

# “Efectividad del método de búsqueda metaheurístico **enfriamiento simulado** en la determinación de una secuencia consenso a partir de múltiples secuencias”

Aspirante: Ing. Adrián Díaz  
Promotora: Dra. Gabriela Minetti

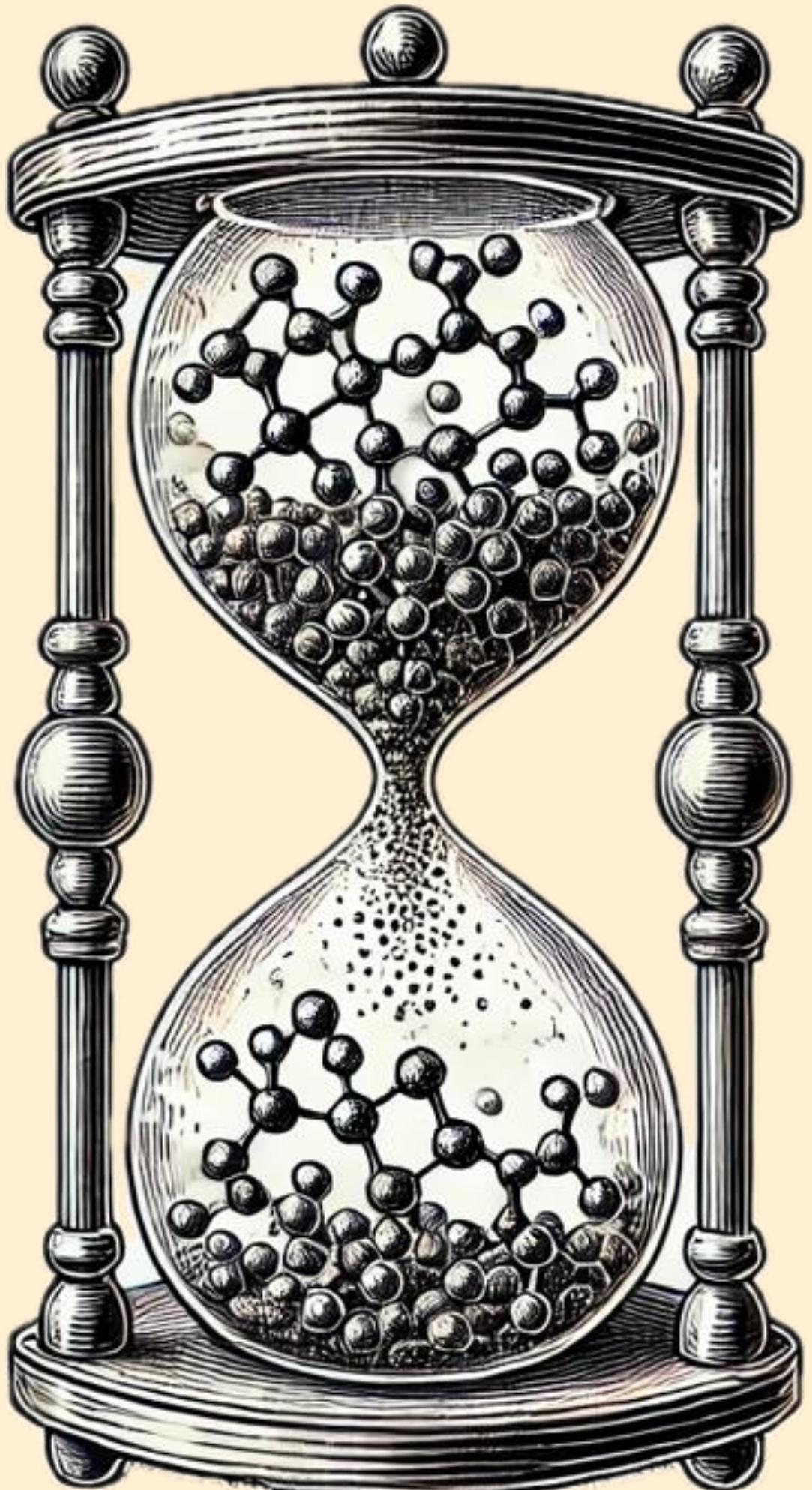
25 de octubre de 2024 - Online - 11h30 (ARG) / 16h30 (BEL)



Universidad  
Nacional  
de Quilmes



# AGENDA



1. MOTIVACIONES. HIPÓTESIS. OBJETIVOS
2. BREVE INTRODUCCIÓN BIOINFORMÁTICA
3. DISEÑO EXPERIMENTAL
4. RESULTADOS
5. PERSPECTIVAS



# MOTIVACIONES



La importancia de la Bioinformática en los hitos científicos.



Compartir una metodología con la comunidad científica.



Implementación de MSASA y una metodología de evaluación de diferentes *software MSA* utilizando una base de datos de prueba.



HIPÓTESIS

OBJETIVOS

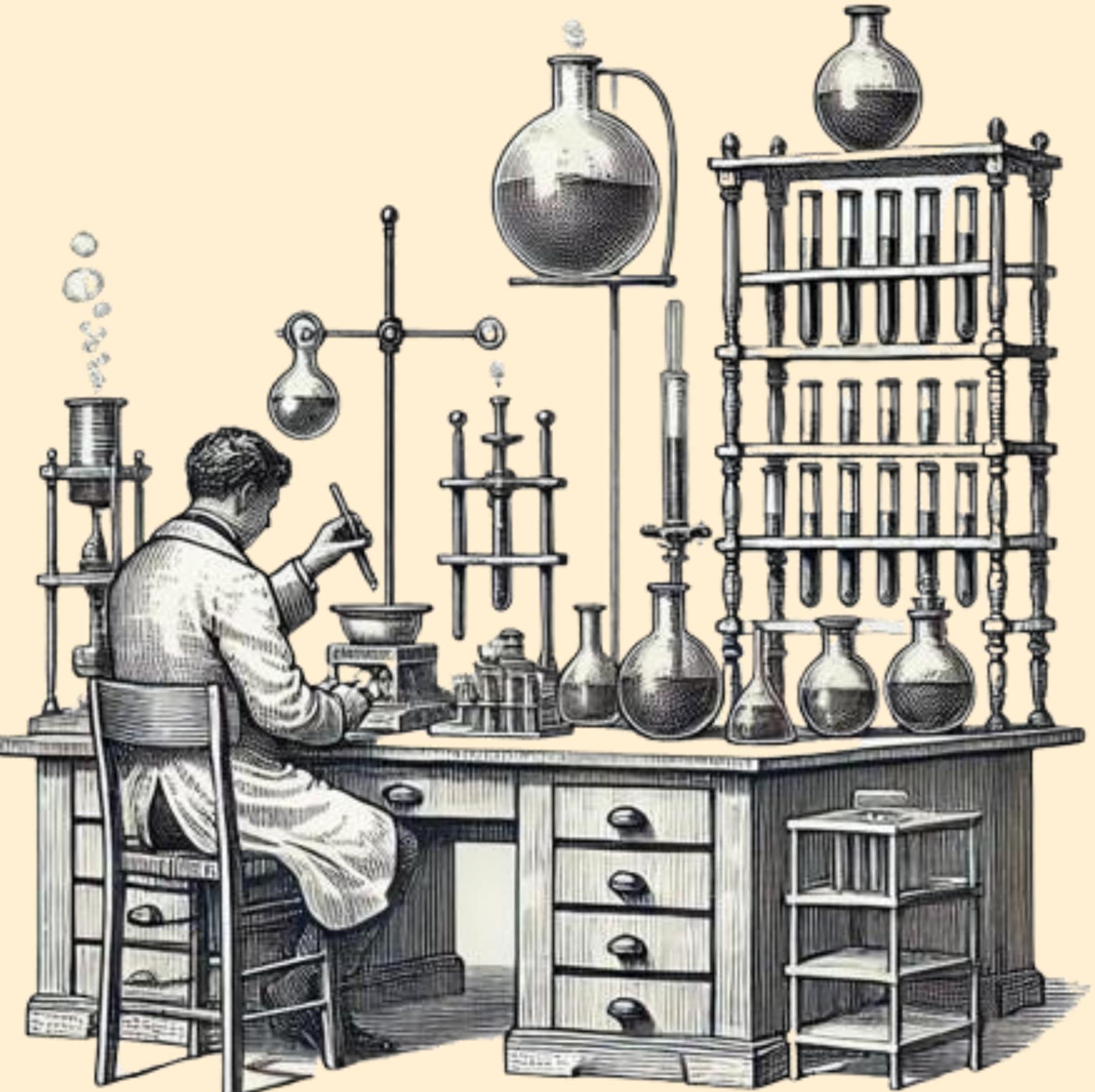
# HIPÓTESIS

H1. Implementaciones de **Enfriamiento Simulado** (SA) resuelven el problema MSA.

H2. Estos algoritmos son **competitivos** con otras soluciones MSA del estado del arte.

H3. Es posible **combinar** diferentes metaheurísticas para encontrar soluciones MSA suficientemente buenas.

H4. Se puede **crear un marco de trabajo** para comparar diferentes soluciones MSA para clasificarlos según la calidad de los resultados.



# OBJETIVOS



**O1.** Implementar un algoritmo SA para resolver el problema MSA y combinarlo con (meta)heurísticas para mejorar su desempeño (algoritmo **MSASA**).

**O2.** Realizar un estudio comparativo de implementaciones MSA del estado del arte eligiendo a las piezas de *software* más representativas.

**O2a.** Desarrollar un **marco de trabajo** suficientemente flexible para evaluar diferentes implementaciones MSA a través de métricas de calidad.

**O2b.** Evaluar MSASA para determinar si sus resultados son lo suficientemente buenos en comparación con el estado del arte.



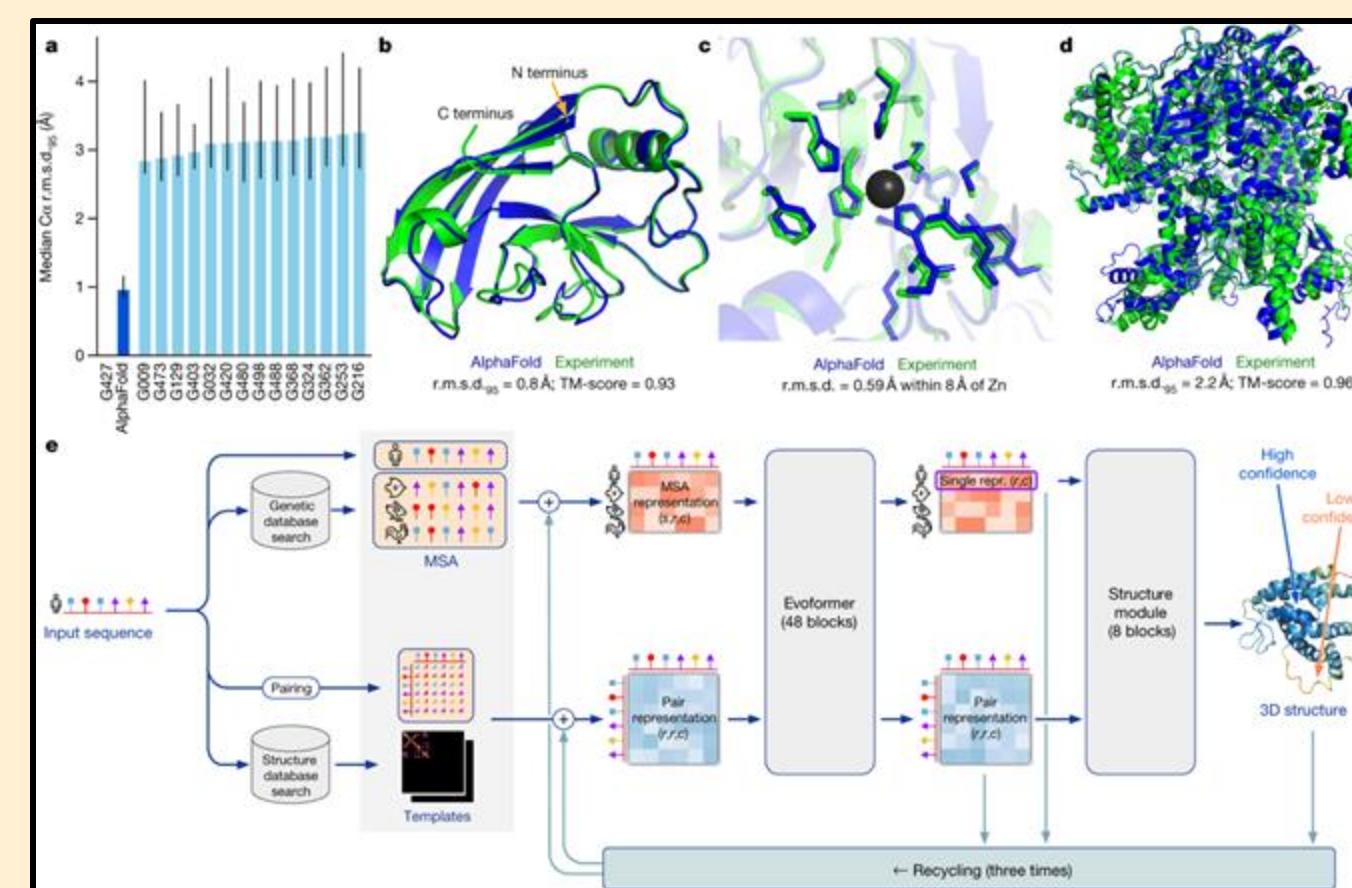
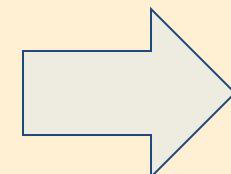
# BREVE INTRODUCCIÓN BIOINFORMÁTICA

# ALPHAFOLD 2

EJEMPLO BIOINFORMÁTICO GANADOR DE  
NOBEL QUÍMICA 2024



Secuencia de la  
proteína



Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 583–589 (2021). <https://doi.org/10.1038/s41586-021-03819-2>

David Baker

"for computational protein design"



David Baker. Ill. Niklas Elmehed © Nobel Prize Outreach

Demis Hassabis

"for protein structure prediction"



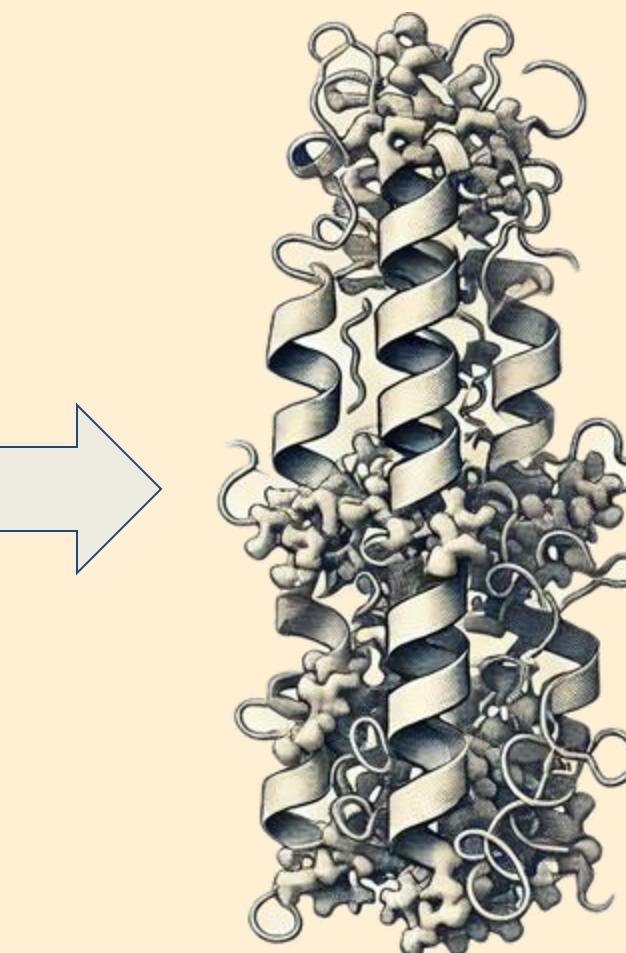
Demis Hassabis. Ill. Niklas Elmehed © Nobel Prize Outreach

John Jumper

"for protein structure prediction"

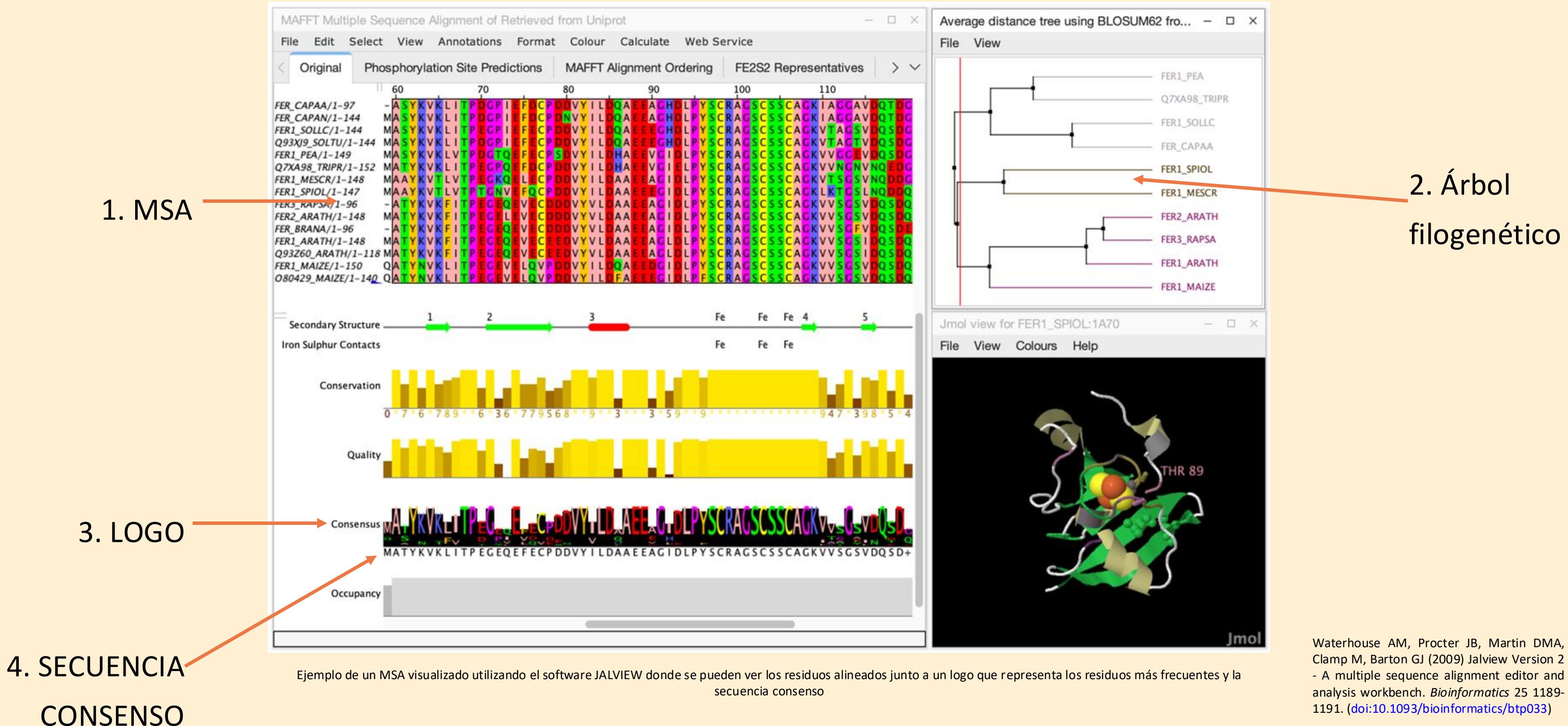


John Jumper. Ill. Niklas Elmehed © Nobel Prize Outreach

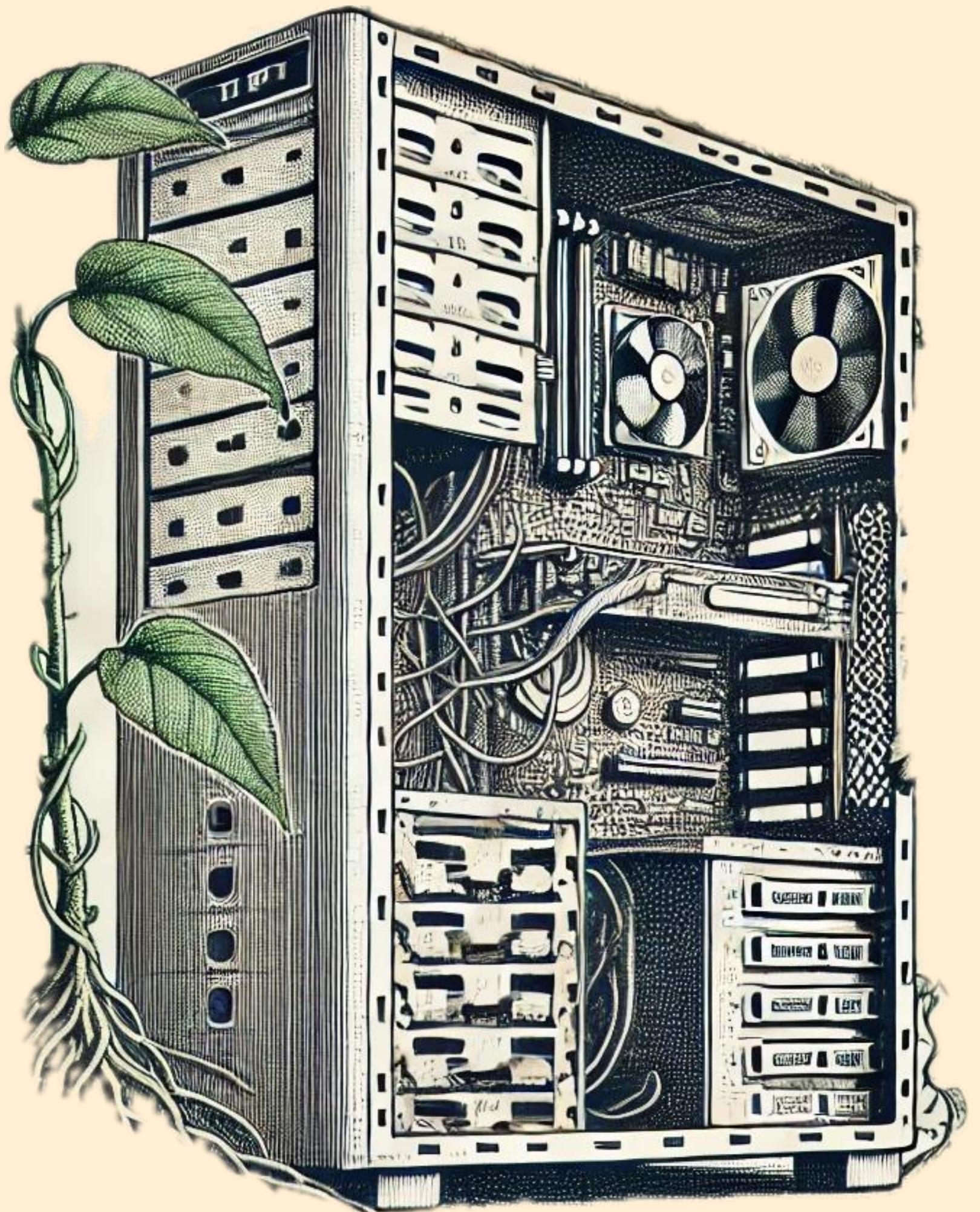


Proteína plegada\*\*\*

# SECUENCIAS CONSENSO & MSA



# COMPUTABILIDAD



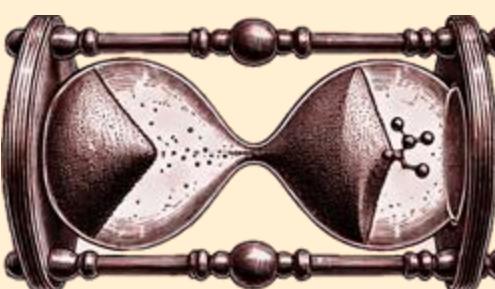
## 1. COMPLEJIDAD

- Teoría de la Complejidad Computacional: tiempo y recursos.
- La máquina de Turing.



## 2. PROBLEMAS P

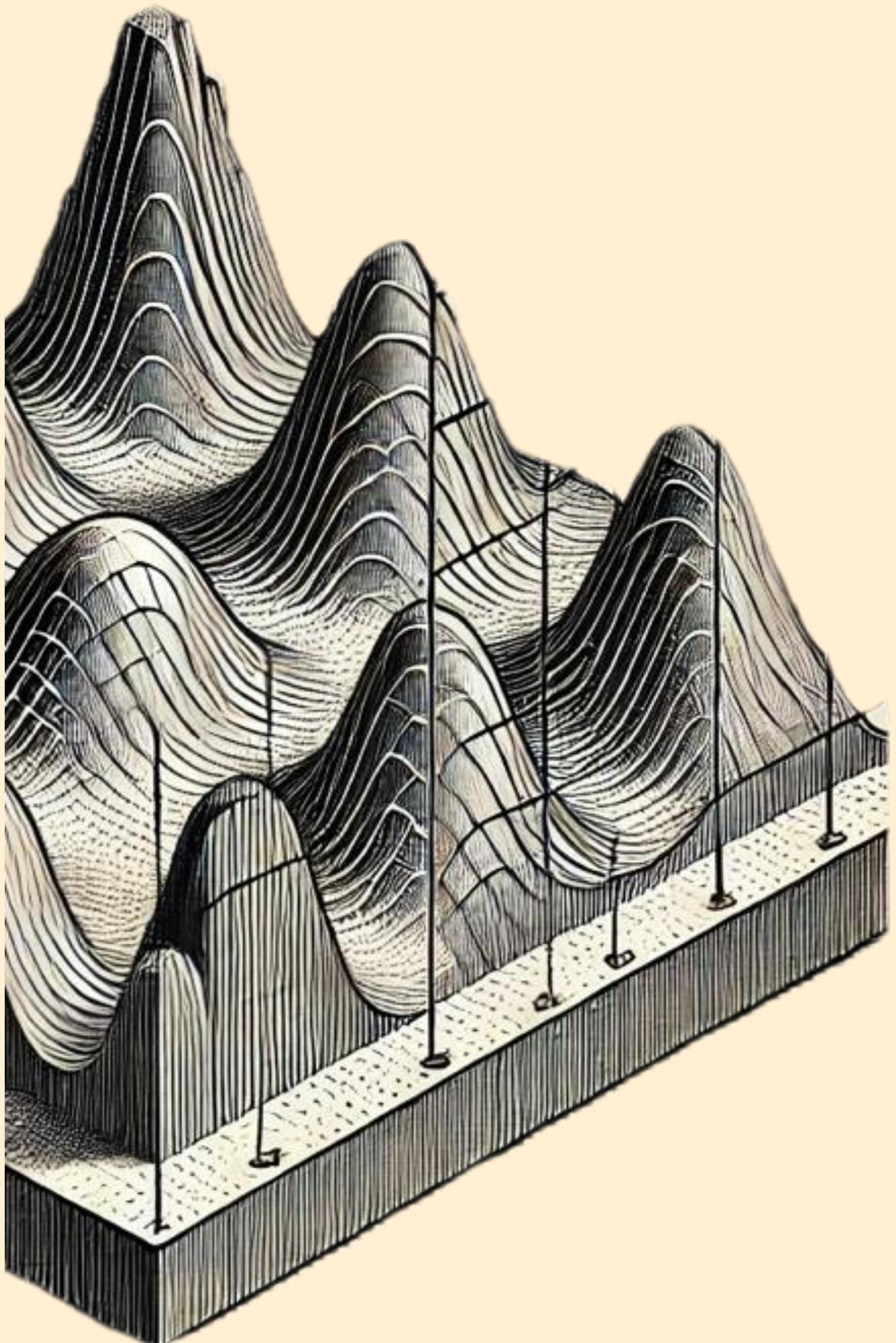
- Pueden ser resueltos en un tiempo polinomial por una máquina de Turing determinista.



## 3. PROBLEMAS NP

- No pueden ser resueltos en un tiempo polinomial por una máquina de Turing determinista.
- Las soluciones si pueden ser verificadas en un tiempo polinomial.

# (META)HEURÍSTICAS



## 1. HEURÍSTICAS

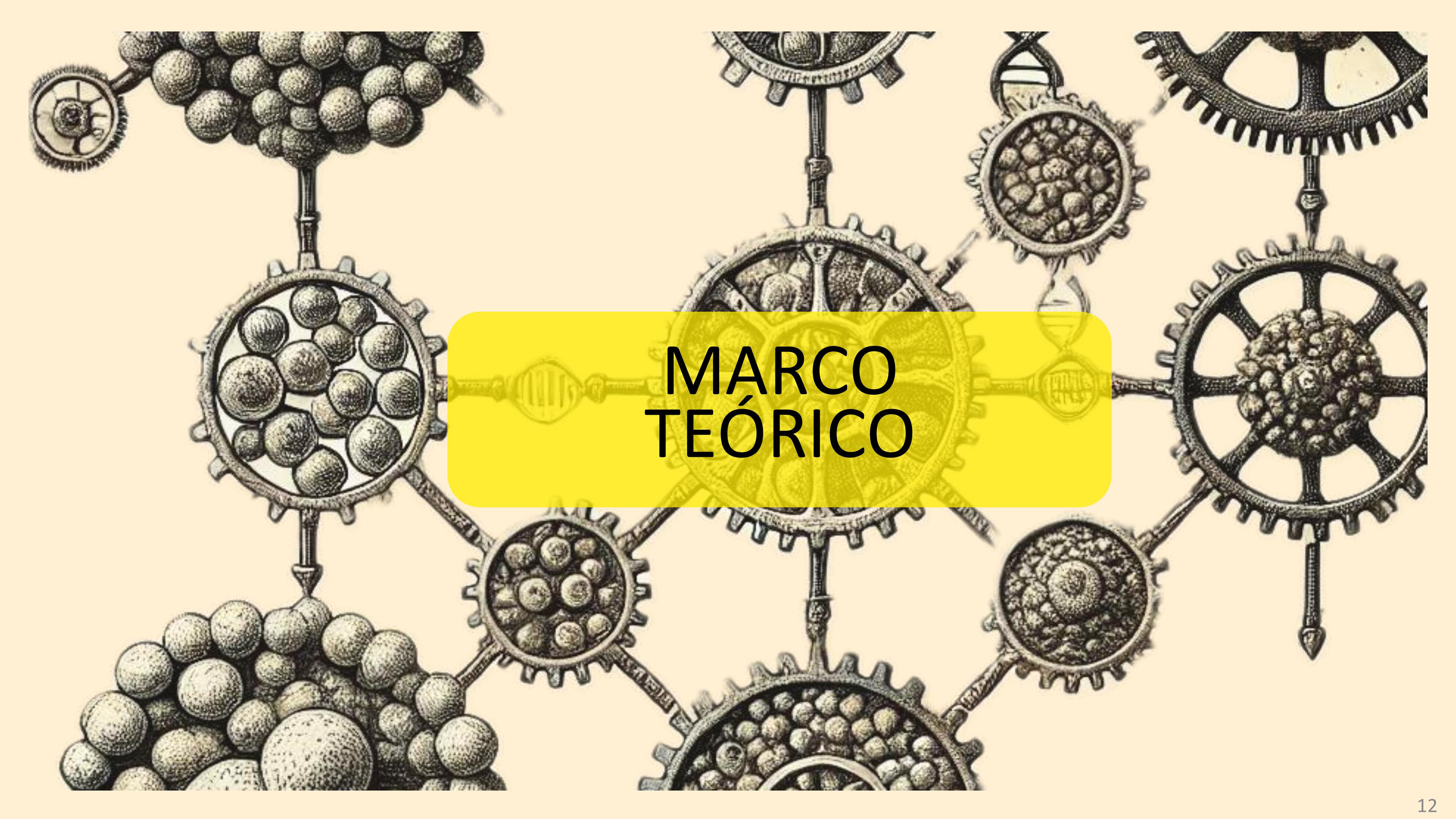
- Lo contrario a lo exacto.
- Obtención de resultado suficientemente buenos.
- Tiempos de ejecución.

## 2. METAHEURÍSTICAS

- Mejorar los resultados de las heurísticas.
- Marco general para algoritmos híbridos.

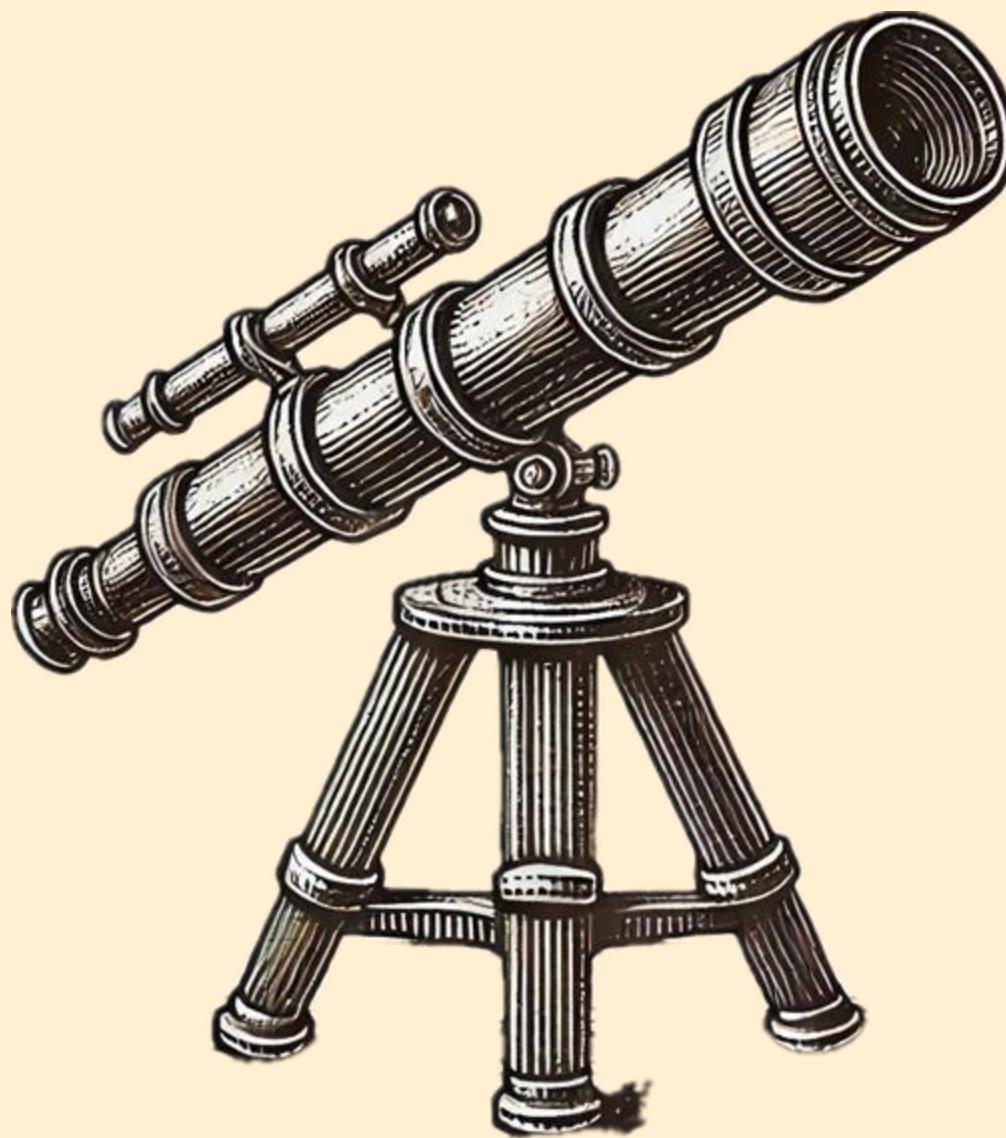
## 3. INSPIRACIÓN

- Son creadas a partir de modelos de otras ciencias como la Física, la Genética, la Biología.



# MARCO TEÓRICO

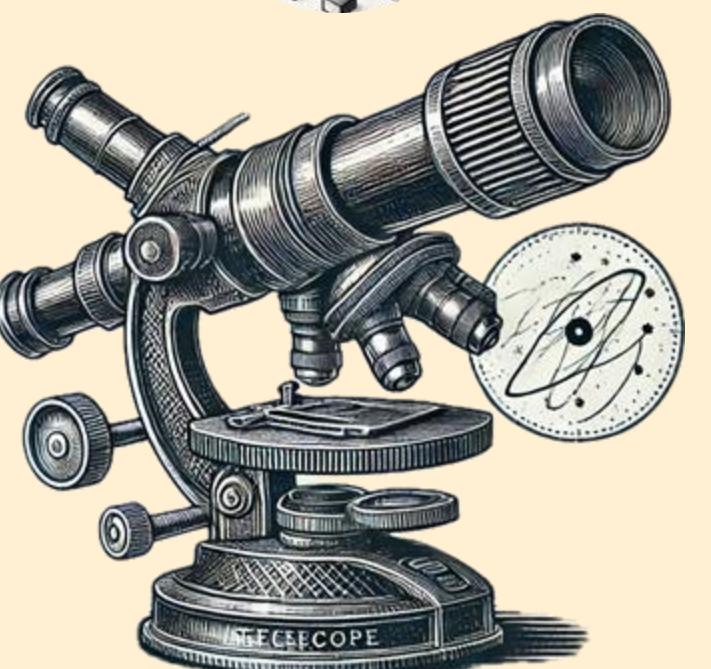
# ALGORITMOS MSA



GLOBALES



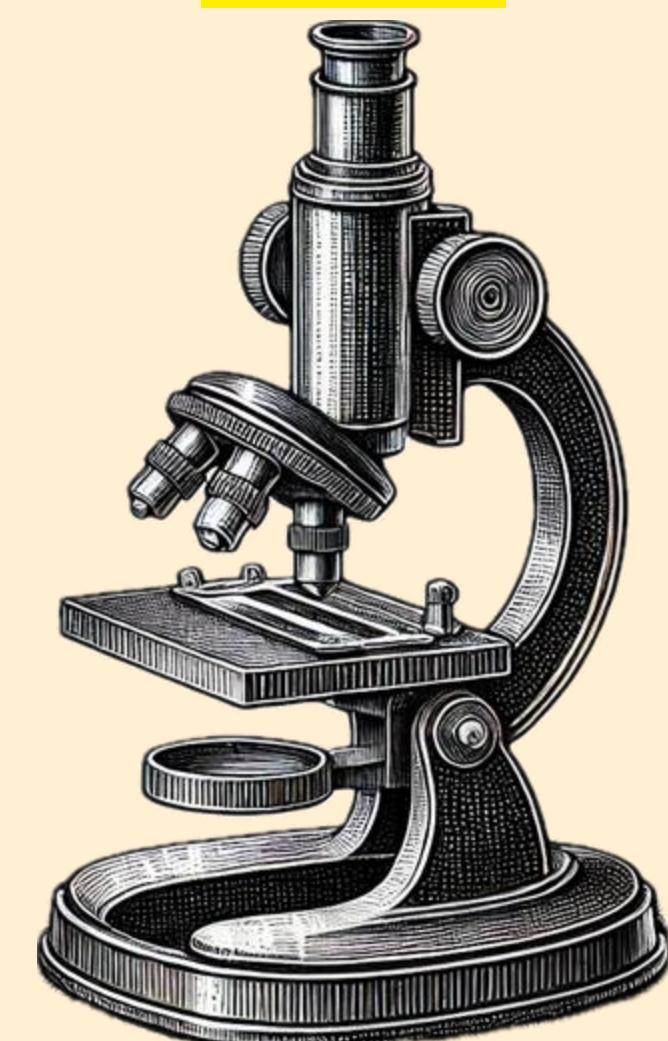
NEEDLEMAN-WUNSCH



HÍBRIDOS



FUSIÓN  
PROGRESIVA

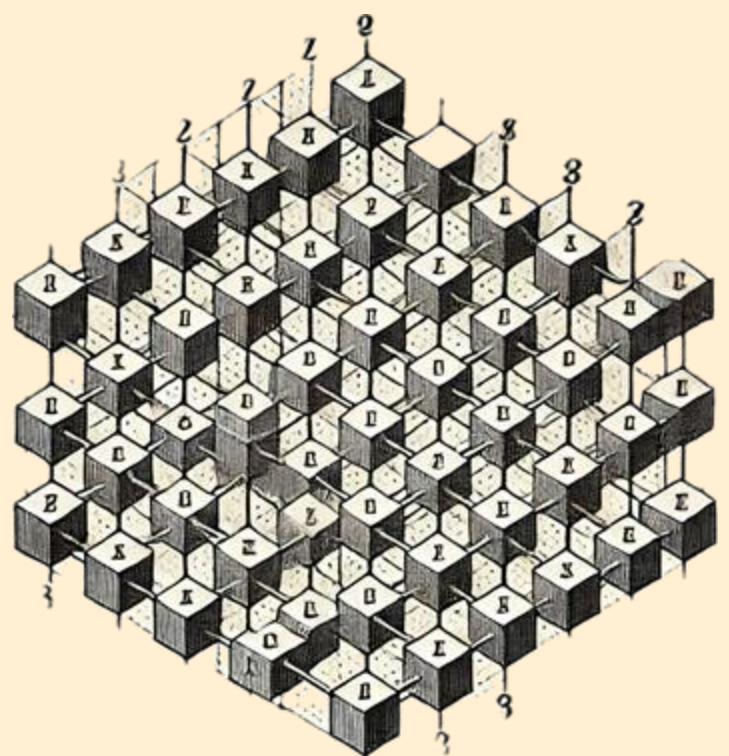


LOCALES

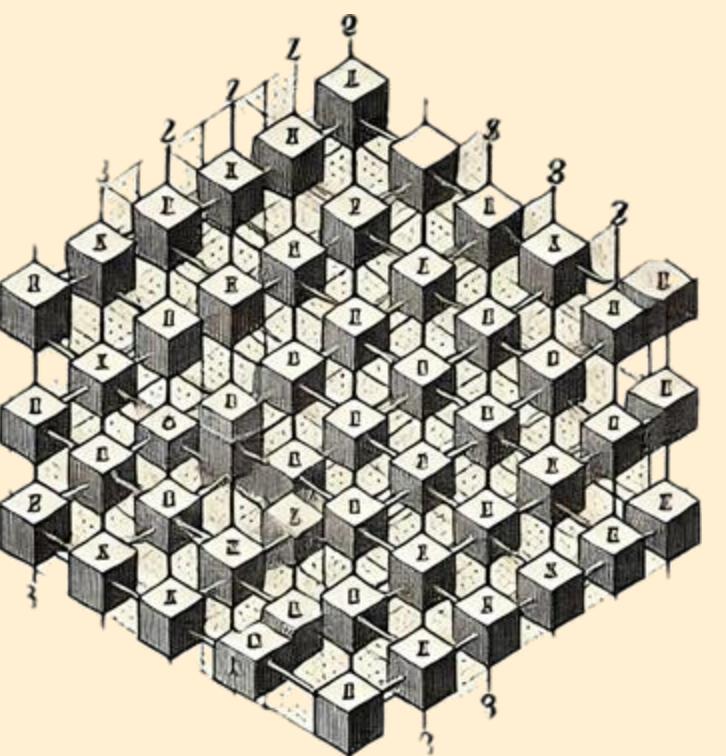


SMITH-WATERMAN

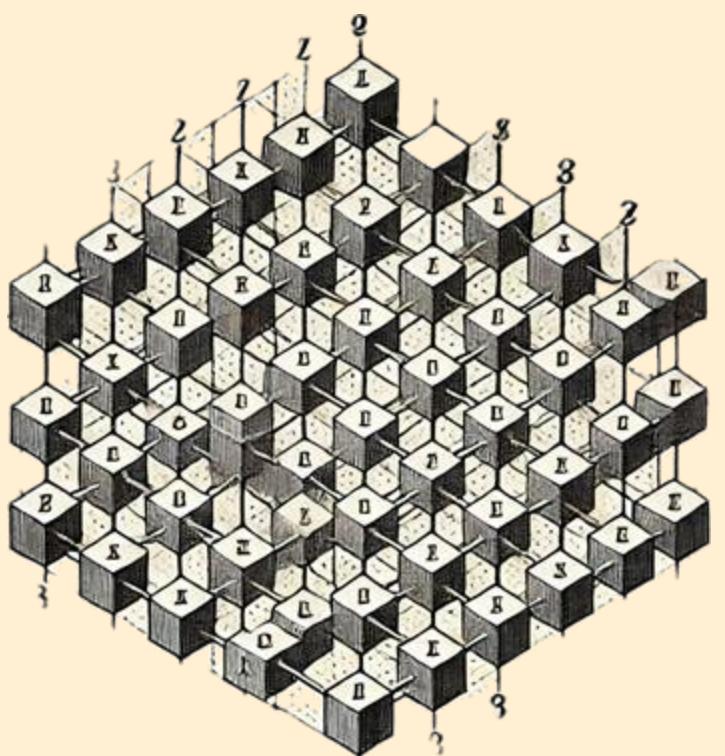
# INFORMACIÓN BIOLÓGICA



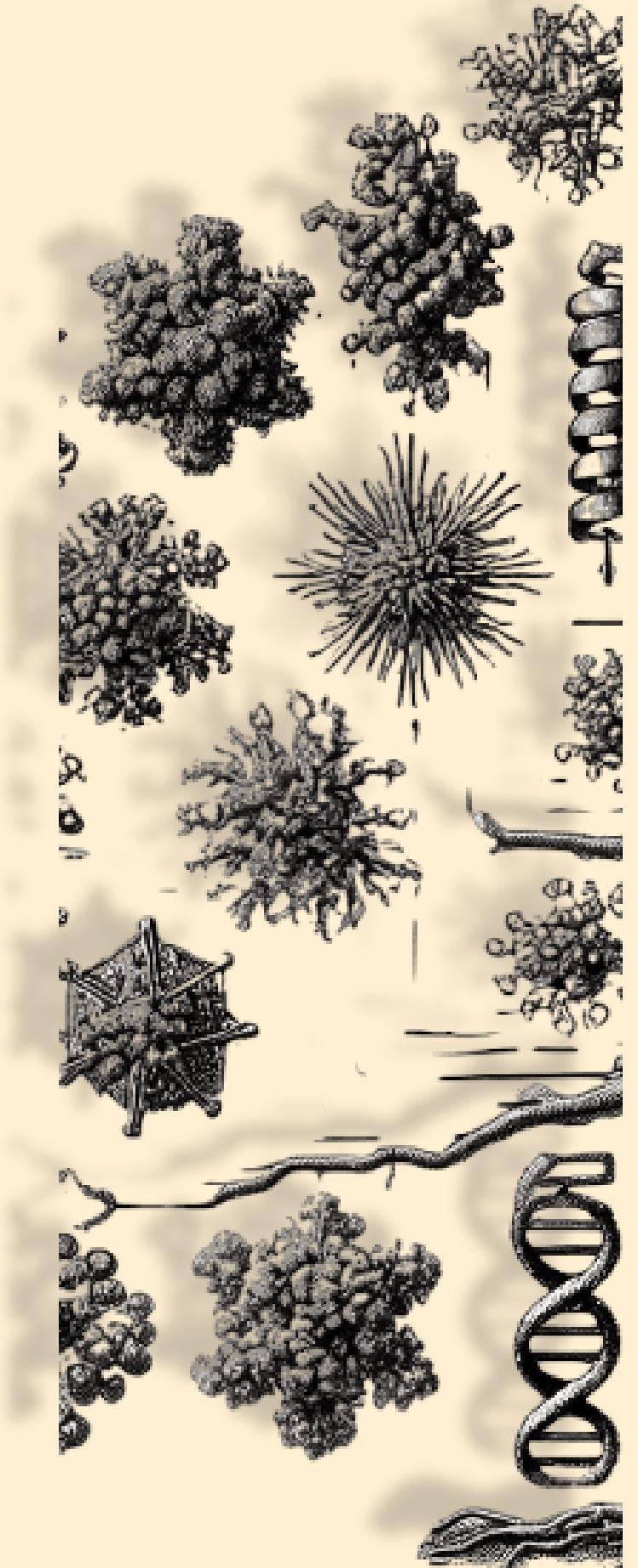
# 1. PAM



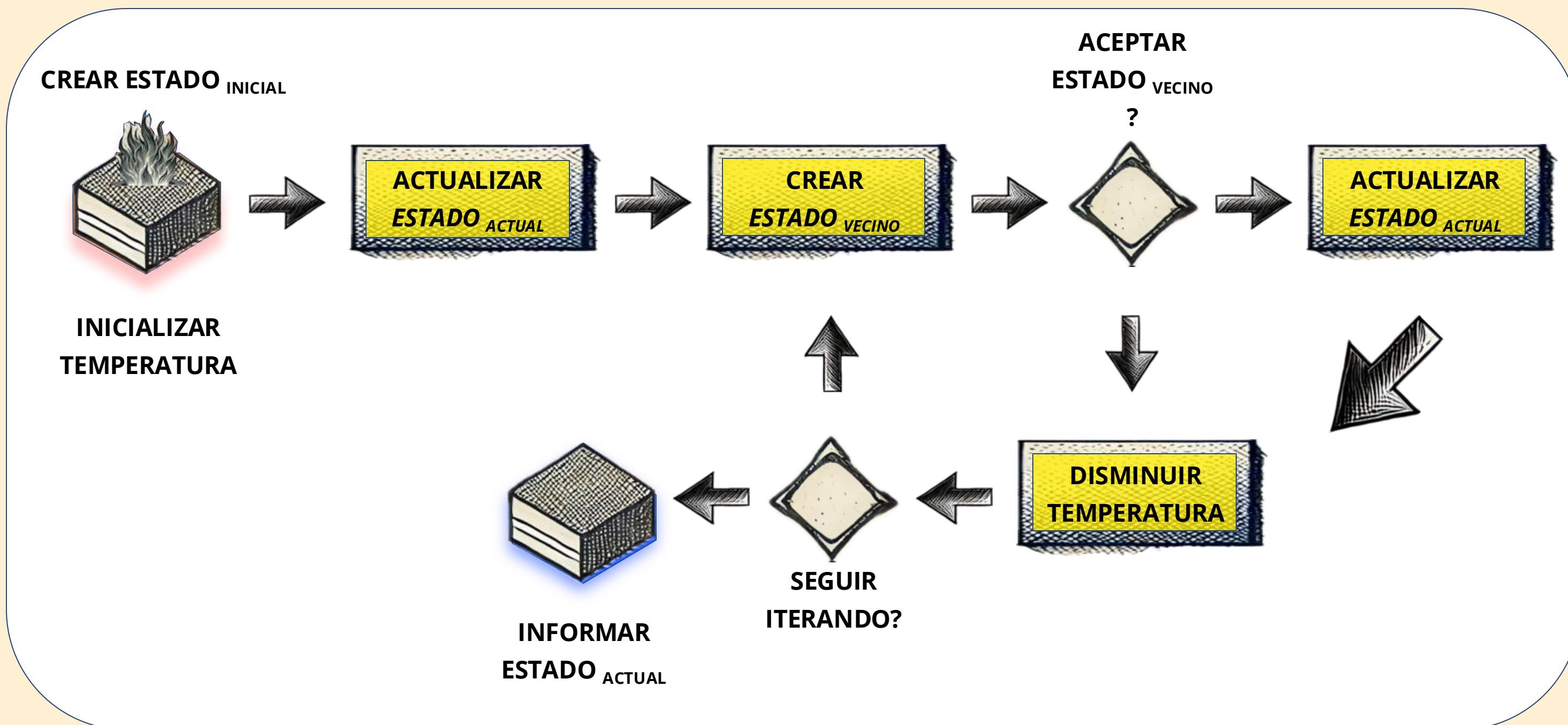
## 2. BLOSUM



# **3. GONNET**



# ENFRIAMIENTO SIMULADO



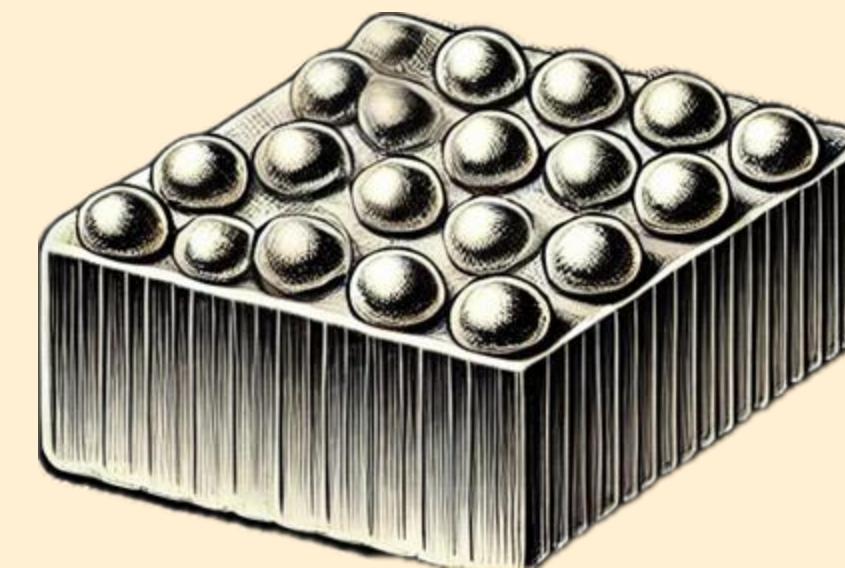
CRITERIO DE  
BOLTZMANN

# ANALOGÍAS EN EL PROBLEMA MSA

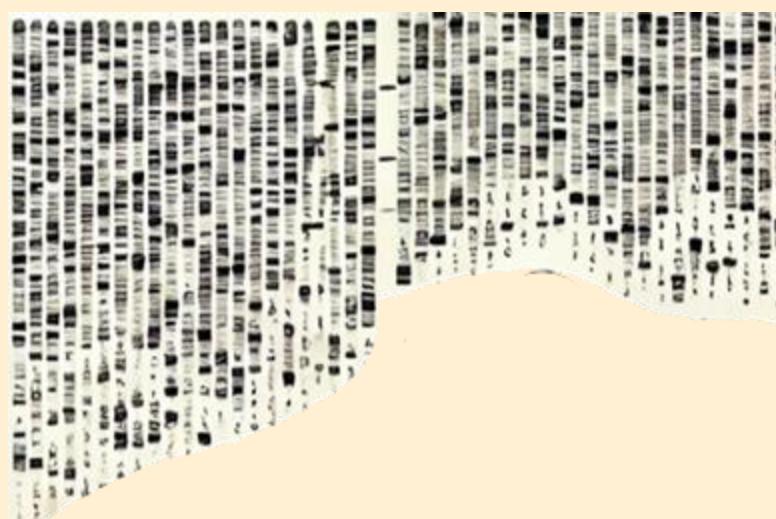
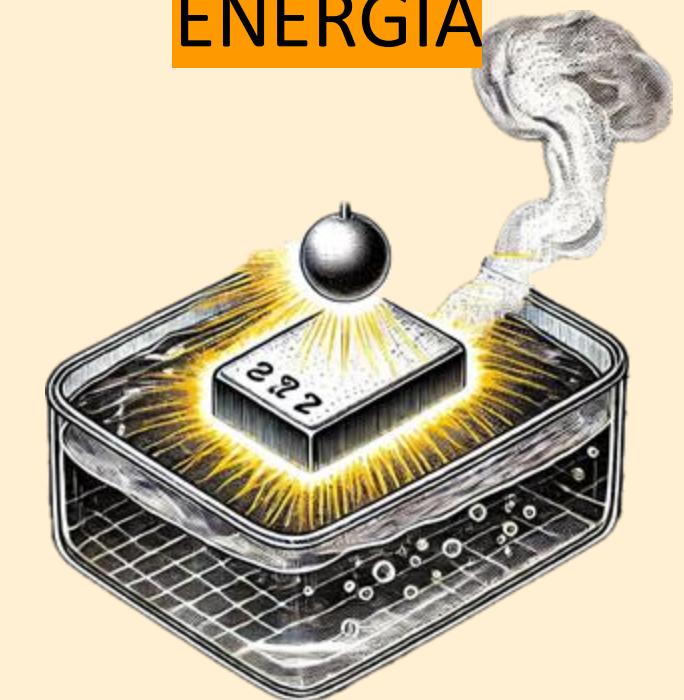
PIEZA MÉTALICA



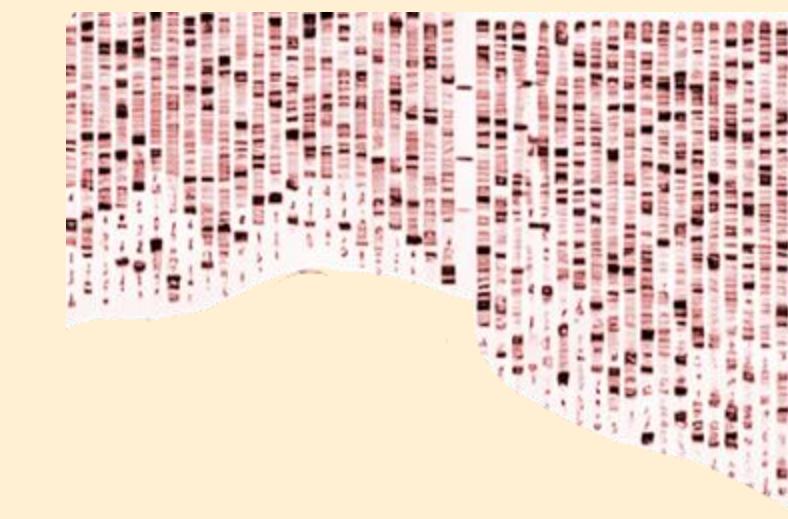
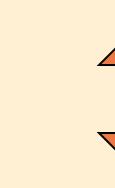
ESTADO VECINO



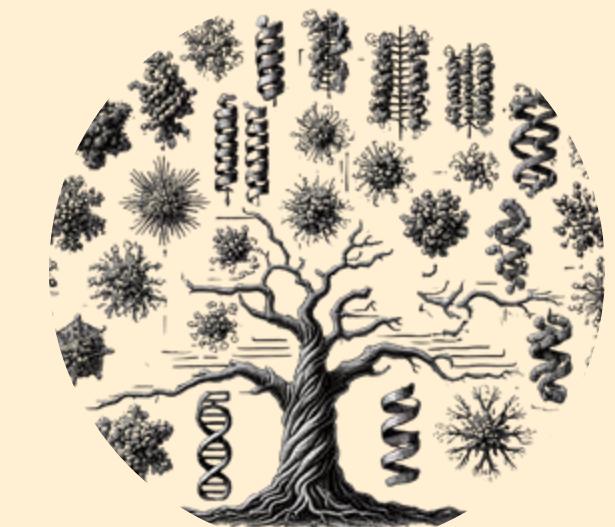
ENERGÍA



MSA

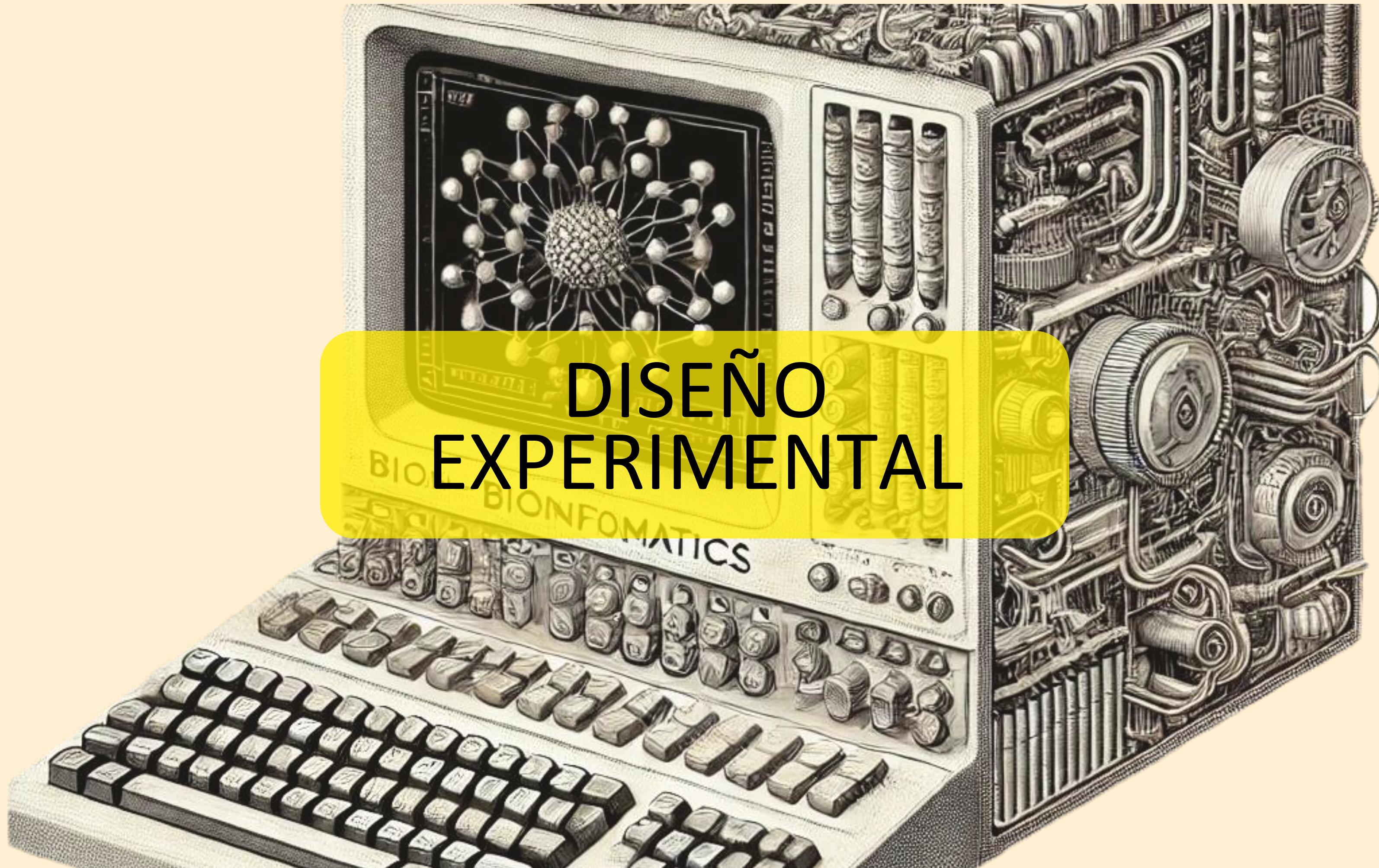


MSA MODIFICADO

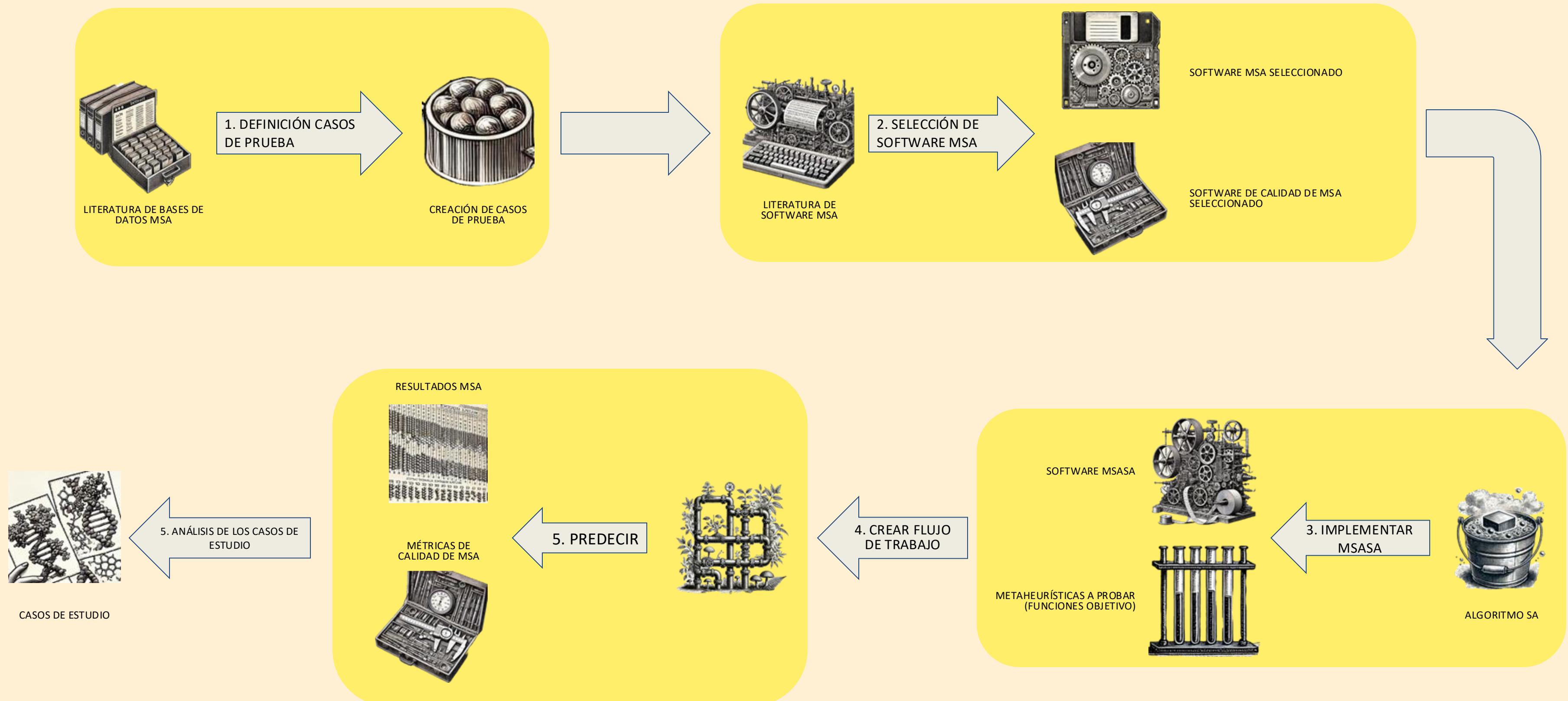


CALIDAD DEL MSA

# DISEÑO EXPERIMENTAL



# DISEÑO EXPERIMENTAL



# LA BASE DE DATOS DE PRUEBAS



## BALI-SCORE

- Similitud.
- Regiones altamente conservadas.
- Regiones de divergencia.



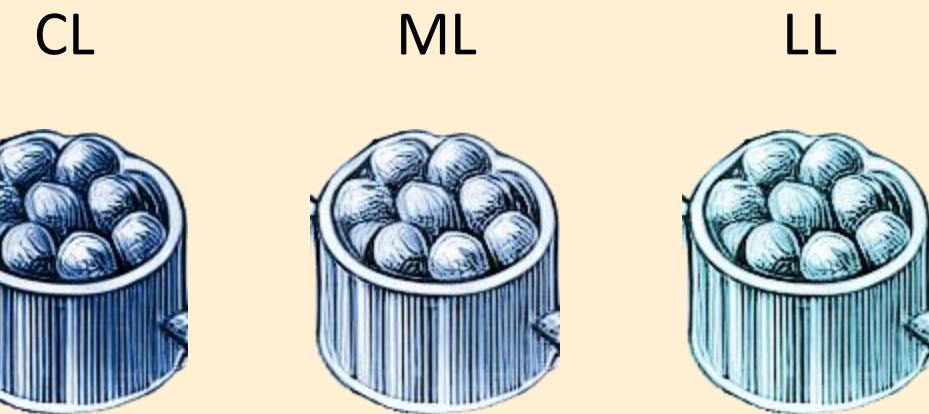
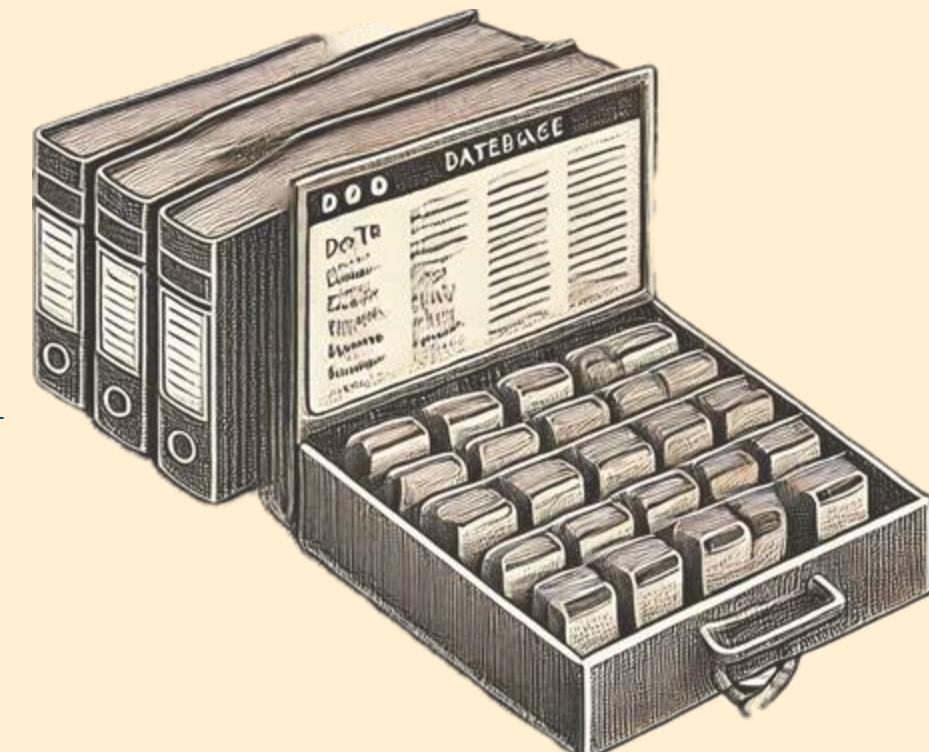
## DIMENSIONES

- Longitud de las secuencias.
- Cantidad de secuencias.
- Grupos de Bali Score.



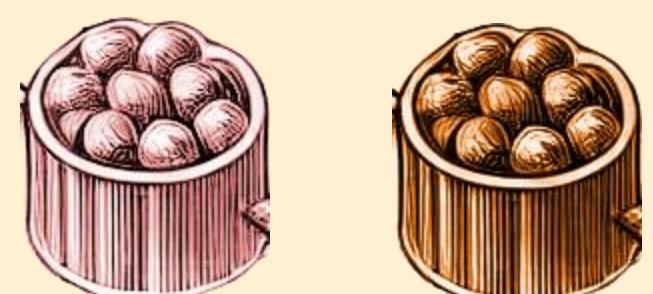
## GRUPOS

- 10 grupos organizados por similitud y complejidad.
- Selección de los 23 casos de prueba.



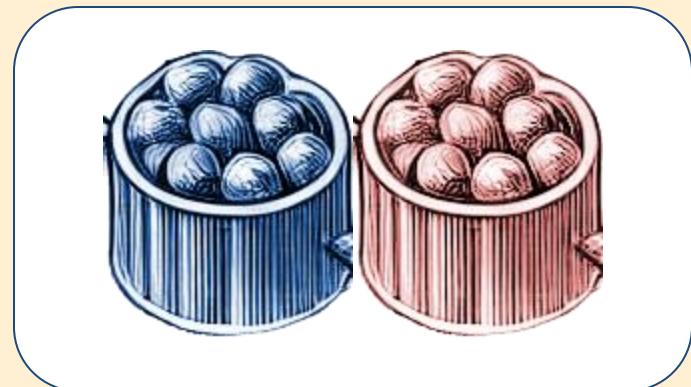
**LONGITUD  
DE LAS SECUENCIAS**

**CANTIDAD DE SECUENCIAS**



BC      AC

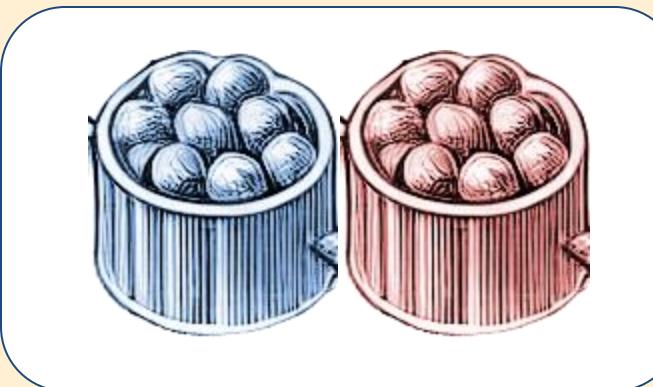
## CL-BC



CL: Hasta 100 residuos por secuencia  
BC: Hasta 100 secuencias a alinear

C1 (Ref 1) C4 (Ref 2)  
C7 (Ref 3) C18 (Ref 10)

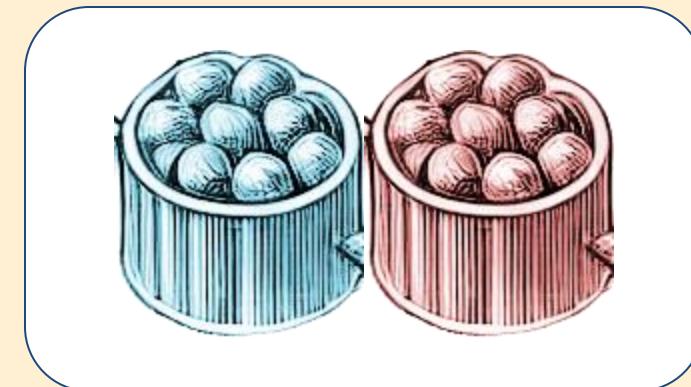
## ML-BC



ML: Entre 100 y 400 residuos por secuencia  
BC: Hasta 100 secuencias a alinear

C2 (Ref 1) C5 (Ref 2)  
C8 (Ref 3) C12 (Ref 4)  
C15 (Ref 9) C20 (Ref 10)

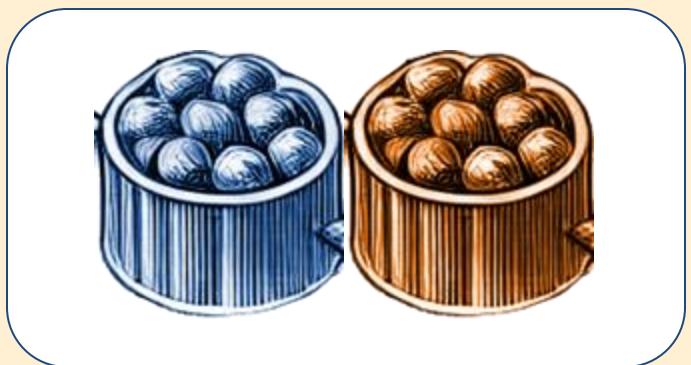
## LL-BC



LL: Más de 400 residuos por secuencia  
BC: Hasta 100 secuencias a alinear

C3 (Ref 1) C6 (Ref 2)  
C10 (Ref 3) C13 (Ref 4)  
C14 (Ref 5) C16 (Ref 9)  
C22 (Ref 10)

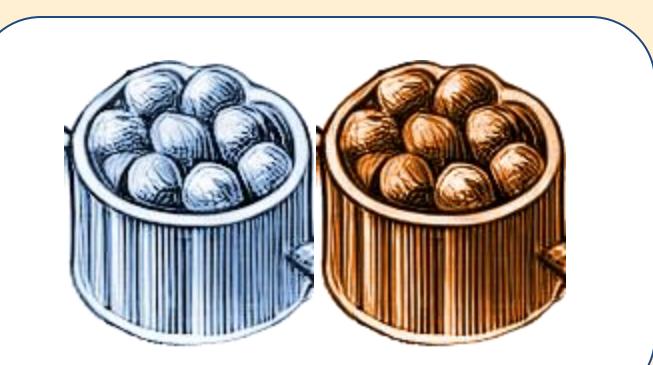
## CL-AC



CL: Hasta 100 residuos por secuencia  
AC: Más de 100 secuencias a alinear

C19 (Ref 10)

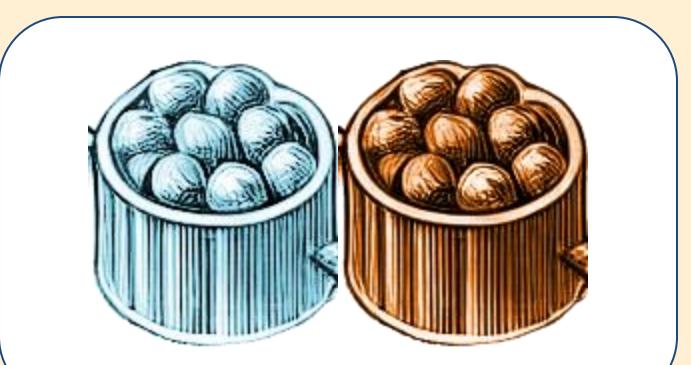
## ML-AC



ML: Entre 100 y 400 residuos por secuencia  
AC: Más de 100 secuencias a alinear

C9 (Ref 3)  
C21 (Ref 10)

## LL-AC



LL: Más de 400 residuos por secuencia  
AC: Más de 100 secuencias a alinear

C11 (Ref 3)  
C17 (Ref 9)  
C23 (Ref 10)

# SOFTWARE DEL ESTADO DEL ARTE

+  
G  
L  
O  
B  
A  
L



## CLUSTAL OMEGA

Basado en N-W.  
Alineamiento progresivo.



## MUSCLE

Similar Clustal Omega.  
Con un alineamiento rápido al principio y  
luego otro más lento y preciso.



## KALIGN

Alineamiento progresivo.



+  
H  
Í  
B  
R  
I  
D  
O



## T COFFEE

Algoritmo Híbrido.  
incorpora información de alineamientos  
locales para mejorar la precisión.  
Puede incluso recibir información  
estructural.

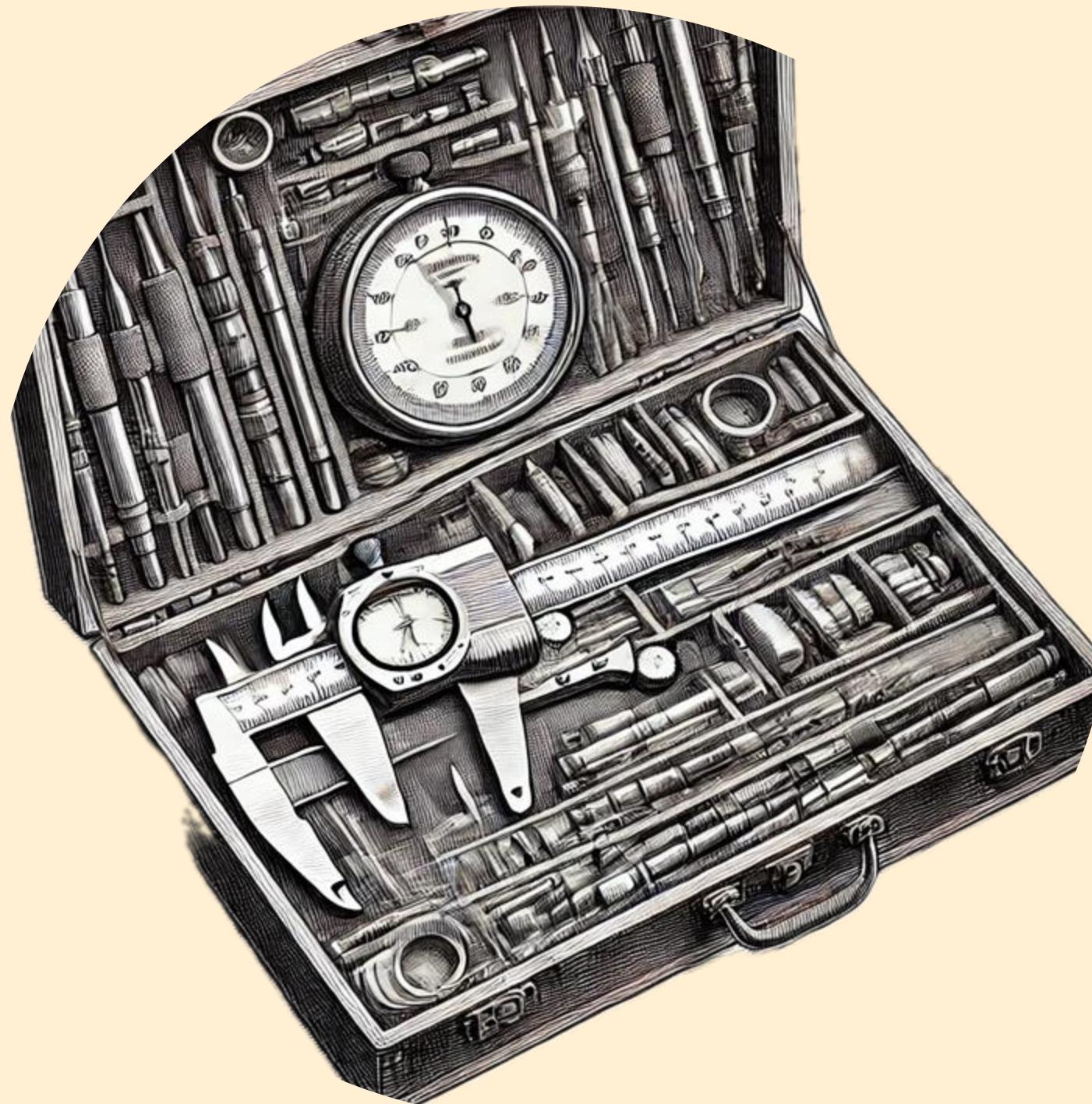


## MAFFT

Global o Híbrido.  
Progresivo pero también puede usar  
información de algoritmos locales.



# MUMSA - OVERLAP SCORE



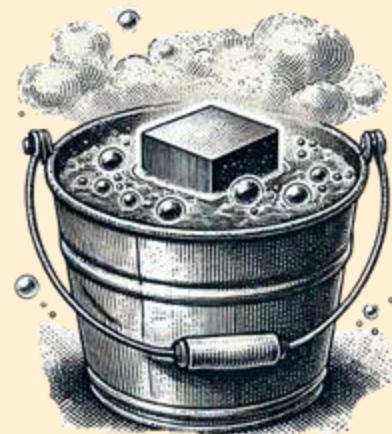
MUMSA evalúa la calidad de alineamientos múltiples automáticamente

El puntaje AOS es muy importante para determinar cuán confiable puede ser un alineamiento.

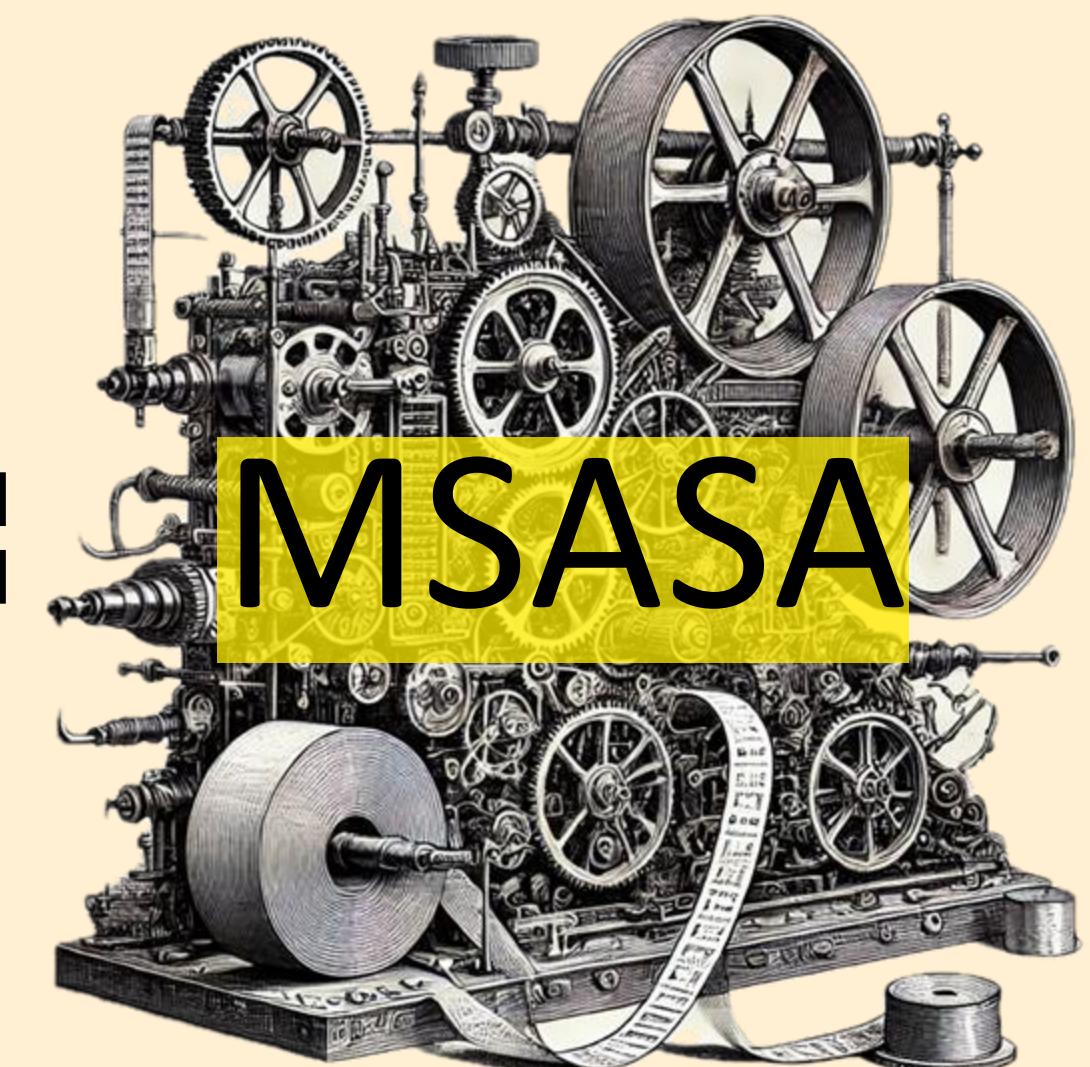
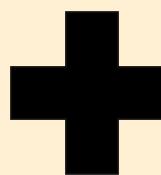
El puntaje AOS mide la dificultad de alinear secuencias

Un AOS mayor a 0.8 indica que las secuencias son fáciles de alinear y el resultado es confiable. Si el AOS es menor a 0.5, las secuencias son difíciles de alinear y los resultados deben tratarse con precaución.

# IMPLEMENTACIÓN MSASA

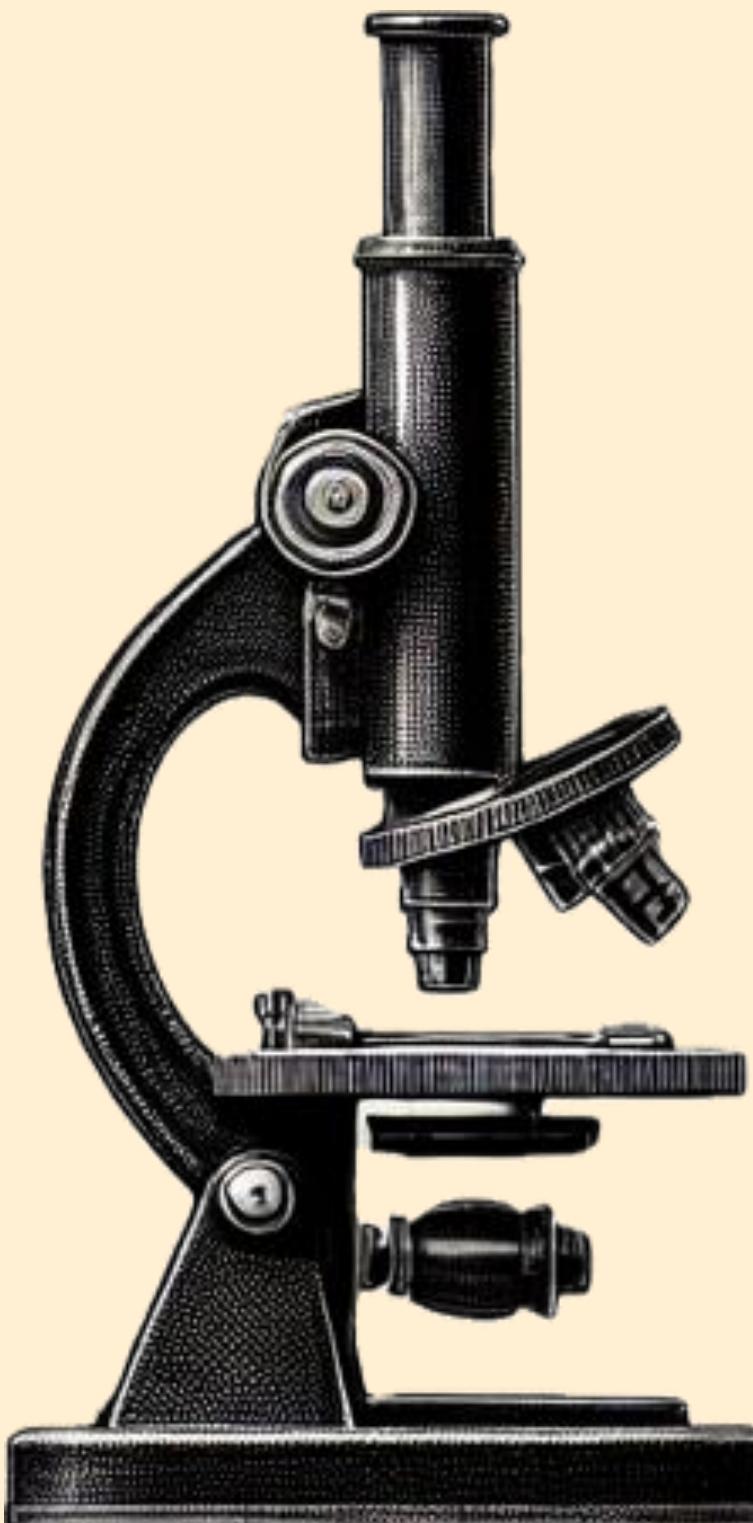


**SA**



**MSASA**

# FUNCIONES OBJETIVO



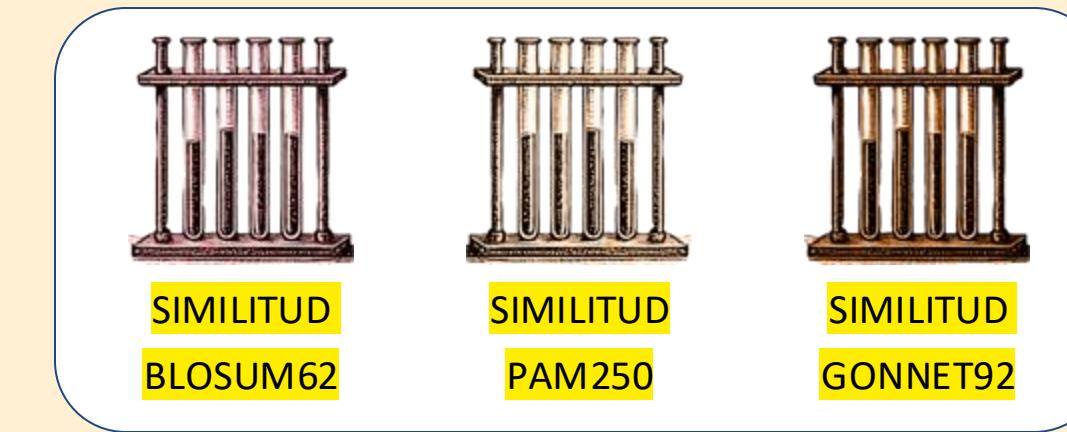
## Metaheurísticas más sencillas

- Cuentan grupos de residuos por columna.



## Metaheurísticas basadas información biológica

- Utilizan los puntajes propuestos por diferentes matrices de identidad/substitución.



## Metaheurísticas basadas los algoritmos MSA clásicos

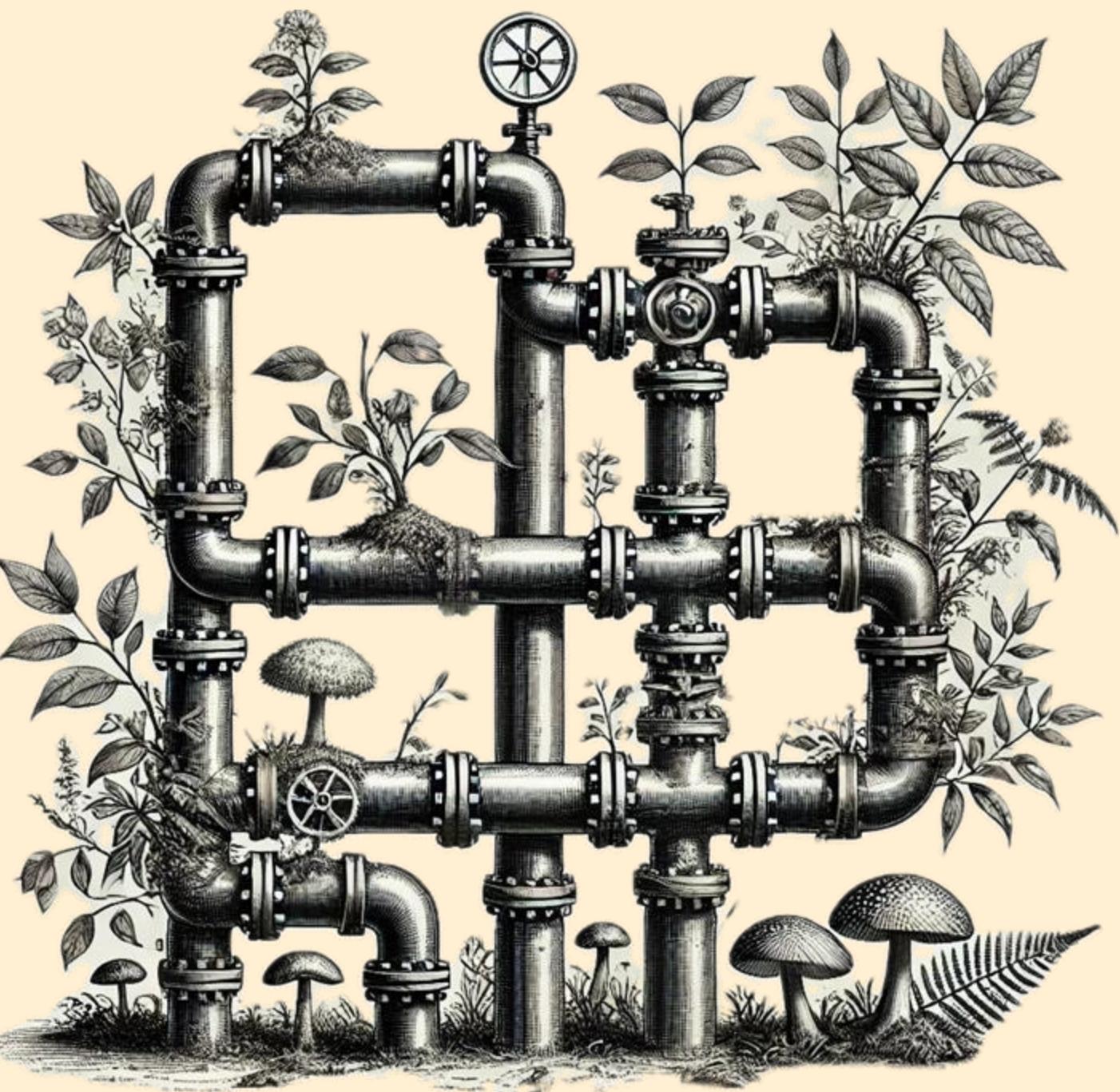
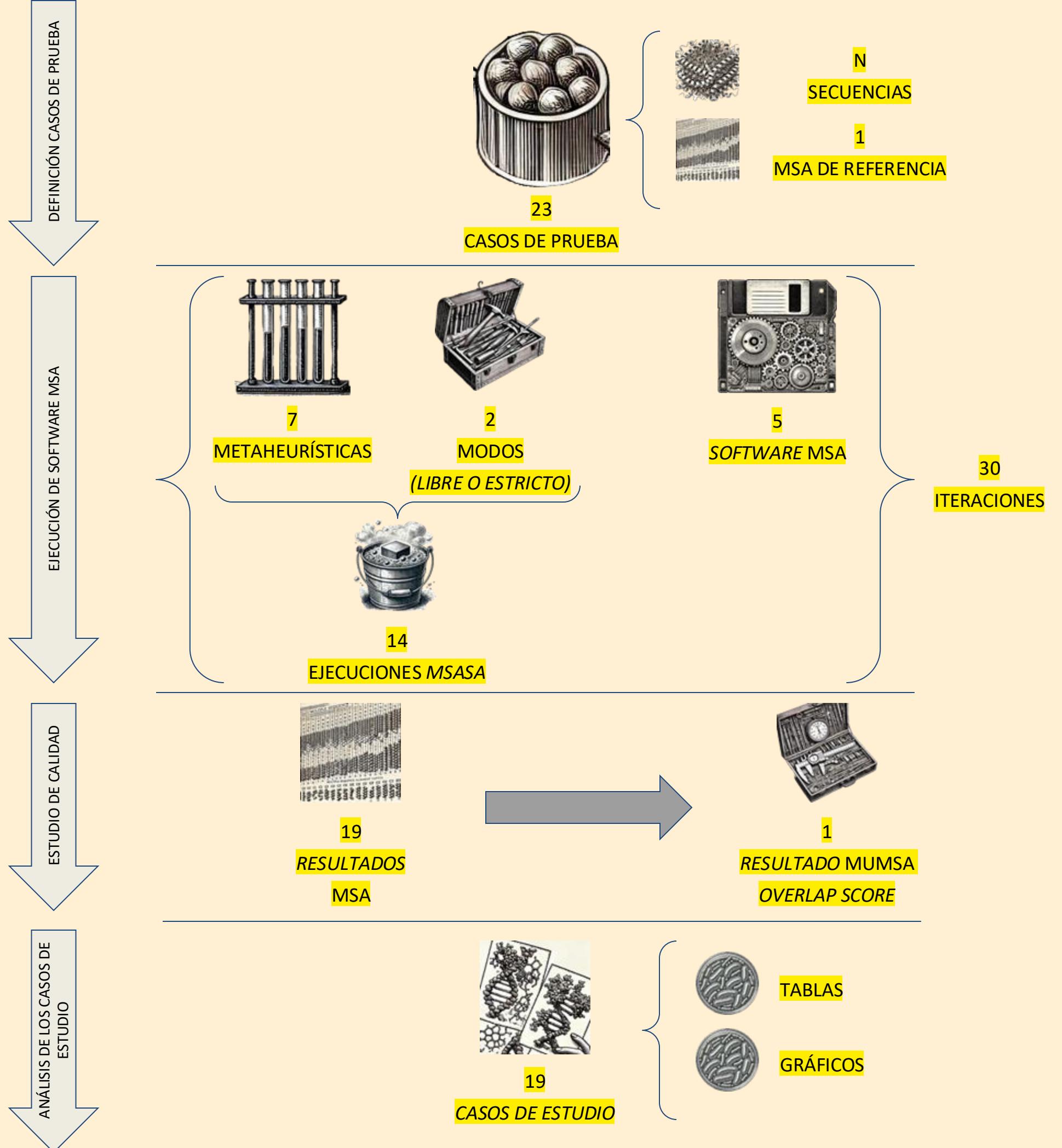
- Intentan simular N-W y S-W mediante una aproximación hecha con las penalidades y puntos por coincidencia.





# RESULTADOS

# FLUJO DE TRABAJO



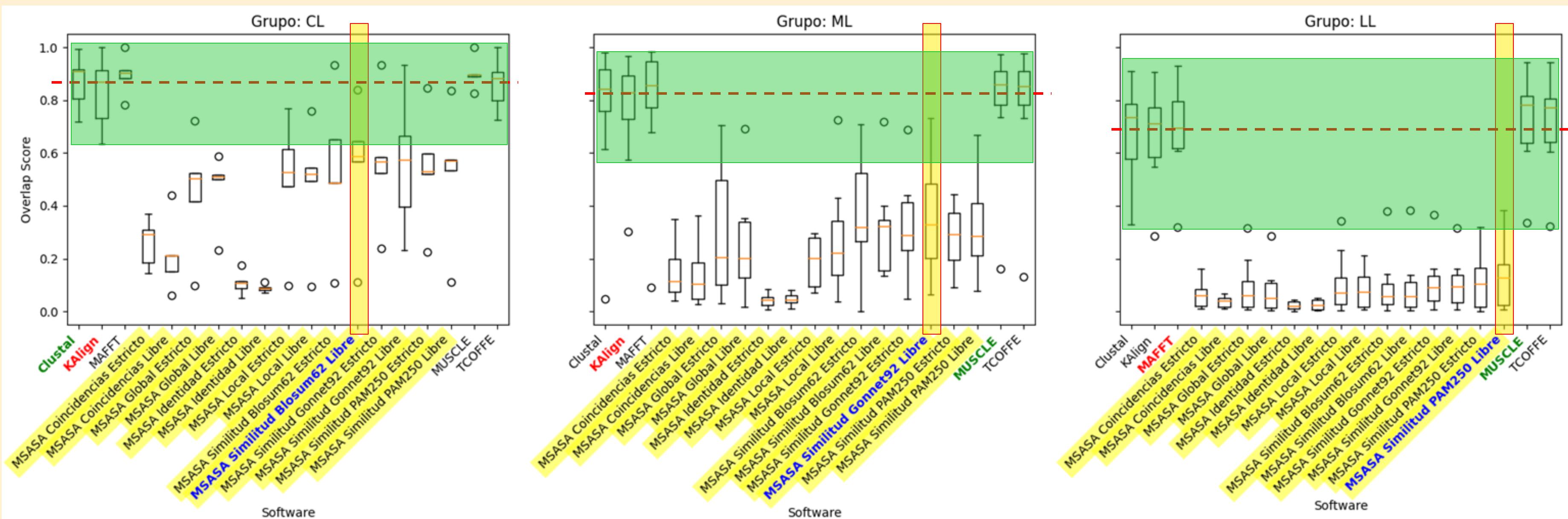
# OVERLAP SCORE (OS)

## LONG. DE SECUENCIAS

CL

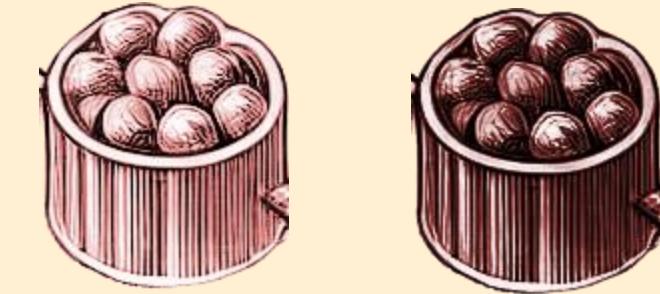
ML

LL



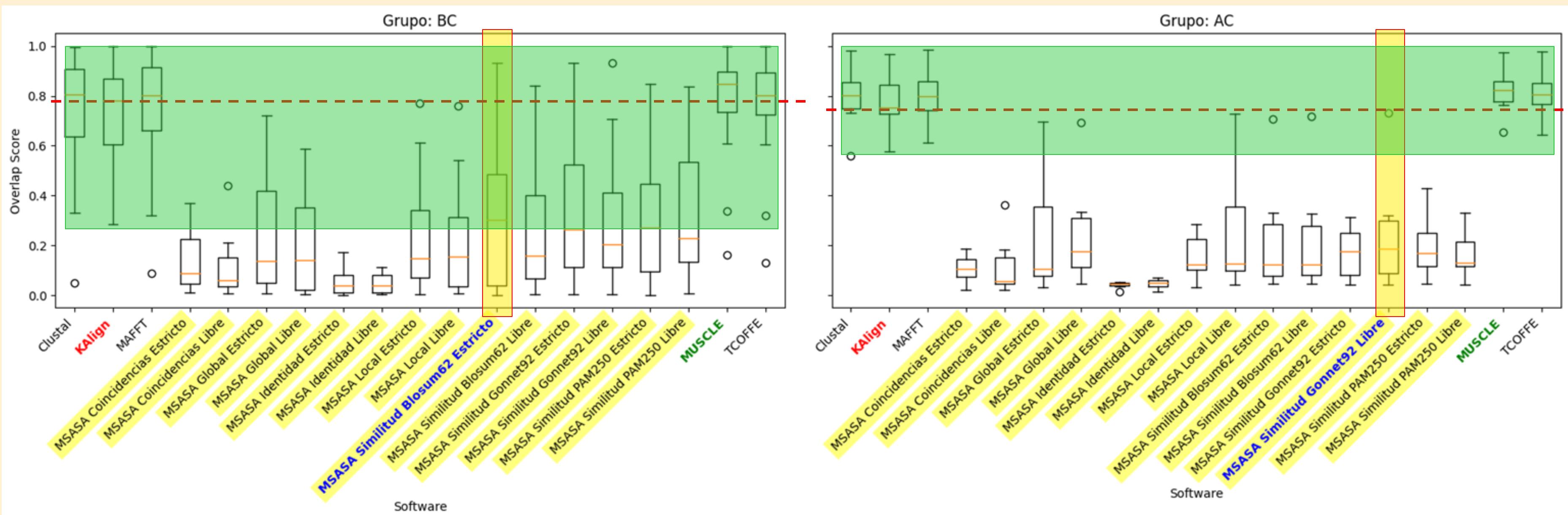
# OVERLAP SCORE (OS)

## CANT. DE SECUENCIAS



BC

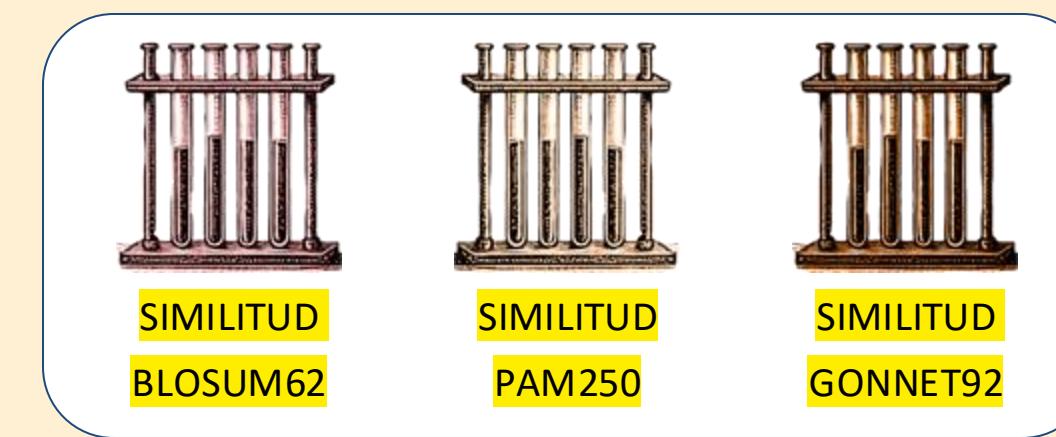
AC



# OVERLAP SCORE (OS)



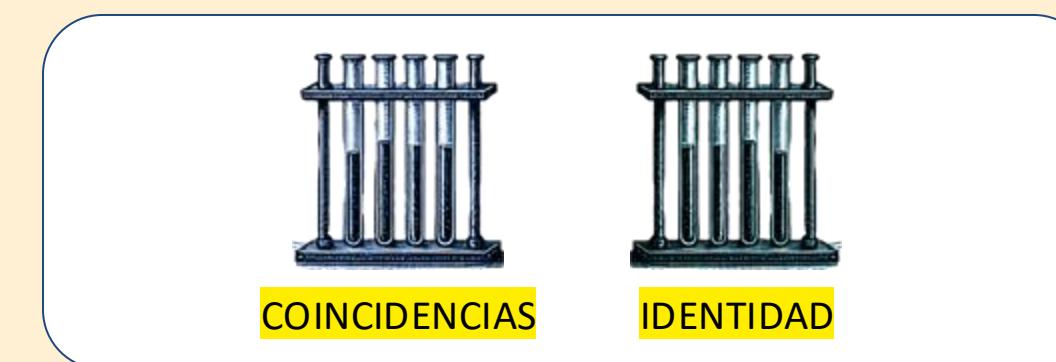
73% - 78%



26% - 29%



21% - 24 %



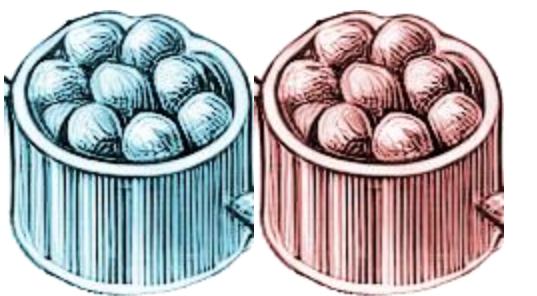
4% - 13%

# CASO DE ESTUDIO C16

Secuencia BOX212 (Ref. 9) - Long. máxima 625 residuos - Cantidad 8 secuencias.

Ref 9: Alineamientos que incluyen motivos lineales.

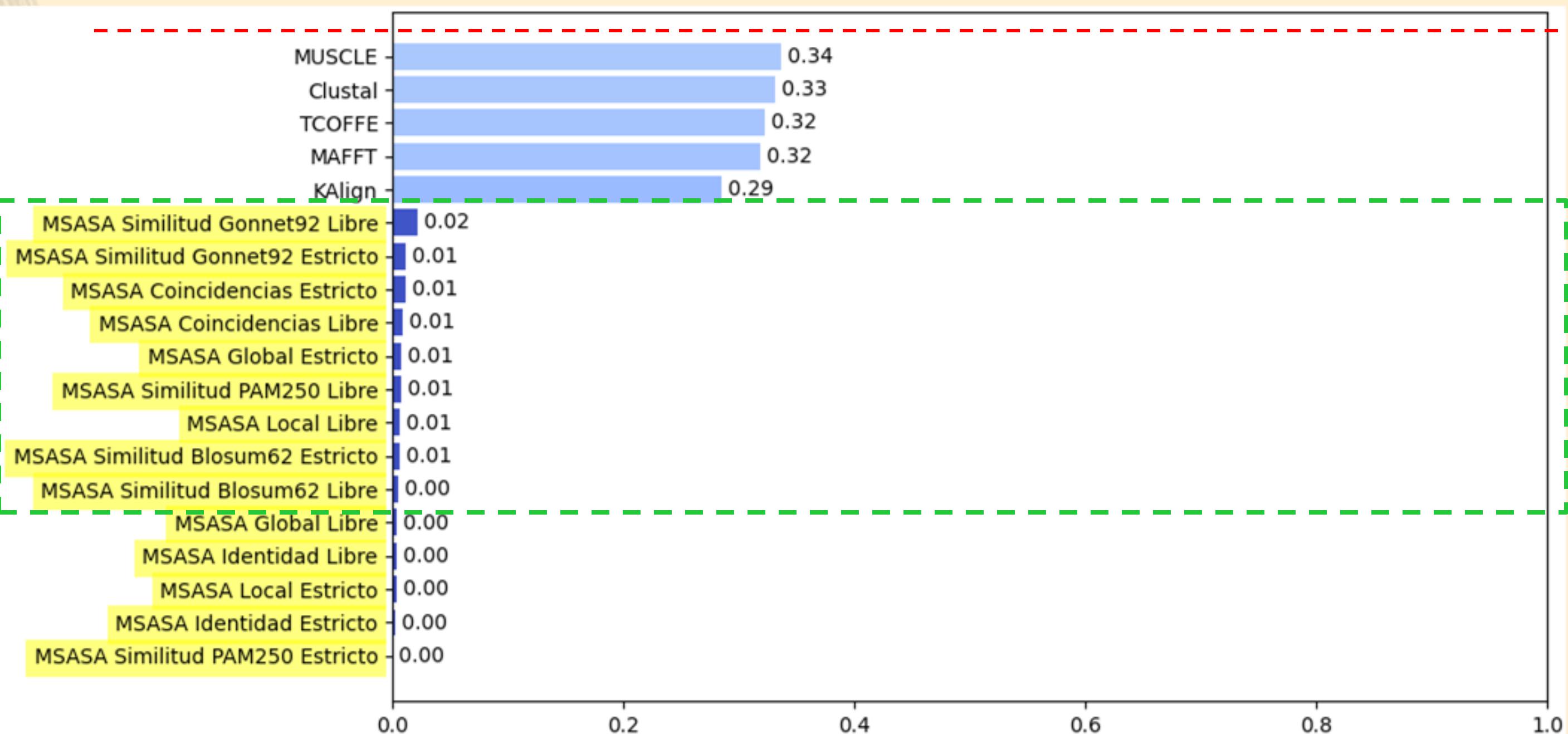
LL-BC



LL: Más de 400 residuos por secuencia

BC: Hasta 100 secuencias a alinear

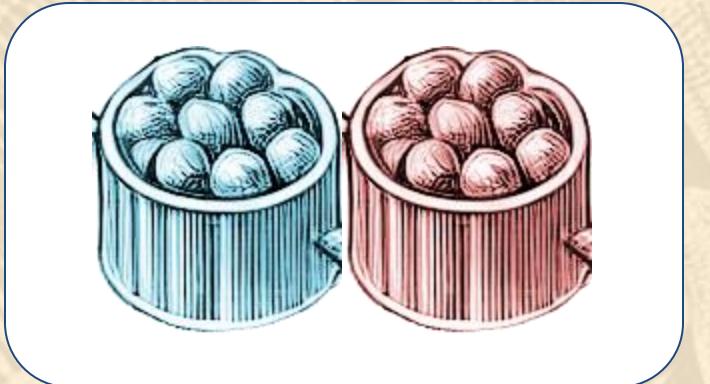
- C3 (Ref 1)
- C6 (Ref 2)
- C10 (Ref 3)
- C13 (Ref 4)
- C14 (Ref 5)
- C16 (Ref 9)**
- C22 (Ref 10)



# CASO DE ESTUDIO C16

Secuencia BOX212 (Ref. 9) - Long. máxima 625 residuos - Cantidad 8 secuencias.  
Ref 9: Alineamientos que incluyen motivos lineales.

LL-BC



LL: Más de 400 residuos por secuencia

## BC: Hasta 100 secuencias a alinear

- C3 (Ref 1)
  - C6 (Ref 2)
  - C10 (Ref 3)
  - C13 (Ref 4)
  - C14 (Ref 5)
  - C16 (Ref 9)**
  - C22 (Ref 10)

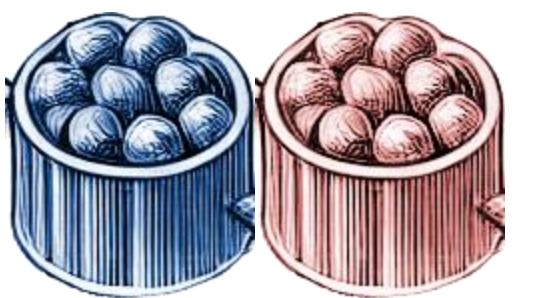
Length: 1463  
Identity: 184/1463 (12.6%)  
Similarity: 115/1463 ( 7.9%)  
Gaps: 1185/1463 (81.0%)  
**Score: 54.5**

ExxxxxxxxxMLixxxxxGLsLLxILxxxQLplxxxxxxxxenxxxxpACxDx  
eQlgxCxxCSRCPPGkxmTxxCGxxHSxDtxCxpCxPDxYxDxwSxexKCx1Cx  
xxFxEkaxCSxtSKixCxxCxxPGFHCxSxxxSDCxxCkxHxxxxCqpGxxVxx  
kDteCxPCsxtGTFSdxxSxxgxCRpwlTnCSxxGxxxxxpGttxSDaICxPPpTx  
xx  
xtxNpGHMpxxxLLxxfILvsxLVL  
ivaLiFvxxrkxxkQlxkqxxxxxxxxxxxxsx1xxxxkxxxxxxxxxxxxsc  
xxxxxxxxx1xx  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxpfxxxxxxxxxNgxLtqxxxGxxxx  
xxpxxxxxxxxxxx  
xx  
xx  
xx  
ExxxxxxxxxxxxLxxxxqEDGxSxxhFPexExxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxDxxxxtGmxxxxxE1

```
# Length: 1518  
# Identity:      0/1518 ( 0.0%)  
# Similarity:    0/1518 ( 0.0%)  
# Gaps:          1518/1518 (100.0%)  
# Score: 0.0
```

# CASO DE ESTUDIO C1

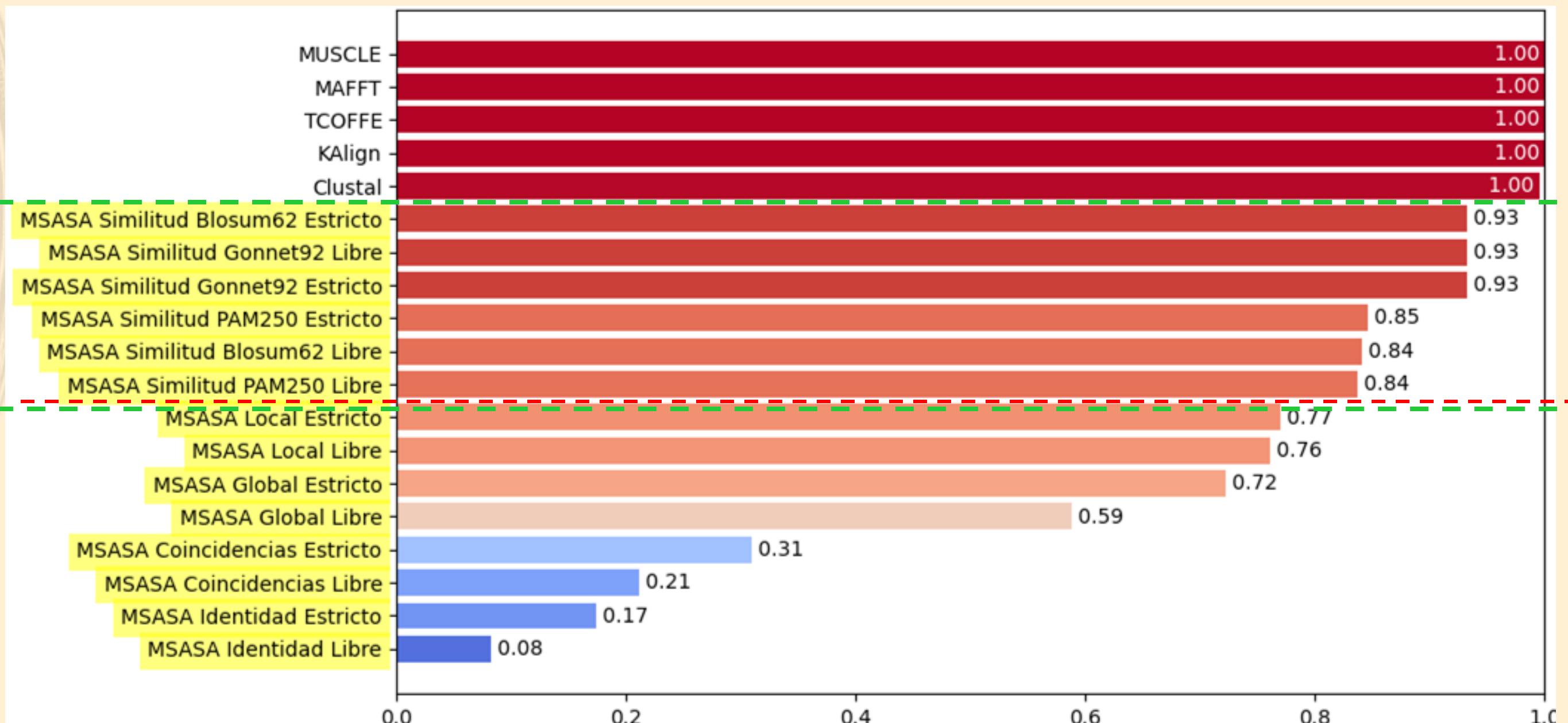
CL-BC



CL: Hasta 100 residuos por secuencia  
BC: Hasta 100 secuencias a alinear

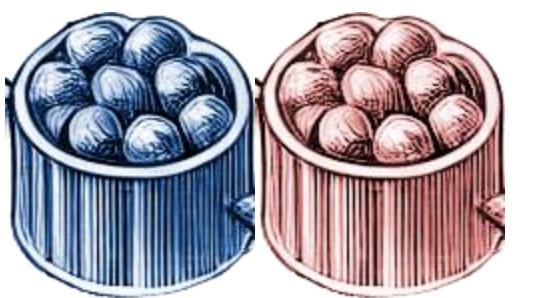
C1 (Ref 1)  
C4 (Ref 2)  
C7 (Ref 3)  
C18 (Ref 10)

Secuencia BB12014 (Ref. 1) - Long. Máxima 52 residuos - Cantidad 9 secuencias.  
Ref 1: Alineamiento con bajo porcentaje de identidad.



# CASO DE ESTUDIO C1

CL-BC

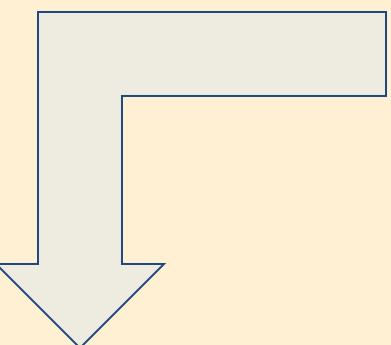


CL: Hasta 100 residuos por secuencia  
BC: Hasta 100 secuencias a alinear

C1 (Ref 1)  
C4 (Ref 2)  
C7 (Ref 3)  
C18 (Ref 10)

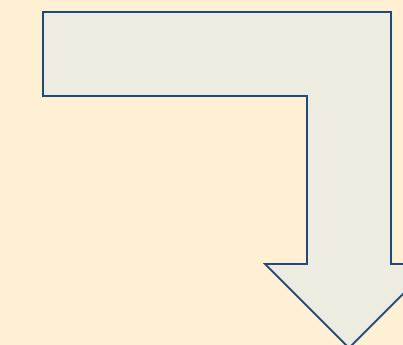
# Length: 52  
# Identity: 52/52 (100.0%)  
# Similarity: 39/52 (75.0%)  
# Gaps: 0/52 ( 0.0%)  
# Score: 196.0

# Length: 55  
# Identity: 47/55 (85.5%)  
# Similarity: 37/55 (67.3%)  
# Gaps: 6/55 (10.9%)  
# Score: 176.0



>BALIBASE

FTQQQLxxLEKxFQKxxxKQYLxTxxREELAQxLGLTEX  
QIKIWFQNRRxKx



>MUSCLE

FTQQQLxxLEKxFQKxxxKQYLxTxxREELAQxLGLTEX  
QIKIWFQNRRxKx

>BLOSUM62\_ESTRICTO

xxxFTxQQLxxLEKxFHxKQYLxTxxREELAQxLGLTEX  
QIKIWFQNRRxKx



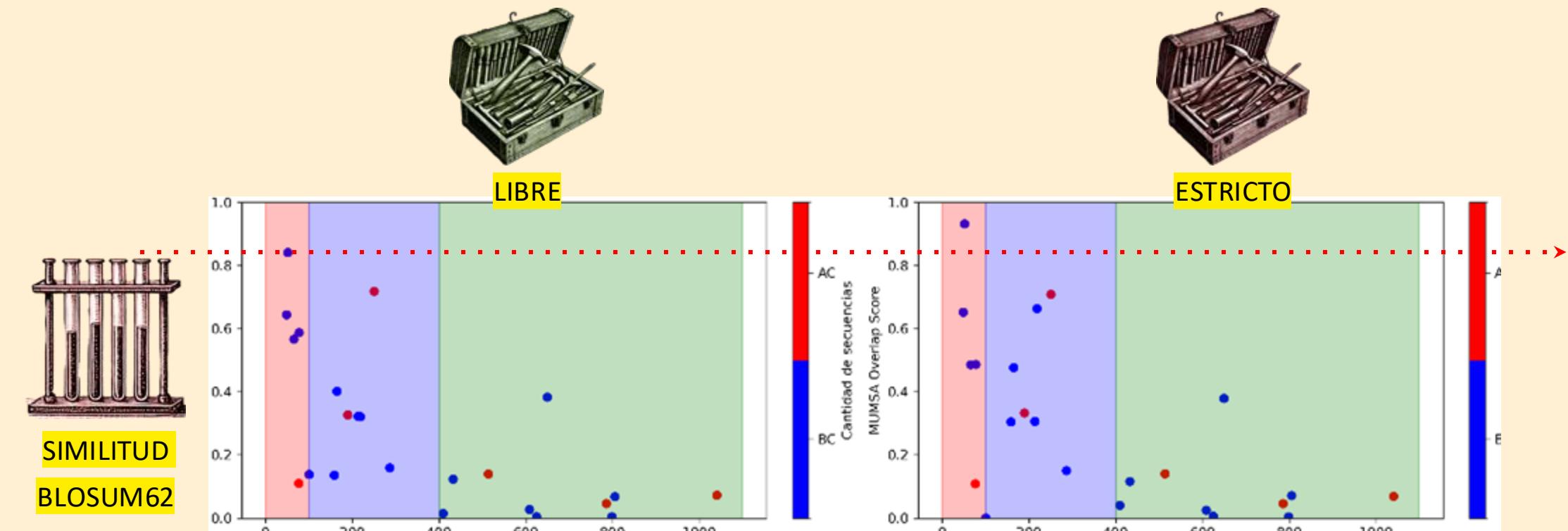
# **CONCLUSIONES**

# LA IMPLEMENTACIÓN MSASA

## 1. CALIDAD DE LOS RESULTADOS

Alguno grupos de estudio encuentran una solución similar a los software MSA del estado del arte.

Estricto vs Libre



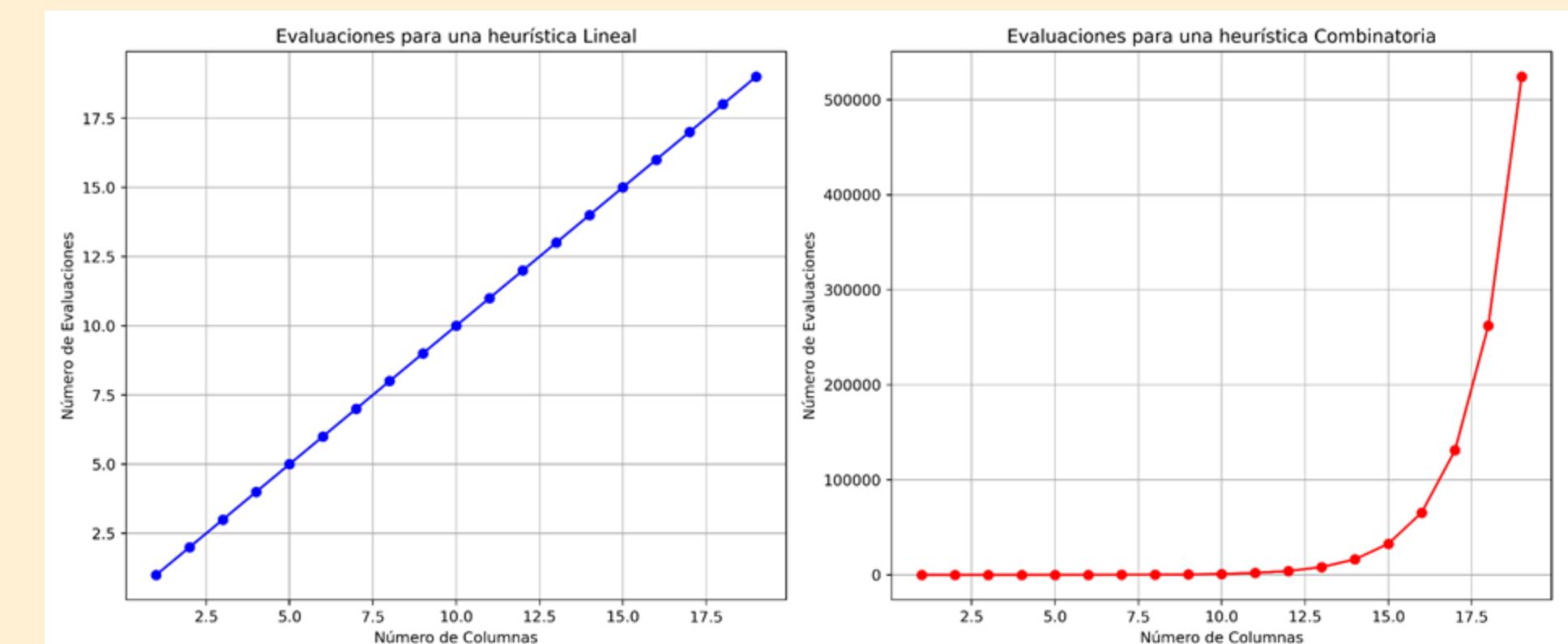
## 2. IMPORTANCIA DE LA

### INFORMACIÓN BIOLÓGICA

Justamente son las FO que incorporan matrices de similitud las que tienen mejores rendimientos.

## 3. COMPLEJIDAD COMPUTACIONAL

Comparación columna por columna.



# HIPÓTESIS

**H1.** Implementaciones de **Enfriamiento Simulado** (SA) resuelven el problema MSA.

“Se puede validar que MSASA puede obtener resultados alineamientos de secuencias múltiples.”

**H2.** Estos algoritmos son competitivos con otras soluciones MSA del estado del arte.

“En particular la implementación MSASA para ciertos escenarios de prueba obtiene valores que se acercan a los obtenidos con las otras alternativas del estado del arte al comparar los resultado con los MSA definidos por los expertos que confeccionaron la base de datos de prueba.”

**H3.** Es posible combinar diferentes metaheurísticas para encontrar soluciones MSA suficientemente buenas.

“El algoritmo MSASA ofrece un método de trabajo lo suficientemente flexible para incorporar nuevas funciones objetivo que puedan aumentar la calidad de los resultados.”

**H4.** Se puede crear un marco de trabajo para comparar diferentes soluciones MSA para clasificarlos según la calidad de los resultados.

“El marco general de trabajo es lo suficientemente flexible para incorporar nuevas implementaciones MSA como así métricas de calidad para evaluar los resultados.”

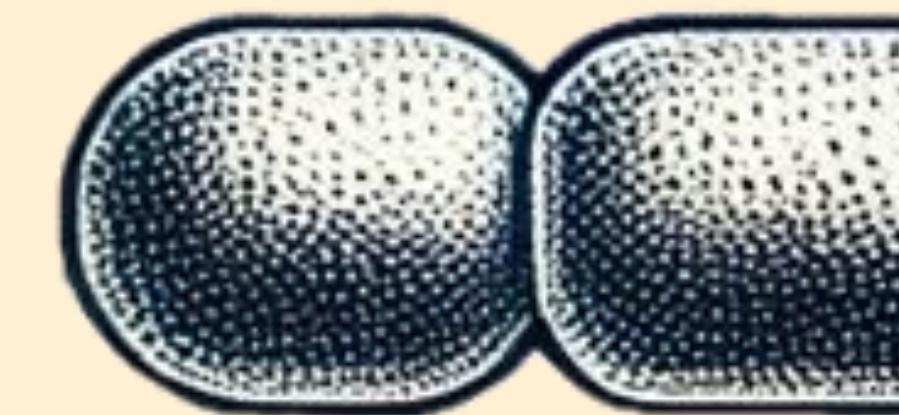
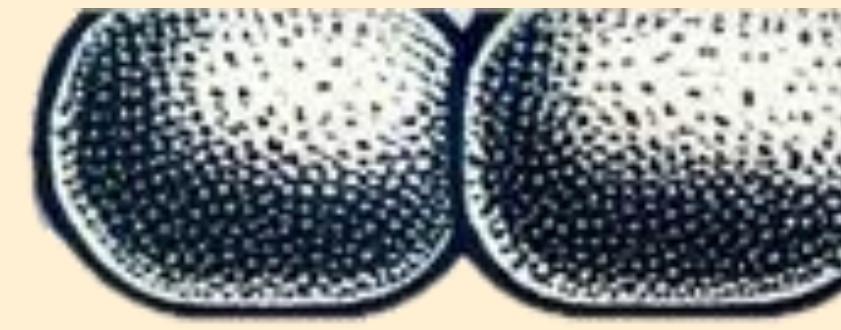




# PERSPECTIVAS

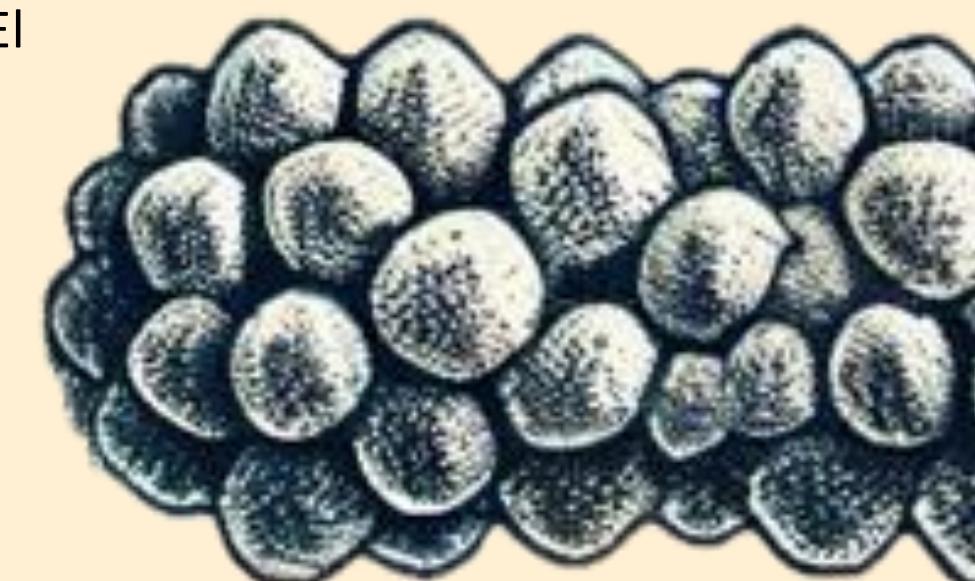
## 1. OBTENCIÓN DE DATOS BIOLÓGICOS

El problema de secuenciar, reconstruir y almacenar las secuencias biológicas está resuelto. El problema hoy en día es como convertir esos datos en información.



## 2. NUEVAS FUNCIONES OBJETIVO

Adaptación del código fuente para evaluar más de una columna a la vez. Además podrían analizarse las filas junto a las columnas para encontrar mejores funciones objetivo (diferenciación de los tipos de GAP).



## 3. MARCO DE TRABAJO FLEXIBLE

El problema de secuenciar, reconstruir y almacenar las secuencias biológicas está resuelto. El problema hoy en día es como convertir esos datos en información.



## CÓDIGO FUENTE & ARCHIVOS:

[https://github.com/agdiaz/msasa\\_2024](https://github.com/agdiaz/msasa_2024)

L

**CIERRE**



# ¡MUCHAS GRACIAS! MERCI BEAUCOUP ! DANK JE WEL !

♪ Tarda en llegar y al final hay recompensa ...



Al Dr. Patricio Yankilevich, UTN-FRBA por introducirme en la Bioinformática.



A mi directora Dra. Gabriela, los jurados Dr. Gustavo, Dra. Silvana y Dr. Galo, a los miembros de la UNQ en especial a la Dra. Carolina Cerrudo y a mis compañeros del cohorte 2017.



Al Dr. Wim Vranken, todo el equipo de Bio2Byte, y al Dr. Rachid Tahzima de ILVO en Bélgica.



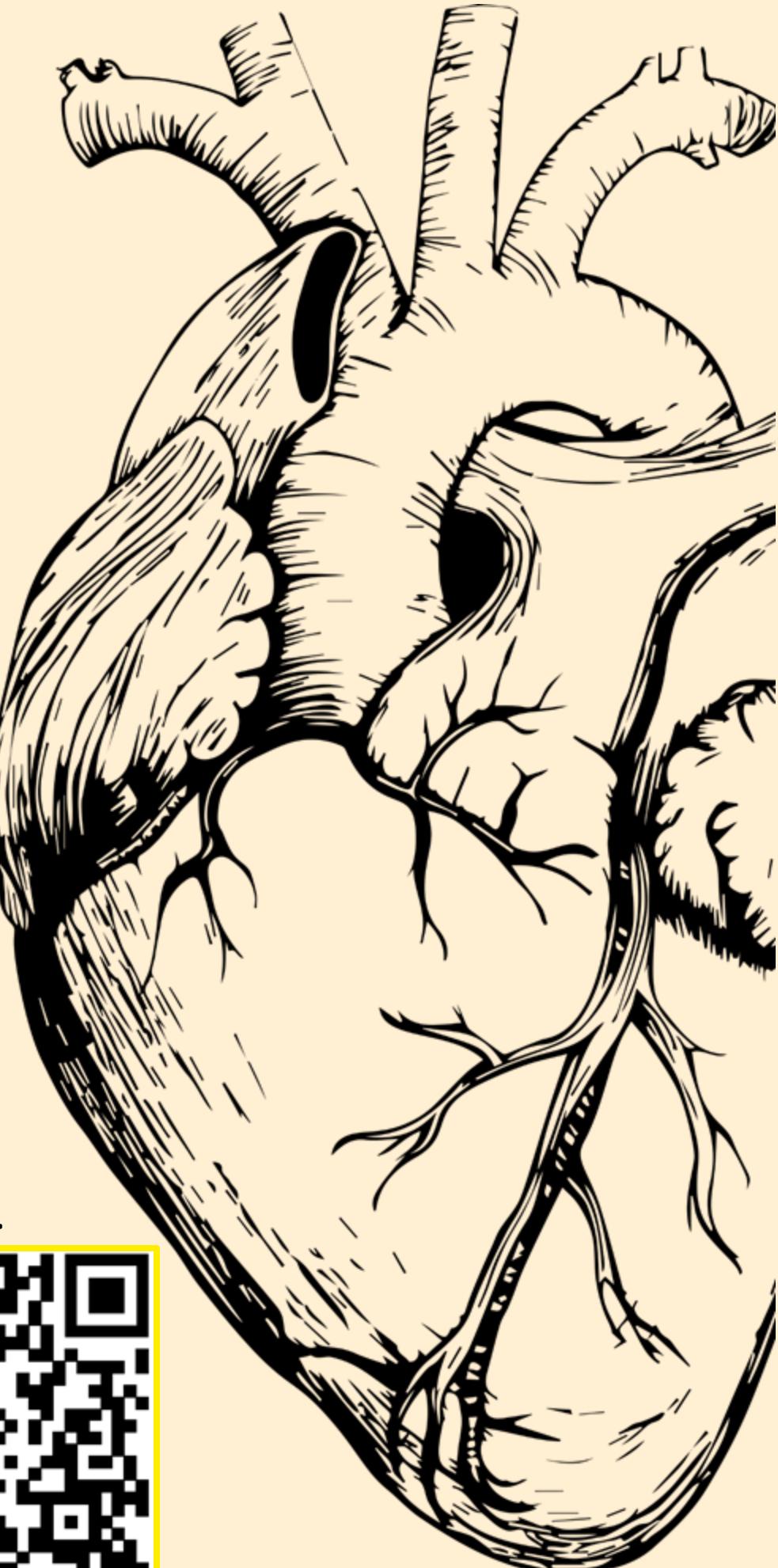
A la familia de Restorando (luego TheFork), a Franco y Gabriel por todas las oportunidades junto a libertad de trabajar y estudiar la maestría a la vez. A Norberto por su apoyo mientras estudiaba ingeniería y trabajaba en ZL.



A mis amigos que voy descubriendo en mi aventura de vivir en el extranjero, a Caroline y la familia Vankerkhoven.



A mis grandes amigos que están siempre conmigo y a los familiares en Argentina que me siguen a la distancia.





# PREGUNTAS

“Efectividad del método de búsqueda  
metaheurístico **enfriamiento simulado** en la  
determinación de una secuencia consenso a  
partir de múltiples secuencias”

**ANEXOS**

Aspirante: Ing. Adrián Díaz  
Promotora: Dra. Gabriela Minetti

25 de octubre de 2024 - Online - 11h30 (ARG) / 16h30 (BEL)



Universidad  
Nacional  
de Quilmes

# LA CÉLULA

Unidad mínima de la vida

- Estructura.
- Clasificación.

El ADN

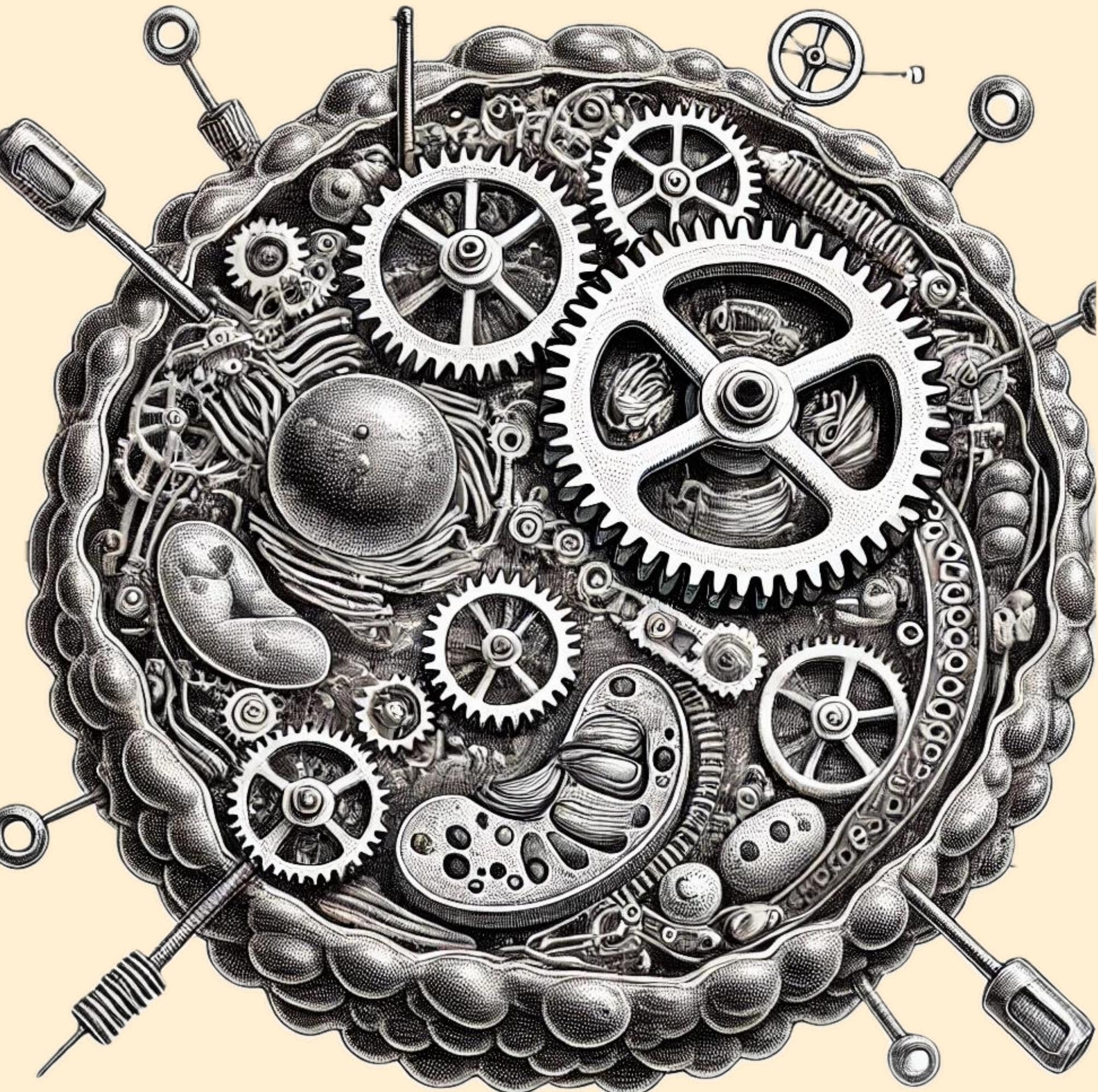
- Descubrimiento.
- Indicios.
- La información genética.

El ARN

- Los intermediarios.
- Mensajeros.

Las proteínas

- Los ejecutores en la vida.
- Plegamiento.
- Niveles de estructura.
- Funciones.



# EVOLUCIÓN DE LA BIOINFORMÁTICA

## 1. DESCUBRIMIENTO DE LAS ESTRUCTURAS MOLECULARES

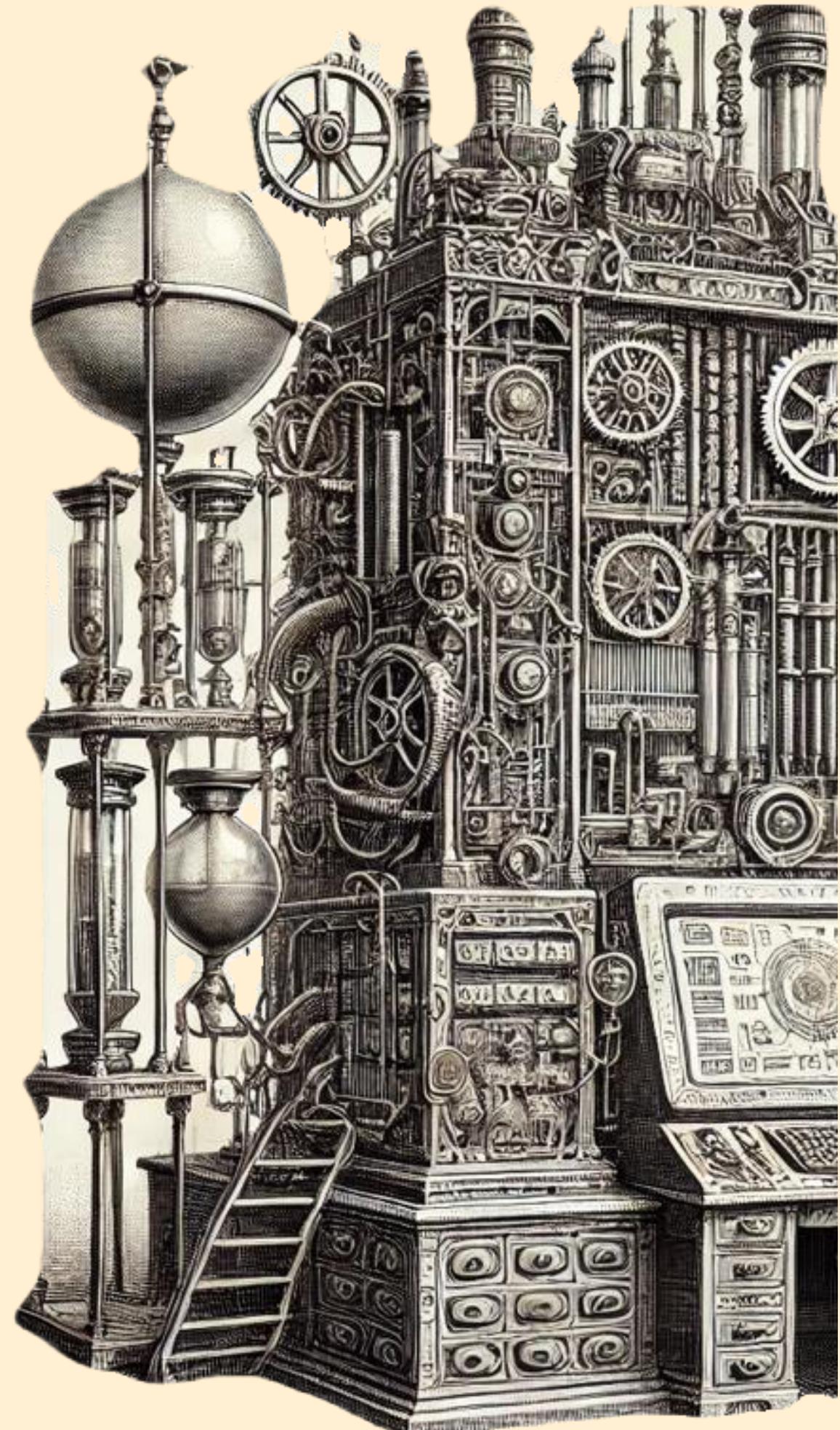
- Estructura molecular.
- Funciones biológicas.

## 2. EVOLUCIÓN DE LAS TÉCNICAS DE SECUENCIACIÓN

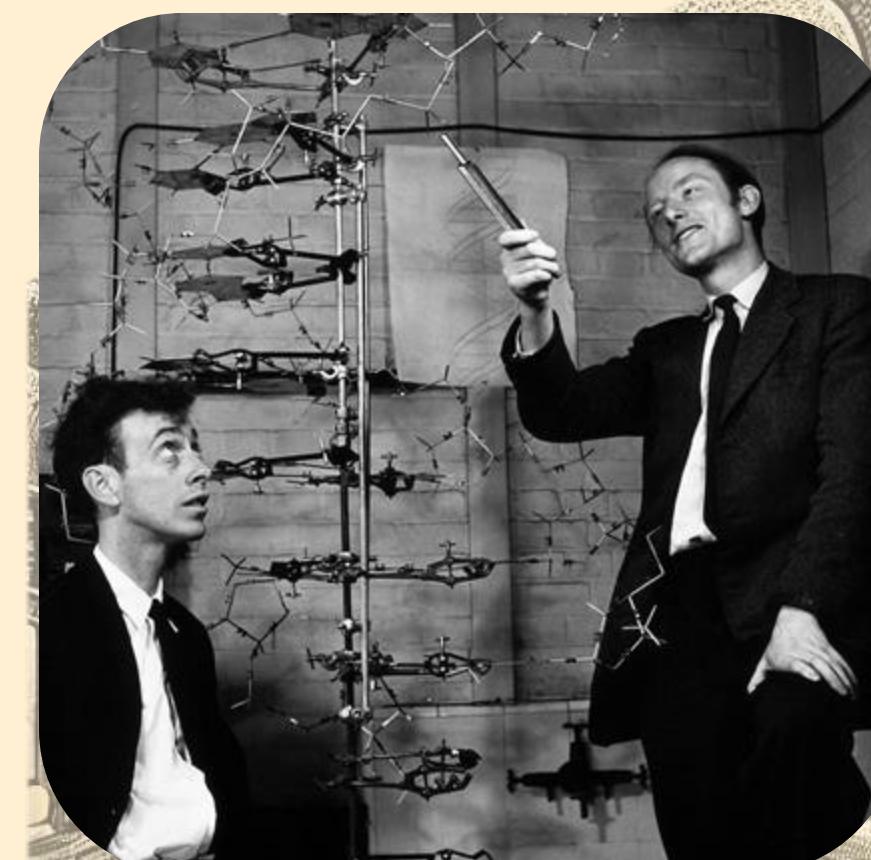
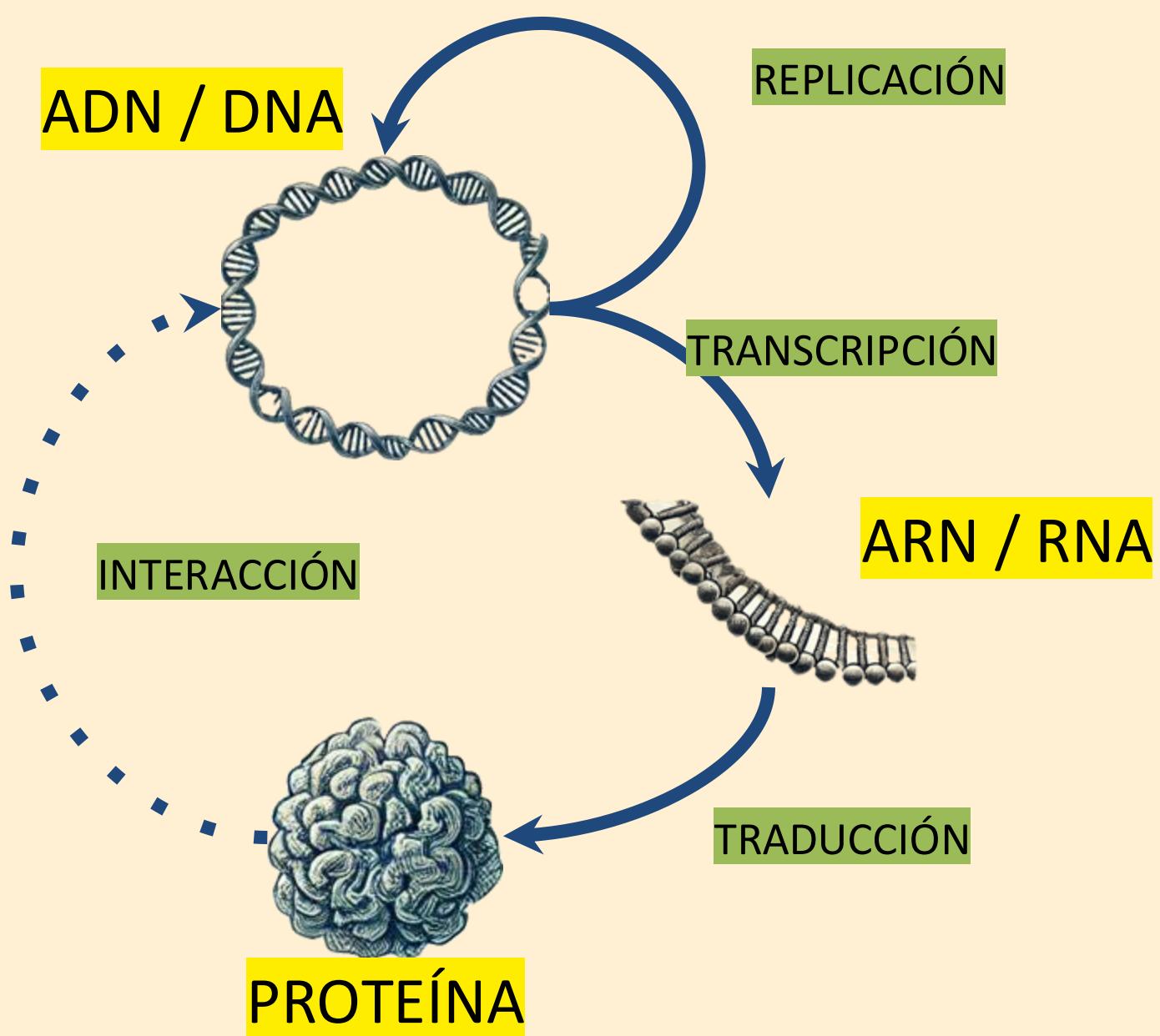
- PCR.
- Método Sanger.

## 3. IMPACTO DE LA SECUENCIACIÓN MASIVA

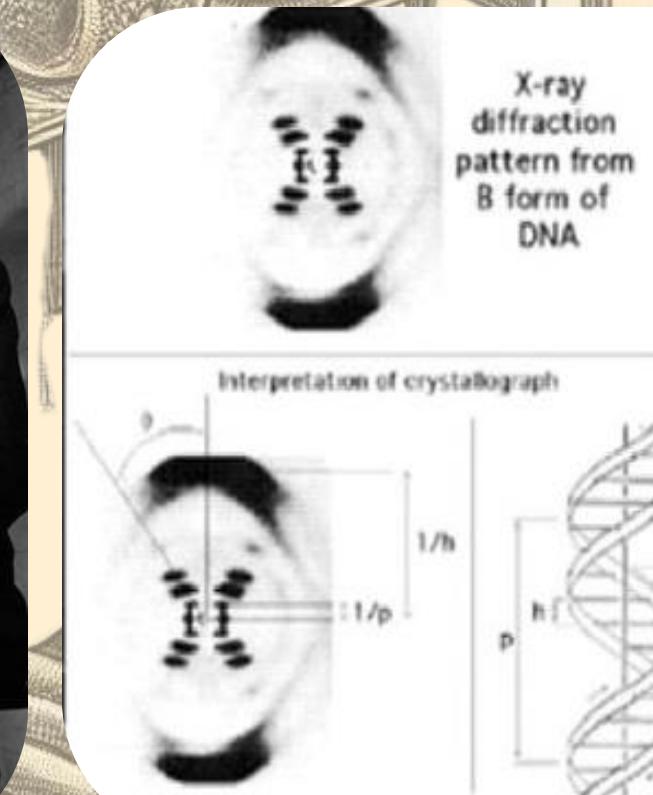
- Nuevas tecnologías.
- Estudios de grupos.
- Aparición de las -ómicas.



# DOGMA CENTRAL DE LA BIOLOGÍA MOLECULAR



*Watson y Crick*



*Franklin*



# DOGMA CENTRAL DE LA BIOLOGÍA MOLECULAR

## 1. La cognición celular gobierna el flujo de información biológica

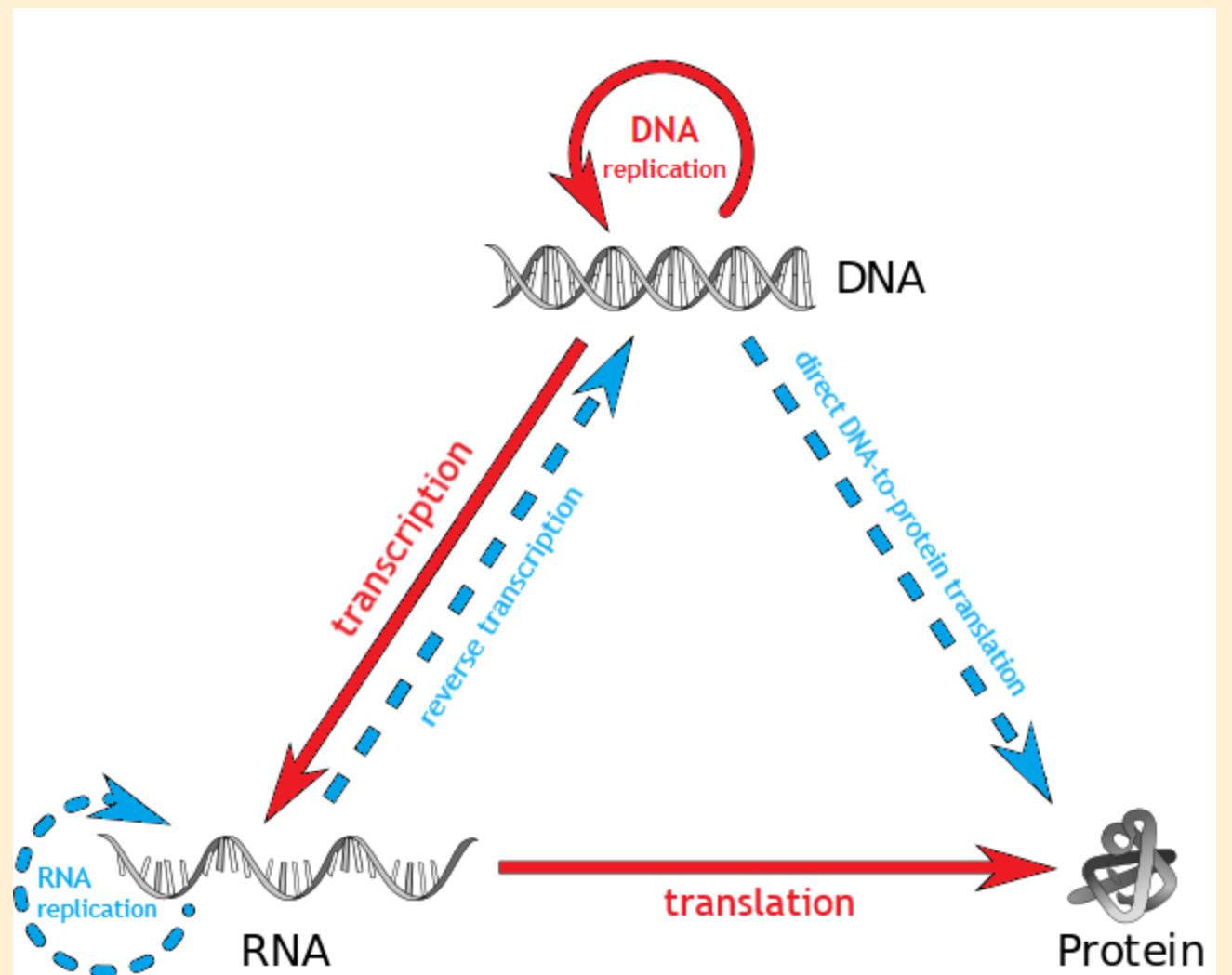
Las células no solo responden a estímulos pasivos, sino que activamente procesan y dirigen la información biológica. Este control es clave para entender cómo los organismos manejan su entorno y funciones internas.

## 2. El nuevo dogma debe ser una teoría celular, no genética

A diferencia de la visión clásica centrada en los genes, ahora se reconoce que las células son las protagonistas en la regulación de la información. La importancia del ambiente celular es fundamental para comprender los procesos biológicos.

## 3. El flujo de información es recíproco entre todos los niveles biológicos

En lugar de un flujo unidireccional de la información desde el ADN, la comunicación es bidireccional y dinámica. Esto implica que la influencia entre los niveles biológicos es mutua, sin un nivel superior de control.



Central Dogma, Philippe Hupé, Wikimedia, 2022,  
[https://commons.wikimedia.org/wiki/File:Central\\_dogma\\_of\\_molecular\\_biology\\_colorized%2Bspecial\\_transfer.png](https://commons.wikimedia.org/wiki/File:Central_dogma_of_molecular_biology_colorized%2Bspecial_transfer.png)

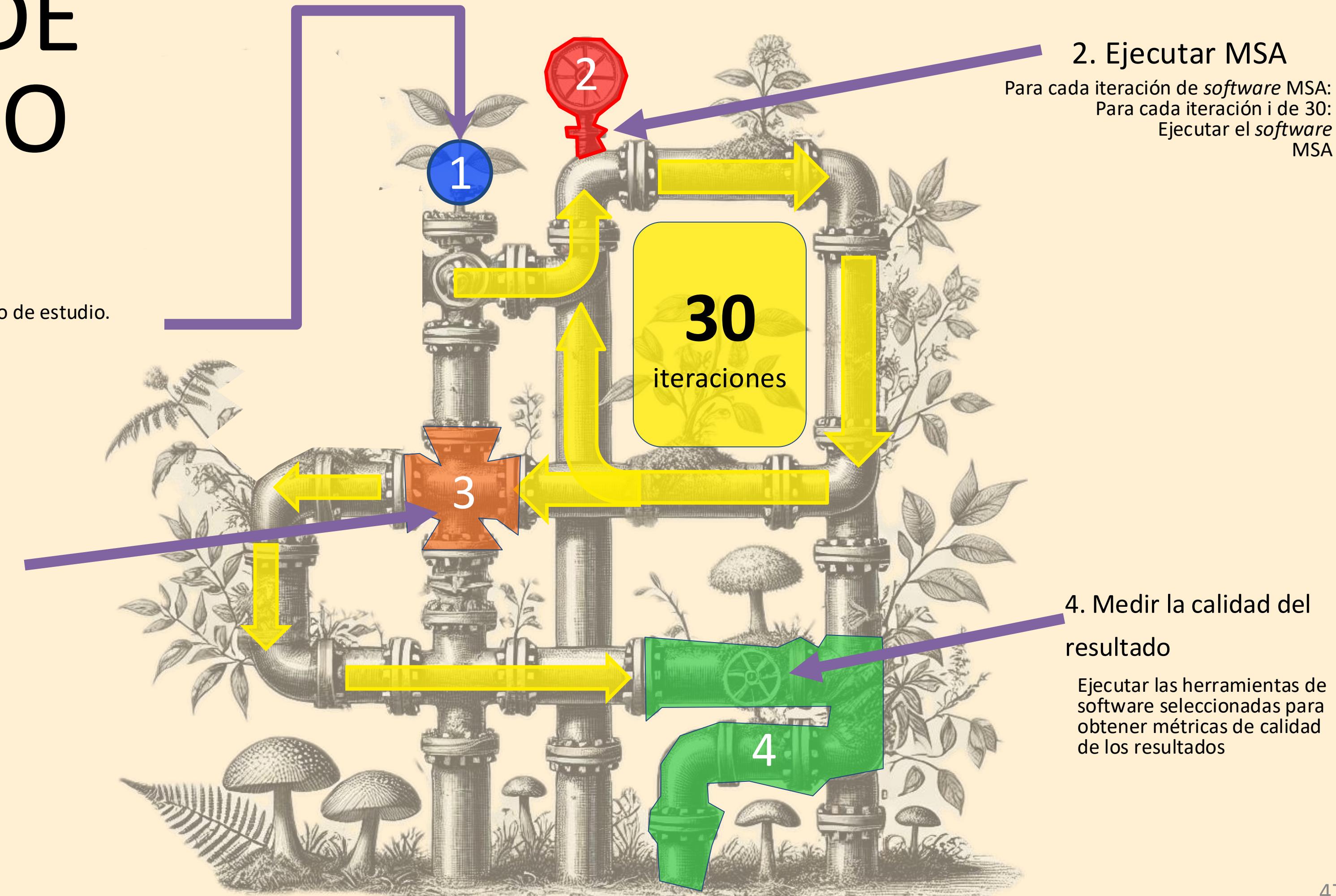


# FLUJO DE TRABAJO

## 1. Archivos de entrada

- a. Secuencias a alinear de un caso de estudio.
- b. MSA esperado (Bali Score).

3. Tomar uno de los 30 resultados como la muestra representativa



# ENERGÍA & ESTADOS VECINOS

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	...	$C_m$
$SECUENCIA_a$	M	V	Y	M	-	-	M	...	Y
$SECUENCIA_b$	M	I	Y	M	-	M	-	...	I
$SECUENCIA_c$	M	I	Y	I	-	M	-	...	I
$SECUENCIA_d$	M	I	Y	I	-	M	-	...	I
$SECUENCIA_e$	M	I	Y	T	-	M	T	...	I
...	M	...	...	...	...	...	...	...	...
$SECUENCIA_n$	M	V	V	S	-	M	M	...	I
<b>Puntaje</b>	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	...	$P_m$

Ejemplo de un MSA visualizado utilizando el software JALVIEW donde se pueden ver los residuos alineados junto a un logo que representa los residuos más frecuentes y la secuencia consenso

Ejemplo de un MSA visualizado utilizando el software JALVIEW donde se pueden ver los residuos alineados junto a un logo que representa los residuos más frecuentes y la secuencia consenso

## Algoritmo 8 Función Objetivo genérica

**Require:** secuencias

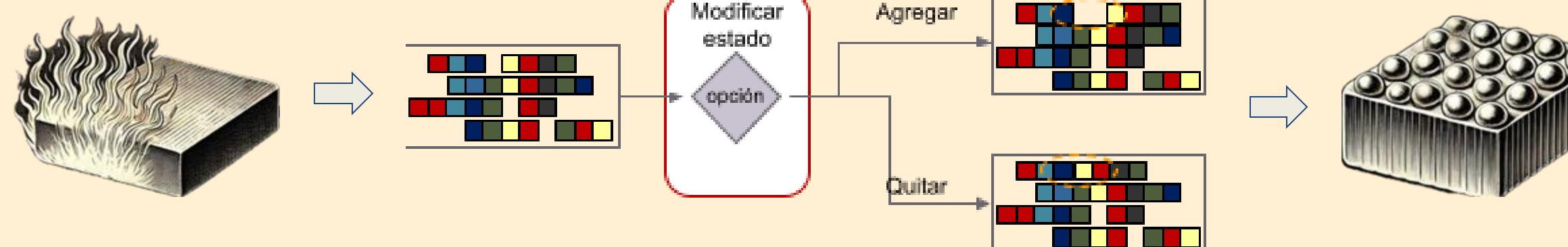
**Require:** cantidad\_secuencias

**Require:** longitud\_máxima

**Require:** columnas

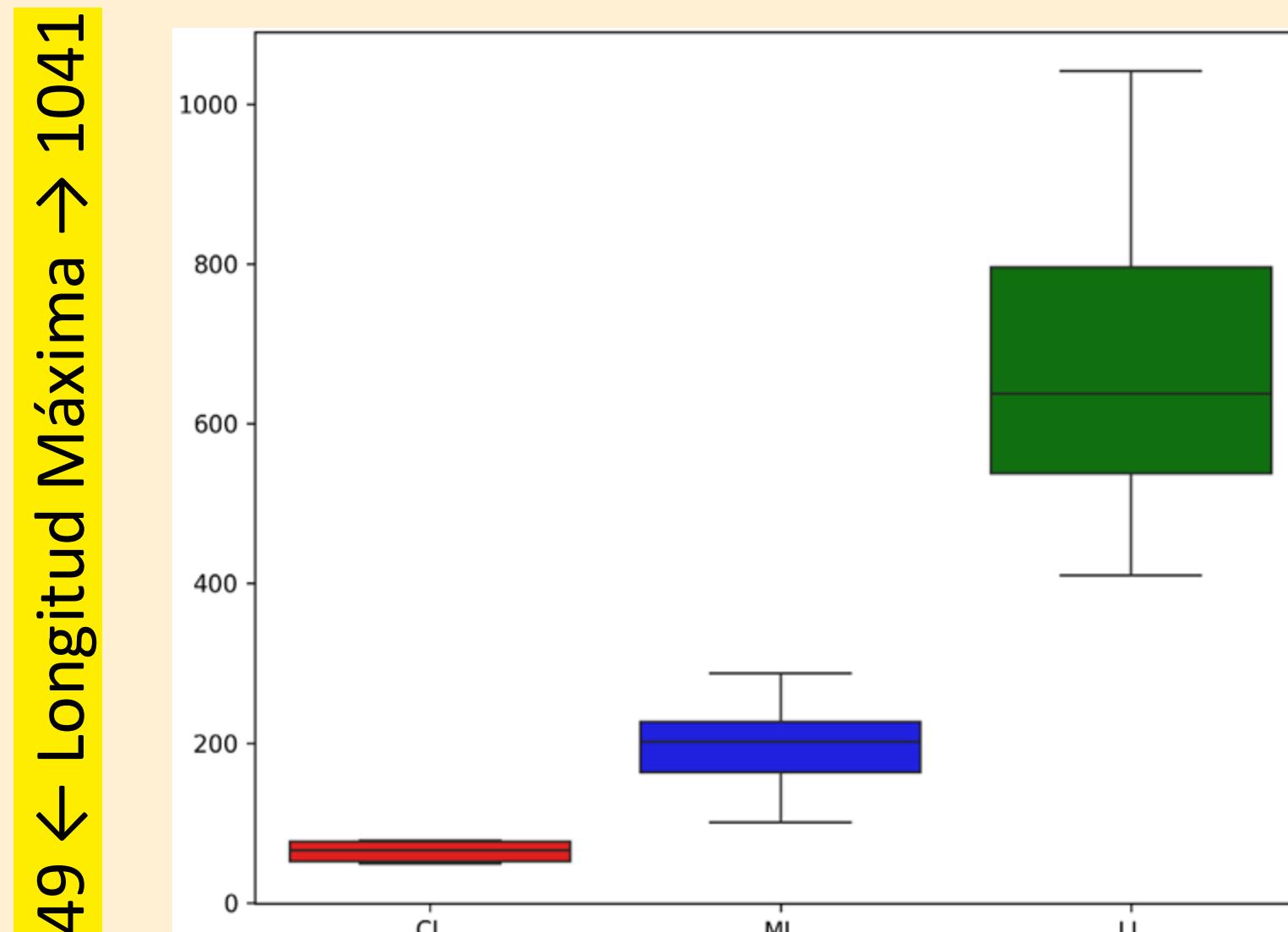
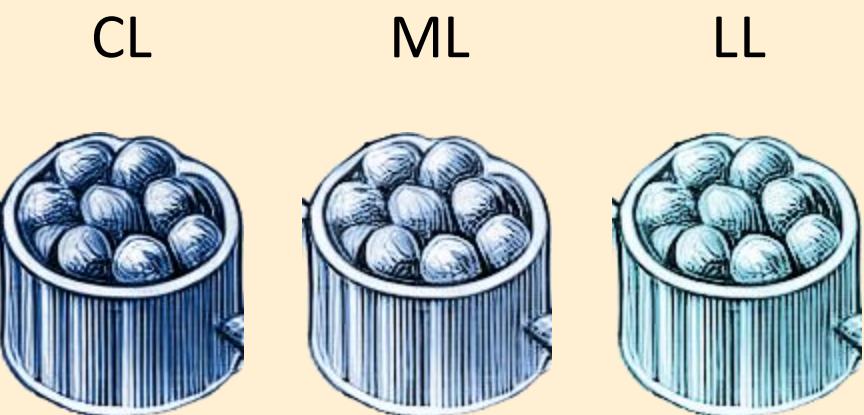
```

1: puntaje_total ← 0
2: for columna en columnas(secuencias) do
3:   puntaje_columna ← heurística(columna)
4:   puntaje_total ← puntaje_total + puntaje_columna
5: end for
6: return puntaje_total / divisor(cantidad_secuencias, longitud_máxima)
```



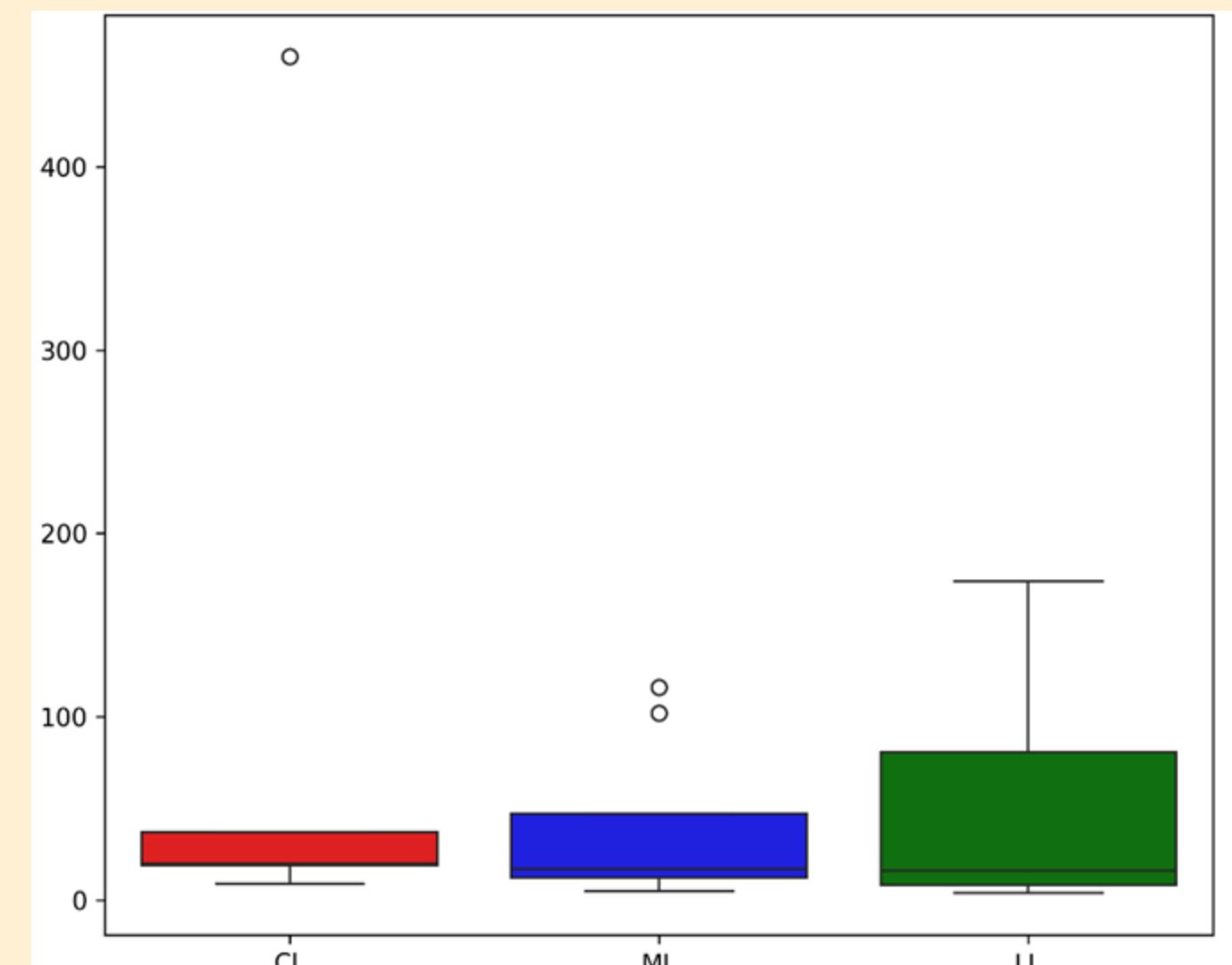
Algoritmo para determinar un estado vecino a partir de un MSA

# DIMENSIÓN LONGITUD DE SECUENCIAS



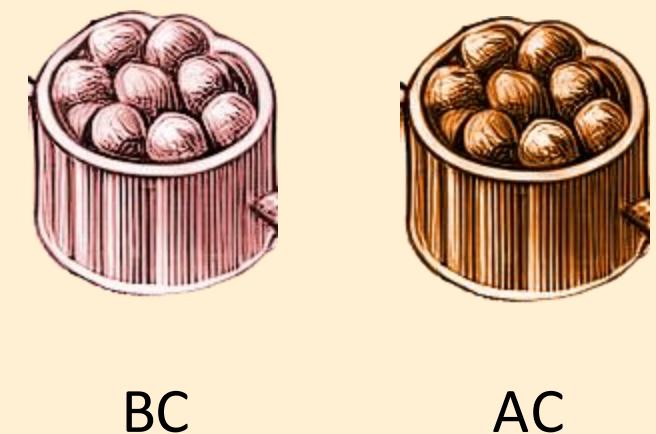
CL ← Grupo → LL

49 ← Longitud Máxima → 1041

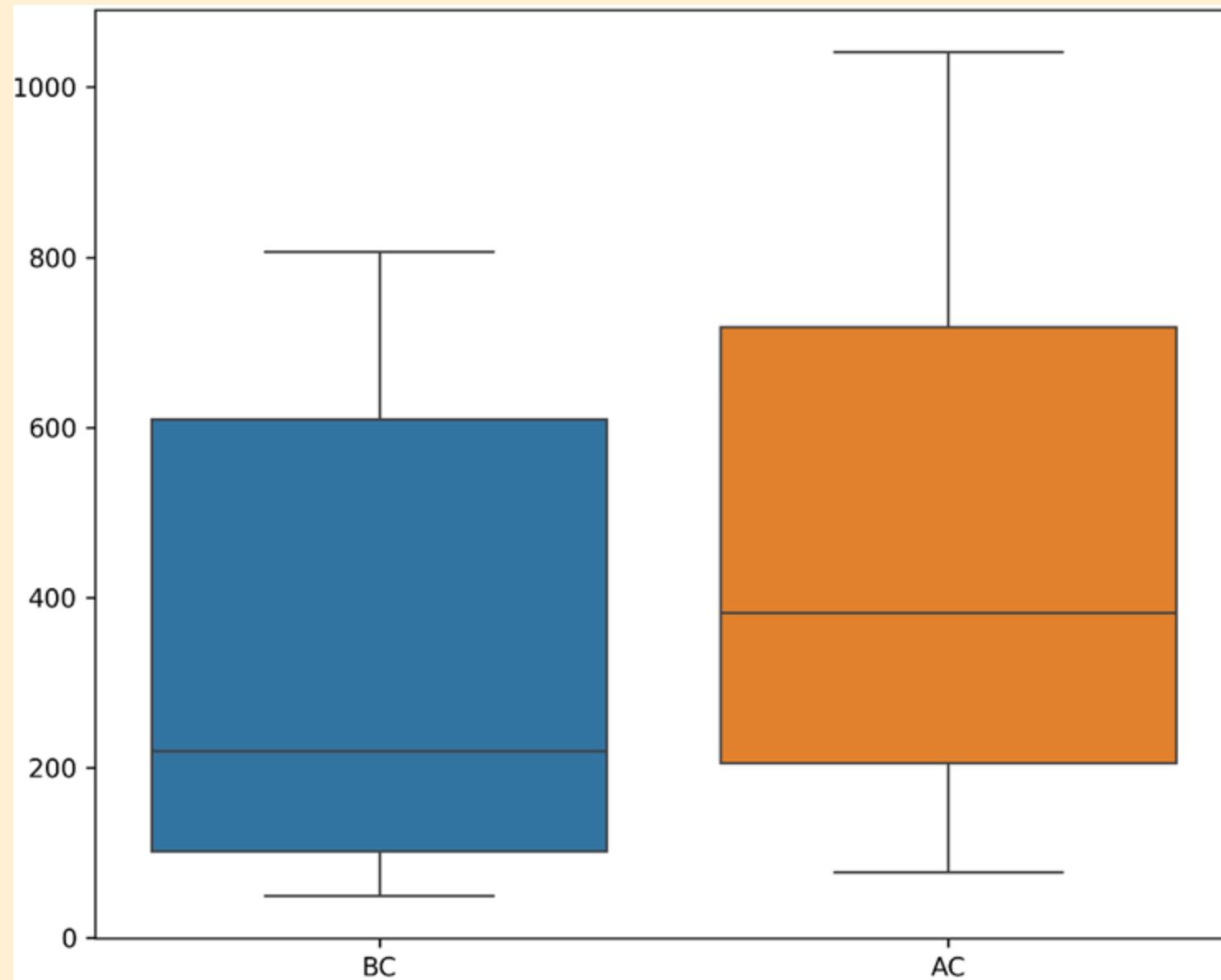


CL ← Grupo → LL

# DIMENSIÓN CANTIDAD DE SECUENCIAS

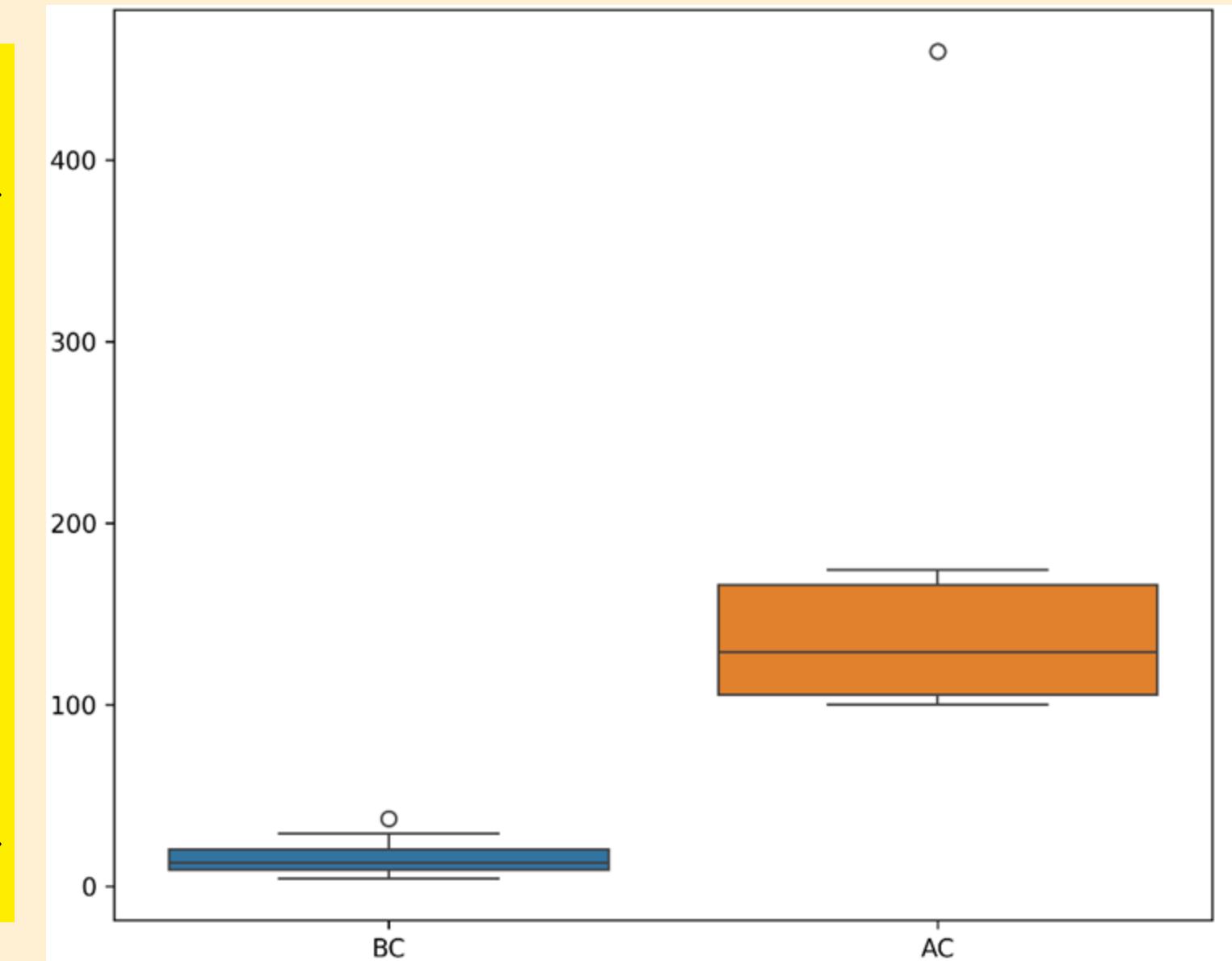


49 ← Longitud Máxima → 1041



BC ← Grupo → AC

4 ← Cant. de secuencias → 460



BC ← Grupo → AC

# ALGORITMOS SA

## Algoritmo 1 Algoritmo Needleman-Wunsch

- 1: Inicializar la matriz de puntuaciones con ceros.
- 2: **for** cada casilla de la matriz, excepto la primera fila y la primera columna **do**
- 3:     Calcular el puntaje de coincidencia si los caracteres de las secuencias son iguales.
- 4:     Calcular el puntaje de eliminación sumando la puntuación de la casilla superior y la penalización por brecha.
- 5:     Calcular el puntaje de inserción sumando la puntuación de la casilla izquierda y la penalización por brecha.
- 6:     Asignar el puntaje máximo entre los puntajes calculados a la casilla actual.
- 7: **end for**
- 8: Recorrer la matriz de puntuaciones desde la esquina inferior derecha hasta la esquina superior izquierda:
  - 9:     Si el puntaje proviene de una coincidencia o un empate: agregar los caracteres correspondientes a la secuencia alineada.
  - 10:    Si el puntaje proviene de una eliminación: agregar un espacio en la secuencia alineada.
  - 11:    Si el puntaje proviene de una inserción: agregar un espacio en la otra secuencia alineada.
- 12: **return** las secuencias alineadas.

## Algoritmo 2 Algoritmo Smith-Waterman

- 1: Inicializar la matriz de puntuaciones con ceros.
- 2: **for** cada casilla de la matriz, excepto la primera fila y la primera columna **do**
- 3:     Calcular el puntaje de coincidencia si los caracteres de las secuencias son iguales.
- 4:     Calcular el puntaje de eliminación sumando la puntuación de la casilla superior y la penalización por brecha.
- 5:     Calcular el puntaje de inserción sumando la puntuación de la casilla izquierda y la penalización por brecha.
- 6:     Asignar el puntaje máximo entre los puntajes calculados a la casilla actual.
- 7:     Si el puntaje máximo es negativo, establecerlo en cero.
- 8: **end for**
- 9: Encontrar el puntaje máximo en toda la matriz y ubicar la casilla correspondiente.
- 10: Recorrer la matriz de puntuaciones desde la casilla con el puntaje máximo hasta llegar a una casilla con puntaje cero:
  - 11:     Si el puntaje proviene de una coincidencia o un empate: agregar los caracteres correspondientes a la secuencia alineada.
  - 12:     Si el puntaje proviene de una eliminación: agregar un espacio en la secuencia alineada.
  - 13:     Si el puntaje proviene de una inserción: agregar un espacio en la otra secuencia alineada.
- 14:     Mover a la casilla vecina con el puntaje máximo.
- 15: **return** las secuencias alineadas.

# ALGORITMOS SA

## Algoritmo 6 Generación de nuevas secuencias con inserción o eliminación de *gaps*

**Require:** secuencias

**Require:** probabilidad\_adición\_delección = 0.5 ▷ Misma probabilidad por defecto

```
1: dirección_completar ← valor aleatorio entre 0 y 1
2: posiciones_gaps ← buscarPosicionesConGap(secuencias)
3: if valor aleatorio < probabilidad_adición_delección o posiciones_gaps está vacío then
4:     secuencias ← agregarGap(secuencias)
5: else
6:     secuencias ← removerGap(secuencias)
7: end if
8: return secuencias
```

## Algoritmo 7 Eliminación de columnas conteniendo únicamente el símbolo *gap* al inicio y al final de la matriz

**Require:** secuencias

```
1: while columna(secuencias, 0) contiene_solo gap o columna(secuencias, -1) contiene_solo
   gap do
2:     if la primera columna de secuencias contiene solo gap then
3:         secuencias ← secuencias desde la segunda columna hasta la última
4:     end if
5:     if la última columna de secuencias contiene solo gap then
6:         secuencias ← secuencias desde la primera columna hasta la penúltima
7:     end if
8: end while
9: return secuencias
```

**Algoritmo 13** Función Simulated Annealing modificado

**Require:** secuencias, temperatura\_actual, tasa\_de\_enfriamiento, temperatura\_min, límite\_sin\_cambios, cantidad\_de\_cambios, cantidad\_de\_estados\_vecinos, modo

```

1: iteración ← 0
2: sin_cambios ← 0
3: energía_actual ← función_objetivo(secuencias)
4: energía_mejor ← energía_actual
5: while temperatura_actual > temperatura_min do
6:   energía_iteración ← energía_actual
7:
8:   for contador_de_vecinos en rango(cantidad_de_estados_vecinos) do
9:
10:    nuevas_secuencias ← copia(secuencias)
11:    for contador_de_cambios en rango(aleatorio(1, cantidad_de_cambios)) do
12:      nuevas_secuencias ← generar_nuevo_estado(nuevas_secuencias)
13:    end for
14:    energía_estado_vecino ← función_objetivo(nuevas_secuencias)
15:    if mejor_energía_estado_vecino == Nulo or mejor_energía_estado_vecino > mejor_energía_estado_vecino then
16:      mejor_estado_vecino ← nuevas_secuencias
17:      mejor_energía_estado_vecino ← energía_estado_vecino
18:    end if
19:  end for
20:
21:  if modo == libre then
22:    acepta_estado ← debe_aceptar(energía_actual, mejor_energía_estado_vecino, temperatura_actual)
23:  else
24:    acepta_estado ← debe_aceptar(energía_mejor, mejor_energía_estado_vecino, temperatura_actual)
25:  end if
26:  if acepta_estado == Verdadero then
27:    secuencias ← copiar(nuevo_estado)
28:    energía_actual ← nueva_energía
29:  end if
30:  if mejor_energía_estado_vecino > energía_mejor then
31:    energía_mejor ← mejor_energía_estado_vecino
32:  end if
33:  if energía_iteración == energía_actual then
34:    sin_cambios ← sin_cambios + 1
35:  else
36:    sin_cambios ← 0
37:  end if
38:  temperatura_actual ← temperatura_actual × tasa_de_enfriamiento
39:  iteración ← iteración + 1
40:  if sin_cambios ≥ límite_sin_cambios then
41:    break
42:  end if
43: end while
44: return secuencias

```

▷ Interrumpir el ciclo de iteraciones

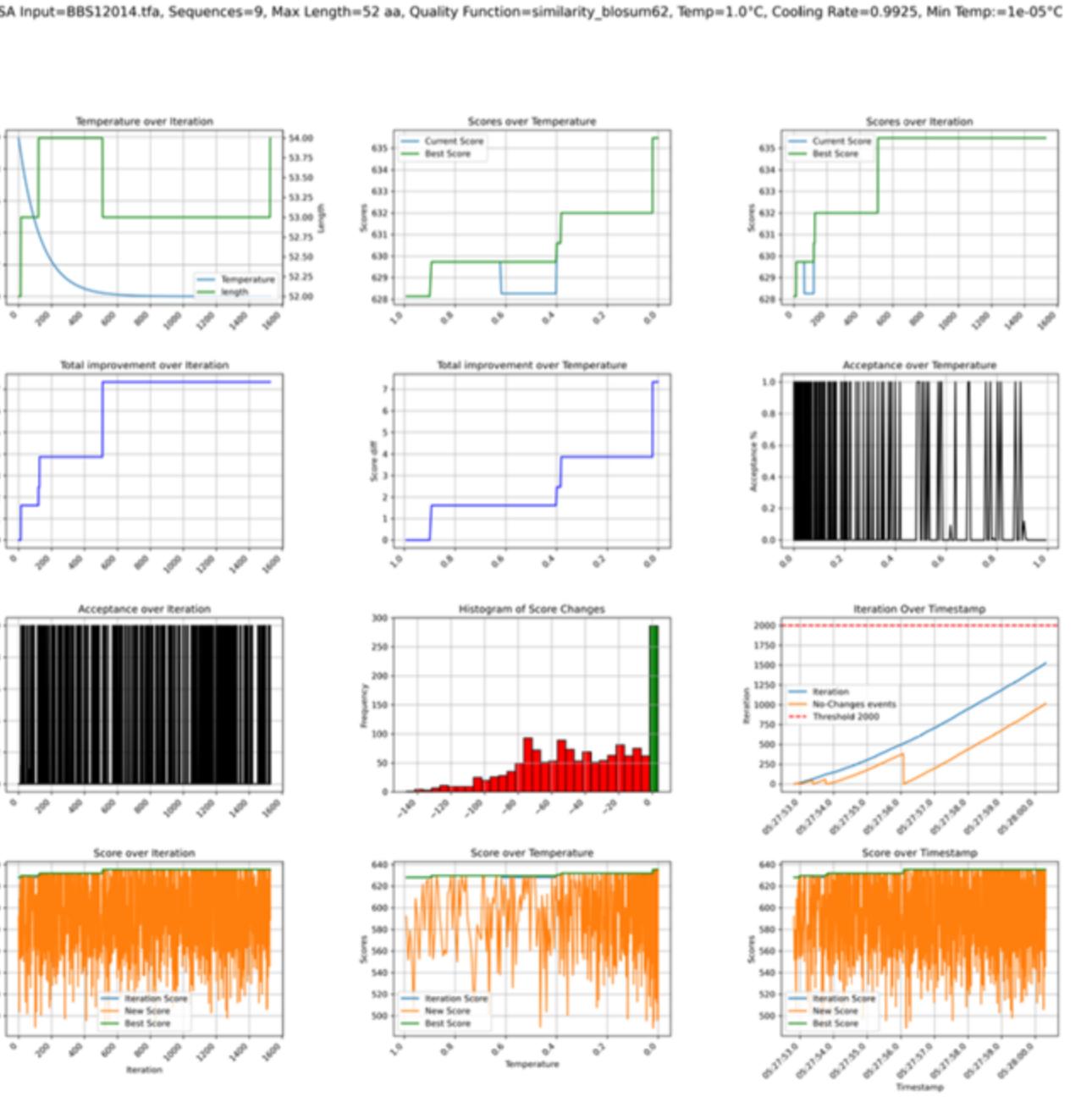


Figura 9: Gráficas diagnóstico para la confección del MSA del caso BBS12014 usando la heurística Similitud Blosum62 en modo libre.

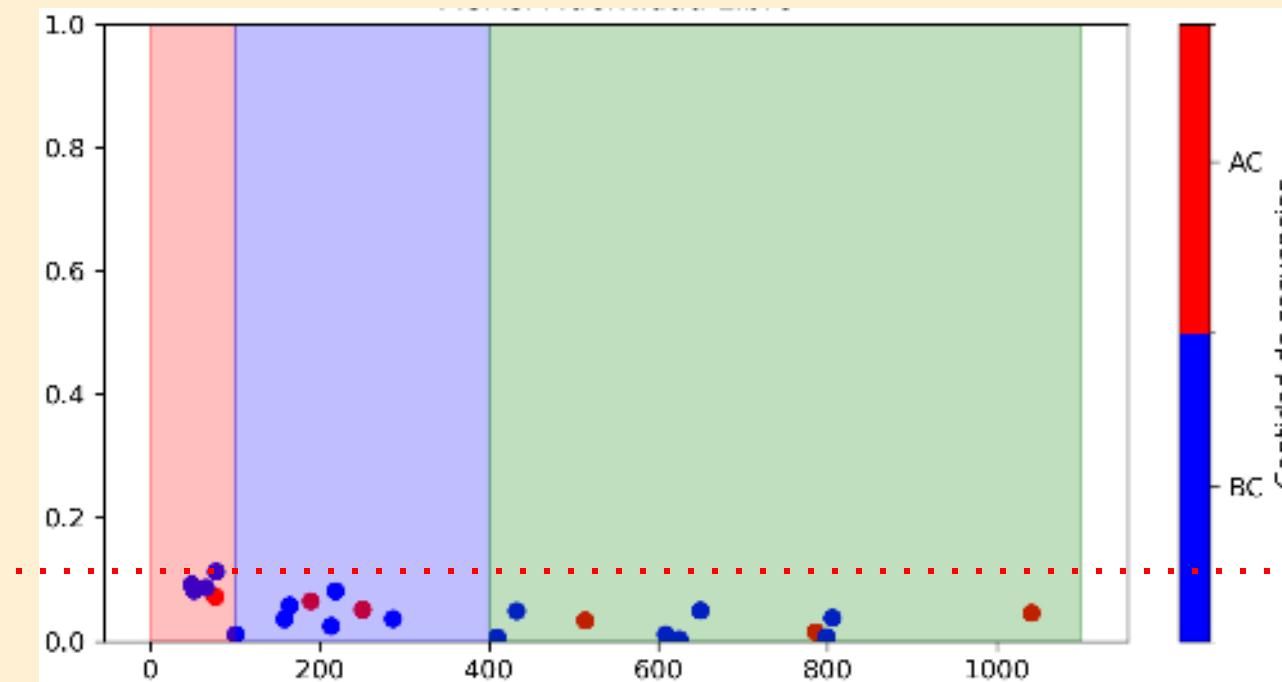
# OS



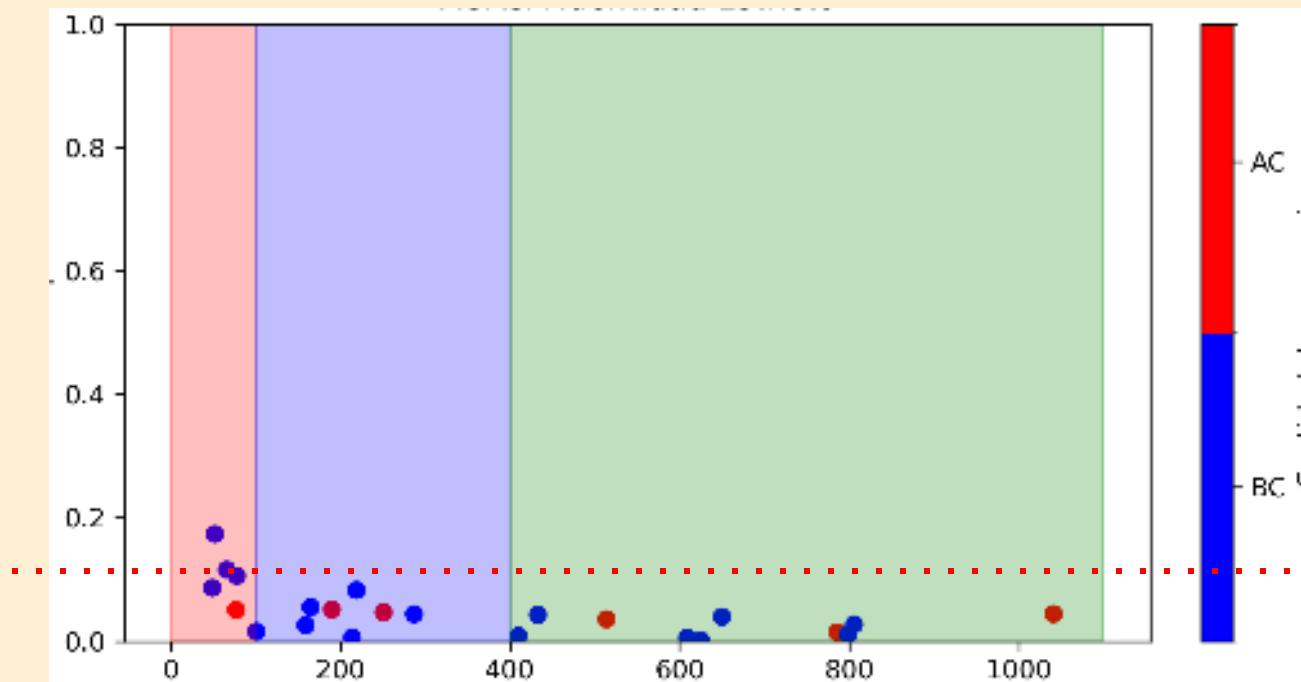
IDENTIDAD



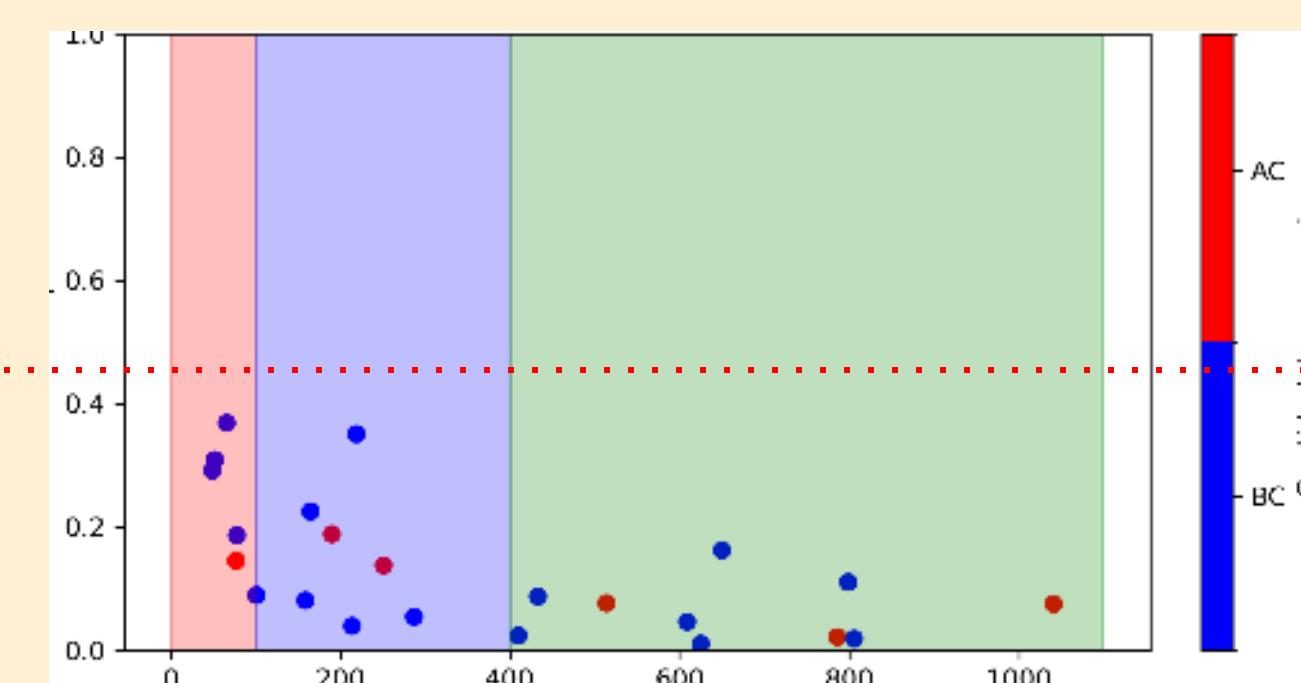
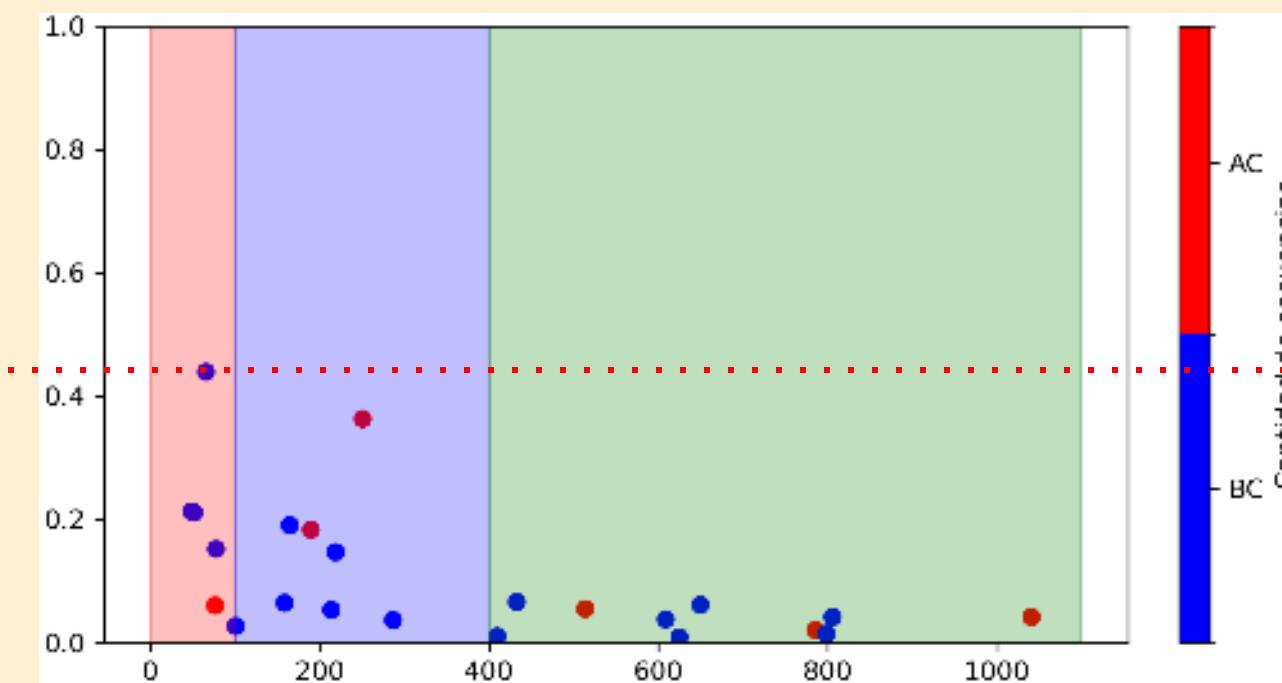
LIBRE



ESTRICTO



COINCIDENCIAS



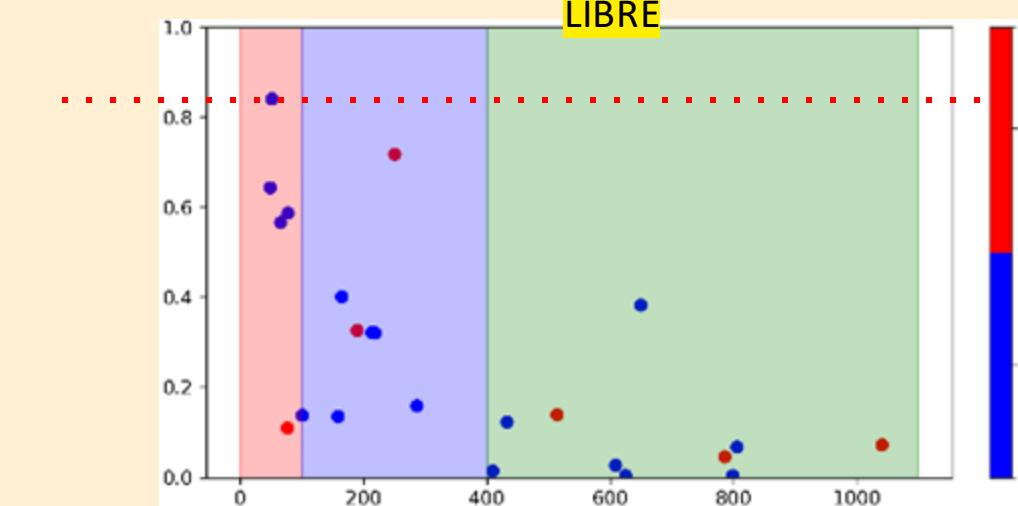
# OS



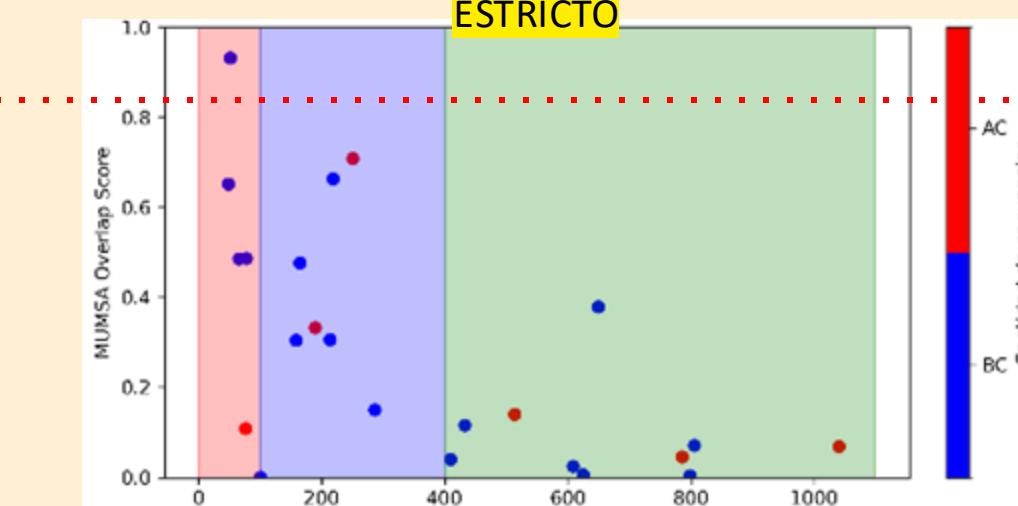
SIMILITUD  
BLOSUM62



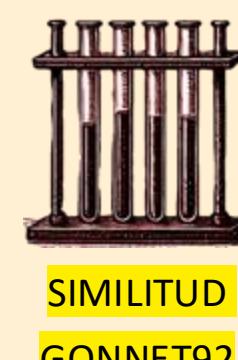
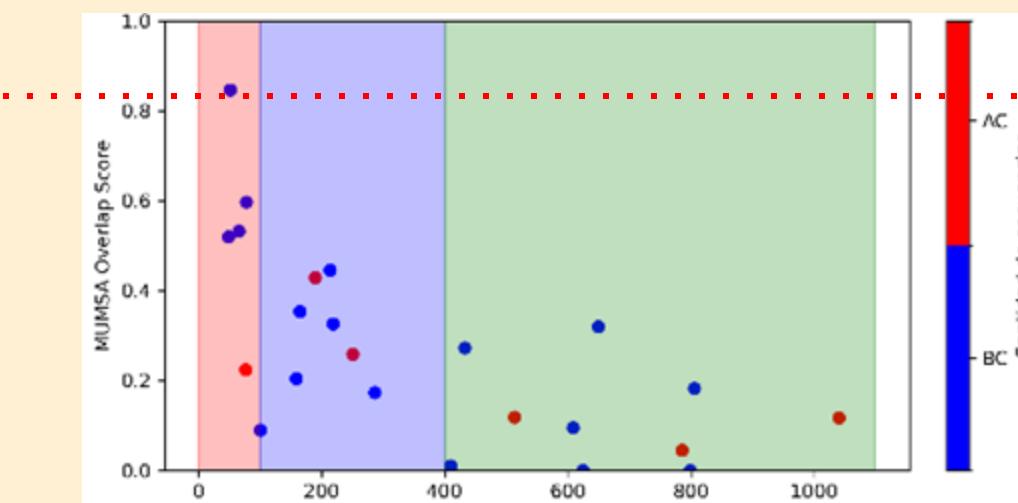
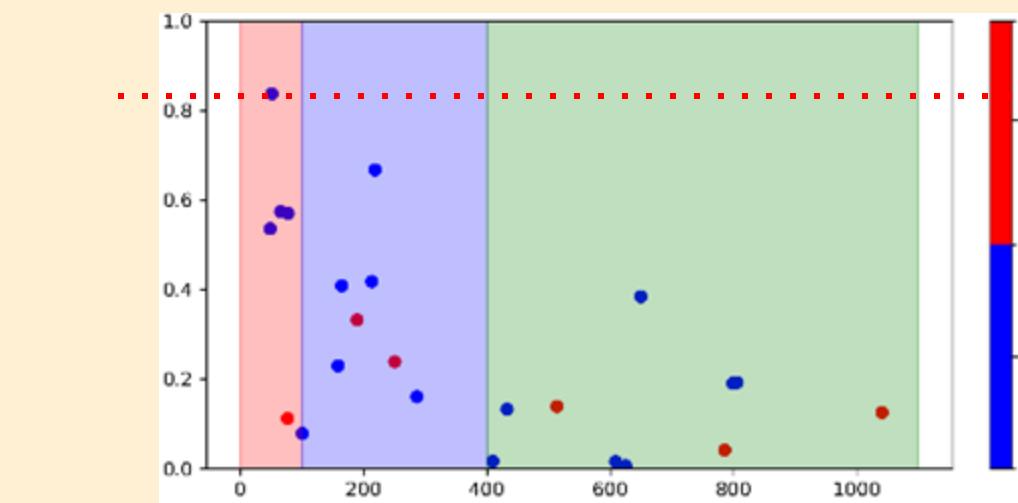
LIBRE



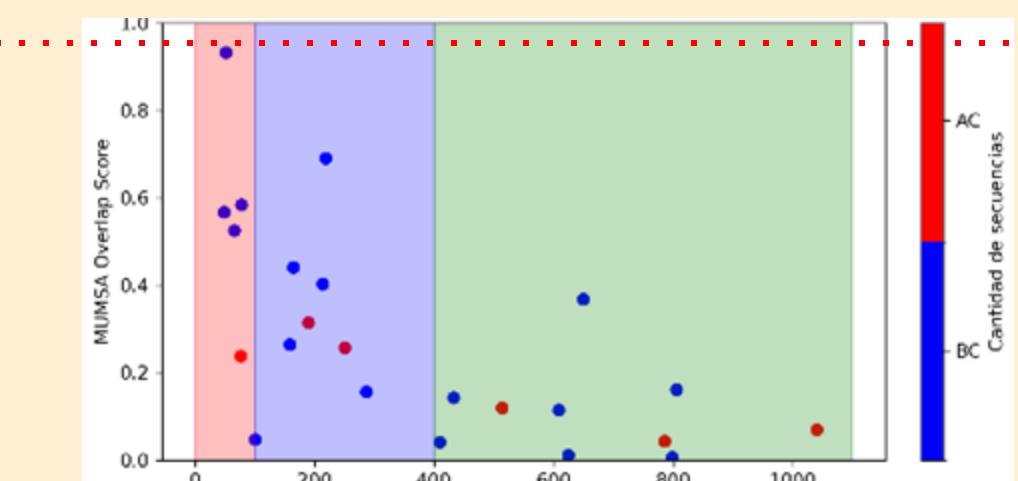
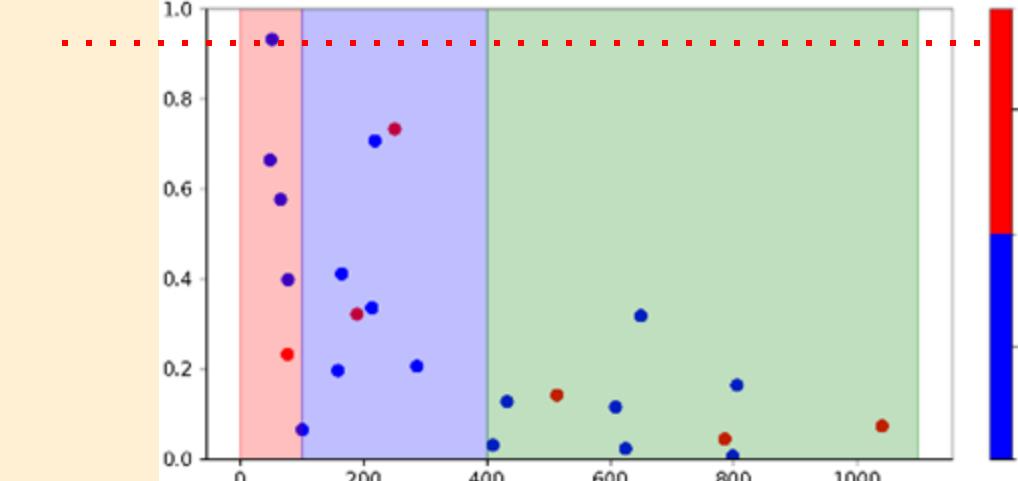
ESTRICTO



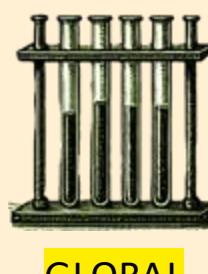
SIMILITUD  
PAM250



SIMILITUD  
GONNET92



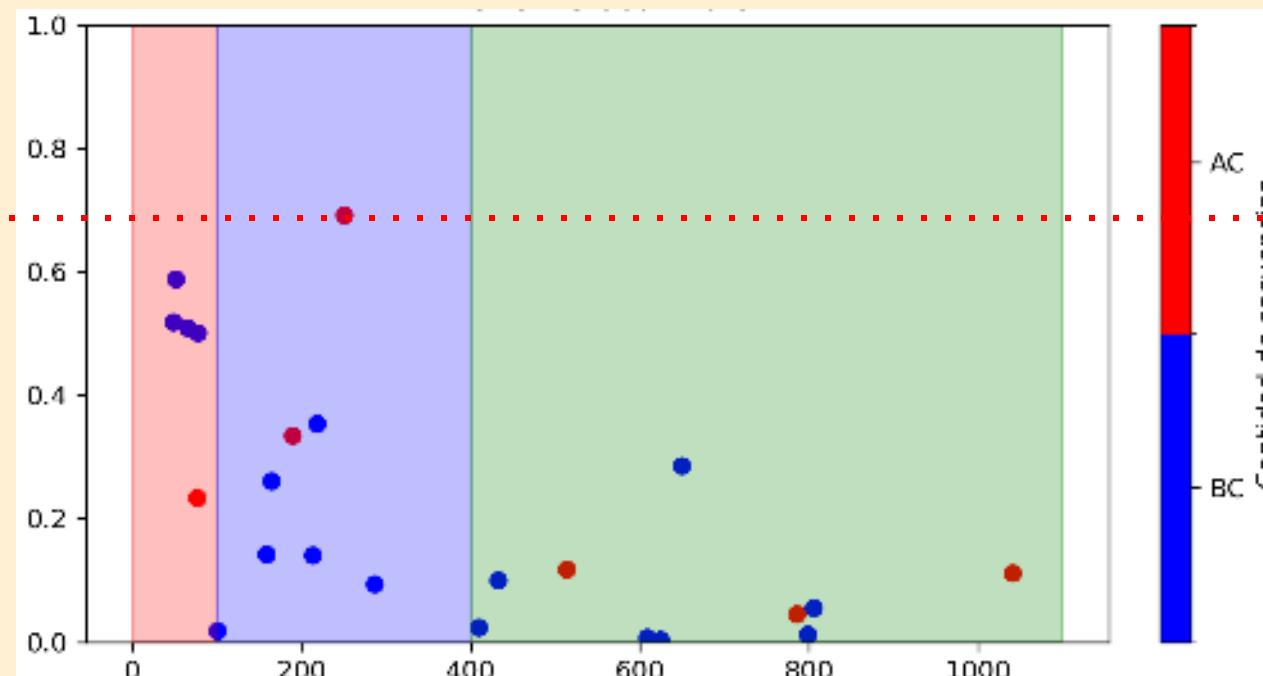
# OS



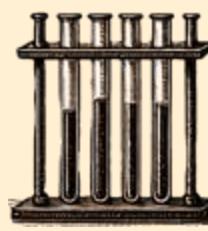
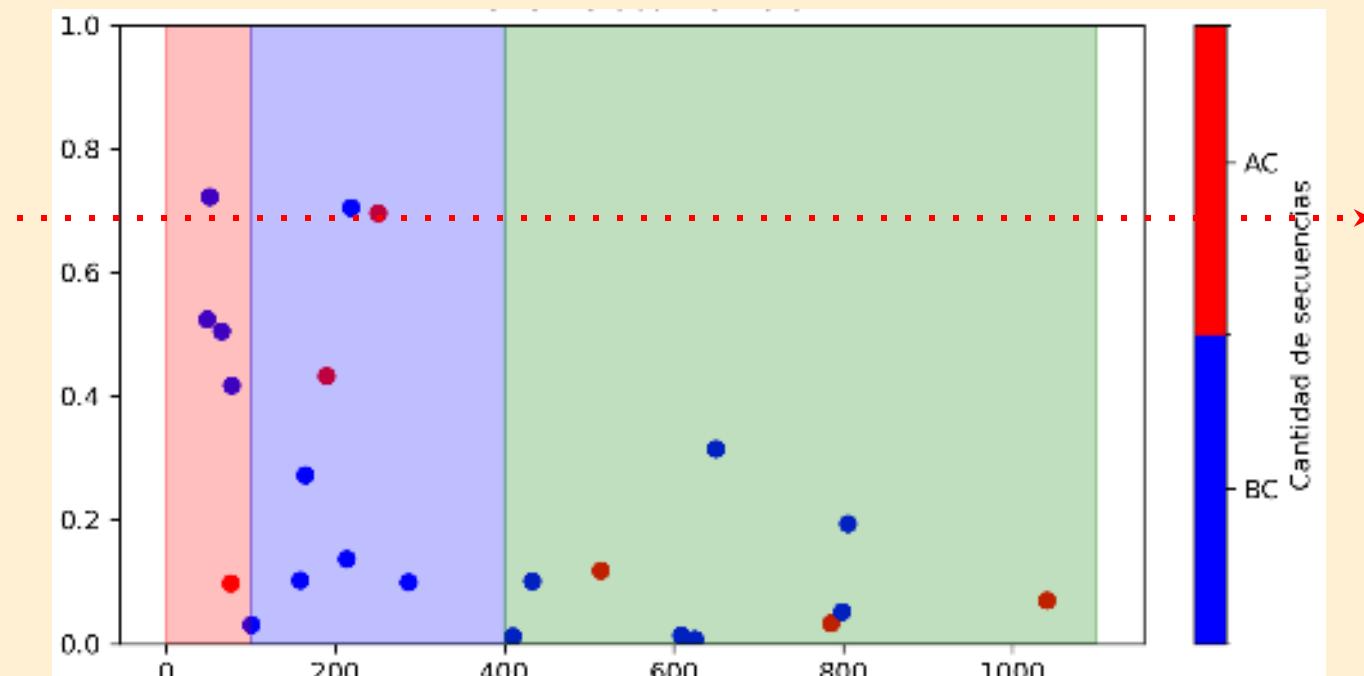
GLOBAL



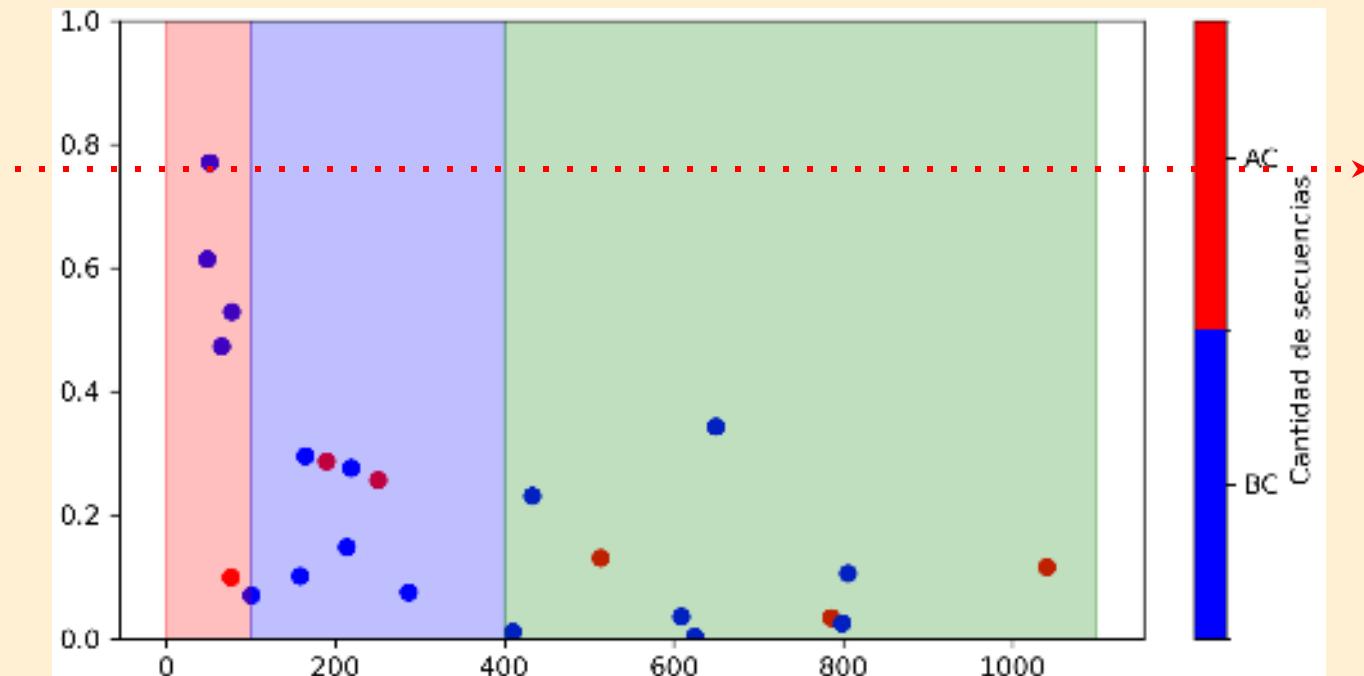
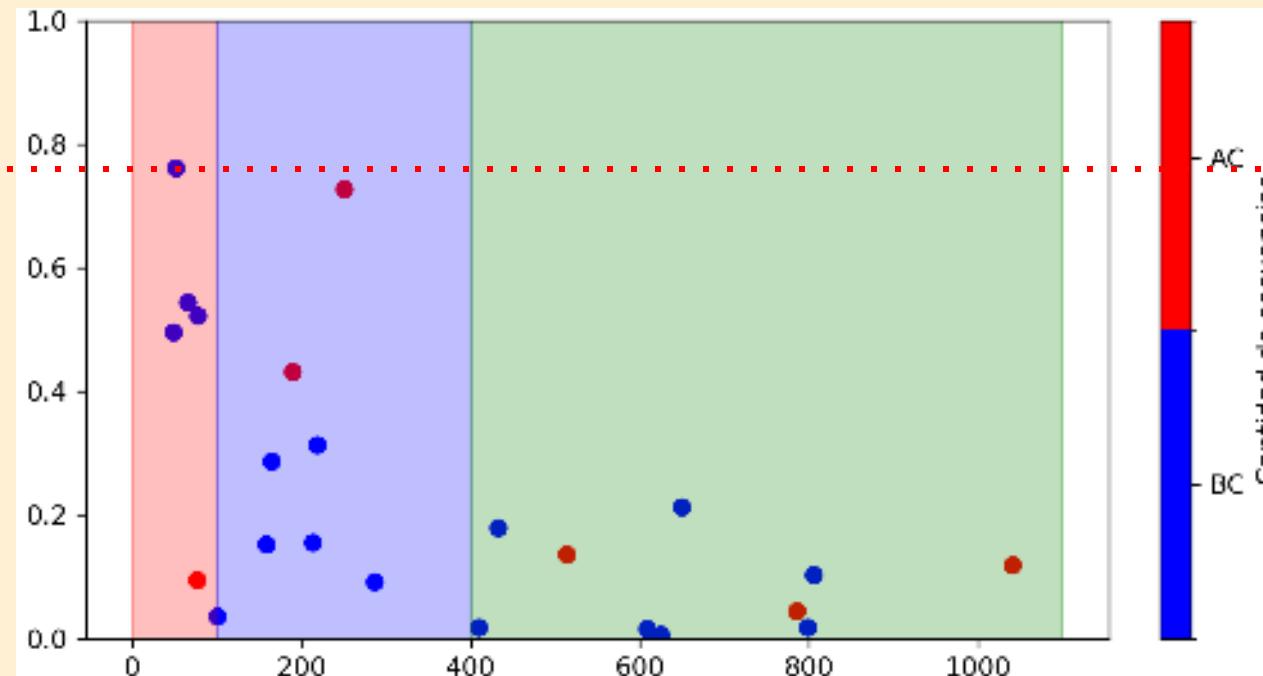
LIBRE



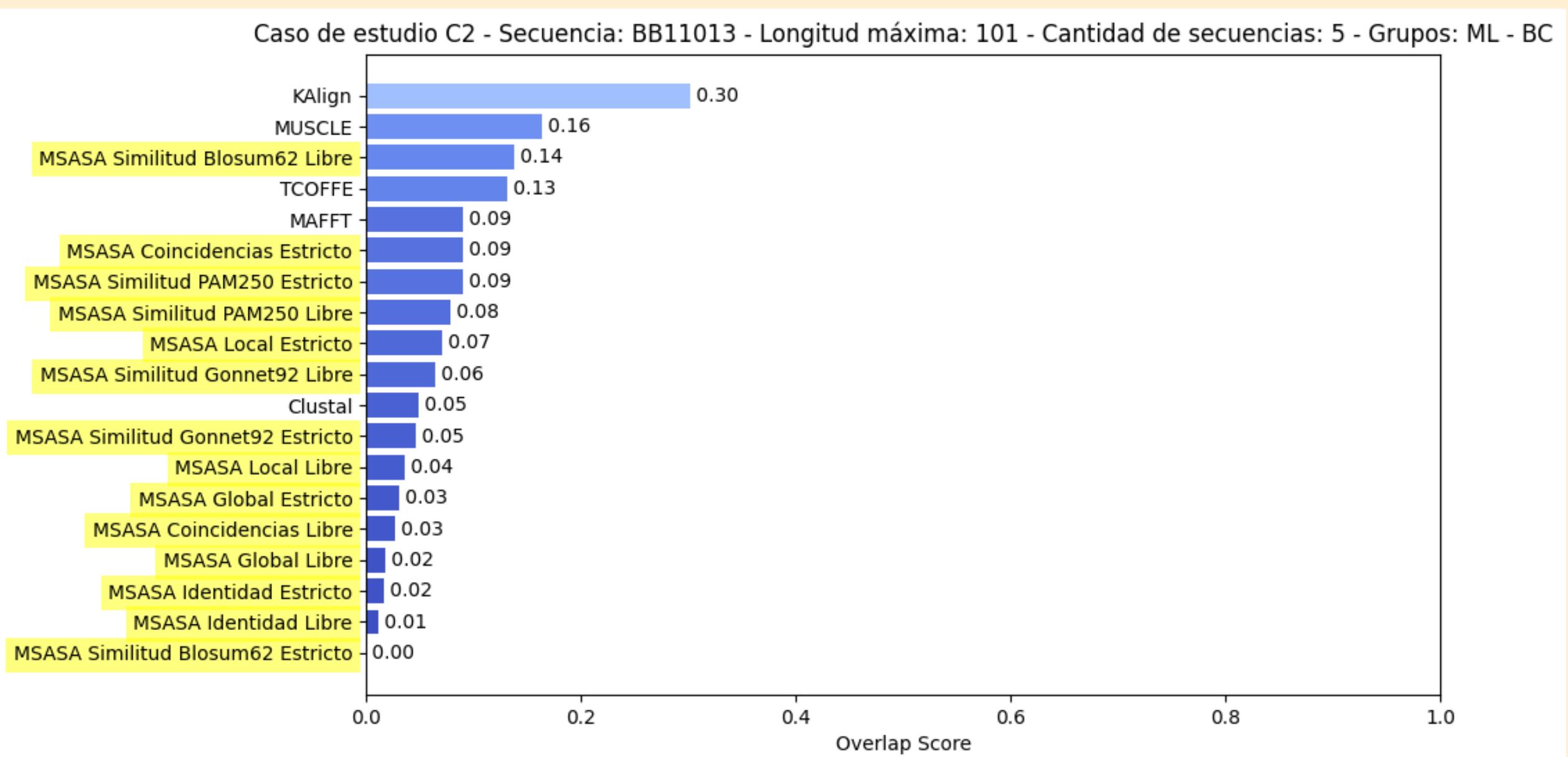
ESTRICTO



LOCAL



# CASO DE ESTUDIO C2



# CASO DE ESTUDIO C14

