Desarrollo y optimización de un pipeline para el análisis dual del transcriptoma planta/patógeno: identificando los mecanismos de patogenicidad y defensa

Jorge Martín García

Máster en Bioinformática y Bioestadística

Análisis de datos ómicos

Nombre Consultor: Guillem Ylla Bou
Nombre Profesor responsable de la asignatura: David Merino Arranz

05 Junio 2018

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desarrollo y optimización de un pipeline para el análisis dual del trancriptoma patógeno/planta: identificando los mecanismos de patogenicidad y defensa*
Nombre del autor:	*Jorge Martín García*
Nombre del consultor/a:	*Guillem Ylla Bou*
Nombre del PRA:	*David Merino Arranz*
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Análisis de datos ómicos*
Idioma del trabajo:	*Español*
Palabras clave	*Patogenicidad, RNA-seq, expresión génica*

Resumen del Trabajo (máximo 250 palabras):

El conocimiento de los mecanismos de infección y defensa en un patosistema vegetal es clave para el desarrollo e implementación de una estrategia de control. Tradicionalmente, los estudios en patología vegetal estudiaban por separado los mecanismos de infección del patógeno y los de defensa de la planta, en función de los objetivos específicos del trabajo en cuestión. Dificultando notablemente la comprensión de las interacciones entre ambos organismos. Recientemente, los nuevos métodos de secuenciación masiva han permitido desarrollar la técnica de RNA-seq. Sin embargo, aunque su uso ha crecido sustancialmente en la última década por su gran aplicabilidad, todavía son infrecuentes los estudios duales hospedante/patógeno del transcriptoma.

El presente Trabajo Fin de Máster se centra en el análisis dual del transcriptoma del patosistema *Eucalyptus grandis/ Chrysoporthe austroafricana*, testando la resistencia genética del hospedante y el efecto de la infección del patógeno. Se ha desarrollado un pipeline y su correspondiente script para la descarga y procesado de secuencias, y el posterior análisis de RNA-seq. Los requerimientos de computación para llevar a cabo este tipo de análisis de RNA-seq y la importancia del número de réplicas biológicas son discutidos.

Los niveles de expresión génica en el hospedante se ven influenciados tanto por la genética de la planta como por la infección de la planta. El análisis de enriquecimiento funcional ha supuesto la mejora del conocimiento de los procesos biológicos implicados en los mecanismos de respuesta de la planta frente a la infección del patógeno en este patosistema.

Abstract (in English, 250 words or less):

Unravelling the infection and defence mechanisms in a plant pathosystem is key for development and implementation of a proper management strategy. Traditionally, research in plant pathology analysed infection and defence process separately, according to the aims of the study. This makes it difficult to elucidate the potential interactions between host/pathogen. Recently, High-Throughput Sequencing Technologies allowed to develop the RNA-seq analysis. Nevertheless, although its application has significantly increased in the current decade, the dual host/pathogen transcriptomic analyses are still limited.

The present Master's Thesis is focused on the dual host/pathogen transcriptomic analyses for the *Eucalyptus grandis/ Chrysoporthe austroafricana* pathosystem, testing the host genetic resistant and the effect of the pathogen infection. A pipeline and its script were developed for downloading and pre-processing of sequences, and subsequent RNA-seq analysis. Computational requirement and the importance of a proper number of biological replicates are discussed.

In host, gene expression is determined by host resistant genetic and infectious disease process. The functional enrichment analyses improved the knowledge on biological process involved in the response mechanisms in plant as a result of the pathogen infection for this pathosystem.

Índice

1. Introducción.....1
1.1.Contexto y justificación del Trabajo.....1
1.2.Objetivos del Trabajo……2
1.3.Enfoque y método seguido.....2
1.4.Planificación del Trabajo.....3
1.5.Breve sumario de productos obtenidos.....5
1.6.Breve descripción de los otros capítulos de la memoria.....5
2. Material y métodos 5
2.1.Descarga y preparación de los archivos originales de la secuenciación.....5
2.2.Control de calidad de la secuenciación.....8
2.3.Alineamiento de las secuencias.....9
2.4.Cuantificación.....13
2.5.Análisis de expresión diferencial.....14
2.6.Análisis de enriquecimiento funcional.....20
3. Resultados y discusión.....23
3.1.Control de calidad de la secuenciación.....23
3.2.Alineamiento de las secuencias.....27
3.3.Cuantificación.....28
3.4.Análisis de expresión diferencial.....28
3.5.Análisis de enriquecimiento funcional.....34
4. Conclusiones.....41
5. Glosario.....42
6. Bibliografía.....43
7. Anexos.....46

Lista de figuras

Figura 1: Evolución del número de trabajos que aplican la técnica de RNA-seq en la última década. Fuente: Elaboración propia (búsqueda en Web of Science de artículos que contengan la palabra RNA-seq en el título; Acceso el 15 Marzo 2018) **1**

Figura 2. Centro de descargas "ncbi" de los genomas de referencia y archivos de anotación **7**

Figura 3. Estadísticas del análisis de control de calidad con fastQC **8**

Figura 4. Captura del proceso de mapeado del hospedante mediante TopHat **10**

Figura 5. Visualización de la interface del software Blast2GO PRO **22**

Figura 6. Pipeline para llevar a cabo un análisis de enriquecimiento funcional. Fuente: Blast2GO **22**

Figura 7. Calidad de secuenciación por base de la secuencia SRR1951761 **23**

Figura 8. Estimador de calidad de la secuencia SRR1951761 **24**

Figura 9. Mapa de calor de la calidad de la secuencia SRR1951761 **24**

Figura 10. Proporción del contenido de cada nucleótido por base (localización) de la secuencia SRR1951761 **25**

Figura 11. Proporción del contenido de GC de la secuencia SRR1951761 **25**

Figura 12. Presencia de N de la secuencia SRR1951761 **26**

Figura 13. Niveles de duplicación de la secuencia SRR1951761 **26**

Figura 14. Contenido de adaptadores de la secuencia SRR1951761 **26**

Figura 15. Heatmap de la correlación de los conteos (interacción resistencia:inoculación) **29**

Figura 16. Gráfico del análisis de componentes principales (interacción resistencia:inoculación) **29**

Figura 17. Niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (interacción resistencia:inoculación) **30**

Figura 18. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (interacción resistencia:inoculación) **31**

Figura 19. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un mayor log2FoldChange (interacción resistencia:inoculación) **31**

Figura 20. Heatmap de la correlación de los conteos (efecto inoculación en clon resistente) **32**

Figura 21. Gráfico del análisis de componentes principales (efecto inoculación en clon resistente) **32**

Figura 22. Niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (efecto inoculación en clon resistente) **33**

Figura 23. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (efecto inoculación en clon resistente) **34**

Figura 24. Distribución de especies (interacción resistencia:inoculación) **34**

Figura 25. Procesos biológicos desencadenados en el hospedante (interacción resistencia:inoculación) **35**

Figura 26. WorldCloud de los procesos biológicos desencadenados en el hospedante (interacción resistencia:inoculación) **35**

Figura 27. Funciones moleculares desencadenados en el hospedante (interacción resistencia:inoculación) **36**

Figura 28. WorldCloud de las funciones moleculares desencadenados en el hospedante (interacción resistencia:inoculación) **36**

Figura 29. Componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (interacción resistencia:inoculación) **37**

Figura 30. WorldCloud de los componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (interacción resistencia:inoculación) **37**

Figura 31. Distribución de especies (efecto inoculación en clon resistente) **38**

Figura 32. Procesos biológicos desencadenados en el hospedante (efecto inoculación en clon resistente) **38**

Figura 33. WorldCloud de los procesos biológicos desencadenados en el hospedante (efecto inoculación en clon resistente) **38**

Figura 34. Funciones moleculares desencadenados en el hospedante (efecto inoculación en clon resistente) **39**

Figura 35. WorldCloud de las funciones moleculares desencadenados en el hospedante (efecto inoculación en clon resistente) **39**

Figura 36. Componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (efecto inoculación en clon resistente) **40**

Figura 37. WorldCloud de los componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (efecto inoculación en clon resistente) **40**

Lista de tablas

Tabla 1. Análisis de riesgos y plan de contingencia **4**

Tabla 2. Diseño experimental del estudio **6**

Tabla 3. Porcentajes de alineamiento frente al hospedante y patógeno para cada muestra **27**

1. Introducción

1.1. Contexto y justificación del Trabajo

El Trabajo Fin de Master (TFM) consistirá en el desarrollo y optimización de un pipeline para el análisis dual del transcriptoma patógeno/planta mediante la utilizando de software libre, el cual pueda ser implementado posteriormente en cualquier patosistema vegetal. Para ello, se utilizarán los datos disponibles en el repositorio "Gene Expression Omnibus (GEO)" (GSE67554; http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE67554) procedentes de un experimento en el cual se analizó el transcriptoma tanto del patógeno (*Chrysoporthe austroafricana)* como del hospedante (*Eucalyptus grandis*) cuando dos clones de distinta susceptibilidad de este último fueron inoculados con el hongo (Mangwanda et al. 2015).

El presente TFM se desarrolla en el área de la patología forestal y en la actualidad hay una notable falta de estudios que afronten el estudio de la interacción hospedante/patógeno de una forma dual. De hecho, la mayoría de trabajos disponibles estudian por separado tanto los mecanismos de infección del patógeno como los de defensa de la planta. Dificultando notablemente la comprensión de las interacciones entre ambos organismos. Recientemente, los nuevos métodos de secuenciación masiva han permitido desarrollar la técnica de RNA-seq desde finales de la pasada década. Sin embargo, aunque su uso ha crecido sustancialmente en la última década por su gran aplicabilidad (Figura 1), todavía son infrecuentes los estudios duales hospedante/patógeno del transcriptoma. De hecho la búsqueda de la combinación "RNA-seq" y "Dual" en Web of Science sólo devolvió 38 referencias (Acceso el 15 Marzo 2018), de las cuales únicamente 8 eran anteriores a 2016. El mayor limitante para este tipo de estudios es la baja proporción de RNA del patógeno frente al del hospedante (incluso inferior al 1%; Bradshaw et al., 2016) lo que implica la necesidad de una gran profundidad de secuenciación, con la consecuente limitación económica, y a su vez dificulta notablemente la capacidad de análisis.

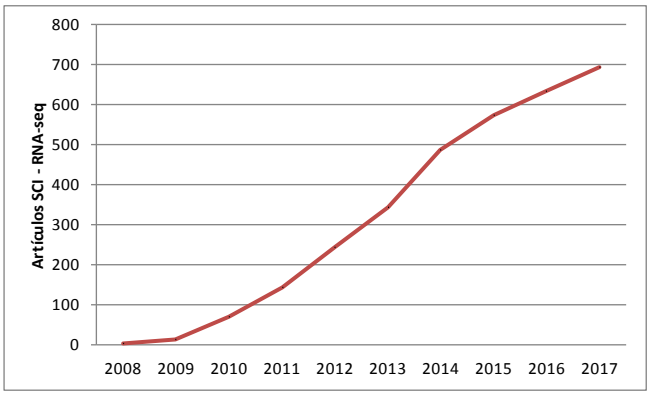


Figura 1: Evolución del número de trabajos que aplican la técnica de RNA-seq en la última década. Fuente: Elaboración propia (búsqueda en Web of Science de artículos que contengan la palabra RNA-seq en el título; Acceso 15 Marzo 2018)

1.2. Objetivos del Trabajo

1.2.1. Objetivos generales

El presente TFM tiene dos objetivos generales:

- ✓ Desarrollar el pipeline completo para el análisis dual hospedante/patógeno de datos de RNA-seq.
- ✓ Extraer conclusiones de nuestros resultados y desarrollar una discusión crítica de los mismos basada en la biobiografía disponible.

1.2.2. Objetivos específicos

Los objetivos específicos del TFM podrían ser encuadrados como logros personales y científicos. En cuanto a los primeros:

- ✓ Optimizar el uso de los recursos bibliográficos disponibles.
- ✓ Familiarizarse con el acceso y descarga de datos depositados en distintos repositorios de datos bioinformáticos.
- ✓ Adquirir los conocimientos necesarios para la utilización de software bioinformático en el análisis de datos de RNA-seq.
- ✓ Desarrollar la capacidad de iniciativa para afrontar de un modo autodidacta el aprendizaje y resolución de los problemas que pudieran surgir durante el desarrollo del trabajo.
- ✓ Mejorar la capacidad organizativa para alcanzar los objetivos planteados en los plazos propuestos en el cronograma.
- ✓ Desarrollar un pensamiento crítico que permita discutir los resultados propios con los de otros investigadores.

También son esperables los siguientes logros científicos:

- ✓ Determinar la profundidad de secuenciación para análisis de RNa-seq Dual.
- ✓ Obtener una serie de scripts que puedan ser implementados en el análisis de datos de RNA-seq dual en otros patosistemas.

1.3. Enfoque y método seguido

La secuenciación del transcriptoma de un patosistema vegetal supone un coste económico inabordable para el presente Trabajo Fin de Máster (TFM), por ello se optó por la utilización de datos de libre acceso en el repositorio GEO.

En cuanto al desarrollo del pipeline y el software a utilizar para el análisis bioinformático no se puede decir que exista una estrategia más apropiada que otra, pues para cada tarea existen diferentes programas bioinformáticos que son aceptados por la

comunidad científica. En este trabajo se ha optado por el desarrollo del pipeline utilizando software libre para que pueda ser utilizado posteriormente por cualquier investigador. Con la excepción de enriquecimiento funcional que, por falta de tiempo en la fase final del trabajo y para agilizar este proceso, se optó por usar una versión de prueba de BLAST2GO PRO.

Debido a las altas necesidades de computación fue inviable llevar a cabo los análisis con un ordenador personal y como alternativa se trabajó con el supercomputador del Campus Científico de la Universidad de Valladolid en escritorio remoto a través de la aplicación Apache Guacamole. Dicho supercomputador consta de 16 CPU con 16 GB de RAM.

1.4. Planificación del Trabajo

1.4.1. Tareas

El presente trabajo se desglosa en las siguientes tareas:

- Recopilación bibliográfica tanto de estudios que se hayan llevado a cabo análisis transcriptómico dual hospedante/patógeno, como de la información necesaria para la utilización del software a utilizar.
- Descarga y preparación de los archivos originales de la secuenciación.
- Elaboración de la PEC1
- Comprobación del control de calidad de la secuenciación.
- Pre-procesado de las secuencias (si fuera requerido)
- Mapeo de las secuencias
- Elaboración PEC2
- Cuantificación
- Normalización y análisis de expresión diferencial
- Análisis de enriquecimiento funcional
- Elaboración PEC3
- Redacción de la memoria final, la cual incluirá la presentación de los resultados obtenidos y la discusión de los mismos.
- Elaboración de la presentación
- Defensa pública del TFM

1.4.2. Calendario

La obtención de permisos de acceso al supercomputador se dilataron en el tiempo, lo que unido al hecho que el acceso no fuese como administrador sino como usuario (imposibilitando la instalación de software, lo cual debía ser solicitado al administrador) supuso un retrasó en los plazos propuestos en el cronograma inicial. El cual tuvo que ser convenientemente reajustado. Ambos cronogramas se adjuntan en Anexo I y II.

1.4.3. Hitos

En el presente TFM cada tarea en realidad es un hito en sí mismo, pues se obtendrá un nuevo archivo de datos con el que se trabajará en la siguiente tarea. Para facilitar la visualización del diagrama de Gantt, en el cronograma únicamente se han marcado como hitos los entregables de las PECs, la memoria y la defensa del TFM.

1.4.4. Análisis de riesgos

En el inicio del presente TFM se identificaron varios factores que podrían repercutir en el cumplimiento de las tareas, proponiéndose en cada caso un plan de contingencia (Tabla 1).

Tabla 1. Análisis de riesgos y plan de contingencia

Riesgo	Plan de contingencia
Alguno de los archivos fastq descargados desde el repositorio podría estar dañado	Buscar otro experimento similar que haya compartido sus datos de secuenciación en algún repositorio
Alguno de los software podría causar problemas en el análisis	Utilizar otro software alternativo que permita llevar a cabo ese proceso
Alguna tarea requiera más tiempo del planificado por falta de experiencia	La planificación temporal podría requerir ajustes pero es de esperar que haya tareas que se alarguen mientras otras ahorren tiempo
Podrían surgir retrasos por motivos personales o profesionales	Este riesgo es difícil de prever y solucionar. Se procurará adelantar las tareas por si en algún momento surge cualquier imprevisto.

Una vez desarrollado el trabajo, se comprobó que otros aspectos que no fueron considerados inicialmente como potenciales riesgos fueron claramente un hándicap para la consecución de los objetivos finales y por lo tanto deberían ser considerados en futuros trabajos. Entre ellos cabe destacar los requerimientos de computación necesarios para llevar a cabo este tipo de análisis y la falta de genomas de referencia y/o anotación (o pobremente anotados) para las especies objeto de análisis.

1.5. Breve sumario de productos obtenidos

- Una profunda recopilación bibliográfica de los software disponibles para llevar a cabo un análisis transcriptómico dual hospedante/patógeno mediante técnicas de RNA-seq.
- Desarrollo de un pipeline y su script completo para la descarga y preparación de archivos de libre acceso en los repositorios disponibles y su posterior análisis de RNA-seq.
- Identificación de problemática asociada a la ejecución de un análisis transcriptómico mediante RNA-seq, tales como capacidad de computación para desarrollar los procesos, requerimientos de almacenamiento, etc.

1.6. Breve descripción de los otros capítulos de la memoria

Desarrollada la introducción y los objetivos el presente TFM presenta la siguiente estructura:

- Capítulo 2 – Material y métodos. En este capítulo se presentará el pipeline desarrollado para llevar a cabo un análisis de RNAseq en el patosistema "*C. austroafricana/E. grandis*".
- Capítulo 3 – Resultados y discusión. Se presentarán los resultados obtenidos y serán objetos de discusión.
- Capítulo 4 – Conclusiones. Se plasmarán las principales conclusiones
- Capítulo 5 – Glosario.
- Capítulo 6 – Bibliografía.

2. Material y métodos

2.1. Descarga y preparación de los archivos originales de la secuenciación

Como se comentó anteriormente para el presente TFM se seleccionaron los datos disponibles en el repositorio "Gene Expression Omnibus (GEO)" (GSE67554; http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE67554) procedentes de un experimento en el cual se analizó el transcriptoma tanto del patógeno (*Chrysoporthe austroafricana)* como del hospedante (*Eucalyptus grandis*) cuando dos clones de distinta susceptibilidad fueron inoculados con el hongo (Mangwanda et al. 2015). En dicho trabajo se testaron dos clones de eucalipto (TAG5 y ZG14) con distinto grado de susceptibilidad al

patógeno, para ello se utilizaron 3 réplicas biológicas tanto para el tratamiento inoculado como el control. Así, en total se secuenciaron un total de 12 muestras (Tabla 2).

Tabla 2. Diseño experimental del estudio

SRA File	Clon	Tratamiento
SSR1951761	TAG5 – Moderadamente Resistente	Control
SSR1951762	TAG5 – Moderadamente Resistente	Control
SSR1951763	TAG5 – Moderadamente Resistente	Control
SSR1951764	TAG5 – Moderadamente Resistente	Inoculado
SSR1951765	TAG5 – Moderadamente Resistente	Inoculado
SSR1951766	TAG5 – Moderadamente Resistente	Inoculado
SSR1951767	ZG14 - Susceptible	Control
SSR1951768	ZG14 - Susceptible	Control
SSR1951769	ZG14 - Susceptible	Control
SSR1951770	ZG14 - Susceptible	Inoculado
SSR1951771	ZG14 - Susceptible	Inoculado
SSR1951772	ZG14 - Susceptible	Inoculado

A pesar de que en el artículo indica que se llevó a cabo una secuenciación paired-end de 50 bp, en el repositorio sólo se aporta un único archivo SRA por muestra. Lo cual a priori apuntaba a que los autores habrían procedido a juntar las dos reads (forward and reverse) de cada extremo en un único read.

Para la descarga de archivos muy pesados en formato ".sra" (~ 11-12 Gb) y su posterior conversión a formato fastq se utilizó "sratoolkit" (NCBI 2018). Una vez modificadas las variables de entorno (PATH) vinculándolo a sratoolkit, se procede a la descarga mediante la función "prefetch"

```
> E:\Usuarios\Jorge\Desktop\sratoolkit.2.9.0-win64\bin>prefetch –v SRR1951761
…..#idem para las otras 11 muestras
```

Una vez en el directorio donde se encuentran los ejecutables de "sratoolkit" se puede correr directamente la función con el identificador de la secuencia. Se incluye la opción -v para visualizar el progreso.

La conversión del archivo SRA a formato "fastq" se lleva a cabo mediante la función "fastq-dump"

```
> E:\Usuarios\Jorge\Desktop >fastq-dump --outdir secuencias -v SRR1951761
```

…..#idem para las otras 11 muestras

En la función fastq-dump se indica donde se quiere crear el nuevo archivo mediante –outdir y finalmente se indica el archivo SRA a convertir

La descarga del genoma de referencia y los archivos de anotación del patógeno y hospedante se llevó a cabo desde la base de datos del National Center for Biotechnology Information (ncbi) (Figura 2)

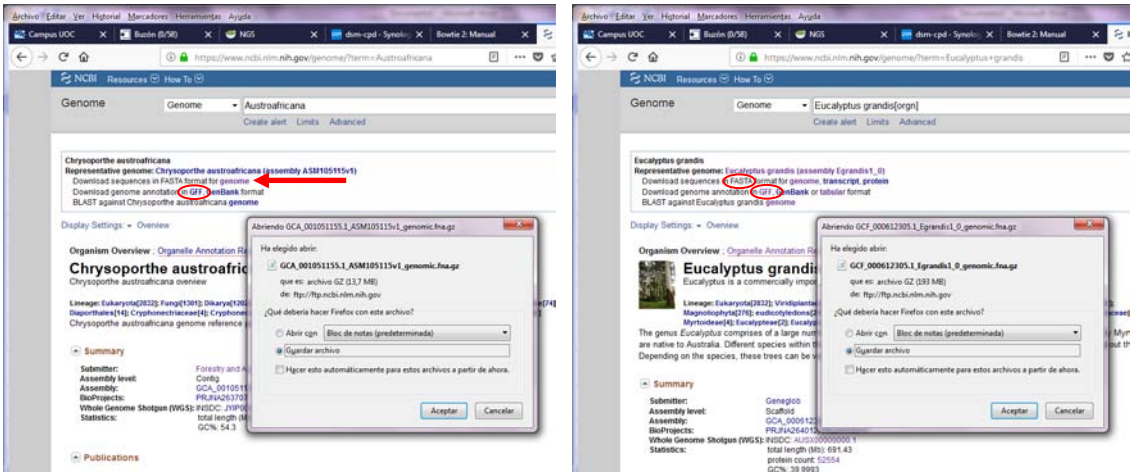


Figura 2. Centro de descargas "ncbi" de los genomas de referencia y archivos de anotación

Los genomas de referencia se encuentran en formato ".fna.gz" para convertirlos en formato ".fa" que son necesarios para posteriores análisis, se procedió a descomprimirlos mediante la función "gunzip" y a su transformación con la función "cp".

> gunzip GCA_001051155.1_ASM105115v1_genomic.fna.gz###Patogeno

> gunzip GCF_000612305.1_Egrandis1_0_genomic.fna.gz###Hospedante

> cp GCA_001051155.1_ASM105115v1_genomic.fna GCA_001051155.1_ASM105115v1_genomic.fa

> cp GCF_000612305.1_Egrandis1_0_genomic.fna GCF_000612305.1_Egrandis1_0_genomic.fa

En primer lugar hay que aportar el input en formato ".fna" y a continuación el output con el nombre que deseemos en formato ".fa"

Los archivos de anotación tuvieron una extensión ".gff.gz". Así, aunque inicialmente se procedió a descomprimir (gunzip) y transformar el archivo ".gff" a ".gtf" (mediante el software gffread (CCB 2018), se comprobó que el formato gff era adecuado para llevar a cabo el alineamiento.

2.2. Control de calidad de la secuenciación

Una vez descargadas las secuencias y convertidas a formato fastq se procede al control de calidad para garantizar que las secuencias cumplen unos requerimientos mínimos de calidad. El análisis se llevó a cabo con el software fastQC (Andrews 2010). Los parámetros que provee el software son:

- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✓ Per base sequence content
- ✓ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ✓ Overrepresented sequences
- ✓ Adapter Content

Como se comentó anteriormente, para cada muestra únicamente hay disponible un archivo SRA en el repositorio. Tras el análisis de control de calidad se pudo comprobar que las muestras tenían ~40 millones de reads de una longitud de 98 bp. Esto último confirma que las secuencias son fruto de la unión de los dos archivos paired-end (de 50 bp cada uno).

El hecho que las secuencias tengan 98 bp en vez de 100 bp y la alta calidad de las mismas (ver sección resultados), podría indicar que las secuencias subidas al repositorio habían sido pre-procesadas. Por ello, en el presente TFM no fue necesario llevar a cabo el trimming de las secuencias (para el cual se tenía previsto haber usado el software Trimmomatic (Bolger et al. 2014)).

Basic Statistics

Measure	Value
Filename	SRR1951761.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	39273762
Sequences flagged as poor quality	0
Sequence length	98
%GC	49

Figura 3. Estadísticas del análisis de control de calidad con fastQC

2.3. Alineamiento de las secuencias

El alineamiento consiste en el mapeo de las reads contra un genoma conocido. Si bien en un paso previo se debe crear un índice de los genomas de referencia, tanto del hospedante como del patógeno. Este proceso fue llevado a cabo con el software Bowtie2 (Langmead and Salzberg 2012)

```
> bowtie2-build GCA_001051155.1_ASM105115v1_genomic.fa CAustroafricanaPatogeno
> bowtie2-build GCF_000612305.1_Egrandis1_0_genomic.fa EgrandisHost
```

Con la función bowtie2-build se procede al indexado, para ello se aporta en primer lugar el input (genoma de referencia en formato .fa) y posteriormente el output (nombre del archivo de salida). Finalmente se generan 4 archivos '1.bt2', '2.bt2', '3.bt2', '4.bt2', 'rev.1.bt2' y 'rev.2.bt2'

Hay numerosos programas de alineamiento disponibles en la actualidad. Entre ellos, bowtie2 (Langmead and Salzberg 2012), STAR (Dobin et al. 2013) y TopHat2 (Kim et al. 2013) son algunos de los más utilizados por la comunidad científica. Para este TFM se desechó inicialmente STAR por requerir de gran capacidad de computación. Se optó por TopHat2 por tener la ventaja frente a bowtie2 de llevar a cabo el alineamiento en presencia de inserciones y delecciones. Además TopHat2 tiene la ventaja de ser capaz de alinear exones más cortos que el tamaño de las reads (Kim et al. 2013).

Teniendo en cuenta que los archivos fastq corresponden a la unión de ambos paired-end, se procedió al mapeado tratándola como si se tratara de un single-end.

```
> TopHat2 --no-coverage-search --segment-length 49 CAustroafricanaPatogeno SRR1951761.fastq
> TopHat2 --no-coverage-search --segment-length 49 EgrandisHost SRR1951761.fastq
```

Para el mapeo se establecieron un par de parámetros opcionales como "--no-coverage-search" porque generaba problemas en el paso "Searching for junctions". De hecho "coverage-search" está indicado sólo para reads <45 bp y secuencias con <=10 millones de reads (https://www.biostars.org/p/202093/; acceso 16/05/2018). De igual forma, se estableción "--segment-length 49" para reducir la longitud de los segmentos a la mitad, de este modo si el read completo es la unión del forward y reverse se forzaría a analizarlos por separados una vez que el alineamiento de la read completa no es posible (https://ccb.jhu.edu/software/tophat/manual.shtml). A continuación se aporta el archivo indexado (sin las extensiones .bt) y finalmente la secuencia a analizar en formato ".fastq!

A pesar de trabajar con el supercomputador del campus científico de la UVa, el alineamiento fue técnicamente inviable. De hecho, por ejemplo, el intento del mapeo de la primera de las secuencias contra el genoma del hospedante transcurrido día y medio no había finalizado el proceso (en la Figura 4 se puede apreciar la fecha y hora de la duración del proceso)

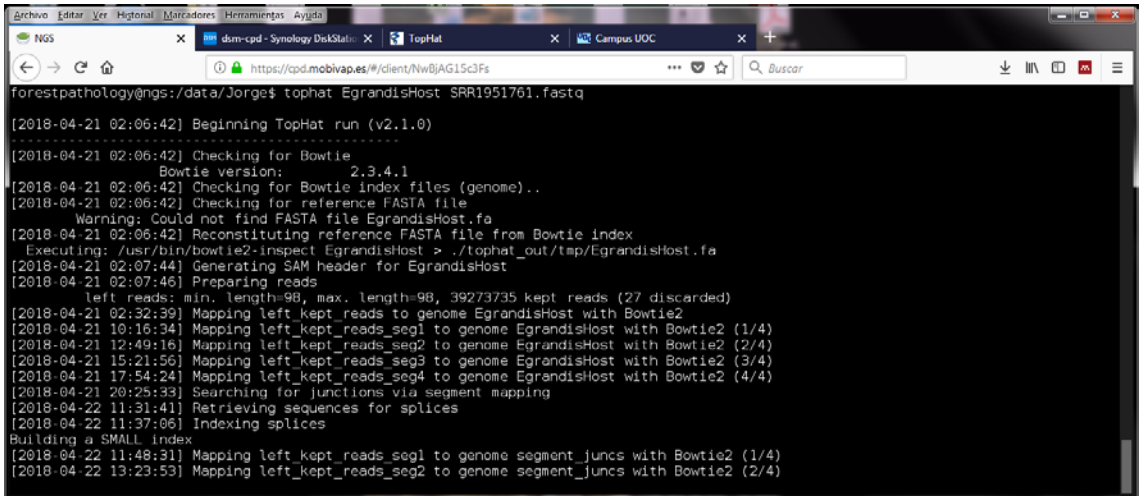


Figura 4. Captura del proceso de mapeado del hospedante mediante TopHat

Ante el gran tiempo de espera durante estos intentos infructuosos, se optó por utilizar un trozo de la secuencia de sólo 100.000 reads de una de las muestras hasta comprobar que el script de alineamiento era correcto. Para ello se usó la función "zcat"

> zcat SRR1951761.fastq.gz | head -400000 >file.fastq

Al utilizar la function "zcat" se puede trabajar con los archivos sin descomprimir, así se indica la secuencia objeto de extracción, seguido del caracter no alfanumérico "|" el comando head - seguido del número de líneas a extraer. En este caso se establecen 400.000 líneas porque cada read de un archivo fastq tiene 4 líneas. La primera corresponde al identificador de la secuencia (@ID), la segunda es la secuencia fasta, la tercera indica el final de la secuencia anterior (+ ó +SeqID) y la cuarta la calidad de la secuencia (ascii). Finalmente se indica el nombre del archivo de salida.

El resultado del mapeo usando este trozo de secuencia fue muy inferior al esperado (a tenor de los resultados presentados por Mangwanda et al., 2015). Al mapear contra el patógeno no se alineó ni una sola read. Sin embargo, teniendo en cuenta que en dicho artículo dan un valor del 0,5% para esa secuencia y que se han usado 100.000 reads, sería de esperar que entorno a 500 reads hubieran mapeado

Reads:
Input : 100000
Mapped : 0 (0.0% of input)
0.0% overall read mapping rate.

Si bien, aún más llamativo fue el resultado el mapeo contra el hospedante, para el cual se obtuvo entorno al 0.1% (menos de 100 reads), que dista mucho del 67% que asignaban a esa muestra en el artículo.

```
Reads:
        Input     :    100000
         Mapped   :       67 ( 0.1% of input)
          of these:       30 (44.8%) have multiple alignments (0 have >20)
0.1% overall read mapping rate.
```

Ante esta nueva situación, y tras comprobar que no había habido ningún error en la descarga de los genomas de referencia de los organismos (hospedante/patógeno), se planteó la hipótesis de que quizás el problema provenía de la unión de los dos archivos paired-end en uno (y que no se había solucionado con la inclusión del comando "--segment-length 49" en Tophat2). Durante el control de calidad también se pudo observar que el inicio de cada read y la posición 50-60 tenían menor calidad (correspondiendo en ambos casos con el inicio de secuenciación de cada una de las dos "left" y "right"; ver sección resultados). Por ello, se planteó que la unión de las dos secuencias podría ser la causante de los problemas de mapeo.

Así, se procedió a desensamblar las secuencias en las dos originales "paired-end". Para ello, se utilizó de nuevo la función "fastq-dump" de sratoolkit:

```
> fastq-dump --split-3 SRR1951761.sra

…..#idem para las otras 11 muestras
```

En este caso la función fastq-dump seguida de la opción "--split-3" sirve para dividir el archivo ".sra" objeto en dos archivos ".fastq". En este caso el resultado de usar "--split-3" es similar a usar ""--split-files" porque no se genera el tercer archivo de salida como resultado de tener una calidad asimétrica (https://www.biostars.org/p/156909/; acceso 12/05/2018)

El tamaño de los archivos resultantes ("file_1.fastq" y "file_2.fastq") fue de ~11Gb cada uno (~264 Gb para las 12 muestras). Por ello, la subida de los archivos al supercomputador era poco viable y se planteó la posibilidad de llevar a cabo la partición de cada archivo usando la función ">split –l [nºreads] file.fastq [output]" para posteriormente ensamblarlo mediante "Samtools merge". Sin embargo, después de varios intentos, se declinó esta opción porque se requeriría de varias semanas para concluir con el mapeado.

Ante este nuevo acontecimiento (y tras discutirlo con el supervisor), se optó por extraer una muestra de 4 millones de reads de cada secuencia y seguir el proceso con esta submuestra. Para proceder a la obtención de dicha submuestra de un modo aleatorio se optó por utilizar la función "seqtk sample" del software "seqtk" (seqtk, 2018), una vez instalado se procedió a correr la función "seqtk sample" estableciendo el mismo "random seed" para mantener pareadas las dos secuencias (left y right):

```
> seqtk sample –s10 SRR1951761_1.fastq 4000000 > SRR1951761_1_4M

> seqtk sample –s10 SRR1951761_2.fastq 4000000 > SRR1951761_2_4M
```

…..#idem para las otras 11 muestras

Las opciones seleccionadas fueron -s10 para establecer un mismo "random seed" y mantener pareadas las dos secuencias (left y right). En este caso el valor numérico corresponde directamente con el número de reads no de líneas.

Sin embargo, de nuevo no se disponía de capacidad computacional para que terminase de completar el proceso y después de muchas horas, se tuvo que abortar el trabajo. Ante este nuevo hándicap se optó por extraer 4 millones de reads localizados entre los 4-8 millones de cada secuencia (evitando extraer los primeros 4 millones pues la calidad de secuenciación inicial podría ser peor).

>zcat SRR1951761_1.fastq.gz | sed –n '16000001, 32000001p;32000002q' > SRR1951761_1_4M.fastq

>zcat SRR1951761_2.fastq.gz | sed –n '16000001, 32000001p;32000002q' > SRR1951762_1_4M.fastq

…..#idem para las otras 11 muestras

Mediante el comando "sed" se procede hacer el filtrado de las líneas entre 16000001 y 32000002. Se establece "-n" para que no lo imprima en pantalla, la letra "p" para imprimir el patrón y "q" para que salga sin procesar nada más. Aquí es fundamental empezar en 1 en vez de cero porque en caso contrario te asigna la fila del read anterior y cuando intentas llevar a cabo el mapeo te da errores porque el archivo ya no empieza con @.

Finalmente se procedió al mapeado contra el hospedante y contra en patógeno en TopHat2 :

>tophat2 --no-coverage-search CAustroafricanaPatogeno SRR1951761_1_4M.fastq SRR1951762_1_4M.fastq

>tophat2 --no-coverage-search EgrandisHost SRR1951761_1_4M.fastq SRR1951762_1_4M.fastq

...#idem para las otras 11 muestras

En este caso ya no se optó por incluir "--segment-length" por tener ahora secuencias de 49 bp. Debido al tiempo requerido para mapear todas las secuencias (tanto para el hospedante como para el patógeno), se recurrió a trabajar en multitarea usando la opción (>screen –r) de bash que te permite crear diferentes sesiones al mismo tiempo (<Ctrl-a><n>) e incluso desconectarte de la sesión (<Ctrl-a><d>) y que siga trabajando el supercomputador de forma remota, pudiendo recuperarla posteriormente (>screen –rx). El output obtenido es un archivo ".bam" que será posteriormente utilizado para la cuantificación.

2.4. Cuantificación

Una vez llevado a cabo el mapeo se procedió a la cuantificación. Para ello, se utilizó el paquete "Rsubread" (Liao et al. 2013) de Bioconductor todo ello implementado en "R" (R Development Core Team 2008). Para ello, se utilizó la función "featureCounts".

```
> library(Rsubread)
> CountsHost<-featureCounts(files=c("SRR1951761.bam", "SRR1951762.bam", "SRR1951763.bam", "SRR1951764.bam", "SRR1951765.bam", "SRR1951766.bam", "SRR1951767.bam", "SRR1951768.bam", "SRR1951769.bam", "SRR1951770.bam", "SRR1951771.bam", "SRR1951772.bam"), annot.ext= "GCF_000612305.1_Egrandis1_0_genomic.gff", isGTFAnnotationFile=TRUE, GTF.featureType= "gene", isPairedEnd=TRUE, nthreads=16)
> CountsPathogen<-featureCounts(files=c("SRR1951761.bam", "SRR1951762.bam", "SRR1951763.bam", "SRR1951764.bam", "SRR1951765.bam", "SRR1951766.bam", "SRR1951767.bam", "SRR1951768.bam", "SRR1951769.bam", "SRR1951770.bam", "SRR1951771.bam", "SRR1951772.bam"), annot.ext= "GCA_001051155.1_ASM105115v1_genomic.gff", isGTFAnnotationFile=TRUE, GTF.featureType= "gene", isPairedEnd=TRUE, nthreads=16)
```

La función featureCounts se alimenta con los archivos ".bam" obtenidos del mapeo, seguido del archivo de anotación en formato y varios parámetros opcionales. En este caso, se estableció el identificador "gene" (GTF.featureType= "gene") como codificador del archivo de anotación. En un primer momento se probó con "gene_id" por ser lo más frecuente, pero tras obtenerse errores se comprobó que la codificación era "gene". También se indicó que se trataba de un análisis paired-end ("isPairedEnd=TRUE"). Finalmente se utilizó la opción "nthreads" para pedir que se procesase el trabajo en paralelo usando las 16 cpu del supercomputador.

Aunque a priori se tenía pensado analizar la expresión génica tanto del hospedante como del patógeno. Al intentar llevar a cabo la cuantificación se obtenía un error para el patógeno. De hecho, al abrir el archivo de anotación ".gff" utilizando función "head" se observó que no había nada anotado salvo las longitudes de los scaffold. Dicho archivo de anotación había sido subido por los mismos autores del artículo (Mangwanda et al. 2015), pero en dicho artículo finalmente se centraron en el hospedante (probablemente por no disponer de la anotación del patógeno). Así, a partir de este punto el resto de análisis se centraron en el hospedante. Si bien, el pipeline desarrollado para el hospedante podría ser implementado de igual modo para el patógeno en aquellos patosistemas con genomas de referencia y anotaciones disponibles para ambos taxones.

Una vez llevado el conteo se procede a exportar la información de los conteos de cada una de las muestras en un archivo ".txt"

```
> write.table(CountsHost$counts,row.names=T,file= “CountHost.txt”)
```

El archivo generado anteriormente con “featureCounts” te genera información de los conteos y de la propia anotación que pueden ser extraídos mediante CountsHost$counts o CountsHost$annotation. En este caso, se procede a generar un archivo de los conteos en formato “.txt”.

2.5. Análisis de expresión diferencial

El análisis de expresión diferencial fue llevado a cabo para identificar aquellos genes que mostraron diferencias estadísticamente significativas. Varios software son utilizados para dicha tarea, entre ellos cabe destacar edgeR, NOISeq y DESeq (Robinson et al. 2010; Love et al. 2014; Tarazona et al. 2015). Para el presente TFM se optó por el paquete DESeq2 de Bioconductor por ser uno de los programas más consistentes (Costa-Silva et al. 2017).

El análisis de expresión diferencial puede ser llevado a cabo para el conjunto de los tratamientos o por el contrario para combinaciones individuales. En el presente TFM se presenta el pipeline tanto para la interacción de ambos tratamientos; resistencia (resistente o susceptible) e inoculación (control o inoculado), como para la comparación entre control vs. inoculado para el clon resistente.

2.5.1. Análisis de expresión diferencial: interacción resistencia:inoculación

Una vez cargada la librería, se procedió a crear la matriz de conteos (CountMatrix) y un dataframe en el que se codifican los tratamientos presentes en la matriz (coldata). Posteriormente, se creó un objeto DESeqDataSet mediante la función “DESeqDataSetFromMatrix” en el cual se asocian los conteos y el diseño experimental para el posterior análisis de expresión diferencial. Para dicho análisis fue utilizada la función “DESEq” (Love et al. 2018a), que lleva a cabo un análisis en tres pasos (estimación del tamaño de los factores, de la dispersión y el ajuste de una distribución binomial negativa GLM y test de Wald).

No se llevó a cabo pre-filtrado de genes con bajos conteos porque la función “results()” de DESeq2 lleva a cabo un filtrado independiente de forma automática. Por ello, dicho pre-filtrado no es necesario salvo para reducir el tamaño del objeto DESeqDataSet y agilizar los procesos de cálculo (Love et al. 2018b).

```
> library(DESeq2)

> CountMatrix<-read.table(“CountHost.txt”, header=T, row.names=1)

> coldata< data.frame(row.names=colnames(CountMatrix),resistance=rep(c(‘Resistant’,’Suscep
tible’),each=6), inoculation=rep(c(‘Control’,’Inoculated’),2,each=3))
```

```
> dds<-DESeqDataSetFromMatrix(countData=CountMatrix,colData=coldata,design= ~ resistance + inoculation + resistance:inoculation)
> dds<-DESeq(dds)
```

Al generar la matriz de conteos mediante la función "read.table" se especifica que el archivo "CountHost.txt" dispone de encabezados que corresponden con los identificadores de los archivos ".bam"y que la primera columna corresponde al identificador de los genes cuantificados. En este caso se establece un diseño para evaluar la interacción de los dos factores analizados en este trabajo (resistencia e inoculación), así se introduce en el modelo el efecto individual de ambos y su interacción.

A continuación una vez cargada la librería ComplexHeatmap (Gu et al. 2016) de Bioconductor se procedió a crear un mapa de calor (heatmap) para comprobar la similitud en expresión génica entre las distintas muestras. Para ello, se aplicó una correlación de Pearson de los conteos de expresión usando la función "cor()", previa transformación logarítmica en base 10.

```
> library(ComplexHeatmap)
> jpeg('Heatmap_correlation_Counts.jpg')
> Heatmap(cor(log10(counts(dds)+1)))
> dev.off()
```

Al trabajar en el supercomputador como terminal remoto, se tuvieron que exportar los mapas usando las funciones jpeg() para iniciar el proceso y dev.off() para concluirlo. Al llevar a cabo una transformación logarítmica en base 10 se sumo 1 a los conteos, para evitar el error de log10 (0). Así, para los genes con valores de 0, se establece un log(0+1)=0.

Del mismo modo, se generó un gráfico de análisis de componentes principales (PCA) mediante la función "plotPCA()" usando el objeto dds previa transformación logarítmica en base 2 (función "rlog()") para minimizar las diferencias entre muestras para filas con bajos conteos.

```
> dds_rlog<-rlog(dds)
> jpeg('PCA_Counts.jpg')
> plotPCA(dds_rlog,intgroup=c("resistance","inoculation"))
> dev.off()
```

En este caso al trabajar con la interacción el intgroup fue generada para ambos factores.

Para visualizar los resultados del análisis de expresión diferencial se utilizó la función "results()" que aporta log2 fold changes, errores estándar, test estadístico, p-valores y p-valores ajustados.

```
> alpha<-0.05
```

```
> resTotal<-results(dds,alpha=alpha)
> resTotal
```

Se estableció un umbral de significación de α = 0,05. Aunque este valor es comúnmente usado en la mayoría de test por la comunidad científica, en estudios de RNA-seq no hay un criterio fijado y no es infrecuente estudios menos restrictivos (α = 0,1) o más restrictivos (α = 0,01). Normalmente, el criterio se establece en función del número de genes identificados como diferencialmente expresados.

Posteriormente se comprueba cuántos genes están diferencialmente expresados, cuál son los más destacados y se genera un archivo con los resultados.

```
> resTotalOrdered<-resTotal[order(resTotal$padj),]
> resTotalOrderedCut<-subset(resTotalOrdered,padj<alpha)
> resTotalOrderedCut
> write.table(resTotalOrderedCut,file="resTotalOrderedCut.txt",sep='\t')
```

En primer lugar se procede a ordenar los genes (y sus conteos) en base al criterio de menor a mayor valor del p-valor ajustado. Posteriormente con la función subset se procede a extraer aquellos genes que cumplen el criterio de que el padj sea menor que el nivel de significación marcado anteriormente (α = 0,05). Finalmente, se genera un archivo con dichos genes usando de nuevo la función write.table.

Para la visualización de los genes diferencialmente expresados se genera un heatmap a partir de la tabla de los recuentos normalizados y previa transformación logarítmica en base 2.

```
> counts.norm <- counts(dds, normalized= T)
> log.norm.counts <- log2 (counts.norm +1)
> DEgenes<-rownames(resTotalOrderedCut)
> DEgenes.log.counts.norm<- log.norm.counts[DEgenes,]
> jpeg('DEgenes.log.counts.norm.jpg')
> Heatmap(DEgenes.log.counts.norm)
> dev.off()
```

Se establecen los conteos normalizados mediante la función counts de DESeq2 y se procede a su transformación log2(x+1). A continuación se extraen los identificadores de los genes diferencialmente expresados para extraer (mediante [ ]) únicamente los conteos que correspondan a esos genes.

Con el objetivo de una correcta visualización se procedió a generar un heatmap para los 10 genes con mayores diferencias de expresión génica. El criterio a utilizar para seleccionar estos genes puede estar basado en los p-valores ajustados o en los valores de

log2Fold-change. Cuando el criterio de selección fue el p-valor ajustado, simplemente se seleccionan los primeros 10 genes pues ya estaban previamente ordenados (de menor a mayor) por dicho criterio

```
> resTotalOrderedCut_padj_10<- resTotalOrderedCut[c(1:10),]
> DEgenes_padj_10<-rownames(resTotalOrderedCut_padj_10)
> DEgenes.log.counts.norm_ padj _10<- log.norm.counts[DEgenes_ padj _10,]
> jpeg('DEgenes.log.counts.norm_ padj _10.jpg')
> Heatmap(DEgenes.log.counts.norm_ padj _10)
> dev.off()
```

El proceso es similar al llevado a cabo con anterioridad, con la única diferencia que en este caso se selección (mediante [c(1:10),]) los 10 genes con menor p-valor ajustado.

FoldChange es el ratio de los valores de expresión génica de los tratamientos, por ello sus valores pueden ser positivos o negativos, indicando sobreregulación o infraregulación sobre el tratamiento de referencia respectivamente. Por eso en este caso hay que comprobar cual son los genes con mayores valores absolutos. Debido a que la selección se restringe únicamente a 10 genes, se han identificado fácilmente de forma visual usando las funciones "head()" y "tail()".

```
> resTotalOrderedCut_FoldChange<-resTotalOrderedCut[order(resTotalOrderedCut$log2Fold
Change),]
> head(resTotalOrderedCut_FoldChange,10)
> tail(resTotalOrderedCut_FoldChange,10)
> resTotalOrderedCut_FoldChange_10<- resTotalOrderedCut_FoldChange[c(1:6,217:220),]
> DEgenes_FoldChange_10<-rownames(resTotalOrderedCut_FoldChange_10)
> DEgenes.log.counts.norm_FoldChange_10<- log.norm.counts[DEgenes_FoldChange_10,]
> jpeg('DEgenes.log.counts.norm_FoldChange_10.jpg')
> Heatmap(DEgenes.log.counts.norm_FoldChange_10)
> dev.off()
```

De nuevo el proceso es similar al llevado a cabo con anterioridad, salvo que en esta ocasión para marcar el nuevo criterio de selección se sustituye $padj por $FoldChange.

Para una visualización individual de los niveles de expresión de los genes se utiliza la función "plotCounts()". En concreto se visualizaron los plots de los 10 genes con mayor expresión diferencial bajo el criterio del p-valor ajustado

```
> jpeg('PlotCounts_Total.jpg',height=1000, width=1000)
```

```
> par(mfrow=c(5,2))
> plotCounts(dds,gene="LOC104448775",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104450466",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104454002",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104417226",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104422076",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104449669",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104453136",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104414164",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104418735",intgroup=c("resistance","inoculation")
> plotCounts(dds,gene="LOC104452445",intgroup=c("resistance","inoculation")
> dev.off()
```

Con la función par(mfrow=c(5,2)) se indica que se genere un único gráfico con 10 ventanas en 5 filas y 2 columnas. Posteriormente se utiliza la función "plotCounts" para representar gráficamente los conteos de los genes diana en base a todas las combinaciones de los 2 factores analizados.

2.5.2. Análisis de expresión diferencial: efecto de la inoculación en el clon resistente

El pipeline es similar al caso anterior modificando la matriz de conteos y el diseño experimental. En concreto se seleccionaron únicamente las primeras seis columnas de la matriz completa por ser las correspondientes al clon resistente. También se modificó el dataframe del diseño muestral, para indicar que las tres primeras secuencias corresponden al control y las 3 restantes al tratamiento inoculado. Finalmente el modelo se modificó para analizar únicamente el factor inoculación.

Nota: los scripts de este apartado no son explicados por ser coincidentes con el apartado 2.5.1, cambiando únicamente la codificación de los inputs y outputs.

```
> CountMatrixresistance<-CountMatrix[,1:6]
> coldataresistance<-data.frame(row.names=colnames(CountMatrixresistance),
inoculation=rep(c('Control','Inoculated') ,each=3))
> ddsresistance<-
DESeqDataSetFromMatrix(countData=CountMatrixresistance,colData=coldataresistance,design
=~ inoculation)
> ddsresistance<-DESeq(ddsresistance)
```

```
> jpeg('Heatmap_correlation_Counts_resistance.jpg')
> Heatmap(cor(log10(counts(ddsresistance)+1)))
> dev.off()
> dds_rlog_resistance<-rlog(ddsresistance)
> jpeg('PCA_Counts_resistance.jpg')
> plotPCA(dds_rlog_resistance,intgroup="inoculation")
> dev.off()
> alpha<-0.05
> resResistance<-results(ddsresistance,alpha=alpha)
> resResistance
> resResistanceOrdered<-resResistance[order(resResistance$padj),]
> resResistanceOrderedCut<-subset(resResistanceOrdered,padj<alpha)
> resResistanceOrderedCut
> write.table(resResistanceOrderedCut,file="resResistanceOrderedCut.txt",sep='\t')
> counts.norm_resistance<- counts(ddsresistance, normalized= T)
> log.norm.counts_resistance <- log2 (counts.norm_resistance +1)
> DEgenes_Resistance<-rownames(resResistanceOrderedCut)
> DEgenes.log.counts.norm_resistance <- log.norm.counts_resistance[DEgenes,]
> jpeg('DEgenes.log.counts.norm.jpg')
> Heatmap(DEgenes.log.counts.norm)
> dev.off()
> resResistanceOrderedCut_padj_10<- resResistanceOrderedCut[c(1:10),]
> DEgenes_Resistance_padj_10<-rownames(resResistanceOrderedCut_padj_10)
> DEgenes.log.counts.norm_ padj _10_resistance<-
log.norm.counts_resistance[DEgenes_Resistance_ padj _10,]
> jpeg('DEgenes.log.counts.norm_ padj _10_resistance.jpg')
> Heatmap(DEgenes.log.counts.norm_ padj _10)
> dev.off()

> jpeg('PlotCounts_Resistance.jpg',height=1000, width=1000)
> par(mfrow=c(5,2))
> plotCounts(dds,gene="LOC104422425",intgroup="inoculation")
> plotCounts(dds,gene="LOC104447317",intgroup="inoculation)
```

```
> plotCounts(dds,gene="LOC104421112",intgroup="inoculation")
> plotCounts(dds,gene="LOC104421444",intgroup="inoculation")
> plotCounts(dds,gene="LOC104415812",intgroup="inoculation")
> plotCounts(dds,gene="LOC104440756",intgroup="inoculation")
> plotCounts(dds,gene="LOC104439413",intgroup="inoculation")
> plotCounts(dds,gene="LOC104421134",intgroup="inoculation")
> plotCounts(dds,gene="LOC104443948",intgroup="inoculation")
> plotCounts(dds,gene="LOC104449755",intgroup="inoculation")
> dev.off()
```

2.6. Análisis de enriquecimiento funcional

El análisis funcional del conjunto de datos de expresión diferencial se llevó a cabo con el software BLAST2GO (Götz et al. 2008). Como se comentó anteriormente todo el pipeline fue desarrollado en software libre, salvo este apartado que por agilizar el proceso se recurrió a la versión de prueba de BLAST2GO PRO.

El análisis funcional fue llevado a cabo con los 10 genes más diferencialmente expresados (bajo el criterio padj) para agilizar el proceso. En primer lugar se generó un archivo ".fasta" con las secuencias de todos los genes del archivo de anotación. Para ello, se utilizó el software gffread (CCB 2018) alimentándolo tanto con el genoma de referencia como del archivo de anotación

```
> gffread -w genes.fa -g GCF_000612305.1_Egrandis1_0_genomic.fa GCF_000612305.1_Egrandis1_0_genomic.gff
```

Para extraer las secuencias en un formato fasta se utiliza la función gffread seguida de "-w" para indicar el nombre de la salida, de "-g" para indicar el genoma de referencia y finalmente incluyendo el archivo de anotaciones .

Con la finalidad de manipular y extraer las secuencias de una forma eficiente se procedió a la lectura del archivo ".fasta" con las secuencias de los genes mediante la función "readDNAStringSet()" del paquete "Biostrings" (Pagès et al. 2018) de Bioconductor.

```
> library(Biostrings)
> library(Biostrings)
> Genes<-readDNAStringSet("genes.fa",use.names=T)
> sequence<-paste(Genes)
```

```
> seq_name<-names(Genes)
> Genes_df<-data.frame(sequence, seq_name)
```

La obtención del dataframe con dos columnas (nombre y secuncia) se basó en un script publicado en el siguiente post https://stackoverflow.com/questions/21263636/how-to-read-fasta-into-dataframe-and-extract-subsequences-of-fasta-file-in-r (acceso 18/05/2018)

A partir de dicho dataframe se procedió a extraer la secuencia de cada uno de los 10 genes con p-valores más bajos en el análisis de expresión diferencial.

```
> LOC104448775_seq<- Genes_df[grep("LOC104448775", Genes_df$seq_name),2]
> LOC104448775_label<-c("LOC104448775")
> write.table(LOC104448775_seq, LOC104448775_label,file=" LOC104448775.fasta")
…..###Idem para el resto de genes
```

La extracción de la secuencia se llevo a cabo usando la función "grep()", posteriormente se creó una etiqueta con el identificador de la secuencia y finalmente se exportó la secuencia en formato fasta

A partir de todos los archivos fasta individuales se genera un archivo multi-fasta concatenando todos ellos mediante la función de Linux "cat". Esto se hizo para las dos comparaciones llevadas a cabo (resistancia:inoculación e inoculación para el clon resistente).

```
> cat LOC104448775.fasta LOC104450466.fasta LOC104454002.fasta LOC104417226.fasta LOC104422076.fasta LOC104449669.fasta LOC104453136.fasta LOC104414164.fasta LOC104418735.fasta LOC104452445.fasta > Total_pvalue.fasta

> cat LOC104422425.fasta LOC104447317.fasta LOC104421112.fasta LOC104421444.fasta LOC104415812.fasta LOC104440756.fasta LOC104439413.fasta LOC104421134.fasta LOC104443948.fasta LOC104449755.fasta > Resistance_pvalue.fasta
```

Una vez se dispuso del archivo multifasta con las secuencias de los 10 genes se procedió a cargarlas directamente en BLAST2GO (File **Load > Load Sequences > Load Fasta File)** (Figura 5)**.**

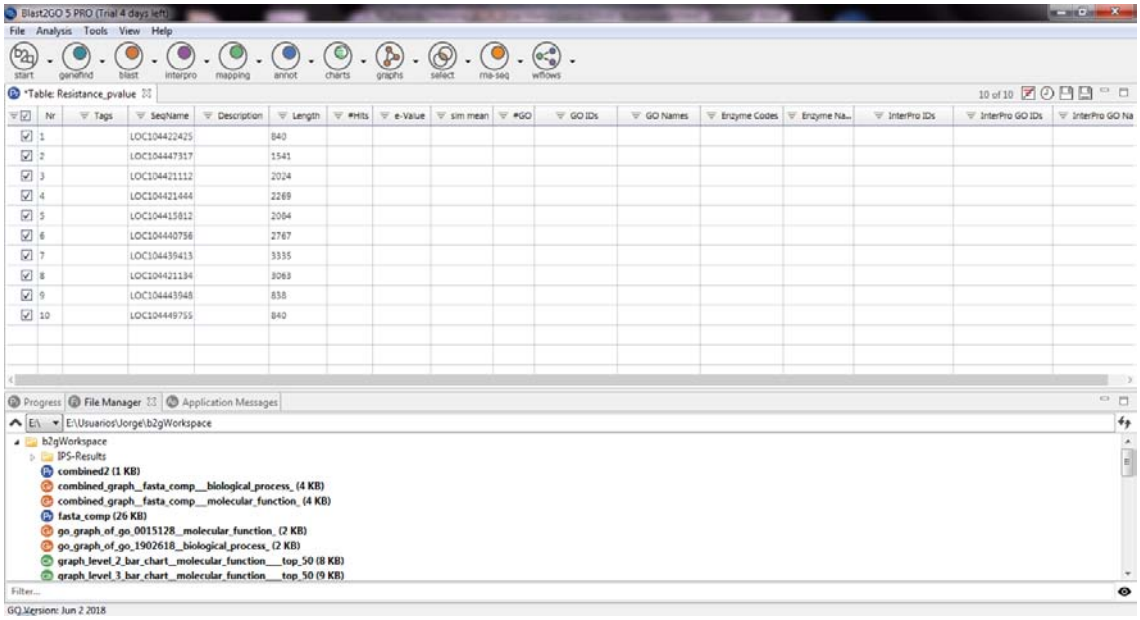


Figura 5. Visualización de la interface del software Blast2GO PRO

A continuación se procedió a llevar a cabo la "Gene ontology annotation" siguiendo el pipeline de la Figura 6.

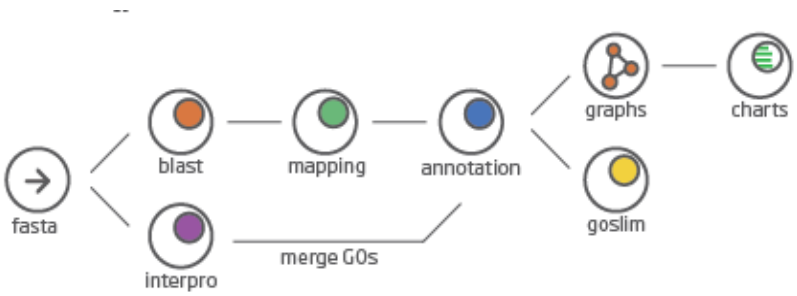


Figura 6. Pipeline para llevar a cabo un análisis de enriquecimiento funcional. Fuente: Blast2GO

Así en primer lugar se llevó a cabo el Blast, en concreto "blastx" para identificar secuencias potencialmente homólogas, comparando la secuencia de nucleótidos traducida para todos los marcos de lectura frente a una base de datos de secuencias de proteínas (http://docs.blast2go.com/user-manual/blast/). En paralelo a Blast se corrió InterProScan para mejorar la calidad de la anotación (http://docs.blast2go.com/user-manual/interproscan-annotation/#InterProScanAnnotation-RunInterProScan). Posteriormente, se llevó cabo el "mapping gene ontology" para machear todos los IDs de las secuencias obtenidas durante el "blast" contra la base de datos "Gene Ontology Annotation". Una vez concluido el mapeo se lleva a cabo el proceso de anotación, para ello evalúa la información obtenida durante el "blast" y "mapping" y asigna los Gene ontology (GO) terms más fiables, teniendo en cuenta la jerarquía GO, la similaridad de las secuencias, así como la abundancia y calidad de la fuente de anotación. Una vez llegado a este punto para mejorar la calidad de la anotación se puede unir la InterProScan Gos a la

anotación (Merge InterProScan GOs to Annotation). Llegado a este punto en la opción "Graphs" se visualizan los resultados de forma conjunta utilizando tres categorías de GO (Procesos biológicos, funciones moleculares y componentes celulares) en distintos formatos (barras, circular, WorldCloud, etc) . En la opción "Chart" se analizó la distribución de especies asociadas a los resultados obtenidos. Finalmente, se identificaron los pathway-maps desde KEGG.

3. Resultados y discusión

3.1. Control de calidad de la secuenciación

Los resultados del control de calidad se presentan y discuten únicamente para la primera de las secuencias, pues todas ellas mostraron resultados similares. El análisis de la calidad de la secuencia muestra el rango de valores de la calidad de cada base en toda la longitud de la secuencia. En este tipo de gráficos es muy común que el inicio y las colas de la secuencia presenten peor calidad de secuenciación. En concreto en las muestras analizadas se observaba claramente como estaban las dos secuencias paired-end de 49 bp unidas. Si bien, la calidad de todas ellas fue muy alta por encima de un valor de 30, incluso en el inicio y final de las mismas (Figura 7). En este caso concreto la calidad media de la secuencia fue 39 (Figura 8), lo cual se considera una calidad excelente. Como umbral mínimo exigido se establece un valor de 20 para el cuartil inferior (barra de error) de todas las bases.

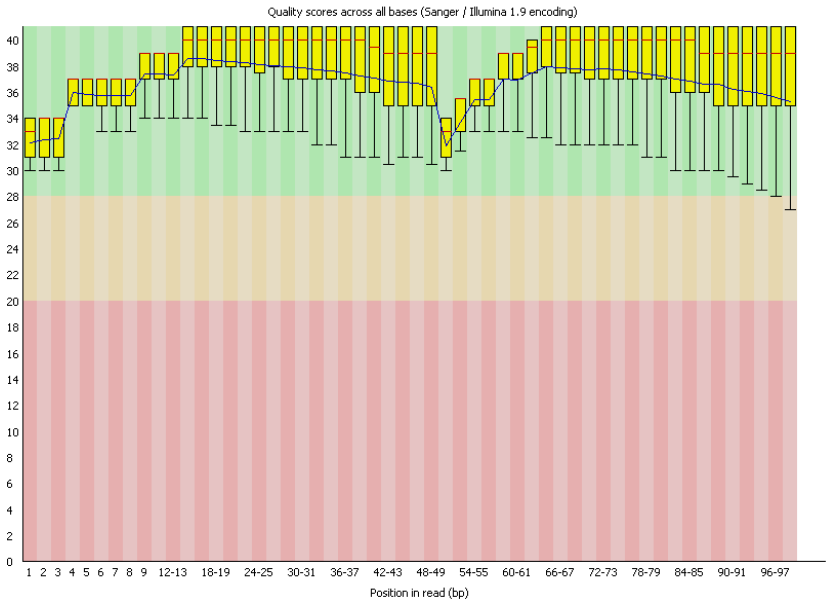


Figura 7. Calidad de secuenciación por base de la secuencia SRR1951761

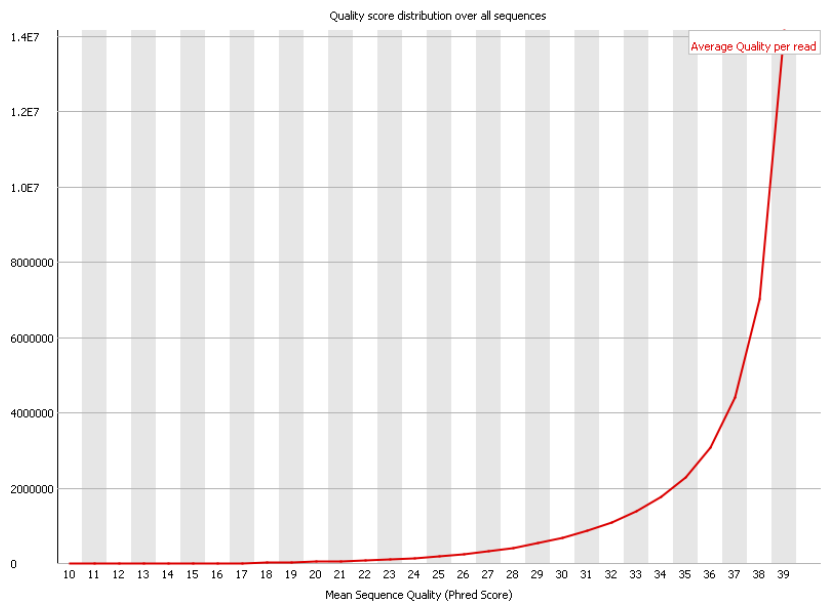


Figura 8. Estimador de calidad de la secuencia SRR1951761

De igual forma en el gráfico de escala de color no se apreciaron posiciones en color rojo que vendría a indicar una peor calidad en comparación con otras para la misma base (Figura 9).

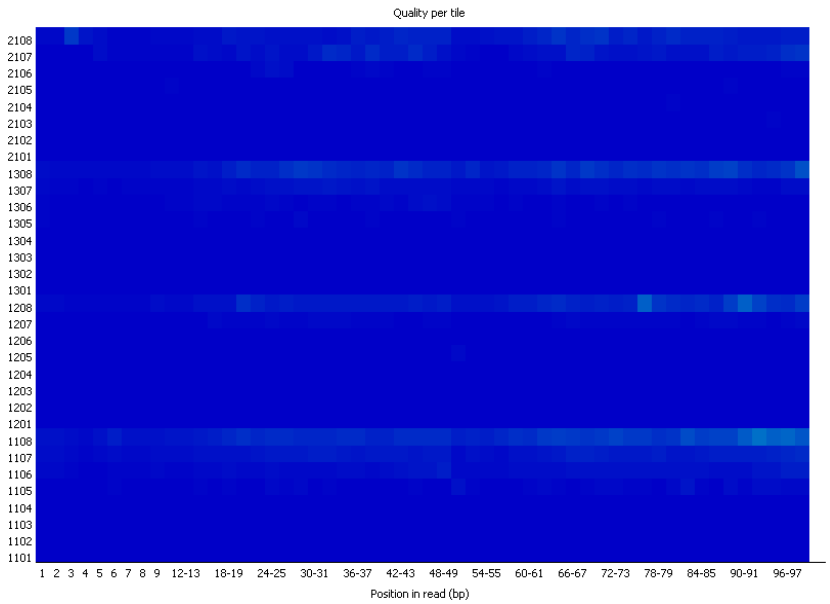


Figura 9. Mapa de calor de la calidad de la secuencia SRR1951761

El análisis de contenido de los distintos nucleótidos en cada posición del read muestra como es muy similar salvo en los primeros nucleótidos (algo normal por tener peor calidad de secuenciación) y en la posición 50-60 (también lógico teniendo en cuenta que se corresponde con la unión/inicio del segundo read del paired-end) (Figura

10). Este resultado es normal para un experimento de RNA-seq y el aviso que marca el software es debido a que está diseñado para experimentos de DNA.

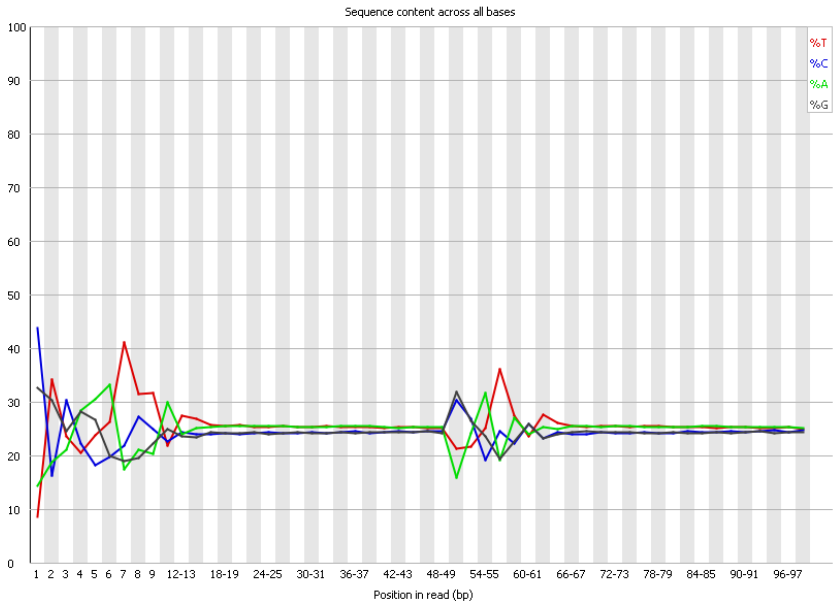


Figura 10. Proporción del contenido de cada nucleótido por base (localización) de la secuencia SRR1951761

La proporción del contenido de GC también se aproxima a la distribución esperada (teórica) (Figura 11). No se encontró presencia de N en las secuencias, lo cual de nuevo indica la alta calidad de las mismas (Figura 12). De igual forma se comprobó que no hubo problemas con los niveles de duplicación (Figura 13) y que no presentaban adaptadores (Figura 14)

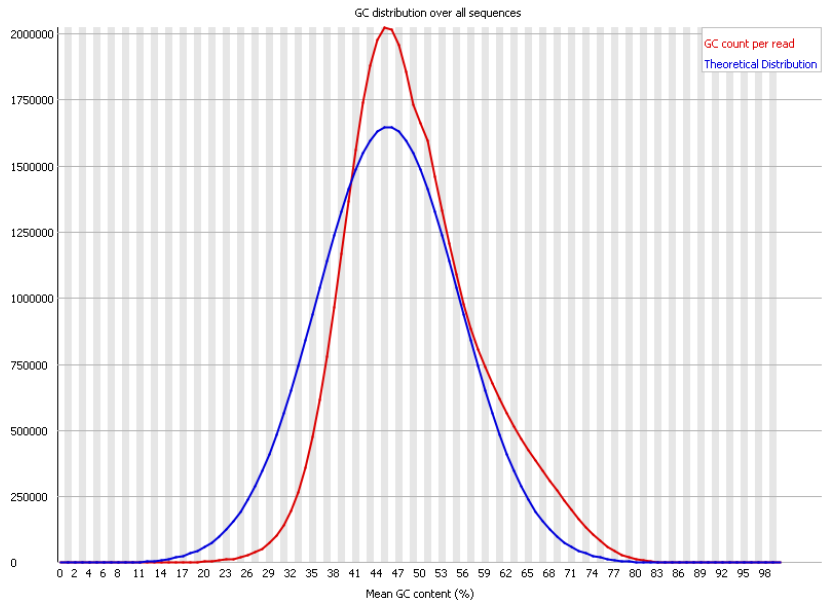


Figura 11. Proporción del contenido de GC de la secuencia SRR1951761

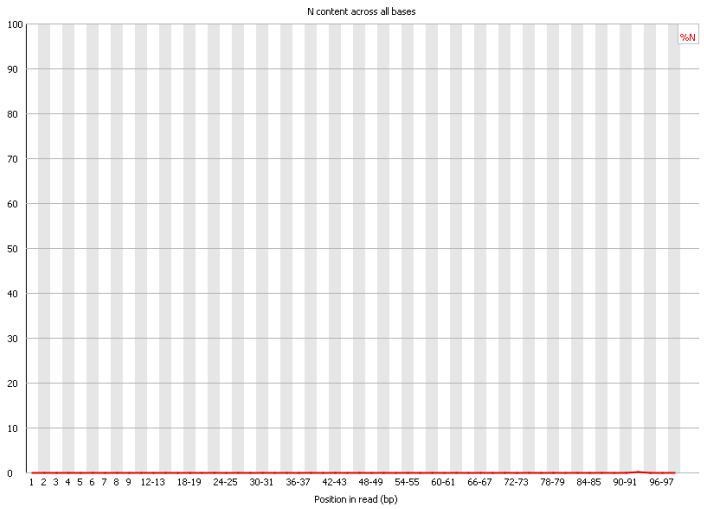


Figura 12. Presencia de N de la secuencia SRR1951761

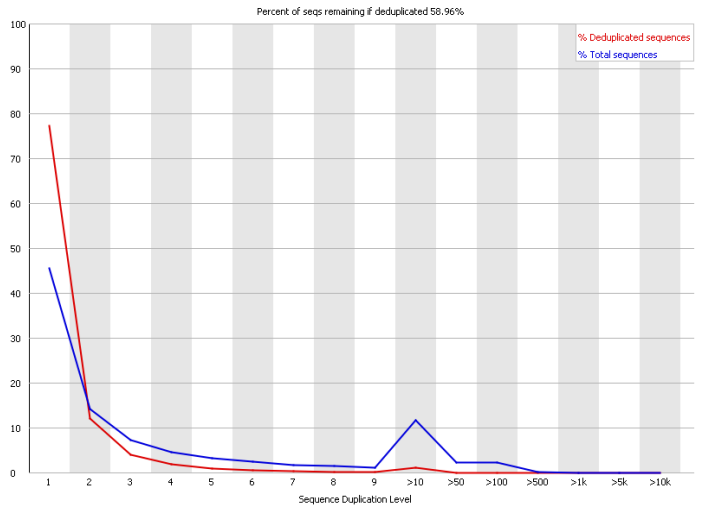


Figura 13. Niveles de duplicación de la secuencia SRR1951761

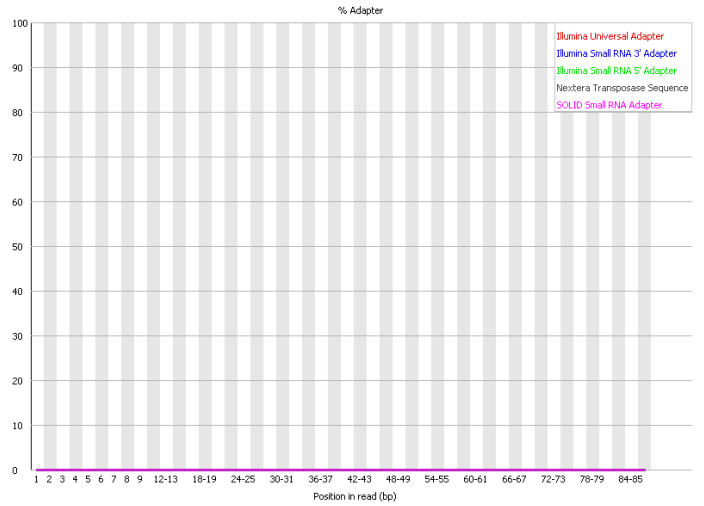


Figura 14. Contenido de adaptadores de la secuencia SRR1951761

Como se apuntó en la metodología, no se procedió a llevar a cabo un pre-procesado de las muestras porque esta principalmente indicado para mejorar la calidad de umbrales intermedios (Q entre 20 y 30) (Del Fabbro et al. 2013). La excelente calidad de las secuencias podría ser debido a que antes de unir el forward y reverse en un único archivo fueran sometidos a un pre-procesado y por ello cada read tenga 98 bp en vez de 100 bp, si como se expone en el artículo se llevo a cabo una secuenciación paired-end de 50 bp.

3.2. Alineamiento de las secuencias

Los resultados del mapeo mediante TopHat2 son presentados en la tabla 3.

Tabla 3. Porcentajes de alineamiento frente al hospedante y patógeno para cada muestra

Muestra*	SRA File	Total PE reads	% Reads mapeadas frente al genoma del hospedante (*E. grandis*)	% Reads mapeadas frente al genoma del patógeno (C. *austroafricana*)
TAG5 Control BR1	SSR1951761	4.000.000	88,1 %	0,1 %
TAG5 Control BR2	SSR1951762	4.000.000	88,1 %	0,8 %
TAG5 Control BR3	SSR1951763	4.000.000	88,1 %	0,1 %
TAG5 Infected BR1	SSR1951764	4.000.000	88,7 %	1.0 %
TAG5 Infected BR2	SSR1951765	4.000.000	87,7 %	1.5 %
TAG5 Infected BR3	SSR1951766	4.000.000	87,2 %	0,9 %
ZG14 Control BR1	SSR1951767	4.000.000	55,1 %	0,1 %
ZG14 Control BR2	SSR1951768	4.000.000	86,8 %	0,1 %
ZG14 Control BR3	SSR1951769	4.000.000	88,4 %	0,1 %
ZG14 Infected BR1	SSR1951770	4.000.000	88,1 %	1.4 %
ZG14 Infected BR2	SSR1951771	4.000.000	88,2 %	1.4 %
ZG14 Infected BR3	SSR1951772	4.000.000	87,9 %	1.5 %

* La codificación de las muestras es la original en el artículo (Mangwanda et al. 2015)

Los resultados muestran que la gran mayoría de las reads (~90%) mapearon contra el genoma del hospedante. Estos resultados a pesar de ser de una submuestra de 4 millones de reads se asemejan bastante a los resultados presentados en Mangwanda et

al., (2015) (material suplementario S1). Salvo para la secuencia SRR1951767 que mostró un resultado anómalo al mapear contra el hospedante. Por el contrario, el porcentaje de reads mapeadas contra el genoma del hospedante fue muy inferior (~1-1.5% en las muestras inoculadas). Este valor también es similar al obtenido en Mangwanda et al., (2015) y en otros trabajos de RNAseq dual (Bradshaw et al. 2016). Lo cual es debido a la baja proporción de biomasa de los patógenos respecto al hospedante, la cual en otros patosistema forestal (*Pinus radiata*/*Fusarium circinatum*) se ha establecido en niveles entre 0,1-1% (Martín-Rodrigues et al. 2013).

3.3. Cuantificación

Entorno al 77% de los genes de *E. grandis* fueron identificados en el conjunto de secuencias analizadas. Aunque Mangwanda et al., (2015) no aporta información sobre la profundidad de secuenciación, dicho porcentaje y el hecho de que no todos los genes expresados son detectados (García-Ortega and Martínez 2015) parece indicar que dicha profundidad de secuenciación fue suficiente.

3.4. Análisis de expresión diferencial

3.4.1. Interacción resistencia:inoculación

Un total de 220 genes fueron diferencialmente expresados, de los cuales 92 fueron infraregulados y 128 sobreregulados. En el heatmap de la correlación de los conteos de cada muestra se observa que las mayores diferencias se dan entre clones (SRR1951761-66 vs SRR1951767-72) más que por el efecto de la inoculación (color azul intenso) (Figura 15). El análisis de componentes principales explicó en sus dos primeros ejes un 86% de la varianza, mostrando claramente que las diferencias entre clones se encuentran en el eje X, mientras que el efecto inoculación queda recogido en el eje Y (Figura 16). Mientras que en el clon susceptible la diferencia entre inoculado y control se produce en todas las muestras, en el caso del clon resistente hay dos muestras (inoculada y control) que parecen comportarse como outlier. Estos resultados se basan en una muestra aleatoria de 4 millones de reads y por lo tanto deberían ser tomados con cautela, pero todo parece indicar que quizás una cuarta réplica biológica para el clon resistente hubiera sido recomendable. De hecho, un mínimo de 6 réplicas biológicas para estudios de RNAseq ha sido sugerido (Schurch et al. 2016). En esta línea, un incremento del número de réplicas parece ser una mejor estrategia que un incremento de la profundidad de secuenciación (más allá de 10 M reads) (Liu et al. 2014).

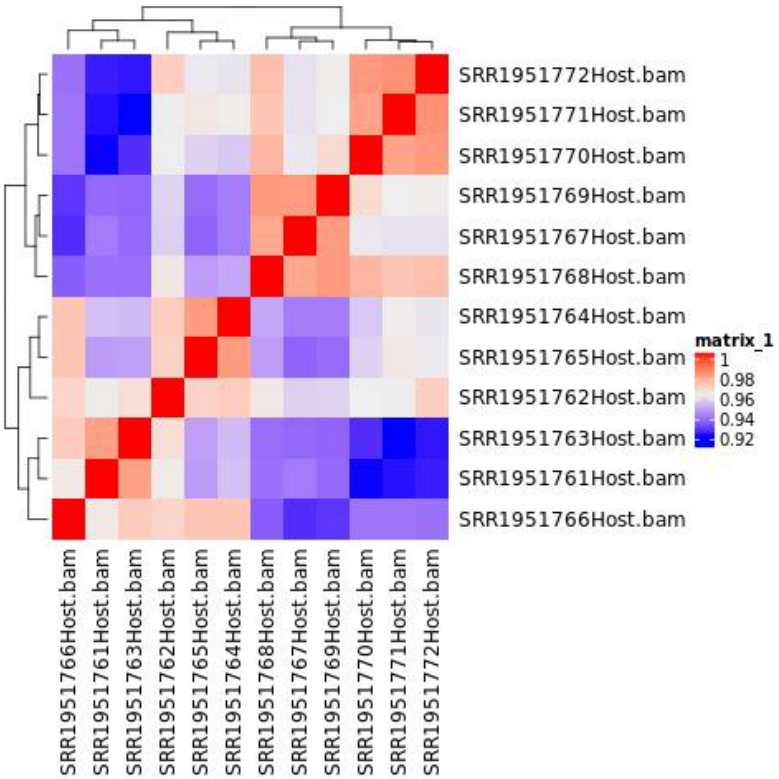


Figura 15. Heatmap de la correlación de los conteos (interacción resistencia:inoculación)

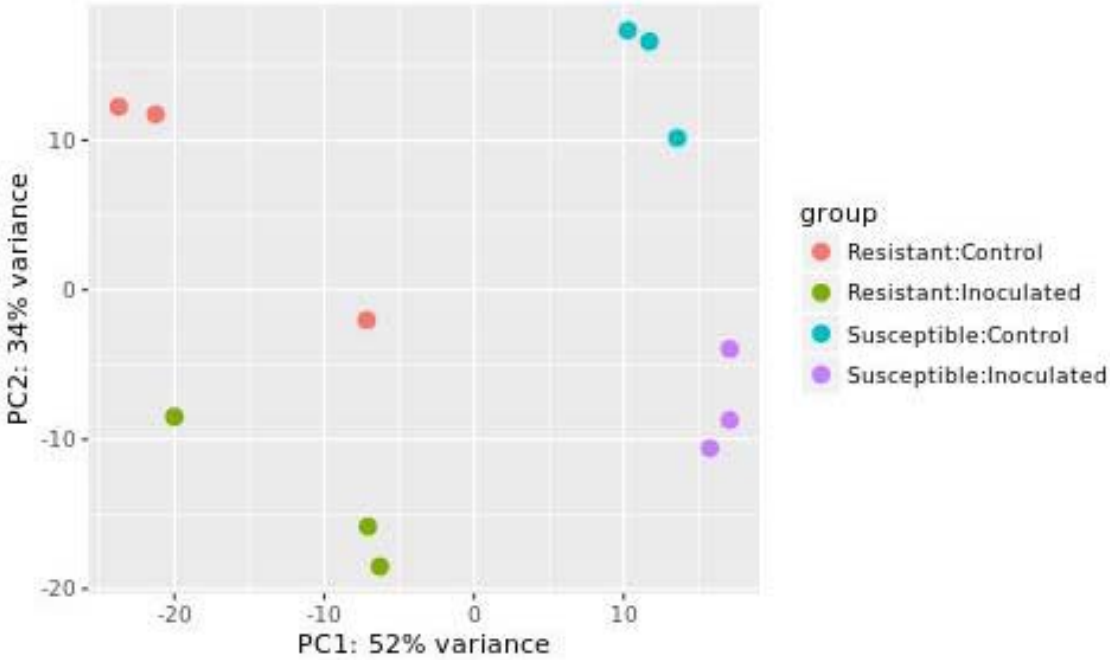


Figura 16. Gráfico del análisis de componentes principales (interacción resistencia:inoculación)

De los 10 genes más diferencialmente expresados (bajo el criterio de menor p-valor ajustado), 8 de ellos(LOC104448775, LOC104422076, LOC104453136, LOC104418735, LOC104450466, LOC104417226, LOC104414164 y LOC104452445) respondieron al efecto del clon exclusivamente (Control:Resistente vs Control:Susceptible). A su vez 2 genes (LOC104454002 y LOC104449669) apuntan a la distinta reacción de los dos clones tras la infección del patógeno (Figura 17).

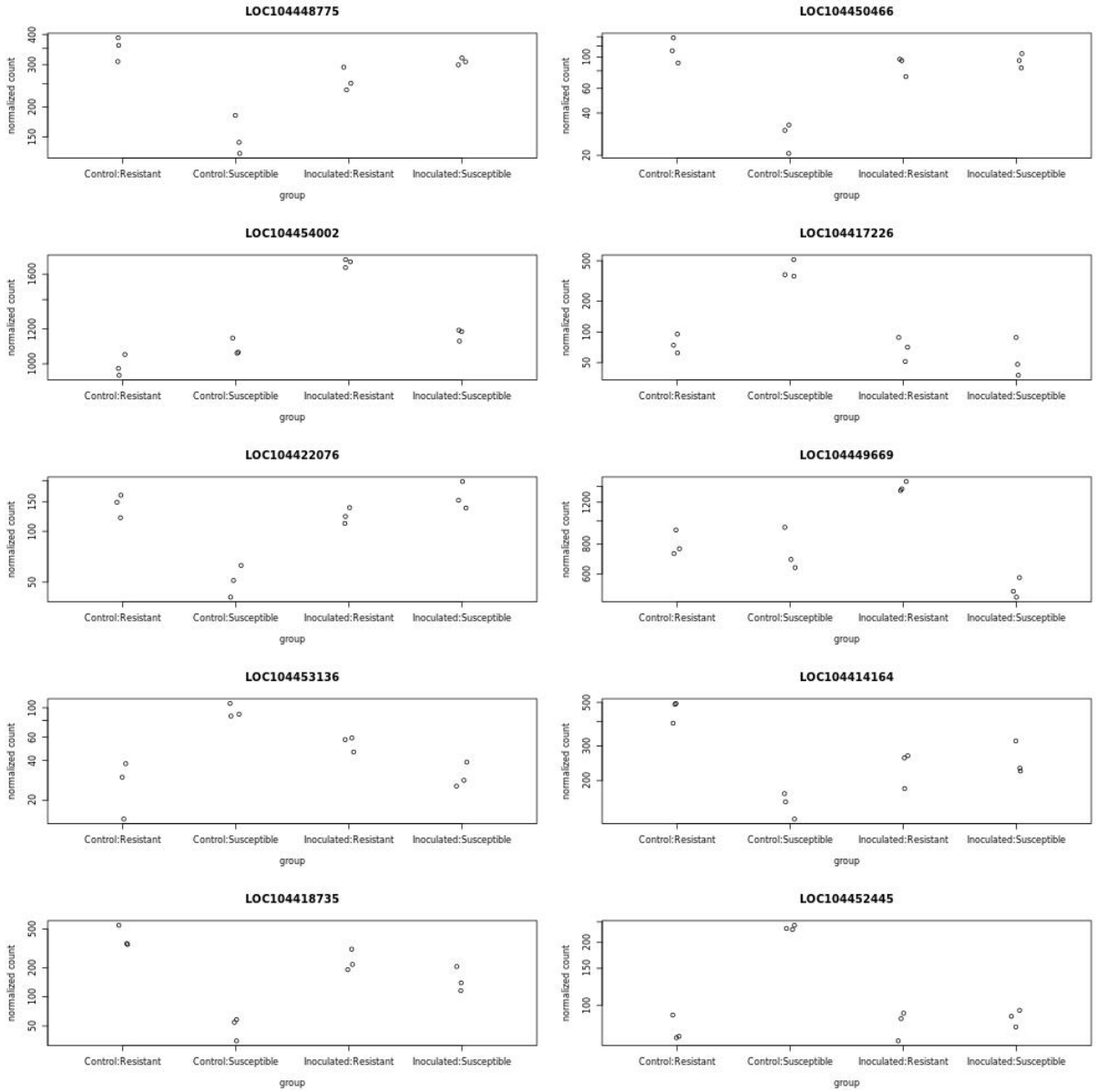


Figura 17. Niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (interacción resistencia:inoculación)

Los mapas de calor basados en los dos criterios de selección (menores p-valores ajustados o mayor log2FoldChange) muestran como pueden variar los resultados obtenidos y la importancia de fijar un criterio para posibles comparaciones (Figuras 18 y 19).

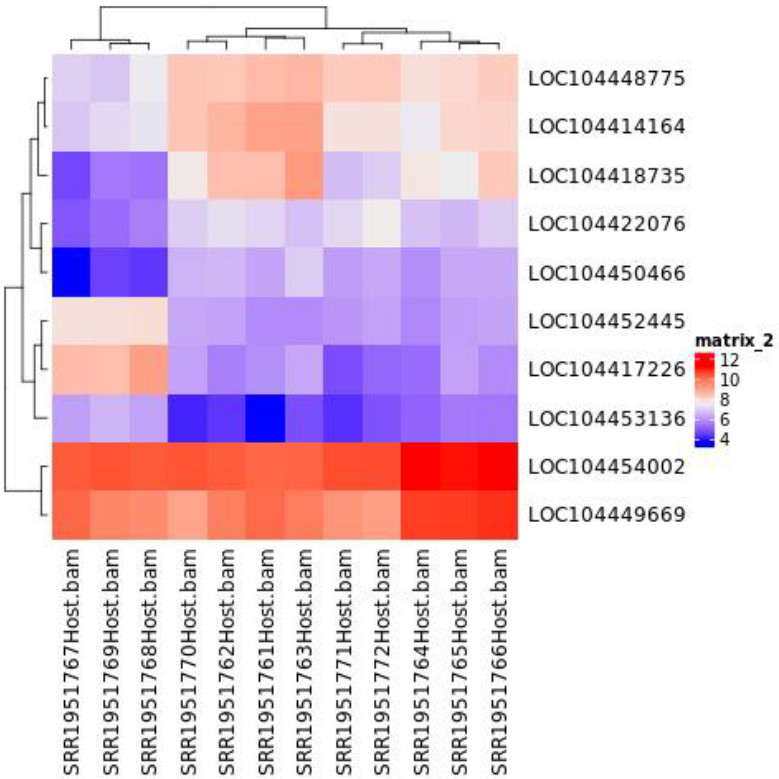


Figura 18. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (interacción resistencia:inoculación)

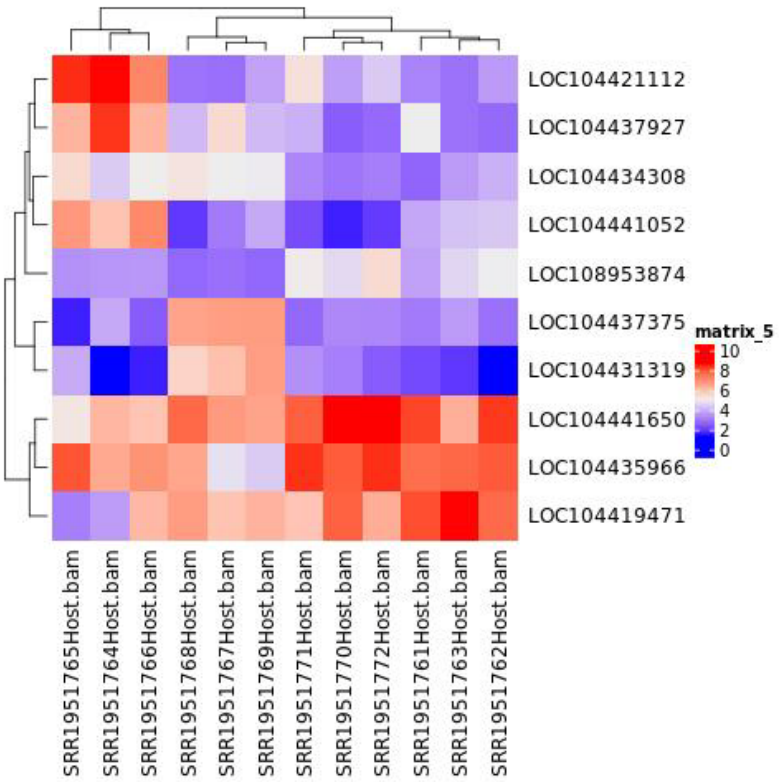


Figura 19. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un mayor log2FoldChange (interacción resistencia:inoculación)

3.4.2. Efecto de la inoculación en el clon resistente

Un total de 2429 genes fueron diferencialmente expresados, de los cuales 1112 fueron infraregulados y 1317 sobreregulados. Este valor es superior al aportado por Mangwanda et al., (2015), quienes encontraron 856 genes sobreregulados y 682 infraregulados. La discrepancia podría ser debido a que en para este TFM se trabajó únicamente con 4 de las 40 M reads. En el heatmap de la correlación de los conteos de cada muestra no se observa un patrón claro porque las dos muestras que fueron identificadas como potenciales outliers en el PCA enmascaran los resultados (Figura 20). Esto pone de nuevo de manifiesto la importancia de tener suficientes réplicas biológicas para poder descartar potenciales outliers.

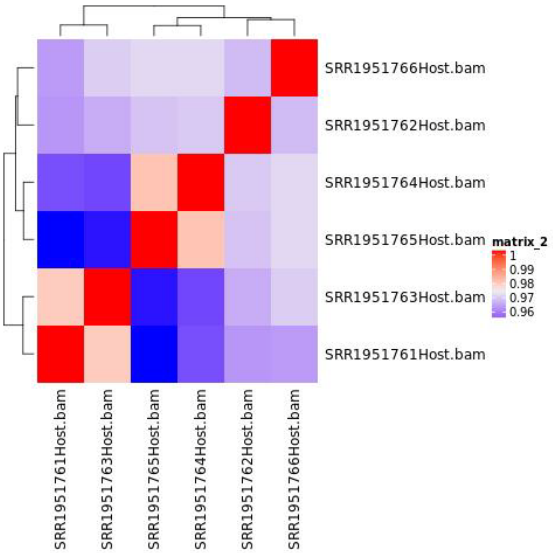


Figura 20. Heatmap de la correlación de los conteos (efecto inoculación en clon resistente)

El análisis de componentes principales, el cual explica el 88% de la varianza en sus dos primeros ejes, diferencia el tratamiento inoculado y el control en el eje de las X (Figura 21). Si bien, de nuevo se observan esas dos muestras que se agrupan de forma conjunta que dificultan el análisis.

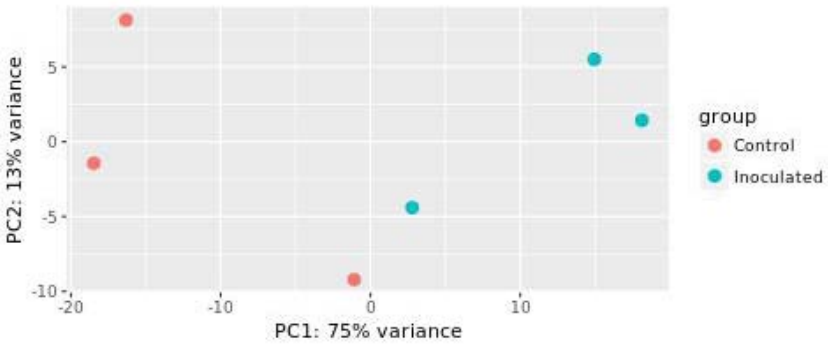


Figura 21. Gráfico del análisis de componentes principales (efecto inoculación en clon resistente)

Los 10 genes más diferencialmente expresados (bajo el criterio de menor p-valor ajustado) fueron LOC104422425, LOC104421112, LOC104415812, LOC104439413, LOC104443948, LOC104447317, LOC104421444, LOC104440756, LOC104421134, LOC104449755). El tratamiento inoculado estuvo sobreregulado frente al control en 7 de ellos, lo que podría indicar una activación consecuencia de la infección (Figura 22).

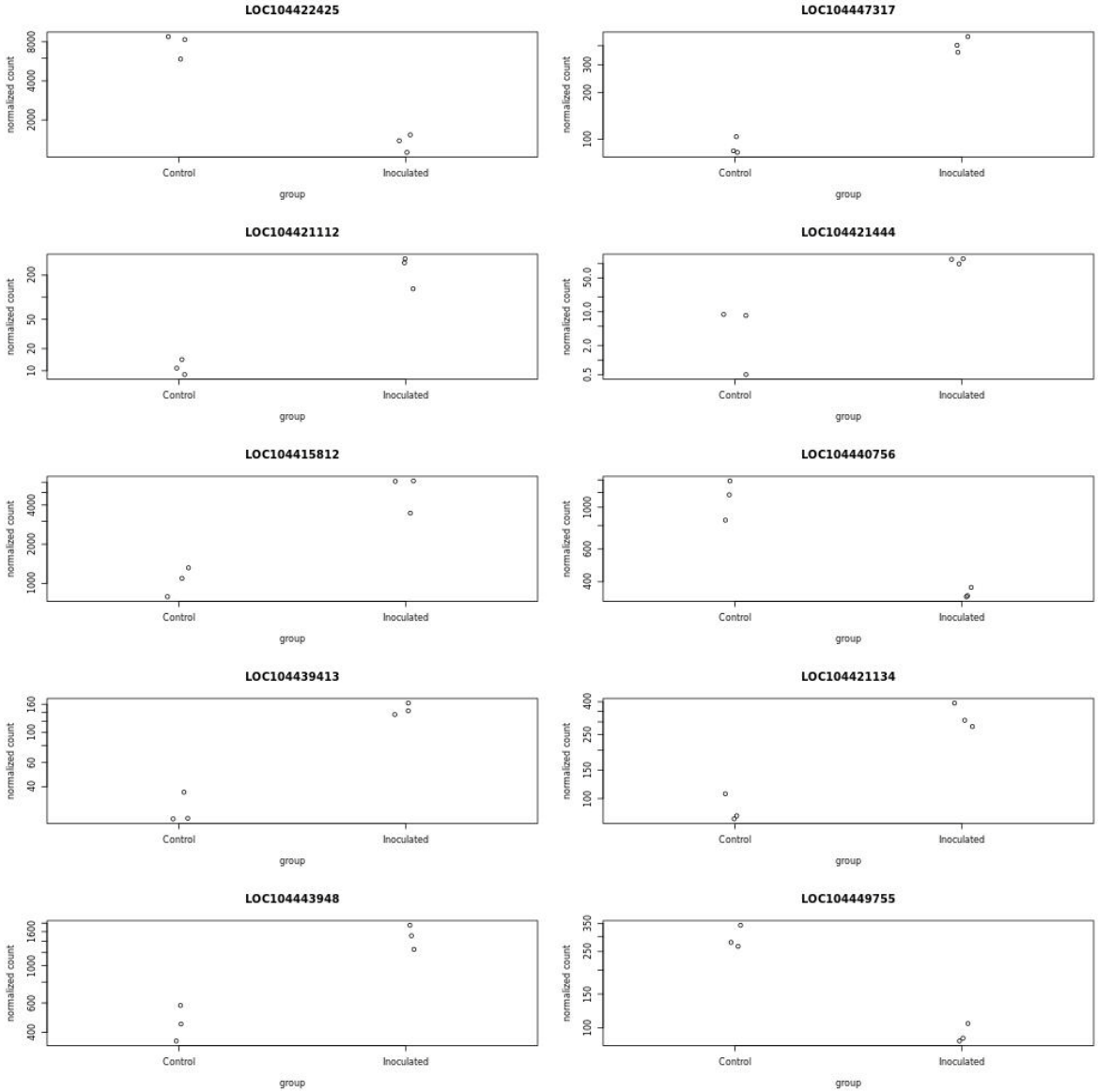


Figura 22. Niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (efecto inoculación en clon resistente)

El mapa de calor de los 10 genes más diferencialmente expresados, muestra diferencias claras entre tratamientos, no observándose resultados anómalos en las dos muestras identificadas como potenciales outliers (Figura 23). Esto apunta a la importancia de complementar el análisis general de todos los genes diferencialmente expresados con uno más en detalle en aquellos más relevantes.

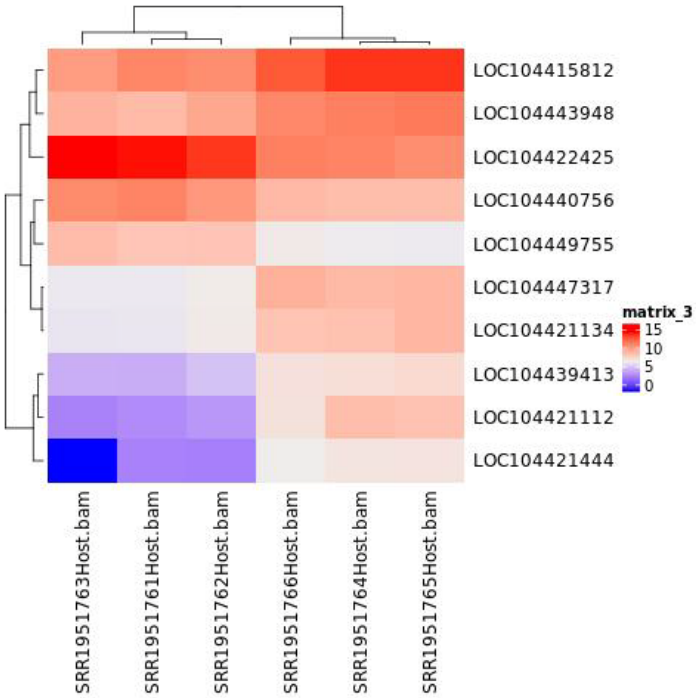


Figura 23. Heatmap de niveles de expresión génica de los 10 genes diferencialmente expresados con un menor p-valor ajustado (efecto inoculación en clon resistente)

3.5. Análisis de enriquecimiento funcional

3.5.1. Interacción resistencia:inoculación

El blast del análisis funcional (a partir de 10 genes seleccionados bajo el criterio de menor pvalor ajustado) muestra que la especie que más macheo con dichas secuencias de la base de datos fue *E. grandis* (lo cual era de esperar), seguido de otros organismos vegetales (Figura 24).

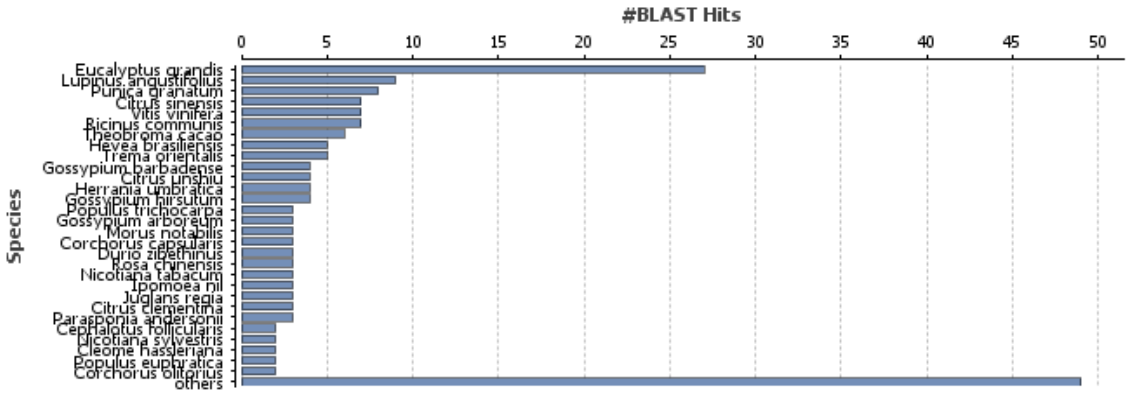


Figura 24. Distribución de especies (interacción resistencia:inoculación)

El análisis funcional también determinó que los procesos biológicos implicados fueron procesos metabólicos, localización, proceso celular, regulación biológica y respuesta a estímulos (Figura 25). Entre otras, los procesos metabólicos, de oxidación-reducción, procesos relacionados con el transporte y asimilación del nitrógeno, respuesta al ácido abcísico y "reactive oxygen species (ROS) biosynthetic process" (Figura 26). De estos, ROS es conocido por ser uno de las primeras respuestas celulares de la planta tras el reconocimiento del patógeno (Torres et al. 2006), el ácido abcísico está relacionado con la inducción de defensas (Verma et al. 2016) y la asimilación de nitrato puede estar asociada a los procesos de infección hospedante/patógeno (Divon and Fluhr 2007).

Para un mayor detalle del análisis funcional ver Anexo III

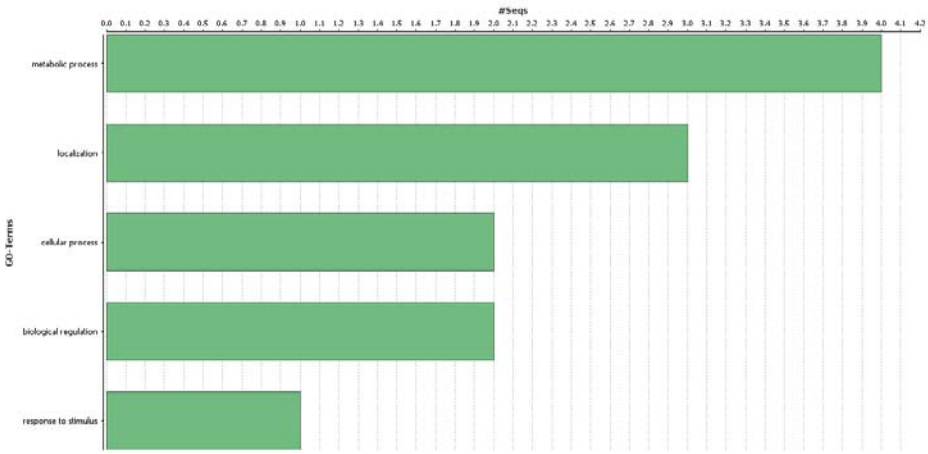


Figura 25. Procesos biológicos desencadenados en el hospedante (interacción resistencia:inoculación)

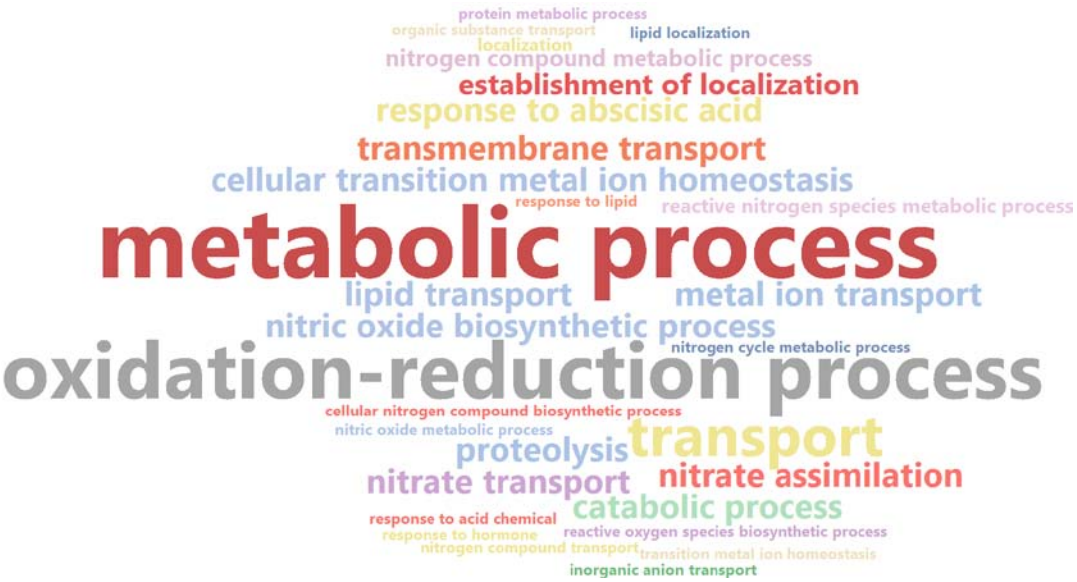


Figura 26. WorldCloud de los procesos biológicos desencadenados en el hospedante (interacción resistencia:inoculación)

A su vez, las funciones moleculares implicadas fueron actividad catalítica y “binding” (Figura 27). En concreto, “cofactor binding”, “transition metal ion binding”, “nitrate reductase (NADPH) activity, etc (Figura 28). Para un mayor detalle del análisis funcional ver Anexo IV

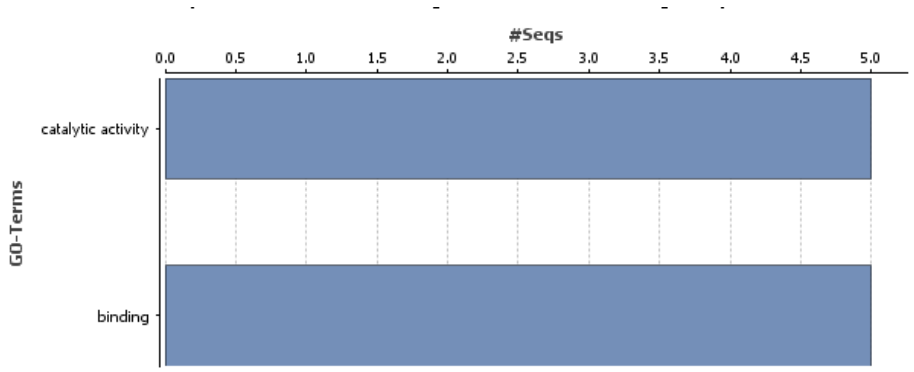


Figura 27. Funciones moleculares desencadenados en el hospedante (interacción resistencia:inoculación)

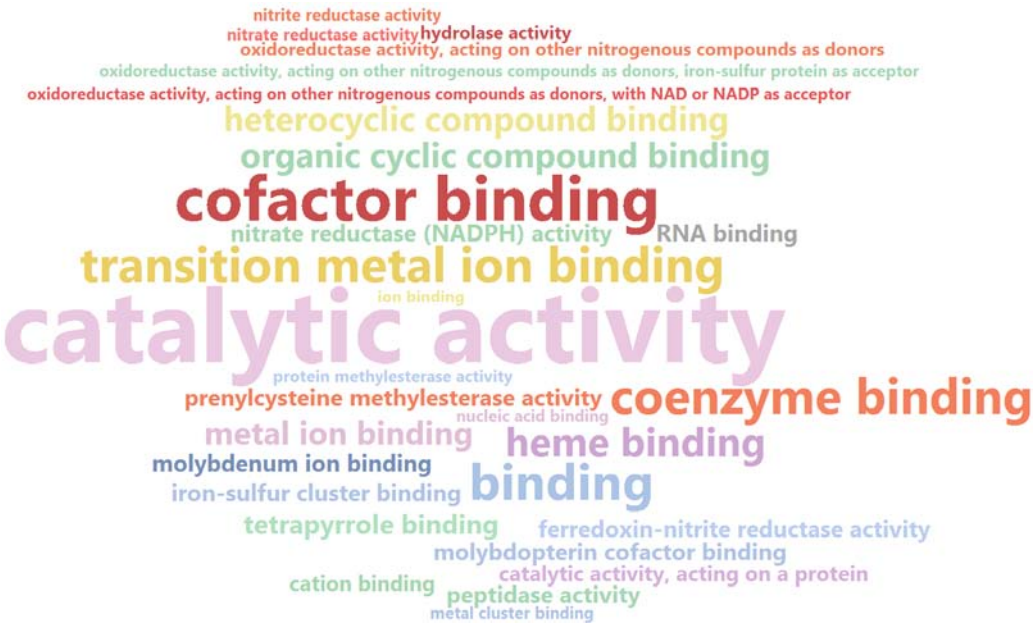


Figura 28. WorldCloud de las funciones moleculares desencadenados en el hospedante (interacción resistencia:inoculación)

En lo referente a los componentes celulares, partes de la célula y de la membrana parecen estar implicados en los procesos (Figura 29). En concreto componentes integrales de la membrana, cytoplasma, partes intracelulares, etc (Figura 30). Para un mayor detalle del análisis funcional ver Anexo V

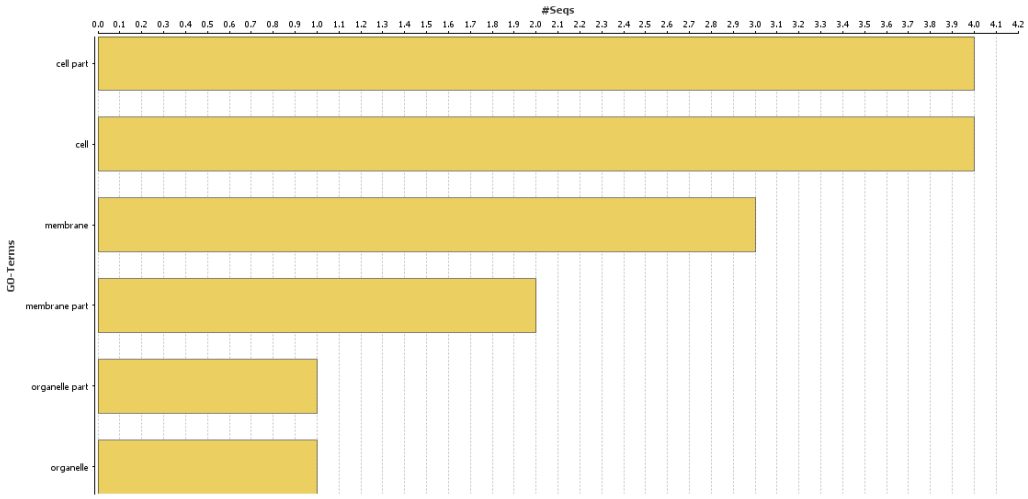


Figura 29. Componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (interacción resistencia:inoculación)

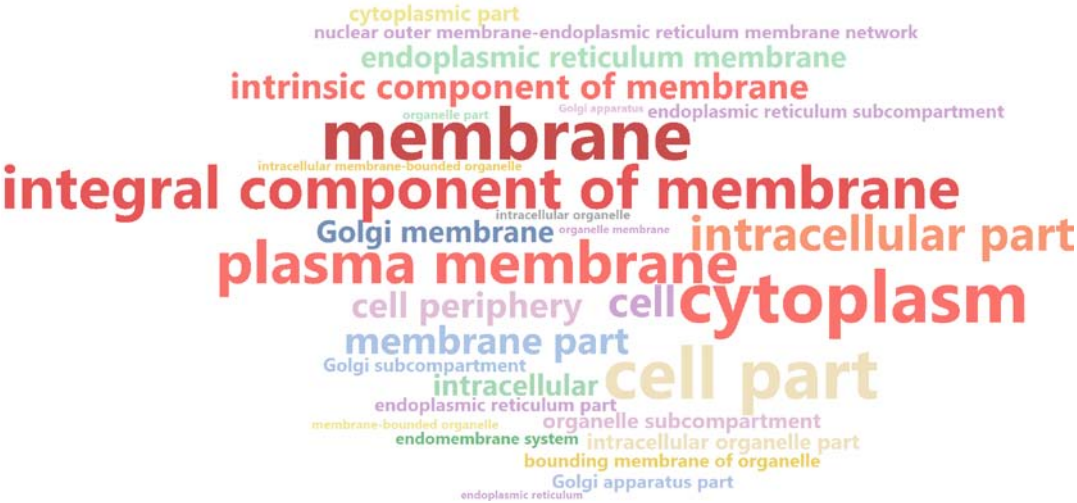


Figura 30. WorldCloud de los componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (interacción resistencia:inoculación)

Los pathway-maps identificados desde KEGG fueron “Nitrogen metabolism” y “Drug Metabolism - Other Enzymes” (Anexos VI y VII).

3.5.2. Interacción resistencia:inoculación

El blast del análisis funcional en este caso fue muy similar al obtenido con la interacción (Figura 31), si bien en este caso un par de especies del chopo (*Populus* sp.) mostraron un mayor porcentaje de hits que en el caso anterior.

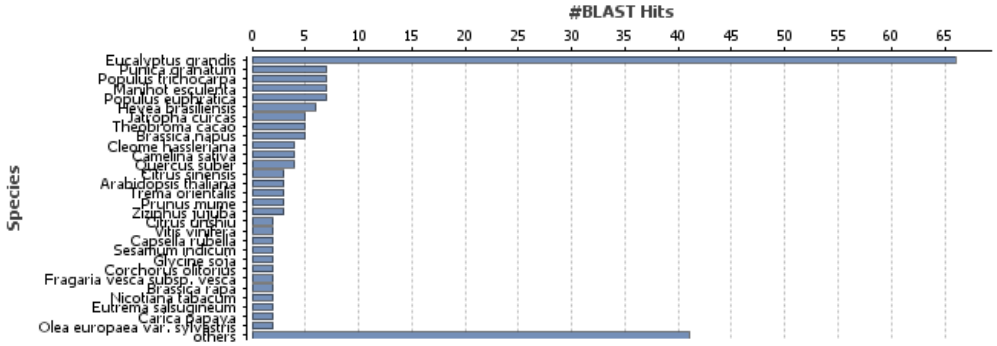


Figura 31. Distribución de especies (efecto inoculación en clon resistente)

El análisis funcional muestra que los procesos biológicos implicados fueron procesos metabólicos, celulares, de respuesta a estímulos, regulación biológica, etc (Figura 32). Entre otras, “lipid metabolic process”, procesos de oxidación-reducción, de biosíntesis celular, etc (Figura 33). Para un mayor detalle del análisis funcional ver Anexo VIII.

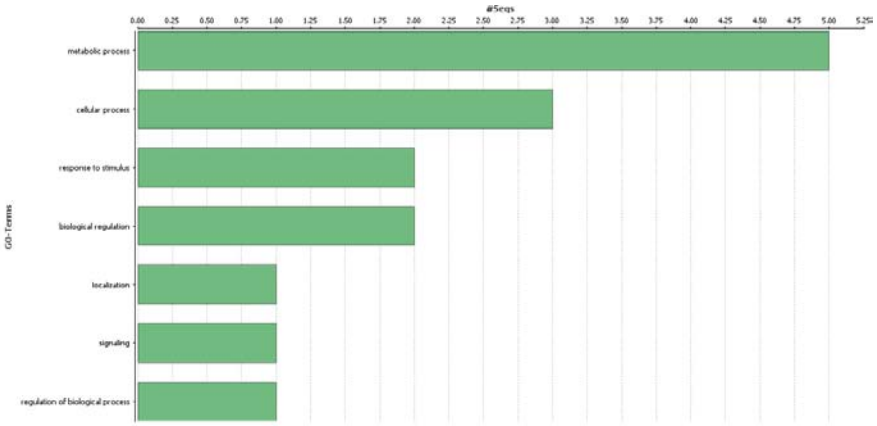


Figura 32. Procesos biológicos desencadenados en el hospedante (efecto inoculación en clon resistente)

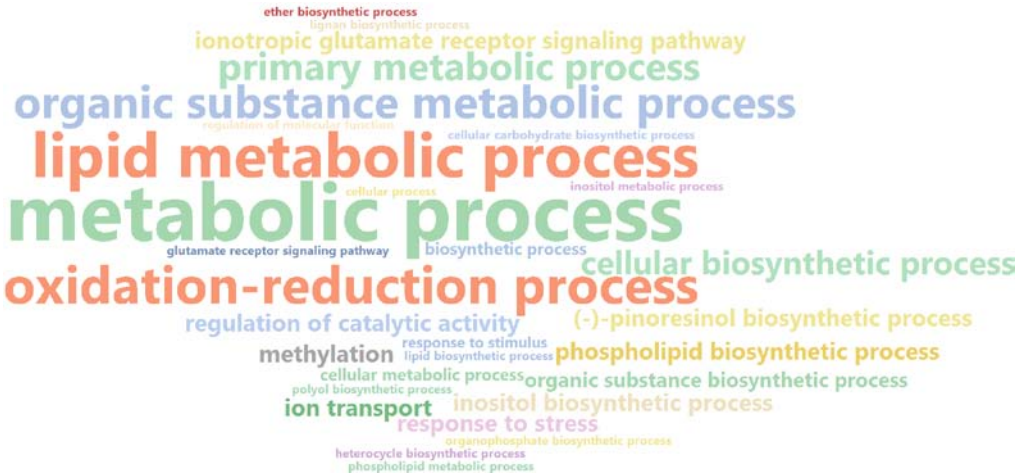


Figura 33. WorldCloud de los procesos biológicos desencadenados en el hospedante (efecto inoculación en clon resistente)

De nuevo entre las funciones moleculares implicadas destaca la actividad catalítica y ya en menor medida actividad de transporte, reguladores de funciones moleculares y actividad de traductor molecular (Figura 34). Entre otras, actividad oxidoreductasa, “transferase activity”, etc (Figura 35). Para un mayor detalle del análisis funcional ver Anexo IX

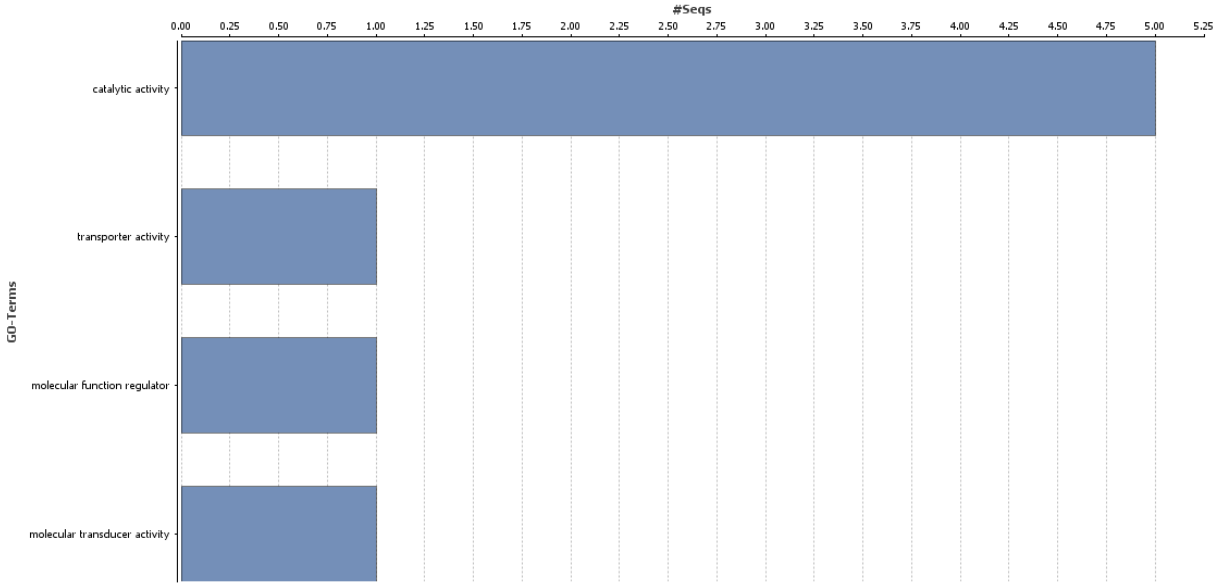


Figura 34. Funciones moleculares desencadenados en el hospedante (efecto inoculación en clon resistente)

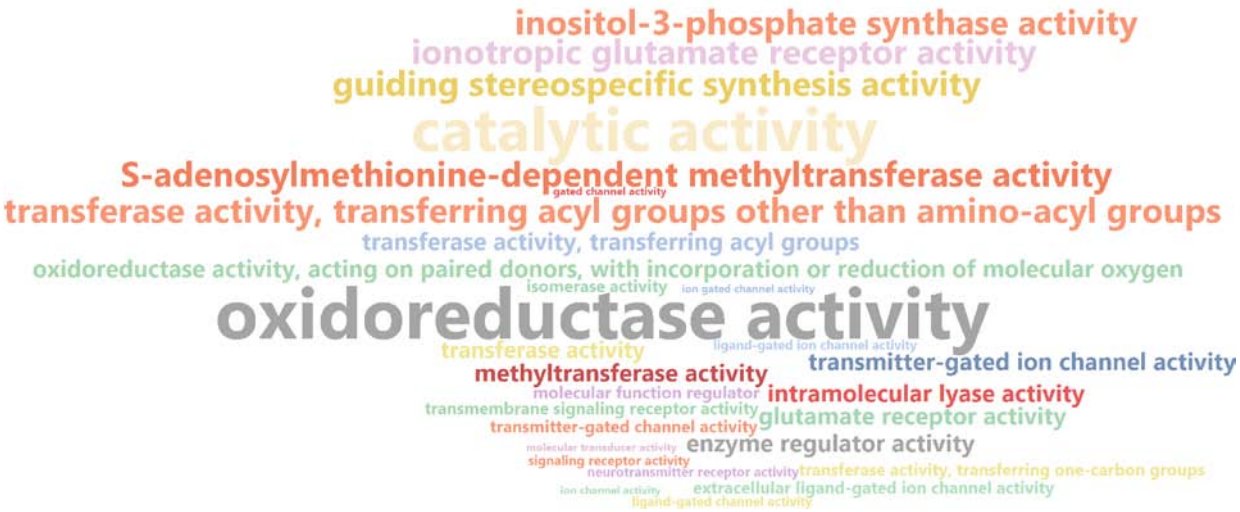


Figura 35. WorldCloud de las funciones moleculares desencadenados en el hospedante (efecto inoculación en clon resistente)

En lo referente a los componentes celulares, al igual que en el caso de la interacción, partes de la membrana y de la célula parecen estar implicados en los procesos (Figura 36). En concreto componentes integrales e intrínsecos de la membrana, el cytoplasma, partes intracelulares, etc (Figura 37). Para un mayor detalle del análisis funcional ver Anexo X

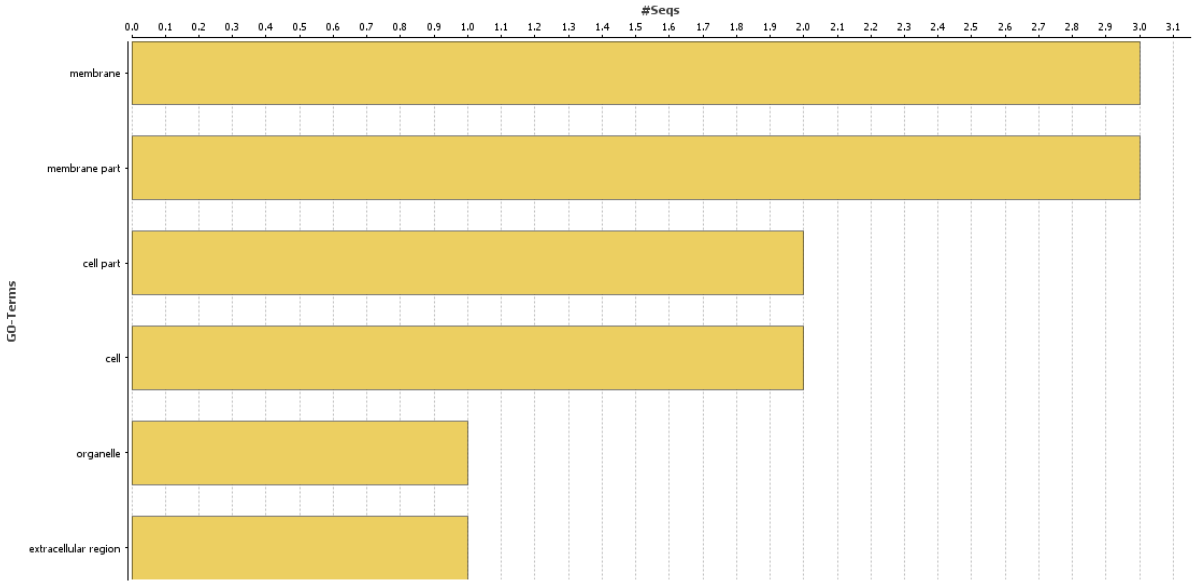


Figura 36. Componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (efecto inoculación en clon resistente)

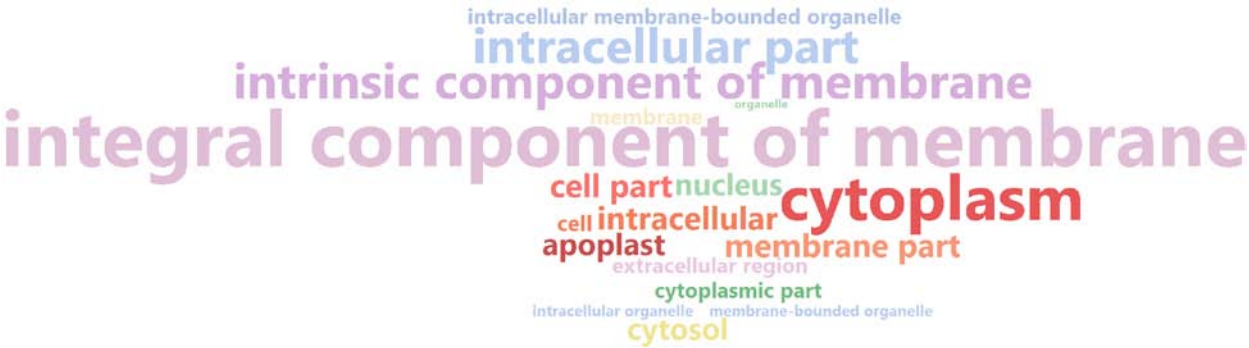


Figura 37. WorldCloud de los componentes celulares relacionados con los niveles de expresión desencadenados en el hospedante (efecto inoculación en clon resistente)

Los pathway-maps identificados desde KEGG fueron “Biosynthesis of antibiotics”, “Inositol Phosphate metabolism” y “Streptomycin biosynthesis” (Anexos XI, XII y XIII)

4. Conclusiones

El análisis RNA-seq es una técnica muy potente pero requiere de una alta capacidad de computación. En el presente TFM se ha podido comprobar que un supercomputador de 16 cpu con16 Gb de RAM permitió analizar secuencias de 4 millones de reads (~2 Gb) en un tiempo aceptable. Sin embargo, secuencias de 40 M de reads (~10 Gb) no pudieron ser procesadas.

En la actualidad se dispone de software libre para llevar a cabo un análisis de RNA-seq. A su vez existe un gran feedback entre bioinformáticos distintos blogs que permiten solucionar muchos de los problemas encontrados durante el desarrollo del pipeline.

Para cada paso del pipeline hay disponibles distintos software, cada uno de los cuales implementan procesos y algoritmos diferentes por lo que se debe ser consciente que la comparación entre estudios tiene ciertas limitaciones.

El análisis funcional es una herramienta muy potente para identificar los procesos biológicos implicados. Sin embargo, actualmente son todavía pocos los estudios en sistemas vegetales que han testado la importancia y funcionamiento de esos procesos de forma empírica. Por ello, una interpretación y discusión de muchos de los resultados obtenidos puede ser demasiado especulativa.

En general se han cumplido los objetivos planteados inicialmente. Si bien, no se consiguió completamente el objetivo de desarrollar un pipeline y sus correspondientes scripts completamente en software libre, porque por falta de tiempo el análisis funcional fue llevado a cabo con un software comercial.

Otro de los hándicaps del presente TFM fue que al trabajar en el supercomputador en control remoto me imposibilitaba copiar y pegar los resultados que se proyectaban en el terminal. De hecho la idea inicial era trabajar con Rstudio y exportar los scripts y el informe de resultados con R-Markdown, pero tampoco fue posible.

La planificación tuvo que ser reajustada por los retrasos originados por los problemas de computación. De hecho, aunque finalmente se obtuvo permiso para trabajar con un supercomputador, este no fue capaz de mapear las secuencias completas (40 M reads; ~12Gb) y se tuvo que trabajar con una submuestra de 4 M reads. Esto último imposibilitó una comparación fidedigna con el artículo publicado por los autores del experimento.

Varias son las líneas de trabajo pendientes que no han podido ser llevadas a cabo en este TFM. Entre ellas, cabría destacar:

- ✓ Comparar del porcentaje de mapeo entre los distintos software disponibles
- ✓ Testar el efecto de la profundidad de secuenciación en los resultados finales

5. Glosario

BLAST	Basic Local Alignment Search Tool
DNA	Ácido desoxirribonucleico
GEO	Gene Expression Omnibus
GFF	General Feature Format
GO	Gene ontology
Heatmap	Mapa de calor
KEGG	Kyoto Encyclopedia of Genes and Genomes
NCBI	National Center for Biotechnology Information
PCA	Análisis de Componentes Principales
Pipeline	Proceso informático continuo
RNA	Ácido ribonucleico
RNA-seq	Secuenciación de ARN
ROS	Reactive Oxygen Species
Script	Archivo de procesamiento
SRA	Sequence Read Archive
TFM	Trabajo Fin de Máster

6. Bibliografía

Andrews S (2010) FastQC: a quality control tool for high throughput sequence data.

Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics 30:2114–2120. doi: 10.1093/bioinformatics/btu170

Bradshaw RE, Guo Y, Sim AD, et al (2016) Genome-wide gene expression dynamics of the fungal pathogen Dothistroma septosporum throughout its infection cycle of the gymnosperm host Pinus radiata. Molecular Plant Pathology 17:210–224. doi: 10.1111/mpp.12273

CCB (2018) gffread utility. http://ccb.jhu.edu/software/stringtie/gff.shtml#gffread. Accessed 10 May 2018

Costa-Silva J, Domingues D, Lopes FM (2017) RNA-Seq differential expression analysis: An extended review and a software tool. PLoS ONE 12:e0190152. doi: 10.1371/journal.pone.0190152

Del Fabbro C, Scalabrin S, Morgante M, Giorgi FM (2013) An extensive evaluation of read trimming effects on illumina NGS data analysis. PLoS ONE 8:e85024. doi: 10.1371/journal.pone.0085024

Divon HH, Fluhr R (2007) Nutrition acquisition strategies during fungal infection of plants. FEMS Microbiology Letters 266:65–74. doi: 10.1111/j.1574-6968.2006.00504.x

Dobin A, Davis CA, Schlesinger F, et al (2013) STAR: Ultrafast universal RNA-seq aligner. Bioinformatics 29:15–21. doi: 10.1093/bioinformatics/bts635

García-Ortega LF, Martínez O (2015) How many genes are expressed in a transcriptome? Estimation and results for RNA-Seq. PLoS ONE 10:e0130262. doi: 10.1371/journal.pone.0130262

Götz S, García-Gómez JM, Terol J, et al (2008) High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic Acids Research 36:3420–3435. doi: 10.1093/nar/gkn176

Gu Z, Eils R, Schlesner M (2016) Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics 32:2847–2849. doi: 10.1093/bioinformatics/btw313

Kim D, Pertea G, Trapnell C, et al (2013) TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biology 14:R36. doi: 10.1186/gb-2013-14-4-r36

Langmead B, Salzberg S (2012) Fast gapped-read alignment with Bowtie 2. Nature

Methods 9:357–359.

Liao Y, Smyth GK, Shi W (2013) The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. Nucleic Acids Research 41:e108. doi: 10.1093/nar/gkt214

Liu Y, Zhou J, White KP (2014) RNA-seq differential expression studies: More sequence or more replication? Bioinformatics 30:301–304. doi: 10.1093/bioinformatics/btt688

Love M, Anders S, Huber W (2018a) Package " DESeq2 " Differential gene expression analysis based on the negative binomial distribution. 53.

Love MI, Anders S, Huber W (2018b) Analyzing RNA-seq data with DESeq2. http://www.bioconductor.org/packages/3.7/bioc/vignettes/DESeq2/inst/doc/DESeq2.html. Accessed 14 May 2018

Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology 15:550. doi: 10.1186/s13059-014-0550-8

Mangwanda R, Myburg AA, Naidoo S (2015) Transcriptome and hormone profiling reveals Eucalyptus grandis defence responses against Chrysoporthe austroafricana. BMC Genomics 16:1–13. doi: 10.1186/s12864-015-1529-x

Martín-Rodrigues N, Espinel S, Sanchez-Zabala J, et al (2013) Spatial and temporal dynamics of the colonization of *Pinus radiata* by *Fusarium circinatum* , of conidiophora development in the pith and of traumatic resin duct formation. New Phytologist 198:1215–1227. doi: 10.1111/nph.12222

NCBI (2018) SRA Toolkit.

Pagès H, Aboyoun P, Gentleman R, DebRoy S (2018) Biostrings: Efficient manipulation of biological strings.

R Development Core Team (2008) R: A language and environment for statistical computing. R Foundation for Statistical Computing.

Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26:139–140. doi: 10.1093/bioinformatics/btp616

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? Rna 22:1641–1641. doi: 10.1261/rna.058339.116

Tarazona S, Furió-Tarí P, Turrà D, et al (2015) Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package. Nucleic Acids Research 43:e140. doi: 10.1093/nar/gkv711

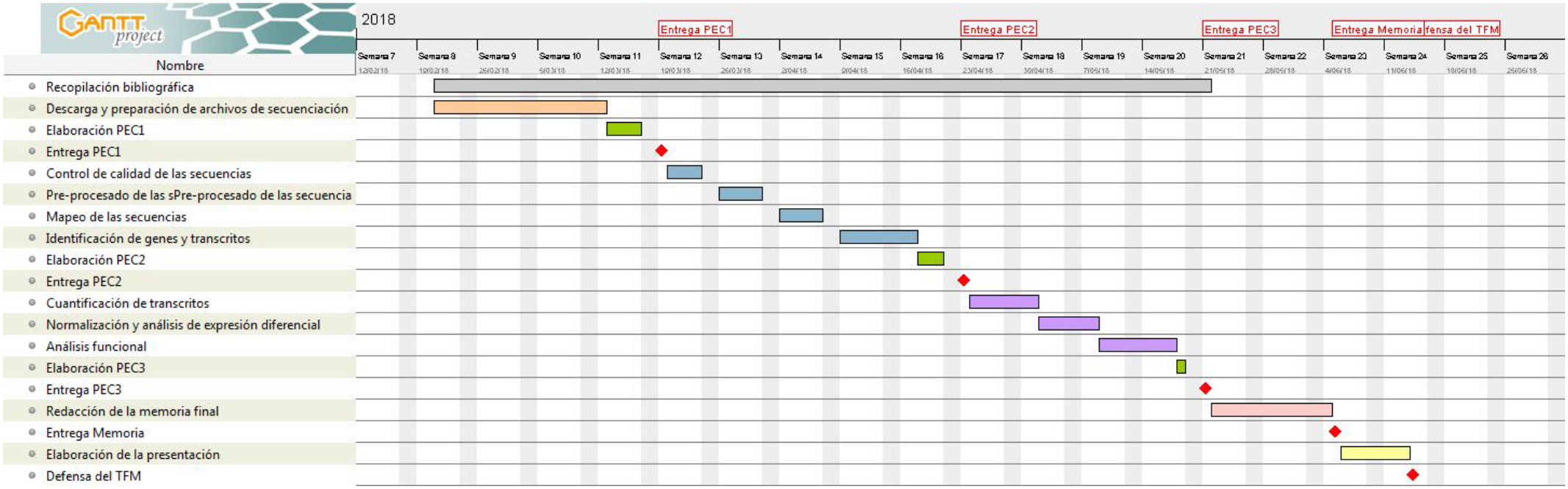
Torres MA, Jones JDG, Dangl JL (2006) Reactive oxygen species signaling in response to pathogens. Plant physiology 141:373–378. doi: 10.1104/pp.106.079467.conjunction

Verma V, Ravindran P, Kumar PP (2016) Plant hormone-mediated regulation of stress responses. BMC Plant Biology 16:86. doi: 10.1186/s12870-016-0771-y

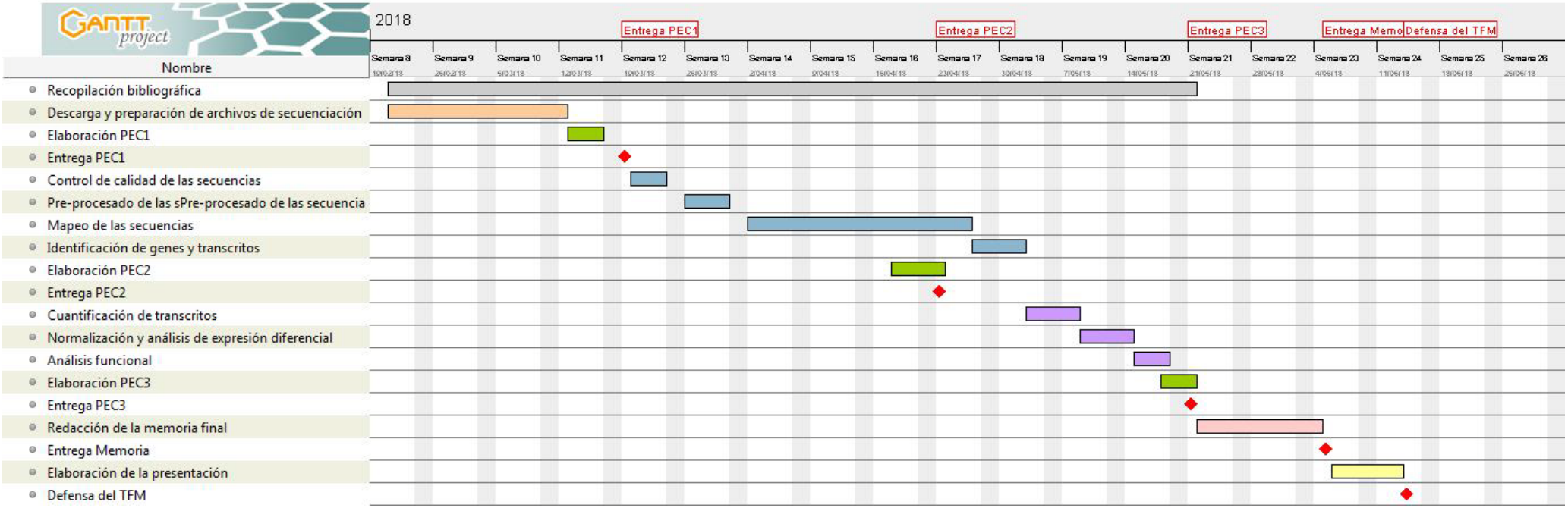
(2018) seqtk. https://github.com/lh3/seqtk. Accessed 1 May 2018

7. Anexos

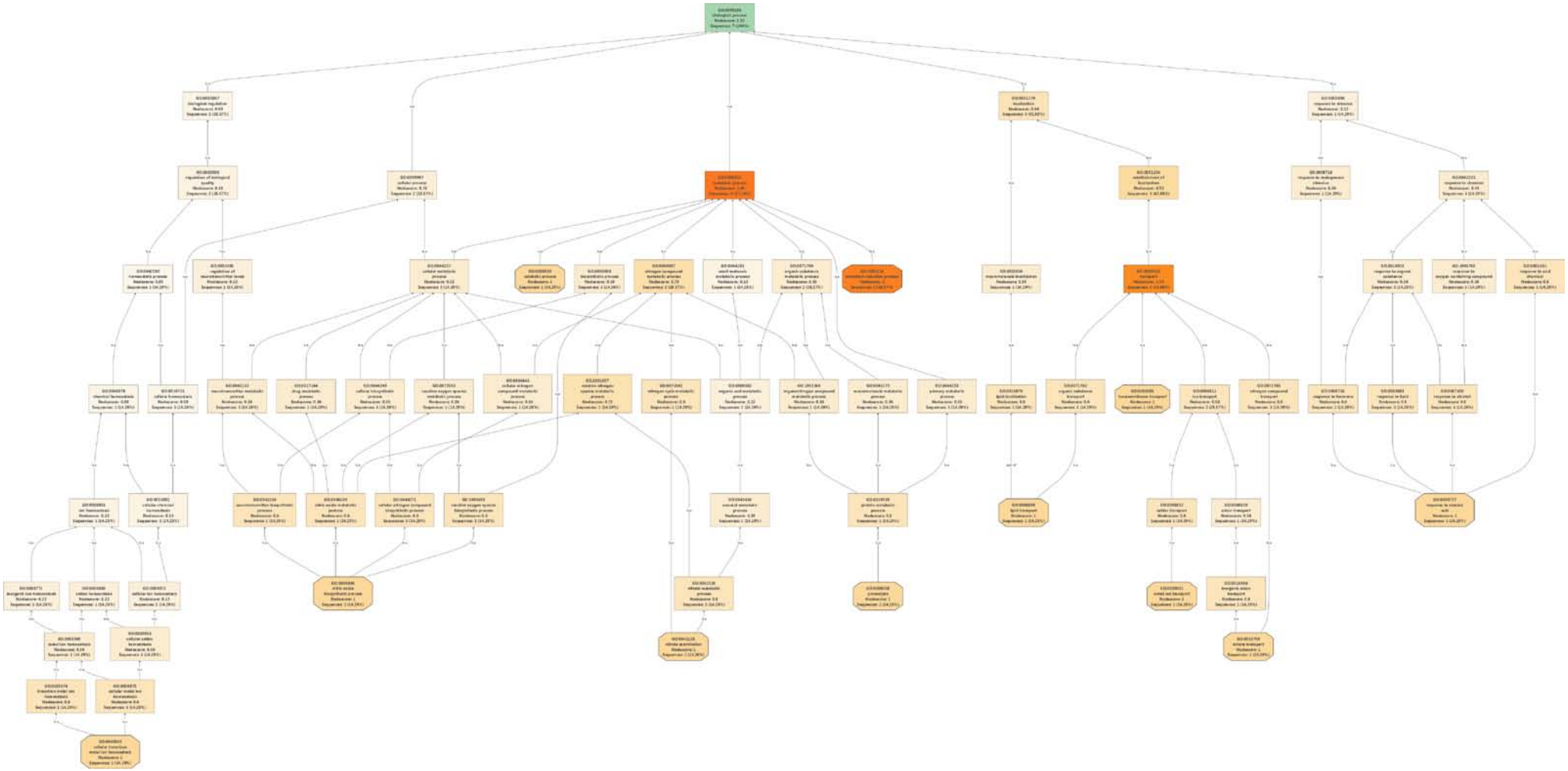
Anexo I. Cronograma inicialmente propuesto en la PEC1



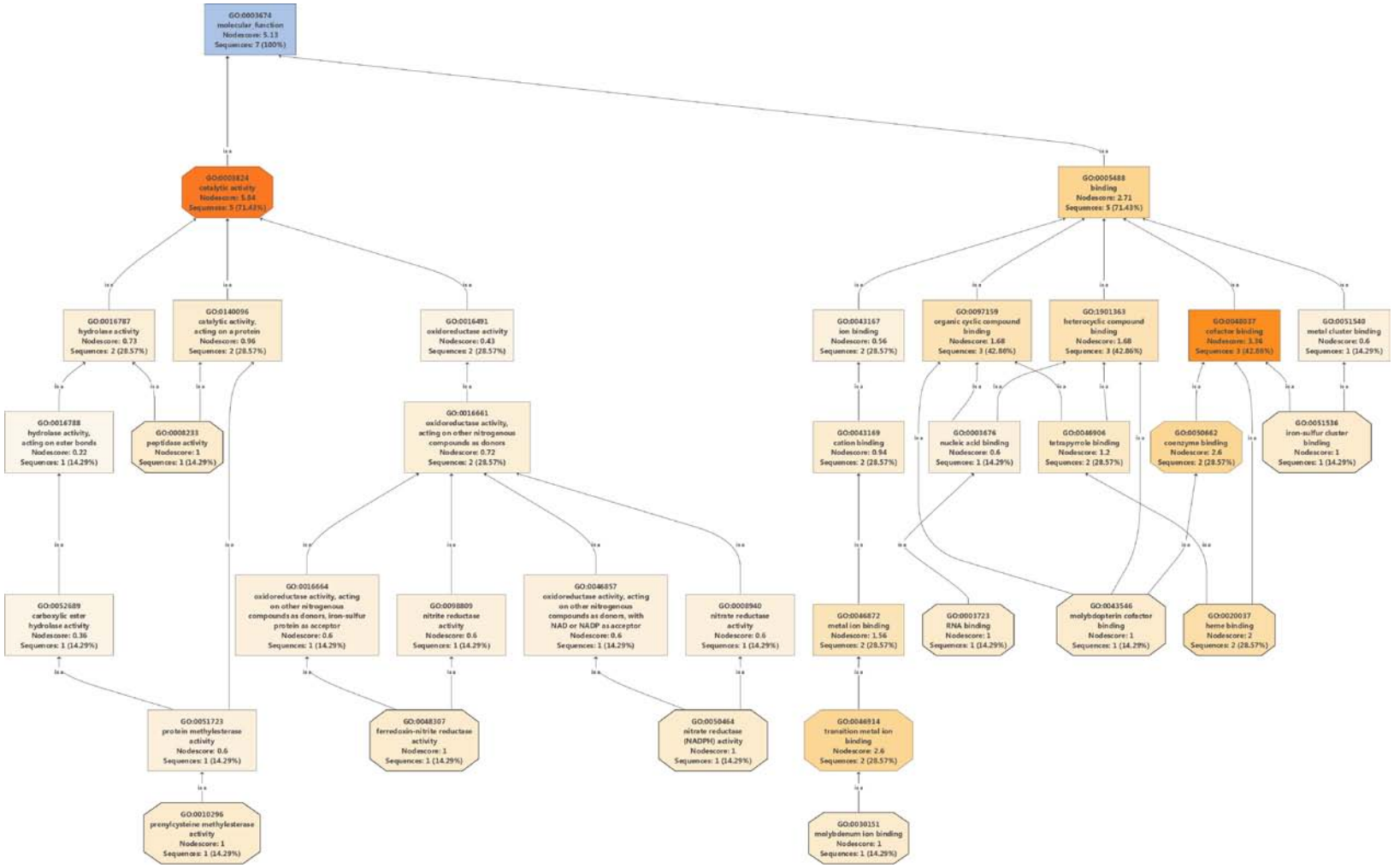
Anexo II. Cronograma modificado durante la PEC2



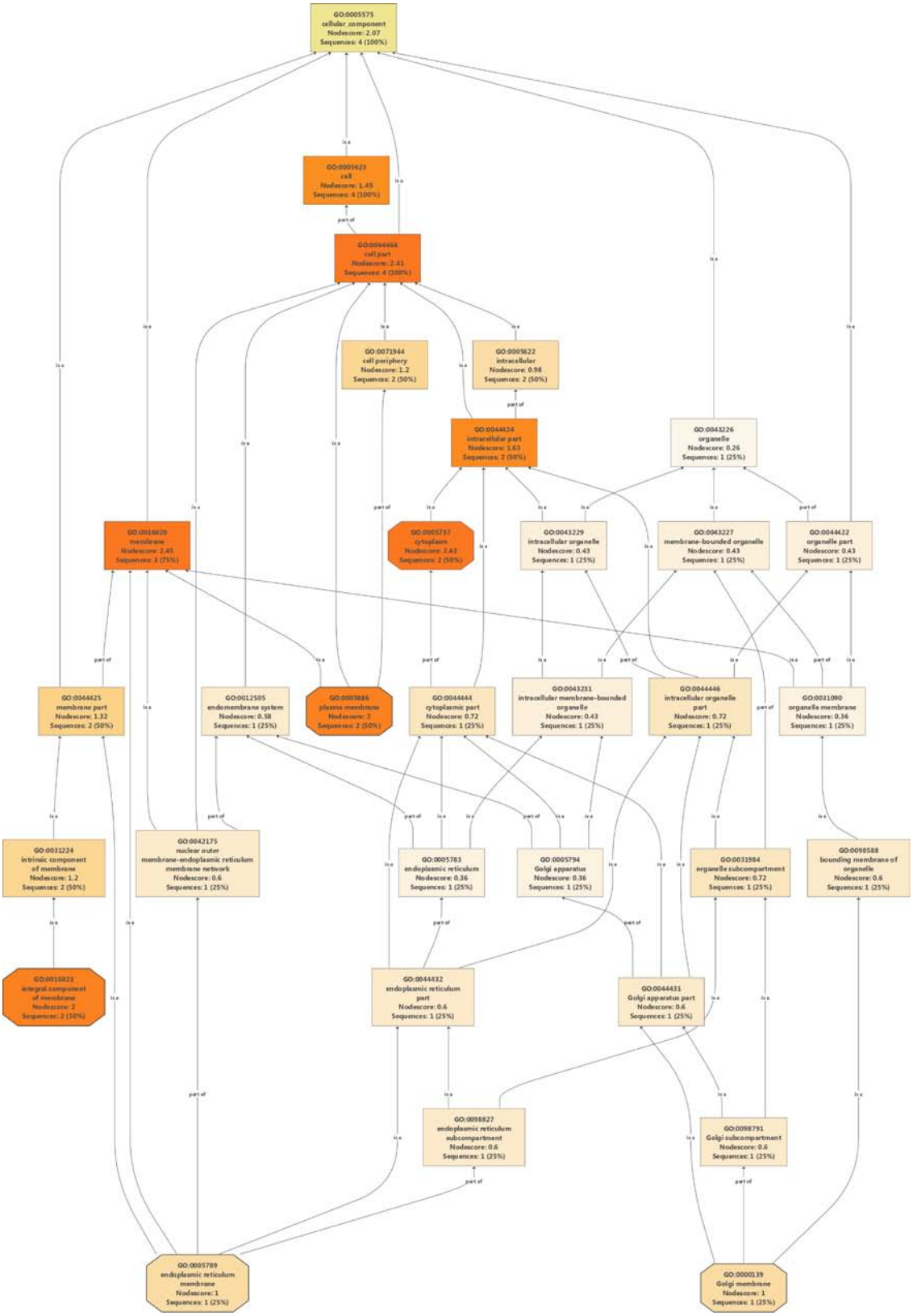
Anexo III. Procesos biológicos del análisis funcional (interacción resistencia:inoculación)



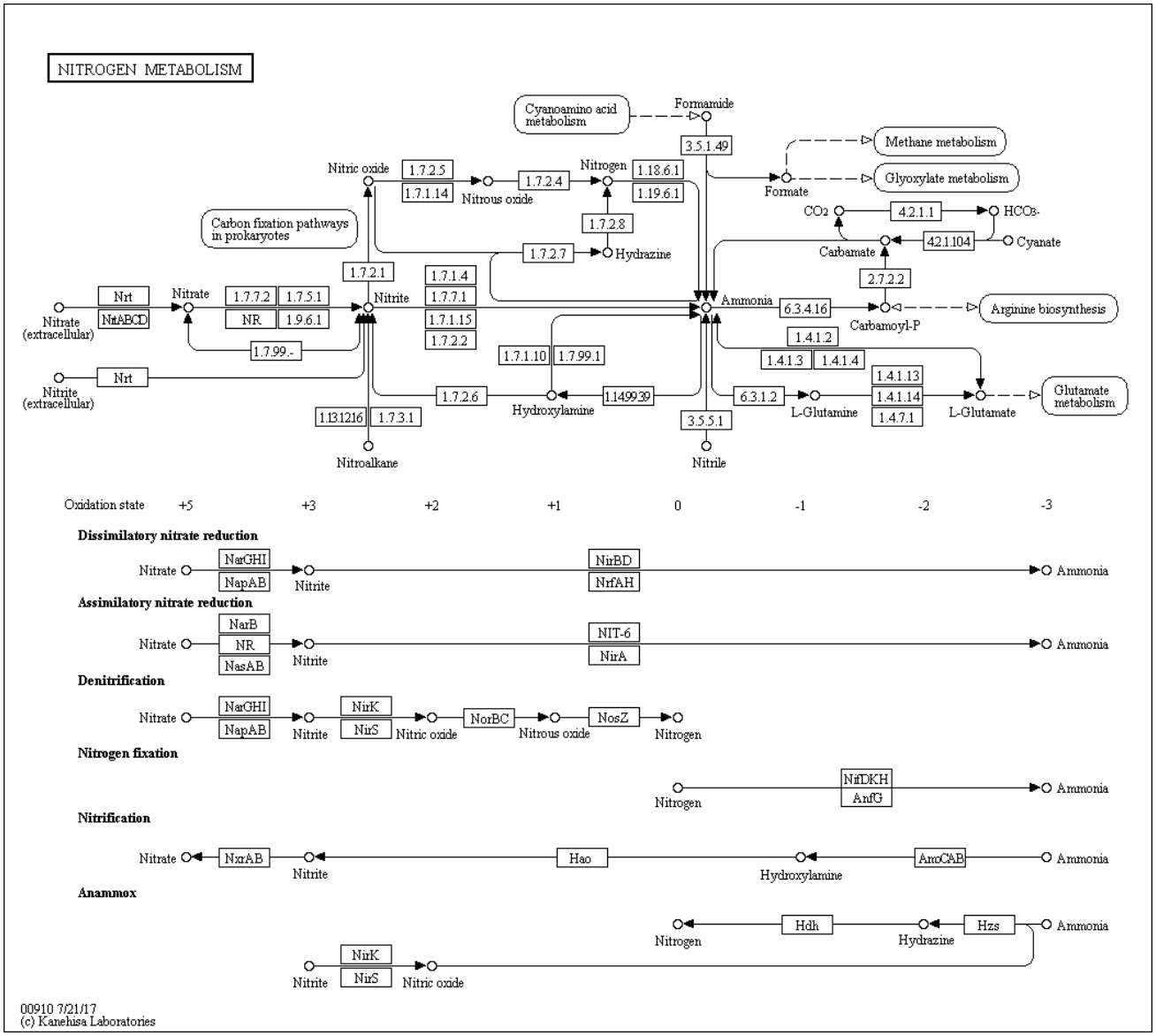
Anexo IV. Funciones moleculares del análisis funcional (interacción resistencia:inoculación)



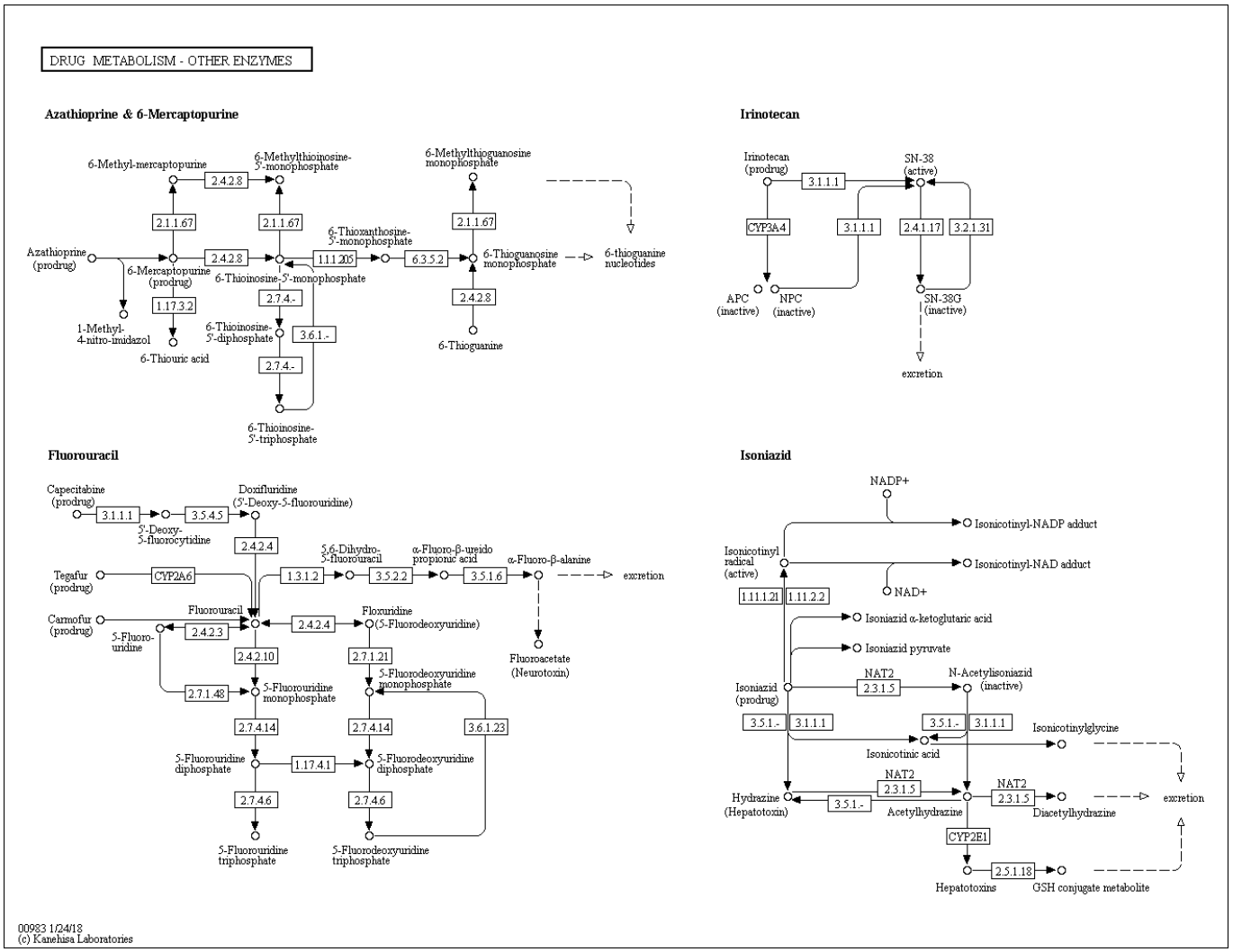
Anexo V. Componentes celulares del análisis funcional (interacción resistencia:inoculación)



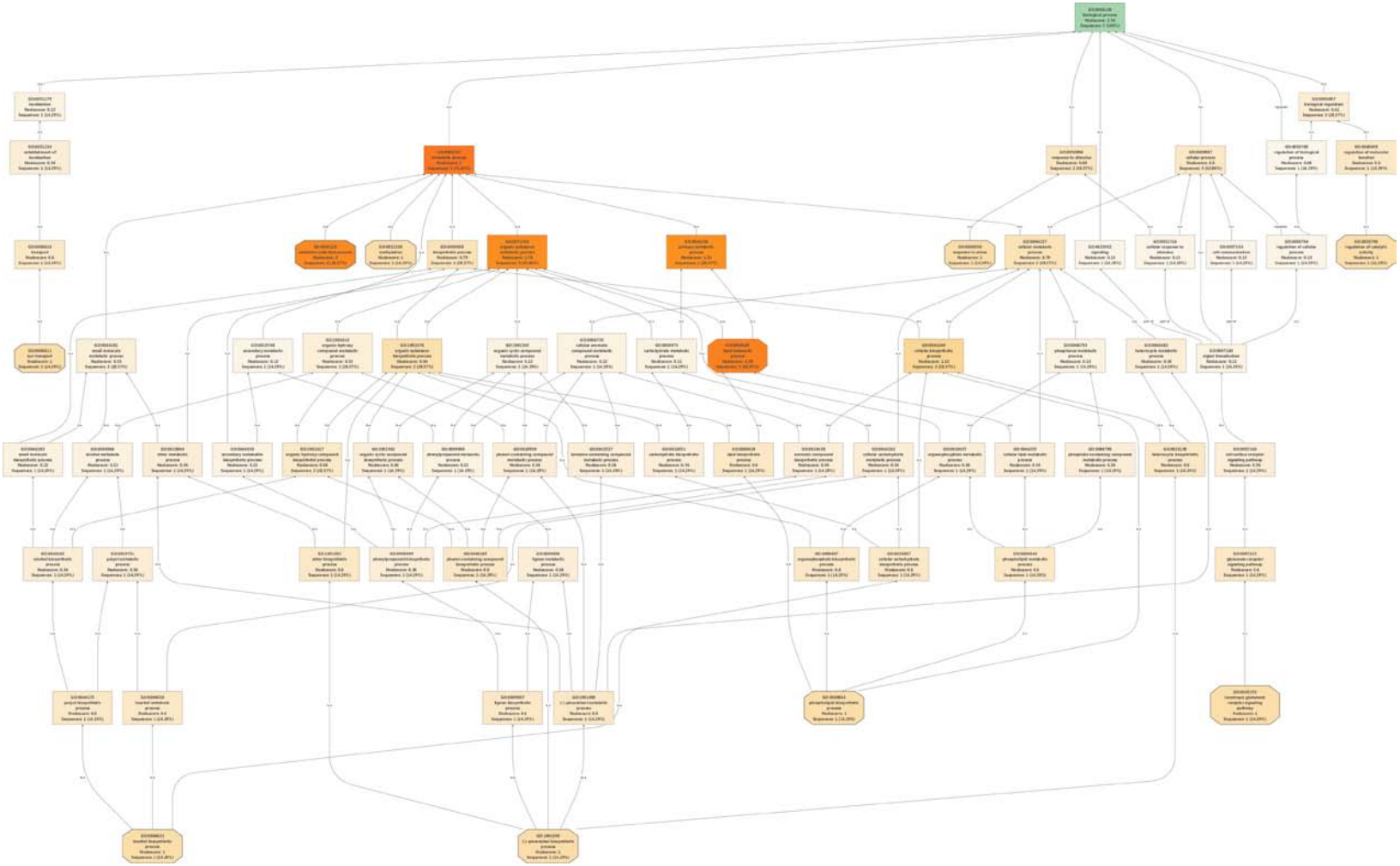
Anexo VI. Pathway Nitrogen Metabolism (interacción resistencia:inoculación)



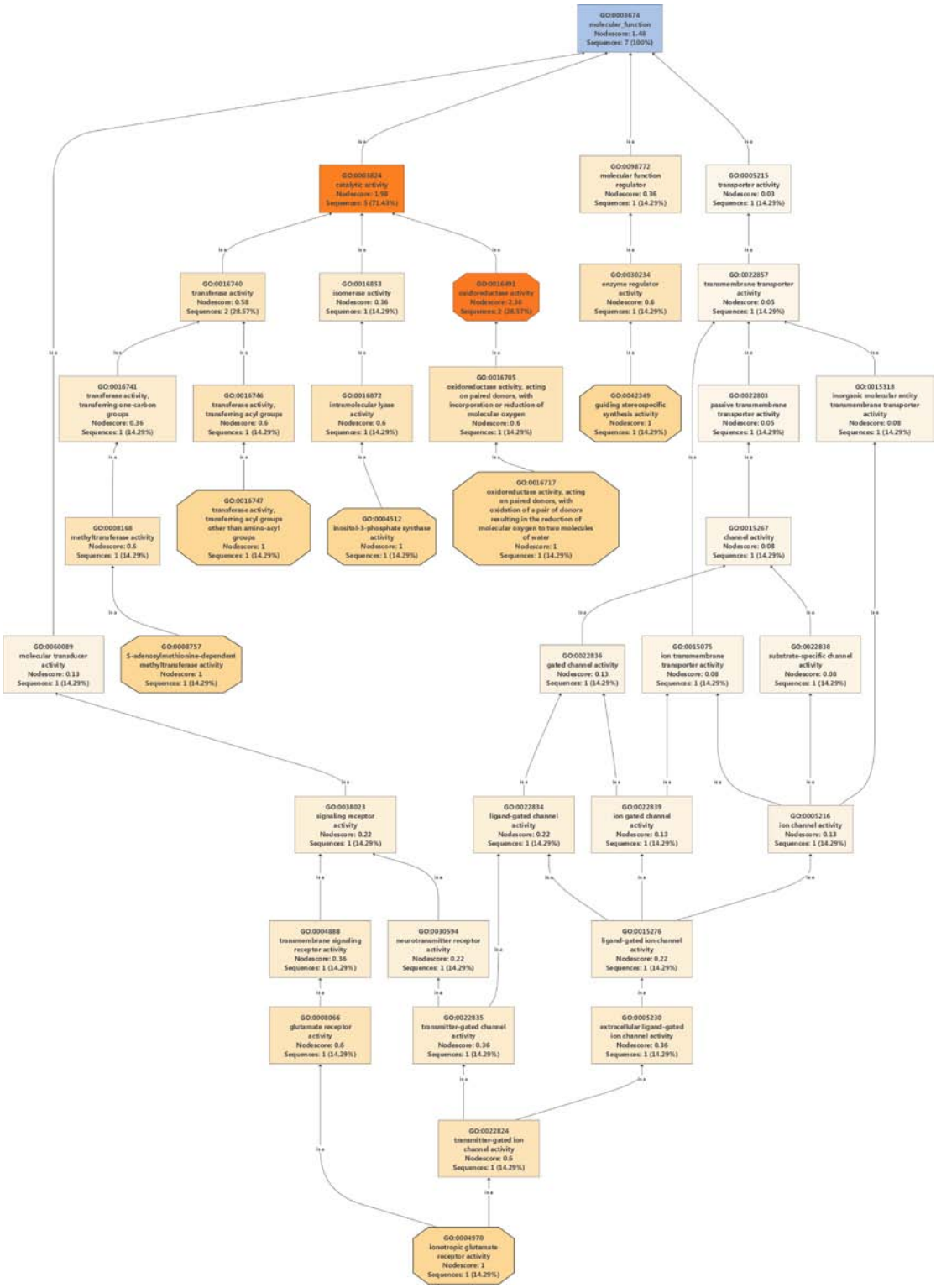
Anexo VII. Pathway Drug metabolism - other enzymes (interacción resistencia:inoculación)



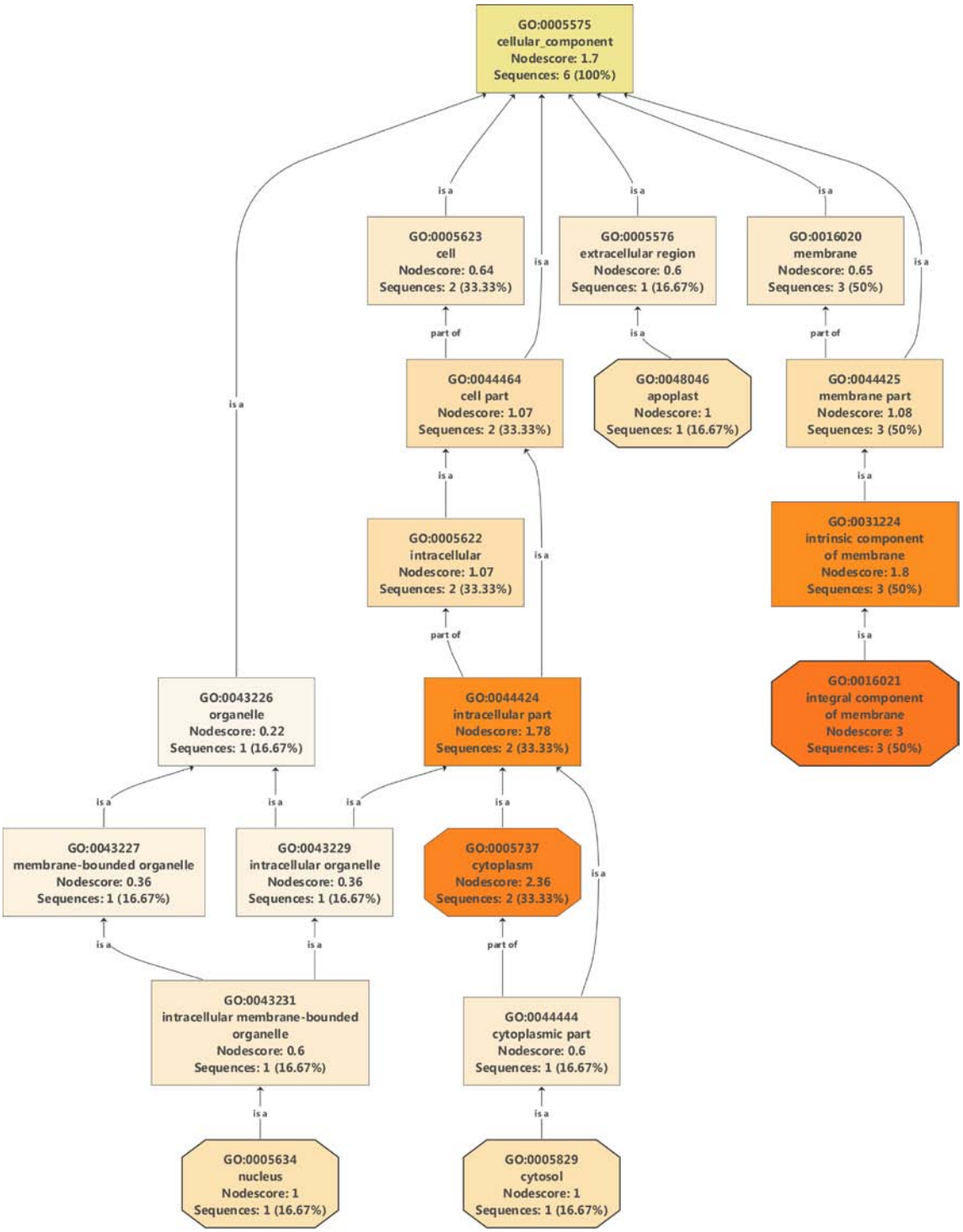
Anexo VIII. Procesos biológicos del análisis funcional (efecto inoculación en clon resistente)



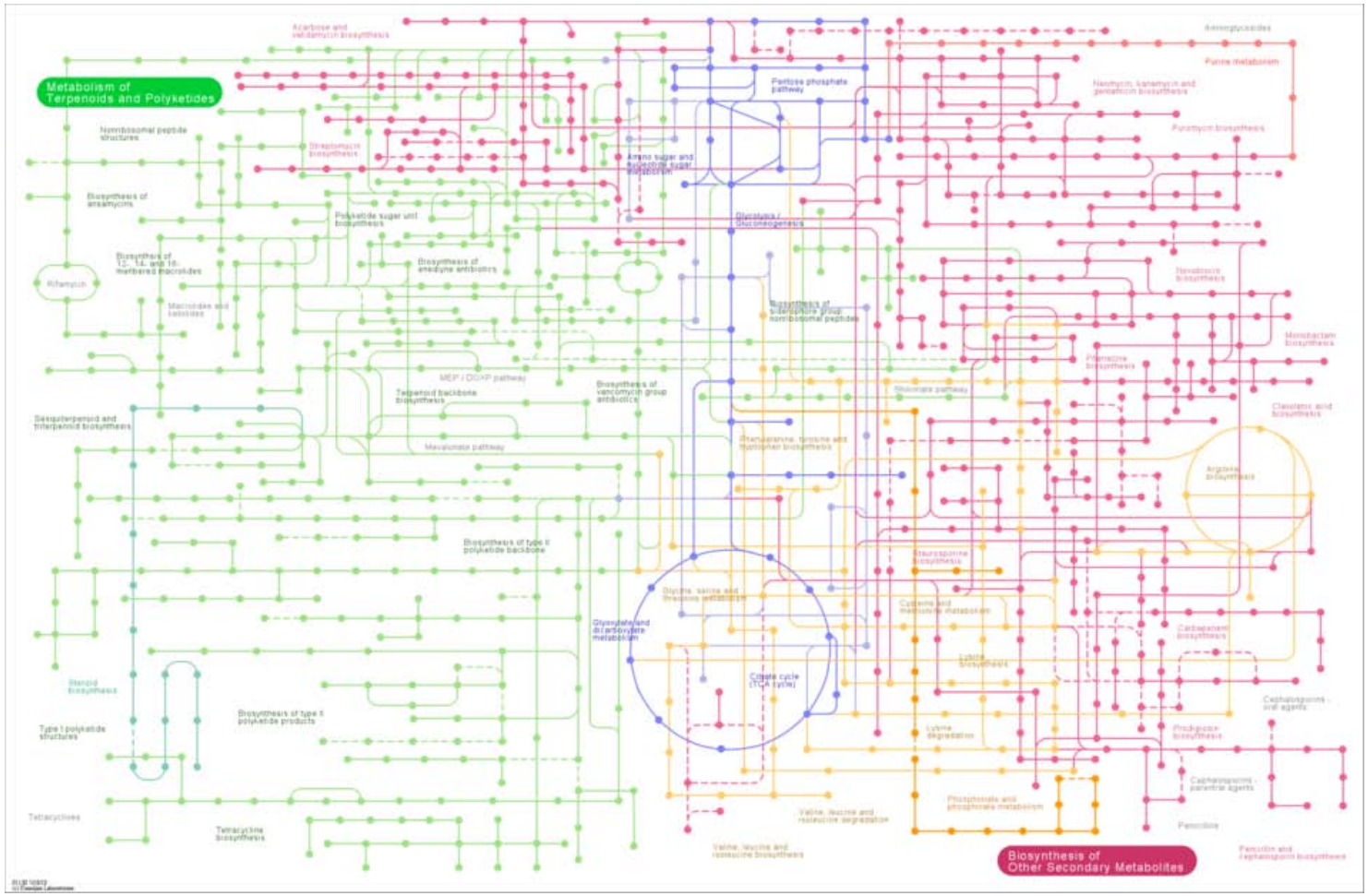
Anexo IX. Funciones moleculares del análisis funcional (efecto inoculación en clon resistente)



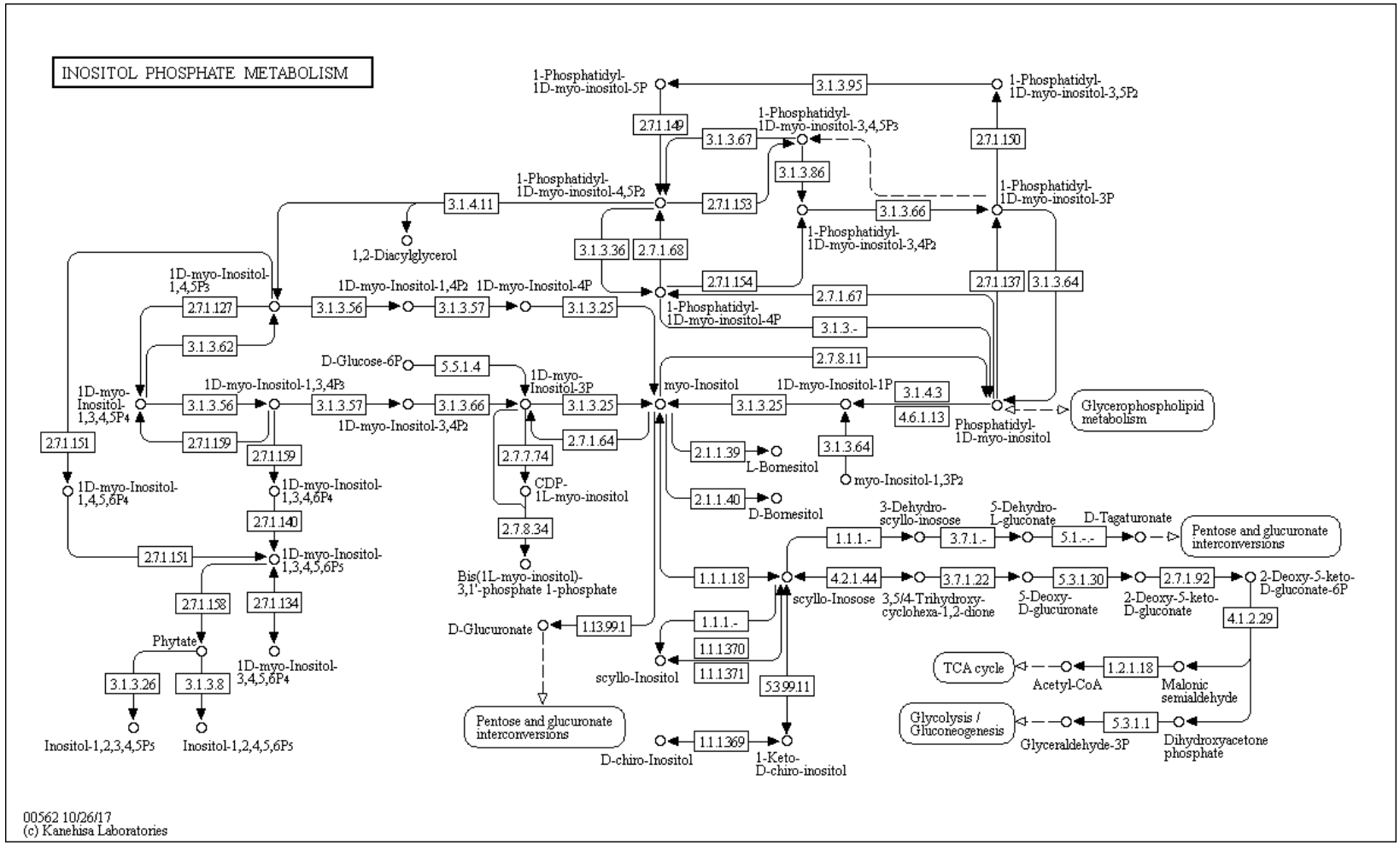
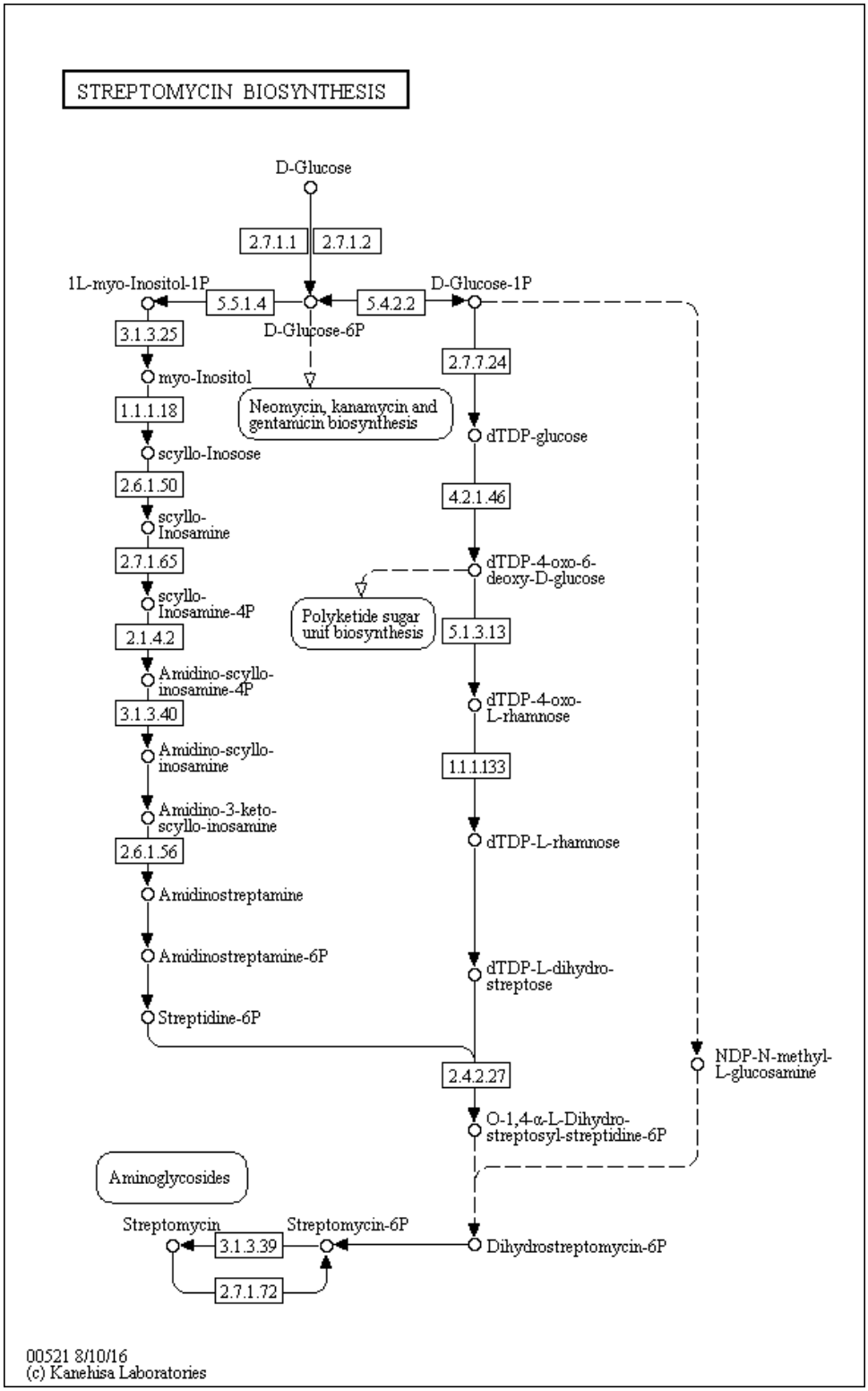
Anexo X. Componentes celulares del análisis funcional (efecto inoculación en clon resistente)



Anexo XI. Pathway Biosynthesis of antibiotics (efecto inoculación en clon resistente)



Anexo XII. Pathway Inositol Phosphate metabolism (efecto inoculación en clon resistente)



Anexo XIII. Pathway Streptomycin biosynthesis (efecto inoculación en clon resistente)