structure of terms referred to as “bins”, which describe biological concepts. The major biological processes (e.g., photosynthesis) are encompassed in the top-level bin, and each sub-bin represents a more narrowly focused subprocess or component within the context of the parent bin. MapMan ontology has been recently completely redesigned into version 4, with almost tripled number of bins ***(30)***; also see Chapter 9). The important difference from the previous version 3 is that proteins are assigned to specific bins only given ample evidence. Due to this strictness, only about a third of plant genes are assigned to any of the bins ***(30)***.

Another concept that can help with experimental data interpretation are determined orthologous relationships between plant gene products. Most of the plant genes have been functionally validated in model species *Arabidopsis thaliana*, while for crop species, less data is available. Nevertheless, by assuming that orthologue genes will have their function conserved ***(31)***, ontology annotations can be transferred from the model to related orthologue genes in the crop species.

To aid with organization, curation and visualization of plant gene annotations, we have developed the GoMapMan database (www.gomapman.org, ***(32)*** where protein-coding genes, metabolites and small RNAs are annotated with MapMan ontology terms (version 3; see Note 24). The subsection ‘protein GoMapMan’ (http://protein.gomapman.org/) contains protein-coding genes of several (currently 13, but continuously expanding) plant species, grouped according to various implemented orthologue groupings which enable the transfer of existing knowledge across species ***(32)***. The database functionalities are presented below (see Note 25).

4.1. Visualising genes in the biological and orthology context

To gather additional information about the gene of interest, you can visualize plant protein-coding genes of different plant species grouped according to different orthologue groupings in the MapMan ontology (Figure 8).

1. **Go to Protein Ontology**: From the main GoMapMan site (www.gomapman.org), select Protein GoMapMan and Protein Ontology tab http://protein.gomapman.org/ontology).
2. **Select plant species**: Currently, 13 species are included in GoMapMan. As visualizing them all in parallel will be difficult, you can select just the ones that you are interested in. For potato, gene model stNIB-v1 (combining PGSC Gene Model v3.4, ITAG Potato Gene Model v.1, POCI and StGIv13; ***(32)*** is used (see Note 25).
3. **Select ortholog grouping**: Orthologue identification relies on the algorithm used, but also the number and evolutionary distance of plant species selected for the orthologue prediction. In GoMapMan, 11 different algorithm and species combinations are available. For potato, we recommend the "OCD_all" grouping that combines orthologues identified by the ITAG consortium ***(11)*** using the reciprocal smallest distance, RSD ("ITAG_RSD") and orthoMCL ("ITAG_orthoMCL") algorithms with PLAZA 3.0 dicot subfamilies ("PLAZA_dicots_sub"; ***(33)***. Different identifiers will be shown according to the selected orthologue grouping, while different identifiers combined in the "OCD_all", PLAZA and ITAG groups will contain only ITAG identifiers (eg Sotub01g029420.1.1; see Note 25).
4. **Browse the ontology**: Open sub-bins by clicking on the selected bin. Opened bins are shown as opened and closed as closed folders. Ortholog groups are presented by a white icon. Mouseover a gene name will show its annotation. Select a gene or an ortholog by clicking on the icon and Gene/Ortholog Details view will open.
5. **Search**: By selecting the Search tab (http://protein.gomapman.org/search/gmm), you can search for individual genes, orthologs, and under the option "microarray probes" for all connected transcript information (both alternative gene nomenclature for other gene models and actual microarray probes). Search is available by identifiers or for gene names, also by descriptions and short names.
6. **View Gene details**: For each gene, detailed information, collected from different sources are shown (Figure 9). For potato, iTAG identifiers link to the respective OrcAE (https://bioinformatics.psb.ugent.be/orcae/) entry. The ontology tree annotations (one or more) are linked to the position in the Protein Ontology. All possible ortholog groups that

gene belongs to are shown (with link to the particular group). For potato genes, the corresponding transcripts and POCI microarray probes are listed. Moreover, a variety of annotations with links are shown, such as (GO, PFAM, SMART; see Note 25).

7. **View ortholog details**: This page lists all the genes that belong to the particular group (see Note 25).

4.2. Exports for other tools

An important feature of the GoMapMan database is the export of mappings (positions of genes within the ontology) that are used by a variety of third-party tools for data analysis and interpretation, such as MapMan (see Chapter 9), gene set enrichment analysis (GSEA, Section 5), DiNAR (Section 6), Biomine Explorer (Section 7) and PaintOmics (http://www.paintomics.org/; ***(34)***. On top of that, "general exports" contain an export of the ontology and connections of the genes with transcripts and microarray features for the individual species and orthologue groups for different algorithms. The most current exports are available at the Exports tab (http://www.gomapman.org/export/current). The tool-specific exports contain several mappings, manly differing between the identifiers/gene models used. There are seven different ones for potato (prefixed with stu_; Figure 10) We consider stu_stNIB-v1 the most comprehensive one since it is a combination of PGSC and ITAG genome models, with the addition of POCI and StGI unigene identifiers ***(32)***. The others are derivatives, so stu_Agilent_4x44k is the POCI, ITAG and PGSC are the gene models (PGSC is split into genes, proteins and transcripts). Lastly, PGSC-ITAG-merged is the non-redundant combination of PGSC and ITAG genes based on the actual gene locations on the PGSC-v4.04 chromosomes (as opposed to stNIB-v1 which is based on amino acid sequence similarity disregarding gene's chromosomal locations).

5. Gene Set Enrichment Analysis

Gene Set Enrichment Analysis (GSEA) ***(35)*** is a knowledge-based approach for interpreting genome-wide expression profiles. First, the genes are ranked according to their expression values and then those the lists are queried for pathways or gene groups (gene sets) whose genes are enriched at the top or bottom of the ranked gene list at a given condition, more so than expected by chance alone. This requires information, to which a particular pathway a gene belongs to. GSEA is usually performed using GO ontology, but for plants, MapMan ontology might be a better option, as it is well curated and updated within the GoMapMan site. The latter will thus be used in this example.

5.1. File Preparation for GSEA

For GSEA to run successfully, three file types are required (https://github.com/NIB-SI/PTDA/blob/master/5.1_gsea-input-files.zip):

1. Your **expression dataset** (TXT file format), which is normalized log2 unfiltered data for your samples. We are using all the genes (48922, without any filtering), then calculate normalisation factors to scale the raw library sizes, and then transform the count data to log2-counts per million (logTMM). Finally, these values are then written into a GSEA-ready file ('results_normalised-matrix_full_GSEA.txt').

   ```
   keep.exprs <- rowSums(reads.raw$counts>50)>=0
   ```

   If your data contains blank values, they should be left blank (no n/a, NA, etc.). The file for this example will be 'results_normalised-matrix_reduced_GSEA_antiLOG.txt', with 10 columns (geneID and the 9 samples) and expression values for 48922 genes before the differential gene expression analysis (Subsection 2.2.5).

2. **Phenotype annotation** in a Categorical Class (CLS) file format, which defines your samples according to their phenotype label (e.g. mock or treated samples, time points etc.). In our example, the first row will contain tree numerical values: the first one will be the total number of samples (9), followed by thenumber of distinct classes (3) and last is always 1. The next row will start with '#' and contain the class names, whereas the last row will have the samples described using these class names (order is important, it has to be the same as in the expression dataset file!). To create this CLS file, open a text file editor, write in the three rows as pertaining for your experiment and save it with a '.cls' extension.

   ```
   9 3 1

   # MOCK WILGAA WILGAB
   ```

```
MOCK MOCK MOCK WILGAA WILGAA WILGAA WILGAB WILGAB WILGAB
```

3. **Gene Matrix Transposed (GMT) file or Ontology mapping file**. These files contain information on gene ontological annotations (i.e all positions in the ontology for each gene) in a certain format. You can download the GMT file with MapMan ontology from the GSEA export subsection in GoMapMan: http://protein.gomapman.org/export/current/gsea/stu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmt.gz (see Note 26).

5.2. Running GSEA

1. Download and install GSEA according to their instructions (GSEA v4.0.3 used in this example): https://www.gsea-msigdb.org/gsea/login.jsp
2. Load the files into the program (Figure 11a) by browsing for them or dragging them (then clicking 'Load these files!'. If the import was successful, the 'Object cache' window on the bottom right will be populated with your dataset ('Datasets'), mapping ('Gene set databases') and the comparison list based on the class file ('Phenotypes').
3. Next, select 'Run GSEA' in the left corner, which opens the GSEA parameter selection window (Figure 11b).
    a. **Required fields**: Select your loaded expression dataset, and select your loaded mapping file under the 'Gene matrix (local gmx/gmt) tab of the pop-up window. Leave the number of permutations as default, and select the comparison you want to perform (options are set according to your CLS file; e.g. 'WILGAA_vs_MOCK' or 'WILGAB_vs_MOCK'). Select 'No_Collapse' under Collapse/Remap to gene symbols (see Note 27), permutation type "by gene_set", and leave "chip platform" blank.
    b. **Basic fields**: Here you can name your analysis ('WILGAA_vs_MOCK' or 'WILGAB_vs_MOCK'), select the folder where the results will be saved (by default results will be saved to your user folder). All the other settings were left as default for

this example. See the application manual (https://www.gsea-msigdb.org/gsea/doc/GSEAUserGuideFrame.html) for their description.

c. Select ‘Run’ on the bottom and wait for the results to be generated.

Once the GSEA runs end, all the files will be placed in a folder that you specified in the above step, with a numerical finish (e.g. WILGAA_vs_MOCK.Gsea.1595418540255; https://github.com/NIB-SI/PTDA/blob/master/5.2_gsea-results.zip). The folder within (edb) will contain your input data files, whereas the folder itself will contain several HTML/EXCEL files (information is structured in the same way in both file formats) for each enriched BIN of the analysis. The two most important ones are the ones, that their names start with ‘gsea_report_for’ (gsea_report_for_gsea_report_for_WILGAA_1595418540255.xls; gsea_report_for_MOCK_1595418540255.xls). These two contain up-regulated BINs in WILGAA and up-regulated BINs in MOCK (as our comparison was WILGAA_vs_MOCK). Once opened in EXCEL, they can be examined and sorted and/or filtered for significance (columns titled NOM p-val and FDR q-val) to list the gene sets that were enriched in the mock (upregulated) or other (downregulated) samples. Usually, FDR q-val < 0.05 is used as a cutoff for enriched gene sets. A subsequent sorting by bin can be useful for result interpretation.

5.3. Combine GSEA results from several runs

When performing several separate GSEA runs, it is beneficial to combine results of these runs into a single file for easier comparisons, data interpretation and further analyses.

1. Download the ‘gseaparsing.pl’ script from (in https://github.com/NIB-SI/PTDA/blob/master/5.3_gsea-results-combined.zip) and place it into a location of your choice.
2. Create a folder for your GSEA results file, e.g. ‘input’. Then for each GSEA run you’ve performed, extract the gsea_report xls files, and rename them appropriately by adding the suffixes ‘-up’ and ‘-down’:

```
gsea_report_for_WILGAA_1595418540255.xls > WILGAA_vs_MOCK-up.xls
```

```
gsea_report_for_MOCK_1595418540255.xls > WILGAA_vs_MOCK-down.xls
gsea_report_for_WILGAB_1595418584006.xls > WILGAB_vs_MOCK-up.xls
gsea_report_for_MOCK_1595418584006.xls > WILGAB_vs_MOCK-down.xls
```

3. Run the 'gseaparsing.pl' script three times; each time you define the input folder name ('gseaparse/'), while you change the output format ('-outfmt=') that extracts either the tags statistic ('tags'), the nominal p-value ('pval') and the false discovery rate corrected q-value ('qval').

```
perl gseaparsing.pl input/ -outfmt=tags wilga-gsea_tags.txt
perl gseaparsing.pl input/ -outfmt=pval wilga-gsea_pval.txt
perl gseaparsing.pl input/ -outfmt=qval wilga-gsea_qval.txt
```

4. You can combine each of these files into a single file in a spreadsheet editor and start examining the significant results. An example of the combined file is supplied as 'GSEA-results-combined' (Figure 12; GSEA-results-combined.xlsx in https://github.com/NIB-SI/PTDA/blob/master/5.3_gsea-results-combined.zip).

6. DiNAR

Web application Differential Network Analysis in R (DiNAR; ***(36)***), allows for static and dynamic visualisation of experimental data of different molecular levels. It currently implements the *Arabidopsis thaliana* large comprehensive network ***(3)***, and the plant immune signalling subnetwork translated to potato ***(36)***. It is available online from https://nib-si.shinyapps.io/DiNAR/, with an additional subset of helper scripts from https://github.com/NIB-SI/DiNAR.

6.1. Preparing the import files for DiNAR

1. The script can be downloaded from Github (https://github.com/NIB-SI/DiNAR/tree/master/IDprioritization; https://github.com/NIB-SI/PTDA/blob/master/6.1_potato-prioritization.zip), together with additional example files

and the example of the folder structure. Place this script at the root of your folder in which you will be running the prioritization, and create folders named 'input' and 'output'.

2. The potato network implemented in DiNAR is not on the level of genes, but on the level of gene families. For that reason, you will need to use the supplied converter between gene family ID and the potato genes belonging to that gene family. It is available as a converter table file 'PlantImmuneSignalling_NODES_STU.txt' (in the input folder of https://github.com/NIB-SI/PTDA/blob/master/6.1_potato-prioritization.zip). Place this file in folder 'input'.
3. Lastly, for each comparison, you will need to prepare a list of differentially expressed genes in a separate file, structured as Table 3 (see Subsection 2.2.5). For these files, you create yet another folder titled 'NGS_DE' in the folder 'input/' and place them there (see https://github.com/NIB-SI/PTDA/blob/master/6.1_potato-prioritization.zip for the structure).
4. After that, you just run the whole data prioritisation script, and it will automatically generate the files you require for DiNAR in the folder 'output/organism_NGS_to_PIS'. Details on the prioritization see 'decision tree for ID prioritisation' section on https://github.com/NIB-SI/DiNAR/tree/master/IDprioritization.

6.2. Visualising the data in the web application

Here we will show you an example of visualisation of transcriptomics data in potato immune signaling model.

1. Start by navigating to DiNAR (ShinyApps) and in the section 'Select Network' choose *Solanum tuberosum*. Under the browse option, select the two files prepared in Subsection 6.1 (in folder 'output/organism_NGS_to_PIS'; 'PIS_dinar_comparison_secA-mock.txt', 'PIS_dinar_comparison_secB-mock.txt' of https://github.com/NIB-SI/PTDA/blob/master/6.1_potato-prioritization.zip; Figure 13a).
2. Upload will start automatically, and you will proceed to the second step, where you will need to define the columns which contain geneIDs, measure of statistical significance (e.g. FDR adjusted P values as in our case) and the measure of difference of differential expression (e.g.

log fold change, logFC). In addition, you can change the cut-off values for both measures (Figure 13b).

3. Selecting ‘Proceed’ (Figure 13b) will automatically open the ‘Dynamic-visNetwork’ section (under the header bar option ‘Differential expression per cluster’). Change this to the ‘Dynamic-animatorR’ option under the same header, which will open up a selection menu, where animation speed can be adjusted, (e.g. to 0.2 in this example; Figure 13c). Pressing the triangle below the ‘Time/Condition’ bar, will start displaying the differential expression of various genes in time-frame, or in our case, two different conditions being used for this example (Figure 13c).
4. At any time, the animation can be paused and observed in more detail (Figure 14). Dynamic graphs can be exported by selecting the ‘Create dynamic .html’ option; same for static images (‘Download static image’) in pdf and png formats.

7. Biomine Explorer

Biomine Explorer is a web application that enables interactive exploration of large heterogeneous biological networks constructed from selected publicly available biological knowledge sources ***(37)***. It is available online from https://biomine.ijs.si/. In the example, we will show how to discover connections between genes EIN3 from the ethylene pathway and NPR1 from the salicylic acid pathway.

1. Start by navigating to Biomine Explorer and selecting the ‘plants_2_aug_2018’ database under the database section (see Note 28). Continue by entering your genes of interest into the ‘find term id’ section (Figure 15). Add the target genes to the source (EIN3; Sotub06g030450.1.1 in potato) or to the target (NPR1; Sotub07g011600.1.1 in potato). Depending on the question and results, one can play around with output graph size (under ‘network size’); here we will leave the default.
2. Applying ‘search’ will run Biomine’s underlying link search and prediction algorithm and result in a browsable (‘advanced view’) and downloadable (‘download graph’) graph. The

latter can be used in third-party network visualisation software (e.g. Pajek or Cytoscape; ***(38, 39)***.

3. Biomine Explorer offers several options to examine the network further, such as freezing the graph, or deletion of nodes/edges/singletons (Figure 16). Clicking on the nodes and edges also offers more information, e.g. on the PUBMED IDs from which the information was gathered. Using this approach you might find some new knowledge about the genes of interest and generate new biological hypothesis based on that.

Notes

1. Each additional sample will greatly increase the robustness and reliability of the results ***(40)***.
2. Illumina offers a '*Library Prep and Array Kit Selector*' (https://www.illumina.com/library-prep-array-kit-selector.html), other providers also similarly.
3. Illumina offers a '*Sequencing Coverage calculator*' (https://support.illumina.com/downloads/sequencing_coverage_calculator.html), where you select your specific sequencer and it returns the flow cells and coverage you will require. If you want to perform assembly and/or variant analysis from the same dataset, you should increase the depth to 100x.
4. The provider will usually re-do all the QC steps and consult you about discrepancies. Upon your confirmation, they can still do the analysis even though QC parameters are not perfect. Still, you should avoid this situation.
5. In Linux, you can use the command 'wget'. Note that 'Original data' section might not always be available, in which case NCBI offers the SRA Toolkit for batch downloads, for instructions see https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc.
6. MD5 sum is sensitive to the file format, so the unarchived FastQ file will have a different calculated MD5 value, as the same corresponding archived (e.g. *.gz) file.
7. If you have several files, and do not want to be calling them one by one, you can use the following command which will write them into a single file: md5sum *.gz > md5_example.txt

8. For a large number of files (e.g. 20 or more) with deep coverage, you can increase the memory given to Java Runtime Enviroment (JRE) by modifying the ‘run_fastqc.bat’ file in the unzipped FastQC folder in an appropriate text editor (e.g. Notepad++). The default in v0.11.9 is ‘java -Xmx250m’, so you can increase it to 1GB by changing to ‘java –Xmx1000m’ and so forth.
9. The non-interactive mode of FastQC allows you to specify all the files for analysis in the command line, with automatic generation of HTML reports for each file, without launching a user interface. Also, see Chapter 8 for example of command line usage.
10. Our recommendation is, that for data of unsufficient quality, it makes more sense to repeat the library preparation and re-sequencing of the samples, than to analyse the low quality dataset.
11. Mostly, FastQC is capable of figuring out which encoding your samples are formatted with, and you don’t need to worry about that. Sometimes, e.g. for the early Illumina datasets, the encoding is not always determined correctly, so you need to check it manually.
12. If your sequencing data does contain adapter contamination, the intermediate step is to trim the sequences, i.e. remove adaptors from the end of the reads and adaptor dimers. This can be done using a variety of tools. Some of these tools also include filtering of reads below a user-set Phred quality value (e.g. below 30).
13. Differences between samples could be due to the quality of the sample or the specific methodology used for sequencing. Often the quality of calls on most platforms will deteriorate as the run progresses – it is common to see base calls quality fall into the orange area towards the end of the read.
14. For advanced Linux users, the whole STAR folder can also be placed under a folder which is in your path. This enables you to call the STAR binary executable file from wherever you are positioned in the system. Alternatively, STAR can be installed and run using the Conda package manager.
15. Advanced Linux command line users can write a bash loop to sequentially map several samples.
16. To allow for only unique mapping of reads to the genome using STAR set the parameter ‘--outFilterMultimapNmax’ to 1. The counting algorithm in STAR (turned on by ‘--quantMode GeneCounts’) only counts uniquely mapped reads and discards the multimapped.

17. If unmapped reads will not be used for any other analyses, the option '--outReadsUnmapped Fastx' can be omitted and some disk space saved.
18. When mapping to the same chromosomal reference (i.e. fasta file) and using the same GTF/GFF file, you can merge the read files without cross-checking within files, as STAR keeps the order in which these features appear in the GTF/GFF file. If using various references however, be vary and implement a check that you are connecting appropriate identifiers (e.g. EXCEL vlookup or a script).
19. Depending on the software used for mapping reads, you could have exported only the raw counts (as is the case with STAR). However, other mapping softwares can already normalise the reads and include them in the same export file, e.g. reads/fragments per kilobase million (RPKM/FPKM), counts per million (CPM), trimmed mean of M values (TMM). RPKM and FPKM are not suitable for differential expression statistical models, whereas CPM and TMM are ***(41)***.
20. The threshold selected for 'low' expressed genes depends on the sequencing depth that varies between experiments and analyses, but generally having 5-10 read counts per gene is considered to be very low.
21. Be careful that sample identifiers don't overlap with the IDs of samples in your other experiments or the ones of other people in your lab.
22. For the ease of further analysis, separate all the factors: it is always easier for downstream analyses to have two columns (one for treatment and one for genotype e.g. Rywal in one, PVY in other, than just a single one that combines both e.g.Rywal-PVY).
23. Upon submission, your data can stay private until paper publication. It is also possible to create private links for reviewers.
24. In GoMapMan, we are currently using v3 ontology, as it has a bigger percentage of mapped genes, which allows us to get (although less accurate) information on a bigger portion of our experimental data.

25. For more information see Introduction and Help pages of the ‘protein GoMapMan’ (http://protein.gomapman.org/) and two video tutorials (https://youtu.be/dHSJbk-ow9U, https://youtu.be/m6g_a1jxuHM).
26. As explained for MapMan above, there are 7 different mappings for potato.
27. GSEA offers the option of collapsing (remapping) your expression dataset, which is useful for microarray experiments (where several probes can map to the same gene). For this, it can handle the multiple occurences of expression values either by retaining either only the maximum or the median expression value, which can however lead to unanticipated results. It is also recommended by the developers, to only use the ‘Collapse’ option for the ranked list, if (and only if) all its features are unique and have a one to one correspondence to the gene symbols. With HTS data, mapping to a genome reference, this problem is generally avoided.
28. Biomine Explorer was built on the original Biomine ***(42)***, which contained human specific data. Biomine Explorer was then expanded for plants, as well as keeping the original human datasets.
29. If you create your own username on the application, you will gain access to additional functionalities, as in the ability to save graphs for further modifications or usage.

Acknowledgements

We thank Henrik Krnec for providing the script gseaparsing.pl. This project was supported by the Slovenian Research Agency (grants P4-0165, J4-1777, J4-9302 and Z7-1888), COST actions CA15110 (CHARME) and CA15109 (COSTNET).

References

1. Lavarenne J, Guyomarc’h S, Sallaud C, et al (2018) The Spring of Systems Biology-Driven Breeding. Trends Plant Sci 23:706–720
2. The Potato Genome Sequencing Consortium (2011) Genome sequence and analysis of the tuber crop potato. Nature 475:189–195

3. Ramšak Ž, Coll A, Stare T, et al (2018) Network Modeling Unravels Mechanisms of Crosstalk between Ethylene and Salicylate Signaling in Potato. Plant Physiol 178:488–499

4. Lieshout N van, Burgt A van der, Vries ME de, et al (2020) Solyntus, the New Highly Contiguous Reference Genome for Potato (Solanum tuberosum). G3 Genes, Genomes, Genet 10:3489–3495

5. Petek M, Zagorščak M, Ramšak Ž, et al (2020) Cultivar-specific transcriptome and pan-transcriptome reconstruction of tetraploid potato. Sci Data 7:1–15

6. Papoutsoglou EA, Faria D, Arend D, et al (2020) Enabling reusability of plant phenomic datasets with MIAPPE 1.1. New Phytol 227:260–273

7. Cock PJA, Fields CJ, Goto N, et al (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38:1767–1771

8. Ewing B, Hillier L, Wendl MC, et al (1998) Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment. Genome Res 8:175–185

9. Ewing B and Green P (1998) Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. Genome Res 8:186–194

10. Leisner CP, Hamilton JP, Crisovan E, et al (2018) Genome sequence of M6, a diploid inbred clone of the high-glycoalkaloid-producing tuber-bearing potato species Solanum chacoense , reveals residual heterozygosity. Plant J 94:562–570

11. The Tomato Genome Consortium (2012) The tomato genome sequence provides insights into fleshy fruit evolution. Nature 485:635–641

12. Schaarschmidt S, Fischer A, Zuther E, et al (2020) Evaluation of Seven Different RNA-Seq Alignment Tools Based on Experimental Data from the Model Plant Arabidopsis thaliana. Int J Mol Sci 21:1–17

13. Baruzzo G, Hayer KE, Kim EJ, et al (2017) Simulation-based comprehensive benchmarking of RNA-seq aligners. Nat Methods 14:135–139

14. Dobin A, Davis CA, Schlesinger F, et al (2013) STAR: ultrafast universal RNA-seq aligner. 29:15–21

15. Thorvaldsdottir H, Robinson JT, and Mesirov JP (2013) Integrative Genomics Viewer (IGV):

high-performance genomics data visualization and exploration. Brief Bioinform 14:178–192

16. Lukan T, Pompe-Novak M, Baebler Š, et al (2020) Precision transcriptomics of viral foci reveals the spatial regulation of immune-signaling genes and identifies RBOHD as an important player in the incompatible interaction between potato virus Y and potato. Plant J 104:645–661

17. R Core Team (2017) R Core Team (2017). R: A language and environment for statistical computing. R Found Stat Comput Vienna, Austria URL http//www R-project org/, page R Found Stat Comput

18. Law CW, Alhamdoosh M, Su S, et al (2018) RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR. 5:1–29

19. Anders S, McCarthy DJ, Chen Y, et al (2013) Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. Nat Protoc 8:1765–1786

20. Wilkinson MD, Dumontier M, Aalbersberg IJ, et al (2016) The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 6:1–6

21. Sansone S-A, McQuilton P, Rocca-Serra P, et al (2019) FAIRsharing as a community approach to standards, repositories and policies. Nat Biotechnol 37:358–367

22. Barrett T, Wilhite SE, Ledoux P, et al (2012) NCBI GEO: archive for functional genomics data sets—update. Nucleic Acids Res 41:D991–D995

23. Athar A, Füllgrabe A, George N, et al (2019) ArrayExpress update – from bulk to single-cell expression data. Nucleic Acids Res 47:D711–D715

24. Leinonen R, Sugawara H, and Shumway M (2011) The Sequence Read Archive. Nucleic Acids Res 39:D19–D21

25. Leinonen R, Akhtar R, Birney E, et al (2011) The European Nucleotide Archive. Nucleic Acids Res 39:D28–D31

26. The Gene Ontology Consortium (2019) The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Res 47:D330–D338

27. Klie S and Nikoloski Z (2012) The Choice between MapMan and Gene Ontology for Automated Gene Function Prediction in Plant Science. Front Genet 3:1–14

28. Kanehisa M, Sato Y, Furumichi M, et al (2019) New approach for understanding genome variations in KEGG. Nucleic Acids Res 47:D590–D595

29. Thimm O, Bläsing O, Gibon Y, et al (2004) mapman: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. Plant J 37:914–939

30. Schwacke R, Ponce-Soto GY, Krause K, et al (2019) MapMan4: A Refined Protein Classification and Annotation Framework Applicable to Multi-Omics Data Analysis. Mol Plant 12:879–892

31. Fulton TM, Hoeven R Van der, Eannetta NT, et al (2002) Identification, Analysis, and Utilization of Conserved Ortholog Set Markers for Comparative Genomics in Higher Plants. Plant Cell 14:1457–1467

32. Ramšak Ž, Baebler Š, Rotter A, et al (2014) GoMapMan: integration, consolidation and visualization of plant gene annotations within the MapMan ontology. Nucleic Acids Res 42:D1167–D1175

33. Proost S, Bel M Van, Vaneechoutte D, et al (2015) PLAZA 3.0: an access point for plant comparative genomics. Nucleic Acids Res 43:D974–D981

34. Hernández-de-Diego R, Tarazona S, Martínez-Mira C, et al (2018) PaintOmics 3: a web resource for the pathway analysis and visualization of multi-omics data. Nucleic Acids Res 46:W503–W509

35. Subramanian A, Tamayo P, Mootha VK, et al (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci 102:15545–15550

36. Zagorščak M, Blejec A, Ramšak Ž, et al (2018) DiNAR: revealing hidden patterns of plant signalling dynamics using Differential Network Analysis in R. Plant Methods 14:1–9

37. Podpečan V, Ramšak Ž, Gruden K, et al (2019) Interactive exploration of heterogeneous biological networks with Biomine Explorer. 35:5385–5388

38. Batagelj V and Mrvar A (1998) Pajek – Program for Large Network Analysis. 21:47–57

39. Shannon P (2003) Cytoscape: A Software Environment for Integrated Models of Biomolecular

Interaction Networks. Genome Res 13:2498–2504

40. Schurch NJ, Schofield P, Gierliński M, et al (2016) How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? 22:839–851

41. Zhao S, Ye Z, and Stanton R (2020) Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols. 26:903–909

42. Eronen L and Toivonen H (2012) Biomine: predicting links between biological entities using network models of heterogeneous databases. BMC Bioinformatics 13:1–21

Figure Captions

Figure 1: Typical Bioanalyzer electropherogram of DNase-treated RNA, isolated from potato leaves, analyzed by 2100 Bioanalyzer and RNA 6000 Nano LabChip Kit (concentration = 207 ng/µl, RIN = 7.7, 25S/18S rRNA ratio = 1.8).

Figure 2: Overview of steps in a classic pipeline of transcriptomics data analysis, starting from raw reads to identification of differentially expressed genes and transcripts.

Figure 3: (a) Sequence Read Archive (SRA) information on the selected sample, with information on the experimental design, platform and library preparation details. (b) Metadata tab of the SRA Run Browser page, from which you can download the file containing all the reads for that sample.

Figure 4: FastQ file format example for two sequential sequences. The first line will contain the read record information (header), starting with a '@' followed by a sequence identifier. The second line will contain the raw sequence letters. The third line will begin with a '+' separator and optionally contain the same sequence identifier as the first line. The fourth line will contain ASCII encoded quality values for each nucleotide in the second line. In the fastQ quality encoding, we can deduce the quality score from the ASCII value of each symbol (e.g. letter F with the ASCII value of 70), by deducing 33 from it (hence 37 for our particular example of the letter F being the quality score at that position).

Figure 5: Quality control analysis by FastQC. (a) Basic information given by FastQC on the content of your FastQ file. (b) Example of overrepresented sequences (not related to the used fastq files in our

example, as they did not contain any overrepresented sequences). (c, d) Per base sequence quality calculated two samples, of good and worse quality, respectively; on the X-axis are specific read positions and on the Y-axis the Phred score boxplots. Central red lines represent the median quality value, the blue line the mean quality value, the yellow boxes the inter-quartile range, upper/lower whiskers the 10% and 90% points.

Figure 6: Effect of low-expressed reads filtering on the boxplots (upper panel) and density plots (lower panel) of expression values for the full data set of all genes (left) and the data after filtering (right; 12447 out of 48922 genes).

Figure 7: A snapshot of a metadata spreadsheet for deposition to GEO. Sample and file information for only three samples is shown.

Figure 8: Visualization of plant genes in GoMapMan. The PR1b gene orthologues in Arabidopsis, potato and tomato belonging to bin 20.1.1.7 “stress.biotic.PR-proteins.PR1 (antifungal)”.

Figure 9: Gene details view for the Sotub09g06900 gene in GoMapMan.

Figure 10: Exports from GoMapMan database. Different types of exports available from the database (left). There are 7 different mapping files of potato genes for MapMan (right).

Figure 11: (a) Entry screen of the Gene Set Enrichment Analysis (GSEA) Java application. (b) The parameter selection window of the application.

Figure 12: Combined results of several GSEA runs as viewed in a spreadsheet editor.

Figure 13: a) Selection of the networks and upload of the files in DiNAR. b) Expanded window for defining the parameters of the input files. c) Slider to adjust for animation speed of the network being visualised.

Figure 14: Visualisation of Cluster 1 using our differential expression comparisons of WILGAA vs. mock and WILGAB vs. mock. The x-axis represents the step in time or condition (in this case we only have 2 conditions), and the color scale colors from up-regulated (red) to down-regulated (blue) based on log2FC values of differentially expressed genes.

Figure 15: Search window of Biomine Explorer, where you define the source and target genes (can also be several at the same time). The database used should be “plants”; whereas the network size is dependant on the source and target nodes and is completely adjustable.

Figure 16: Results from the search set in Figure 15; with green nodes representing protein information; red gene information; orange linking to PubMed article identifiers; blue representing orthology information and gray various groupings. The graph can be further modified by freezing the node position, deletion of specific nodes or edges (see Note 29).

Table Captions

Table 1: Example of the first lines of the 'Reads per gene' file, with four columns corresponding to different strandedness options. The first column represents the geneID, followed by three columns representing various strandedness options (unstranded, forward and reverse strand). For visualization simplicity, only 5 genes are shown as example. The first four lines following the header always contain information on the number of reads that were not used in counting - unmapped, multimapped, reads not overlapping with annotated genes (noFeature), and reads that cannot unambiguously be assigned to one gene (N_ambiguous).
Table 2: Example of differential expression analysis results, with gene IDs in the first column, followed by log2 fold-change values for each comparison (treat_secA-mock_logFC, treat_secB-mock_logFC) and the adjusted P values for each comparison (treat_secA-mock_padj, treat_secB-mock_padj). The resulting file, when generated with the above commands, will also contain the logTMM normalized values for all samples; for sake of clarity we do not show them here.
Table 3: Differential expression analysis results in a comparison between treatment (section A of Rywal-NahG) and control. The file contains at minimum 3 columns: gene identifiers (probeID), log fold-change values for the comparison (LogFC_secA_mock) and the adjusted P-values of the comparison (pAdj_secA_mock). The table is only shown partially for brevity; the complete file contains all the comparisons.

Tables

Table 1

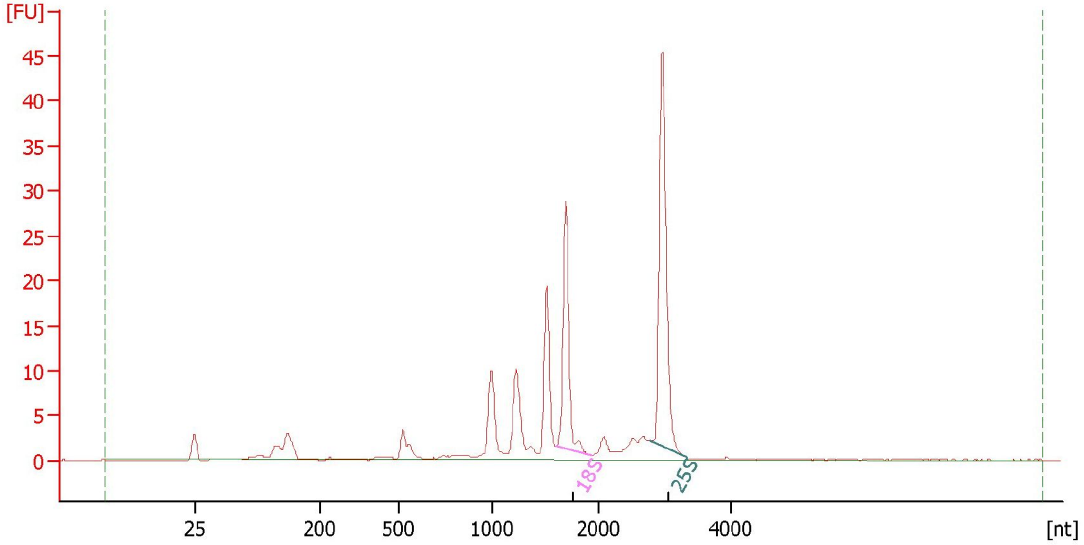
Gene ID	Unstranded	Forward	Reverse
N_unmapped	6231825	6231825	6231825
N_multimapping	952201	952201	952201
N_noFeature	1459482	9706775	9817014
N_ambiguous	383791	115278	113096
PGSC0003DMG400013995	4	2	2
PGSC0003DMG400038530	0	0	0
Sotub09g013810	6	2	4
PGSC0003DMG400042446	0	0	0
…	…	…	…
PGSC0003DMG400040970	0	0	0

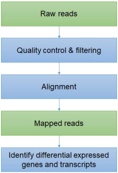
Table 2

geneID	**treat_secA-mock_logFC**	**treat_secB-mock_logFC**	**treat_secA-mock_padj**	**treat_secB-mock_padj**
PGSC0003DMG400000001	0.212806881	0.34224743	0.597742835	0.397772366
PGSC0003DMG400000037	1.094805399	1.104419385	0.021211752	0.024477414
PGSC0003DMG400000048	-0.800759053	-0.535353749	0.009885513	0.05701151
PGSC0003DMG400000108	-0.790719464	-0.755999207	0.004989111	0.008288538

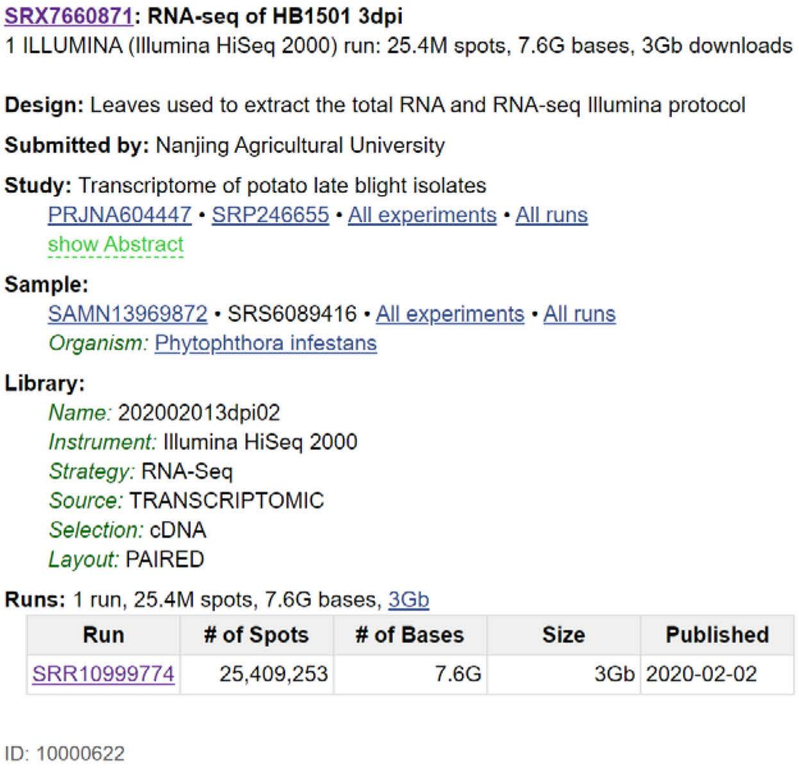
Table 3

probeID	LogFC_secA_mock	pAdj_secA_mock
PGSC0003DMG400000001	0.212806881	0.597742835
PGSC0003DMG400000037	1.094805399	0.021211752
...	...	...
Sotub12g032910	-1.348761902	0.003866681
Sotub12g032950	-0.45213611	0.309225523


[FU]
45
40
35
30
25
20
15
10
5
0
18S
25S
25
200
500
1000
2000
4000
[nt]


Raw reads
Quality control & filtering
Alignment
Mapped reads
Identify differential expressed genes and transcripts

a)



SRX7660871: RNA-seq of HB1501 3dpi
1 ILLUMINA (Illumina HiSeq 2000) run: 25.4M spots, 7.6G bases, 3Gb downloads

Design: Leaves used to extract the total RNA and RNA-seq Illumina protocol

Submitted by: Nanjing Agricultural University

Study: Transcriptome of potato late blight isolates
PRJNA604447 • SRP246655 • All experiments • All runs
show Abstract

Sample:
SAMN13969872 • SRS6089416 • All experiments • All runs
Organism: Phytophthora infestans

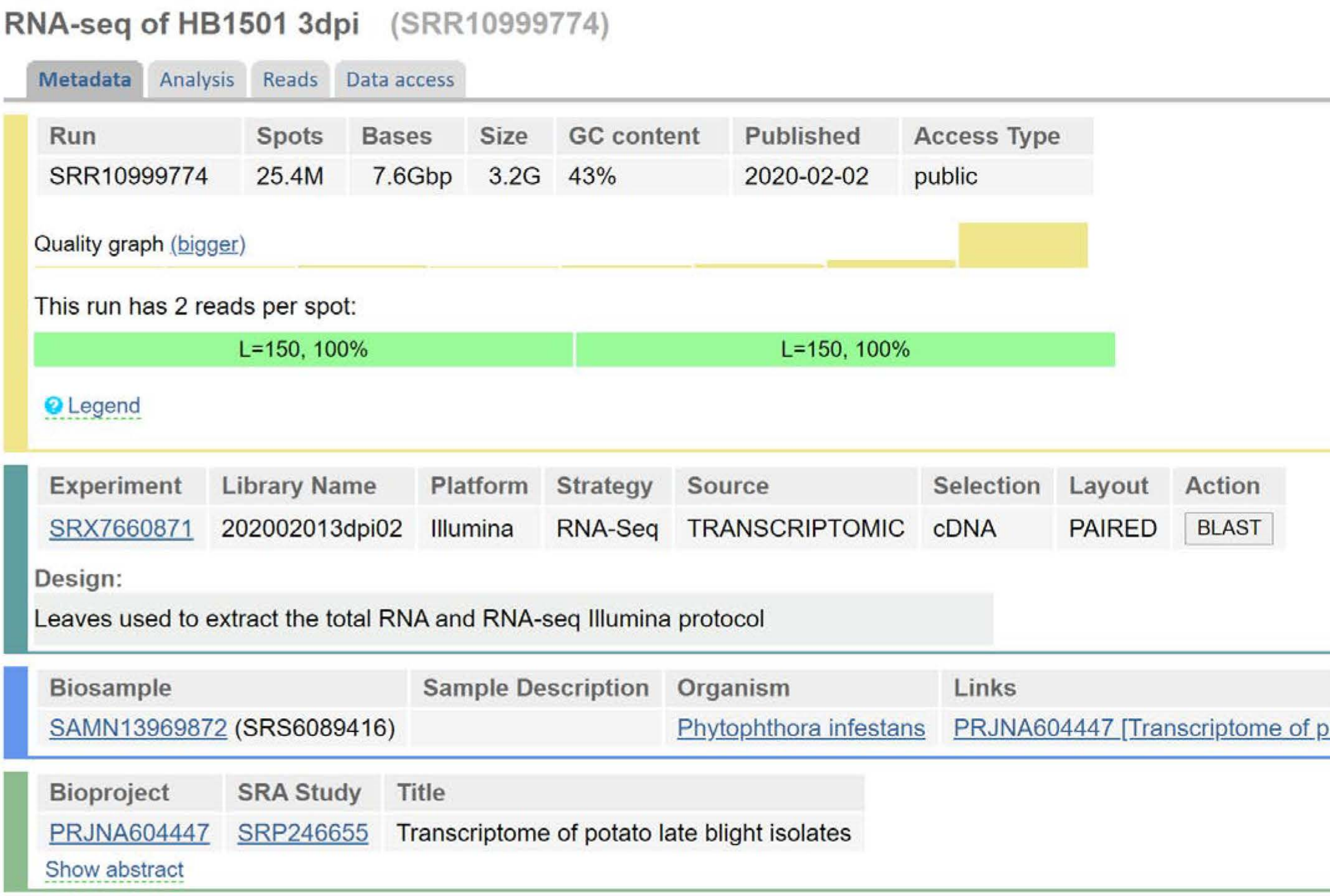
Library:
Name: 202002013dpi02
Instrument: Illumina HiSeq 2000
Strategy: RNA-Seq
Source: TRANSCRIPTOMIC
Selection: cDNA
Layout: PAIRED

Runs: 1 run, 25.4M spots, 7.6G bases, 3Gb

Run	# of Spots	# of Bases	Size	Published
SRR10999774	25,409,253	7.6G	3Gb	2020-02-02

ID: 10000622

b)



RNA-seq of HB1501 3dpi (SRR10999774)

Run	Spots	Bases	Size	GC content	Published	Access Type
SRR10999774	25.4M	7.6Gbp	3.2G	43%	2020-02-02	public

Quality graph (bigger)

This run has 2 reads per spot:

L=150, 100%	L=150, 100%

Legend

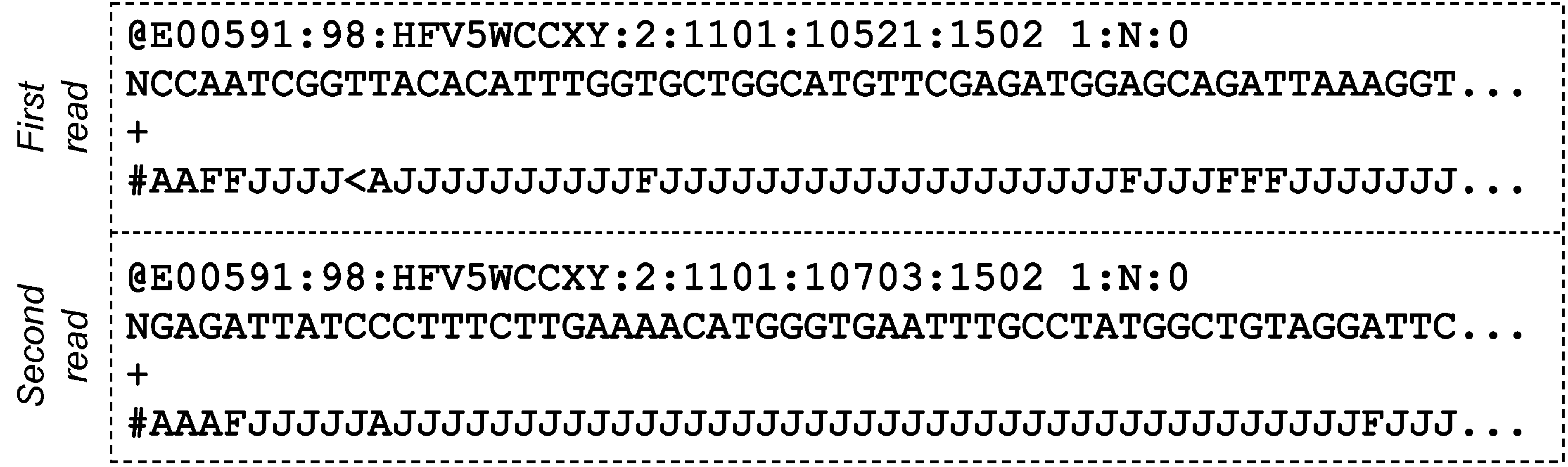
Experiment	Library Name	Platform	Strategy	Source	Selection	Layout	Action
SRX7660871	202002013dpi02	Illumina	RNA-Seq	TRANSCRIPTOMIC	cDNA	PAIRED	BLAST

Design:
Leaves used to extract the total RNA and RNA-seq Illumina protocol

Biosample	Sample Description	Organism	Links
SAMN13969872 (SRS6089416)		Phytophthora infestans	PRJNA604447 [Transcriptome of p

Bioproject	SRA Study	Title
PRJNA604447	SRP246655	Transcriptome of potato late blight isolates

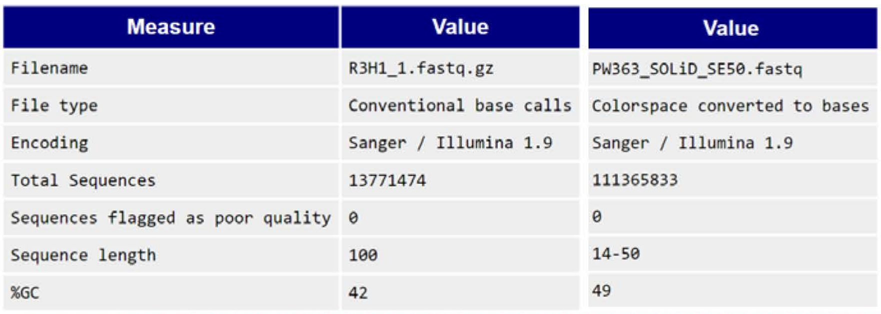
Show abstract


First read
Second read

```
@E00591:98:HFV5WCCXY:2:1101:10521:1502 1:N:0
NCCAATCGGTTACACATTTGGTGCTGGCATGTTCGAGATGGAGCAGATTAAAGGT...
+
#AAFFJJJJ<AJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJFJJJFFFJJJJJJJ...
```

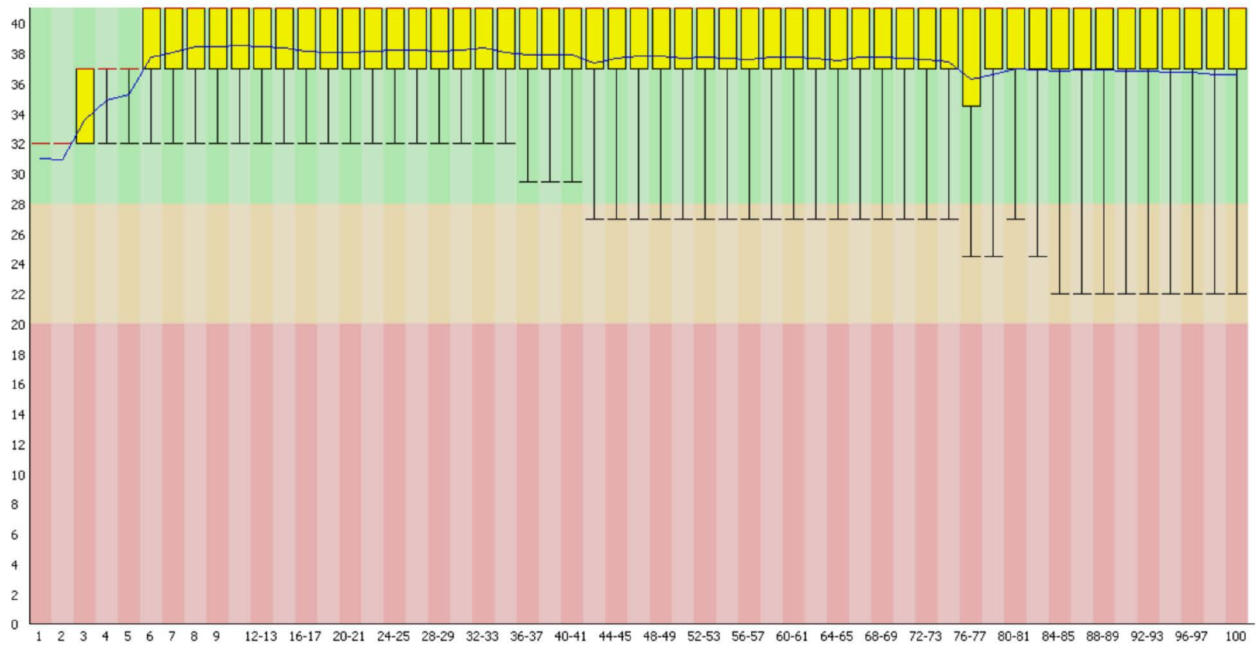
```
@E00591:98:HFV5WCCXY:2:1101:10703:1502 1:N:0
NGAGATTATCCCTTTCTTGAAAACATGGGTGAATTTGCCTATGGCTGTAGGATTC...
+
#AAAFJJJJJAJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFJJJ...
```

a)

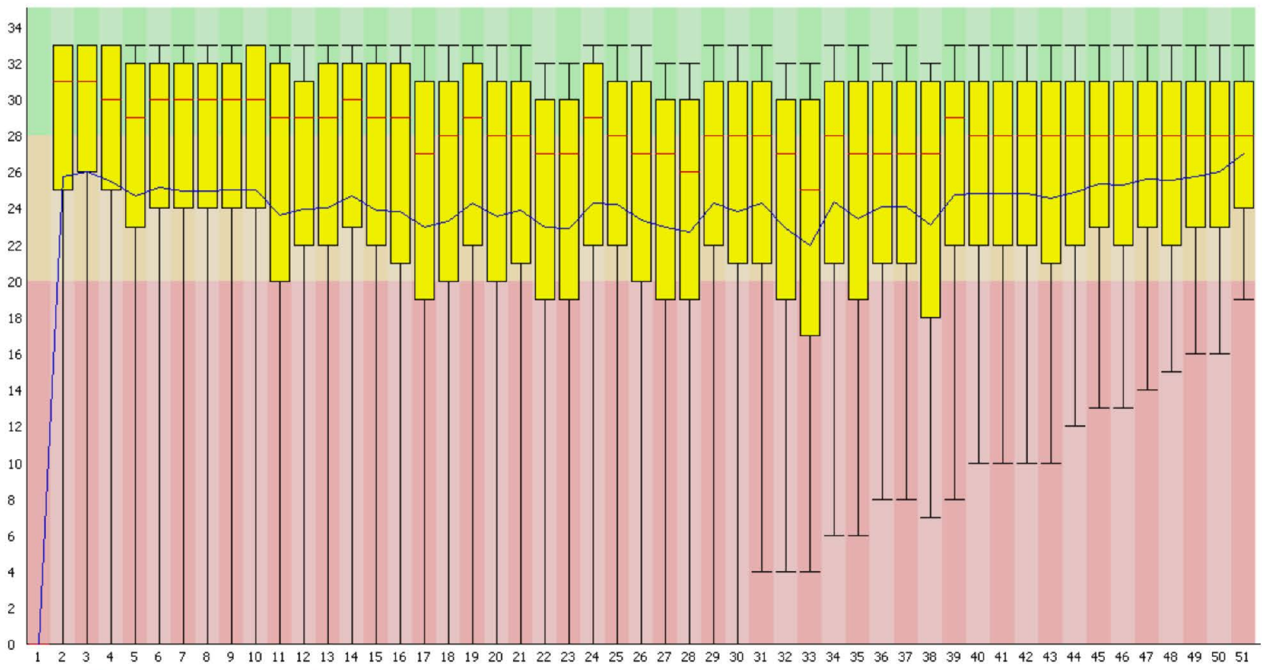


Measure	Value	Value
Filename	R3H1_1.fastq.gz	PW363_SOLiD_SE50.fastq
File type	Conventional base calls	Colorspace converted to bases
Encoding	Sanger / Illumina 1.9	Sanger / Illumina 1.9
Total Sequences	13771474	111365833
Sequences flagged as poor quality	0	0
Sequence length	100	14-50
%GC	42	49

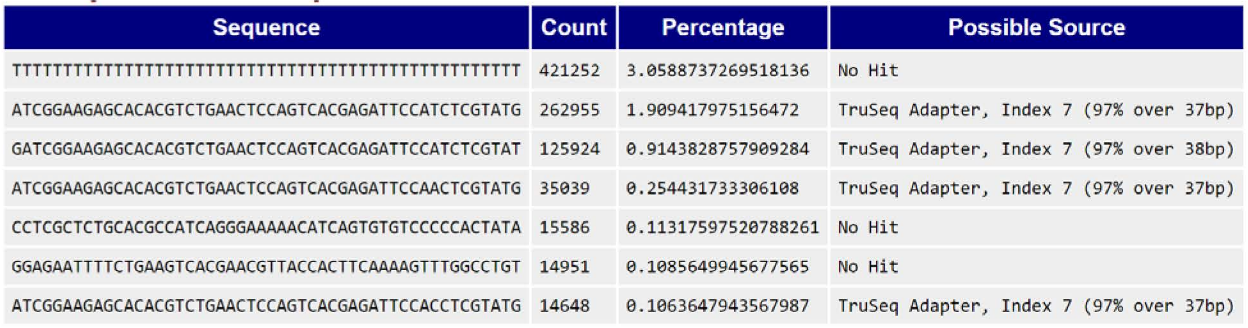
b)



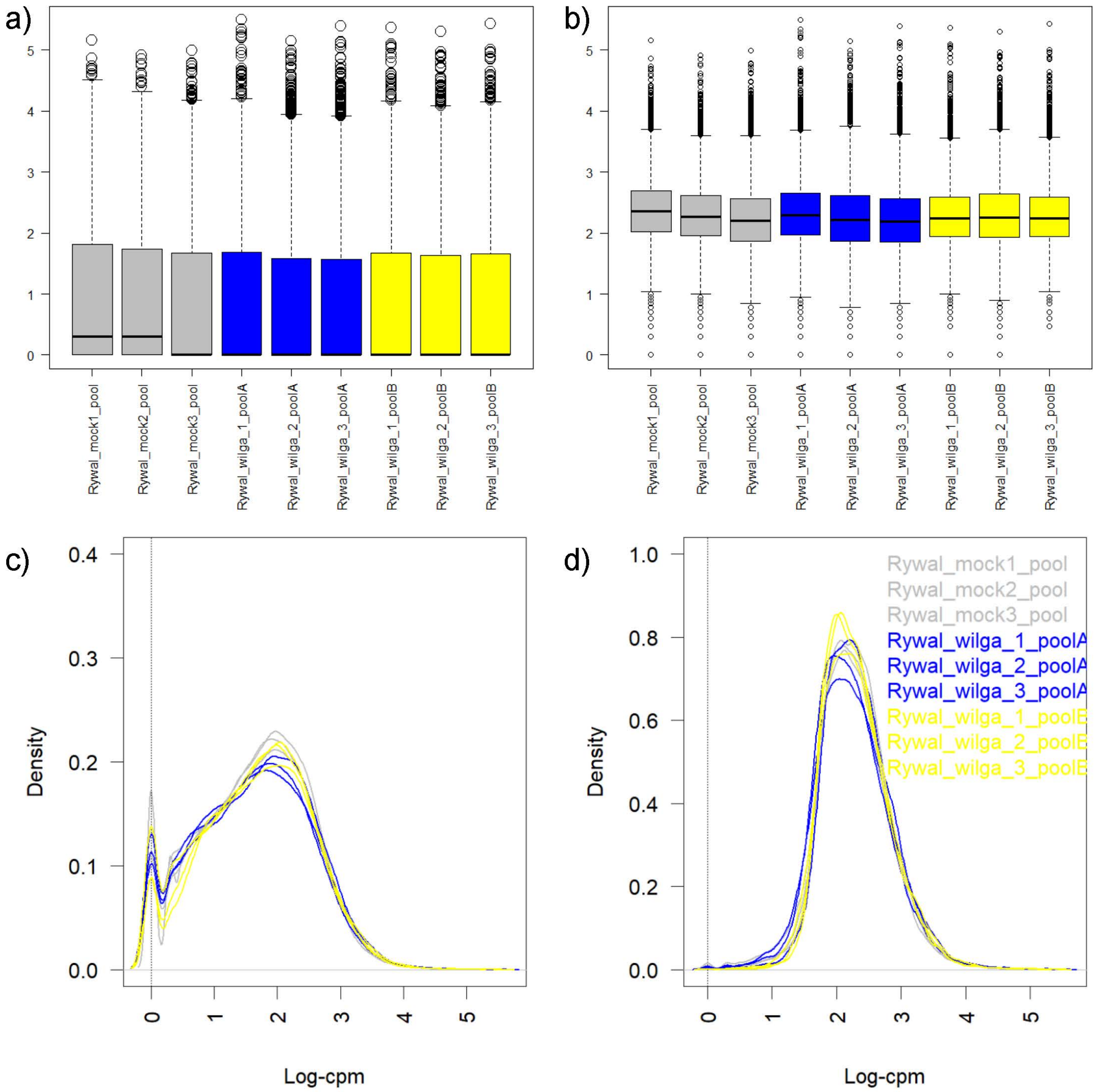
c)

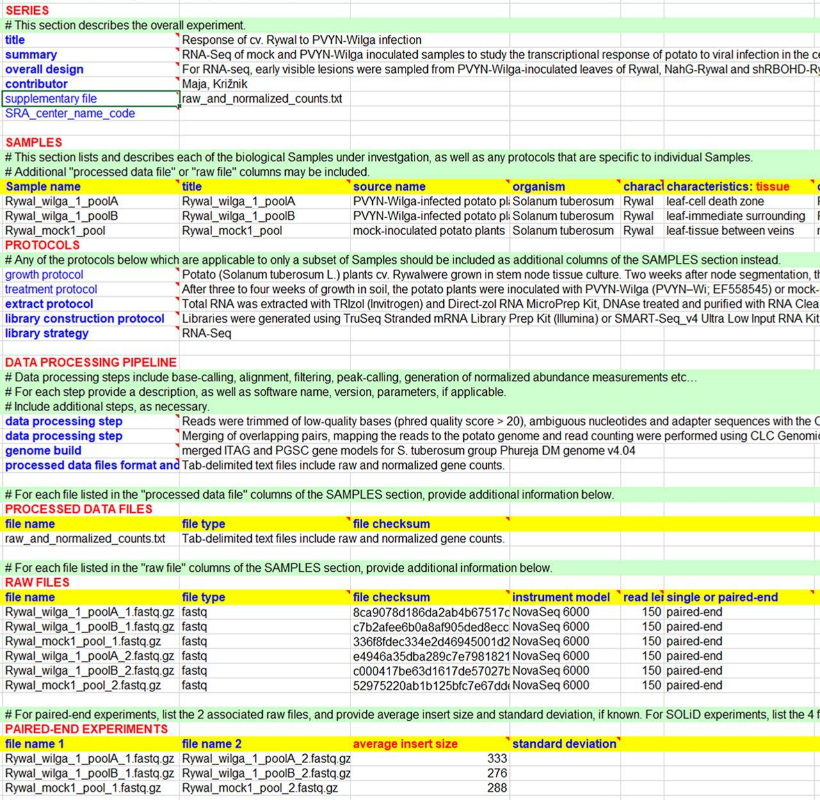


d)



Sequence	Count	Percentage	Possible Source
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	421252	3.0588737269518136	No Hit
ATCGGAAGAGCACACGTCTGAACTCCAGTCACGAGATTCCATCTCGTATG	262955	1.909417975156472	TruSeq Adapter, Index 7 (97% over 37bp)
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGAGATTCCATCTCGTAT	125924	0.9143828757909284	TruSeq Adapter, Index 7 (97% over 38bp)
ATCGGAAGAGCACACGTCTGAACTCCAGTCACGAGATTCCAACTCGTATG	35039	0.254431733306108	TruSeq Adapter, Index 7 (97% over 37bp)
CCTCGCTCTGCACGCCATCAGGGAAAAACATCAGTGTGTCCCCCACTATA	15586	0.11317597520788261	No Hit
GGAGAATTTTCTGAAGTCACGAACGTTACCACTTCAAAAGTTTGGCCTGT	14951	0.1085649945677565	No Hit
ATCGGAAGAGCACACGTCTGAACTCCAGTCACGAGATTCCACCTCGTATG	14648	0.1063647943567987	TruSeq Adapter, Index 7 (97% over 37bp)


a)
b)
c)
d)
Rywal_mock1_pool
Rywal_mock2_pool
Rywal_mock3_pool
Rywal_wilga_1_poolA
Rywal_wilga_2_poolA
Rywal_wilga_3_poolA
Rywal_wilga_1_poolB
Rywal_wilga_2_poolB
Rywal_wilga_3_poolB
Density
Log-cpm



SERIES

This section describes the overall experiment.

title	Response of cv. Rywal to PVYN-Wilga infection
summary	RNA-Seq of mock and PVYN-Wilga inoculated samples to study the transcriptional response of potato to viral infection in the ce
overall design	For RNA-seq, early visible lesions were sampled from PVYN-Wilga-inoculated leaves of Rywal, NahG-Rywal and shRBOHD-Ry
contributor	Maja, Križnik
supplementary file	raw_and_normalized_counts.txt
SRA_center_name_code	

SAMPLES

This section lists and describes each of the biological Samples under investgation, as well as any protocols that are specific to individual Samples.
Additional "processed data file" or "raw file" columns may be included.

Sample name	**title**	**source name**	**organism**	**charact**	**characteristics: tissue**	**c**
Rywal_wilga_1_poolA	Rywal_wilga_1_poolA	PVYN-Wilga-infected potato pl	Solanum tuberosum	Rywal	leaf-cell death zone	F
Rywal_wilga_1_poolB	Rywal_wilga_1_poolB	PVYN-Wilga-infected potato pl	Solanum tuberosum	Rywal	leaf-immediate surrounding	F
Rywal_mock1_pool	Rywal_mock1_pool	mock-inoculated potato plants	Solanum tuberosum	Rywal	leaf-tissue between veins	r

PROTOCOLS

Any of the protocols below which are applicable to only a subset of Samples should be included as additional columns of the SAMPLES section instead.

growth protocol	Potato (Solanum tuberosum L.) plants cv. Rywalwere grown in stem node tissue culture. Two weeks after node segmentation, th
treatment protocol	After three to four weeks of growth in soil, the potato plants were inoculated with PVYN-Wilga (PVYN–Wi; EF558545) or mock-
extract protocol	Total RNA was extracted with TRIzol (Invitrogen) and Direct-zol RNA MicroPrep Kit, DNAse treated and purified with RNA Clea
library construction protocol	Libraries were generated using TruSeq Stranded mRNA Library Prep Kit (Illumina) or SMART-Seq_v4 Ultra Low Input RNA Kit
library strategy	RNA-Seq

DATA PROCESSING PIPELINE

Data processing steps include base-calling, alignment, filtering, peak-calling, generation of normalized abundance measurements etc…
For each step provide a description, as well as software name, version, parameters, if applicable.
Include additional steps, as necessary.

data processing step	Reads were trimmed of low-quality bases (phred quality score > 20), ambiguous nucleotides and adapter sequences with the C
data processing step	Merging of overlapping pairs, mapping the reads to the potato genome and read counting were performed using CLC Genomic
genome build	merged ITAG and PGSC gene models for S. tuberosum group Phureja DM genome v4.04
processed data files format and	Tab-delimited text files include raw and normalized gene counts.

For each file listed in the "processed data file" columns of the SAMPLES section, provide additional information below.

PROCESSED DATA FILES

file name	**file type**	**file checksum**
raw_and_normalized_counts.txt	Tab-delimited text files include raw and normalized gene counts.	

For each file listed in the "raw file" columns of the SAMPLES section, provide additional information below.

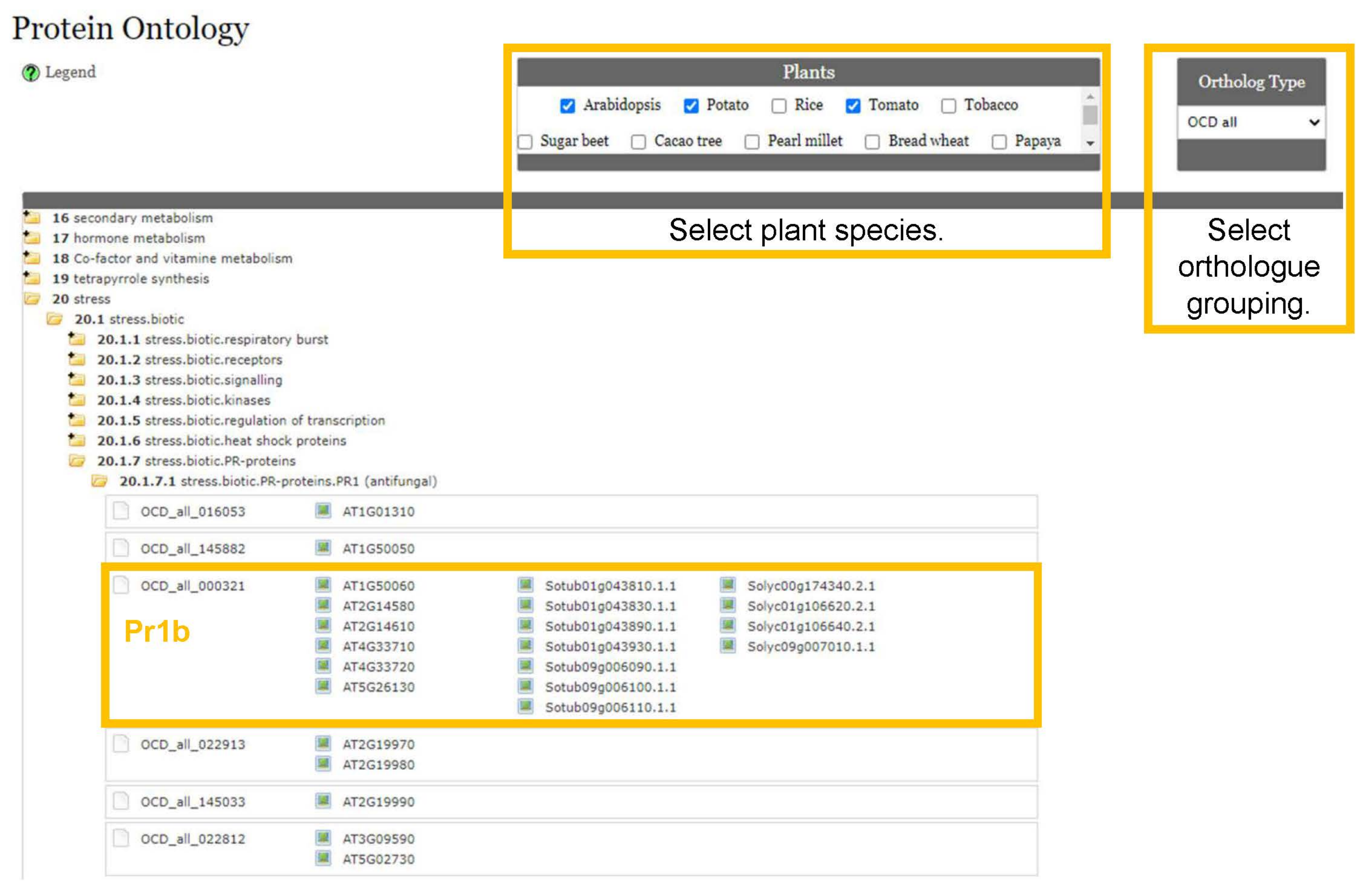
RAW FILES

file name	**file type**	**file checksum**	**instrument model**	**read le**	**single or paired-end**
Rywal_wilga_1_poolA_1.fastq.gz	fastq	8ca9078d186da2ab4b67517c	NovaSeq 6000	150	paired-end
Rywal_wilga_1_poolB_1.fastq.gz	fastq	c7b2afee6b0a8af905ded8ecc	NovaSeq 6000	150	paired-end
Rywal_mock1_pool_1.fastq.gz	fastq	336f8fdec334e2d46945001d2	NovaSeq 6000	150	paired-end
Rywal_wilga_1_poolA_2.fastq.gz	fastq	e4946a35dba289c7e7981821	NovaSeq 6000	150	paired-end
Rywal_wilga_1_poolB_2.fastq.gz	fastq	c000417be63d1617de57027b	NovaSeq 6000	150	paired-end
Rywal_mock1_pool_2.fastq.gz	fastq	52975220ab1b125bfc7e67dd	NovaSeq 6000	150	paired-end

For paired-end experiments, list the 2 associated raw files, and provide average insert size and standard deviation, if known. For SOLiD experiments, list the 4 f

PAIRED-END EXPERIMENTS

file name 1	**file name 2**	**average insert size**	**standard deviation**
Rywal_wilga_1_poolA_1.fastq.gz	Rywal_wilga_1_poolA_2.fastq.gz	333	
Rywal_wilga_1_poolB_1.fastq.gz	Rywal_wilga_1_poolB_2.fastq.gz	276	
Rywal_mock1_pool_1.fastq.gz	Rywal_mock1_pool_2.fastq.gz	288	



Protein Ontology

Legend

Plants

[✓ Arabidopsis] [✓ Potato] Rice [✓ Tomato] Tobacco
Sugar beet Cacao tree Pearl millet Bread wheat Papaya

Select plant species.

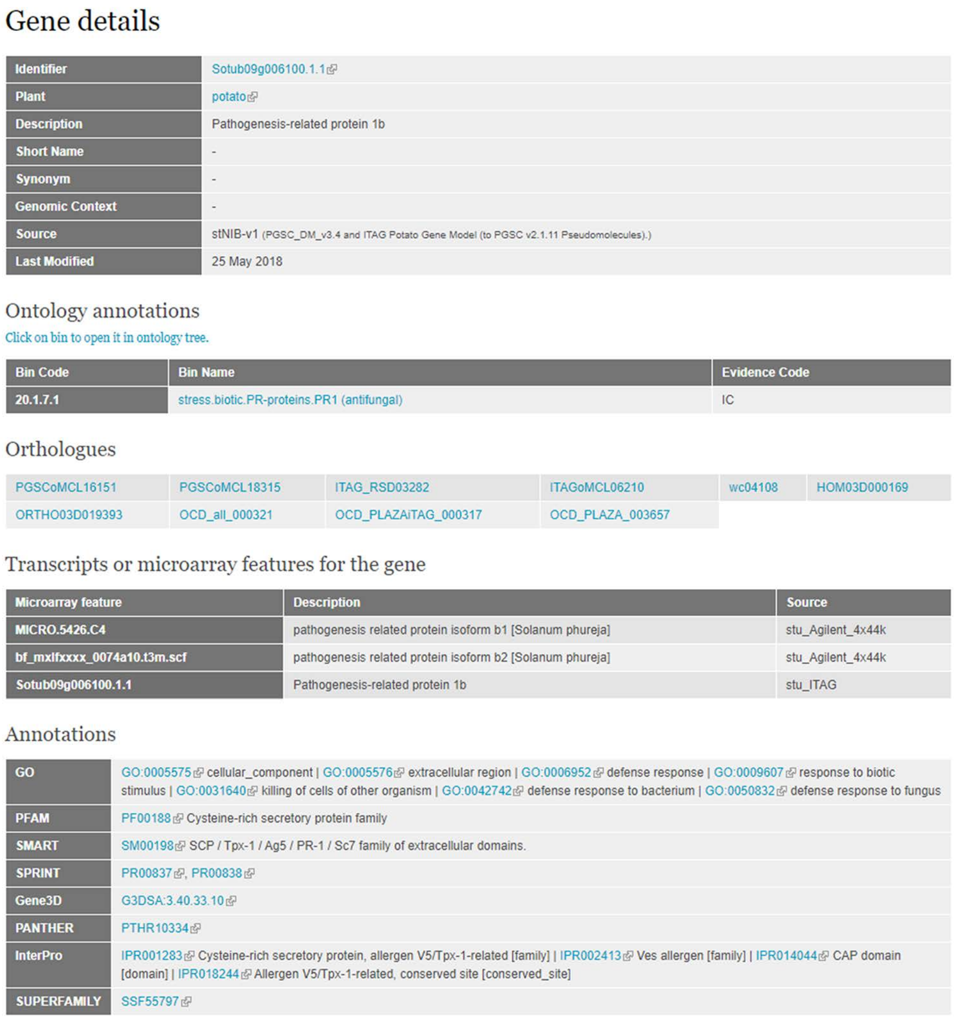
Ortholog Type

OCD all

Select orthologue grouping.

- **16** secondary metabolism
- **17** hormone metabolism
- **18** Co-factor and vitamine metabolism
- **19** tetrapyrrole synthesis
- **20** stress
  - **20.1** stress.biotic
    - **20.1.1** stress.biotic.respiratory burst
    - **20.1.2** stress.biotic.receptors
    - **20.1.3** stress.biotic.signalling
    - **20.1.4** stress.biotic.kinases
    - **20.1.5** stress.biotic.regulation of transcription
    - **20.1.6** stress.biotic.heat shock proteins
    - **20.1.7** stress.biotic.PR-proteins
      - **20.1.7.1** stress.biotic.PR-proteins.PR1 (antifungal)

OCD_all_016053	AT1G01310		
OCD_all_145882	AT1G50050		
OCD_all_000321 Pr1b	AT1G50060 AT2G14580 AT2G14610 AT4G33710 AT4G33720 AT5G26130	Sotub01g043810.1.1 Sotub01g043830.1.1 Sotub01g043890.1.1 Sotub01g043930.1.1 Sotub09g006090.1.1 Sotub09g006100.1.1 Sotub09g006110.1.1	Solyc00g174340.2.1 Solyc01g106620.2.1 Solyc01g106640.2.1 Solyc09g007010.1.1
OCD_all_022913	AT2G19970 AT2G19980		
OCD_all_145033	AT2G19990		
OCD_all_022812	AT3G09590 AT5G02730		



Gene details

Identifier	Sotub09g006100.1.1
Plant	potato
Description	Pathogenesis-related protein 1b
Short Name	-
Synonym	-
Genomic Context	-
Source	stNIB-v1 (PGSC_DM_v3.4 and ITAG Potato Gene Model (to PGSC v2.1.11 Pseudomolecules).)
Last Modified	25 May 2018

Ontology annotations

Click on bin to open it in ontology tree.

Bin Code	Bin Name	Evidence Code
20.1.7.1	stress.biotic.PR-proteins.PR1 (antifungal)	IC

Orthologues

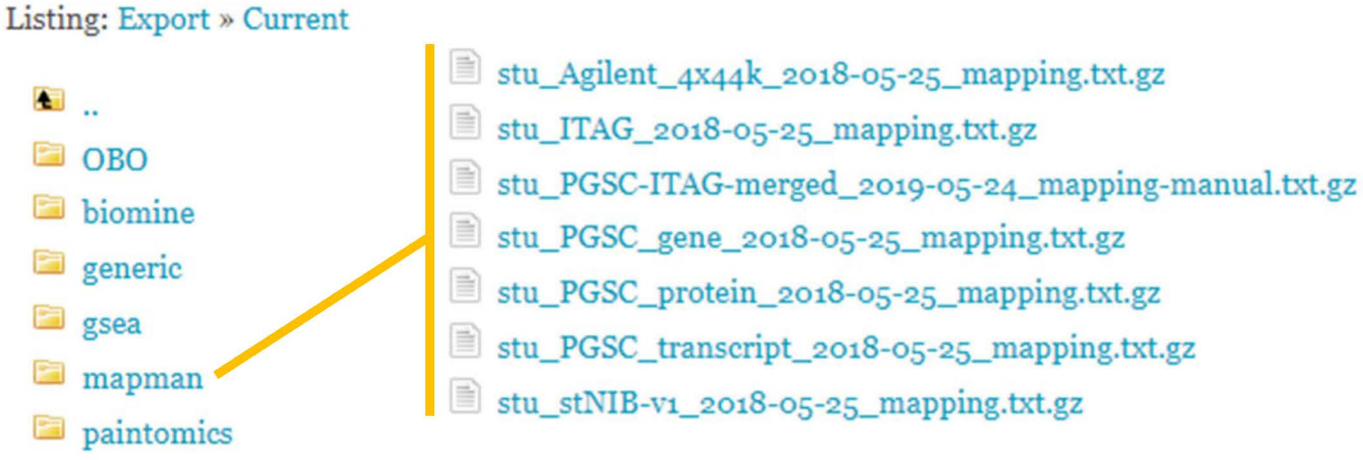
PGSCoMCL16151	PGSCoMCL18315	ITAG_RSD03282	ITAGoMCL06210	wc04108	HOM03D000169
ORTHO03D019393	OCD_all_000321	OCD_PLAZAiTAG_000317	OCD_PLAZA_003657		

Transcripts or microarray features for the gene

Microarray feature	Description	Source
MICRO.5426.C4	pathogenesis related protein isoform b1 [Solanum phureja]	stu_Agilent_4x44k
bf_mxlfxxxx_0074a10.t3m.scf	pathogenesis related protein isoform b2 [Solanum phureja]	stu_Agilent_4x44k
Sotub09g006100.1.1	Pathogenesis-related protein 1b	stu_ITAG

Annotations

GO	GO:0005575 cellular_component | GO:0005576 extracellular region | GO:0006952 defense response | GO:0009607 response to biotic stimulus | GO:0031640 killing of cells of other organism | GO:0042742 defense response to bacterium | GO:0050832 defense response to fungus
PFAM	PF00188 Cysteine-rich secretory protein family
SMART	SM00198 SCP / Tpx-1 / Ag5 / PR-1 / Sc7 family of extracellular domains.
SPRINT	PR00837, PR00838
Gene3D	G3DSA:3.40.33.10
PANTHER	PTHR10334
InterPro	IPR001283 Cysteine-rich secretory protein, allergen V5/Tpx-1-related [family] | IPR002413 Ves allergen [family] | IPR014044 CAP domain [domain] | IPR018244 Allergen V5/Tpx-1-related, conserved site [conserved_site]
SUPERFAMILY	SSF55797

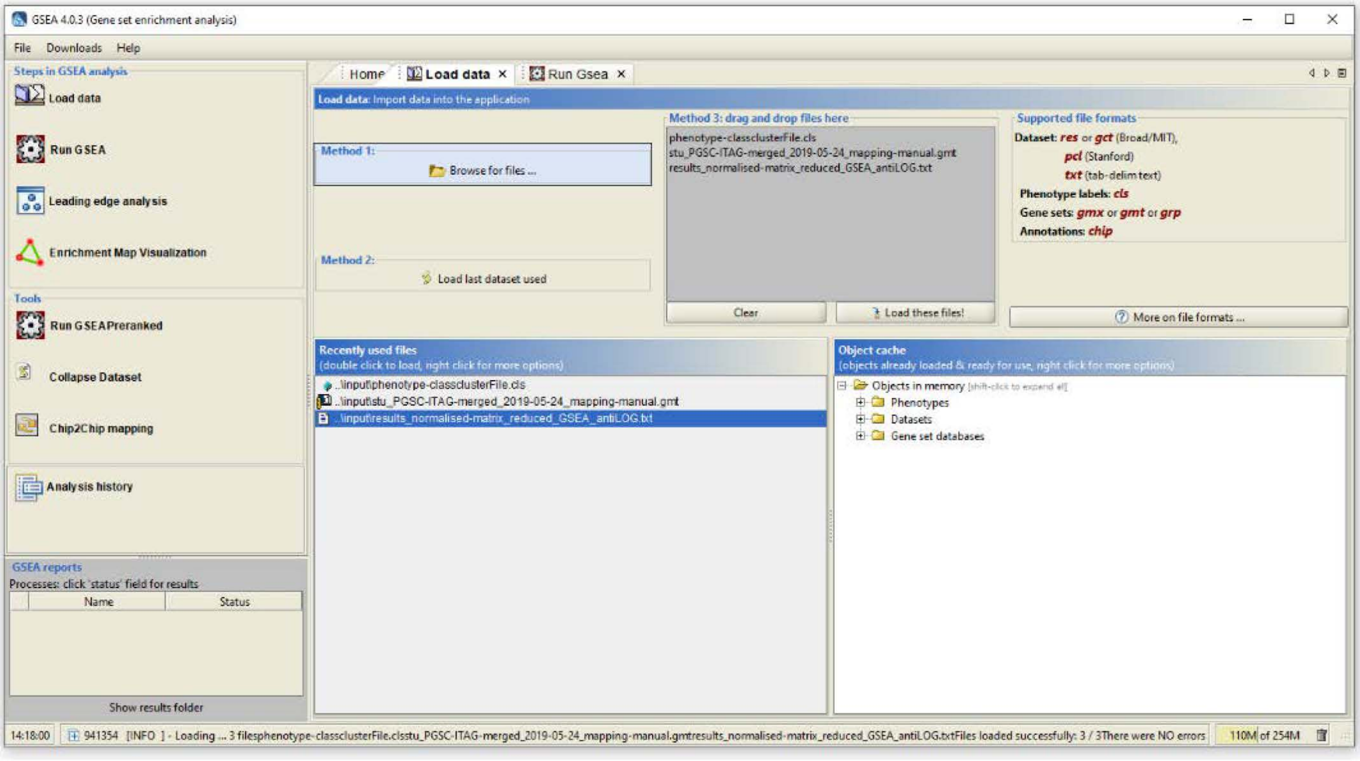


Listing: Export » Current

- ..
- OBO
- biomine
- generic
- gsea
- mapman
- paintomics

- stu_Agilent_4x44k_2018-05-25_mapping.txt.gz
- stu_ITAG_2018-05-25_mapping.txt.gz
- stu_PGSC-ITAG-merged_2019-05-24_mapping-manual.txt.gz
- stu_PGSC_gene_2018-05-25_mapping.txt.gz
- stu_PGSC_protein_2018-05-25_mapping.txt.gz
- stu_PGSC_transcript_2018-05-25_mapping.txt.gz
- stu_stNIB-v1_2018-05-25_mapping.txt.gz

a)



Load data: Import data into the application

Method 3: drag and drop files here
phenotype-classclusterFile.cls
stu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmt
results_normalised-matrix_reduced_GSEA_antiLOG.txt

Supported file formats
Dataset: *res* or *gct* (Broad/MIT),
pcl (Stanford)
txt (tab-delim text)
Phenotype labels: *cls*
Gene sets: *gmx* or *gmt* or *grp*
Annotations: *chip*

Recently used files
(double click to load, right click for more options)
..\input\phenotype-classclusterFile.cls
..\input\stu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmt
..\input\results_normalised-matrix_reduced_GSEA_antiLOG.txt

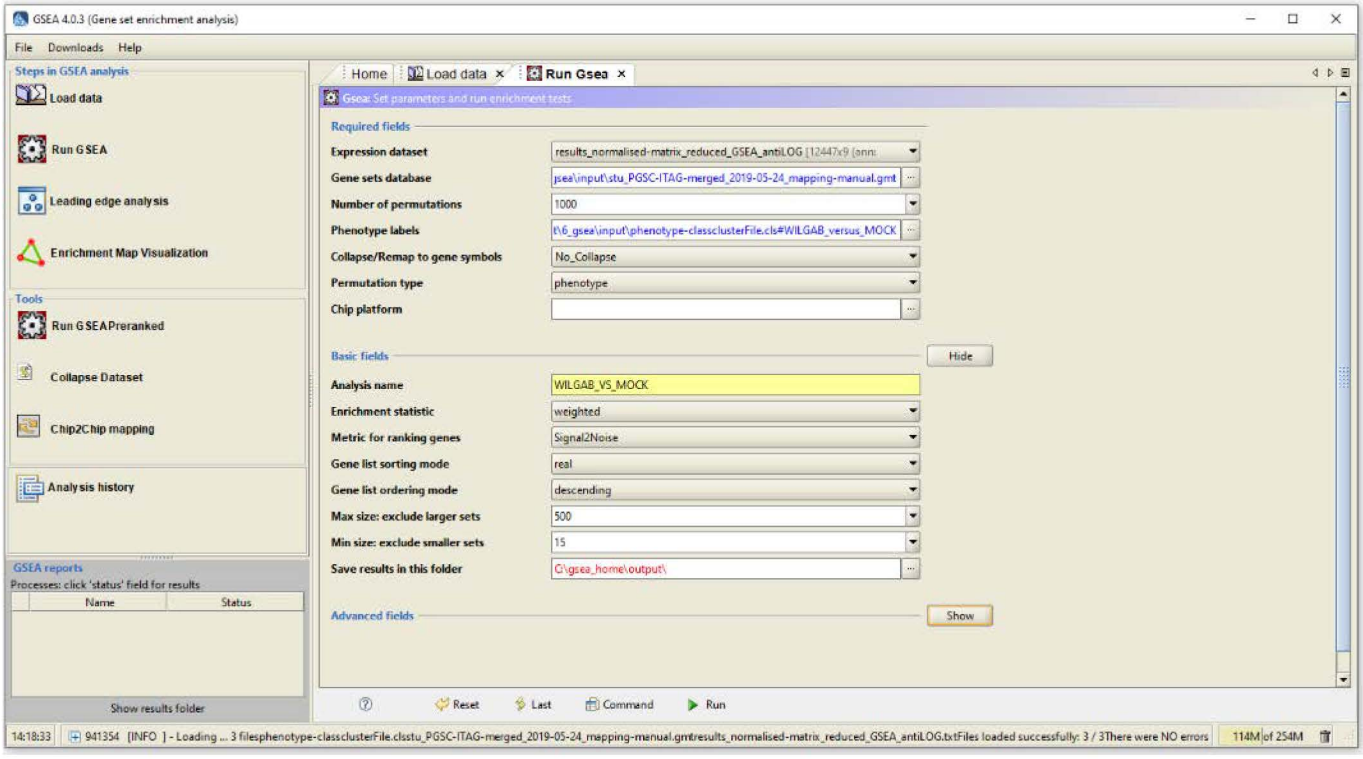
Object cache
(objects already loaded & ready for use, right click for more options)
Objects in memory [shift-click to expand all]
Phenotypes
Datasets
Gene set databases

Processes: click 'status' field for results

Name	Status

```
14:18:00 | 941354 [INFO ] - Loading ... 3 filesphenotype-classclusterFile.clsstu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmtresults_normalised-matrix_reduced_GSEA_antiLOG.txtFiles loaded successfully: 3 / 3There were NO errors | 110M of 254M
```

b)



Gsea: Set parameters and run enrichment tests

Required fields

Expression dataset	results_normalised-matrix_reduced_GSEA_antiLOG [12447x9 (ann:
Gene sets database	sea\input\stu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmt
Number of permutations	1000
Phenotype labels	t\6_gsea\input\phenotype-classclusterFile.cls#WILGAB_versus_MOCK
Collapse/Remap to gene symbols	No_Collapse
Permutation type	phenotype
Chip platform	

Basic fields

Analysis name	WILGAB_VS_MOCK
Enrichment statistic	weighted
Metric for ranking genes	Signal2Noise
Gene list sorting mode	real
Gene list ordering mode	descending
Max size: exclude larger sets	500
Min size: exclude smaller sets	15
Save results in this folder	C:\gsea_home\output\

Advanced fields

Processes: click 'status' field for results

Name	Status

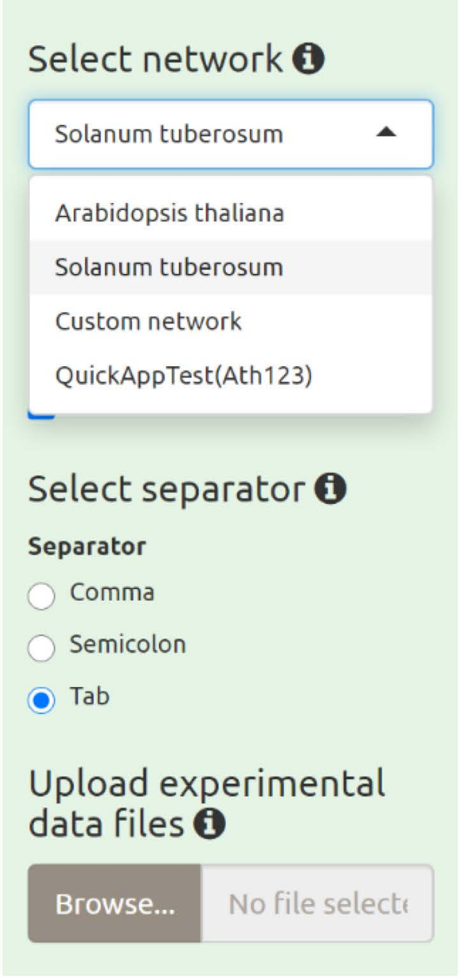
```
14:18:33 | 941354 [INFO ] - Loading ... 3 filesphenotype-classclusterFile.clsstu_PGSC-ITAG-merged_2019-05-24_mapping-manual.gmtresults_normalised-matrix_reduced_GSEA_antiLOG.txtFiles loaded successfully: 3 / 3There were NO errors | 114M of 254M
```

P-value threshold 0.05

Q-value threshold 0.5

BIN NAME	BIN-SIZE	WILGAA_vs_MOCK		WILGAA_vs_MOCK	
		P-value		Q-value	
1 PS	154	66%	64%	66%	64%
1.1 PS.LIGHTREACTION	103	76%	69%	76%	69%
1.1.1 PS.LIGHTREACTION.PHOTOSYSTEM II	46			91%	57%
1.1.1.1 PS.LIGHTREACTION.PHOTOSYSTEM II.LHC-II	22				
1.1.1.2 PS.LIGHTREACTION.PHOTOSYSTEM II.PSII POLYPEPTIDE SUBUNITS	23	83%	83%	83%	83%
1.1.2 PS.LIGHTREACTION.PHOTOSYSTEM I	24	96%			96%
1.1.2.2 PS.LIGHTREACTION.PHOTOSYSTEM I.PSI POLYPEPTIDE SUBUNITS	15	100%		100%	93%
1.2 PS.PHOTORESPIRATION	17	59%	53%	59%	53%
1.3 PS.CALVIN CYCLE	33	73%	67%		67%
2 MAJOR CHO METABOLISM	60	30%	30%	30%	30%
2.1 MAJOR CHO METABOLISM.SYNTHESIS	18	44%	50%		50%
2.1.2 MAJOR CHO METABOLISM.SYNTHESIS.STARCH	15		53%	40%	53%
2.2 MAJOR CHO METABOLISM.DEGRADATION	42				
2.2.1 MAJOR CHO METABOLISM.DEGRADATION.SUCROSE	24	46%			
2.2.2 MAJOR CHO METABOLISM.DEGRADATION.STARCH	18	39%		39%	39%

a)



Select network

Solanum tuberosum

- Arabidopsis thaliana
- Solanum tuberosum
- Custom network
- QuickAppTest(Ath123)

Select separator

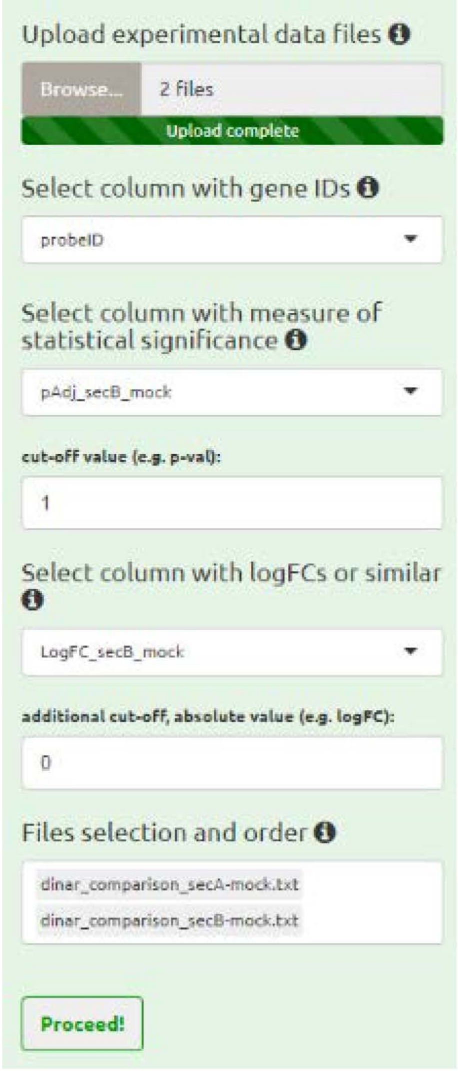
Separator

Comma
Semicolon
[✓ Tab]

Upload experimental data files

Browse... No file selecte

b)



Upload experimental data files

Browse... 2 files
Upload complete

Select column with gene IDs

probeID

Select column with measure of statistical significance

pAdj_secB_mock

cut-off value (e.g. p-val):

1

Select column with logFCs or similar

LogFC_secB_mock

additional cut-off, absolute value (e.g. logFC):

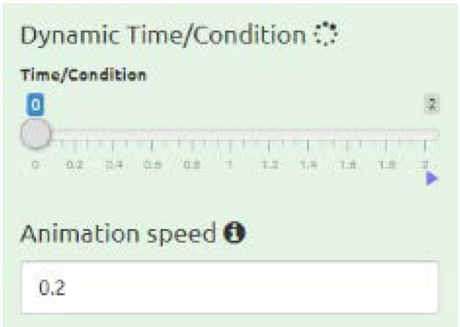
0

Files selection and order

dinar_comparison_secA-mock.txt
dinar_comparison_secB-mock.txt

Proceed!

c)



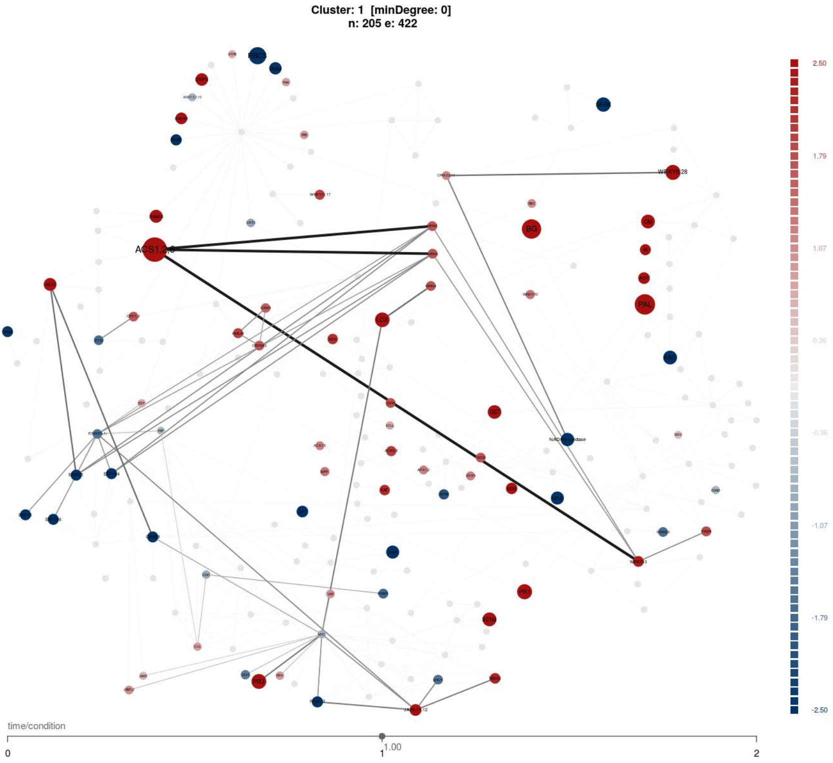
Dynamic Time/Condition

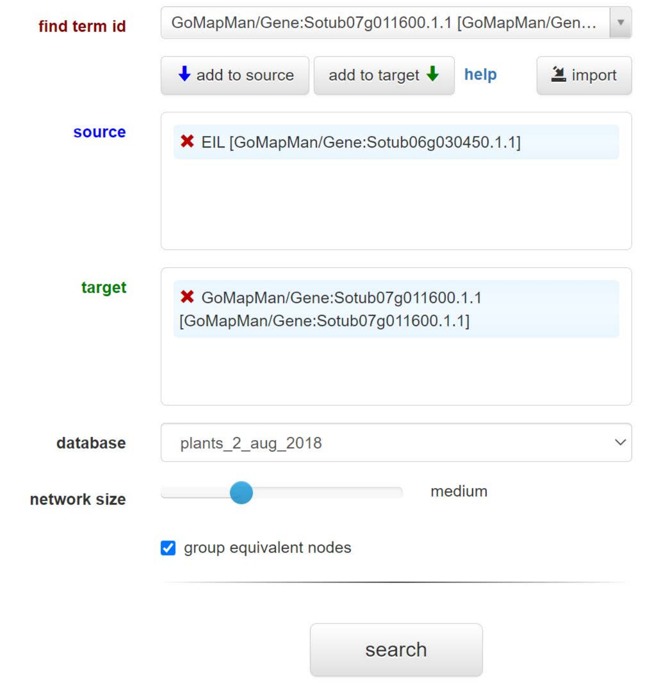
Time/Condition

0

Animation speed

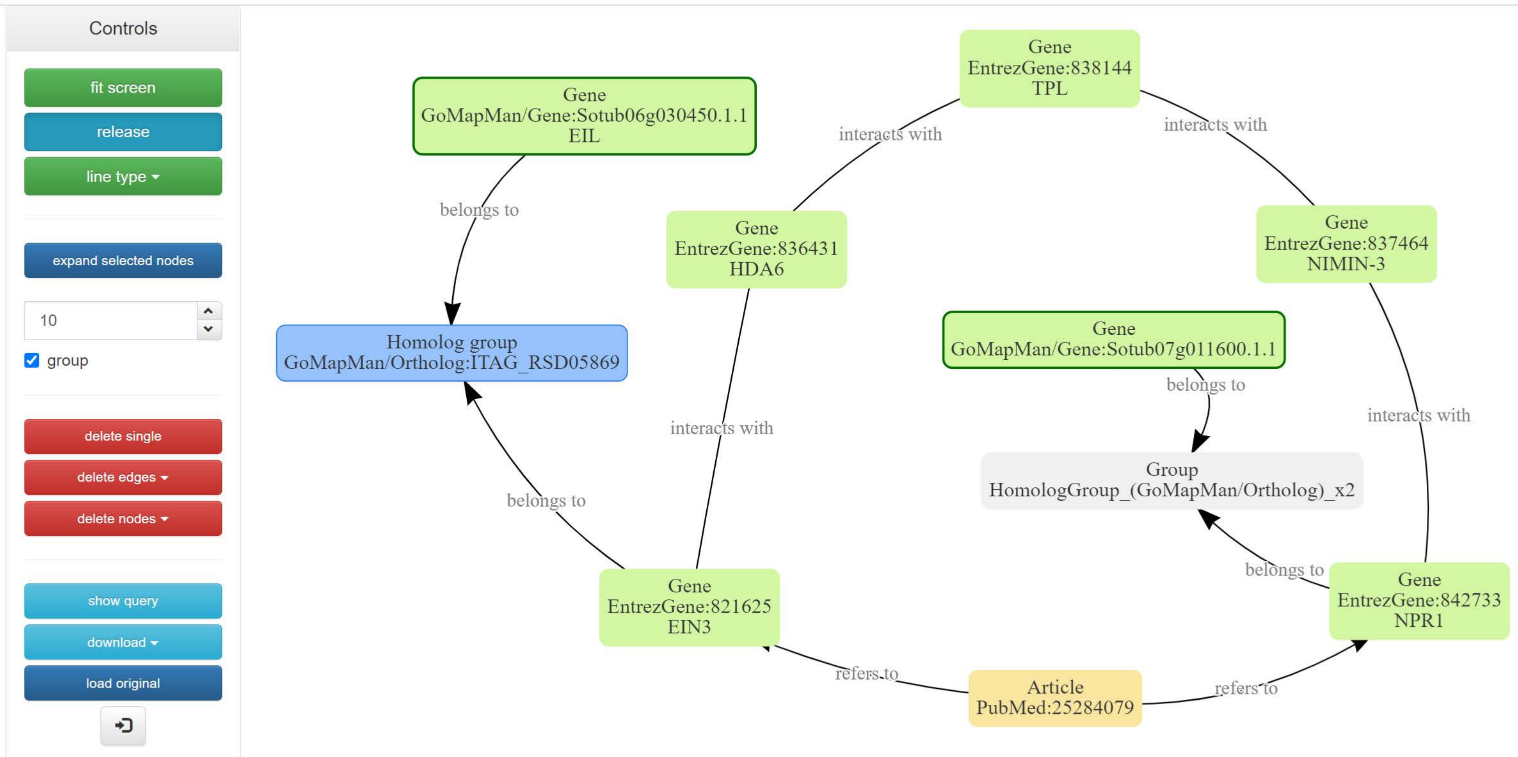
0.2


Cluster: 1 [minDegree: 0]
n: 205 e: 422
RBCS
ACS1,2,6
WRKY8,28
BG
PAL
LOX
GST
PR1
SEN4
PR3
NAD(P)H oxidase
2.50
1.79
1.07
0.36
-0.36
-1.07
-1.79
-2.50
time/condition
0
1
1.00
2



find term id	GoMapMan/Gene:Sotub07g011600.1.1 [GoMapMan/Gen…
source	EIL [GoMapMan/Gene:Sotub06g030450.1.1]
target	GoMapMan/Gene:Sotub07g011600.1.1 [GoMapMan/Gene:Sotub07g011600.1.1]
database	plants_2_aug_2018
network size	medium

[✓] group equivalent nodes

Controls
fit screen
release
line type
expand selected nodes
10
group
delete single
delete edges
delete nodes
show query
download
load original
Gene
GoMapMan/Gene:Sotub06g030450.1.1
EIL
belongs to
Homolog group
GoMapMan/Ortholog:ITAG_RSD05869
Gene
EntrezGene:838144
TPL
interacts with
interacts with
Gene
EntrezGene:836431
HDA6
Gene
EntrezGene:837464
NIMIN-3
Gene
GoMapMan/Gene:Sotub07g011600.1.1
belongs to
interacts with
interacts with
Group
HomologGroup_(GoMapMan/Ortholog)_x2
belongs to
belongs to
Gene
EntrezGene:821625
EIN3
Gene
EntrezGene:842733
NPR1
refers to
Article
PubMed:25284079
refers to