



UNIVERSIDAD PERUANA  
**CAYETANO HEREDIA**

## Bioinformática II

**Proyecto del curso:** Análisis de la sintenia y conservación de rutas metabólicos en organismos fotosintetizadores: desde un alga unicelular a plantas especializadas

**Grupo: 1**

**Integrantes:**

- Armas Torres Valery Leonor
- Cahuana Mamani, Nataly Catherine
- Chalco González Adrián Alexandro
- Alvarao Caballero Johan Isaac

**Profesores:**

- Claudia Machicado
- Felipe Yon

LIMA - PERÚ

2024

## Índice:

<b>Premisa</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>3</b>
<b>3. Materiales y métodos</b>	<b>4</b>
3.1. Parámetros de selección de secuencias	4
3.2. Evaluación de continuidad y integridad de los ensamblajes	5
3.3. Asignar funciones a genes o secuencias de proteínas	6
3.4. Identificación de grupos de genes ortólogos conservados: OrthoFinder	6
3.5. Preparación de Archivos de anotación GFF para MCScanX	7
3.6. Preparación de Secuencias Proteicas Fasta para MCScanX en Bash	11
3.7. Ejecución en MCScanX	15
3.8 Tratamiento de datos MCScanX: Obtención de estadísticas de los bloques sinténicos y análisis de términos GO diferenciales y enriquecidos	17
<b>4. Resultados</b>	<b>25</b>
<b>5. Discusión</b>	<b>29</b>
<b>6. Conclusiones</b>	<b>32</b>
<b>7. Bibliografía</b>	<b>33</b>
<b>8. Anexos</b>	<b>33</b>

## Premisa

A lo largo de la evolución, los organismos fotosintetizadores basados en clorofila han colonizado diferentes ambientes y adaptándose a condiciones diferentes de hábitat, partiendo del agua con algas hasta plantas parasíticas y carnívoras. Se busca identificar la sintenia y sus rearrreglos entre especies representativas de algas, plantas fotosintetizadoras, parasíticas y carnívoras, y así hallar dentro de los bloques conservados cuáles clusters de genes relacionados a rutas metabólicas se mantienen conservados a lo largo del tiempo y evolución. Se sugiere escoger una especie representativa de cada grupo, con genoma pequeño, y delimitar el análisis de los clusters de genes a grupos funcionales para englobar procesos conservados en el tiempo.

# 1. Introducción

La historia evolutiva de las plantas terrestres nació en el mar, dentro del grupo de las algas verdes carófitas (Jianchao Ma, 2022). Este proceso evolutivo involucró no solo la ganancia y la cooptación de genes, sino también la pérdida de otros. Además, existen estudios sobre familias de genes entre especies fotosintéticas, como *Arabidopsis thaliana* y *Chlamydomonas Reinhardtii*. Sin embargo, hasta ahora no se ha comparado con *Cuscuta australis* (planta parásita) y *Roridula gorgonias* (planta carnívora). Las familias de genes conservados en plantas terrestres suelen estar relacionados con la respuesta al estrés, transporte de iones y metabolitos (Jianchao Ma, 2022).

Por ello, en este trabajo evaluará los bloques sintéticos compartidos en estas cuatro especies: tres plantas terrestres (*Arabidopsis thaliana*, *Cuscuta australis* y *Roridula gorgonias*) y un alga unicelular (*Chlamydomonas reinhardtii*).

El objetivo es determinar la sintenia de rutas metabólicas conservadas entre estas especies fotosintetizadoras. Se espera identificar la conservación de algunas rutas metabólicas clave entre los organismos analizados, tales como: el transporte de iones (como K, Fe, Cu, etc.), regulación del ciclo celular, mecanismos de respuesta al estrés abiótico (particularmente relevantes en plantas terrestres), fotosíntesis (posiblemente de manera parcial en *Cuscuta australis*), metabolismos de compuestos secundarios para la adaptación biótica y desarrollo embrionario (Guo, 2012, Sun et al., 2018, Jianchao Ma, 2022 y Fleck et al, 2023).

Para alcanzar estos objetivos, se emplean una serie de herramientas bioinformáticas. Los datos genómicos fueron descargados de bases de datos públicos como NCBI y DRYAD. La calidad de los genomas serán evaluadas con las herramientas *QUAST* y *BUSCO*, disponibles en Galaxy. Los lenguajes de programación como Python, Bash y R se utilizarán para la edición de archivos, el procesamiento y análisis de datos. La anotación e identificación de genes ortólogos conservados se realizará utilizando eggNOG-mapper y OrthoFinder, respectivamente. Para la representación gráfica de los resultados se empleará R, complementado con Excel en casos particulares como la generación de gráficos de barras. La determinación de bloques sintéticos se llevará a cabo con MCScanX y su visualización se gestionará con Synvisio. Finalmente, el enriquecimiento de términos GO en los bloques sintéticos será analizado mediante las plataformas DAVID y PANTHER.

## 2. Objetivos

Objetivo principal: Determinar la sintenia de rutas metabólicas conservadas entre las especies fotosintetizadoras: un alga unicelular, planta modelo, planta parásita y una carnívora.

Objetivos secundarios:

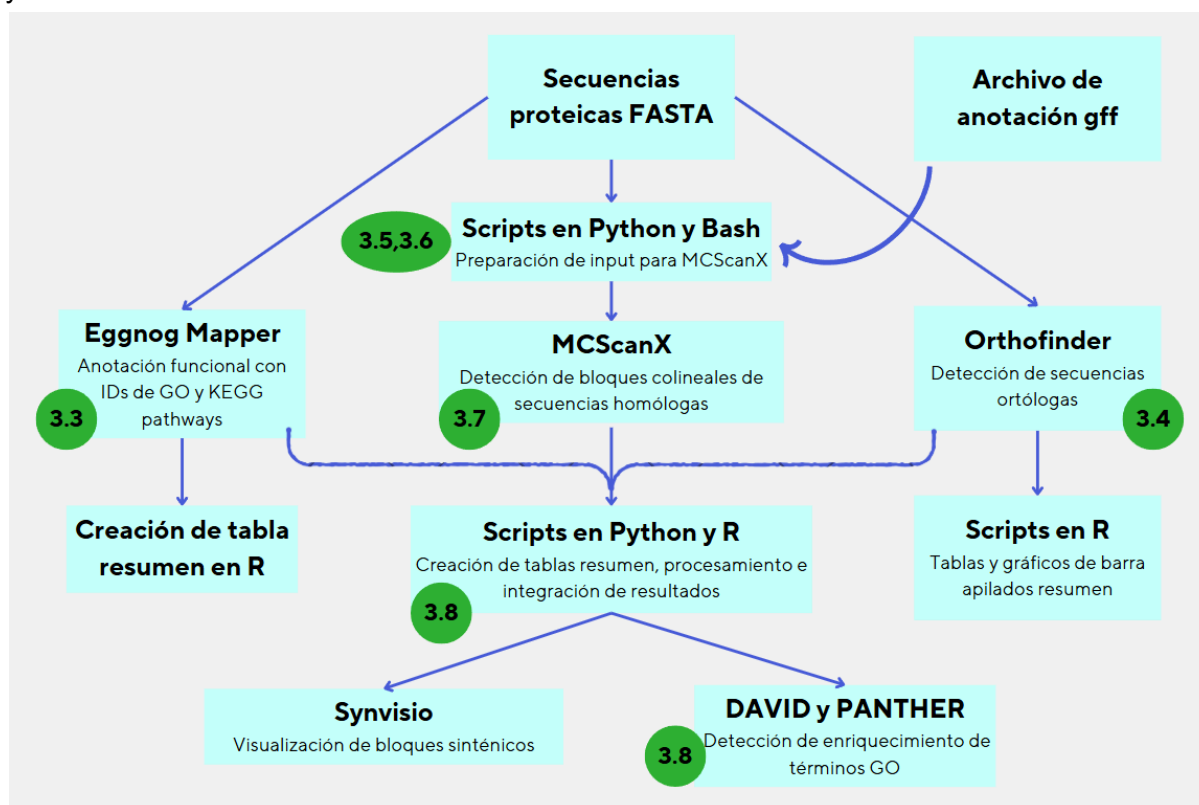
- Criterios de exclusión de genomas
- Identificar genes ortólogos conservados entre las especies seleccionadas, utilizando herramientas como eggNOG-mapper y OrthoFinder.
- Analizar la sintenia de los genes ortólogos de interés en vías metabólicas específicas utilizando MCScanX y Synvisio

### 3. Materiales y métodos

**Nota importante:** A partir de este punto, en el presente trabajo se mencionan varios scripts en python o R, de los cuales solo se toman capturas como muestra general. El/La gentil lector/a puede acceder a cada uno de estos scripts por su nombre en la carpeta “Scripts” del comprimido.

#### Flujograma guía

Los números corresponden a la sección donde se discute el cuadro. No se considera Quast y BUSCO.



#### 3.1. Parámetros de selección de secuencias

Para el proceso de selección de secuencias, se consideraron los siguientes criterios: Organismo fotosintético, disponibilidad del genoma, proteínas y anotaciones en base de datos como NCBI o DRYAD, además, tamaño del genoma entre 100 Mb a 300Mb (pequeño), para un manejo computacional eficiente dado nuestros recursos limitados.

La búsqueda realizada en NCBI, tuvo como filtro organismos fotosintéticos conocidos. Se priorizaron aquellos con genomas ensamblados, acompañados de archivos de secuencia genómica (fasta.), proteínas (fasta.) y anotaciones completas (gff.). Aquellos organismos que cumplieron con los criterios de selección fueron seleccionados para descargar sus secuencias en los siguientes formatos:

- Genoma y proteínas en formato **FASTA** .

- Anotaciones en formato **GFF** .

Tabla 1. Visualización general de los organismos escogidos

	<i>Arabidopsis thaliana</i>	<i>Chlamydomonas reinhardtii</i>	<i>Cuscuta australis</i>	<i>Utricularia gibba</i>
Identificador	GCF_0000001735.4	GCF_000002595.2	GCA_003260385.1	GCA_002189035.1
Fuente	RefSeq	RefSeq	GenBank	GenBank
Documentos descargados	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Secuencia y anotaciones → GBFF  No hay proteínas

En un inicio se consideró utilizar la secuencia de la planta carnívora *Utricularia gibba*, sin embargo, como carecía de secuencia de proteínas y anotaciones, se optó por *Roridula gorgonias*. Aunque se buscaron genomas anotados de *Utricularia gibba* en múltiples fuentes y páginas de archivos suplementarios de artículos, no se encontró la anotación del genoma necesaria.

Entonces, la tabla 1. con los genomas utilizados queda de este modo:

	<i>Arabidopsis thaliana</i>	<i>Chlamydomonas reinhardtii</i>	<i>Cuscuta australis</i>	<i>Roridula gorgonias</i>
Identificador	GCF_0000001735.4	GCF_000002595.2	GCA_003260385.1	_____
Fuente	RefSeq	RefSeq	GenBank	DRYAD
Documentos descargados	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa	Secuencias del genoma (FASTA) → fna  Funciones de anotación → GFF  Proteínas (FASTA)→ faa

## 3.2. Evaluación de continuidad y integridad de los ensamblajes

Se evaluó la calidad de los ensamblajes, para ello se utilizó las herramientas QUAST y BUSCO dentro de Galaxy Europa.

### Quast

1. Selección de secuencias de entrada en formato fna. (fasta) - Input: genomas ensamblados.
2. Selección del tipo de organismos → escoger organismos **Eucariotas**
3. Selección de los archivos de salida Informe HTML y PDF.
4. Los demás parámetros por default.

## Flujo de trabajo del análisis de integridad de ensamblaje con BUSCO

Análisis de proteínas en BUSCO. Se dividió en 2 operaciones esto por los linajes de la especie acuática y las terrestres, el linaje de Chlorophytas(*Chlamydomonas reinhardtii*) y Eudicots(*Arabidopsis thaliana*, *Cuscuta australis*, *Roridula gorgonias*). Se subieron las secuencias de proteínas, en la primera operación se colocó esta configuración:

- Lineage data source: Download lineage data
- Mode: Annotated gene sets(protein)
- Auto-detect or select lineage: Select lineage
- Lineage: Chlorophyta

y en la siguiente se colocó esta configuración:

- Lineage data source: Download lineage data
- Mode: Annotated gene sets(protein)
- Auto-detect or select lineage: Select lineage
- Lineage: Eudicots

Nos arroja varios resultados, se resalta una tabla para ver los genes fragmentados, completos y desaparecidos, en excel se recopila y se armará un gráfico de barras.

## 3.3. Asignar funciones a genes o secuencias de proteínas

Para la asignación de funciones a las secuencias proteicas de cada especie se utilizó la herramienta “**EggNOG Mapper** functional sequence annotation by orthology”, contenida en Galaxy Europa. Aquí se subieron los archivos de proteínas en formato faa.

1. Parámetros de trabajo
  1. Input: la secuencia de proteínas en formato faa.
  2. Tipo de secuencia: proteínas
  3. Matriz de puntuación y costes de brecha: BLOSUM62
  4. Base para la anotación: Diamond
  5. Modo de sensibilidad diamond: Sensible
  6. Los demás parámetros por default.
2. Archivos resultantes (Output)
  1. seed\_orthologs (no se utilizó)
  2. anotaciones: de este archivo se extrajeron los identificadores de términos GO y pathways del KEGG mencionados más adelante

## 3.4. Identificación de grupos de genes ortólogos conservados: OrthoFinder

*OrthoFinder* es una herramienta para identificar genes ortólogos (y parálogos) entre diferentes genomas. Su github está en <https://github.com/davidemms/OrthoFinder>

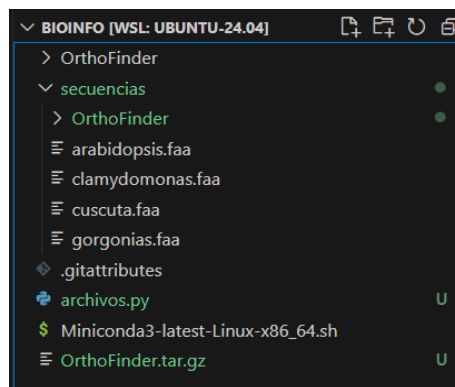
- **Instalación de OrthoFinder:** Clona el repositorio desde GitHub e instala las dependencias.

```

(base) valery0905@Valery:~/repo/bioinfo$ wget https://github.com/davideemms/OrthoFinder/releases/latest/download/OrthoFinder.tar.gz
(base) valery0905@Valery:~/repo/bioinfo$ tar xzvf OrthoFinder.tar.gz
(base) valery0905@Valery:~/repo/bioinfo$ cd OrthoFinder/
(base) valery0905@Valery:~/repo/bioinfo$ ./orthofinder -h

```

- **Preparación de los datos:**
  - Reúne los archivos de secuencias proteicas en formato **FASTA** para cada genoma.
  - Coloca los archivos en una carpeta y asegúrate de que estén nombrados de forma clara.



- **Ejecución de OrthoFinder:** Ejecuta OrthoFinder apuntando al directorio donde tienes los archivos FASTA.

```

(base) valery0905@Valery:~/repo/bioinfo$ ./orthofinder -f /home/valery0905/repo/bioinfo/secuencias

```

4. **Revisión de los resultados:** OrthoFinder crea una carpeta de resultados (por ejemplo, **Results\_date**) que contiene archivos clave:

- **Orthogroups.csv:** Lista los ortogrupos con los genes de cada genoma.
- **Orthologues:** Relación de ortólogos entre genomas.
- **Gene\_Trees:** Árboles filogenéticos de los genes.
- **Visualización de los resultados:** Uso del script **figures.RMD** en R para los resultados colocados en las figuras X

```

SpeciesTree_rooted.txt
secuencias > OrthoFinder > Results_Nov24 > Species_Tree > SpeciesTree_rooted.txt
1 (clamydomonas:0.541445,(arabidopsis:0.214068,(gorgonias:0.224884,cuscutea:0.227796)0.551674:0.0713023)1:0.541445);

```

## 3.5. Preparación de Archivos de anotación GFF para MCSanX

(para obtener all\_new.gff)

Para correr MCSanX se necesitan dos archivos: primero, el resultado tabular de la realización de un blastp all vs all las secuencias proteicas (todas contra todas). Segundo, un archivo gff (que realmente es un bed) único, correspondiente a todas estas secuencias. Cada secuencia proteica corresponde a un ARNm en el genoma y archivo gff de cada especie. Cabe resaltar que un gen puede tener múltiples ARNm, y, por ende, múltiples secuencias proteicas. Nosotros hemos trabajado a partir de ARNm.

1. Primero, se creó y ejecutó el script gff\_to\_bed de abajo (el nombre del script se cambió, antes era edit\_gff), que extrae sólo los mRNA del gff y los lleva a mi nuevo archivo bed y, además, añade prefijo de especie (AT, CR, CU o RO) a la identificador de cada cromosoma en la columna de los cromosomas. Esto último para que se pueda visualizar correctamente al hacer el análisis de sintenia, e identificar de qué especie viene cada cromosoma. El script inferior consta de dos funciones (cada una inicia en def), una para *Roridula gorgonias*, y las otras tres para el resto de las especies. La razón es que la estructura del gff de *Roridula* era distinta a las de las otras tres especies porque no provenía del Refseq del NCBI. El script tiene 3 inputs, el input o gff original, el output en archivo bed, y, por último, el prefijo, que son dos letras mayúsculas que varían entre cada especie.

```
CHALCO > eproceso > edit_gff.py > ...
5 def gff_to_bed(gff_file, bed_file, species_prefix):
6     #Función del script: Extraer mRNAs con siempre y cuando tengan una CDS correspondiente
7     import sys
8
9
10    def gff_to_bed(gff_file, bed_file, species_prefix):
11        # Diccionarios para almacenar información
12        mrna_info = {} # Almacena información de mRNA: mRNA_ID -> {'chrom', 'start', 'end'}
13        mrna_to_protein = {} # Mapea mRNA_ID a protein_id
14
15        with open(gff_file, 'r') as gff:
16            for line in gff:
17                if line.startswith('#'):
18                    continue # Ignorar comentarios
19                fields = line.strip().split('\t')
20                if len(fields) < 9:
21                    continue # Saltar líneas mal formateadas
22                chrom = fields[0]
23                source = fields[1]
24                feature_type = fields[2]
25                start = int(fields[3]) - 1 # Convertir a 0-based
26                end = fields[4]
27                strand = fields[6]
28                attributes = fields[8]
29
30                # Procesar atributos en un diccionario
31                attr_dict = {}
32                for attr in attributes.split(';'):
33                    key, value = attr.split('=')
34                    attr_dict[key] = value
35
36                # Extraer mRNA_ID de los atributos
37                mrna_id = attr_dict.get('ID', '')
38                # Verificar si el mRNA_ID está en el diccionario mrna_info
39                if mrna_id in mrna_info:
40                    # Si está, actualizar la información con los datos de la línea actual
41                    mrna_info[mrna_id]['chrom'] = chrom
42                    mrna_info[mrna_id]['start'] = start
43                    mrna_info[mrna_id]['end'] = end
44                    mrna_info[mrna_id]['strand'] = strand
45                    mrna_info[mrna_id]['feature_type'] = feature_type
46
47                # Si no está, crear una nueva entrada en el diccionario
48                else:
49                    mrna_info[mrna_id] = {'chrom': chrom, 'start': start, 'end': end, 'strand': strand, 'feature_type': feature_type}
50
51                # Mapear mRNA_ID a protein_id
52                protein_id = attr_dict.get('protein_id', '')
53                mrna_to_protein[mrna_id] = protein_id
54
55                # Escribir la información en el archivo bed
56                # Formato: chrom, start, end, score, strand, feature_type, mrna_id, protein_id
57                bed_line = f'{chrom}\t{start}\t{end}\t0\t{strand}\t{feature_type}\t{mrna_id}\t{protein_id}\n'
58                with open(bed_file, 'a') as bed:
59                    bed.write(bed_line)
60
61    # Ejecutar la función gff_to_bed
62    gff_to_bed(gff_file, bed_file, species_prefix)
```

#Ejecución del script en la terminal

```
python3 gff_to_bed.py arabidopsis.gff arabidopsis.bed AT
python3 gff_to_bed.py chlamydomonas.gff chlamydomonas.bed CR
python3 gff_to_bed.py cuscuta.gff cuscuta.bed CU
python3 gff_to_bed.py roridula.gff roridula.bed RG
```



2. Con los comandos en linux mostrados se confirmó que el número de features en cada bed de cada especie era igual que el número de proteínas en el archivo proteico fasta de cada especie

```
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$ grep -c '^>' ../proteins/arabidopsis_protein.faa
48265
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$ wc -l arabidopsis.bed
48265 arabidopsis.bed

(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$ wc -l roridula.bed
22655 roridula.bed
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$ grep -c '^>' ../proteins/roridula_protein.faa
22655
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$ grep -c '^>' ../proteins/roridula_protein.faa
22655
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/eprocso/bed$
```

3. Se concatenaron (o unieron) todos los bed de las especies en un solo bed llamado all.bed
4. Para poder mejorar la distinción de genes al entender el futuro output de MCScanX y manejar mejor Synvisio, se creó el siguiente script en bash para editar el archivo all.bed. Lo que se hizo fue añadir los prefijos de especie AT, CR, CU y RO a los identificadores de cada secuencia proteica en el BED, para que también las proteínas, por ejemplo en *Arabidopsis*, tengan AT al inicio, y no solo los cromosomas

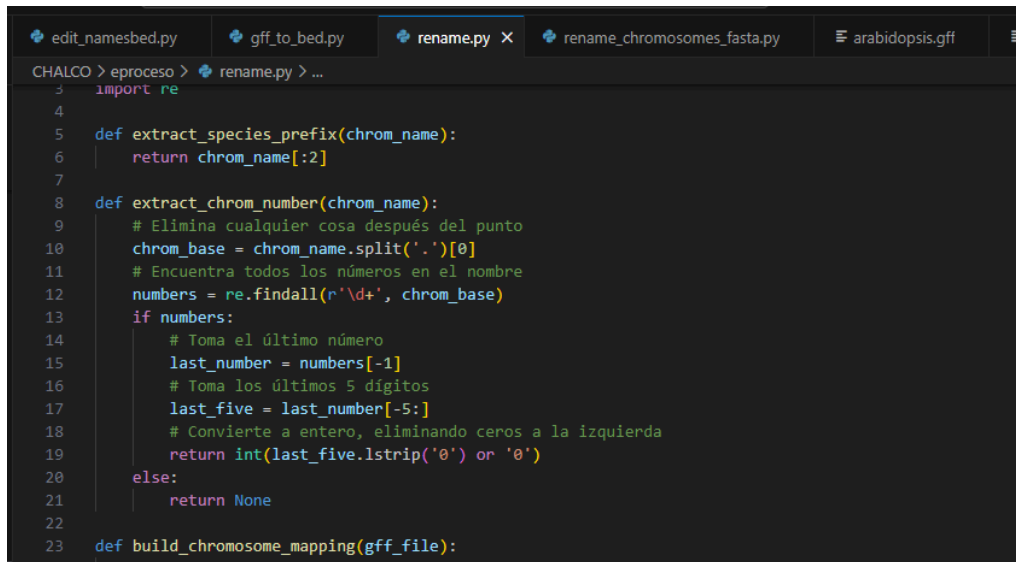
#Creación del script

```
cat << 'EOF' > add_prefixes_bed.awk
BEGIN { OFS = "\t" } { prefix = substr($1, 1, 2) $4 = prefix $4 print $0
} EOF
```

#Ejecucion del script

```
awk -f add_prefixes_bed.awk all.bed > all_prefixed.bed
#Luego de añadir los prefijos a los genes, reordeno las columnas
awk '{print $1, $4, $2, $3}' OFS='\t' all_prefixed.bed > all.gff
```

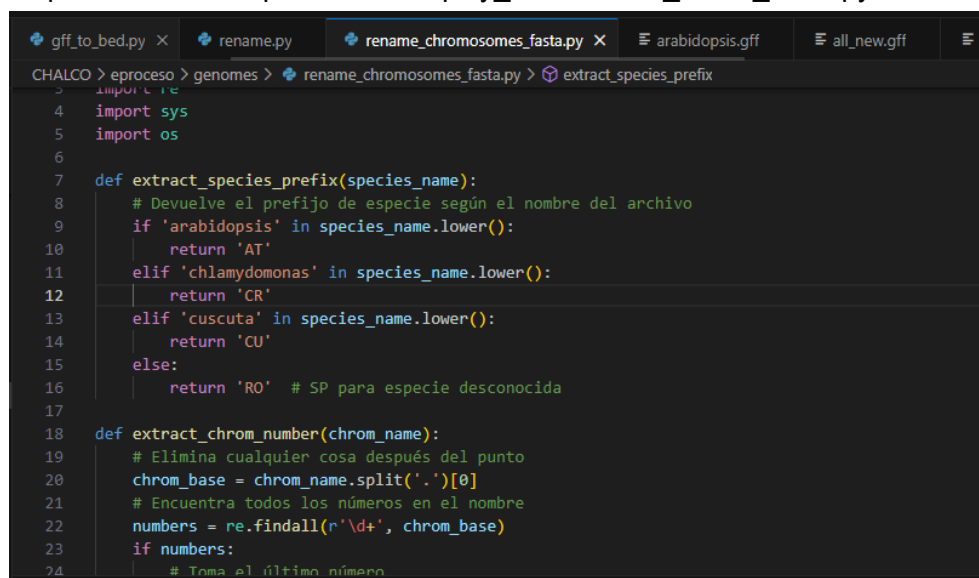
5. Esto terminaría con el proceso, pero al ejecutar MCScanX es mejor acortar el nombre de cada cromosoma, por ejemplo, de ATNC0009.1 a AT1. Para ello utilicé el script en python simplify\_chromosome\_name\_gff.py para renombrar el nombre de todos los cromosomas en all.gff y simplificarlo.



```
edit_namesbed.py  gff_to_bed.py  rename.py X  rename_chromosomes_fasta.py  arabidopsis.gff  E
CHALCO > eproceso > rename.py ...
3 import re
4
5 def extract_species_prefix(chrom_name):
6     return chrom_name[:2]
7
8 def extract_chrom_number(chrom_name):
9     # Elimina cualquier cosa después del punto
10    chrom_base = chrom_name.split('.')[0]
11    # Encuentra todos los números en el nombre
12    numbers = re.findall(r'\d+', chrom_base)
13    if numbers:
14        # Toma el último número
15        last_number = numbers[-1]
16        # Toma los últimos 5 dígitos
17        last_five = last_number[-5:]
18        # Convierte a entero, eliminando ceros a la izquierda
19        return int(last_five.lstrip('0') or '0')
20    else:
21        return None
22
23 def build_chromosome_mapping(gff_file):
24     species_chrom_numbers = {}
```

### PASO EXTRA:

- También se simplificaron los nombres de los cromosomas en los fasta del genoma completo de cada especie con `simplify_chromosome_name_fasta.py`



```
gff_to_bed.py X  rename.py  rename_chromosomes_fasta.py X  arabidopsis.gff  all_new.gff  E
CHALCO > eproceso > genomes > rename_chromosomes_fasta.py > extract_species_prefix
3 import re
4 import sys
5 import os
6
7 def extract_species_prefix(species_name):
8     # Devuelve el prefijo de especie según el nombre del archivo
9     if 'arabidopsis' in species_name.lower():
10        return 'AT'
11    elif 'chlamydomonas' in species_name.lower():
12        return 'CR'
13    elif 'cuscuta' in species_name.lower():
14        return 'CU'
15    else:
16        return 'RO' # SP para especie desconocida
17
18 def extract_chrom_number(chrom_name):
19     # Elimina cualquier cosa después del punto
20    chrom_base = chrom_name.split('.')[0]
21    # Encuentra todos los números en el nombre
22    numbers = re.findall(r'\d+', chrom_base)
23    if numbers:
24        # Toma el último número
```

- Posterior a ello utilizó el script `get_length_chromosomes_fasta.py` para calcular las longitudes de cada cromosoma de cada especie y visualizar los genomas de mayor tamaño con Synvisio (ya que la mayoría son cromosomas pequeños). Además, de poder ver dónde podrían estar los mayores bloques de sintenia

```

CHALCO > eproceso > genomes > get_length_chromosomes_fasta.py > calcular_longitudes
10 def calcular_longitudes(fasta_path, output_path):
30     secuencia = ''
31
32     for linea in f:
33         linea = linea.strip()
34         if linea.startswith('>'): # New sequence header
35             if nombre_secuencia:
36                 secuencias[nombre_secuencia] = len(secuencia)
37             # Extract the sequence name (identifier only, no extra info)
38             nombre_secuencia = linea[1:].split()[0]
39             secuencia = ''
40         else:
41             secuencia += linea
42
43     # Store the last sequence
44     if nombre_secuencia:
45         secuencias[nombre_secuencia] = len(secuencia)
46
47     # Sort sequences by length in descending order
48     secuencias_ordenadas = sorted(secuencias.items(), key=lambda x: x[1], reverse=True)

```

- El output del script se ve así, donde los cromosomas se encuentran ordenados de mayor a menor tamaño:

```

CHALCO > eproceso > genomes > gorgonias_lengths.tsv > data
1  Contig  Longitud
2  R018683 191047
3  R018619 157010
4  R018400 147796
5  R018538 139688
6  R019179 137243
7  R019186 135246
8  R019298 135229
9  R018362 134847
10 R018680 130525
11 R019316 130372
12 R018322 129125
13 R018639 128845
14 R018677 128545
15 R018463 125426
16 R017590 124691
17 R019085 124442
18 R018109 123763

```

### 3.6. Preparación de Secuencias Proteicas Fasta para MCSScanX en Bash

1. Primero se eliminaron las demás palabras del header y mantengo solo la primera palabra (accesión del header)

```

sed 's/^\(>[^]*\)*/\1/' arabidopsis_protein.faa >
arabidopsis_protein_edited.faa
sed 's/^\(>[^]*\)*/\1/' chlamydomonas_protein.faa >
chlamydomonas_protein_edited.faa
sed 's/^\(>[^]*\)*/\1/' cuscuta_protein.faa >
cuscuta_protein_edited.faa
sed 's/^\(>[^]*\)*/\1/' roridula_protein.faa >

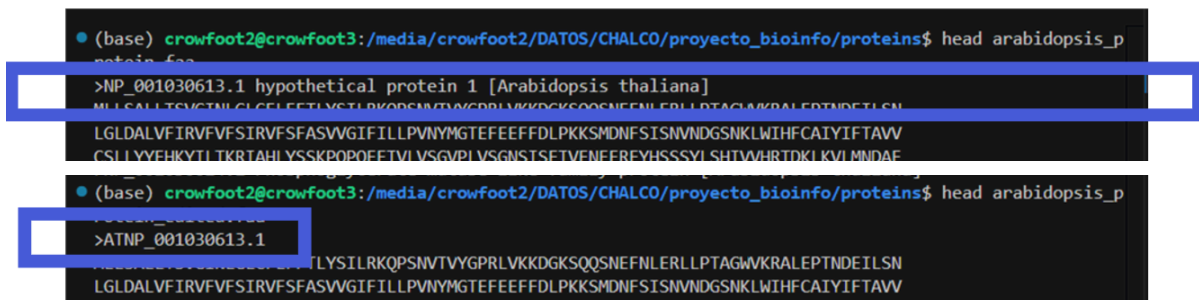
```

roridula\_protein\_edited.faa

2. Añadió el prefijo correspondiente a la accesión de cada header (de cada secuencia proteica) del archivo de proteínas fasta

```
sed -i 's/^>\(.*\)/>AT\1/' arabidopsis_protein_edited.faa
sed -i 's/^>\(.*\)/>CR\1/' chlamydomonas_protein_edited.faa
sed -i 's/^>\(.*\)/>CU\1/' cuscuta_protein_edited.faa
sed -i 's/^>\(.*\)/>RO\1/' roridula_protein_edited.faa
```

Ejemplo del cambio en el header de Arabidopsis thaliana, antes y después de la ejecución de los comandos previos (1 y 2).



```
(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/proyecto_bioinfo/proteins$ head arabidopsis_p
tein.faa
>NP_001030613.1 hypothetical protein 1 [Arabidopsis thaliana]
MLLSALLTGVGTLGLCELEETLVSTLRKQPCINQWCPDLVKKDGKQQQNEFNLERLLPTAGWVKRALEPTNDEILSN
LGLDALVFIRVFVSIRVFSFASVWGIFILLPVNYMGTEFEFFDLPKKSMDNFSISNVNDGSNKLWIHFCAIYIFTAVV
CSLIYYEHKVTITKRTAHLVSSKPOPOEETVLVSGVPLVSGNSTSETVNEFEYHSSSYLSHTVWVRTDKLKVLMNDAE

(base) crowfoot2@crowfoot3:/media/crowfoot2/DATOS/CHALCO/proyecto_bioinfo/proteins$ head arabidopsis_p
tein.faa
>ATNP_001030613.1
MLLSALLTGVGTLGLCELEETLVSTLRKQPCINQWCPDLVKKDGKQQQNEFNLERLLPTAGWVKRALEPTNDEILSN
LGLDALVFIRVFVSIRVFSFASVWGIFILLPVNYMGTEFEFFDLPKKSMDNFSISNVNDGSNKLWIHFCAIYIFTAVV
CSLIYYEHKVTITKRTAHLVSSKPOPOEETVLVSGVPLVSGNSTSETVNEFEYHSSSYLSHTVWVRTDKLKVLMNDAE
```

3. A continuación, se creó la base de datos de cada especie para hacer el BLASTP múltiple intra e interespecies (como se mencionó, MCScanX requiere como input el resultado de un BLAST):

```
makeblastdb -in arabidopsis_protein_edited.faa -dbtype prot -out arabidopsis_db
makeblastdb -in chlamydomonas_protein_edited.faa -dbtype prot -out chlamydomonas_db
makeblastdb -in cuscuta_protein_edited.faa -dbtype prot -out cuscuta_db
makeblastdb -in roridula_protein_edited.faa -dbtype prot -out roridula_db
```

4. Ejecución del BLASTP intra y interespecífico (entre las secuencias de una misma especie y entre las secuencias de diferentes especies)

En los comandos de abajo, el parámetro -outfmt 6 indica que la salida estará en formato tabular, lo que es requerido por MCScanX. Este formato contiene columnas predefinidas como las identidades, valores de e-valor y posiciones de inicio/fin de los alineamientos. Por su parte, el parámetro -max\_target\_seqs 5 limita el número máximo de secuencias objetivo (hits) reportadas por consulta, mostrando únicamente las 5 mejores coincidencias según el puntaje. Esto optimiza la salida al enfocarse en los hits más relevantes y también se recomienda al correr MCScanX

#Arabidopsis vs Arabidopsis

```
blastp -query arabidopsis_protein_edited.faa -db arabidopsis_db -evaluate  
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
arabidopsis_vs_arabidopsis.blast
```

#### # Arabidopsis vs Chlamydomonas

```
blastp -query arabidopsis_protein_edited.faa -db chlamydomonas_db  
-evaluate 1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
arabidopsis_vs_chlamydomonas.blast
```

#### # Arabidopsis vs Cuscuta

```
blastp -query arabidopsis_protein_edited.faa -db cuscuta_db -evaluate 1e-5  
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
arabidopsis_vs_cuscuta.blast
```

#### # Arabidopsis vs Roridula

```
blastp -query arabidopsis_protein_edited.faa -db roridula_db -evaluate  
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
arabidopsis_vs_roridula.blast
```

#### # Chlamydomonas vs Arabidopsis

```
blastp -query chlamydomonas_protein_edited.faa -db arabidopsis_db  
-evaluate 1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
chlamydomonas_vs_arabidopsis.blast
```

#### # Chlamydomonas vs Chlamydomonas

```
blastp -query chlamydomonas_protein_edited.faa -db chlamydomonas_db  
-evaluate 1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
chlamydomonas_vs_chlamydomonas.blast
```

#### # Chlamydomonas vs Cuscuta

```
blastp -query chlamydomonas_protein_edited.faa -db cuscuta_db -evaluate  
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
chlamydomonas_vs_cuscuta.blast
```

#### # Chlamydomonas vs Roridula

```
blastp -query chlamydomonas_protein_edited.faa -db roridula_db -evaluate  
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
chlamydomonas_vs_roridula.blast
```

#### # Cuscuta vs Arabidopsis

```
blastp -query cuscuta_protein_edited.faa -db arabidopsis_db -evaluate 1e-5  
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out  
cuscuta_vs_arabidopsis.blast
```

#### # Cuscuta vs Chlamydomonas

```
blastp -query cuscuta_protein_edited.faa -db chlamydomonas_db -evaluate
```

```
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out
cuscuta_vs_chlamydomonas.blast
```

#### # Cuscuta vs Cuscuta

```
blastp -query cuscuta_protein_edited.faa -db cuscuta_db -evaluate 1e-5
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out
cuscuta_vs_cuscuta.blast
```

#### # Cuscuta vs Roridula

```
blastp -query cuscuta_protein_edited.faa -db roridula_db -evaluate 1e-5
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out
cuscuta_vs_roridula.blast
```

#### # Roridula vs Arabidopsis

```
blastp -query roridula_protein_edited.faa -db arabidopsis_db -evaluate
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out
roridula_vs_arabidopsis.blast
```

#### # Roridula vs Chlamydomonas

```
blastp -query roridula_protein_edited.faa -db chlamydomonas_db -evaluate
1e-5 -outfmt 6 -num_threads 28 -max_target_seqs 5 -out
roridula_vs_chlamydomonas.blast
```

#### # Roridula vs Cuscuta

```
blastp -query roridula_protein_edited.faa -db cuscuta_db -evaluate 1e-5
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out
roridula_vs_cuscuta.blast
```

#### # Roridula vs Roridula

```
blastp -query roridula_protein_edited.faa -db roridula_db -evaluate 1e-5
-outfmt 6 -num_threads 28 -max_target_seqs 5 -out
roridula_vs_roridula.blast
```

5. Se realizó la concatenación de los resultados de todos los BLASTP

```
cat arabidopsis_vs_arabidopsis.blast \
arabidopsis_vs_chlamydomonas.blast \ arabidopsis_vs_cuscuta.blast \
arabidopsis_vs_roridula.blast \ chlamydomonas_vs_arabidopsis.blast \
chlamydomonas_vs_chlamydomonas.blast \ chlamydomonas_vs_cuscuta.blast \
chlamydomonas_vs_roridula.blast \ cuscuta_vs_arabidopsis.blast \
cuscuta_vs_chlamydomonas.blast \ cuscuta_vs_cuscuta.blast \
cuscuta_vs_roridula.blast \ roridula_vs_arabidopsis.blast \
roridula_vs_chlamydomonas.blast \ roridula_vs_cuscuta.blast \
roridula_vs_roridula.blast \ > all_vs_all.blast
```

### 3.7. Ejecución en MCScanX

El programa MCScanX primero detecta los genes homólogos con el blastp dado como input, y luego detecta los genes homólogos que están juntos, en un rango especificado, para construir bloques sinténicos colineales.

Input: all.blast y all.gff

1. Primero se corrió con estos parámetros por default

```
./MCScanX /media/crowfoot2/DATOS/CHALCO/eprocso/mcscanx_analysis/all -s  
5 -m 25 -w 5
```

2. Luego se corrió con los parámetros con valores más relajados, para permitir obtener más bloques sinténicos colineales entre las especies, por ser lejanas

```
./MCScanX  
/media/crowfoot2/DATOS/CHALCO/eprocso/mcscanx_analysis_relaxed/all_new -s  
3 -m 50 -w 10
```

#### Explicación de los parámetros

-w OVERLAP\_WINDOW: Distancia máxima (en número de genes) entre genes BLAST coincidentes para ser colapsados como parte del mismo bloque. En la “segunda” corrida relajada (paso 2) este parámetro se incrementó para permitir generar bloques sinténicos incluso entre genes homólogos BLAST relativamente lejos (hasta 10 genes distantes).

-m MAX\_GAPS: Máximo número de brechas (genes ausentes o sin homólogos) permitidas dentro de un bloque sinténico. En la corrida relajada del paso 2 este parámetro se incrementó a 50 para permitir que el bloque sinténico se mantenga incluso si hay hasta 50 genes que no son homólogos.

-s MATCH\_SIZE: Número mínimo de genes requeridos para considerar un bloque como sinténico. El valor de este parámetro se disminuyó a 3 en la segunda corrida, para permitir llamar bloques de menor tamaño (hasta de mínimo 3 genes).

Es importante resaltar que aunque se trabajó con ambas formas, tanto con parámetros más estrictos por default (paso 1), como con relajados (paso 2), tras la inspección de la data realizada en “Tratamiento de datos MCScanX”, se decidió usar los parámetros estrictos por default, porque se conseguían bloques sinténicos consistentes con funciones (de términos GO) similares.

```
(base) crowfoot2@crowfoot3:~/MCScanX-master$ ./MCScanX_h  
[Usage] ./MCScanX_h prefix_fn [options]  
-k MATCH_SCORE, final score=MATCH_SCORE+NUM_GAPS*GAP_PENALTY  
  (default: 50)  
-g GAP_PENALTY, gap penalty (default: -1)  
-s MATCH_SIZE, number of genes required to call collinear blocks  
  (default: 5)  
-e E_VALUE, alignment significance (default: 1e-05)  
-m MAX_GAPS, maximum gaps allowed (default: 25)  
-w OVERLAP_WINDOW, maximum distance (# of genes) to collapse BLAST matches (default: 5)  
-a only builds the pairwise blocks (.collinearity file)  
-b patterns of collinear blocks. 0:intra- and inter-species (default); 1:intra-species; 2:inter-species  
-c whether to consider homology scores. 0:not consider (default); 1: lower preferred; 2: higher preferred  
-h print this help page
```

**Nota:** Acá el uso de 2 pasos, no ilustra que solo se corrió dos veces el programa. Realmente MCScanX se ejecutó múltiples veces, según se afinaba el input gff y el input de blast. Se tuvieron también problemas similares a los mencionados en el siguiente “Issue” de su página de github (con el orden de las columnas del bed o gff entrante):

<https://github.com/wyp1125/MCScanX/issues/53>

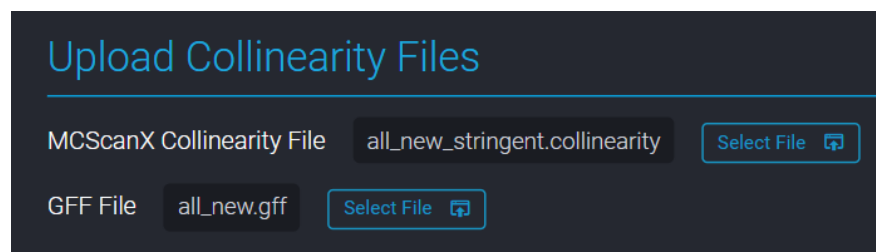
En general, la documentación del input de MCScanX necesita mejorar.

3. Visualización de output de MCScanX, archivo all.collinearity:

```
CHALCO > eproceso > mcscanx_analysis_stringent > all_new.collinearity > data
1 ##### Parameters #####
2 # MATCH_SCORE: 50
3 # MATCH_SIZE: 5
4 # GAP_PENALTY: -1
5 # OVERLAP_WINDOW: 5
6 # E_VALUE: 1e-05
7 # MAX GAPS: 25
8 ##### Statistics #####
9 # Number of collinear genes: 17297, Percentage: 15.93
10 # Number of all genes: 108604
11 #####
12 ## Alignment 0: score=300.0 e_value=2.6e-10 N=6 AT1&AT1 minus
13 0- 0: ATNP_051099.1 ATNP_051123.1 0
14 0- 1: ATNP_051100.1 ATNP_051122.1 8e-66
15 0- 2: ATNP_051101.1 ATNP_051121.1 0
16 0- 3: ATNP_051103.2 ATNP_051119.2 0
17 0- 4: ATNP_051104.1 ATNP_051118.1 1e-112
18 0- 5: ATNP_051105.1 ATNP_051117.1 0
19 ## Alignment 1: score=4136.0 e_value=0 N=92 AT2&AT2 plus
20 1- 0: ATNP_001321164.1 ATNP_177524.2 5e-78
21 1- 1: ATNP_001322884.1 ATNP_177527.3 0
22 1- 2: ATNP_564051.1 ATNP_001323057.1 3e-76
```

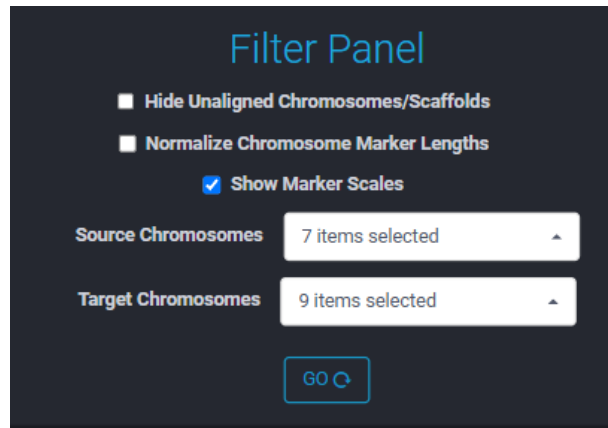
Nótese que cada línea “## Alignment X” define un bloque sinténico. Los primeros bloques son entre la misma especie (fíjese el amable lector en los prefijos de los genes comparados en ambas columnas, ambos son AT, por lo que los genes comparados en estos bloques sinténicos son de *Arabidopsis*)

- Este archivo all.collinearity, junto con el gff aceptado por MCScanX, sirvieron como input para Synvisio. La dirección web de Synvisio es: <https://synvisio.github.io/#/>.



- Se colocaron los cromosomas para visualizar:





- Y de ese modo se produjeron las figuras 6, 11 y 12.

### 3.8 Tratamiento de datos MCSanX: Obtención de estadísticas de los bloques sinténicos y análisis de términos GO diferenciales y enriquecidos

1. Primero, se quisieron obtener estadísticas generales sobre los bloques sinténicos. Como estamos interesados en los bloques entre especies (no dentro de una misma especie), se obtuvo el número de genes que participan en bloques interespecies de cada cromosoma, para saber qué cromosomas aportan más a la conservación interespecie. Además, se obtuvo el número de parejas cromosomales interespecie con mayor número de bloques sinténicos y de genes. Todo lo mencionado se obtuvo procesando los resultados del archivo `all.collinearity` con el script en python `"summarise_statistics_collinearity_file.py"`:

```

18 import os
19 from collections import defaultdict
20
21 def process_collinearity(file_path, bed_file, output_file, interaction_output_file):
22     """ ...
23     # Map genes to their respective chromosomes using the BED file
24     gene_to_chromosome = {}
25     with open(bed_file, 'r') as bed:
26         for line in bed:
27             fields = line.strip().split('\t')
28             gene = fields[1]
29             chromosome = fields[0]
30             gene_to_chromosome[gene] = chromosome
31
32     # Initialize data structures for syntenic gene and interaction analysis
33     syntenic_genes = set()
34     chromosome_counts = defaultdict(int)
35     interaction_counts = defaultdict(int)
36     interaction_genes = defaultdict(set)
37
38     with open(file_path, 'r') as collinearity:
39         current_alignment = None
40         current_pair = None
41
42         for line in collinearity:
43             # Skip non-relevant lines

```

2. Número de genes que participan en bloques de colinealidad interespecies en cada cromosoma, ordenado de cromosomas con más a menos genes:

```
CHALCO > eproceso > mcscanx_analysis_relaxaed_names >
```

1	Chromosome	Syntenic_Gene_Count
2	AT2	2353
3	AT6	2262
4	AT4	1914
5	AT3	1493
6	AT5	1476
7	CU160	457
8	CU148	358
9	CU87	335
10	CU142	275
11	CU155	248
12	CU31	245
13	CU130	238

3. Número de parejas cromosomales, el conteo de bloques sintéticos entre cada pareja, así como el total de genes en cada una, ordenado de parejas con mas bloques a menos bloques sinténicos:

```
CHALCO > eproceso > mcscanx_analysis_relaxaed_names > chromosome_interactions.tsv
```

1	Chromosome_Pair	Syntenic_Block_Count	Total_Genes_Involved
2	AT6&CU160	20	404
3	AT2&CU16	16	305
4	AT2&CU143	15	271
5	AT6&CU31	15	328
6	AT4&CU142	14	245
7	AT4&CU155	14	278
8	AT2&CU140	13	231
9	AT2&CU144	13	269
10	AT2&CU160	13	277
11	AT6&CU87	13	363

4. Luego de obtener estadísticas generales de los bloques sinténicos, pensamos en cómo evaluar el enriquecimiento de términos GO entre los genes de diferentes especies, y analizar los GO diferenciales como los conservados. Primero analizamos los GO **diferenciales**, analizando su conteo entre diferentes especies. Para ello usamos el script en python “count\_go\_per\_eggnogfile.py”. Este script analiza el output annotations del eggnog de cada especie para (i) sacar el número de veces que aparece determinado GO en el archivo anotado de cada especie, (ii) darme todos los genes asociados a cada GO, y (iii) darme un archivo tsv con el conteo de cada GO en cada cromosoma de cada especie.

```

CHALCO > eproceso > egglog > count_go_per_egglogfile.py > ...
13
14 import os
15 from collections import defaultdict
16
17 # Define specific GO terms of interest
18 specific_gos = {
19     "GO:0019253", "GO:0010110", "GO:0019685", "GO:0015979", "GO:0006796",
20     "GO:0040011", "GO:1902019", "GO:0009399", "GO:1902025"
21 }
22
23 # Species prefixes mapped to their EggNOG filenames
24 species_prefix = {
25     "arabidopsis_egglog.tsv": "AT",
26     "chlamydomonas_egglog.tsv": "CR",
27     "cuscuta_egglog.tsv": "CU",
28     "gorgonias_egglog.tsv": "RO",
29 }
30
31 # File paths and directories
32 input_dir = "/media/crowfoot2/DATOS/CHALCO/eprocso/egglog"
33 gff_file = "/media/crowfoot2/DATOS/CHALCO/eprocso/mcscanx_analysis_relaxaed_names/all_new.gff"
34 # Output files
35 output_file = "/media/crowfoot2/DATOS/CHALCO/eprocso/egglog/go_genes.tsv"
36 summary_file = "/media/crowfoot2/DATOS/CHALCO/eprocso/egglog/go_summary.tsv"
37 chromosome_file = "/media/crowfoot2/DATOS/CHALCO/eprocso/egglog/go_chromosome_summary.tsv"
38
39 # Data structures for storing results
40 go_to_genes = defaultdict(lambda: defaultdict(list))

```

- El resultado principal fue el summary del número de veces que aparece cada GO en cada especie. Esto se realizó con GOs asociados a fotosíntesis, importación del nitrato y cilios, todo mediante el script en R **figures.RMD** (ver PASO 5 en el script) y se obtuvo lo siguiente:

```
CHALCO > eproceso > egglog > go_summary.tsv > data
```

	GO	arabidopsis	chlamydomonas	cuscuta	roridula	Total
1	GO:0015979	261	4	63	93	421
2	GO:0019253	9	1	5	4	19
3	GO:0019685	9	1	5	4	19
4	GO:1902025	12	0	4	7	23
5	GO:0010110	3	1	1	0	5
6	GO:1902019	0	5	0	0	5

A partir de esta tabla se produjo la tabla 3 y la figura 7

- El siguiente paso fue analizar los términos GO **conservados** entre especies, que es el objetivo principal del proyecto. Para ello, se generó el script de python más grande creado, con múltiples funciones, llamado "integrate\_MCScanX\_GO\_orthologues.py". Este script toma como input los resultados de annotations de Egglog para cada especie, los nombres asociados a cada GO term y KEGG pathway de los API de cada una de estas bases de datos, los grupos de ortólogos detectados por Orthofinder y el collinearity de MCScanX, los archivos de anotación gff originales de cada especie, y se genera lo descrito en los siguientes pasos

```
CHALCO > eproceso > mcsanx_analysis_stringent > integrate_MCSanX_GO_orthologues.py > ...
19 # New output file for protein IDs and locus_tags
20 OUTPUT_PROTEIN_LOCUS_PATH = "/media/crowfoot2/DAT05/CHALCO/eprocso/eggnog/syntenic_blocks_protein_locus.tsv"
21
22 # New output files for interspecies blocks
23 OUTPUT_GO_FILTERED_INTERSPECIES_PATH = "/media/crowfoot2/DAT05/CHALCO/eprocso/eggnog/syntenic_go_filtered_interspecie
24 OUTPUT_KEGG_INTERSPECIES_PATH = "/media/crowfoot2/DAT05/CHALCO/eprocso/eggnog/syntenic_kegg_interspecies_stringent.ts
25
26 > def download_go_ontology(go_obo_url): ...
68
69 > def download_kegg_pathways(): ...
87 > def parse_gff_files(gff_dir): ...
123
124 > def parse_n0_tsv(n0_tsv_path): ...
160
161 > def parse_eggnog_files(eggnog_dir): ...
207
208 > def parse_collinearity_file(collinearity_path): ...
253
254 def process_blocks(blocks, block_gene_count, block_species, gene_to_go, gene_to_kegg_pathways, gene_to_og, go_dict, ke
255 """
256     Processes each block, collects GO terms, KEGG pathways, and OGs, sorts the blocks and terms as required,
257     and writes the outputs.
258     Also generates additional files for interspecies blocks.
259     """
260     print("Processing blocks to collect GO terms, KEGG pathways, OGs, and descriptions...")
261     # Sort blocks by total gene count in descending order
262     sorted_blocks = sorted(blocks.items(), key=lambda x: len(x[1]), reverse=True)
```

7. El resultado probablemente más importante del script “integrate\_MCScanX\_GO\_orthologues.py” es el archivo tsv con el conteo de términos GO presentes en cada bloque sinténico, que básicamente me permite conocer qué GO están enriquecidos en ese bloque sinténico conservado entre especies, cuántas proteínas corresponden a ese GO y cuáles IDs de esas proteínas:

```
CHALCO > eproceso > egglog > syntenic_go_filtered_interspecies_stringent.tsv > data
```

	Block	GO_Term	GO_Description	GO_Count	Genes
1	Block411	N=47	AT3&CU90 plus	GO:0005622	intracellular anatomical structure 49 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
2	Block411	N=47	AT3&CU90 plus	GO:0043226	organelle 47 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
3	Block411	N=47	AT3&CU90 plus	GO:0043229	intracellular organelle 47 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
4	Block411	N=47	AT3&CU90 plus	GO:0043231	intracellular membrane-bounded organelle 47 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
5	Block411	N=47	AT3&CU90 plus	GO:0043227	membrane-bounded organelle 47 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
6	Block411	N=47	AT3&CU90 plus	GO:0005737	cytoplasm 38 CURAL46123.1,ATNP_001318308.1,CURAL45920.1,ATNP_565640.1
7	Block411	N=47	AT3&CU90 plus	GO:0065007	biological regulation 34 ATNP_001318308.1,ATNP_565640.1,ATNP_180441.1,CURAL45920.1
8	Block411	N=47	AT3&CU90 plus	GO:0009987	cellular process 33 ATNP_001318308.1,ATNP_180441.1,CURAL45920.1,ATNP_565640.1
9	Block411	N=47	AT3&CU90 plus	GO:0005634	nucleus 31 ATNP_001318308.1,ATNP_565640.1,ATNP_180441.1,CURAL45920.1
10	Block411	N=47	AT3&CU90 plus	GO:0050789	regulation of biological process 28 ATNP_001318308.1,ATNP_180441.1,CURAL45920.1,ATNP_565640.1
11	Block411	N=47	AT3&CU90 plus	GO:0050794	regulation of cellular process 27 ATNP_001318308.1,ATNP_180441.1,CURAL45920.1,ATNP_565640.1
12	Block411	N=47	AT3&CU90 plus	GO:0005488	binding 26 ATNP_001318308.1,ATNP_180441.1,ATNP_001323611.1,ATNP_565640.1
13	Block411	N=47	AT3&CU90 plus	GO:0005086	response to stimulus 23 CURAL46123.1,ATNP_001318308.1,ATNP_565640.1
14	Block411	N=47	AT3&CU90 plus	GO:0044238	primary metabolic process 23 ATNP_001318308.1,CURAL45920.1,ATNP_565640.1
15	Block411	N=47	AT3&CU90 plus	GO:0008152	metabolic process 23 ATNP_001318308.1,CURAL45920.1,CURAL46123.1,ATNP_565640.1
16	Block411	N=47	AT3&CU90 plus	GO:0003824	catalytic activity 23 ATNP_001318308.1,ATNP_850122.1,CURAL46123.1,ATNP_565640.1
17	Block411	N=47	AT3&CU90 plus	GO:0043170	macromolecule metabolic process 19 ATNP_001318308.1,CURAL46123.1,ATNP_565640.1
18	Block411	N=47	AT3&CU90 plus	GO:0008000	regulation of primary metabolic process 19 ATNP_001318308.1,CURAL46123.1,ATNP_565640.1

- Se obtuvo lo mismo para las vías metabólicas del KEGG, aunque al final resultó no contribuir tanto al análisis (con los GO terms bastó)

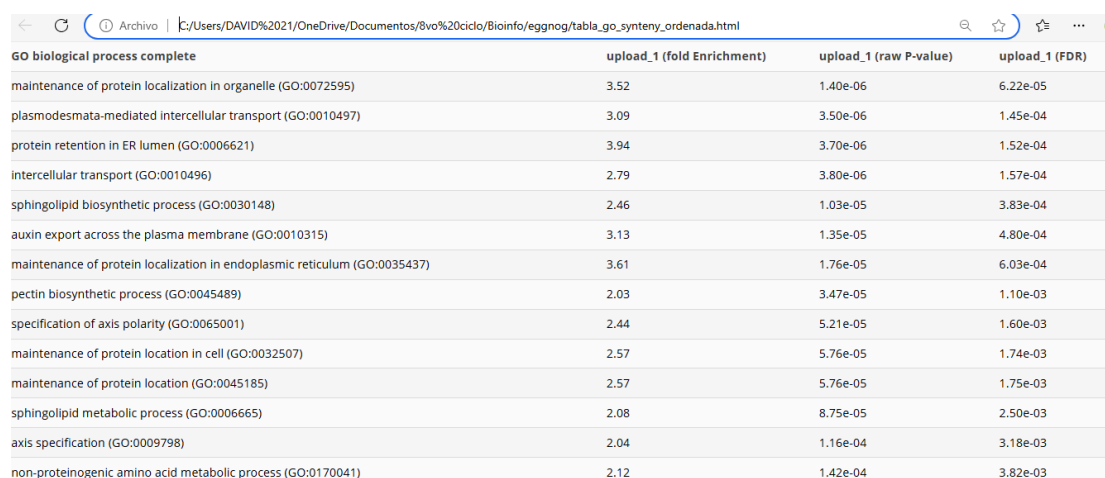
1	Block	KEGG_Pathway_ID	KEGG_Pathway_Description	Pathway_Count	Genes
2	Block411	N=47	AT3&CU90 plus map01100	Metabolic pathways	8 ATNP_850122.1,ATNP_565658.1,CURAL4589
3	Block411	N=47	AT3&CU90 plus map05034	Alcoholism	6 CURAL45922.1,ATNP_180441.1,ATNP_180440.1,CURA
4	Block411	N=47	AT3&CU90 plus map05152	Tuberculosis	6 CURAL45922.1,ATNP_001318308.1,ATNP_850122
5	Block411	N=47	AT3&CU90 plus map01110	Biosynthesis of secondary metabolites	6 CURAL46123.1,ATNP
6	Block411	N=47	AT3&CU90 plus map04626	Plant-pathogen interaction	5 CURAL45922.1,CURAL46078.1,ATN
7	Block411	N=47	AT3&CU90 plus map04921	Oxytocin signaling pathway	4 CURAL45922.1,ATNP_850094.1,AT
8	Block411	N=47	AT3&CU90 plus map04371	Apelin signaling pathway	4 CURAL45922.1,ATNP_850094.1,AT
9	Block411	N=47	AT3&CU90 plus map04722	Neurotrophin signaling pathway	4 CURAL45922.1,ATNP_0013183
10	Block411	N=47	AT3&CU90 plus map05418	Fluid shear stress and atherosclerosis	4 CURAL45922.1,ATNP
11	Block411	N=47	AT3&CU90 plus map05133	Pertussis	4 CURAL45922.1,ATNP_001318308.1,CURAL46144.1,AT
12	Block411	N=47	AT3&CU90 plus map04910	Insulin signaling pathway	4 CURAL45922.1,ATNP_850094.1,AT
13	Block411	N=47	AT3&CU90 plus map04922	Glucagon signaling pathway	4 CURAL45922.1,ATNP_850094.1,AT
14	Block411	N=47	AT3&CU90 plus map05322	Systemic lupus erythematosus	4 ATNP_180441.1,ATNP_180440
15	Block411	N=47	AT3&CU90 plus map05203	Viral carcinogenesis	4 ATNP_180441.1,ATNP_180440.1,CURAL
16	Block411	N=47	AT3&CU90 plus map04750	Inflammatory mediator regulation of TRP channels	2 CURAL
17	Block411	N=47	AT3&CU90 plus map04740	Olfactory transduction	2 CURAL45922.1,ATNP_180271.1
18	Block411	N=47	AT3&CU90 plus map05031	Amphetamine addiction	2 CURAL45922.1,ATNP_180271.1

9. Otro de los resultados relevantes fue la obtención del locus\_tag para cada proteína de cada bloque sinténico. En la imagen inferior, por ejemplo, vemos para el Block431 (como el lector apreciará, se trata de un bloque sinténico intraespecie) cada una de las proteínas de *A. thaliana* correspondientes en la segunda columna, y su locus\_tag en la cuarta columna (la tercera columna es lo mismo que la segunda, se refiere a la misma proteína, solo que sin el prefijo de especie de dos letras al inicio de su código). ¿Y para qué quiero obtener el locus\_tag de cada proteína en cada bloque sinténico? Lo veremos más adelante.

1	Block	Gene_ID	Protein_ID	Locus_Tag	
2	Block431	N=96	AT4&AT6 plus	ATNP_001326403.1	NP_001326403.1 AT3G03010
3	Block431	N=96	AT4&AT6 plus	ATNP_197086.1	NP_197086.1 AT5G15820
4	Block431	N=96	AT4&AT6 plus	ATNP_187034.1	NP_187034.1 AT3G03840
5	Block431	N=96	AT4&AT6 plus	ATNP_197233.1	NP_197233.1 AT5G17310
6	Block431	N=96	AT4&AT6 plus	ATNP_187056.1	NP_187056.1 AT3G04060
7	Block431	N=96	AT4&AT6 plus	ATNP_197344.2	NP_197344.2 AT5G18430
8	Block431	N=96	AT4&AT6 plus	ATNP_197069.1	NP_197069.1 AT5G15650
9	Block431	N=96	AT4&AT6 plus	ATNP_197314.1	NP_197314.1 AT5G18130
10	Block431	N=96	AT4&AT6 plus	ATNP_186995.1	NP_186995.1 AT3G03450
11	Block431	N=96	AT4&AT6 plus	ATNP_186929.2	NP_186929.2 AT3G02800
12	Block431	N=96	AT4&AT6 plus	ATNP_197169.2	NP_197169.2 AT5G16660
13	Block431	N=96	AT4&AT6 plus	ATNP_001327779.1	NP_001327779.1 AT3G04350
14	Block431	N=96	AT4&AT6 plus	ATNP_187043.1	NP_187043.1 AT3G03930

10. Seguidamente, se utilizó el primer resultado (paso 7) del conteo de GO en cada bloque sinténico interespecies para ubicar fácilmente algunos bloques sinténicos grandes en synvisio, visualizarlos, y poder cerciorarme de que estén enriquecidos en determinados términos GO, como se ve en la imagen inferior. Aunque en un inicio la inspección de enriquecimiento fue visual, solo viendo la descripción de los términos GO más frecuentes en cada bloque sinténico (ver imagen del paso 7), luego se usó el locus tag de las proteínas de mi bloque sinténico de interés (del paso 9) y lo introduje en la plataforma DAVID (<https://davidbioinformatics.nih.gov/tools.jsp>), el cual dio como resultado el enriquecimiento de términos GO del bloque sinténico específico, que se ve en la figura 11.

11. Por otra parte, se quisieron hacer comparaciones entre todos los bloques sinténicos entre cada par de especies, pero, como se mencionó anteriormente, solo entre *Arabidopsis* y *Cuscuta* se obtuvo un buen número de bloques conservados, decidimos centrarnos en estas especies. Entonces, se eligieron todos los bloques sinténicos entre *Arabidopsis* y *Cuscuta*, extraje todos sus locus\_tag (del paso 9) y los introduje en Panther a través de la plataforma gene ontology (<https://geneontology.org/>, no se usó DAVID porque no aceptaba un número tan grande de locus\_tag). El resultado fue una tabla, la cual se descargó y procesó con el script de R **figures.RMD** (ver PASO 11 en el script), para filtrar los que tuvieron mayor Fold\_Enrichment y p-value ajustado para pruebas múltiples. El output fue el siguiente archivo html.



GO biological process complete	upload_1 (fold Enrichment)	upload_1 (raw P-value)	upload_1 (FDR)
maintenance of protein localization in organelle (GO:0072595)	3.52	1.40e-06	6.22e-05
plasmodesmata-mediated intercellular transport (GO:0010497)	3.09	3.50e-06	1.45e-04
protein retention in ER lumen (GO:0006621)	3.94	3.70e-06	1.52e-04
intercellular transport (GO:0010496)	2.79	3.80e-06	1.57e-04
sphingolipid biosynthetic process (GO:0030148)	2.46	1.03e-05	3.83e-04
auxin export across the plasma membrane (GO:0010315)	3.13	1.35e-05	4.80e-04
maintenance of protein localization in endoplasmic reticulum (GO:0035437)	3.61	1.76e-05	6.03e-04
pectin biosynthetic process (GO:0045489)	2.03	3.47e-05	1.10e-03
specification of axis polarity (GO:0065001)	2.44	5.21e-05	1.60e-03
maintenance of protein location in cell (GO:0032507)	2.57	5.76e-05	1.74e-03
maintenance of protein location (GO:0045185)	2.57	5.76e-05	1.75e-03
sphingolipid metabolic process (GO:0006665)	2.08	8.75e-05	2.50e-03
axis specification (GO:0009798)	2.04	1.16e-04	3.18e-03
non-proteinogenic amino acid metabolic process (GO:0170041)	2.12	1.42e-04	3.82e-03

De esta tabla se examinaron los términos GO más enriquecidos al considerar todos los bloques sinténicos colineares entre *Arabidopsis* y *Cuscuta* y se obtuvieron las figuras 8, 9 y 10.

# 4.Resultados

Figura 1.Resultados de continuidad de los genomas ensamblados

	Cuscuta australis	Arabidopsis thaliana	Chlamydomonas reinhardtii	Roridula gorgonia
Statistics without reference	GCA_003260385_1_Cau_v1_0_geno...	GCF_000001735_4_TAIR10_1_geno...	GCF_000002595_2_Chlamydomonas...	R_gorgonias_assembly_fasta
# contigs	218	7	53	20 615
# contigs (>= 0 bp)	218	7	53	20 615
# contigs (>= 1000 bp)	218	7	53	20 615
Largest contig	10 192 992	30 427 671	9 730 733	191 047
Total length	262 630 465	119 668 634	111 098 438	284 227 596
Total length (>= 0 bp)	262 630 465	119 668 634	111 098 438	284 227 596
Total length (>= 1000 bp)	262 630 465	119 668 634	111 098 438	284 227 596
N50	3 625 894	23 459 830	7 783 580	46 984
N90	946 644	18 585 056	3 826 814	5024
auN	3 615 237	24 546 704	7 093 996	44 941
L50	27	3	7	2209
L90	84	5	15	8938
GC (%)	36.4	36.06	64.08	36.61

Figura 2.Resultados de integridad de los genomas ensamblados

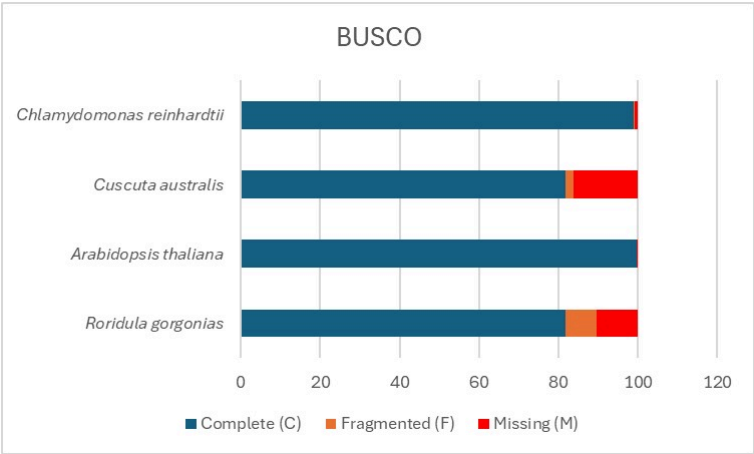


Figura 3. Eventos de duplicación génica

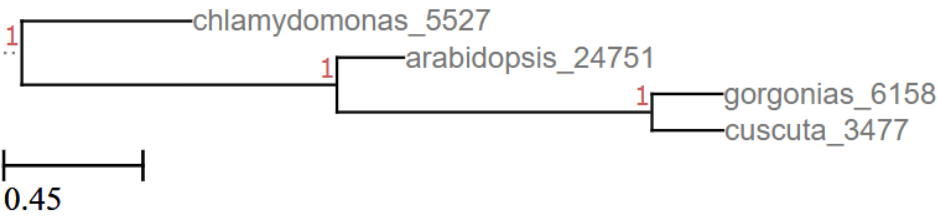


Tabla 2. Matriz de ortólogos en común entre especies.

Especie	A. thaliana	C. reinhardtii	C. australis	R. gorgonias
A. thaliana	0	5454	13042	17778
C. reinhardtii	14939	0	6356	8102
C. australis	26432	5159	0	15939
R. gorgonias	29358	5415	12891	0

Cada entrada (i,j) es el número de proteínas en j que tiene ortólogos en la especie i. *Chlamydomonas reinhardtii* tiene pocas secuencias ortólogas en común con las demás especies. Mientras que *Cuscuta* y *Roridula* tienen un alto número de genes ortólogos en común con *Arabidopsis* y entre ellas mismas.

Figura 4. Porcentaje de secuencias de cada especie asignados a singletons (grupos ortólogos de una única secuencia), grupos ortólogos de dos secuencias, y grupos ortólogos con más de tres secuencias.

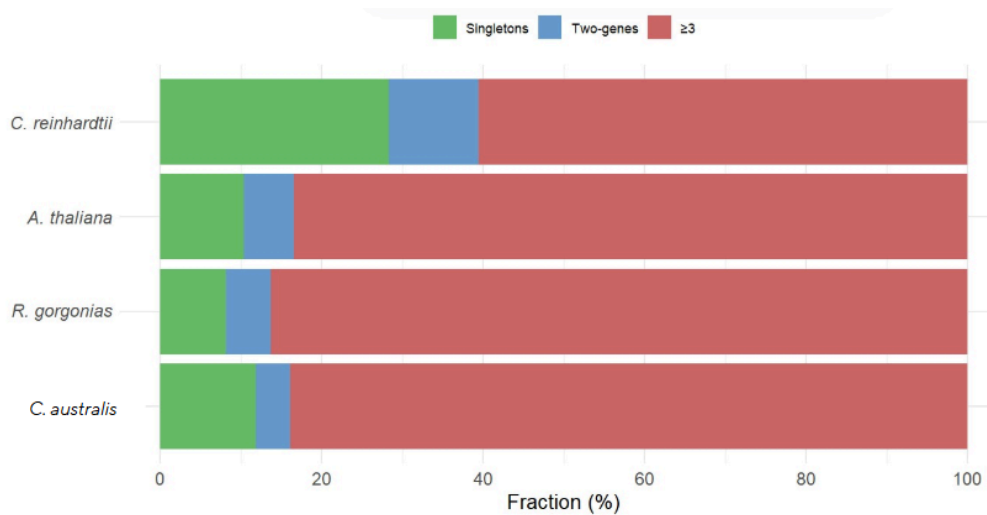


Figura 5. Porcentaje de genes de cada especie asignados a singletons, grupos ortólogos solo detectados en esa especie (Species specific), y grupos ortólogos compartidos con al menos una especie (Shared orthogroups).

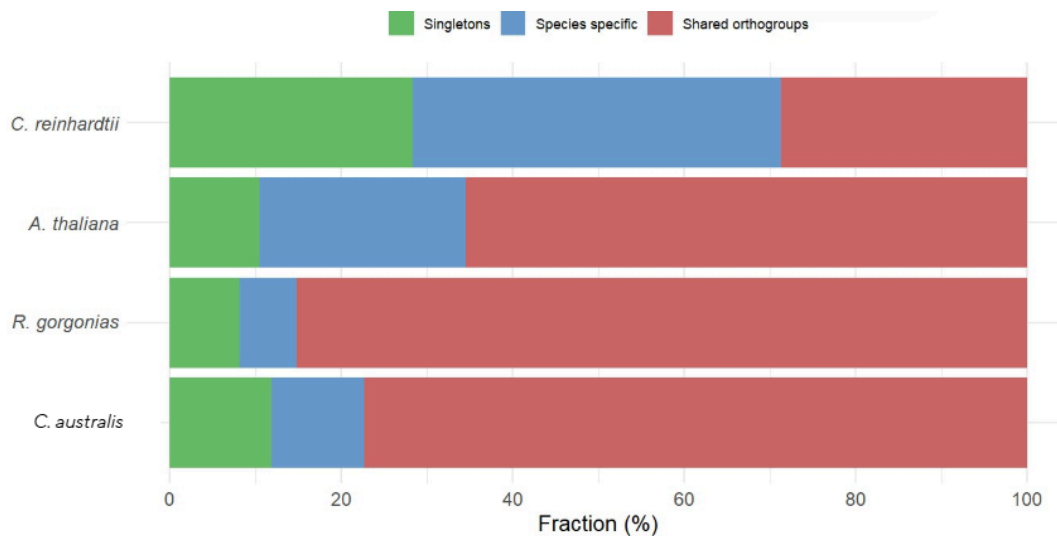
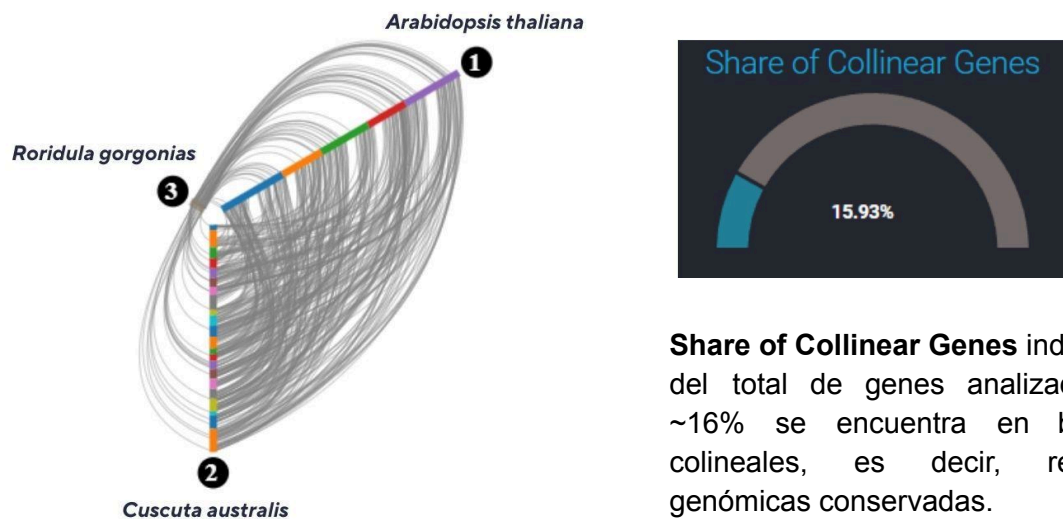




Figura 6. Colinealidad de tres especies fotosintéticas terrestres



Los fragmentos de colores representan los cromosomas, mientras que las líneas grises indican los bloques sinténicos en común.

Tabla 3. Recuento de término GO por cada especie

Recuentos de términos GO por especie						
GO	Nombre	Arabidopsis	Chlamydomonas	Cuscuta	Roridula	Total
GO:0015979	Fotosíntesis	261	4	63	93	421
GO:0019253	Ciclo reductivo de pentosa-fosfato	9	1	5	4	19
GO:0019685	Reacción oscura de la fotosíntesis	9	1	5	4	19
GO:1902025	Importación del nitrato	12	0	4	7	23
GO:0010110	Regulación de la reacción oscura de la fotosíntesis	3	1	1	0	5
GO:1902019	Regulación de la motilidad celular dependiente de cilios	0	5	0	0	5

Se muestra cada uno de los GO con respectivos nombres, además, el número de genes contenidos en cada GO según la especie.

Figura 7. Gráfico de barras de los recuentos por términos GO (nº de genes) y especies

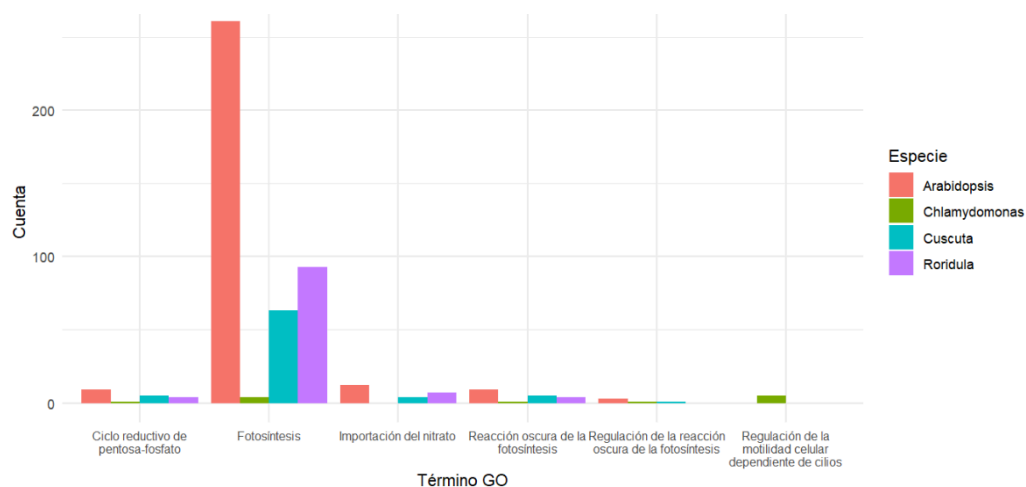


Figura 8. Términos GO enriquecidos en bloques sinténicos entre *Arabidopsis* y *Cuscuta*.

GO biological process complete	upload_1 (fold Enrichment)	upload_1 (raw P-value)	upload_1 (FDR)
regulation of cyclin-dependent protein kinase activity (GO:1904029)	2.17	4.59e-04	1.02e-02
regulation of protein serine/threonine kinase activity (GO:0071900)	2.17	4.59e-04	1.02e-02
regulation of DNA endoreduplication (GO:0032875)	2.17	4.59e-04	1.03e-02
regulation of cyclin-dependent protein serine/threonine kinase activity (GO:0000079)	2.17	4.59e-04	1.03e-02
regulation of actin nucleation (GO:0051125)	4.33	6.55e-04	1.31e-02
positive regulation of actin nucleation (GO:0051127)	4.33	6.55e-04	1.31e-02
regulation of Arp2/3 complex-mediated actin nucleation (GO:0034315)	4.33	6.55e-04	1.31e-02
positive regulation of cytoskeleton organization (GO:0051495)	3.03	2.13e-03	3.61e-02
regulation of cytokinesis (GO:0032465)	3.61	3.17e-03	4.69e-02

Se observan términos GO asociados a la regulación de señales intracelulares (actividad kinasa), y ciclo celular (nucleación de la actina, replicación, ciclinas, citocinesis).

Figura 9. Términos GO enriquecidos en bloques sinténicos entre *Arabidopsis* y *Cuscuta*. Transporte iónico.

GO biological process complete	upload_1 (fold Enrichment)	upload_1 (raw P-value)	upload_1 (FDR)
lead ion transport (GO:0015692)	4.33	6.55e-04	1.32e-02
potassium ion transmembrane transport (GO:0071805)	2.01	7.87e-04	1.54e-02
magnesium ion transmembrane transport (GO:1903830)	3.71	8.48e-04	1.61e-02
iron ion transport (GO:0006826)	2.10	9.44e-04	1.79e-02
iron ion transmembrane transport (GO:0034755)	2.41	2.90e-03	4.41e-02

Se observan términos GO asociados al transporte iónico

Figura 10. Términos GO enriquecidos en bloques sinténicos entre *Arabidopsis* y *Cuscuta*.

GO biological process complete	upload_1 (fold Enrichment)	upload_1 (raw P-value)	upload_1 (FDR)
auxin metabolic process (GO:0009850)	2.00	3.69e-04	8.60e-03
positive regulation of flower development (GO:0009911)	2.17	1.43e-04	3.82e-03
axis specification (GO:0009798)	2.04	1.16e-04	3.18e-03
specification of axis polarity (GO:0065001)	2.44	5.21e-05	1.60e-03
auxin export across the plasma membrane (GO:0010315)	3.13	1.35e-05	4.80e-04
negative gravitropism (GO:0009959)	2.25	1.55e-03	2.75e-02
polarity specification of adaxial/abaxial axis (GO:0009944)	2.26	2.27e-03	3.78e-02
auxin biosynthetic process (GO:0009851)	2.17	2.33e-03	3.86e-02

Se observan términos GO asociados al desarrollo embrionario y establecimiento de patrones

Figura 11. Bloque sinténico más grande

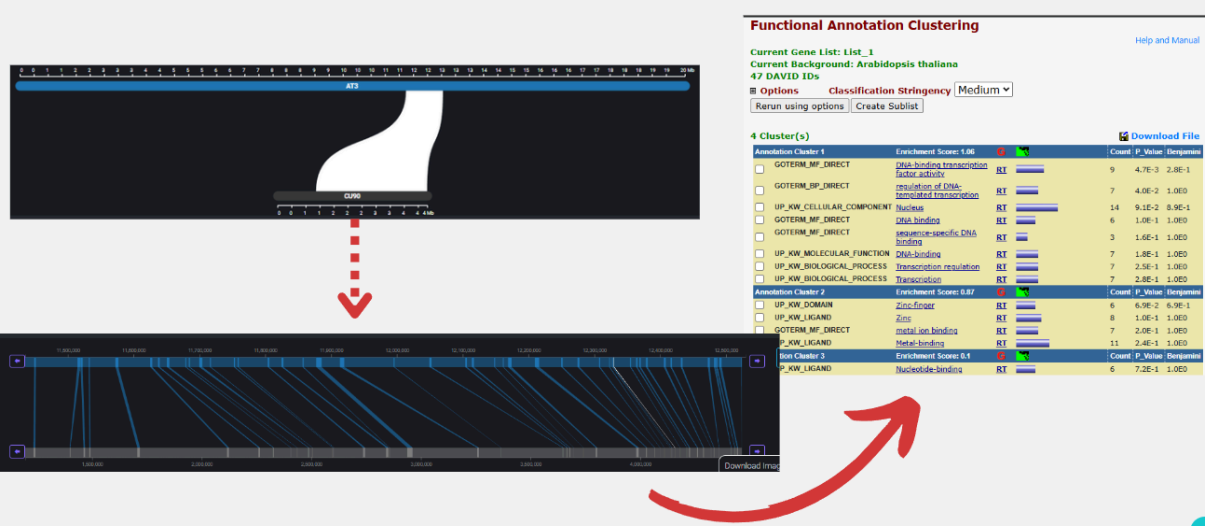
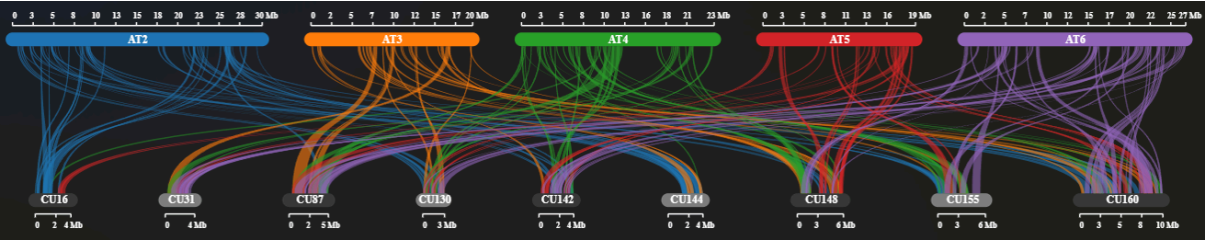


Figura 12. Panorama global de sintenia entre Arabidopsis y Cuscuta



## 5. Discusión

Del análisis realizado con QUAST (ver figura 1), el ensamblaje con mayor continuidad (mayores N50 y menores L50) corresponde a *Arabidopsis thaliana*, seguida por *Chlamydomonas reinhardtii*. Esto porque ambas especies son modelos ampliamente estudiados en biología molecular y evolutiva. Además, estos genomas fueron obtenidos de la base de datos NCBI RefSeq Assembly y asegura su calidad. Por otro lado, el tamaño de los genomas es relativamente pequeño, donde *Roridula gorgonia* tiene el genoma más grande entre los cuatro organismos, sin embargo, es el más fragmentado. Este genoma fue descargado de DRYAD, una plataforma en donde se encuentran disponibles y se puede reutilizar datos de investigación de otros artículos (DRYAD, 2024). En este caso, la secuencia de *Roridula gorgonia* obtuvo del artículo titulado “*Annotated genome sequences of the carnivorous plant Roridula gorgonias and a non-carnivorous relative, Clethra arborea*” dentro de este se mencionan las limitaciones del conjunto de datos, ya que se trata de un borrador del genoma, por lo tanto, algunos genes localizados al final de los andamios podría estar incompletos, y las regiones repetidas podrían estar ausente o estar mal ensambladas (Hartmann, S. et al., 2020). Esto se corrobora con los resultados del QUAST visto al principio. Asimismo, el genoma fragmentado concuerda con el resultado del BUSCO (figura 2), donde *Roridula* aparece como la especie con ortólogos de copia única más fragmentados.

En la Figura 3 (Eventos de duplicación génica), vemos la agrupación de las especies. La escala de 0.45 sustituciones por sitio indica que el genoma total del clado de *Roridula gorgonias* y *Cuscuta* a divergido en distancia nucleotídica casi tanto del de *Arabidopsis*, como el de *Arabidopsis* ha divergido del de *Chlamydomonas*, lo que es resultado de la gran distancia evolutiva de *Chlamydomonas* con respecto a las demás especies, y de la gran cantidad de cambios sufridos por *Roridula* y *Cuscuta* con respecto a *Arabidopsis*, probablemente por la reducción del genoma. Valores como 5527 en “chlamydomonas\_5527” representan el número de duplicaciones génicas con al menos un 50% de soporte ocurridas en esa rama. La expansión de familias génicas por duplicación en *Chlamydomonas* ya ha sido reportada en otros artículos, debido a su distancia evolutiva con respecto a las demás plantas terrestres (Guo, 2012), las adaptaciones específicas a ambientes acuáticos y características de la adaptación de sus genes a la vida unicelular. Asimismo, el clado que lleva a *Arabidopsis* tiene **24 751 duplicaciones**, que no comparte con *Roridula gorgonias* y *Cuscuta*. *Cuscuta* y *Roridula* han sufrido reducciones en su genoma, por lo que probablemente las duplicaciones de *Arabidopsis* reflejan la gran cantidad de duplicaciones y expansión de familias génicas de las angiospermas (Guo, 2012; Cannon et al., 2004). Como *Roridula* y *Cuscuta* son las únicas especies angiospermas contra las cuales el programa Orthofinder puede comparar, una buena porción del número de ortólogos que se obtiene solo en *Arabidopsis* realmente corresponde a la mayoría de dicotiledóneas, pero se asigna solo a *Arabidopsis* porque ya no están en *Roridula* y *Cuscuta*.

También, un buen número de las duplicaciones reportadas puede deberse a las rondas de duplicación, expansión génica y rearrreglos detectados en el orden Brassicales específico de *Arabidopsis* (Cannon et al., 2004). Esto también concuerda con el número reducido de proteínas de ambas especies carnívora y parásita al compararlas con *Arabidopsis* como

referencia (Sun et al., 2018; Fleck & Jobson, 2023). Asimismo, no se puede descartar que el número alto de duplicaciones y ortólogos en *Arabidopsis* también puede ocurrir por una anotación más detallada en su genoma, ya que es una planta modelo.

En Tabla 2 (matriz de ortólogos en común entre especies) se nos muestra las proteínas ortólogas en común entre especies. Donde *Chlamydomonas reinhardtii* tiene pocas secuencias ortólogas en común con las demás especies (aproximadamente solo un 25% de sus proteínas). Por su parte, *Cuscuta* y *Roridula* tienen un alto número de genes ortólogos en común con *Arabidopsis* y entre ellas mismas. De hecho, el 72% de las secuencias de *Cuscuta* (13042/18157) y el 78% de las de *Roridula*, son ortólogas con las de *Arabidopsis* (17778/22655). Asimismo, es interesante notar que un número similar (70%) comparten entre ellas mismas. Estas proteínas ortólogas podrían ser parte de un proteoma indispensable, de funciones celulares básicas, compartido entre todas estas Angiospermas.

En la Figura 4 vemos que *Chlamydomonas* es la especie que tiene un mayor número de singletons o genes sin ningún homólogo (genes huérfanos). Por su parte, también tienen el mayor número de secuencias que participan en grupos ortólogos de 2 genes, que probablemente también sean grupos ortólogos específicos de esta especie. Por otro lado, la Figura 5 nos muestra que la abrumadora mayoría de sus proteínas pertenecen a grupos ortólogos específicos de la especie *Chlamydomonas*, además de los singletons (que por definición solo están en la especie en la que se detectan). El porcentaje de proteínas en grupos ortólogos compartidos es del 25%, lo que también se verificó en la Tabla 2. (Matriz de ortólogos en común entre especies). Estos resultados concuerdan con el reporte de Guo (2013). Asimismo, *Arabidopsis* tiene un número alto de grupos ortólogos reportados como únicos para esta especie, por razones ya mencionadas como la expansión de genes sufrida en el linaje de Brassicales y de las angiospermas, la pérdida de muchos de estos genes durante la reducción de los genomas de las plantas parasítica y carnívora, así como por la mejor anotación del genoma de *Arabidopsis*.

De acuerdo al resultado de MCScanX (ver figura 6) no se encontraron bloques sinténicos en común entre *Chlamydomonas* y las demás especies, lo que es esperable dado el bajo porcentaje de genes ortólogos encontrados en común con las demás especies y el alto número de singletons o genes huérfanos (Figura 5 y Tabla 2). *Chlamydomonas* ha experimentado una divergencia evolutiva muy significativa de aproximadamente 725 MA (Guo 2012, ver Figura 3. Eventos de duplicación de genes). Por otra parte, en la imagen de colinealidad, se puede observar que el genoma de *Roridula* se encuentra altamente fragmentado en comparación con *Cuscuta* y *Arabidopsis*. Si bien *Roridula* posee el genoma más grande entre los cuatro organismos, desde el inicio estaba atada una serie de limitaciones en el conjunto de datos, dado que se trata de un borrador del genoma sin ensamble a nivel de scaffolds. Ello impidió la recuperación de bloques sinténicos grandes y por esta razón, la identificación de bloques sintéticos se centró exclusivamente en *A. thaliana* y *C. australis*.

Antes de analizar los bloques sinténicos, describiremos algunas funciones diferenciales detectadas en base a los términos GO. La tabla 3 y figura 7 nos muestran el conteo de términos GO (n° de genes con ese término) y especies. *Arabidopsis thaliana* tiene la mayor cantidad de genes en cada término GO, exceptuando al GO de regulación de motilidad celular dependiente de cilios (que como es de esperar es único de *Chlamydomonas*, un

organismo unicelular que se moviliza con cilios). Como se explicó antes, esto puede deberse a una mayor anotación y a la duplicación de genes o incluso, en menor medida, la transferencia horizontal de genes (HGT). Aunque el papel de la HGT en organismos eucariotas multicelulares es mucho menos claro y a veces se considera anecdótico (Jianchao Ma, 2022). Sin embargo, no desestimamos la transferencia horizontal de genes (HGT) debido a que ocurrieron dos eventos significativos de HGT en la evolución de las plantas terrestres. Además, la gran mayoría de los genes adquiridos en los dos eventos se han conservado en los grupos descendientes y están involucrados en las respuestas al estrés, el transporte de iones y metabolitos, así como en el crecimiento y el desarrollo; y el metabolismo especializado como en el caso de plantas carnívoras y parasítica (Jianchao Ma, 2022). Así mismo, se observa que *Cuscuta* tiene un número reducido de genes involucrados en la mayoría de vías, incluido fotosíntesis, importación del nitrato y la reacción oscura de la fotosíntesis. Es posible que ello se deba a eventos de pérdida de genes resultante de su estilo de vida parasitario. Esta hipótesis se respalda en estudios previos, que reportan que *Cuscuta* ha perdido el 11.7% de los ortólogos normalmente conservados en plantas autótrofas (Guiling Sun, 2018), incluidos genes estructurales que participan en el ciclo de los electrones obtenidos en el fotosistema I (los genes *ndh*) (Guiling Sun, 2018). El caso de *Roridula* es muy similar al de *Cuscuta*, y esperable, porque también se conoce que las especies carnívoras tienen un genoma reducido y sufren cambios en la obtención de nutrientes (Fleck & Jobson, 2023). En ambas especies se observa una caída en los genes de la fase oscura de la fotosíntesis y su regulación, por el mismo hecho de que obtienen energía y carbono de otras fuentes.

Finalmente, en el resultado principal, los términos GO enriquecidos en bloques sinténicos entre *Arabidopsis* y *Cuscuta* están relacionados con la regulación de señales intracelulares (actividad kinasa), y ciclo celular (nucleación de la actina, replicación, ciclinas, citocinesis) (ver figura 8), el transporte iónico (ver figura 9), y el desarrollo embrionario y establecimiento de patrones (ver figura 10). Estos resultados corresponden a lo mencionado en el artículo de Jianchao Ma (2022), quien señala algunos genes involucrados en transporte de metabolitos e iones, así como en el crecimiento y desarrollo, fueron adquiridos durante dos eventos importantes de HGT y que a menudo se han acumulado, duplicado o diferenciado funcionalmente en grupos descendientes, así contribuyendo a la diversificación y la evolución a largo plazo de las plantas terrestres (Jianchao Ma, 2022). En adición, decidimos examinar qué genes o vías metabólicas estaban presentes en el bloque sinténico de mayor tamaño entre *Arabidopsis* y *Cuscuta*. Resultó que el bloque sinténico de mayor tamaño (Ver Figura 11) contiene genes conservados de factores de transcripción reguladores del ADN. Estos factores de transcripción se agrupan en 47 genes colineales, que contienen reguladores maestros de respuesta al estrés abiótico, estímulos lumínicos, radiación, respuesta a químicos, defensa bacteriana y regulación de la producción de macromoléculas (Ver Anexos). Por ello, entendemos que son parte crítica de la adaptación a la vida terrestre entre estas dos plantas, para lidiar con estreses bióticos y abióticos que tienen en común. Asimismo, tiene sentido que este resultado no se haya obtenido con *Chlamydomonas*, un organismo acuático. Es posible también que *Roridula* tuviera resultados similares a los encontrados en *Arabidopsis* y *Cuscuta*, sin embargo, para corroborarlo es necesario un genoma menos fragmentado y con mayor cantidad de anotaciones.

## 6. Conclusiones

- La mayor parte de las proteínas de *Chlamydomonas* son únicas de esta especie, carecen de términos GO, y no encontraron grupos ortólogos ni bloques sinténicos al compararlas con otras especies, probablemente porque debe enfrentarse a estreses particulares del ambiente acuático
- *Cuscuta* y *Roridula* tienen un número reducido de proteínas en comparación a *Arabidopsis*, incluido la reducción en las proteínas asociadas a la fotosíntesis.
- El mayor número de bloques sinténicos se encontró entre *Arabidopsis* y *Cuscuta*, y aunque ambas especies tienen numerosos genes ortólogos con *Roridula*, no se armaron bloques sinténicos por la fragmentación del genoma de esta última.
- Las rutas metabólicas conservadas entre *Arabidopsis* y *Cuscuta* incluyen genes de la regulación del ciclo celular, el transporte iónico, el desarrollo embrionario, cascadas de señalización intracelular y factores de transcripción reguladores de la respuesta a estrés biótico y abiótico.

## 7. Bibliografía

- Cannon, S. B., Mitra, A., Baumgarten, A., Young, N. D., & May, G. (2004). The roles of segmental and tandem gene duplication in the evolution of large gene families in *Arabidopsis thaliana*. *BMC plant biology*, 4, 1-21.
- DRYAD. (2024). Who we are. <https://datadryad.org/stash/about>
- Fleck, S. J., & Jobson, R. W. (2023). Molecular Phylogenomics Reveals the Deep Evolutionary History of Carnivory across Land Plants. *Plants*, 12(19), 3356. <https://doi.org/10.3390/plants12193356>
- Guiling Sun, Yuxing Xu, Hui Liu, Ting Sun, Jingxiong Zhang, Christian Hettenhausen, Guojing Shen, Jinfeng Qi, Yan Qin, Jing Li, Lei Wang, Wei Chang, Zhenhua Guo, Ian T. Baldwin y Jianqiang Wu. (2018). Large-scale gene losses underlie the genome evolution of parasitic plant *Cuscuta australis*. *Nature Communications*. Vol 9. <https://www.nature.com/articles/s41467-018-04721-8>
- Guo Y. L. (2012). Gene family evolution in green plants with emphasis on the origination and evolution of *Arabidopsis thaliana* genes. *The plant Journal*. <https://onlinelibrary.wiley.com/doi/10.1111/tpj.12089>
- Hartmann, S., Preick, M., Abelt, S., Scheffel, A. y Hofreiter, M. (2020). Annotated genome sequences of the carnivorous plant *Roridula gorgonias* and a non-carnivorous relative, *Clethra arborea*. *BMC Research Notes*. <https://bmcrsnotes.biomedcentral.com/articles/10.1186/s13104-020-05254-4>
- Jianchao Ma, Shuanghua Wang, Xiaojing Zhu, Guiling Sun, Guanxiao Chang, Linhong Li, Xiangyang Hu, Shouzhou Zhang, Yun Zhou, Chun-Peng Song, y Jinling Huang. (2022). Major episodes of horizontal gene transfer drove the evolution of land plants. *Molecular Plant*. Volume 15, pag. 857-871. <https://www.sciencedirect.com/science/article/pii/S1674205222000491?via%3Dihub>
- Sun, G., Xu, Y., Liu, H. et al. (2018). Large-scale gene losses underlie the genome evolution of parasitic plant *Cuscuta australis*. *NatCommun* 9, 2683. <https://www.nature.com/articles/s41467-018-04721-8#citeas>



## 8. Anexos

Visualización de los términos GO presentes en el bloque sinténico más grande en común entre *Arabidopsis* y *Cuscuta*. Se puede ver su descripción en la tercera columna, y el número de genes en los que aparecen en la cuarta columna.

83	Block411	N=47	AT3&CU90	plus	GO:0009607	response to biotic stimulus	6	ATNP_001318308.1,CURAL46144.1,C
84	Block411	N=47	AT3&CU90	plus	GO:0009605	response to external stimulus	6	ATNP_001318308.1,CURAL46144
85	Block411	N=47	AT3&CU90	plus	GO:0043207	response to external biotic stimulus	6	ATNP_001318308.1,CU
86	Block411	N=47	AT3&CU90	plus	GO:0051707	response to other organism	6	ATNP_001318308.1,CURAL46144.1,C
87	Block411	N=47	AT3&CU90	plus	GO:1902680	positive regulation of RNA biosynthetic process	6	ATNP_565640
88	Block411	N=47	AT3&CU90	plus	GO:0010557	positive regulation of macromolecule biosynthetic process	6	
	Block411	N=47	AT3&CU90	plus	GO:0009628	response to abiotic stimulus	8	CURAL46123.1,ATNP_180439.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0009314	response to radiation	8	CURAL46123.1,ATNP_180439.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0009416	response to light stimulus	8	CURAL46123.1,ATNP_180439.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0006950	response to stress	8	CURAL46123.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0048518	positive regulation of biological process	8	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0009266	response to temperature stimulus	4	CURAL46123.1,ATNP_180439.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0009409	response to cold	4	CURAL46123.1,ATNP_180439.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0042742	defense response to bacterium	4	ATNP_001318308.1,CURAL46144.1,C
	Block411	N=47	AT3&CU90	plus	GO:0004842	ubiquitin-protein transferase activity	4	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0098542	defense response to other organism	4	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:2000026	regulation of multicellular organismal development	4	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0032446	protein modification by small protein conjugation	4	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0006952	defense response	4	ATNP_001318308.1,CURAL46144.1,C
	Block411	N=47	AT3&CU90	plus	GO:0009617	response to bacterium	4	ATNP_001318308.1,CURAL46144.1,C
	Block411	N=47	AT3&CU90	plus	GO:0003676	nucleic acid binding	16	ATNP_001318308.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0140110	transcription regulator activity	16	ATNP_001318308.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0032502	developmental process	14	ATNP_565640.1,CURAL46144.1,C
	Block411	N=47	AT3&CU90	plus	GO:0003700	DNA-binding transcription factor activity	14	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0016020	membrane	14	ATNP_850122.1,CURAL46169.1,ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0032991	protein-containing complex	13	ATNP_001318308.1,C
	Block411	N=47	AT3&CU90	plus	GO:0003677	DNA binding	13	ATNP_180441.1,ATNP_001323611.1,C