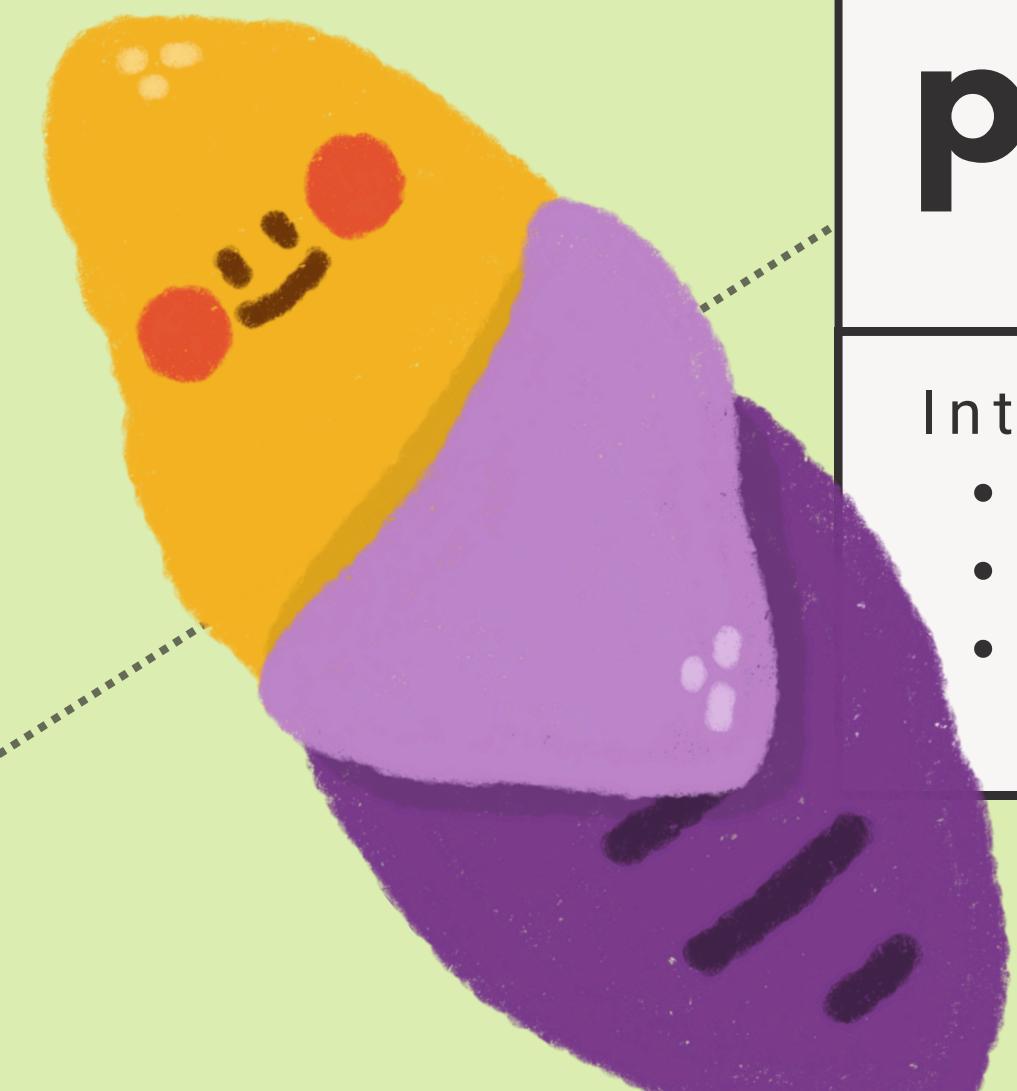
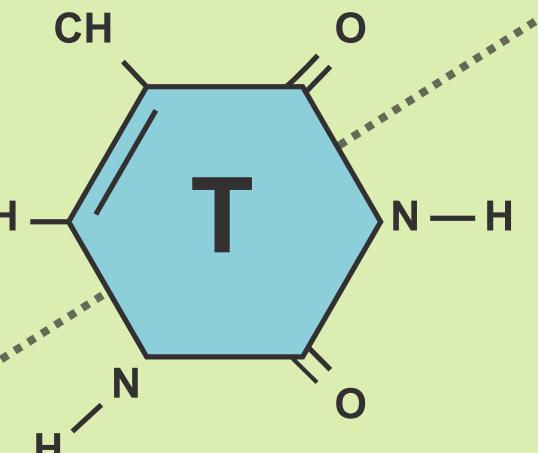
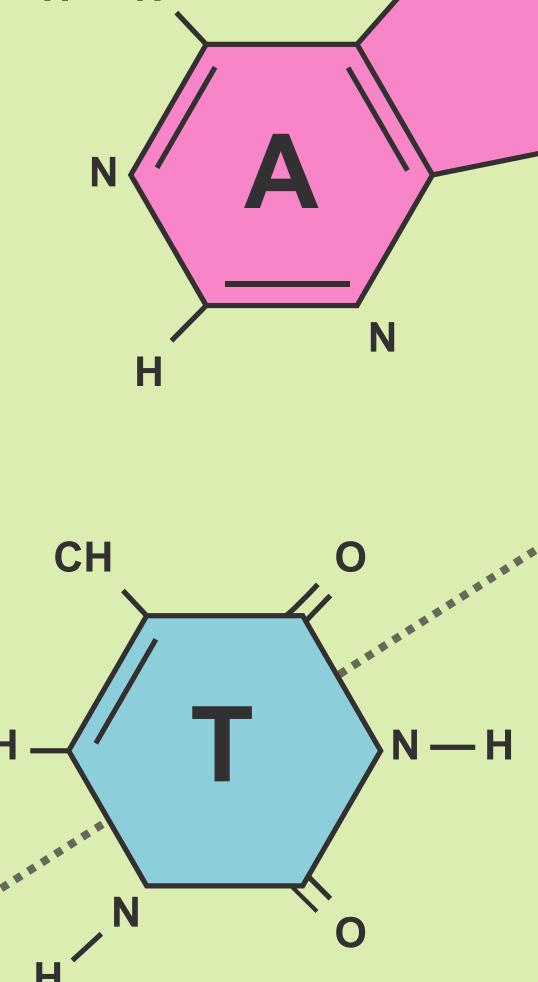
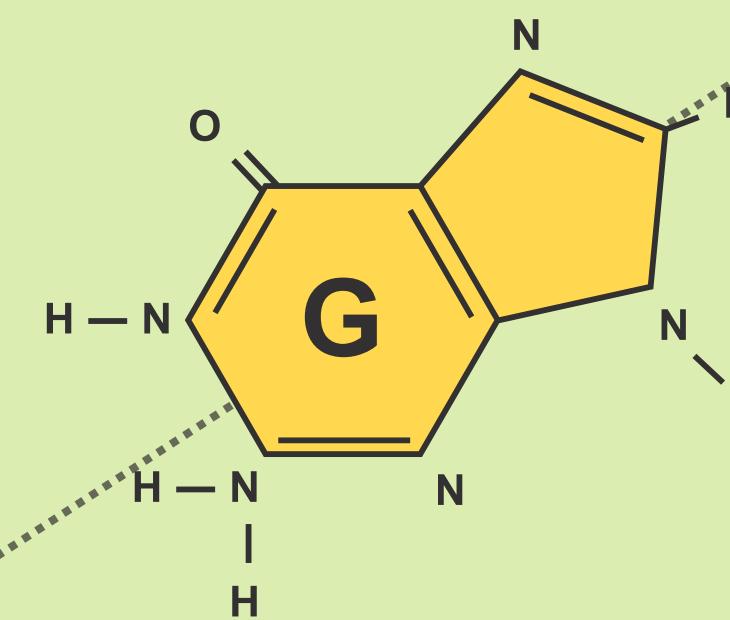


Analyzing TANDEM repeats in *Ipomoea batatas* (sweet potato)

Integrantes:

- Adrián Chalco
- Alanis Castillo
- Joaquin Espejo
- Valeria Garcia
- Gabriel Muñoz



General information about Ipomoea Batatas



Challenges behind
the study of its
genome



Importance

- Tolerance to adverse climates and soil conditions
- High starch content
- Rich in vitamins (vitamin A) and flavonoids

(Zhao et al., 2022; Lee, 2019)

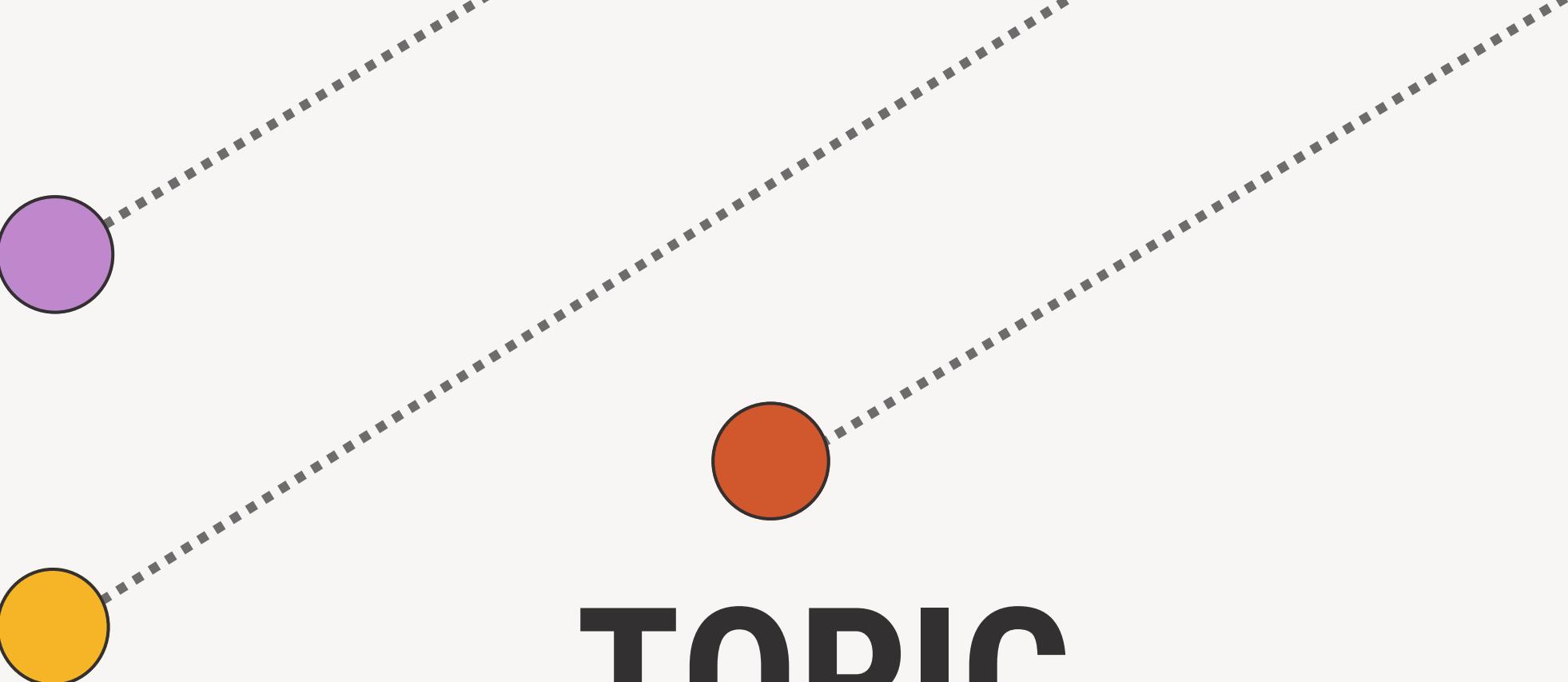
(Yan et al., 2015; Katayama et al., 2017; Xiao et al., 2022; Yan et al., 2022)

Genomic features

- Complex genome and autohexaploid
- Has 90 chromosomes
- 50% consists of repetitive regions
- High genetic variability (heterozygosity)

- Multiple copies of genes
- Due to its polyploid nature, heterozygosity increases.
- It has complex inheritance patterns due to its polyploidy.
- It requires greater complexity and costs in its genomic analysis.
- Lack of a pan-genome, which shows the various haplotypes.

(Yan et al., 2015; Xiao et al., 2022)



TOPIC, QUESTION, AND OBJECTIVES

MAIN OBJECTIVE

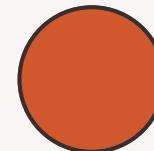
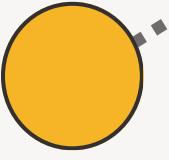
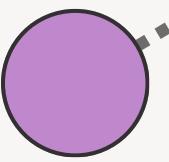
Analyze repeats in the genome of *Ipomoea batatas* to identify SSRs useful as molecular markers.

RESEARCH QUESTION

Which microsatellites in the genome of *Ipomoea batatas* can serve as suitable molecular markers?

SPECIFIC OBJECTIVES

- Identify repetitive elements in the genome of *Ipomoea batatas* using the TRF and RepeatMasker tools.
- Describe the best SSR candidates, including their location, length, and number of repeats.
- Validate the specificity of SSRs in *Ipomoea batatas* and evaluate their variability within the genus using Blast analysis.



CONTEXT

TANDEM REPEATS

MINISATELLITES

MICROSATELLITES

SATELLITES

MEJORA GENÉTICA EN
AGRICULTURA



They are short fragments of continuous repetitive DNA. They occur in non-coding DNA. (National Human Genome Research Institute, 2024)

DNA sequences of up to 6 bp repeated in TANDEM (Liao et al., 2023)

What characteristics make them good markers?

- High degree of polymorphism
- Codominant
- Easy to measure and analyze
- 100% reliability
- Repeatable and automatable

(Cadima et al., 2013; Aranguren Méndez et al., 2003; Zawedde et al., 2015)

Its use allows us to understand the genetic diversity of plants.

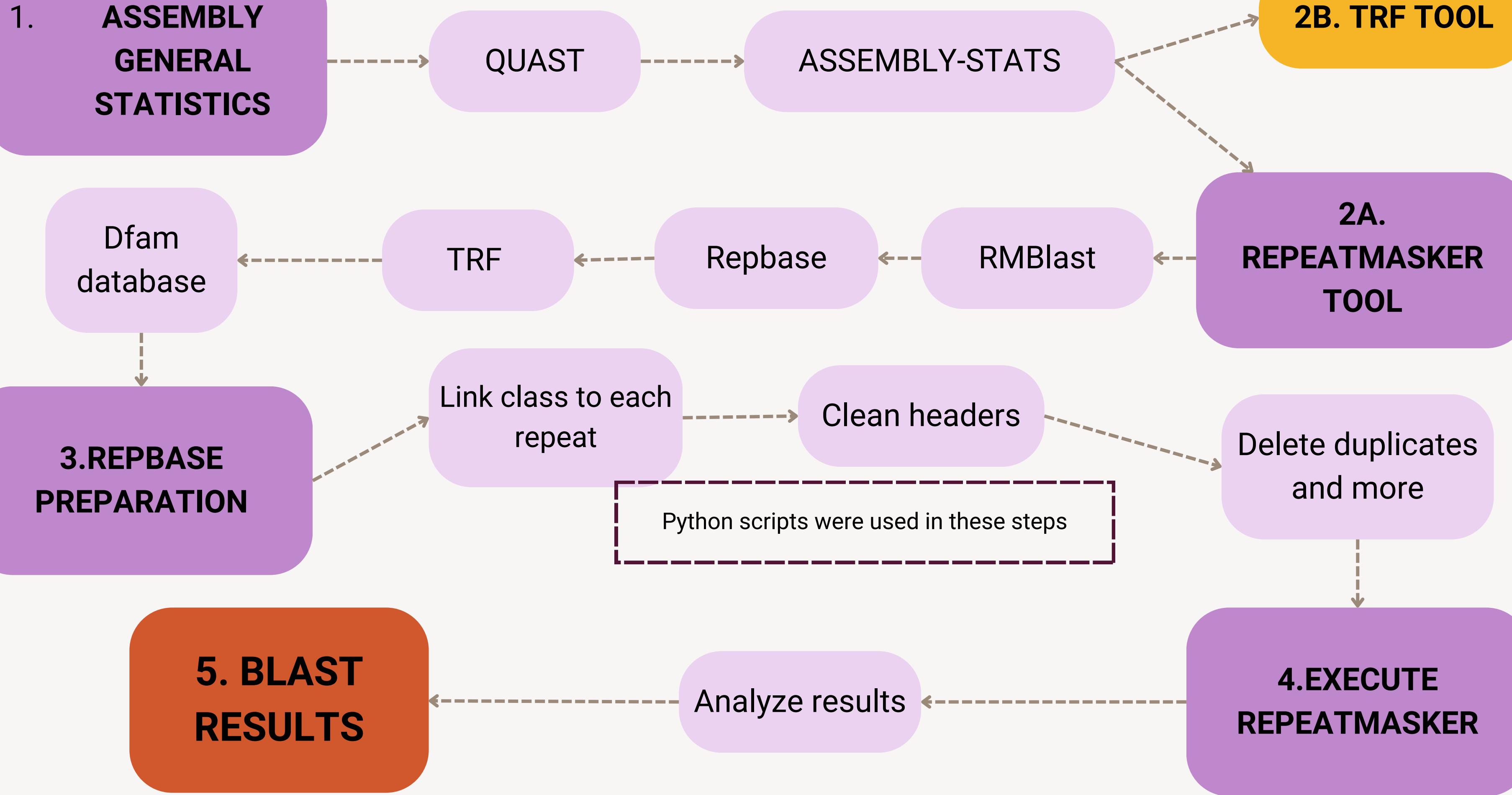
- It enables the establishment of effective and efficient practices for the conservation of plant genetic resources (Lee et al., 2019).

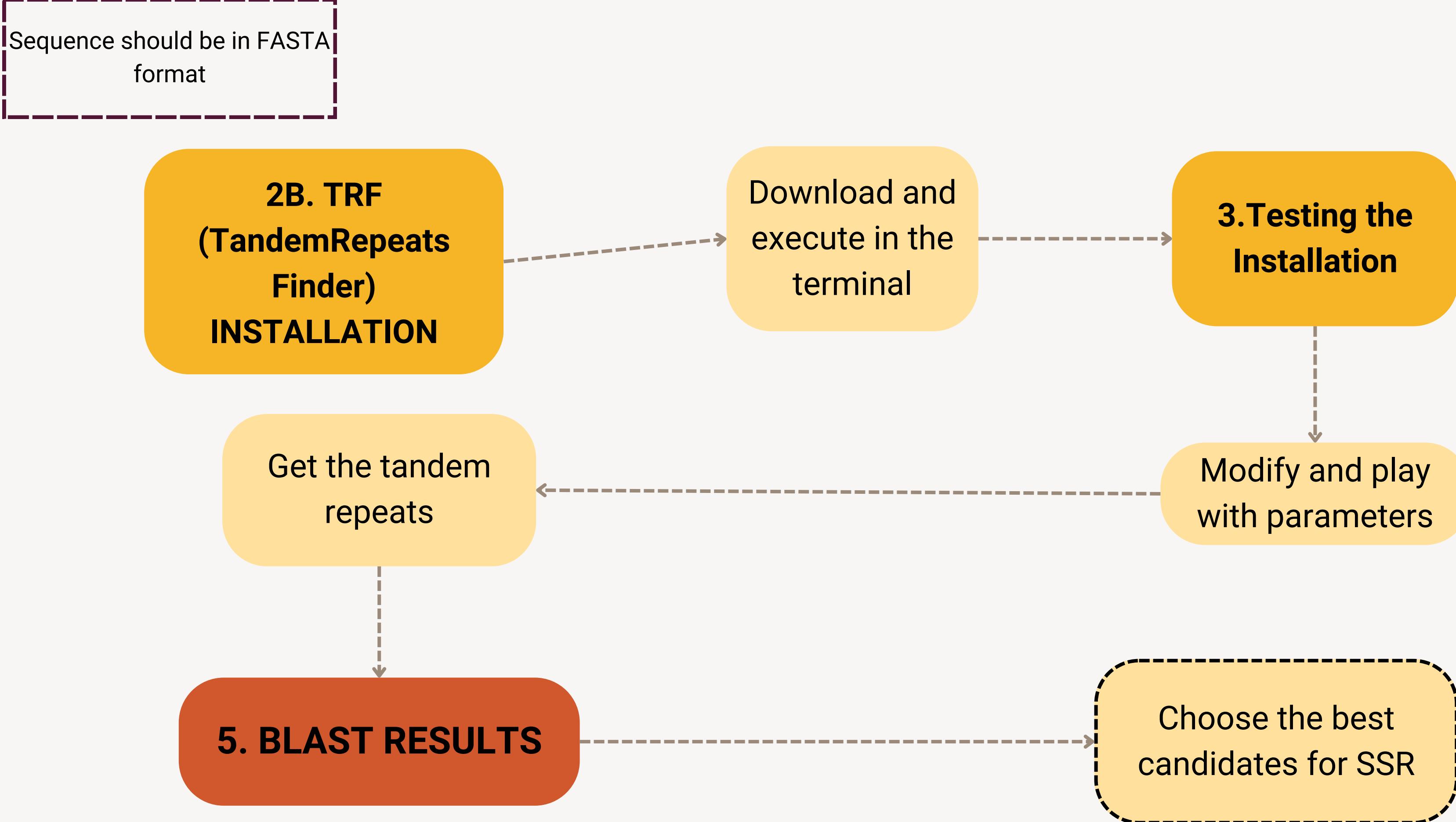
SELECTING THE IDEAL MICROSATELLITE

- Sequences of between 1 and 6 base pairs (Dirlewanger et al., 2002; Hu, 2004)
- They are mono-, di-, or trinucleotides (Provan et al., 1996)
- They are found around conserved regions (Dirlewanger et al., 2002)
- They exhibit polymorphism (Wenne, 2023)
- They are specific to Ipomoea (Madesis et al., 2013)
- They are evenly distributed (Sagar et al., 2023)
- They have a minimum of 5 repeats (Ellegren, 2004 ;Rasoarahona et al.,2023)



METHODOLOGY





2A. REPEAT MASKER CONFIGURATION

Algoritmo de búsqueda: RMBlast

Blastear cada repeat contra el genoma de Ipomoea

Comando de descarga:

```
wget  
http://www.repeatmasker.org/rmblast/rmblast-2.14.1+-x64-linux.tar.gz  
tar zxvf rmblast-2.14.1+-x64-linux.tar.gz
```

Ejecutable de TRF

Potenciar el hallazgo de microsatélites

TANDEM REPEATS FINDER

Concatenación preliminar de RepBase

Red base de elementos repetitivos.- Juntar todos los archivos fasta

```
cat *.ref > ../all_repbase.fasta  
cd appendix  
cat *.ref >> ../../all_repbase.fasta
```

Descarga de base de datos Dfam

Red base de elementos transponibles.- Se extrajo elementos pertenecientes de **Viriplantae**

```
./famdb.py -i Libraries/famdb families --format fasta_name --ancestors  
--descendants 'Viriplantae' --include-class-in-name >  
viriplantae_repeats.fasta
```

3. REPBASE PREPARATION

ANTES



all_repbase.fasta

```
>ZOMBI_C      Mariner/Tc1 Eutheria
>CAA@1
>AluMacYb4   SINE1/7SL   Macaca
>RPS2        Pseudogene  Homo sapiens
```

DESPUES

```
>ZOMBI_C_Mariner/Tc1_Eutheria#DNA/TcMar-Tigger
>CAA_1#Simple_repeat
>AluMacYb4_SINE1/7SL_Macaca#SINE/Alu
>RPS2_Pseudogene_Homo_sapiens#Unknown
```

Header / Formato a desear

The recommended format for IDs in a custom library is:

```
>repeatname#class/subclass
or simply
>repeatname#class
```



RMRBMeta.embl

```
ID  Mariner1_CB      repeatmasker; DNA;
CC  Mariner1_CB  DNA
XX
DE  RepbaseID: MARINER1_CB
XX
KW  DNA/TcMar-Tc1.
XX
XX
CC  RepeatMasker Annotations:
CC  Type: DNA
CC  SubType: TcMar-Tc1
CC  Species: Caenorhabditis briggsae
```

Asocia **RepbaseID** con **Type** y
SubType

3. REPBASE PREPARATION

1

Asociar clase y subclase con el RepbaseID del archivo “RMRBMeta.embl”

```
def read_repbase_annotations(file):
    """
    Esta funcion lee un archivo de anotaciones de RepBase en formato EMBL y extrae las anotaciones
    de interes, como el ID original, el ID de RepBase, el tipo y subtipo de cada entrada.
    Devuelve un diccionario con estas anotaciones, donde las claves son los IDs en minusculas
    y los valores son diccionarios con las anotaciones correspondientes.
    """

    annotations = {}
    with open(file, 'r') as f:
        entry = None
        for line in f:
            if line.startswith('ID'):
                entry = {}
                rep_id = line.split()[1].strip()
                entry['OriginalID'] = rep_id
            elif line.startswith('DE'):
                repbase_id = line.split(': ')[1].strip()
                entry['ID'] = repbase_id
            elif line.startswith('CC      Type'):
                entry['Type'] = line.split(': ')[1].strip()
            elif line.startswith('CC      SubType'):
                entry['SubType'] = line.split(': ')[1].strip()
            elif line.startswith('///'):
                if entry:
                    # Usa RepbaseID como clave si esta disponible, de lo contrario usa el ID original
                    key = entry.get('ID', entry.get('OriginalID')).lower()
                    annotations[key] = entry
                    entry = None
```

3. REPBASE PREPARATION

1

Para asociar clase y subclase con el RepbaseID del archivo “RMRBMeta.embl”

```
# Funcion para modificar el archivo FASTA
def modify_fasta(file, annotations, output):
    """
    Esta funcion lee un archivo FASTA, busca coincidencias de los encabezados de las secuencias
    con las anotaciones de RepBase y modifica los encabezados agregando el tipo y subtipo de las anotaciones
    correspondientes. Escribe el archivo modificado en un nuevo archivo de salida.
    """

    with open(file, 'r') as f_in, open(output, 'w') as f_out:
        for line in f_in:
            if line.startswith('>'):
                header = line.strip()
                header_lower = header[1:].lower()
                header_parts = header_lower.split()
                key = header_parts[0] # Primera palabra del header
                matched = False

                # Compara la primera palabra del encabezado
                if key in annotations:
                    header += f" #{annotations[key]['Type']}/{annotations[key]['SubType']}"
                    matched = True
                else:
                    # Compara las primeras 5 letras con cualquier clave que comience con esas letras
                    for annotation_key in annotations:
                        if annotation_key.startswith(key[:5]):
                            header += f" #{annotation_key}/{annotations[annotation_key]['SubType']}"
                            matched = True
                            break

                if matched:
                    f_out.write(header + '\n')
                    sequence = ''
                    for line in f_in:
                        if line.startswith('>'):
                            break
                        sequence += line
                    f_out.write(sequence)
```

3. REPBASE PREPARATION

1

Para asociar clase y subclase con el RepbaseID del archivo “RMRBMeta.embl”

```
def read_repbase_annotations(f):
    """
    Esta función lee un archivo FASTA, extrae los encabezados de las secuencias que no contienen el símbolo '#', y los valora
    """
    annotations = []
    with open(f) as f_in:
        entry = None
        for line in f_in:
            if line.startswith('>'):
                if entry:
                    annotations.append(entry)
                entry = {}
                entry['header'] = line[1:]
            elif line.startswith('#'):
                entry['comment'] = line[1:]
            elif line.startswith(' '):
                entry['sequence'] = entry.get('sequence', '') + line
            elif line.startswith('/'):
                entry['subclasses'] = entry.get('subclasses', []) + [line[1:]]
            elif line.startswith('!'):
                entry['classes'] = entry.get('classes', []) + [line[1:]]
            elif line.startswith('@'):
                entry['id'] = line[1:]
            elif line.startswith('*'):
                entry['length'] = int(line[1:])
            elif line.startswith('<'):
                entry['subclasses'].append(line[1:])
            elif line.startswith('>'):
                entry['subclasses'].append(line[1:])
            elif line.startswith('>>'):
                entry['subclasses'].append(line[1:])
            elif line.startswith('>>>'):
                entry['subclasses'].append(line[1:])

    return annotations
```

Función para modificar el archivo FASTA

```
def modify_fasta(file, annotations, output):
```

Esta función lee un archivo FASTA, extrae los encabezados de las secuencias que no contienen el símbolo '#', y los valora

```
def extract_headers_without_hash(fasta_file, output_file):
```

Esta función lee un archivo FASTA, extrae los encabezados de las secuencias que no contienen el símbolo '#', y los valora

Parámetros:

- fastas_file (str): El nombre del archivo FASTA de entrada.
- output_file (str): El nombre del archivo donde se guardarán los encabezados sin '#'.

Retorno:

- None

```
headers_without_hash = []
```

```
with open(fasta_file, 'r') as f:
    for line in f:
        if line.startswith('>') and '#' not in line:
            headers_without_hash.append(line.strip())
```

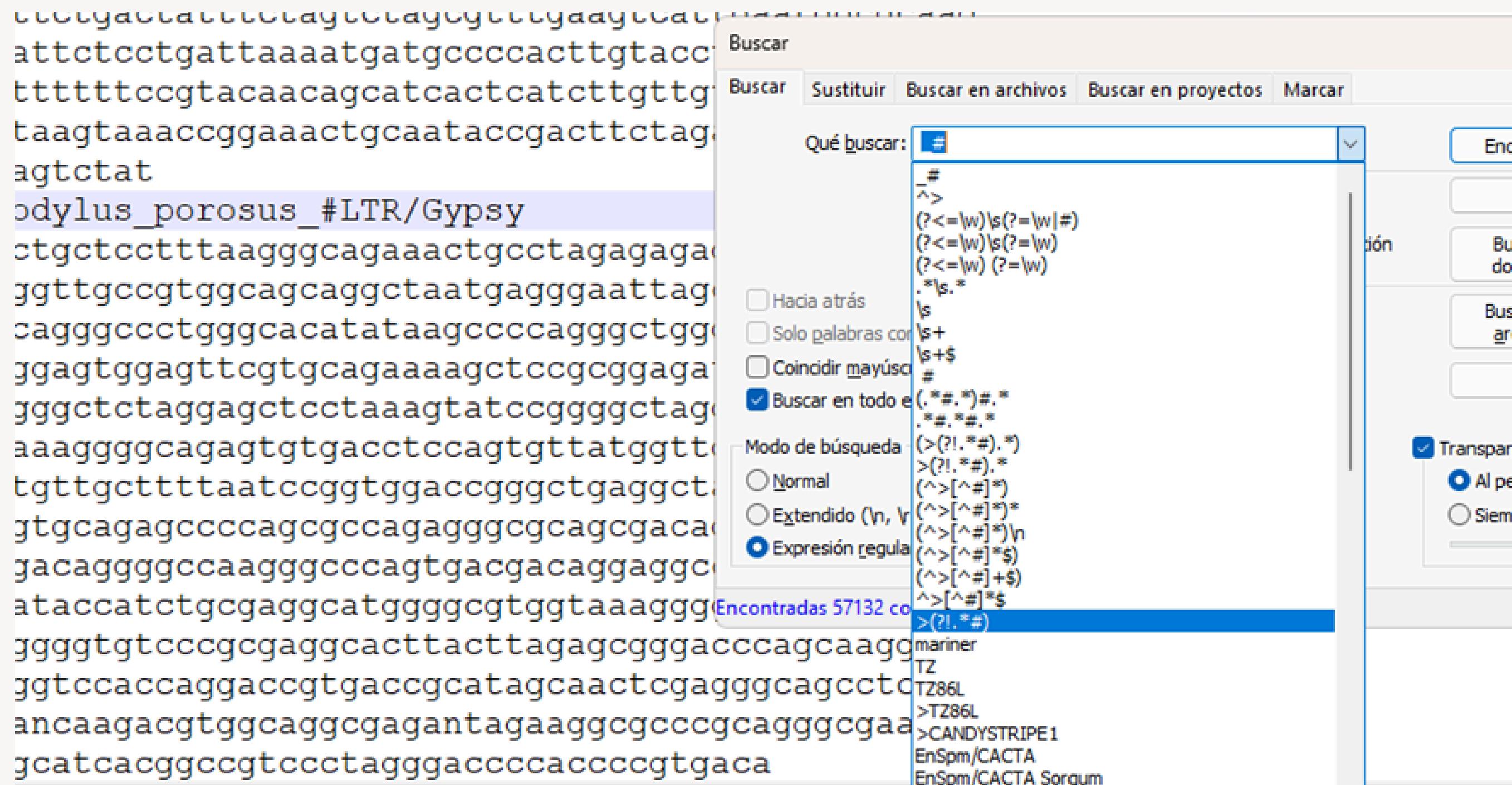
```
with open(output_file, 'w') as f_out:
    for header in headers_without_hash:
        f_out.write(header + '\n')
```

3. REPBASE PREPARATION

2

Limpiar los encabezados

Edición de espacios en blanco y disposición de símbolos



3. REPBASE PREPARATION

2

Limpiar los encabezados

Edición de espacios en blanco y disposición de símbolos

The screenshot shows a code editor with a dark theme. On the left, there is a sequence of DNA bases (A, T, C, G) in a FASTA file format. To the right of the sequence is a Python script. The script defines a function `clean_fasta_headers` that takes two parameters: `input_file` and `output_file`. The function documentation states that it reads a FASTA file, removes headers, trims whitespace, and writes the sequences to a new output file. It includes parameter descriptions for `input_file` and `output_file`, and a return value of `None`. The script uses `open` statements to read from the input file and write to the output file, and a `for` loop to iterate through each line of the input file. Inside the loop, it checks if a line starts with a greater-than symbol (`>`). If so, it removes trailing whitespace and replaces spaces with underscores. The script then continues to read the sequence lines. At the bottom of the editor, there is a status bar showing the file path and some file statistics.

```
def clean_fasta_headers(input_file, output_file):
    """
    Esta funcion lee un archivo FASTA, limpia los encabezados de las secuencias removiendo espacios y barras al final,
    reemplaza los espacios con guiones bajos, y escribe las secuencias con los encabezados limpiados en un nuevo archivo de salida.

    Parametros:
    input_file (str): El nombre del archivo FASTA de entrada.
    output_file (str): El nombre del archivo donde se guardaran las secuencias con encabezados limpiados.

    Retorno:
    None
    """
    with open(input_file, 'r') as infile, open(output_file, 'w') as outfile:
        for line in infile:
            if line.startswith('>'):
                # Remover espacios y barras al final del encabezado
                clean_line = line.rstrip().rstrip('/')
                # Reemplazar espacios con guiones bajos
                outfile.write(clean_line + '\n')
            else:
                outfile.write(line)
```

3. REPBASE PREPARATION

3

Eliminar secuencias
duplicadas y otros ajustes

```
def elim_dup(fasta_file):
    """
    Esta función remueve las secuencias duplicadas en un archivo FASTA.

    Parámetros:
    fasta_file (str): El nombre del archivo FASTA de entrada.

    Retorno:
    None
    """

    output_file = fasta_file.replace('.fasta', '_full_line_unique.fasta')
    seen_lines = set()
    write_line = True
    with open(fasta_file, 'r') as infile, open(output_file, 'w') as outfile:
        for line in infile:
            if line.startswith('>'): # Verifica duplicados en las líneas de encabezado
                full_header = line.strip().upper() # Elimina espacios finales
                if full_header in seen_lines:
                    write_line = False # Si el encabezado ya está en el set, no escribe línea
                else:
                    seen_lines.add(full_header.upper())
            else:
                outfile.write(line)
```

3. REPBASE PREPARATION

3

Eliminar secuencias duplicadas y otros ajustes

Reemplazar símbolos extraños a “ATGC” por “N”.

```
def elim_dup(fasta_file):
    """
    Esta función reemplaza cualquier letra que no sea ATGC por N en un archivo FASTA.

    Parámetros:
    fasta (str): El nombre del archivo FASTA de entrada.

    Retorno:
    None
    """

    output_file = fasta.replace('.fasta', '_replaced.fasta')
    with open(fasta, 'r') as infile, open(output_file, 'w') as outfile:
        for line in infile:
            if line.startswith('>'): # Si la línea es un encabezado, la copia tal cual
                outfile.write(line) # Elimina los espacios finales y agrega un salto de línea
            else: # Si no es un encabezado, realiza la sustitución
                updated_line = line.upper() # Convierte a mayúsculas
                updated_line = re.sub(r'[^ATGC\n]', 'N', updated_line) # Reemplaza cualquier carácter que no sea ATGC por N
                outfile.write(updated_line) # Elimina los espacios finales y agrega un salto de línea
    print(f"Archivo fasta limpio guardado como: {output_file}")

def replace(fasta):
    """
    Esta función reemplaza cualquier letra que no sea ATGC por N en un archivo FASTA.

    Parámetros:
    fasta (str): El nombre del archivo FASTA de entrada.

    Retorno:
    None
    """

    output_file = fasta.replace('.fasta', '_replaced.fasta')
    with open(fasta, 'r') as infile, open(output_file, 'w') as outfile:
        for line in infile:
            if line.startswith('>'): # Si la línea es un encabezado, la copia tal cual
                outfile.write(line) # Elimina los espacios finales y agrega un salto de línea
            else: # Si no es un encabezado, realiza la sustitución
                updated_line = line.upper() # Convierte a mayúsculas
                updated_line = re.sub(r'[^ATGC\n]', 'N', updated_line) # Reemplaza cualquier carácter que no sea ATGC por N
                outfile.write(updated_line) # Elimina los espacios finales y agrega un salto de línea
    print(f"Archivo fasta limpio guardado como: {output_file}")
```

2.B TRF INSTALLATION

parámetros para considerar en el código para ejecutar TRF.

```
Please use: trf409.dos64.exe File Match Mismatch Delta PM PI Minscore MaxPeriod [options]

Where: (all weights, penalties, and scores are positive)
  File = sequences input file
  Match = matching weight
  Mismatch = mismatching penalty
  Delta = indel penalty
  PM = match probability (whole number)
  PI = indel probability (whole number)
  Minscore = minimum alignment score to report
  MaxPeriod = maximum period size to report
  [options] = one or more of the following:
    -m      masked sequence file
    -f      flanking sequence
    -d      data file
    -h      suppress html output
    -r      no redundancy elimination
    -l <n>  maximum TR length expected (in millions) (eg, -l 3 or -l=3 for 3 million)
            Human genome HG38 would need -l 6

See more information on the TRF Unix Help web page: https://tandem.bu.edu/trf/trf.unix.help.html

Note the sequence file should be in FASTA format:

>Name of sequence
aggaaaacctgccatggcctctggtagactgtccatccactgtcgctgccttcag
atactctgaccatggatccccctgggtgcagccaagccacaatggccatggcgccgtgt
actccccaccccccacccctcctgatecctgcatggacatggccttccacatccctgtg
```

- La mayoría de parámetros del código se usaron por defecto.
- Se realizaron pruebas del código para refinar la búsqueda de secuencias TANDEM final.

comando por defecto

>trf409.dos64.exe genomeconc.fasta 2 5 7 80 10 50 2000 -l 10

comando modificado

>trf409.dos64.exe genomeconc.fasta 1 1 10 80 10 50 6 -l 10

BLAST

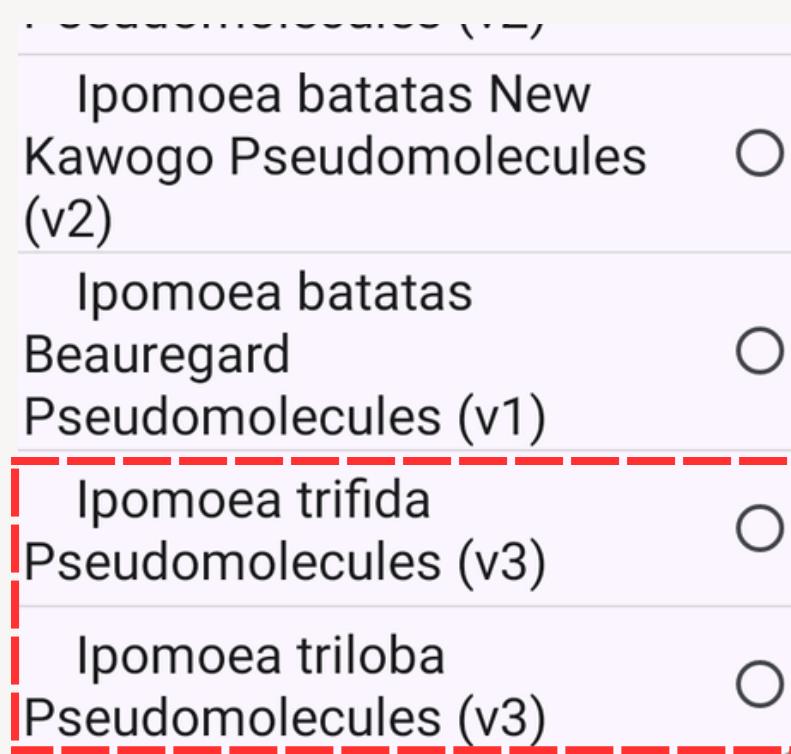
1. Se accedió a la herramienta BLAST de la organización Sweetpotato Genomic Resources

<http://sweetpotato.uga.edu/blast.shtml>

2. Se realizó un blastn

Se seleccionaron como base de datos *I. trifida* e *I. triloba*

Los demás parámetros se establecieron por defecto.



3 . El output es un archivo ".blast" en el que se analizaron las coincidencias significativas.

Sweetpotato Genomics Resource

Home Genome Browser Search Tools Download Links Contact

Sweetpotato Genomics Resource BLAST Server

The Sweetpotato Genomics Resource BLAST server allows searching the genome assemblies and the genome annotation released by the GT4SP and SweetGAINS projects.

Select type of BLAST Search: blastn

Select database: Ipomoea trifida transcripts - HC Set (v3)

Enter Query Sequence: Enter FASTA sequence: [Large input field]

Clear Form

Optional BLAST Parameters:

Expect Threshold: 1e-5

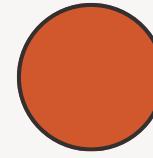
Max Number of Alignments Reported: 10

Word Length: 11

Filter: none

Submit BLAST Search: Search using blastn

Submit

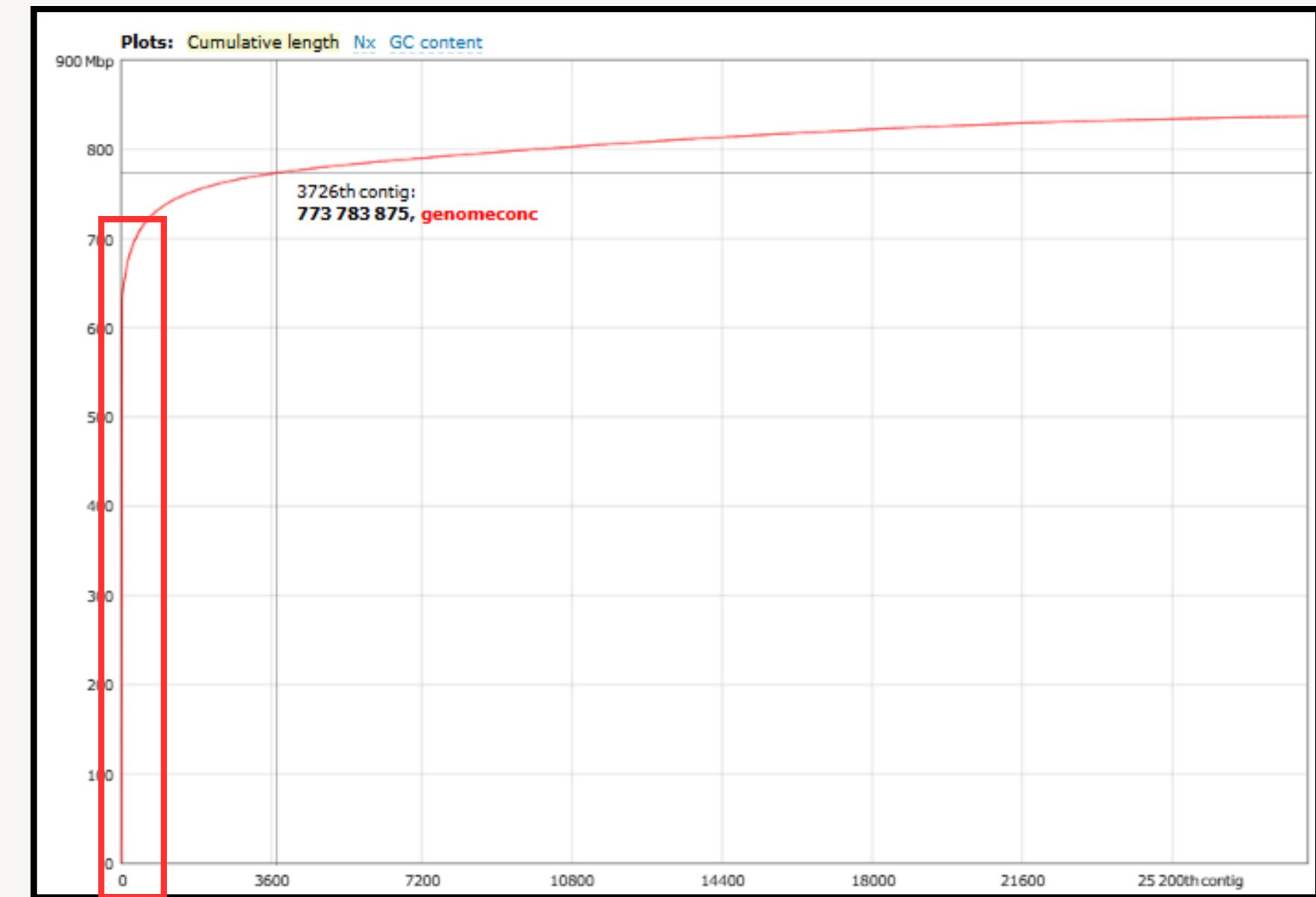


RESULTS

1. GENERAL ASSEMBLY STATISTICS (QUAST)

- La mayor parte de la longitud del genoma se cubre con un número reducido de contigs

Statistics without reference genomeconc	
# contigs	28 454
# contigs (>= 0 bp)	28 461
# contigs (>= 1000 bp)	25 620
# contigs (>= 5000 bp)	5019
# contigs (>= 10000 bp)	2640
# contigs (>= 25000 bp)	1227
# contigs (>= 50000 bp)	657
Largest contig	52 844 149
Total length	837 010 047
Total length (>= 0 bp)	837 013 208
Total length (>= 1000 bp)	834 766 598
Total length (>= 5000 bp)	781 103 081
Total length (>= 10000 bp)	765 089 693
Total length (>= 25000 bp)	743 099 674
Total length (>= 50000 bp)	723 216 748
N50	41 463 214
N75	32 277 352
L50	10
L75	15
GC (%)	35.15
Matches	
# N's	100 978 753
# N's per 100 kbp	12 064



1. GENERAL ASSEMBLY STATISTICS (QUAST)

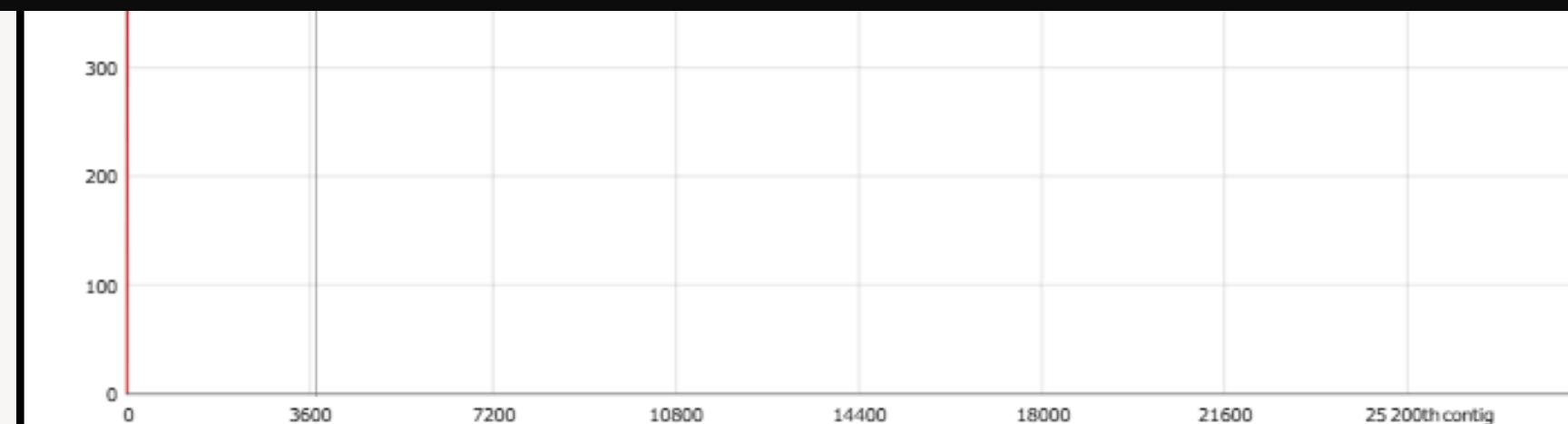
- La mayor parte de la longitud del genoma se cubre con un número reducido de contigs
- Existen 15 cromosomas reconstruidos

Statistics without reference genomeconc	
# contigs	28 454
# contigs (>= 0 bp)	28 461
# contigs (>= 1000 bp)	25 620



```
(summer) bradford@bradford:~/Documents/bioinfo$ assembly-stats -s -l 650000 genomeconc.fasta
total_length 633423954
number 15
mean_length 42228263.60
longest 52844149
shortest 32277352
N_count 83283976
```

Total length (>= 20000 bp)	718899871
Total length (>= 50000 bp)	723 216 748
N50	41 463 214
N75	32 277 352
L50	10
L75	15
GC (%)	35.15
Mismatches	
# N's	100 978 753
# N's per 100 kbp	12 064



RUN REPEATMASKER

- Total de elementos repetitivos: se obtuvo 423 millones bp (50.64%) vs. 382 millones bp (45.619%)
- Gran cantidad de retroelementos (LTR como Ty1/Copia) y secuencias desconocidas (aprox. 22%)
- Cantidad moderada de Simple Repeats: se obtuvo 245 mil elementos (1.17%) vs 324 mil (1.740%)

Publicación: RepeatModeler

Table 2 | Summary of repeat sequence identification

Type of elements	Number of elements	Length occupied(bp)	Percentage of genome*
LTR	213,439	92,066,503	10.987
DNA elements	256,260	50,863,135	6.070
LINE	40,146	20,814,265	2.484
Simple repeat	324,390	14,578,880	1.740
RC/Helitron	12,067	5,610,834	0.670
Low complexity	45,912	2,324,352	0.277
SINE	4,168	618,723	0.074
rRNA	322	97,450	0.012
Satellite	456	33,443	0.004
snRNA	219	29,584	0.004
Unknown	985,978	195,232,724	23.299
Total	1,883,357	382,269,893	45.619

(Yang et al., 2017)

	number of elements*	length occupied	percentage of sequence
Retroelements	440005	157434983 bp	18.81 %
SINES:	60647	6995137 bp	0.84 %
Penelope:	2812	437399 bp	0.05 %
LINEs:	60605	23963438 bp	2.86 %
L1/CIN4	43371	22059232 bp	2.64 %
LTR elements:	318753	126476408 bp	15.11 %
BEL/Pao	2363	267860 bp	0.03 %
Ty1/Copia	151381	61948684 bp	7.40 %
Gypsy/DIRS1	146353	61145368 bp	7.31 %
Retroviral	3543	572885 bp	0.07 %
DNA transposons	312501	59302071 bp	7.08 %
hobo-Activator	86205	15456922 bp	1.85 %
Tc1-I5630-Pogo	7912	761070 bp	0.09 %
En-Spm	0	0 bp	0.00 %
MULE-MuDR	87145	20521028 bp	2.45 %
PiggyBac	248	15805 bp	0.00 %
Tourist/Harbinger	45153	7666218 bp	0.92 %
Other (Mirage, P-element, Transib)	457	24391 bp	0.00 %
Rolling-circles	36587	10804194 bp	1.29 %
Unclassified:	1005838	178668080 bp	21.35 %
Total interspersed repeats:	395842533 bp	47.29 %	
Small RNA:	94516	12084885 bp	1.44 %
Satellites:	1797	158874 bp	0.02 %
Simple repeats:	245053	9807686 bp	1.17 %
Low complexity:	34697	1653991 bp	0.20 %
bases masked:	423896995 bp	(50.64 %)	

Nuestro Resultado: RepeatMasker

RUN REPEATMASKER

- El genoma de *Ipomoea batatas* es adecuado para la identificación de microsatélites
- Las diferencias en los resultados pueden deberse a los métodos utilizados; RepeatModeler es más exhaustivo en la identificación de nuevas repeticiones, mientras que RepeatMasker se basa en bases de datos conocidas.

Publicación: RepeatModeler

Table 2 | Summary of repeat sequence identification

Type of elements	Number of elements	Length occupied(bp)	Percentage of genome*
LTR	213,439	92,066,503	10.987
DNA elements	256,260	50,863,135	6.070
LINE	40,146	20,814,265	2.484
Simple repeat	324,390	14,578,880	1.740
RC/Helitron	12,067	5,610,834	0.670
Low complexity	45,912	2,324,352	0.277
SINE	4,168	618,723	0.074
rRNA	322	97,450	0.012
Satellite	456	33,443	0.004
snRNA	219	29,584	0.004
Unknown	985,978	195,232,724	23.299
Total	1,883,357	382,269,893	45.619

(Yang et al., 2017)

	number of elements*	length occupied	percentage of sequence
Retroelements	440005	157434983 bp	18.81 %
SINES:	60647	6995137 bp	0.84 %
Penelope:	2812	437399 bp	0.05 %
LINEs:	60605	23963438 bp	2.86 %
L1/CIN4	43371	22059232 bp	2.64 %
LTR elements:	318753	126476408 bp	15.11 %
BEL/Pao	2363	267860 bp	0.03 %
Ty1/Copia	151381	61948684 bp	7.40 %
Gypsy/DIRS1	146353	61145368 bp	7.31 %
Retroviral	3543	572885 bp	0.07 %
DNA transposons	312501	59302071 bp	7.08 %
hobo-Activator	86205	15456922 bp	1.85 %
Tc1-I5630-Pogo	7912	761070 bp	0.09 %
En-Spm	0	0 bp	0.00 %
MULE-MuDR	87145	20521028 bp	2.45 %
PiggyBac	248	15805 bp	0.00 %
Tourist/Harbinger	45153	7666218 bp	0.92 %
Other (Mirage, P-element, Transib)	457	24391 bp	0.00 %
Rolling-circles	36587	10804194 bp	1.29 %
Unclassified:	1005838	178668080 bp	21.35 %
Total interspersed repeats:	395842533	bp	47.29 %
Small RNA:	94516	12084885 bp	1.44 %
Satellites:	1797	158874 bp	0.02 %
Simple repeats:	245053	9807686 bp	1.17 %
Low complexity:	34697	1653991 bp	0.20 %

Nuestro Resultado: RepeatMasker

TRF

TANDEM REPEATS FINDER

Secuencias TANDEM identificadas

Please cite:
G. Benson,
"Tandem repeats finder: a program to analyze DNA sequences
Nucleic Acid Research(1999)
Vol. 27, No. 2, pp. 573-580.

archivo summary

Multiple Sequence Summary

Only sequences containing repeats are shown!

[Click on sequence description to view repeat table.](#)

Sequence Index	Sequence Description	Number of Repeats
1	<u>NXFB01000001.1 Ipomoea batatas cultivar Taizhong6 chromosome 1 chr1, whole genome shotgun sequence</u>	1198
2	<u>NXFB01000002.1 Ipomoea batatas cultivar Taizhong6 chromosome 2 chr2, whole genome shotgun sequence</u>	1629
3	<u>NXFB01000003.1 Ipomoea batatas cultivar Taizhong6 chromosome 3 chr3, whole genome shotgun sequence</u>	1055
4	<u>NXFB01000004.1 Ipomoea batatas cultivar Taizhong6 chromosome 4 chr4, whole genome shotgun sequence</u>	1644
5	<u>NXFB01000005.1 Ipomoea batatas cultivar Taizhong6 chromosome 5 chr5, whole genome shotgun sequence</u>	1533
6	<u>NXFB01000006.1 Ipomoea batatas cultivar Taizhong6 chromosome 6 chr6, whole genome shotgun sequence</u>	1535
7	<u>NXFB01000007.1 Ipomoea batatas cultivar Taizhong6 chromosome 7 chr7, whole genome shotgun sequence</u>	1225
8	<u>NXFB01000008.1 Ipomoea batatas cultivar Taizhong6 chromosome 8 chr8, whole genome shotgun sequence</u>	1429
9	<u>NXFB01000009.1 Ipomoea batatas cultivar Taizhong6 chromosome 9 chr9, whole genome shotgun sequence</u>	1247
10	<u>NXFB01000010.1 Ipomoea batatas cultivar Taizhong6 chromosome 10 chr10, whole genome shotgun sequence</u>	1535
11	<u>NXFB01000011.1 Ipomoea batatas cultivar Taizhong6 chromosome 11 chr11, whole genome shotgun sequence</u>	1438

Length: 3602548

Tables: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

This is table 1 of 10 (1198 repeats found)

Click on indices to view alignment

Table Explanation

Indices: 41272--41415 Score: 98
Period size: 3 Copynumber: 48.0 Consensus size

Alignment explanation

Statistics

Matches: 115, Mismatches: 26, Indels: 0
0.82 0.18 0.00

Matches are distributed among these distances

3 115 1.0

ACGTcount: A:0.66, C:0.04, G:0.02, T:0.28

Consensus pattern (3 bp): TAA

FINAL SSR

Se filtraron las secuencias repetidas en Tándem y se procedió a encontrar los mejores candidatos a SSR.

chr	Posición	Motivo	PeriodSize	Tamaño total del SSR	Score	Percent Indels
1	96293-96334	(ATA)14	3	42	50	0
2	231084, 231113	(ATTGTT)5	6	30	51	0
3	1354277-1354327	(TTA)17	3	51	51	0
4	82024-82077	(ATT)18	3	54	56	0
5	1771121-1771174	(AATAAT)9	6	54	55	0
6	628374-628417	(TAT)12	3	36	53	0
7	2782385-2782441	(TAT)19	3	57	58	0
8	700815-700865	(TAA)17	3	51	52	0
9	392419-392454	(ATAT)9	4	36	51	0
10	10924503-10924550	(ATA)16	3	48	54	0
11	901637--901684	(TCA)16	3	48	50	0
12	413271-413324	(ATT)18	3	54	64	0
13	3621855-3621898	(ATAT)11	4	44	56	0
14	114839-114889	(TTA)17	3	51	54	0
15	2500907-2500960	(TAA)18	3	54	54	0

BLAST

Se blastearon contra las secuencias de otros dos géneros de Ipomoea: *I. trifida* e *I. triloba*

I. trifida

Length = 32247286
Score = 105 bits (53), Expect = 5e-22
Identities = 53/53 (100%)
Strand = Plus / Plus

Query: 16 tcatcatcatcatcatcatcatcatcatcatcatcatcatcatcatcttt 68
 ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| |||||
Sbjct: 11027363 tcatcatcatcatcatcatcatcatcatcatcatcatcatcatcttt 11027415

I. triloba

BLAST

Se blastearon contra las secuencias de otros dos géneros de Ipomoea: *I. trifida* e *I. triloba*

I. trifida

And so on for the other SSRs.

Identities = 59/60 (98%)
Strand = Plus / Plus

Identities = ...
Strand = Plus / Plus

Identities
Strand = Plus / Plus

Query: 1

Score Ident

Score = 99.6 bits (50), Expect =
Identities = 50/50 (100%)
Strand = Plus / Plus

Query: 1 agcacaac score = 99.6 bits (50), Expect =
Sbjct: 3552805 agcacaac Identities = 50/50 (100%)
Strand = Plus / Plus

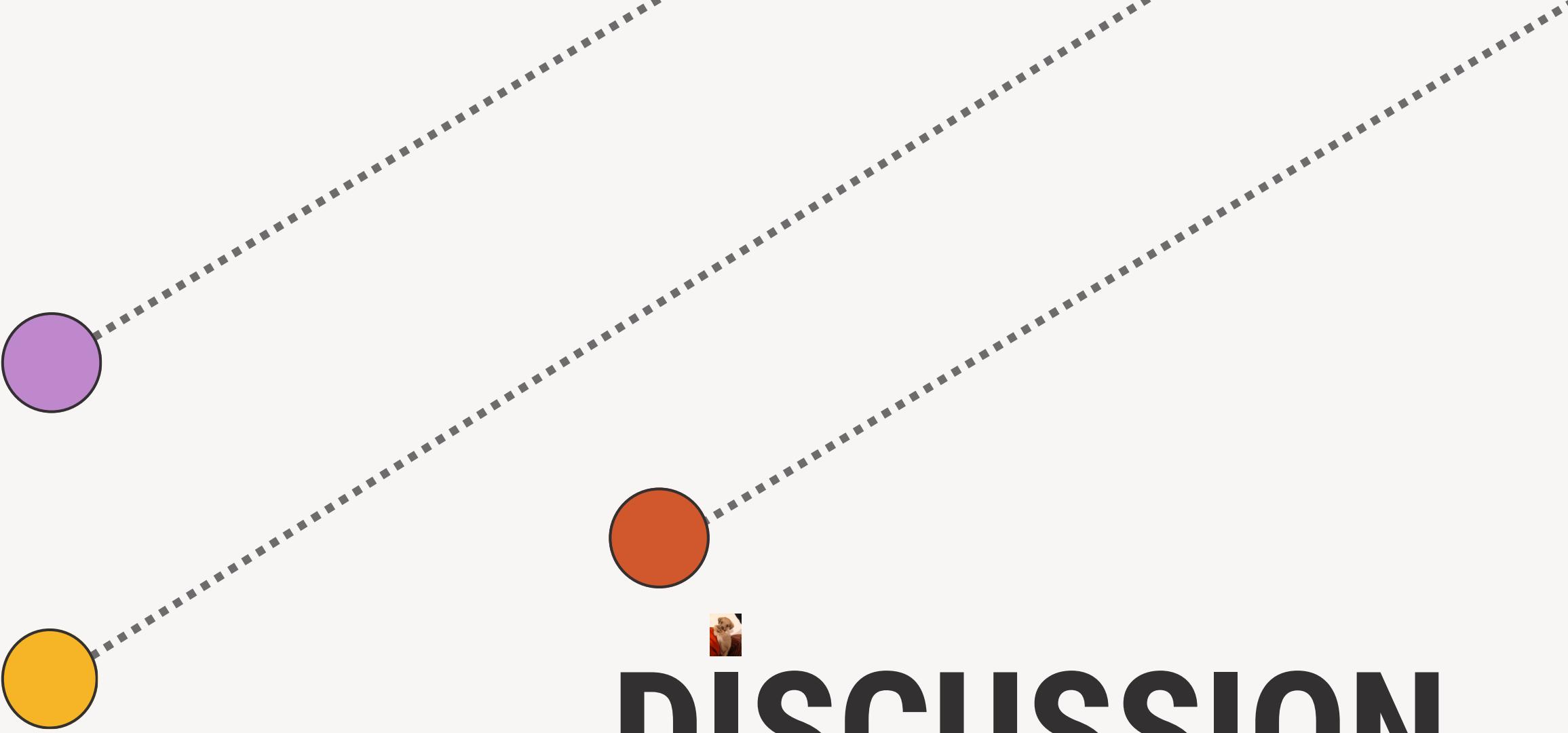
Score: *Query: 16*

Strasse *Sbjekt:* 119.

JRC-1102

jQuery .

Subjct: 23302054 aa



DISCUSSION

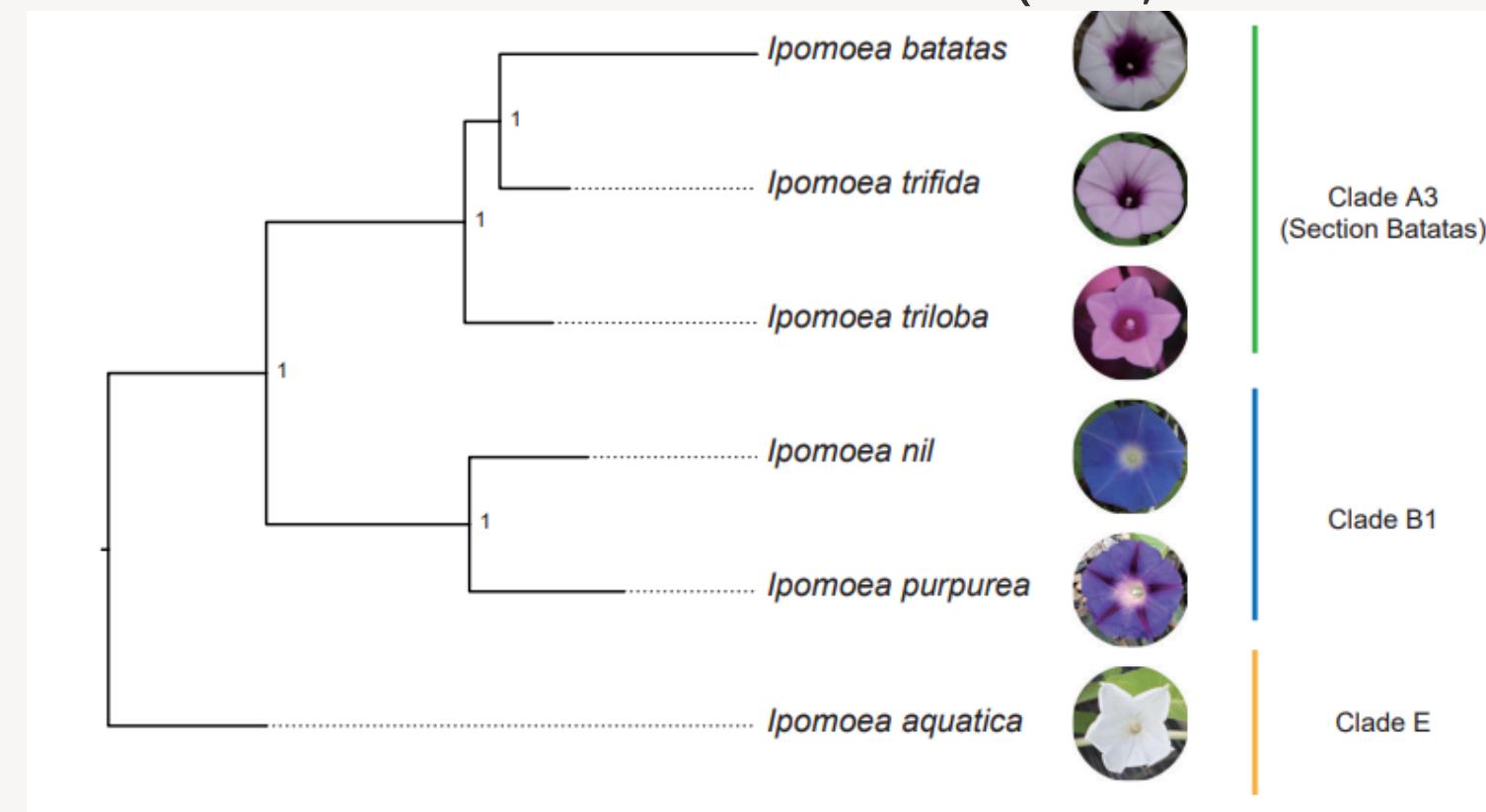


DISCUSSION

Referencia	Marcadores seleccionados	Accesiones y resultados
Hu et al., 2004	12 SSR de <i>I. trifida</i>	Los primers desarrollados fueron capaces de amplificar el ADN genómico de <i>I. batata</i> y otras tres especies silvestres relacionadas.
Schafleitner et al., 2010	223 loci SSR en <i>I.batatas</i>	6 accesiones hexaploides de <i>I. batatas</i> , 2 diploides de <i>I. trifida</i> . Mayoría de loci amplificados en ambas especies
Buteler et al., 1999	63 loci SSR en <i>I.batatas</i>	2 poblaciones hexaploides de <i>I. batatas</i> , 1 tetraploide y 1 diploide de <i>I. trifida</i> . Amplificación en <i>I. batatas</i> y productos similares en <i>I. trifida</i>

DISCUSSION

(Yan, et. al. 2022)



- *I. batatas*, *I. trifida*, and *I. triloba* form the same clade.
- According to molecular data, *I. batata*, *I. trifida*, and *I. triloba* are closely related.

(McComish et. al., 2024)

Ancient and Modern Genomes Reveal Microsatellites Maintain a Dynamic Equilibrium Through Deep Time

Bennet J. McComish ^{1,2}, Michael A. Charleston ¹, Matthew Parks ^{3,4}, Carlo Baroni ^{5,6}, Maria Cristina Salvatore ^{5,6}, Ruiqiang Li ⁷, Guojie Zhang ^{8,9}, Craig D. Millar ¹⁰, Barbara R. Holland ¹, and David M. Lambert ^{3,*}

¹School of Natural Sciences, University of Tasmania, Hobart, TAS 7001, Australia

²Menzies Institute for Medical Research, University of Tasmania, Hobart, TAS 7001, Australia

³Australian Research Centre for Human Evolution, Griffith University, Nathan, QLD 4111, Australia

⁴Department of Biology, University of Central Oklahoma, Edmond, OK 73034, USA

⁵Dipartimento di Scienze della Terra, University of Pisa, Pisa, Italy

⁶CNR-IGG, Institute of Geosciences and Earth Resources, Pisa, Italy

⁷Novogene Bioinformatics Technology Co. Ltd., Beijing 100083, China

⁸China National GeneBank, BGI-Shenzhen, Shenzhen 518083, China

⁹Department of Biology, Centre for Social Evolution, University of Copenhagen, Copenhagen DK-2100, Denmark

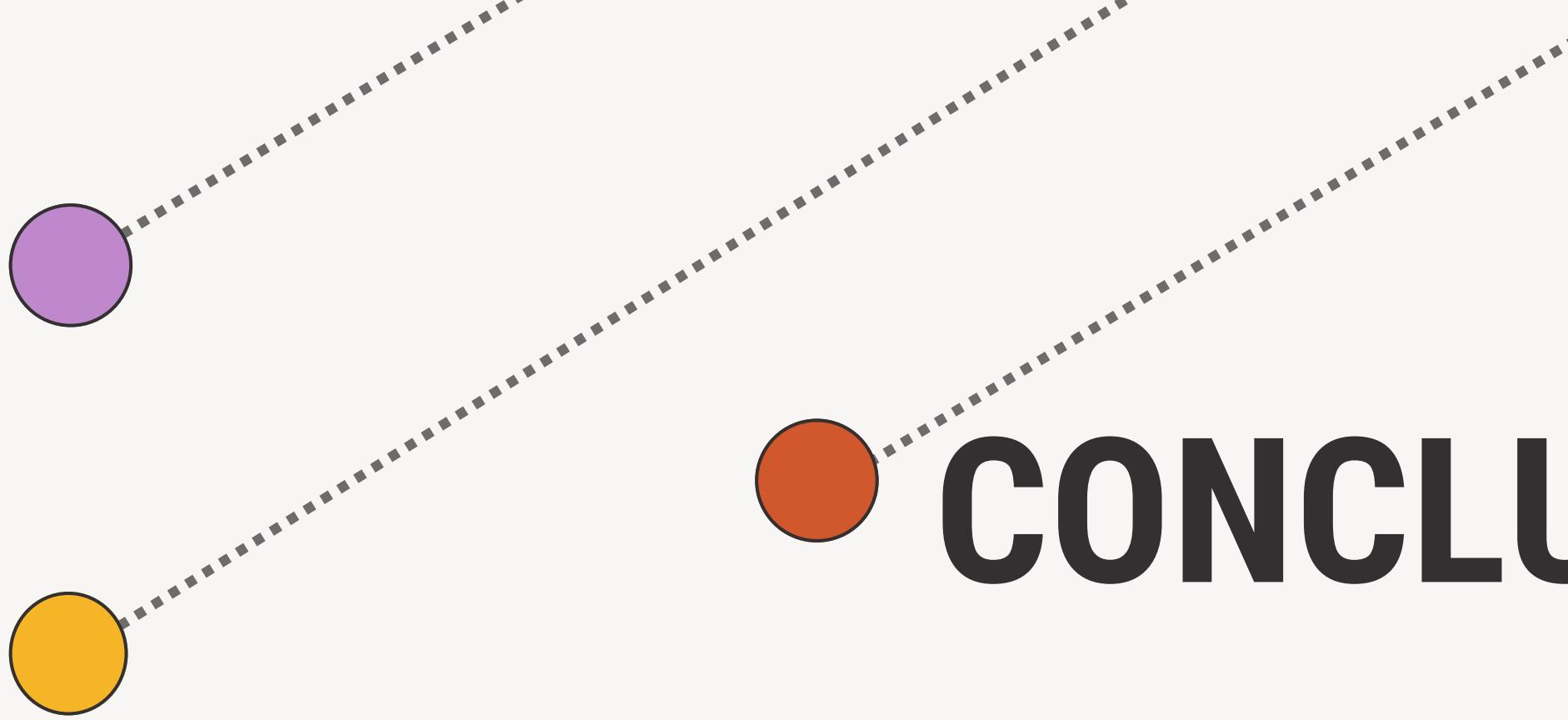
¹⁰School of Biological Sciences, University of Auckland, Auckland, New Zealand

*Corresponding author: E-mail: d.lambert@griffith.edu.au.

Accepted: January 23, 2024

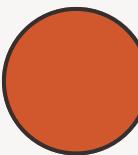
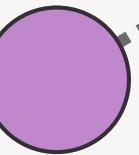
Abstract

- Microsatellites can be found in UTRs (non-coding regions) that regulate gene expression.
- Therefore, they can influence the phenotype and be conserved over evolutionary time.



CONCLUSIONS

- The reconstructed chromosomes are a good representation of the genome. However, the high number of N bases makes it difficult to properly identify repeats.
- A high proportion of retroelements (18.8%) and unknown elements (approx. 22%) were identified in *Ipomoea batatas*. There was also a moderate amount of microsatellites (1.17%).
- Fifteen candidate microsatellites were identified after filtering based on their location, alignment score, percentage of divergence, and number of repeats.
- The ability of SSRs to serve as markers in multiple species indicates a close genetic relationship between *I. batatas*, *I. trifida*, and *I. triloba*. This facilitates comparative studies and genetic conservation in related species.



REFERENCES

- Álvarez-Cubero, M., Mtnez-Gonzalez, L., Saiz, M., Álvarez, J., & Lorente, J. (s. f.-c). Nuevas aplicaciones en identificación genética. https://scielo.isciii.es/scielo.php?pid=S1135-7606201000100002&script=sci_arttext
- Aranguren Méndez, J. A., Román Bravo, R., Isea, W., Villasmil, Y., & Jordana, y. J. (2003). Los microsatélites (STR's), marcadores moleculares de ADN por excelencia para programas de conservación: una revisión. Utoronto.ca. <https://tspace.library.utoronto.ca/bitstream/1807/7085/1/la05005.pdf>
- Billingsley, K. J., Lättekivi, F., Planken, A., Reimann, E., Kurvits, L., Kadastik-Eerme, L., Kasterpalu, K. M., Bubb, V. J., Quinn, J. P., Kõks, S., & Taba, P. (2019). Analysis of repetitive element expression in the blood and skin of patients with Parkinson's disease identifies differential expression of satellite elements. *Scientific Reports*, 9(1), 1–9. <https://doi.org/10.1038/s41598-019-40869-z>
- Buteler, M., Jarret, R. & LaBonte, D. Sequence characterization of microsatellites in diploid and polyploid Ipomoea. *Theor Appl Genet* 99, 123–132 (1999). <https://doi.org/10.1007/s001220051216>
- Cadima, X., Veramendi, S., & Gabriel, J. (s. f.). Uso de marcadores moleculares microsatelite para el análisis de la diversidad genética de papa nativa de Bolivia. http://www.scielo.org.bo/scielo.php?pid=S2072-92942013000100003&script=sci_arttex
- Ellegren, H. (2004). Microsatellites: simple sequences with complex evolution. *Nature Reviews. Genetics*, 5(6), 435–445. doi:10.1038/nrg1348
- Grüneberg, W., Mwanga, R., Andrade, M., & Espinoza, J. (2009). Selection methods. Part 5: Breeding clonally propagated crops.
- Hu, J., Nakatani, M., Lalusin, A. G., & Fujimura, T. (2004). New microsatellite markers developed from reported *Ipomoea trifida* sequences and their application to sweetpotato and its related wild species. *Scientia Horticulturae*, 102(4), 375–386. doi:10.1016/j.scienta.2004.04.014
- Buteler, M., Jarret, R. & LaBonte, D. Sequence characterization of microsatellites in diploid and polyploid Ipomoea. *Theor Appl Genet* 99, 123–132 (1999). <https://doi.org/10.1007/s001220051216>
- Schafleitner, R., Tincopa, L.R., Palomino, O. et al. A sweetpotato gene index established by de novo assembly of pyrosequencing and Sanger sequences and mining for gene-based microsatellite markers. *BMC Genomics* 11, 604 (2010). <https://doi.org/10.1186/1471-2164-11-604>
- Katayama, K., Kobayashi, A., Sakai, T., Kuranouchi, T., & Kai, Y. (2017). Recent progress in sweetpotato breeding and cultivars for diverse applications in Japan. *Breeding science*, 67(1), 3-14.
- Lebot, V. (2010). Sweet potato. *Root and tuber crops*, 97-125.
- Lee, K. J., Lee, G.-A., Lee, J.-R., Sebastin, R., Shin, M.-J., Cho, G.-T., & Hyun, D. Y. (2019). Genetic diversity of sweet potato (*Ipomoea batatas* L. Lam) germplasms collected worldwide using chloroplast SSR markers. *Agronomy (Basel, Switzerland)*, 9(11), 752. <https://doi.org/10.3390/agronomy9110752>
- Liao, X., Zhu, W., Zhou, J., Li, H., Xu, X., Zhang, B., & Gao, X. (2023). Repetitive DNA sequence detection and its role in the human genome. *Communications Biology*, 6(1). <https://doi.org/10.1038/s42003-023-05322-y>
- Mattei, M. G., & Luciani, J. (2022). Heterocromatina, del Cromosoma a la Proteína. *Atlasgeneticsoncology.org*. <https://atlasgeneticsoncology.org/teaching/209067/heterocromatina-del-cromosoma-a-la-prote-na>
- Madesis, P., Ganopoulos, I., & Tsafaris, A. (2013). Microsatellites: evolution and contribution. *Methods in molecular biology (Clifton,*

- Myers, P. (2007) Tandem repeats and morphological variation. *Nature Education* 1(1):1, <https://www.nature.com/scitable/topicpage/tandem-repeats-and-morphological-variation-40690/>
- Méndez, J., & González-Tizón, A. (2011). LOS GENOMAS EUCARIOTAS: ASPECTOS GENERALES. Udc.es. https://ruc.udc.es/dspace/bitstream/handle/2183/9859/CC_30_art_1.pdf?sequence=1&isAllowed=y
- National Human Genome Research Institute. (2024) Repetición en tandem. *Genome.gov*. <https://www.genome.gov/es/genetics-glossary/Repeticion-en-tandem>
- Provan, J., Powell, W. & Waugh, R. Microsatellite analysis of relationships within cultivated potato (*Solanum tuberosum*). *Theoret. Appl. Genetics* 92, 1078–1084 (1996). <https://doi.org/10.1007/BF00224052>
- Ramesh, P., Mallikarjuna, G., Sameena, S., Kumar, A., Gurulakshmi, K., Reddy, B. V., Reddy, P. C. O., & Sekhar, A. C. (2020). Advancements in molecular marker technologies and their applications in diversity studies. *Journal of biosciences*, 45, 123.
- Rasoarahona, R., Wattanadilokchatkun, P., Panthum, T., Thong, T., Singchat, W., Ahmad, S. F., Chaiyes, A., Han, K., Kraichak, E., Muangmai, N., & et al. (2023). Optimizing microsatellite marker panels for genetic diversity and population genetic studies: An ant colony algorithm approach with polymorphic information content. *Biology*, 12(10), 1280. <https://doi.org/10.3390/biology12101280>
- Sagar, T., Kapoor, N., & Mahajan, R. (2023). Microsatellites as Potential Molecular Markers for Genetic Diversity Analysis in Plants. In *Molecular Marker Techniques: A Potential Approach of Crop Improvement* (pp. 81-101). Singapore: Springer Nature Singapore.
- Wenne R. (2023). Microsatellites as Molecular Markers with Applications in Exploitation and Conservation of Aquatic Animal Populations. *Genes*, 14(4), 808. <https://doi.org/10.3390/genes14040808>
- Xiao, Y., Zhu, M., & Gao, S. (2022). Genetic and Genomic Research on Sweet Potato for Sustainable Food and Nutritional Security. *Genes*, 13(10), 1833. <https://doi.org/10.3390/genes13101833>
- Yan, L., Lai, X., Li, X., Wei, C., Tan, X., & Zhang, Y. (2015). Analyses of the complete genome and gene expression of chloroplast of sweet potato [*Ipomoea batata*]. *PloS One*, 10(4), e0124083. <https://doi.org/10.1371/journal.pone.0124083>
- Yan, M., Nie, H., Wang, Y., Wang, X., Jarret, R., Zhao, J., Wang, H., & Yang, J. (2022). Exploring and exploiting genetics and genomics for sweetpotato improvement: Status and perspectives. *Plant Communications*, 3(5), 100332. <https://doi.org/10.1016/j.xplc.2022.100332>
- Yang, J., Moeinzadeh, M. H., Kuhl, H., Helmuth, J., Xiao, P., Haas, S., Liu, G., Zheng, J., Sun, Z., Fan, W., Deng, G., Wang, H., Hu, F., Zhao, S., Fernie, A. R., Boerno, S., Timmermann, B., Zhang, P., & Vingron, M. (2017). Haplotype-resolved sweet potato genome traces back its hexaploidization history. *Nature plants*, 3(9), 696–703. <https://doi.org/10.1038/s41477-017-0002-z>
- Yang, J., Moeinzadeh, M. H., Kuhl, H., Helmuth, J., Xiao, P., Haas, S., Liu, G., Zheng, J., Sun, Z., Fan, W., Deng, G., Wang, H., Hu, F., Zhao, S., Fernie, A. R., Boerno, S., Timmermann, B., Zhang, P., & Vingron, M. (2017). Haplotype-resolved sweet potato genome traces back its hexaploidization history. *Nature plants*, 3(9), 696–703. <https://doi.org/10.1038/s41477-017-0002-zperspectives>
- Ghislain, M., Magembe, E., Amaro, G. B., Grumet, R., & Hancock, J. (2015). Characterization of the genetic diversity of Uganda's sweet potato (*Ipomoea batatas*) germplasm using microsatellites markers. *Genetic Resources and Crop Evolution*, 62, 501-513
- Zhang, K., Wu, Z., Tang, D., Lv, C., Luo, K., Zhao, Y., Liu, X., Huang, Y., & Wang, J. (2016). Development and Identification of SSR Markers Associated with Starch Properties and β-Carotene Content in the Storage Root of Sweet Potato (*Ipomoea batatas* L.). *Frontiers in plant science*, 7, 223. sa
- Zhao, L., Qi, Z., Xiao, S., Tang, F., Liu, Y., Deng, Y., Dai, X., Zhou, Z., Ou, W., & Cao, Q. (2022). Genetic diversity assessment of Sweetpotato germplasm in China using InDel markers. *Agronomy (Basel, Switzerland)*, 12(12), 3074. <https://doi.org/10.3390/agronomy12123074>

Gracias