PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

MAESTRÍA EN BIOLOGÍA COMPUTACIONAL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MASTER EN BIOLOGÍA COMPUTACIONAL.

TEMA: Desarrollo de un Algoritmo para Clasificar Retrotransposones con LTR en Plantas.

AUTOR: Ing. Tatiana Paola Benalcázar Vayas.

DIRECTOR: PhD. Romain Guyot.

QUITO – ECUADOR.

Julio - 2023.

CERTIFICADO DEL DIRECTOR.

Como director del trabajo de titulación: Desarrollo de un Algoritmo para Clasificar Retrotransposones con LTR en Plantas, desarrollado por Tatiana Paola Benalcázar Vayas, estudiante de la Maestría en Biología Computacional, después de haber supervisado la finalización del trabajo y realizado las correcciones respectivas, apruebo la redacción final del documento escrito para continuar con los trámites correspondientes para sustentar la defensa oral.

_____

PhD. Romain Guyot.

DIRECTOR

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD.

Yo, Tatiana Paola Benalcázar Vayas, portadora de la cédula de ciudadanía número 1720360120, declaro bajo juramento que el presente trabajo de titulación es de mi total autoría, no ha sido presentado previamente para otro grado profesional, no contiene plagio alguno y he consultado las fuentes bibliográficas que respaldan este trabajo.

_____

Tatiana Paola Benalcázar Vayas.

DEDICATORIA.

El presente trabajo lo dedico a mi familia, a mis abuelitos Lic. Arturo Benalcazar y Lic. Rosa Gómez, a mis padres MSc. Luis Benalcazar e Ing. Ayda Vayas, quienes me inculcaron valores de responsabilidad y persistencia, siempre me ayudaron incondicionalmente y estuvieron pendientes de mi desarrollo profesional. También dedico este trabajo a Mitchell Wilcock, quien me brindó su apoyo, confianza y comprensión durante el desarrollo de esta investigación.

En este trabajo se refleja todo mi esfuerzo y dedicación para alcanzar una más de mis metas propuestas, por esta razón dedico este trabajo a mi familia, quienes confiaron en mí, ya que sin su apoyo esto no hubiera sido posible.

AGRADECIMIENTO.

Agradezco a Dios por guiarme para hacer realidad mis sueños, a mi familia en especial a mis abuelitos Lic. Arturo Benalcazar y Lic. Rosa Gómez, quienes me enseñaron a ser constante y a nunca dejarme vencer por las adversidades. Agradezco, a mis padres por toda su ayuda, apoyo y comprensión durante mis estudios de esta maestría, con quienes estaré profundamente agradecida. Adicionalmente, quiero agradecer a Mitchell Wilcock, quien me dio su apoyo y comprensión durante la realización de este proyecto y mis estudios y quien me incentivó a culminar este trabajo También quisiera agradecer a mi director del proyecto de titulación PhD. Romain Guyot, por compartir sus conocimientos y por toda su ayuda para el desarrollo de este trabajo. Mis más sinceros agradecimientos a mis profesores por brindarme sus conocimientos.

ÍNDICE DE CONTENIDO.

RESUMEN

Los elementos transponibles tienen un rol importante en la evolución genética y son los principales componentes de genomas eucariotas, siendo los retrotransposones de larga terminal los más abundantes en genomas de plantas, por esta razón su identificación es un paso crítico para la anotación y el estudio de la regulación de la expresión genética. Se desarrollo una herramienta computacional automatizada, denominada Arthur_LTRanalizer, para identificar y clasificar retrotransposones LTR, en base a sus dominios proteicos, aplicando el método de perfiles de Modelos Ocultos de Markov (HMM). Esta herramienta fue implementada en Python junto con anaconda, cuenta con la función de multiprocesamiento y es capaz de clasificar retrotransposones LTR a nivel del linaje por medio de bases de datos de perfiles de dominios proteicos de elementos transponibles de REXdb y GyDB. Los resultados obtenidos de la búsqueda contra perfiles HMM fueron filtrados y aquellos con mayor puntaje se guardaron en archivos de salida en formato GFF y TSV, el programa retorna un archivo con secuencias anotadas de nucleótidos y aminoácidos, las cuales pueden ser usadas en análisis comparativos subsecuentes y filogenéticos. El rendimiento de la herramienta se comparó con: LTR_retriever, LTRclassifier y TEsorter, usando dos bases de datos curadas de elementos transponibles del arroz y del maíz, con lo que se concluyó que Arthur_LTRanalizer es comparable a herramientas exitosas como LTR_retriever, es confiable, eficiente, exacto, usa un algoritmo de procesamiento rápido y es fácil de usar gracias a su interfaz gráfica.

Palabras clave: Retrotransposones LTR, elementos transponibles, identificación TE, Ty1/Copia, Ty3/Gypsy, pipeline computacional TE, algoritmo, dominios proteicos, REXdb, GyDB, perfiles HMM.

ABSTRACT.

Transposable elements have an important role in the genetic evolving and are the main components of eukaryotic genomes, being long terminal repeat retrotransposons the most abundant in plant genomes, the identification of this elements is a critical step for the annotation and study of the regulation of the genetic expression. A new automatized computational tool was developed called Arthur_LTRanalizer for identifying and classifying LTR retrotransposons taking in count its protein domains and applying the method of profiles of Hidden Markov Models. This tool was implemented in Python together with anaconda, it has the function of multiprocessing and is capable of classifying LTR retrotransposons to the linage level using data bases of protein domains profiles of transponsable elements from REXdb y GyDB. The results of the search against HMM profiles were filtered and were kept those ones that had greater score and saved in output files in GFF and TSV format, the program produce a file of annotated sequences of nucleotides and amino acids which can be used in phylogenic and posterior comparative analysis. The performance of the tool was compared with: LTR_retriever, LTRclassifier and TEsorter by using two curated data bases of transposable elements from sequences of rice and corn, it was concluded that Arthur_LTRanalizer is comparable to successful tools like LTR_retriever, is reliable, efficient, exact, use an algorithm of fast processing and is easy to use thanks to the graphical interface.

Keywords: LTR-retrotransposons, transposable elements, TE identification, Ty1/Copia, Ty3/Gypsy, TE computational pipeline, algorithm, protein domains, REXdb, GyDB, HMM profiles.

CAPÍTULO I.
INTRODUCCIÓN.

1.1. ANTECEDENTES.

Existen diferentes estrategias y herramientas para anotar los elementos transponibles, estos elementos pueden ser identificados por su homología (con RepeatMasker o LTRClassifier), de *novo* (con RepeatModeler o REPET) o por su estructura (con LTR_finder o LTR_harvest). Algunas herramientas pueden cobrar una cierta variedad de transposones, mientras que otras herramientas se enfocan solamente en ciertas clases de transposones (Riehl, Riccio, Miska, & Hemberg, 2022).

Los investigadores de la Universidad Autónoma de Manizales de Colombia elaboraron un algoritmo conocido como Inpactor (https://github.com/simonorozcoarias/Inpactor) para analizar y clasificar una clase particular de elementos: Los retrotransposones con LTR (Long Terminal Retrotransposon), en base a la homología y estructura, lo aplicaron en plantas de piña, en esta investigación se usó el lenguaje de programación bash y C (Orozco Arias et al., 2018).

También, en otra investigación realizada por la Universidad Autónoma de Manizales de Colombia se estudiaron las medidas de desempeño de diversos algoritmos computacionales para determinar y clasificar a los retrotransposones con LTR, utilizando enfoques de aprendizaje automático y aprendizaje profundo (Orozco-Arias et al., 2023), esta investigación fue dirigida por el Doctor Romain Guyot (Orozco-Arias, Isaza, & Guyot, 2019).

1.2. PLANTEAMIENTO DEL PROBLEMA.

Las características que presentan los elementos transponibles como: el gran número de copias en los genomas, una evolución más rápida que los genes codificantes, una gran variedad de fragmentos no autónomos, su evolución dinámica debido a la inserción anidada de otros elementos transponibles, una recombinación desigual y las duplicaciones en tándem, los convierte en elementos difíciles de detectar y clasificar de

forma precisa y rápida. Adicionalmente, la clasificación de los elementos transponibles está estructurada jerárquicamente desde la familia más alta hasta los linajes, por lo que la clasificación a nivel de familias es una tarea compleja, requiere de demasiado tiempo y en algunos casos incluso se necesita realizar una curación manual de secuencias biológicas (Wicker et al., 2007). En las plantas, los elementos retrotransposones LTR, de estructura similar a los retrovirus animales, son los más redundantes, su multiplicación contribuye al aumento del tamaño del genoma. Por ejemplo, el genoma del trigo blando tiene un tamaño de más de 15Gb (5 veces el genoma humano) y contiene casi un 90 % de elementos transponibles (Zhu et al., 2021). Estos elementos pueden clasificarse a nivel de familia/linaje utilizando enfoques de similitud con dominios proteicos de referencia (Neumann, Novák, Hoštáková, & Macas, 2019). De esta forma el desarrollo de un nuevo algoritmo computacional contribuiría a facilitar la tarea de anotación y clasificación de los elementos transponibles de diferentes organismos de una forma más eficaz y precisa (Wicker et al., 2007).

1.3. JUSTIFICACIÓN.

Es importante identificar y clasificar los elementos transponibles en las plantas porque contribuyen a la diversidad genética de los genomas y pueden crear importantes mutaciones en las plantas de cultivo. La actividad de estos elementos tuvo un importante impacto positivo en rasgos de interés agronómico para las especies de cultivo, como: la pigmentación de la piel de la fruta en las manzanas (Zhang et al., 2019) y las uvas (Ferreira et al., 2019), la pigmentación de la pulpa en las naranjas (Huang et al., 2019), el desarrollo de la fruta partenocarpia sin semilla en las manzanas (Joldersma & Liu, 2018), el fenotipo de nectarina en los melocotones (Lu et al., 2021) y la diversidad de las accesiones de tomate, incluyendo la variación en la forma y el color de la fruta (Domínguez et al., 2020). Sin embargo, los elementos transponibles también pueden tener un impacto negativo en los rasgos agronómicos, como lo ilustra la mutación del manto en las palmas aceiteras clonalmente (Vetaryan, Kwan, Namasivayam, Ho, & Syed Alwee, 2018).

A partir de los estudios realizados para determinar el funcionamiento de los elementos transponibles, se han podido desarrollar diversas herramientas biotecnológicas como las de genotipado por ejemplo TIP_finder (Orozco Arias et al., 2020) o las de inducción de nuevas mutaciones en las plantas.

1.4. FORMULACIÓN DE UNA HIPÓTESIS.

El desarrollo de un algoritmo computacional permite identificar y clasificar a los retrotransposones LTR al nivel de linaje de secuencias genómicas de plantas.

1.5. OBJETIVOS DE LA INVESTIGACIÓN.

1.5.1. Objetivo general.

- Diseñar un algoritmo computacional rápido que permita identificar y clasificar los retrotransposones con LTR en las plantas usando dominios proteicos.

1.5.2. Objetivos específicos.

- Realizar una interfaz gráfica de usuario que permita escoger los parámetros de entrada para el análisis de las secuencias biológicas.
- Elaborar las instrucciones de programación para el módulo de identificación de los retrotransposones LTR empleando el lenguaje Python.
- Desarrollar el software para el módulo de clasificación de los retrotransposones LTR usando dominios proteicos de la base de referencia REXdb y GyDB.
- Evaluar y comparar el desempeño del nuevo algoritmo elaborado usando medidas de rendimiento.

1.6. ALCANCE.

El alcance del presente trabajo de titulación consiste en la implementación de un algoritmo computacional que permita clasificar retrotransposones LTR, tomando en cuenta los dominios proteicos de la base de datos de referencia REXdb y GyDB. El

algoritmo será elaborado en el lenguaje de programación de Python, utilizando la metodología de ensayo y error. En este proyecto no se realizará secuenciación de genomas, ya que las secuencias que serán analizadas provienen de especies de plantas que serán obtenidas de bases de datos de plantas disponibles en la web como son: la base de datos publica de Ensembl (IRGSP, 2022), RepetDB (Amselem et al., 2019), InpactorDB (Orozco-Arias et al., 2021) y Oryza Repeat Database (Ouyang & Buell, 2022). Como resultado de este proyecto se pretende obtener un algoritmo eficiente que sea amigable con el usuario y que contribuya al desarrollo de la biología computacional de Ecuador.

CAPÍTULO II.

MARCO TEÓRICO.

2.1. Historia de los elementos transponibles.

Los elementos transponibles son el principal contribuyente al tamaño de genomas de plantas, por un largo tiempo los elementos transponibles tuvieron una connotación negativa ya que primariamente estaban considerados como ADN basura. En años recientes, este punto de vista ha comenzado a cambiar, aunque los elementos transponibles pueden dañar genes presentes en el genoma anfitrión (Thieme & Bucher, 2018), estos elementos presentan algunas funciones importantes en las plantas como: están involucrados en procesos adaptativos y evolutivos, constituyen el principal factor que afecta el tamaño de los genomas de plantas, bajo condiciones de estrés reordenan un genoma, pueden relocalizar genes y generar nuevos genes y pseudo genes, contribuyen a la composición centromérica, son capaces de regular la expresión de genes cercanos mediante diversos mecanismos como: proveer elementos regulatorios promotores y potenciadores e insertarse dentro de los genes para dirigir el sistema regulatorio epigenético. Los transposones se han utilizado en la clonación de genes de plantas, tienen el potencial de mejorar la productividad de cosechas y presentan un rol importante en la domesticación de cosechas (Thieme & Bucher, 2018; Valencia & Girgis, 2019).

La codificación de la información genética para un cierto fenotipo se había asumido que estaba organizada de una forma unidimensional y estática, hasta que se presentó el descubrimiento revolucionario de los elementos transponibles por Barbara McClintock en 1950, con la observación de sitios (loci) mutables que sustentaban la alta diversidad del color del grano del maíz. En la actualidad se conoce que la presencia de elementos genéticos móviles como el sistema Ac/Ds (Transposones) descubierto por Barbara McClintock constituye la regla y no la excepción. En efecto, los elementos transponibles han sido detectados en todos los organismos probados y en casos extremos constituyen más del 80 % del genoma del maíz y la cebada (Thieme & Bucher, 2018).

El aumento del conocimiento en el campo de la investigación de los elementos transponibles ha revelado su función indispensable durante el desarrollo, en la respuesta a factores de estrés ambiental y como controladores de la evolución. También, se han encontrado hallazgos prominentes en diferentes reinos incluyendo a los humanos, en los que los elementos transponibles son considerados como módulos básicos vitales de la vida (Chuong, Elde, & Feschotte, 2017; Thieme & Bucher, 2018).

Actualmente, los cultivadores del sector orgánico tienen grandes retos para desarrollar nuevas variedades de cultivo, para construir las bases para la seguridad de la comida de una población global en crecimiento bajo condiciones ambientales cambiantes. Los elementos transponibles junto con los estudios epigenéticos constituyen una fuente genética valiosa para los retos futuros en agricultura, con los conocimientos de los mecanismos de su regulación (Galindo-González, Mhiri, Deyholos, & Grandbastien, 2017).

2.2. Elementos transponibles en genomas de plantas.

2.2.1. Clasificación.

Los elementos transponibles (TE) son secuencias de ADN que tienen la habilidad de cambiar su posición dentro del genoma y tienen la capacidad de regular la función de un gen y la evolución del genoma. Los elementos transponibles se dividen en dos clases principales, tomando en cuenta su mecanismo de transposición, como: elementos de clase I y elementos de clase II (transposones) y cada clase puede ser subdividida en subclases en base al mecanismo de integración cromosómica (Bourque et al., 2018; Ranganathan, Nakai, Schönbach, & Gribskov, 2019; Thieme & Bucher, 2018).

Los elementos de clase I están relacionados evolutivamente con los retrovirus y se conocen como retrotransposones, estos elementos dependen completamente de la transcripción por la ARN polimerasa II y se amplifican por medio de ARN intermediario, se pueden movilizar por medio del mecanismo copia y pega; para lo cual primeramente el DNA es copiado a una molécula de ARN, este intermediario de RNA es transcrito en reversa usando la enzima reverso-transcriptasa, a una copia de cDNA, la secuencia de

ADN transcrita en reversa es insertada o integrada en una nueva posición dentro del genoma (Bourque et al., 2018; Ranganathan et al., 2019; Thieme & Bucher, 2018).

El sistema de convención de clasificación de elementos transponibles es jerárquico y se basa en una codificación de tres letras, en la que cada letra denota: la clase, el orden y superfamilia, el nombre de la familia o subfamilia, la secuencia en la que cada elemento fue encontrado y el número de corrida que define la inserción individual en la accesión. La clase representa el nivel más alto que divide a los elementos transponibles en función de la presencia o ausencia de ARN como un intermediario en la transposición. Cada clase de elementos transponibles puede ser dividida en subclases, órdenes y superfamilias en función de las características que comparten; las subclases permiten distinguir a los elementos transponibles de acuerdo a su movimiento durante la transcripción reversa, los órdenes organizan a los elementos transponibles conforme al mecanismo de inserción, las superfamilias agrupan a los elementos transponibles en base a los dominios proteicos, presencia y tamaño del sitio de duplicación (TSD), las superfamilias pueden ser divididas en familias tomando en cuenta la conservación de secuencias de ADN (Cho, 2021; Wicker et al., 2007).

Los elementos de clase I o retrotransposones no son clasificados en subclases ya que todos los elementos son copia-inserción y ninguno de los elementos transfieren hebras de ADN al sitio donante y en lugar de esto el ARN intermediario es transcrito desde una copia genómica y después es transcrito en reversa a ADN por medio de la reverso-transcriptasa (RT), por esta razón el orden reemplaza a la subclase de los retrotransposones, dividiéndose en cinco ordenes de acuerdo a la organización y filogenia de la reverso-transcriptasa, estos ordenes son: retrotransposones de repetición terminal larga (LTR o Long Terminal Repeats), secuencias repetitivas intermedias dictyostelium (DIRS), elementos de tipo Penelope (PLE), elementos nucleares largos intercalados (LINE o Long Interleaved Nuclear Elements) y elementos nucleares cortos intercalados (SINE o Short Interleaved Nuclear Elements); estos elementos se agrupan básicamente como retrotransposones LTR y no LTR, los miembros de los cuatro últimos ordenes se conocen como elementos no LTR (Cho, 2021; Wicker et al., 2007).

Los elementos de la clase II también se conocen como ADN transposones, se transponen sin producir un ARN intermediario y siguen el mecanismo corta y pega, (Bourque et al., 2018; Ranganathan et al., 2019; Thieme & Bucher, 2018). Las principales clases de elementos transponibles pueden subdividirse en base a sus características estructurales y enzimáticas (Thieme & Bucher, 2018). Los elementos de la clase II se subdividen en dos subclases: subclase 1 que constituye a los transposones de repetición de terminal invertida (TIR), que a su vez se clasifica en 10 superfamilias en función de las similitudes de secuencia: Tc1-Mariner, hAT, Mutator, Merlin, Transib, P, PiggyBac, Pif-Harbinger, CACTA y Crypton; a su vez la subclase 2 se conoce como transposones de ADN no TIR que está compuesta por dos superfamilias: Helitron y Maverick (Cho, 2021).

El principal criterio para diferenciar a los retrotransposones es la presencia o ausencia de dos repeticiones terminales largas (LTRs) con la misma orientación en la terminación 5' y 3', de esta forma los retrotransposones pueden ser de dos tipos: de repetición terminal larga (LTR) y de repetición sin terminal larga (no-LTR); en los retrotransposones de repetición terminal larga (LTR) la integración se produce por métodos de división y de una reacción de transferencia de hebra catalizada por una integrasa (Bourque et al., 2018; Ranganathan et al., 2019; Thieme & Bucher, 2018).

Tabla 1. Clasificación jerárquica de los retrotransposones (Cho, 2021).

Clase	Orden	Superfamilia	Organismo
Clase I (retrotransposones)	LTR	Copia	Plantas, metazoarios, fungi
		Gypsy	Plantas, metazoarios, fungi
		Bel-Pao	Metazoarios
		Retrovirus	Metazoarios
		ERV	Metazoarios
	DIRS	DIRS	Plantas, metazoarios, fungi
		Ngaro	Plantas, metazoarios
		VIPER	Otro
	PLE	Penelope	Plantas, metazoarios, fungi

	LINE	R2	Metazoarios
		RTE	Metazoarios
		Jockey	Metazoarios
		L1	Plantas, metazoarios, fungi
		I	Plantas, metazoarios, fungi
	SINE	tRNA	Plantas, metazoarios, fungi
		7SL	Plantas, metazoarios, fungi
		5S	Plantas
Clase II (ADN transposones) Subclase I	TIR	Tc1- Mariner	Plantas, metazoarios, fungi
		hAT	Plantas, metazoarios, fungi
		Mutator	Plantas, metazoarios, fungi
		Merlin	Metazoarios
		Transib	Metazoarios, fungi
		P	Plantas, metazoarios
		PiggyBac	Metazoarios
		PIF - Harbinger	Plantas, metazoarios, fungi
		CACTA	Plantas, metazoarios, fungi
	Crypton	Crypton	Fungi
Clase II (ADN transposones) Subclase II	Helitron	Helitron	Plantas, metazoarios, fungi
	Maverick	Maverick	Metazoarios, fungi

Los retrotransposones LTR son particularmente más abundantes en genomas de plantas, están formados por dos secuencias de repeticiones terminales largas que contienen promotores y secuencias regulatorias que son requeridas para la actividad de transposición y que ayudan a controlar su expresión y replicación, están localizadas en los extremos de la región interna en la que se ubican los genes *gag* y *pol* (Cho, 2021).

Los elementos transponibles pueden ser divididos en familias o subfamilias, las cuales son grupos cercanamente relacionados que pueden ser descendientes de una unidad ancestral que está presente como una copia ancestral y se puede inferir como una secuencia consenso que es representativa de la familia o subfamilia (Bourque et al.,

2018). En función del orden de los genes o de los dominios proteicos y de su secuencia, los retrotransposones LTR se han agrupado en tres superfamilias como son: Ty1/Copia (pseudoviridae), Ty3/Gypsy (metaviridae) y endógenos retrovirus (ERV) (Ramakrishnan et al., 2022; Sultana et al., 2022; Thieme & Bucher, 2018). La principal diferencia estructural entre los grupos Copia y Gypsy se basa en el orden de la reverso-transcriptasa (RT) e integrasa (INT), por ejemplo en Ty1/Copia el orden de los genes es PR-INT-RT-RH y en Ty3/Gypsy es PR-RT-RH-INT (Aroh & Halanych, 2021; Z. Ouyang et al., 2021). Los estudios filogenéticos demostraron que la similitud en la estructura genética entre Ty3/Gypsy y retrovirus, indican que los retrovirus se pudieron haber originado de elementos Ty3/Gypsy con la adición de un nuevo gen *env* (Ramakrishnan et al., 2022).

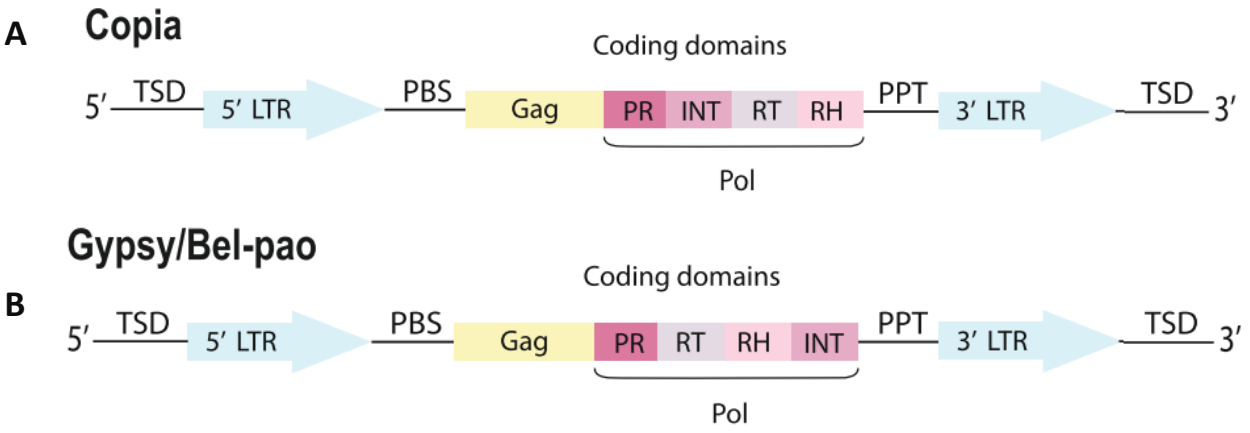

Figura 1. Representación esquemática de los retrotransposones LTR. A. Modelo del elemento Ty1/Copia. B. Modelo del elemento Ty3/Gypsy (Aroh & Halanych, 2021).

Las superfamilias se dividen en linajes dependiendo de la estructura de los elementos y la secuencia (Vangelisti et al., 2020). Recientemente Neumann et al. realizaron una investigación a 13 863 retrotransposones LTR de 80 especies de plantas y establecieron un sistema de clasificación perfeccionado para los retrotransposones LTR en plantas, dividieron a los elementos Ty1/Copia y Ty3/Gypsy en 16 y 14 linajes respectivamente (Jedlicka, Lexa, Vanat, Hobza, & Kejnovsky, 2019; Neumann et al., 2019). Los retrotransposones Ty1/Copia se subdividen en los linajes: Ale, Alesia, Angela, Bianca, Bryco, Lyco, Gymco-I, Gymco-II, Gymco-III, Gymco-IV, Ikeros, Ivana (Sirevirus/Oryco),

Osser (hemivirus), SIRE, TAR, Tork. Los linajes de Ty3/Gypsy que se agrupan en base a la presencia de un cromodominio en la rama de cromovirus incluyen: Chlamyvir, Tcn1, CRM, Galadriel, Tekay (Del/Del1), Reina y los elementos que no pertenecen a la rama de cromovirus son: Phygy, Selgy, Athila, Tat, Ogre, Retand (Neumann et al., 2019; Zhou et al., 2021).

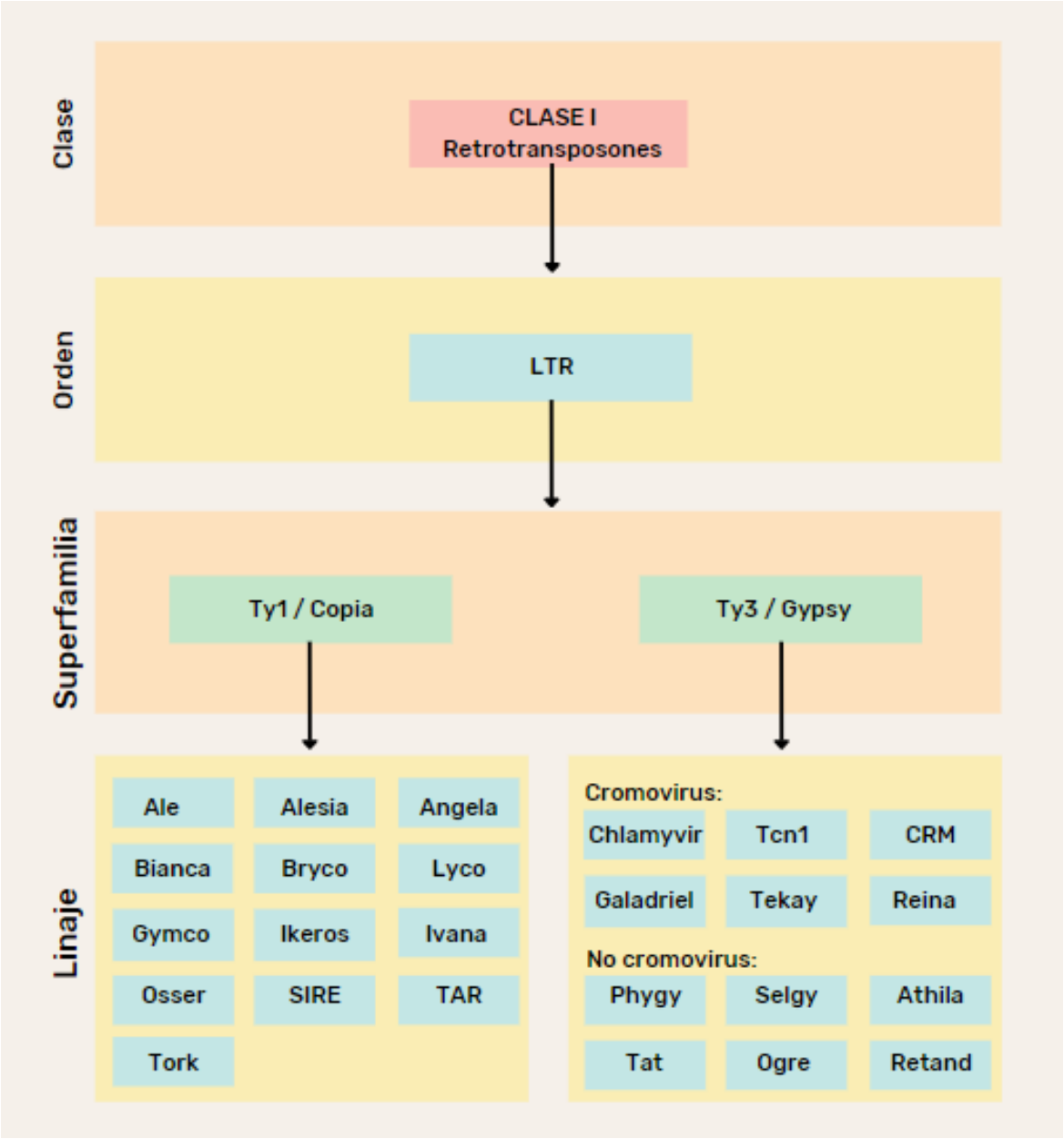

Figura 2. Clasificación de elementos transponibles LTR, adaptado de (Bourque et al., 2018; Neumann et al., 2019; Orozco-Arias et al., 2019; Zhou et al., 2021).

Tabla 2. Correspondencias entre nombres de superfamilias y linajes de sistemas de clasificación y del Comité Internacional de Taxonomía de Virus (ICTV), adaptado de (Orozco-Arias et al., 2019).

REXdb	Wicker and Keller	GyDB	ICTV
Superfamilias			
Copia	Copia	Ty1/Copia	Pseudoviridae
Gypsy	Gypsy	Ty3/Gypsy	Metaviridae
Bel-pao	Bel-pao	Bel-pao	Semotivirus
Linajes — Copia			
Ale	Ale	Sirevirus/Retrofit	Pseudovirus
Alesia	Ale	-	-
Angela	Angela	-	Pseudovirus
Bianca	Bianca	-	-
Bryco	-	-	-
Lyco	-	-	-
Gymco I, II, III, IV	-	-	-
Ikeros	Angela	Tork	Pseudovirus
Ivana	Ivana	Sirevirus/Oryco	-
Osser	-	Osser	Hemivirus
SIRE	Maximus	Sirevirus/SIRE	Sirevirus
TAR	TAR	Tork	-
Tork	-	Tork	Pseudovirus
Linajes — Gypsy			
Chromovirus/CRM	-	Chromoviruses/CRM	-
Chromovirus/Chlamyvir	-	-	-
Chromovirus/Galadriel	-	Chromoviruses/Galadriel	-
Chromovirus/Reina	-	Chromoviruses/Reina	-

Chromovirus/Tekay	-	Chromoviruses/Del	Metavirus (Del1)
Non- Chromovirus/OTA/Athila	-	Athila/Tat/Athila	Metavirus (Athila)
Non- Chromovirus/OTA/Tat/TatI	-	-	-
Non- Chromovirus/OTA/Tat/TatII	-	-	-
Non- Chromovirus/OTA/Tat/TatIII	-	-	-
Non/ Chromovirus/OTA/Tat/Ogre	-	Athila/Tat/Tat/Tat (Ogre)	-
Non/ Chromovirus/OTA/Tat/Retand	-	Athila/Tat/Tat	Metavirus (Tat4)
Non- Chromovirus/Phygy	-	-	-
Non- Chromovirus/Selgy	-	-	-

2.2.2. Características de los retrotransposones LTR.

Los retrotransposones LTR en plantas tienen una longitud desde unos pocos cientos de pares de bases (menos de 1 kb) hasta kilo bases (22 kb), la secuencia nucleotídica inicia con los nucleótidos 5'-TG-3' y termina con 5'-CA-3', además los genes de polimerasa (*pol*) y *gag* están organizados en un marco abierto de lectura (ORF o "Open Reading Frame") (Garcia-Pérez, 2016) y terminan con una secuencia de longitud larga en 3', sin embargo algunos marcos abiertos de lectura pueden estar localizados entre *pol* y la secuencia de terminación larga. El gen *gag* está implicado en la codificación de proteínas estructurales para la transcripción reversa y replicación (Ou & Jiang, 2018), el gen *pol* generalmente codifica varios dominios proteicos como son: proteasa (PR), integrasa (IN), reverso transcriptasa (RT) y RNase H o ribonucleasa H (RH) (Neumann et al., 2019; Ramakrishnan et al., 2022). La reverso-transcriptasa (RT) es la única enzima capaz de catalizar la transcripción reversa y constituye el principal dominio enzimático que es común a todos los retrotransposones autónomos (Gabriel, 2023). Adicionalmente, existen dos motivos conservados que son característicos de estos elementos como son: el sitio de combinación de primers (PBS) y la vía polipurina (PPT, región altamente enriquecida en purinas A y G), que están involucrados en la replicación de los retrotransposones (Sultana et al., 2022; Vangelisti et al., 2020).

Tabla 3. Dominios de los elementos transponibles y sus funciones en el mecanismo de replicación, adaptado de (Orozco-Arias et al., 2019).

Nombre completo del gen	Nombre corto	Función
Reverso transcriptasa	RT	Síntesis de DNA usando RNA como templado.
RNase H	RNAseH	Degradación del templado de RNA en el hibrido DNA-RNA.
Integrasa	INT	Catalización de la inserción del retrotransposón cDNA en el genoma de la célula anfitriona.
Envelope	ENV	Transferencia célula - célula del retrovirus.
Grupo específico de antígeno	GAG	Proteína estructural para partículas similares a virus.
Cromodominio	Chrod	Dirección de la inserción de nuevas copias de retrotransposones LTR en regiones heterocromáticas, por el reconocimiento especifico de marcas de histonas heterocromáticas y otros factores.

Las regiones LTRs pueden tener una longitud de 85 bp hasta 5 kb conteniendo regiones regulatorias y sitios de inicio de transcripción (TSS) necesarios para la transcripción de los elementos transponibles por ARN polimerasa II. Los transcriptos originarios de la región 5' LTR tienen dos funciones importantes en el ciclo de vida de los retrotransposones, la primera función es la codificación para la maquinaria de replicación o poliproteínas que consisten de: proteinasa aspártica (AP), reverso transcriptasa (AR), ribonucleasa RNaseH (RH), integrasa (INT) y proteínas estructurales GAG de cápside que forman una partícula similar a virus (Virus Like Particule), la segunda función es que sirven como templado para la transcripción reversa que resulta en ADN extracromosomal complementario (ecDNA) que es capaz de ingresar al núcleo e integrarse en el ADN genómico (Thieme & Bucher, 2018).

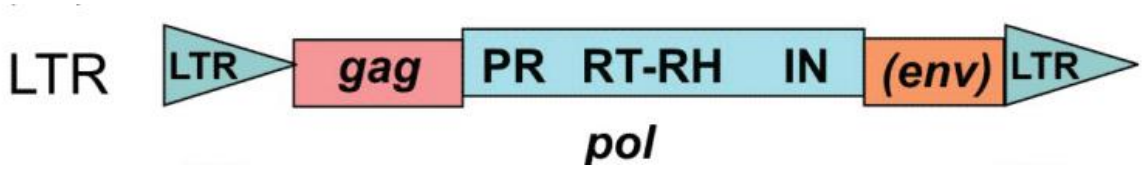

Figura 3. Características estructurales de retrotransposones LTR (Gabriel, 2023).

2.2.3. Mecanismo de transposición.

La transposición es una característica de los transposones y se basa en la movilidad, los transposones pueden ser categorizados como elementos autónomos y no autónomos. Los elementos autónomos tienen marcos abiertos de lecturas (ORF) que codifican los productos requeridos para la transposición, es decir están equipados con la maquinaria necesaria para moverse alrededor del genoma; mientras que los elementos no autónomos que están habilitados para la transposición carecen de la mayoría o de todas las secuencias codificantes, no tienen capacidad de codificación significativa pero mantienen secuencias cis que son necesarias para la transposición y no tienen la configuración necesaria para moverse alrededor del genoma, sin embargo su movimiento es facilitado por los elementos transponibles autónomos. Los elementos autónomos de los retrotransposones LTR contienen al menos dos genes llamados: *gag* y *pol*, el gen *gag* (grupo asociado al antígeno) codifica una proteína estructural llamada cápsida y el gen *pol* (polimerasa) codifica una poliproteína que provee la maquinaria enzimática para transcripción reversa e integración en el genoma anfitrión y es responsable para las actividades de: proteasa (PR), reverso transcriptasa (RT), RNase H (RH) e integrasa (IN) (Liu et al., 2019; Ramakrishnan et al., 2022; Vicient & Casacuberta, 2020).

Como la transcripción de los retrotransposones LTR inicia y termina dentro de la región LTR, la región aguas arriba de TSS en la hebra 5’ y aguas abajo del sitio de terminación en 3’ pierde el transcrito inicial que sirve como templado para la transcripción reversa, estas regiones se perderían a menos que sean realmacenadas en un mecanismo complejo durante la síntesis de cDNA, este mecanismo de restablecimiento de información perdida en los extremos LTR se basa en las secuencias homologas de los

elementos transponibles que incluyen dos dominios internos como son: sitio de unión de un primer (PBS) y la vía polipurina (PPT) (Thieme & Bucher, 2018).

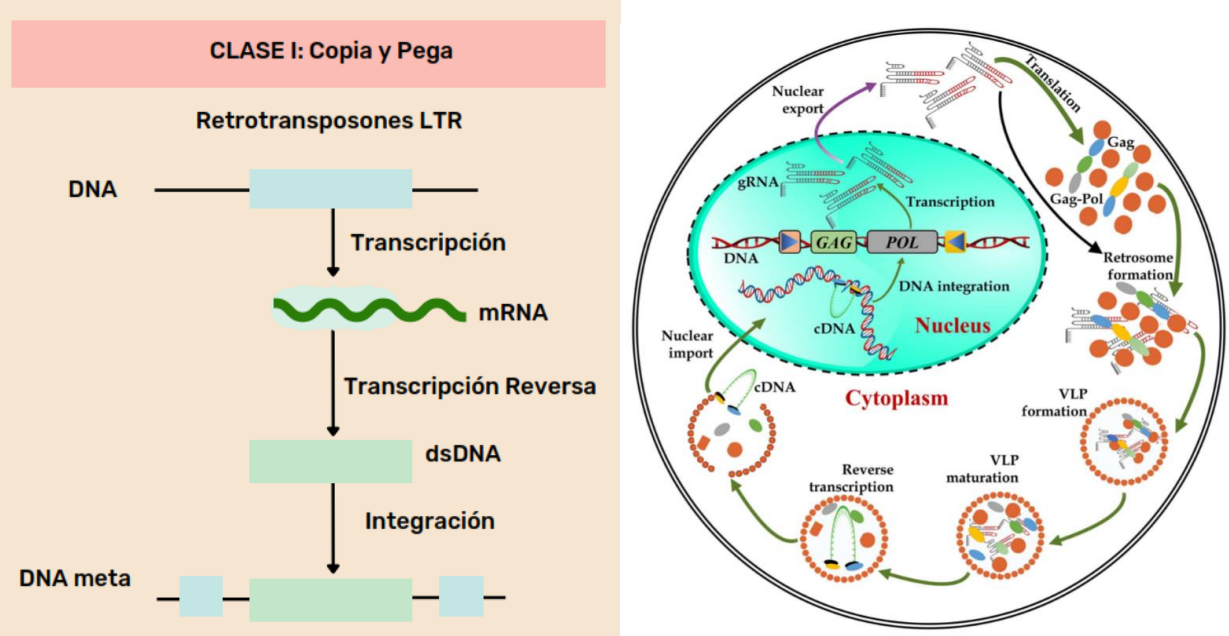

Figura 4. Mecanismo de transposición de los retrotransposones LTR en la célula, adaptado de (Ramakrishnan et al., 2022).

Existen retrotransposones LTR no autónomos que están presentes en plantas, tienen una región interna no codificante cuyo tamaño es de 0.3 kb para los retrotransposones de repetición terminal en miniatura (TRIM) y retrotransposones pequeños LTR (SMART) y hasta 3.5 kb para los retrotransposones largos derivativos (LARD). Estos elementos carecen de regiones codificantes necesarias para la producción de su propia maquinaria de replicación, por lo que dependen de proteínas codificadas por otros elementos autónomos relacionados. Los retrotransposones LTR representan la clase más larga de elementos transponibles en genomas de plantas conocidas (Thieme & Bucher, 2018).

También se ha identificado a una nueva estructura no autónoma de terminal de repetición largo LTR llamada: repetición de terminal con dominio GAG (TR-GAG), la cual ha sido descrita en plantas monocotiledóneas, dicotiledóneas y en genomas de angiospermas basales, estos elementos son relativamente cortos, tienen una longitud menor a 4 kb y presentan características típicas de los retrotransposones LTR, sin embargo tienen un

único marco de lectura que codifica para la proteína GAG que está involucrada en la transposición, el ensamblaje y empaquetamiento del elemento similar a los virus; a pesar de la carencia de la maquinaria enzimática que es necesaria para su movilidad, existe fuerte evidencia que sugiere que los TR-GAG son elementos activos (Chaparro et al., 2015).

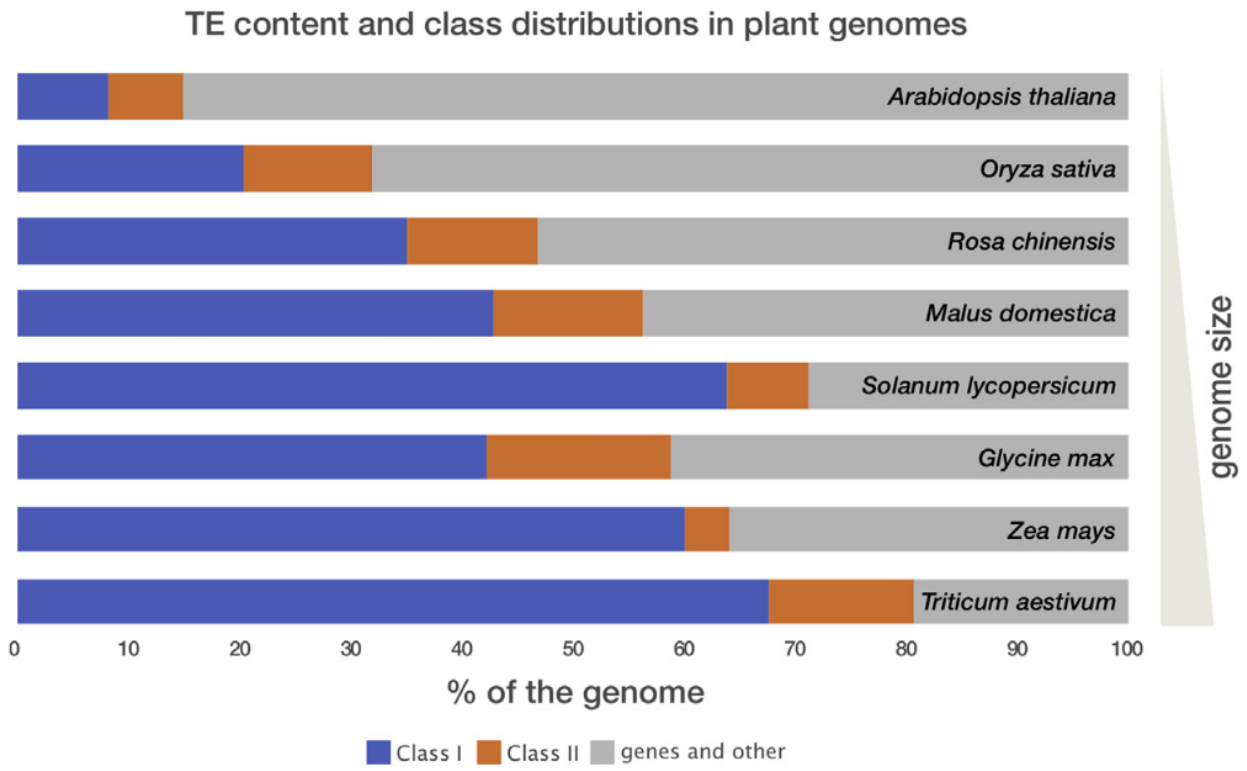

Figura 5. Porcentaje de genomas de plantas ocupados por elementos transponibles de dos clases principales (Thieme & Bucher, 2018).

2.2.4. Impacto de los elementos transponibles en el tamaño del genoma.

Los elementos transponibles representan porciones substanciales de los genomas de plantas, también cumplen un rol como elementos estructurales que dan forma a la arquitectura del genoma. Los análisis estructurales y las comparaciones de varios genomas de angiospermas secuenciados, incluyendo arabidopsis, soya y arroz, han demostrado que existe una fuerte relación entre el contenido de elementos transponibles y el tamaño actual del genoma; la proliferación y el aumento de los retrotransposones a

través del mecanismo de copia y pega, constituye el principal controlador de la expansión genómica en plantas (Thieme & Bucher, 2018).

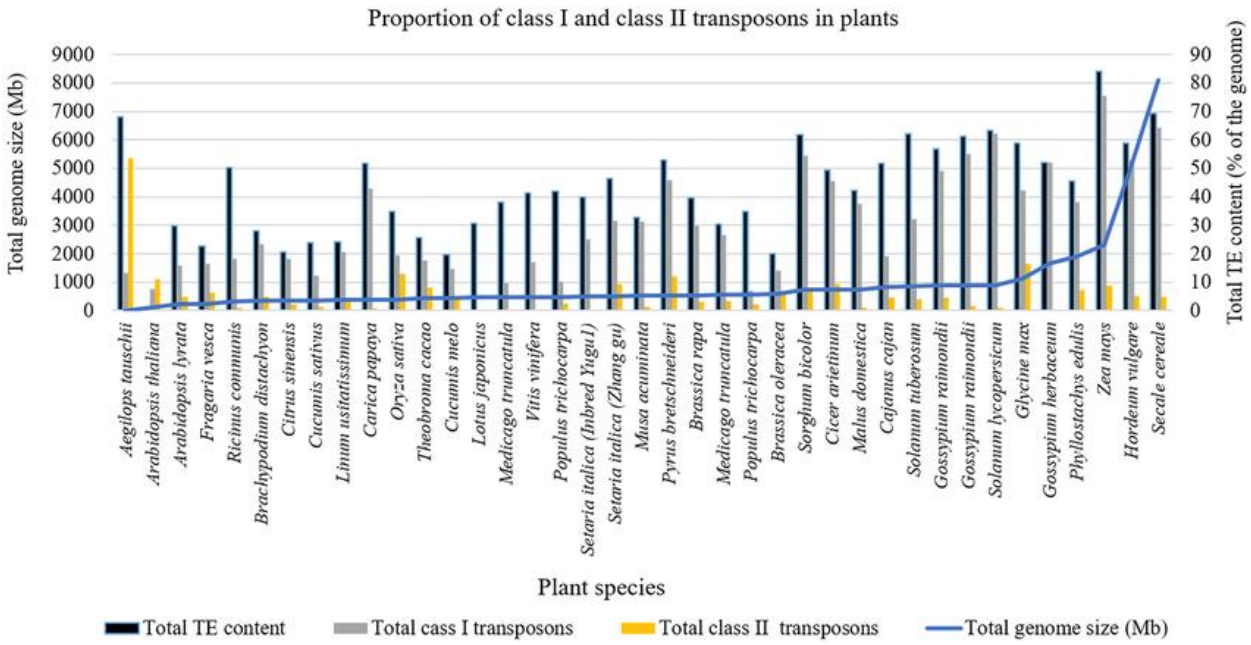

Figura 6. Proporción de transposones en el genoma de diferentes especies de plantas, las barras azules representan el contenido total de elementos transponibles, barras grises los transposones de clase I, barras amarillas los transposones de clase II y la línea azul el tamaño de los genomas en Mb (Ramakrishnan et al., 2022).

2.3. Bases de datos de elementos transponibles.

Las bases de datos de elementos transponibles se clasifican en dos tipos: las primeras analizan y clasifican a los elementos en base a su filogenia por linaje y dominio proteico como GyDB y REXdb, el segundo tipo de bases de datos identifican y caracterizan a los elementos transponibles en especies específicas como GrTEdb y DPTEdb (Zhou et al., 2021) o de tipo general como InpactorDB (Orozco-Arias et al., 2021).

La base de datos de REXdb divide a los retrotransposones Copia y Gypsy en 16 y 14 linajes respectivamente en base a los dominios conservados poliproteicos de muestras de secuencias de 80 especies (Zhou et al., 2021). Dentro de las bases de datos de elementos transponibles de plantas se encuentran: RepetDB, la cual es una de las más

completas, Repbase Update que ya no es disponible libremente desde el 2018 y Oryza repeat Database que contiene solamente secuencias de arroz (Edwards, 2022).

Tabla 4. Lista de bases de datos de elementos transponibles de plantas, adaptado de (Edwards, 2022; Ramakrishnan et al., 2022).

Base de datos	Contenido	Sitio web
RepeatDB	23 especies de plantas	http://urgi.versailles.inra.fr/Data/Transposable-elements
REXdb	Dominios de plantas (retrotransposones LTR)	http://repeatexplorer.org/?page_id=918
InpactorDB	Especies de plantas (retrotransposones LTR)	https://doi.org/10.3390/genes12020190
Dfam	Genomas eucariotas	https://dfam.org/repository
Repbase	Genomas eucariotas	https://doi.org/10.1186/s13100-015-0041-9
Oryza repeat Database	Solo arroz	http://rice.uga.edu/ http://rice.uga.edu/annotation_oryza.shtml
DPTEdb	Plantas dioicas	http://genedenovoweb.ticp.net:81/DPTEdb/index.php
MnTEdb	Solamente plantas de mora	https://morus.swu.edu.cn/mntedb/
ConTEdb	Plantas coníferas	http://genedenovoweb.ticp.net:81/conTEdb/index.php
SPTEdb	Plantas salicáceas	http://genedenovoweb.ticp.net:81/SPTEdb/index.php
SoyTEdb	Plantas de soya	https://www.soybase.org/
GyDB (Gypsy Database)	Dominios de plantas (retrotransposones LTR)	https://gydb.org/index.php/Main_Page
TIGR	Plantas de maíz	https://www.jcvi.org/research/maize-cell-genomics-resources-visualizing-promoter-activity-and-protein-dynamics-using

2.4. Métodos para identificar y clasificar a elementos transponibles.

2.4.1. Homologías a secuencias.

El método más usado para detectar elementos transponibles es por homologías a secuencias de elementos transponibles conocidos, la búsqueda inicia a partir de bases de datos existentes con familias de elementos transponibles identificadas y los nuevos elementos son clasificados en la familia que contiene la secuencia más similar, se realiza en secuencias que codifican para proteínas (reverso transcriptasa) ya que para alcanzar un nivel significante de similitud se requiere de un nivel suficiente de conservación a través de la evolución. Los métodos de detección basados en homologías proteicas presentan varias ventajas como son: el conocimiento previo que ha sido obtenido a partir de un gran número de secuencias reportadas de elementos transponibles, detectan fácilmente elementos transponibles que están presentes en un sola copia en el genoma y pueden proveer una clasificación entre transposones y retrotransposones (Edwards, 2022).

Un dominio puede ser definido como una región conservada y funcional en una secuencia proteica, que puede formar una estructura semiindependiente 3D en las proteínas, una proteína puede tener uno o más dominios; los dominios tienen un rol importante en las funciones proteicas específicas que evolucionan genéticamente y las transfieren dentro de diferentes organismos, algunas familias proteicas son conocidas por haber surgido de ancestros comunes al obtener diferentes combinaciones de dominios, la identificación de estos dominios conservados es importante para entender la función de las proteínas (Forero, 2022; Singh & Pathak, 2022).

Los perfiles de modelos ocultos de Markov (HMM) son modelos probabilísticos usados para predecir homologías, pueden ser generados al realizar alineamientos a un conjunto definido de secuencias que son representativas de familias y que tienen funciones conocidas, capturan información específica de la posición relacionada con cambios evolutivos que han sucedido en un conjunto de secuencias homologas, las cuales han sido alineadas por medio de alineamientos múltiples de secuencias. Un valor umbral es usado para clasificar a una proteína como miembro de una familia basada en HMM,

existe un umbral o límite predefinido para remover falsos positivos que es llamado el umbral de concurrencia (gathering threshold), las secuencias que presentan un valor mayor a este umbral son alineadas a los perfiles definidos para generar un alineamiento completo (Forero, 2022; Ismail, 2022).

Las regiones conservadas en motivos o dominios se observan en alineamientos de secuencias múltiples y pueden ser descritas como perfiles, los cuales constituyen una descripción cuantitativa de un motivo o dominio que pueden ser puntuados en base a la ocurrencia de cada aminoácido, generando un modelo probabilístico usando los Modelos ocultos de Markov (HMM) (Kaufmann, Klinger, & Savelsbergh, 2017). Para crear un perfil HMM primero se crea una matriz de puntuación de posición especifica (PSSM) de los alineamientos múltiples de secuencias, los aminoácidos en cada posición del alineamiento tienen una puntuación de frecuencia. La clasificación proteica que está basada en homología es usada por la base de datos proteicos llamada pfam, los perfiles HMM pueden ser obtenidos de varias fuentes como: TIGRFAMS, PRKs, pfam, entre otros (Forero, 2022; Ismail, 2022).

Los métodos populares como los modelos ocultos de Markov (HMM) y PSI-BLAST son ampliamente usados para detectar dominios en secuencias proteicas (Forero, 2022; Singh & Pathak, 2022). El perfil HMM ha sido considerado uno de los métodos que es capaz de retornar más proteínas correctas que están relacionadas distantemente, que el método de BLAST (Kaufmann et al., 2017; Singh & Pathak, 2022).

El sistema de clasificación de dominios proteicos organiza a las secuencias proteicas en familias tomando en cuenta la similitud de secuencias, los miembros proteicos de una familia comparten una significativa similitud de secuencias que puede ser detectada por el programa llamado HMMER3, el cual es un paquete de software libre que es utilizado para identificar proteínas homologas en base a la comparación de perfiles de modelos ocultos de Markov (HMM) contra una secuencia o a una base de datos de secuencias proteicas (Ismail, 2022). La búsqueda de HMMER de elementos transponibles de una familia fijada involucra tres pasos: primero se alinea las copias conocidas de la familia de elementos transponibles (HMMalign), posteriormente se crea el perfil HMM de esta

familia (HMMbuild y HMMcalibrate) y finalmente se escanea la secuencia genómica con el perfil HMM como modelo (HMMsearch) (Edwards, 2022). Este método se utilizó para clasificar a los retrotransposones en LTR en un reciente pipeline denominado TEsorter (R. G. Zhang et al., 2022).

2.4.2. Estructural.

Otro método usado para detectar elementos transponibles se basa en la estructura, toma en cuenta características estructurales de elementos transponibles conocidos, por ejemplo las secuencias LTR presentan elementos comunes como son: presentan una cadena de repetición corta que marcan los extremos 5' TG…CA 3'en cada extremo, la vía polipurina de 1 bp de longitud y varios dominios proteicos. Los métodos puramente estructurales están limitados por el hecho de que para cada tipo de elemento transponible se tiene que designar e implementar un modelo específico y algunos elementos transponibles son más fuertes estructuralmente por lo tanto son más fáciles de detectar y este método no puede identificar elementos transponibles con nuevas estructuras y tampoco aquellos elementos antiguos que carecen de características estructurales. Existe un tercer método para identificar elementos transponibles que es conocido como ab initio, en el cual no existe presunciones del tipo de transposón buscado y es un método usado para anotar genomas (Edwards, 2022; Peterson, 2013).

2.4.3. Método de *novo*.

El método de *novo* implica una propia comparación que requiere alinear un genoma o parte de él, se cuantifican k-mers exactos y no se necesita de información adicional de la secuencia consultada ("query"), sin embargo el número bajo de copias de los elementos transponibles no puede ser reconocido como secuencias repetitivas en este método, presenta baja sensibilidad para secuencias altamente divergentes o antiguas y se produce una fragmentación de las secuencias largas de elementos transponibles. El método que usa k-mers, enumera el motivo repetido formando subcadenas para después extenderse a las secuencias de los extremos y formar una secuencia más larga que representa a una familia de elementos repetidos, en este tipo de algoritmo se selecciona aleatoriamente un k-mer de alta frecuencia como semilla para buscar secuencias que

contengan aquel k-mer y las secuencias resultantes se alinean para obtener una secuencia consenso. El método basado en genómica comparativa se caracteriza por comparar y relacionar secuencias genómicas completas para encontrar espacios de InDels provocados por los elementos transponibles, se requiere de genomas de referencia muy bien anotados debido a la gran diversidad de los elementos transponibles incluso en especies cercanamente relacionadas, de esta forma se han creado herramientas bioinformáticas para identificar elementos transponibles específicos como Inpactor 2 con aprendizaje profundo (Orozco-Arias et al., 2023; Peterson, 2013; Ramakrishnan et al., 2022).

Tabla 5. Lista de programas de búsqueda o de clasificación de retrotransposones LTR, adaptado de (Edwards, 2022; Orozco-Arias et al., 2019).

Nombre	Método	Tareas	Sistema operativo	Requisitos
LTR_MINER	Homología	Identificación	Todos	Perl
LTR_Finder	Estructura	Identificación	Linux	No
LTR_STRUC	Estructura	Identificación	Windows	No
LTRharvest	Estructura	Identificación	Linux, MacOS	Perl, Python, C compiler, GenomeTools
LTR_Retriever	Estructura	Identificación	Linux, MacOS	Perl, Conda
LTRClassifier	Homología	Clasificación	Todos, sitio web	Perl
MGEScan-LTR	Estructura	Identificación	Linux, MacOS	Python, C compiler, HMMER, EMBOSS, Galaxy
TEsorter	Homología HMM	Clasificación	Linux	Python HMMER

PASTEC	Estructura y homología HMM	Clasificación	Linux	Python
Inpactor	Pipeline, Estructura y homología	Clasificación Otro análisis	Linux	C, LTR_STRUCor Repeat
Inpactor2	De novo, aprendizaje profundo	Identificación y clasificación	Linux	Python

2.5. Medidas de rendimiento de algoritmos.

Los problemas de clasificación se categorizan en tres tipos: binarios, multiclase y multi etiqueta, en las tareas de clasificación binaria solamente se consideran dos clases que se refieren comúnmente como clase positiva y negativa, por ejemplo, casos saludables vs enfermos, subexpresado vs sobreexpresado, etc. Por el contrario, las tareas de multiclase incluyen más de dos clases y algunas de las medidas para clasificación binaria pueden extenderse a problemas multiclase. Una tarea de etiqueta simple quiere decir que un caso pertenece solo a una clase, mientras que la multi etiqueta se refiere a que un caso puede pertenecer simultáneamente a más de una clase (Ranganathan et al., 2019).

Algunas medidas de rendimiento elementales pueden derivarse de una clasificación simple y binaria, sus resultados se representan en una matriz de confusión de 2x2 como la Tabla 6 (Ranganathan et al., 2019), en la que se ubican cuatro valores:

- Verdaderos positivos (TP), constituyen los casos que son realmente positivos y fueron predichos como positivos.
- Falsos positivos (FP), es el número de casos que son realmente negativos pero fueron predichos como positivos.
- Falsos negativos (FN), son los casos que en realidad son positivos pero fueron predichos como negativos.
- Verdaderos negativos (TN), es el número de casos que en realidad son negativos y fueron predichos como negativos (Ranganathan et al., 2019).

Tabla 6. Matriz de confusión de 2x2, adaptada de (Ranganathan et al., 2019).

		Real		
		Clase Positiva	Clase Negativa	
Predich	Clase Positiva	TP	FP	TP+FP
	Clase Negativa	FN	TN	FN+TN
		TP+FN	FP+TN	TP+FP+FN+TN

El número de falsos positivos también se conoce como error de Tipo I y el número de falsos negativos como error de Tipo II. A partir de la matriz de confusión se derivan algunas medidas de rendimiento elementales (Ranganathan et al., 2019).

2.5.1. Sensibilidad.

La sensibilidad de una algoritmo de clasificación, también conocida como memoria (recall) o tasa de verdaderos positivos (TPR), es la proporción de verdaderos positivos con respecto al número total de instancias positivas (Ranganathan et al., 2019).

$$Sensibilidad = \frac{TP}{TP + FN} \quad (\text{I})$$

2.5.2. Precisión.

La precisión o valor predicho positivo es el número de casos positivos predichos correctamente divididos para el numero de todos los casos que son predichos como positivos (Ranganathan et al., 2019).

$$Precisión = \frac{TP}{TP + FP} \quad (\text{II})$$

2.5.3. Especificidad.

La especificidad o tasa de verdaderos negativos (TNR) es el número de casos negativos predichos correctamente divididos para el numero de los casos negativos. Una baja

sensibilidad representa un alto número de falsos negativos y una baja especificidad indica la presencia de muchos falsos positivos (Ranganathan et al., 2019).

$$Especificidad = \frac{TN}{FP + TN} \quad \text{(III)}$$

2.5.4. Exactitud.

La exactitud de un algoritmo de clasificación ("accuracy") se considera como la proporción de las clasificaciones correctas. La forma más simple de estimar la exactitud es calcular el porcentaje de elementos de una clase clasificados correctamente (instancias positivas) y el porcentaje de elementos de una segunda clase clasificados correctamente (instancias negativas) (Ranganathan et al., 2019).

$$Exactitud_{positiva} = 100 \cdot \frac{TP}{TP + FN} \quad \text{(IV)}$$

$$Exactitud_{negativa} = 100 \cdot \frac{TN}{TN + FP} \quad \text{(V)}$$

La exactitud puede ser calculada tomando en cuenta todas las instancias correctamente clasificadas como (*VI*) (Ranganathan et al., 2019).

$$Exactitud = 100 \cdot \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(VI)}$$

La exactitud promedio se obtiene con el porcentaje de exactitud de ambas clases, positiva y negativa (Ranganathan et al., 2019).

$$Exactitud_{promedio} = 100 \cdot \frac{Exactitud_{positiva} + Exactitud_{negativa}}{2} \quad \text{(VII)}$$

2.5.5. Tasa de error.

La tasa de error (error rate) es la proporción de clasificaciones incorrectas (Ranganathan et al., 2019).

$$Tasa\ de\ error = \frac{FP + FN}{TP + FP + FN + TN} \quad (VIII)$$

2.5.6. Valor predicho negativo.

Valor predicho negativo corresponde al número de casos negativos predichos correctamente dividido para el numero de todos los casos que son predichos como negativos (Ranganathan et al., 2019).

$$Valor\ predicho\ negativo = \frac{TN}{TN + FN} \quad (IX)$$

2.5.7. Tasa de descubrimiento de falsos.

La tasa de descubrimiento de falsos (FDR) es el número de falsos positivos en relación con el número de casos que fueron predichos como positivos (Ranganathan et al., 2019).

$$FDR = \frac{FP}{FP + TP} \quad (X)$$

2.5.8. Medida F.

La medida F (F-score) se define como la media armónica de la sensibilidad (recall) y precisión, multiplicada por 2 para obtener una puntuación de 1 cuando la sensibilidad y especificidad es igual a 1 (Ranganathan et al., 2019).

$$F = 2 \cdot \frac{1}{\frac{1}{Sensibilidad} + \frac{1}{Precisión}} = 2 \cdot \frac{Precisión \cdot Sensibilidad}{Precisión + Sensibilidad} \quad (XI)$$

2.5.9. Curva ROC.

La curva ROC (“Receiver Operating Characteristic”) es una representación gráfica de la tasa de verdaderos positivos (sensibilidad) en función de la tasa de falsos positivos (tasa de falsa alarma), un clasificador con una tasa de verdaderos positivos del 100% y sin

falsos positivos correspondería a la coordenada (0,1) en el espacio de ROC y el nivel de oportunidad seria la diagonal del gráfico.

$$Tasa\ de\ falsos\ positivos = \frac{FP}{FP + TN} \qquad (\text{XII})$$

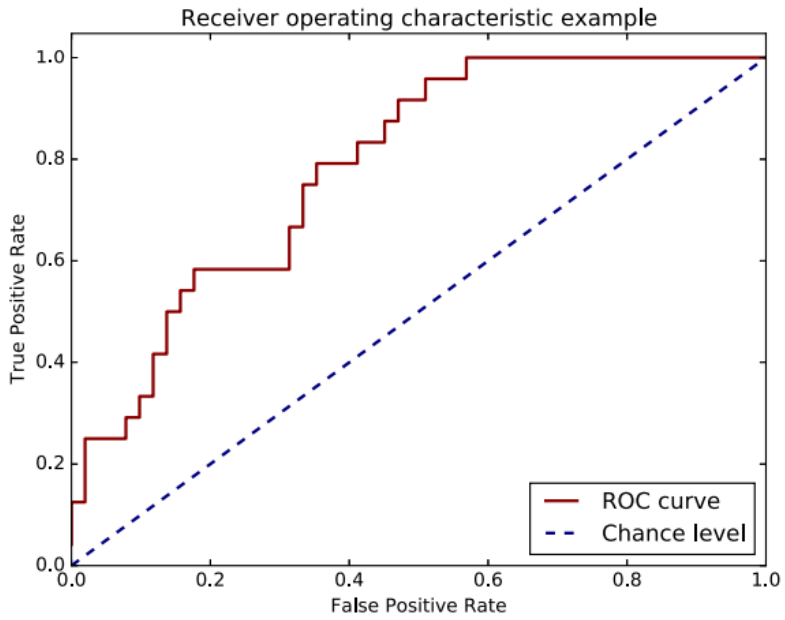

Figura 7. Representación esquemática de una curva ROC (Ranganathan et al., 2019).

2.5.10. Coeficiente de correlación de Matthews.

El coeficiente de correlación de Matthews (MCC) se define como (XIII), toma valores desde -1 a +1, donde +1 indica una correlación perfecta que revela que el modelo predice todos los casos positivos reales como positivos y todos los casos negativos reales como negativos (Ranganathan et al., 2019).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \qquad (\text{XIII})$$

CAPÍTULO III.

MARCO METODOLÓGICO.

3.1. Contexto y clasificación de la investigación.

Para la realización de este proyecto se empleó el diseño de la investigación cuantitativa, de tipo no experimental y transversal, con el método de ensayo y error para probar las instrucciones de programación hasta conseguir resultados adecuados. La variable independiente corresponde a los elementos transponibles de tipo retrotransposones LTR, la variable dependiente es el desarrollo de un algoritmo para identificar elementos transponibles. Las herramientas empleadas constituyen: bases de datos biológicas de plantas, herramientas bioinformáticas para determinar retrotransposones LTR, algoritmos de programación realizados en el lenguaje de programación de Python con sus librerías de análisis biológico.

3.2. Técnicas e instrumentos de recolección de datos.

La recolección de los datos para el desarrollo de este proyecto se realizó mediante el método de datos secundarios, con las técnicas de investigación documental de tipo online. La fuente de datos principal constituye la información disponible en las bases de datos genómicas de plantas, que se encuentran disponibles libremente en la web.

3.3. Técnicas para el procesamiento de datos.

La información que fue recogida de archivos en formato Fasta de bases de datos genómicas, se procesó usando algoritmos de programación escritos en el lenguaje de programación Python; se utilizó técnicas de codificación, tabulación y probabilísticas.

3.4. Técnicas de síntesis de resultados.

El método usado para la discusión y síntesis de resultados de la investigación fue estadístico de medidas de rendimiento como: precisión, exactitud, sensibilidad, especificidad, puntaje F1 y coeficiente de correlación de Matthews, derivadas de una matriz de confusión de clasificación binaria. Los gráficos estadísticos se realizaron en el

entorno de RStudio, el cual es un software libre y usa el lenguaje de programación R para el análisis estadístico y generación de gráficos (Curry, 2021; MacFarland & Yates, 2021).

3.5. Materiales.

3.5.1. Bases de datos de elementos transponibles.

Se usó dos bases de datos de referencia contra las cuales se analizaron las secuencias de entrada, que contienen secuencias de dominios proteicos conservados de elementos retrotransponibles LTR, la primera es REXdb la cual es una base de datos completa que ha sido usada ampliamente como la base de referencia de RepeatExplorer (Novák, Neumann, Pech, Steinhaisl, & Macas, 2013), está compuesta por 13 795 secuencias de elementos LTR, clasificadas a nivel de linaje, pertenecientes a 80 especies viridiplantae, se descargó del sitio web de REXdb (Neumann et al., 2019; Neumann, Novák, Hoštáková, & Macas, 2022), mientras que la segunda base de datos es GyDB que cuenta con secuencias de perfiles de Modelos Ocultos de Markov a nivel de linaje, se obtuvo del sitio web de GyDB (Llorens et al., 2011, 2023).

Para probar el funcionamiento del programa se utilizó bases de datos de elementos transponibles, disponibles libremente en la web, una de ellas es InpactorDB la cual es una base de datos semicurada compuesta de 130511 elementos de retrotransposones LTR clasificados a nivel de linaje, que corresponden a 195 genomas de plantas (pertenecientes a 108 especies), su versión no redundante consiste de 67305 retrotransposones LTR, además existe un archivo disponible de instancias negativas que no corresponden a secuencias de elementos transponibles LTR, estas bases de datos están disponibles en formato fasta, se descargaron del repositorio de Zenodo (Orozco-Arias et al., 2021, 2022). Otra base de datos empleada fue RepetDB (Amselem et al., 2019), que es de acceso público y contiene secuencias estandarizadas de elementos transponibles de genomas de varias especies de plantas, de las cuales se seleccionó: Arabidopsis thaliana, obtenida a través de la página web de URGI (Unité de Recherche en Génomique et bio-Informatique) (Amselem et al., 2022a) y *Zea maize*

ZmB73_RefGen_v3, descargada del sitio web URGI (Unité de Recherche en Génomique et bio-Informatique) (Amselem et al., 2022b).

Se usó dos librerías curadas de elementos transponibles para validar el algoritmo, una de ellas es la librería del arroz (Oryza sativa v6.9.5) que se caracteriza por su alta calidad, las secuencias de elementos transponibles LTR representativos del genoma del arroz se encuentran curadas manualmente, está compuesta de 897 secuencias con una longitud de 2.34 Mb que representan a 508 elementos LTR no redundantes, se obtuvo del repositorio github de EDTA, https://github.com/oushujun/EDTA (Ou et al., 2023); la segunda librería fue del maíz (Zea Mays TE11122019, Maize TE Consortium (MTEC)), también se encuentra curada manualmente, es de alta calidad, compuesta de 1362 secuencias de elementos transponibles, se descargó del repositorio github de EDTA (Ou et al., 2023).

3.5.2. Secuencias genómicas.

Para probar el funcionamiento del algoritmo con secuencias genómicas como entrada, se seleccionaron genomas ensamblados de dos especies de plantas modelo, que han sido ampliamente usadas en diferentes estudios de algoritmos computacionales, como son: del arroz (*Oryza sativa* ssp. japónica IRGSP1) que tiene un tamaño de alrededor 500 Mb, se descargó de la base de datos publica de Ensembl (IRGSP, 2022) y de *Arabidopsis thaliana* (TAIR10) con tamaño de 135 Mb, se obtuvo desde el sitio web de Ensembl (TAIR, 2023).

3.5.3. Recursos computacionales.

El algoritmo implementado fue ejecutado en una computadora personal Intel Core i7 de decima generación, que presenta las siguientes características: sistema operativo Windows 10 de 64 bits, CPU de 2.60 GHz, memoria RAM de 32GB, resolución de pantalla de 1920x1080 Px, en la cual corre la máquina virtual de Ubuntu con (v20.0).

3.5.4. Lenguaje de programación y ambiente de trabajo.

Se trabajó con el sistema operativo Linux con la distribución Debian en la máquina virtual Ubuntu, el programa fue desarrollado en el lenguaje de programación de Python (v3.10) que es un lenguaje de alto nivel, orientado a objetos, de libre acceso, versátil y ampliamente usado en distintas aplicaciones (Mastrodomenico, 2022; The Python Software Foundation, 2022). Se usó la plataforma de anaconda para controlar el ambiente de trabajo de Python, tiene la ventaja de aislar ambientes de trabajo con la finalidad de permitir instalar y actualizar paquetes fácilmente en cado uno de ellos, sin que exista dependencia (Anaconda Inc, 2022; Meador, 2022).

La instalación de los paquetes bioinformáticos se realizó a través de Bioconda, la cual es una distribución de programas especializada para biología computacional, funciona de forma versátil como un canal hacia el director de paquetes de Conda, de esta manera Bioconda soporta los sistemas operativos Linux y macOS, permite instalar rápidamente herramientas y paquetes de programas relacionados a la investigación bioinformática y biomédica por medio de Conda, para mejorar el flujo de trabajo y evitar errores de interdependencia entre paquetes con distintas versiones (Bioconda, 2022; Grüning et al., 2018). Para instalar correctamente los paquetes en Python se configuró los canales, con los comandos indicados en el manual de Bioconda, en el mismo orden (Bioconda, 2022).

Para desarrollar las instrucciones de programación y ejecutar el código de Python fácilmente se instaló el IDE o editor de texto de distribución libre: Visual Studio Code (v 1.78), el cual es muy empleado ya que se puede ejecutar comandos en la consola rápidamente, además cuenta con herramientas de desarrollo y con un amplio ecosistema de extensiones para diferentes lenguajes (Microsoft, 2022; Speight, 2021).

Las librerías usadas fueron instaladas a través del ambiente Anaconda, se instaló el módulo de Biopython el cual contiene librerías, clases y módulos de alta calidad y reusables para el análisis de datos biológicos. Entre las características más importantes que presenta: es su capacidad para leer diferentes archivos bioinformáticos que pueden ser iterados en forma de secuencia por secuencia, la presencia de una clase denominada secuencia que permite acceder fácilmente a la identificación y características de una

secuencia por medio de métodos establecidos y tiene recursos para desarrollar operaciones comunes en las secuencias como por ejemplo traducción y transcripción (Chang et al., 2023; Cock et al., 2009).

3.6. Cronograma de actividades.

Tabla 7. Cronograma de actividades.

	MESES A PARTIR DE SEPT 2022 (INICIO) HASTA JULIO 2023 (FIN).										
ACTIVIDADES	SEP 2022	OCT 2022	NOV 2022	DIC 2022	ENE 2023	FEB 2023	MAR 2023	ABR 2023	MAY 2023	JUN 2023	JUL 2023
ASIGNACIÓN DEL TEMA Y TUTOR DEL PROYECTO											
REVISIÓN BIBLIOGRÁFICA											
REDACCIÓN DEL CAPÍTULO I INTRODUCCIÓN											
DISEÑO DEL ALGORITMO											
CORRECCIÓN DEL CAPÍTULO I INTRODUCCIÓN											
REDACCIÓN DEL CAPÍTULO II MARCO TEÒRICO											
ELABORACIÓN DEL ALGORITMO											
CORRECCIÓN DEL CAPÍTULO II MARCO TEÒRICO											
DISEÑO DE INTERFAZ GRAFICA											
REALIZACIÓN DE PRUEBAS PRELIMINARES.											
REDACCIÓN DEL CAPÍTULO III METODOLOGÍA											
CORRECCIÓN DEL CAPÍTULO III METODOLOGÍA											
MODIFICACIONES DEL ALGORITMO											
VALIDACIÓN DEL ALGORIMO											
REDACCIÓN DEL CAPÍTULO IV RESULTADOS											

CORRECCIÓN DEL CAPÍTULO IV RESULTADOS											
PRESENTACIÒN DEL INFORME FINAL											

3.7. Metódica.

3.7.1. Método para identificar retrotransposones LTR.

Para identificar a los elementos retrotransponibles de larga terminal se empleó un modelo probabilístico que predice homologías y describe a una familia de secuencias relacionadas, el cual se conoce como Modelo Oculto de Markov (HMM), que ha sido ampliamente usado en el área de biología computacional; particularmente este modelo se aplicó en la búsqueda de regiones conservadas o dominios proteicos de secuencias biológicas (Ismail, 2022).

HMM es un método probabilístico empleado para el análisis lineal de secuencias, es utilizado en el área de bioinformática para cualquier tarea que pueda ser descrita como un proceso en el que se analicen secuencias de izquierda a derecha (Baxevanis, Bader, & Wishart, 2020). Las secuencias homólogas se cree que tienen ancestros comunes, por lo que comparten secuencias estructurales y funciones, el modelo HMM ha sido muy efectivo para identificar secuencias homólogas (Singh & Pathak, 2022).

El Modelo Oculto de Markov es superior a las Matrices de Puntuación de Posición Especifica (PSSM) porque puede modelar gaps, es más sensible y toma en cuenta correlaciones entre residuos de vecinos cercanos (Hasija, 2023). HMM incluye todas las posibles combinaciones de coincidencias y no coincidencias, es capaz de manejar eficientemente las longitudes variables de las secuencias y las dependencias condicionales, que no pueden ser calculadas con PSSM (Baxevanis et al., 2020).

Los eventos asociados a las secuencias genómicas son principalmente probabilísticos, como es el caso de la presencia de dominios proteicos dentro de una secuencia, por este motivo los Modelos Ocultos de Markov son usados para proveer una representación estadística de procesos reales biológicos y son muy adecuados para resolver distintos problemas de biología molecular como: búsqueda de perfiles, caracterización y

clasificación de familias proteicas, alineamientos múltiples de secuencias, identificación de sitios regulatorios, comparación de estructuras proteicas, predicción de estructura secundaria de proteínas, entre otros (Baxevanis et al., 2020; Rocha & Ferreira, 2018).

El cálculo de probabilidades sigue un proceso conocido como la cadena de Markov, la cual consiste de una serie de observaciones, en la que la probabilidad de una observación depende de observaciones previas; además constituye una cadena de eventos basados en un conjunto finito de estados, que tienen una dependencia serial, en donde el siguiente estado depende del estado actual (Baxevanis et al., 2020; Rocha & Ferreira, 2018). Una secuencia de ADN puede ser considerada un ejemplo de un Modelo Oculto de Markov porque la probabilidad de observación de una base, en una posición particular, depende de las bases que le preceden, así como también existen dependencias entre codones adyacentes (Baxevanis et al., 2020).

Un HMM está definido por cinco elementos: alfabeto de símbolos, conjunto de estados, probabilidad de estado inicial, probabilidad de emisión (representada como una matriz de emisión de un símbolo en un estado) y probabilidad de transición (representada como una matriz de cambio de un estado a otro). Los símbolos son las letras representativas de los aminoácidos, existen 20 símbolos que corresponden a las 20 letras que caracterizan a los aminoácidos (Rocha & Ferreira, 2018).

Cada una de las observaciones de las secuencias puede ser representada por diferentes estados del Modelo Oculto de Markov. El estado de coincidencias representa a los aminoácidos más probables de ser encontrados en cada posición del alineamiento, es decir considera la probabilidad de encontrar un aminoácido en una posición del alineamiento. El estado de inserción indica la adición de un aminoácido, mientras que el estado deleción representa la pérdida de un aminoácido por lo que la secuencia debe saltar a la siguiente posición para continuar el alineamiento. Cada uno de estos estados, así como la relación entre estados se ilustra gráficamente en la Figura 8, usando la representación original que fue introducida por Anders Krogh (Krogh, Larsson, von Heijne, & Sonnhammer, 2001), los cuadrados en color verde simbolizan los estados de coincidencias, donde “I” determina el inicio del alineamiento y “F” el fin del alineamiento,

en tanto que los rombos en color anaranjado representan a los estados de inserciones y los círculos en color rosado simbolizan los estados de deleción, las flechas representan un movimiento de un estado a otro y cada una está asociada con una probabilidad de desplazamiento entre estados (Baxevanis et al., 2020).

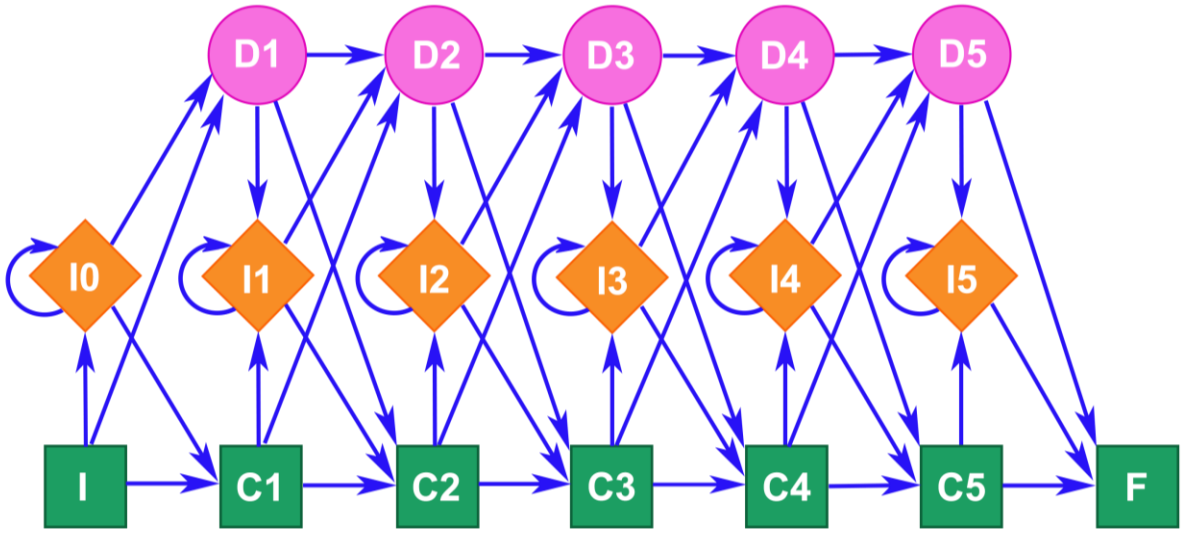

Figura 8. Representación de un HMM, "I" representa el inicio del alineamiento, "F" el fin del alineamiento, "C" coincidencias, "I0-I15" inserciones, "D1-D5" deleciones, adaptado de (Baxevanis et al., 2020).

El usuario puede ver la secuencia de aminoácidos que va a ser analizada, pero no puede realmente observar los estados en los que los aminoácidos se encuentran, de allí se deriva el término oculto, es decir cada estado emite una secuencia particular de aminoácidos con su propia probabilidad de emisión, este estado es oculto pero la secuencia en si es visible. Las probabilidades de transición y de emisión son derivadas de conjuntos de entrenamiento con secuencias que tienen estructura correcta (Baxevanis et al., 2020).

Una de las aplicaciones más populares de los Modelos Ocultos de Markov, en el análisis de secuencias biológicas, constituye los perfiles de Modelos Ocultos de Markov, los cuales producen perfiles probabilísticos de una familia proteica y son similares a las matrices de posición de pesos, pero además proporcionan una ruta más flexible para tratar con las inserciones y deleciones (Rocha & Ferreira, 2018). Los perfiles HMM capturan la diversidad de las secuencias biológicas, constituyen un método muy sensible

para detectar homólogos remotos (Nakaya, 2021); son modelos con estado de inserción y deleción asociados con cada estado de coincidencia, permiten la inserción y deleción en cualquier parte de la secuencia objetivo (Eddy, 1998). Además, modelan la probabilidad de que inserciones y deleciones ocurran más frecuentemente en ciertas secciones de una proteína que en otras (Gupta & Behera, 2021).

Los estados principales de un perfil HMM corresponden a las columnas consenso de un alineamiento múltiple y su cantidad se corresponde con las posiciones en el alineamiento; un estado de coincidencia modela la distribución de residuos, mientras que los estados de inserción representan a las regiones altamente variables en el alineamiento y los estados de deleción son también conocidos como estados de silenciamiento, no coinciden con ningún residuo y hacen posible el salto entre una o más columnas de los alineamientos. Este modelo tiene un estado para cada posición del alineamiento, cada estado presenta veinte resultados que corresponden a cada uno de los posibles aminoácidos y dos estados para las ocurrencias de inserción o deleción llamados InDels (Nakaya, 2021).

Los perfiles HMM usan un sistema de puntuación de posición específica para capturar información del grado de conservación de varias posiciones de un alineamiento múltiple (MathWorks, 2022). De esta forma se genera una matriz de puntuación de posición específica (PSSM), esta matriz permite crear un perfil HMM (Ismail, 2022). Se construyen los perfiles en base a cálculos de probabilidades de ocurrencia de aminoácidos en posiciones específicas, los puntajes se emplean para clasificar a las secuencias y dar una estimación de cuan similar es la nueva secuencia con la referencia (Nakaya, 2021).

Los alineamientos consisten de múltiples secuencias de genes que tienen una relación biológica significativa. La construcción del perfil HMM se lleva a cabo a partir de datos alineados provenientes de alineamientos múltiples de secuencias (MSA) de proteínas que tienen funciones conocidas (Ismail, 2022). En un típico perfil HMM se crea un modelo posicional o modelo basado en las posiciones de cada columna consenso del alineamiento, las columnas no consenso se tratan como inserciones (Wheeler, Clements, & Finn, 2014).

Los perfiles HMM proveen una estructura probabilística para la comparación de secuencias, potenciando la información contenida en un alineamiento de secuencias, para mejorar la detección de secuencias relacionadas distantemente. Son utilizados en la anotación de dominios proteicos y secuencias genómicas derivadas de la expansión de elementos transponibles antiguos (Wheeler et al., 2014).

El método HMM crea perfiles de las secuencias relacionadas y después se usan estos perfiles para identificar nuevas secuencias proteicas (Singh & Pathak, 2022). Los perfiles HMM han sido empleados en diferentes estudios para identificar elementos transponibles en genomas. El perfil correspondiente de una superfamilia de elementos transponibles es usado para buscar miembros de aquella superfamilia en un genoma, incluyendo copias de homólogos distantes (Fischer, Campos, & Barella, 2018).

La utilidad de este modelo proviene de la habilidad de entrenarlo con un conjunto de secuencias en lugar secuencias individuales, para cada secuencia se determina el camino más probable basado en puntuaciones probabilísticas. Las características colectivas de las secuencias de entrada permiten escanear secuencias contra una librería de perfiles HMM, para conocer si una nueva secuencia pertenece a una de las familias caracterizadas previamente (Baxevanis et al., 2020).

3.7.2. Método para clasificar retrotransposones LTR.

La clasificación de retrotransposones LTR se realizó en base al método de perfiles de Modelos Ocultos de Markov. Los perfiles HMM son usados como representantes de familias de secuencias de perfiles en lugar de una única secuencia, por lo que la clasificación de secuencias se basa en la similitud de perfiles HMM (Nakaya, 2021).

Las proteínas con secuencias y estructura relacionada se organizan en familias proteicas, pueden compartir funciones similares y relaciones evolutivas. Cuando se alinean regiones conservadas de múltiples secuencias de una misma familia, se obtiene una matriz de posición de pesos (PWM) que captura los patrones de conservación a lo largo de las diferentes posiciones del alineamiento (Rocha & Ferreira, 2018). Los aminoácidos en cada posición del alineamiento tienen un puntaje en función de su

frecuencia, se utiliza un valor umbral para clasificar a una proteína como miembro de una familia HMM (Ismail, 2022). Las secuencias que presentan una probabilidad o puntaje más alto que el valor umbral son consideradas potenciales miembros de la familia (Rocha & Ferreira, 2018).

El paquete HMMER, desarrollado por Sean Eddy (Potter et al., 2018), es una herramienta completa compuesta por diferentes métodos, usa la técnica de los Modelos Ocultos de Markov para las tareas de análisis de múltiples secuencias biológicas como la búsqueda de secuencias homólogas en bases de datos y en alineamientos múltiples de secuencias (Rocha & Ferreira, 2018). HMMER es un paquete de software libre que es utilizado para identificar proteínas homologas en base a la comparación de perfiles de modelos ocultos de Markov (HMM) contra una secuencia o a una base de datos de secuencias proteicas (Ismail, 2022).

El uso de Modelos Ocultos de Markov ha ayudado a encontrar más elementos transponibles con mayor exactitud, esto se puede observar en la base de datos de elementos repetitivos de Pfam (Finn et al., 2016) y Dfam (Wheeler et al., 2013) la cual usa secuencias de Repbase y las convierte en perfiles HMM (Baxevanis et al., 2020), adicionalmente esta base de datos de proteínas usa HMMER para crear sus perfiles (Sofi, Shafi, & Masoodi, 2022).

3.8. Implementación.

3.8.1. Interfaz gráfica de usuario.

El desarrollo del programa inició con la definición de librerías de Python, a continuación, se elaboraron las instrucciones de programación para crear una interfaz gráfica de usuario (GUI) con la finalidad de que las opciones del proceso de búsqueda de retrotransposones fueran de fácil acceso y manejo para el usuario. Se diseñó la interfaz usando la librería estándar de Python conocida como Tkinter (“Tool Kit Interface”), que ofrece una variedad de complementos como botones, cuadros de texto y de búsqueda, que pueden ser personalizados (Moore, 2021; Roseman, 2021). El algoritmo de la interfaz se mantiene en un lazo infinito hasta que el usuario cierre la aplicación, en la

ventana principal se crearon cada uno de los elementos de control como son: tres botones y 3 botones desplegables.

Se configuró la función de cada botón de la siguiente manera: el primer botón denominado “escoger archivo” muestra una nueva ventana que permite seleccionar el archivo de entrada que se encuentra guardado en el equipo, el primer botón desplegable muestra un menú para que el usuario elija el tipo de secuencias que se encuentran en el archivo de entrada, las cuales pueden ser de elementos transponibles (TE) o genómicas, el segundo botón desplegable permite escoger el tipo de secuencias a ser analizadas estas pueden ser de nucleótidos o proteínas, el tercer botón desplegable muestra un menú para que el usuario seleccione la base de datos de referencia contra la cual se va a analizar el archivo de entrada, los opciones que se presentan son la base de datos de referencia de REXdb y de GyDB, el segundo botón denominado “escoger ruta” despliega una ventana para elegir la ruta en la que se guardaran los archivos de salida y el tercer botón llamado “iniciar búsqueda” permite iniciar el proceso de búsqueda y clasificación de elementos transponibles LTR. El diseño de la interfaz gráfica se presenta en la Figura 9.

Figura 9. Diseño de la interfaz gráfica de usuario.

El algoritmo de la interfaz gráfica se puede apreciar en la Figura 10, el usuario puede escoger el archivo de entrada que contiene las secuencias a analizar, el tipo de secuencias de entrada, la base de referencia contra la cual se desea analizar y la ruta de salida en la cual se almacenaran los archivos obtenidos del análisis de las secuencias.

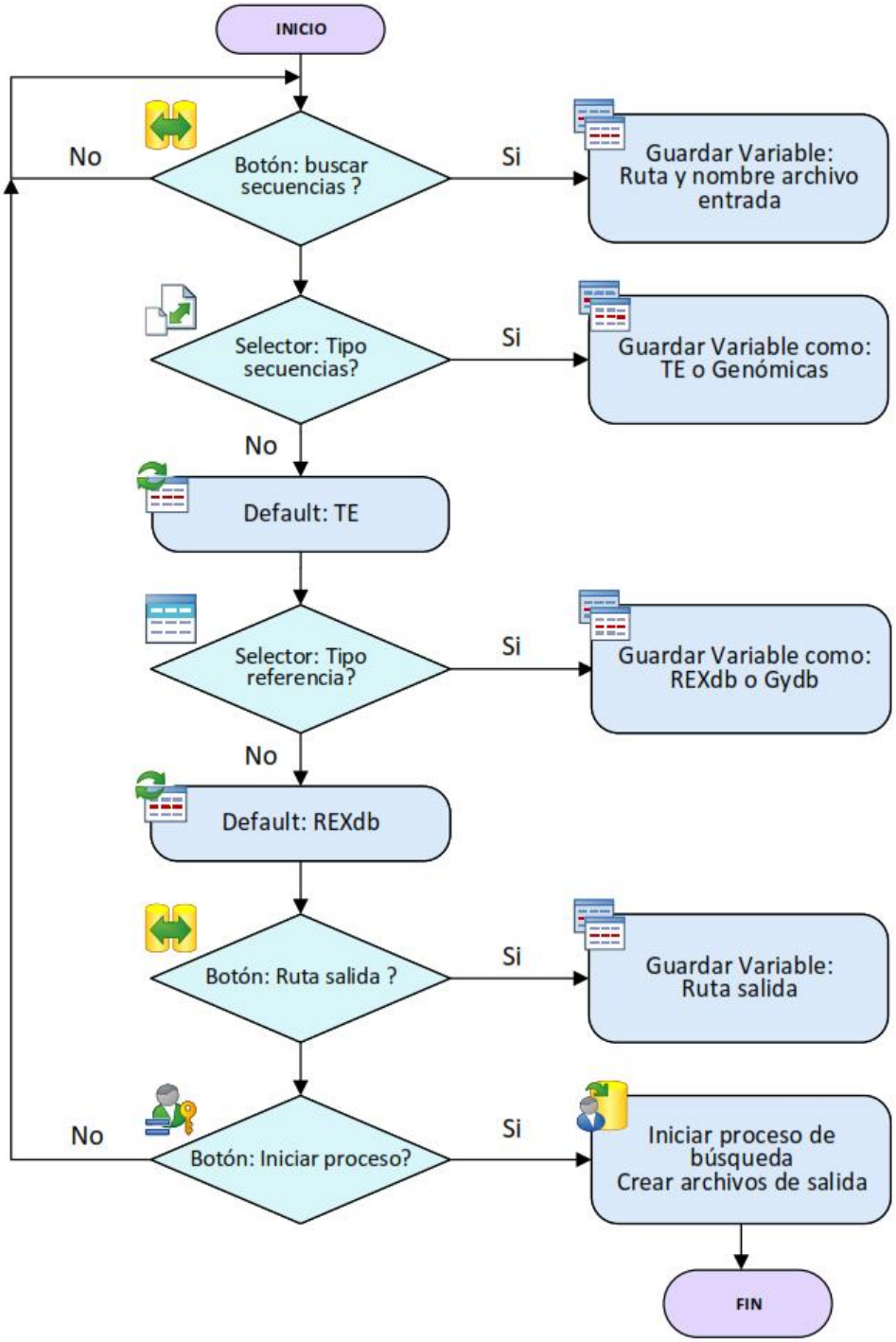

Figura 10. Esquema del algoritmo de la interfaz gráfica de usuario.

El proceso inicia con la verificación de que la secuencia de entrada haya sido escogida por el usuario, si existe un archivo de entrada el programa revisa que se encuentre en formato Fasta. Cuando el usuario no selecciona los campos referentes al tipo de secuencia y la base de datos de referencia, el programa tiene establecidos por defecto estos parámetros y analizará las secuencias como: nucleotídicas de elementos transponibles con la base de datos de referencia REXdb.

El programa no inicia el proceso de identificación de retrotransposones cuando todos los campos necesarios no han sido seleccionados y muestra diferentes mensajes de error, para ayudar a que el usuario escoja adecuadamente todos los campos. En caso de iniciar el proceso sin que el usuario haya seleccionado el archivo de entrada y la ruta de salida el programa mostrara un mensaje de error de campos vacíos. Así como también, en el caso de que las secuencias tengan una longitud menor a 1 Mbp y el usuario hubiera escogido la opción de genoma el programa muestra un mensaje de error de tipo de secuencia y de la misma manera si la longitud de las secuencias fuera mayor a 1 Mbp y el usuario hubiera escogido la opción de elementos transponibles.

3.8.2. Programa principal.

Se desarrolló una herramienta computacional llamada Arthur_LTRanalizer usando el lenguaje de programación de Python (v3.10). Las instrucciones se elaboraron en Visual Studio Code, en el ambiente de trabajo de Conda se configuraron los canales de Bioconda para cargar los paquetes bioinformáticos como son: Biopython para trabajar con secuencias genómicas y leer archivos en formato fasta, HMMER para usar el método de Perfiles de Modelos Ocultos de Markov que se descargó con Bioconda como archivo binario (HMM) (Eddy, 2020, 2022). Se usaron las siguientes librerías de Python: “Tkinter” para desarrollar la interfaz gráfica, “sys” y “os” para obtener rutas y nombres de archivos, “multiprocessing” para usar el multiprocesamiento, “re” para utilizar expresiones, “io” para verificar archivos biológicos y “Bio” para usar secuencias biológicas (Kong, Siauw, & Bayen, 2021).

3.8.3. Verificación y preprocesamiento de datos.

El tipo de archivo de entrada que la aplicación recibe es en formato fasta con secuencias de elementos transponibles o un archivo en formato FASTA con secuencias de un genoma. En esta fase se comprobó que el archivo de entrada se encuentre en formato FASTA y se verificó que exista correspondencia con la longitud de las secuencias y la opción escogida por el usuario en la interfaz gráfica. Para leer el archivo fasta se usó el método “SeqIO.parse” que permite leer secuencias y crear objetos SeqRecord iterables, este método está provisto en la librería be Biopython y ayuda a iterar y a acceder a cada una de las secuencias del archivo fasta. Arthur_LTRanalizer comprueba la longitud de las secuencias de entrada, si las secuencias tienen una longitud menor a 1 Mbp se consideran como secuencias de elementos transponibles, mientras que si la longitud es mayor a 1 Mbp se las considera secuencias genómicas.

En la etapa de preprocesamiento de datos, en el caso de secuencias de elementos transponibles y secuencias genómicas correspondientes a nucleótidos, se realizó el proceso de transcripción y traducción en un marco de seis, se crearon seis secuencias de aminoácidos, de las cuales tres correspondieron al sentido opuesto, para esto se utilizó el método “translate” del paquete de Biopython, junto con la primera tabla de codones presente en la librería. En el caso de que el usuario ocupe secuencias de proteínas, en lugar de secuencias nucleotídicas, se omitió el proceso de transcripción y traducción.

Cuando las secuencias a analizar fueran de tipo genómicas, el programa procesa a estas secuencias en pequeños fragmentos con la finalidad de que el consumo de memoria permanezca bajo, para lo cual se procedió a dividirlas teniendo en consideración una ventana de 270 000 bp y un solapamiento (overlap) de 30 000 bp, se decidió escoger esta estrategia porque incluye la longitud de un elemento transponible LTR, el solapamiento de una secuencia con otra contribuye a asegurar que todos los elementos puedan ser encontrados y que no se pierden partes del elemento, se aumenta la velocidad de búsqueda y ha sido comprobada en otras aplicaciones bioinformáticas como Tip_finder (Orozco Arias et al., 2020).

3.8.4. Identificación y clasificación de elementos transponibles.

El análisis de identificación y clasificación de retrotransposones LTR se basó en el método de perfiles de Modelos Ocultos de Markov (HMM). Se usaron perfiles de referencia de secuencias de dominios proteicos de retrotransposones LTR de REXdb y GyDB, para el procesamiento de la información se necesitó tres archivos adicionales, los cuales se obtuvieron con el comando “Hmmpress” del paquete HMMER , el cual realiza la compresión binaria e indexamiento de los archivos de perfiles de dominios proteicos de REXdb y GyDB.

La identificación de retrotransposones LTR se realizó mediante la búsqueda de homologías contra perfiles de dominios proteicos de REXdb y GyDB, con la ayuda del paquete HMMER se generaron archivos de salida en el formato de tabla de dominios del Modelo Oculto de Markov por medio del comando “Hmmscan”. Se compararon las secuencias de entrada que fueron traducidas contra una de las dos bases de referencia, para ello el query (secuencia de interés) constituyeron las secuencias de entrada que se desea analizar y el target (secuencia objetivo) correspondió a las secuencias de dominios proteicos de la base de referencia de REXdb y GyDB.

Los resultados obtenidos del escaneo contra los perfiles HMM fueron guardados y unificados en un nuevo archivo. Para cada uno de los elementos identificados se normalizó el puntaje (score) tomando en cuenta el puntaje del dominio y la longitud de la secuencia del target (Biryukov & Ustyantsev, 2021) y se calculó la cobertura teniendo en consideración a la longitud del perfil HMM del query y a la longitud del target.

Se hizo un procesamiento aplicando filtros para remover elementos de baja calidad y reducir la tasa de falsos positivos, aquellos elementos que presentaron una cobertura menor al 20% y un valor esperado (e-value) mayor a $1x10^{-3}$ fueron eliminado (Biryukov & Ustyantsev, 2021). Los elementos que se mantuvieron, junto con sus características se guardaron en un nuevo archivo de salida. El esquema general del flujo de trabajo de trabajo se muestra en la Figura 11.

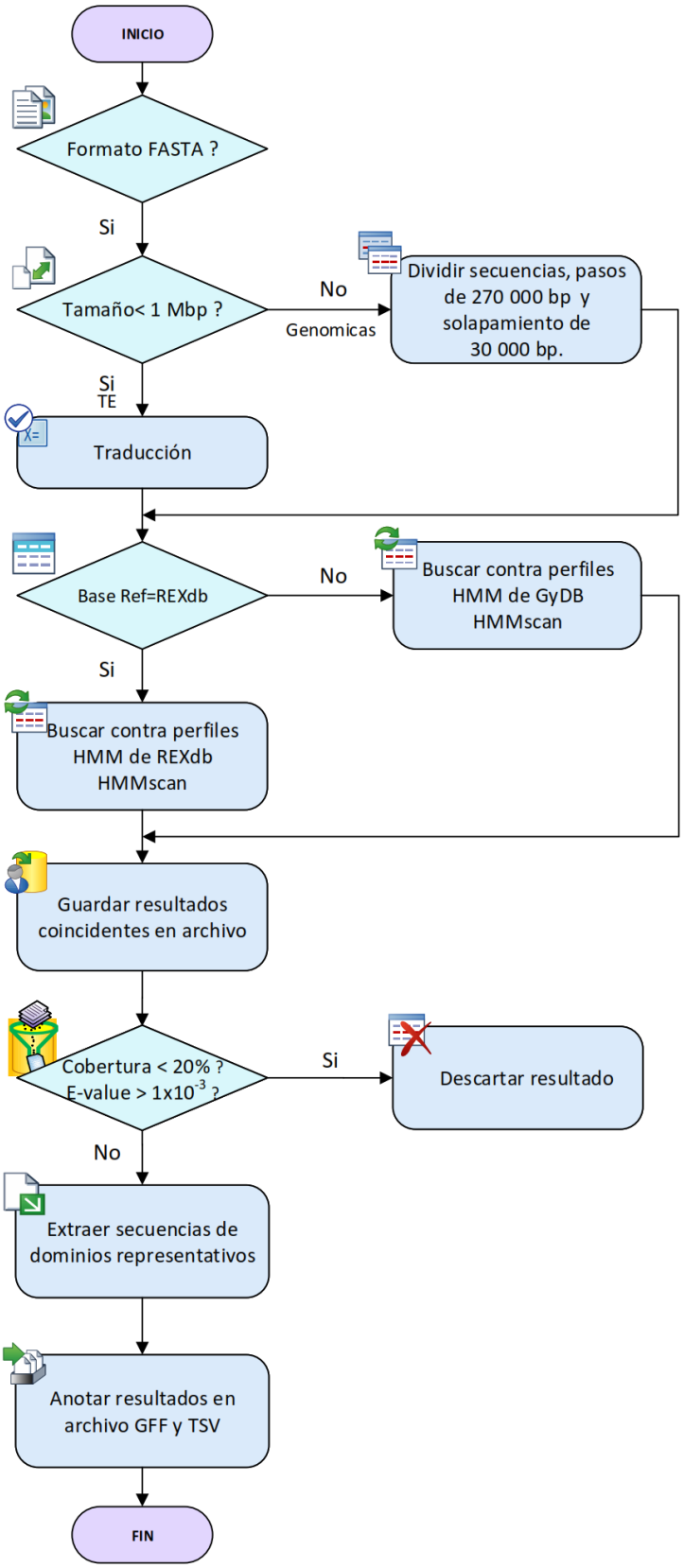

Figura 11. Representación esquemática del flujo de trabajo de Arthur_LTRanalizer. TE: elementos transponibles.

Para la clasificación se emplea una técnica basada en homologías, usando dominios proteicos codificantes que son conocidos de los retrotransposones LTR, de la base de datos de referencia de perfiles de dominios proteicos de REXdb y GyDB (Orozco-Arias et al., 2021). Los retrotransposones LTR fueron clasificados en las superfamilias Copia y Gypsy y en sus respectivos linajes en base a las coincidencias obtenidas de la comparación con los perfiles HMM de dominios proteicos de una de las dos bases de referencia. Además, las superfamilias Copia y Gypsy se consideraron completas tomando en cuenta la presencia y el orden de los dominios proteicos conservados, los cuales son: proteína capsida (GAG) (Chaparro et al., 2015), proteasa aspártica (AP), integrasa (INT), reverso transcriptasa (RT), RNase H (RH), que fueron descritas por Thomas Wicker (Wicker et al., 2007).

Se consideró como elementos retrotransponibles LTR completos de la superfamilia Copia cuando presentaron sus dominios conservados en el orden “GAG, PROT, INT, RT y RH”, descrito por Thomas Wicker (Wicker et al., 2007), en el caso de la superfamilia Gypsy se consideraron elementos completos cuando sus dominios conservados se encontraban en el orden: “GAG, PROT, RT, RH e INT”, descrito por Thomas Wicker (Wicker et al., 2007). En el caso de que existieran dominios con varias coincidencias se dio preferencia a los resultados que presentaron mayor puntaje. Se extrajeron las secuencias de los retrotransposones LTR que fueron clasificados a nivel de su linaje y se guardaron en un nuevo archivo.

3.8.5. Estrategia en paralelo.

Se tomó en cuenta la ventaja de las arquitecturas multinúcleos de las computadoras personales, por lo que se desarrolló el algoritmo del programa usando un proceso computacional de paralelización. Haciendo uso del multiprocesamiento se ejecutó un mismo proceso varias veces de forma simultánea con diferentes datos, esta estrategia ha sido empleada anteriormente en otros programas como Tip_finder (Orozco Arias et al., 2020).

La aplicación implementada cuenta con la función de multiprocesamiento, con el objetivo de reducir el tiempo de ejecución del programa. Este proceso inicia dividiendo el archivo de entrada que contiene secuencias biológicas en “n” archivos (donde n es el número de procesos), de esta manera cada proceso de trabajo se desarrolla de forma simultánea, esta estrategia se puede apreciar en la Figura 12. Cuando las secuencias de entrada son nucleotídicas, se realiza la traducción de las secuencias de cada archivo, posteriormente el archivo de salida o la salida traducida se escanea contra el perfil HMM de REXdb o GyDB, cada proceso genera un archivo de salida HMM, se junta cada uno de estos archivos en un único archivo unificado, finalmente se filtra el archivo y se generan los archivos de salida en formato tabular y GFF.

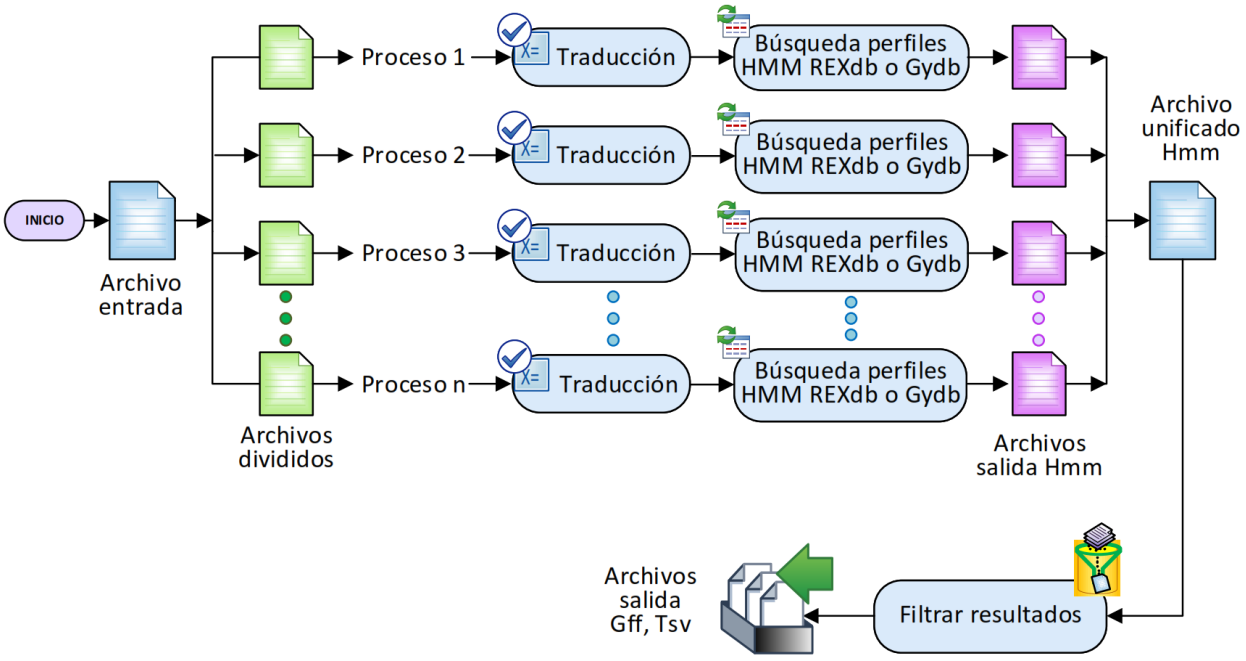

Figura 12. Flujo de trabajo de la estrategia en paralelo implementada en Arthur_LTRanalizer.

3.8.6. Archivos de salida.

Al finalizar el proceso de búsqueda y clasificación de retrotransposones LTR se crearon diferentes archivos de salida en formato tabular TSV y en formato GFF. Cada uno de estos archivos almacena información relevante de las secuencias de retrotransposones identificas, en el archivo de salida de clasificación de los elementos se presenta la

superfamilia y linaje representantivos de cada elemento identificado por Arthur_LTRanalizer.

Se extrajeron las secuencias de nucleótidos de los elementos identificados y secuencias de aminoácidos de los dominios proteicos de los elementos representativos identificados, se guardaron en archivos en formato Fasta, estas secuencias podrían ser usadas para posteriores análisis filogenéticos. El archivo de salida en formato TSV, presenta valores separados por tabuladoras, se incluyen columnas de la identificación del elemento, la superfamilia, clado, si es un elemento completo, hebra y dominios de retrotransposones encontrados, que puede ser analizado posteriormente usando criterios personalizados por el usuario.

3.9. Pruebas de funcionamiento.

En la fase inicial, el nuevo algoritmo se probó con secuencias pertenecientes a elementos no transponibles LTR de la base de datos de especies de plantas de instancias negativas de InpactorDB que incluían a diferentes tipos de ARN (Orozco-Arias et al., 2021, 2022), de estas bases se seleccionaron aleatoriamente las secuencias para la fase de prueba, los resultados obtenidos con estas secuencias fueron satisfactorios, se realizaron varias correcciones en las instrucciones de programación tanto en los archivos de salida como en los diferentes procesos para clasificar a las secuencias, una vez que el programa funcionaba adecuadamente se añadió la función para analizar archivos de entrada como genomas.

3.10. Disponibilidad y requisitos de la aplicación.

Una vez que se terminó la programación de la aplicación, se realizó un archivo ejecutable para facilitar la instalación en Linux, de esta forma el programa se puede correr a través del ejecutable o por medio del script con Python cuando todos los requisitos estén solventados.

- **Nombre del proyecto:** Arthur_LTRanalizer.
- **Licencia:** Acceso libre y gratuito.

- **Requisitos:** Se recomienda usar una computadora o máquina virtual con sistema operativo Linux, se debe tener instalado Python con una versión 3.10 o superior, instalar el paquete anaconda, configurar los canales de Bioconda e instalar los paquetes de Python: Biopython, HMMER y librerías de Python como Tkinter.
- **Repositorio:** El código en Python, la aplicación y el manual de instalación, se subieron al repositorio de github: https://github.com/Tatysb29/arthur_LTRanalizer. El acceso al script principal de Python se ubica dentro de la carpeta de descarga individual, el nombre del script es arthur_ltrAnalizer2023.py y se encuentra en: https://github.com/Tatysb29/arthur_LTRanalizer/blob/main/descargaIndividual/arthur_ltrAnalizer2023.py

3.11. Instalación.

Arthur_LTRanalizer se puede instalar de dos maneras, la primera opción es directamente desde el ejecutable y la segunda opción es corriendo el script de Python directamente.

3.11.1. Opción 1 en Linux a través del ejecutable.

- Descargar la carpeta “descargaConEjecutable”, esta carpeta contiene una subcarpeta Arthur_LTRAnalizer.
- En la terminal ingresar dentro de la carpeta Arthur_LTRAnalizer por medio del comando “cd”.
- En la terminal añadir una ruta a la variable de entorno de Linux hacia la carpeta Arthur_LTRAnalizer con el comando:
  export PATH="$HOME/ miRutaHacialaCarpetaArthur:$PATH"
- Correr el programa con el comando: ./Arthur_LTRAnalizer
- Si aparece el error de permiso denegado, se debe habilitar el archivo como ejecutable, dando click derecho en el archivo Arthur_LTRAnalizer- click en propiedades y habilitar archivo ejecutable

3.11.2. Opción 2 en Linux corriendo el script directamente.

- Descargar la carpeta “descargaIndividual”.

- Instalar el paquete de Anaconda.
- En la terminal crear un ambiente de trabajo con el comando: conda create env miAmbiente.
- Ingresar al ambiente de trabajo creado con el comando: conda activate miAmbiente.
- Dentro de este ambiente configurar los canales de bioconda con los siguientes comandos, en el mismo orden:
  conda config --add channels defaults
  conda config --add channels bioconda
  conda config --add channels conda-forge
  conda config --set channel_priority strict (Bioconda, 2022).
- Instalar las siguientes librerías:
  biopython con: conda install -c conda-forge biopython # https://anaconda.org/conda-forge/biopython
  xopen con: conda install -c bioconda xopen # https://anaconda.org/bioconda/xopen
  hmmer con: conda install -c bioconda hmmer # https://anaconda.org/bioconda/hmmer
- En la terminal, estando dentro del ambiente de trabajo elegido, ingresar dentro de la carpeta Arthur_LTRAnalizer por medio del comando cd.
- Se debe correr el programa con el comando: /arthur_ltrAnalizer2023.py

CAPÍTULO IV.

RESULTADOS.

La evaluación del rendimiento de los programas que clasifican a los elementos transponibles inicia con la categorización del genoma completo en cuatro partes, tomando como referencia el método de evaluación propuesto en la aplicación EDTA (Ou et al., 2019). De esta manera el genoma se dividió en secuencias objetivo (target) y secuencias no objetivo (no target), las secuencias de elementos transponibles LTR se etiquetaron como objetivo y todas las demás secuencias que no correspondieron a retrotransposones LTR se etiquetaron como no objetivo.

Al comparar los elementos retrotransponibles LTR de la librería de referencia con los elementos retrotransponibles LTR identificados por el algoritmo, se puede clasificarlos en verdaderos positivos, falsos negativos, verdaderos negativos y falsos positivos, que son provenientes de una matriz de confusión de clasificación binaria. La representación esquemática de la clasificación de secuencias para el análisis se presenta en la Figura 13 (Rodriguez & Makałowski, 2022).

Se reconoció a los verdaderos positivos como las secuencias de retrotransposones LTR que fueron identificadas por el algoritmo y que coincidieron con las secuencias de retrotransposones LTR de la librería de referencia. Los falsos negativos correspondieron a los elementos clasificados como no LTR por el algoritmo pero que si se encontraron catalogados como elementos retrotransponibles LTR en la librería de referencia. Mientras que los verdaderos negativos fueron los elementos identificados por el algoritmo como no LTR y que también se encontraron en la librería de referencia como elementos no LTR. Los falsos positivos son aquellos elementos que fueron clasificados por el algoritmo como retrotransposones LTR pero que no se encontraron catalogados en la librería de referencia como elementos retrotransponibles LTR. La representación esquemática de la forma de selección de los componentes de la matriz de confusión se indica en la Figura 13 (Rodriguez & Makałowski, 2022).

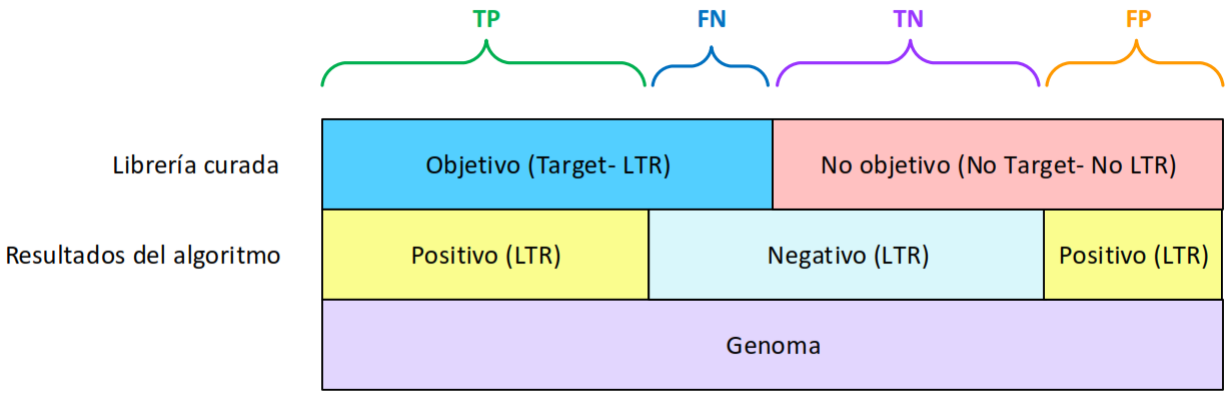

Figura 13. Representación esquemática de la clasificación de secuencias, TP (verdaderos positivos), FN (falsos negativos), TN (verdaderos negativos) y FP (falsos positivos), adaptado de (Ou et al., 2019).

La validación de la nueva herramienta Arthur_LTRanalizer se realizó tomando como referencia el estudio conducido por Lerat (Valencia & Girgis, 2019), en el que se probaron varias herramientas para validar su investigación. Se comparó el rendimiento del nuevo algoritmo con tres herramientas computacionales que identifican y clasifican elementos transponibles como son: LTRretriever (Ou & Jiang, 2023), LTRclassifier (Monat, Tando, Tranchant-Dubreuil, & Sabot, 2016) y TEsorter (R. G. Zhang et al., 2023).

LTRretriever usa la base de referencia Dfam basada en perfiles de Modelos Ocultos de Markov de elementos repetitivos, mientras TEsorter usa varios perfiles de referencia entre ellos REXdb viridiplantae, REXdb metazoa, REXdb-TIR, Yuan_and_Wessler-PNA-TIR y GyDB. Se escogió a estas herramientas para el análisis, ya que permitieron realizar comparaciones directas entre ellas y Arthur_LTRanalizer.

No se escogió a otros programas porque no era posible realizar un análisis comparativo directo; además, la instalación de varias herramientas fue complicada con algunas dependencias que no se pudieron instalar como TERL (https://github.com/muriloHoracio/TERL) y DeepTE (https://github.com/LiLabAtVT/DeepTE), otras herramientas corrieron adecuadamente sin embargo no generaron ningún resultado debido a que se presentaron errores en el análisis, ya que se requería realizar ciertas configuraciones que no se incluían en el

manual de usuario, como es el caso de RepeatModeler (https://github.com/Dfam-consortium/RepeatModeler).

En el caso de Inpactor (Orozco Arias et al., 2018) se necesita características estructurales de los elementos a clasificar y no soportaba secuencias como única entrada, adicionalmente otros programas como: TEclass (https://www.bioinformatics.uni-muenster.de/tools/teclass/index.hbi?), PASTEC como un módulo de REPET (https://urgi.versailles.inra.fr/Tools/PASTEClassifier) y REPCLASS (https://uta-ir.tdl.org/uta-ir/handle/10106/455), solamente proveen clasificaciones confiables a nivel de orden, por lo que no fue posible realizar una comparación objetiva.

Las tres herramientas se evaluaron usando una computadora personal Intel Core i7 de decima generación, con sistema operativo Windows 10 de 64 bits, memoria RAM de 32GB, en la cual corre la máquina virtual de Ubuntu (v20.0). Para la evaluación de las herramientas con la librerías curadas de secuencias de elementos transponibles se empleó el módulo Annotate_TE de LTR_retriever y se analizó con la base de referencia de Dfam (Storer, Hubley, Rosen, Wheeler, & Smit, 2021).

Cada archivo de salida de las distintas herramientas se comparó con la librería curada de elementos transponibles del arroz; tomando en cuenta que el análisis correspondió a una clasificación binaria se obtuvieron los componentes de la matriz de confusión como son: verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, a partir de estos valores se determinaron medidas de rendimiento tales como: sensibilidad, especificidad, exactitud (accuracy), precisión, puntaje F1 y Coeficiente de Correlación de Matthews (CCM), usando las ecuaciones (I-XIII) presentadas en el capítulo I, además se calculó el porcentaje de clados identificados por las herramientas y el tiempo de ejecución de cada algoritmo. Los valores obtenidos con la base de datos curada de elementos transponibles del arroz, para las cuatro herramientas, se muestra en la Tabla 8.

Tabla 8. Rendimiento de herramientas que identifican retrotransposones LTR en la librería de TE del Arroz.

Librería TE	Programa	Sensibilidad	Especificidad	Exactitud	Precisión	F1	MCC ⁑	Clados (%) ∗	Tiempo (min)
Arroz	ArthurLTRanalizer (REXdb)	0.830	0.987	0.908	0.984	0.900	0.827	83.937	2.50
	ArthurLTRanalizer (GyDB)	0.781	0.984	0.883	0.980	0.869	0.781	82.652	3.00
	LTRclassifier	0.682	0.863	0.773	0.833	0.750	0.554	NA	70.00
	LTR_retriever (Annotate_TE) †	0.827	0.954	0.890	0.947	0.883	0.787	NA	2.08
	TEsorter (REXdb)	0.840	0.998	0.919	0.998	0.912	0.849	84.951	2.25
	TEsorter (GyDB)	0.802	0.994	0.898	0.992	0.887	0.811	83.904	2.50

∗ Porcentaje de elementos que fueron asignados a clados.
† El módulo Annotate_TE del programa LTR_retriever usó la base de referencia de Dfam.
⁑ MCC: Coeficiente de correlación de Matthews.
NA: No aplica.
TE: Elementos transponibles.

Al evaluar la sensibilidad de las cuatro herramientas computacionales se encontró que el programa que presentó mayor sensibilidad del 84.00 % es TEsorter con la base de referencia REXdb, en segundo lugar se ubica la nueva herramienta Arthur_LTRanalizer evaluado con la misma librería de REXdb con un 83.00 %, seguido de LTR_retriever con un 82.70 %, a continuación se ubica TEsorter con la base de referencia GyDB con un 80.20 %, seguido de Arthur_LTRanalizer con la librería GyDB que presenta un 78.10 % y en último lugar se ubica LTRclassifier con un 78.00 %, esto se puede apreciar en la Figura 14.

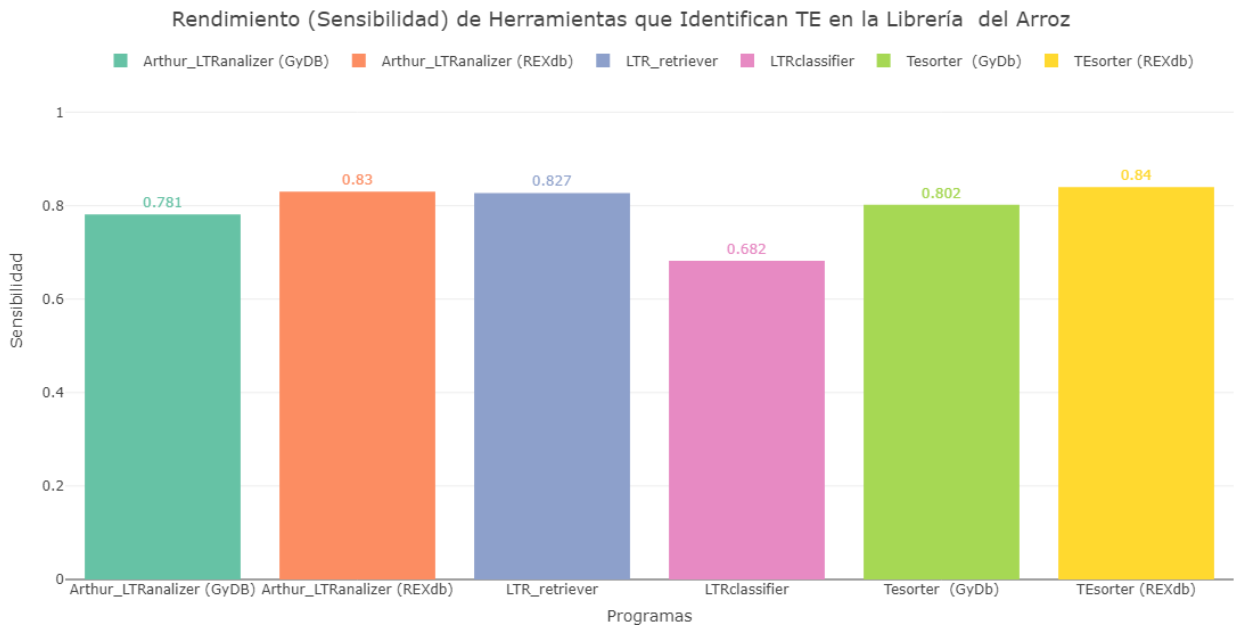

Figura 14. Evaluación de la sensibilidad de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

Se determinó el rendimiento de los cuatro algoritmos teniendo en cuenta la especificidad, los resultados se muestran en la Figura 15, la herramienta que presenta mayor especificidad es TEsorter con la librería REXdb con un 99.80 %, en segundo lugar se ubica la nueva herramienta Arthur_LTRanalizer analizada con la librería REXdb con un 98.70 %, en tercer puesto se encuentra TEsorter con la librería GyDB con un 99.40%, en cuarto lugar se ubica Arthur_LTRanalizer con GyDB con un 98.40 %, a continuación se encuentra LTR_retriever con un 95.40 % y en último lugar estuvo LTRclassifier con un 86.30 %.

Se evaluó la exactitud de los cuatro programas que clasifican elementos retrotransponibles LTR usando la base de datos de elementos transponibles del arroz, los resultados se indican en la Figura 16. En primer lugar se halla TEsorter con REXdb con un 91.90 %, en segundo puesto se ubica Arthur_LTRanalizer con la librería REXdb con un 90.80 %, en tercer lugar se encuentra TEsorter con GyDB con un 89.80 %, seguido de LTR_retriever analizado con la librería de Dfam con un 89.00 %, en quinto puesto se ubica Arthur_LTRanalizer con la librería GyDB con un 88.30 % y en último lugar LTRclassifier con un 77.30 %.

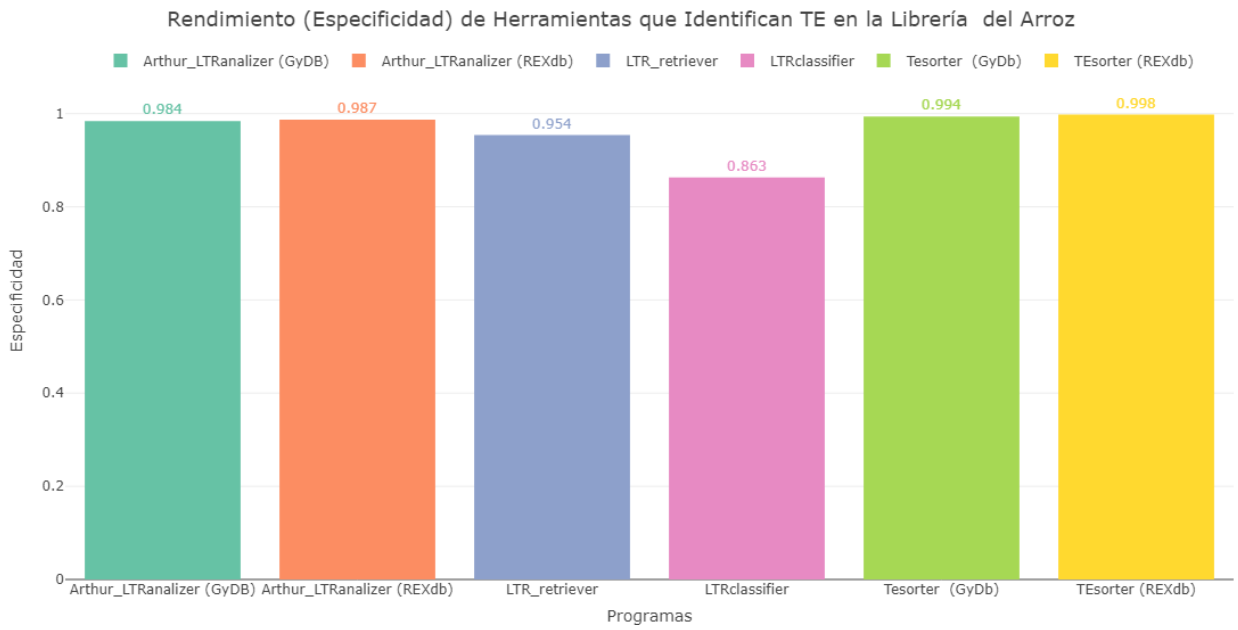

Figura 15. Evaluación de la especificidad de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

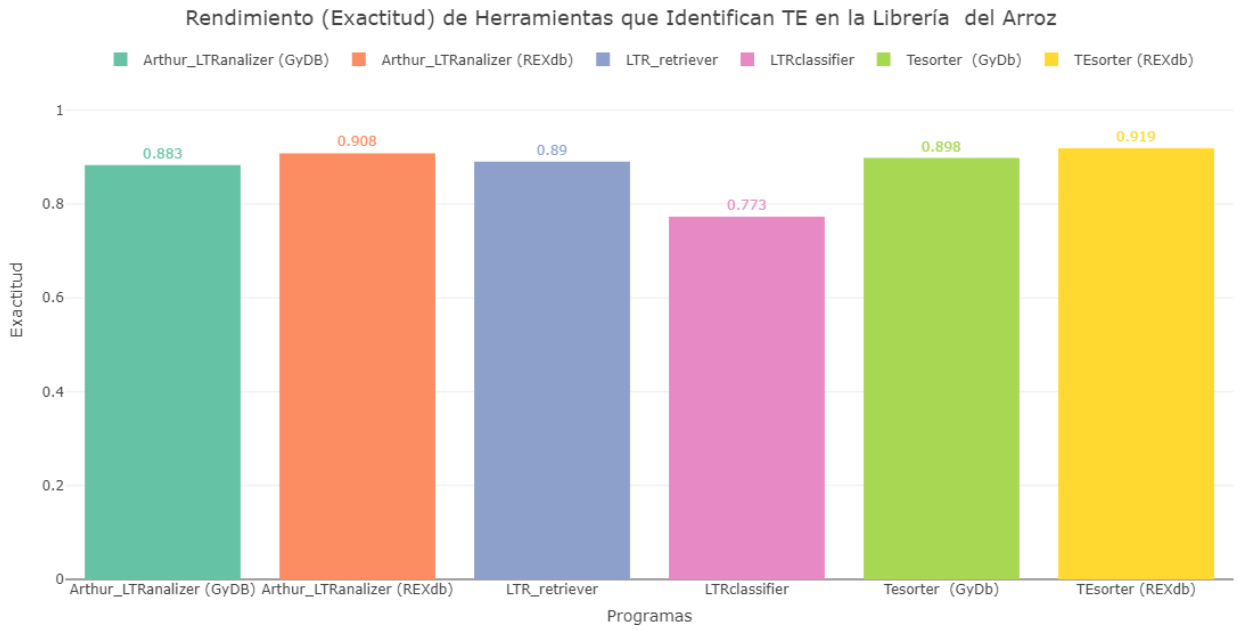

Figura 16. Evaluación de la exactitud de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

Al evaluar la precisión de las herramientas que clasifican elementos transponibles LTR, con la base de referencia curada de elementos transponibles del arroz, se encontró que el programa que presenta mayor precisión es TEsorter con la librería REXdb con un 99.80 %, en segundo lugar se halla TEsorter con GyDB con 99.20 %, en tercer puesto se ubica la nueva herramienta Arthur_LTRanalizer con REXdb con un 98.40 %, en cuarto lugar se encuentra Arthur_LTRanalizer con la librería GyDB con un 98.00 %, a continuación se ubica LTR_retriever con un 94.70 % y la herramienta que tiene la menor precisión es LTRclassifier, esto se puede apreciar en la Figura 17.

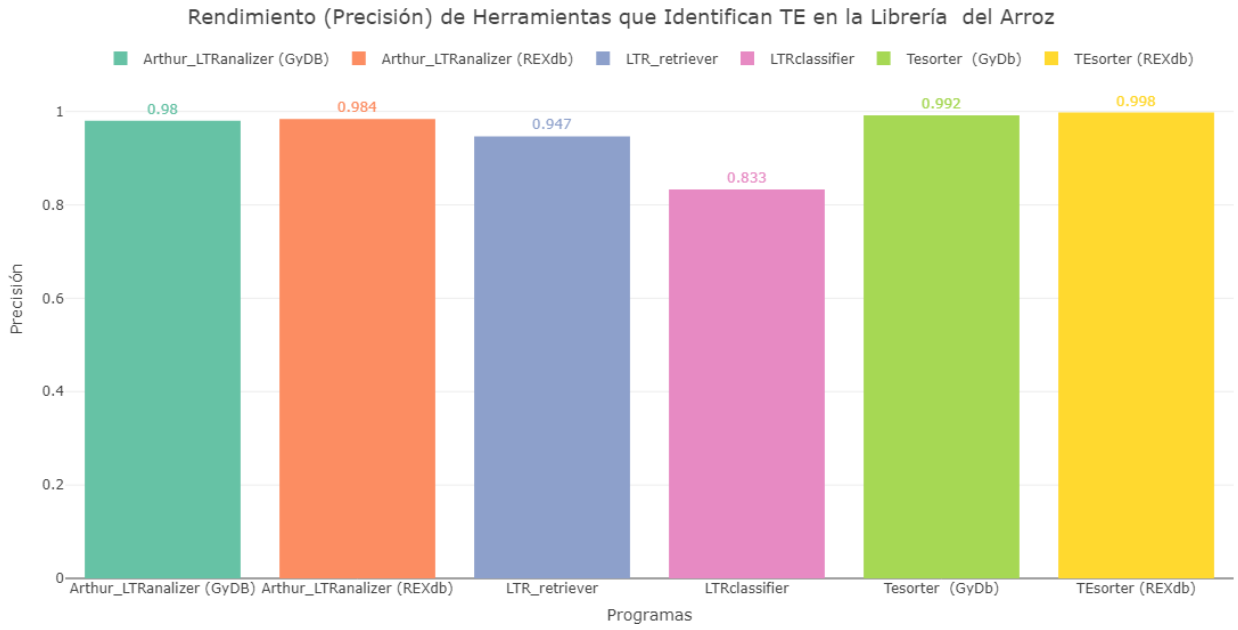

Figura 17. Evaluación de la precisión de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

Se determinó el puntaje F1 con las secuencias de la librería de elementos transponibles del arroz, se obtuvo los resultados que se indican en la Figura 18, la herramienta que tiene el mayor puntaje F1 fue TEsorter con REXdb con un 91.20 %, en segundo lugar se ubica Arthur_LTRanalizer con un 90.00 %, en tercer lugar TEsorter con GyDB con un 88.70 %, en cuarto puesto se ubica Arthur_LTRanalizer con GyDB con un 86.90 %, seguido por LTR_retriever con un 88.30 % y en último lugar LTRclassifier.

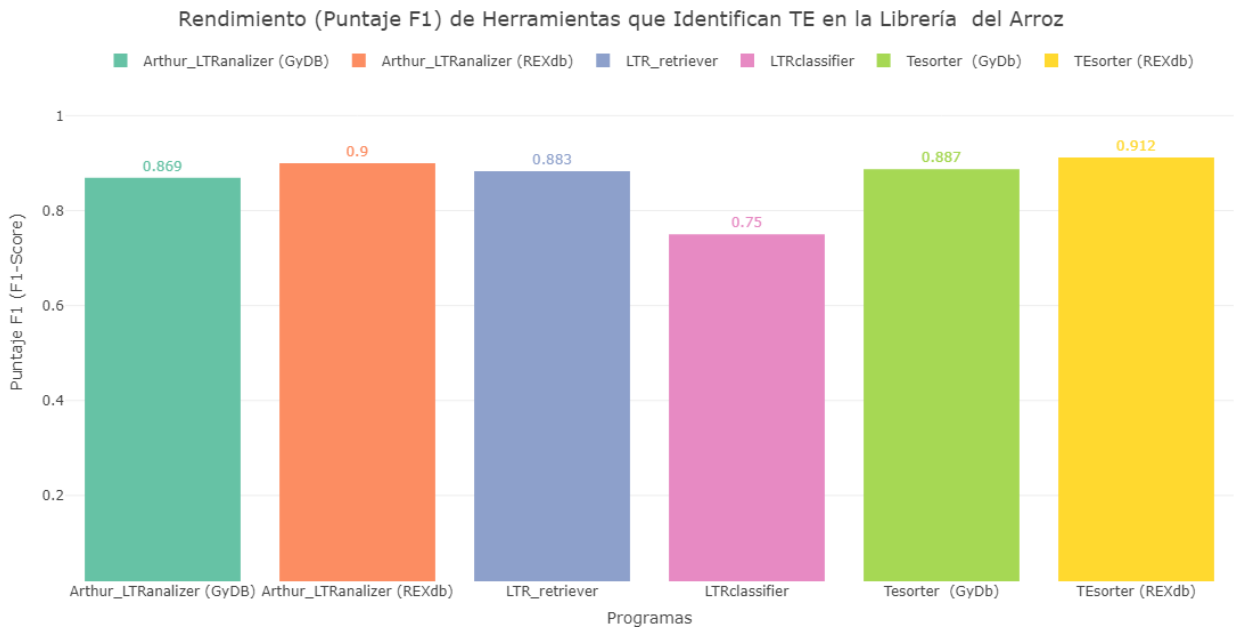

Figura 18. Evaluación del puntaje F1 de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

También se evaluó el Coeficiente de Correlación de Matthews (MCC) con la base de datos referencial compuesta de secuencias de elementos transponibles del arroz, el programa que presenta un mayor coeficiente es TEsorter con REXdb con 84.90 %, en segundo lugar se ubica Arthur_LTRanalizer con REXdb con 82.70 %, en tercer lugar se encuentra TEsorter GyDB con un 81.10 %, seguido de LTR_retriever con 78.70 % y en último puesto LTRclassifier con 55.40 %, el grafico de barras se puede apreciar en la Figura 19.

Además, se calculó el porcentaje de clados identificados por las distintas herramientas usando la base de datos de referencia de elementos transponibles del arroz, el programa que es capaz de identificar un mayor porcentaje de clados es TEsorter con la librería REXdb con un 84.95 %, en segundo puesto se encuentra Arthur_LTRanalizer con REXdb con 83.94 %, en tercer lugar se ubica TEsorter con la librería GyDB con 83.90 %, en cuarto lugar se halla Arthur_LTRanalizer con GyDB con 82.65 %, las herramientas LTR_retriever y LTRclassifier no clasificaron a los elementos en linajes, se puede apreciar en la Figura 20.

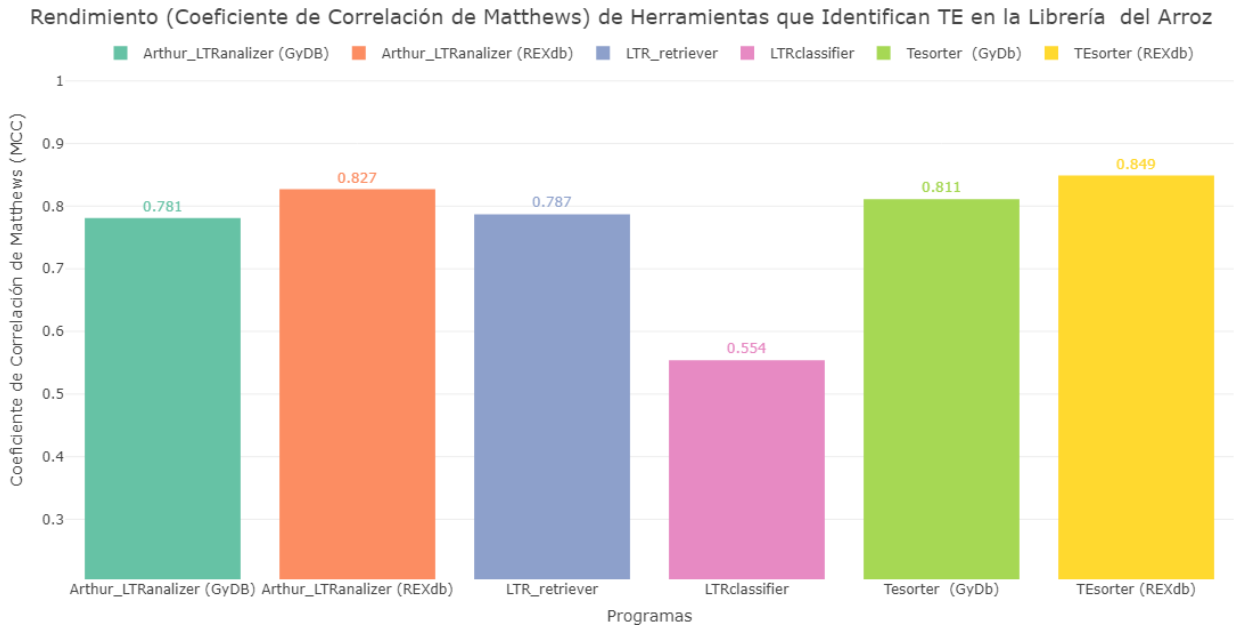

Figura 19. Evaluación del Coeficiente de Correlación de Matthews (MCC) de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

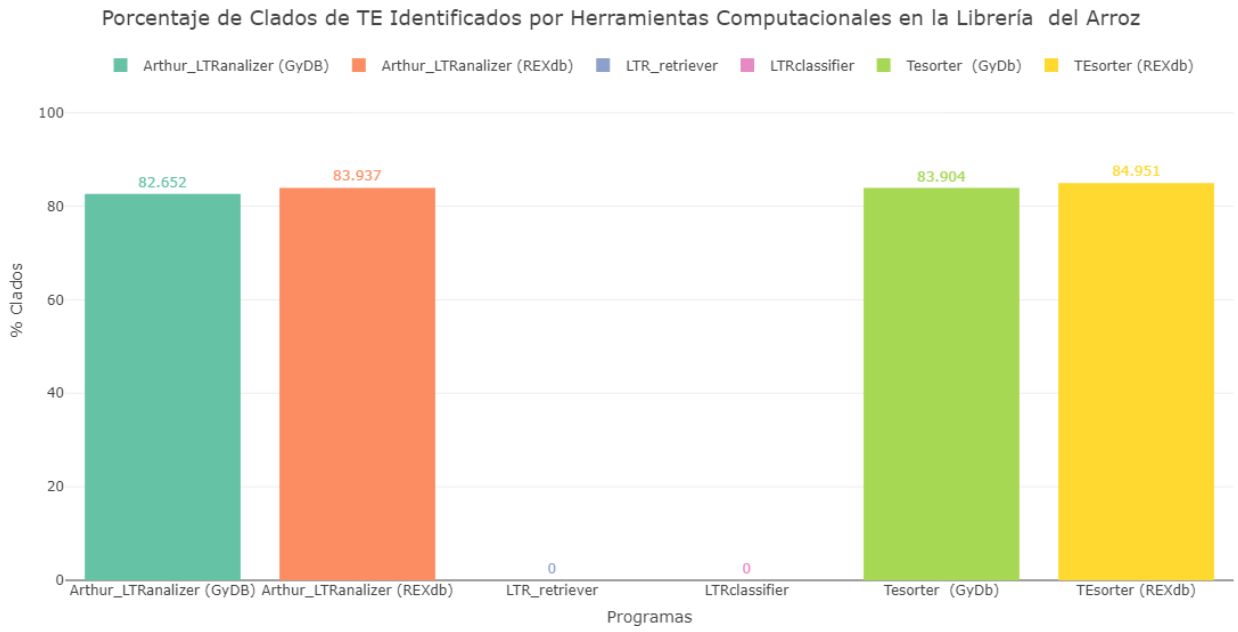

Figura 20. Porcentaje de clados de retrotrasposones LTR identificados por herramientas computacionales en la librería de TE del arroz.

Se determinó el tiempo de ejecución de los cuatro programas con secuencias de la base curada con elementos transponibles del arroz que se presenta en la Figura 21, la

herramienta que presentó el menor tiempo de ejecución fue LTR_retriever con 2.08 minutos, le sigue TEsorter con la librería REXdb con 2.25 minutos, seguido de TEsorter con GyDB con 2.50 minutos, en cuarto lugar se ubica Arthur_LTRanalizer con REXdb con 2.50 minutos, seguido de Arthur_LTRanalizer con GyDB con 3.00 minutos y el programa que tarda más tiempo en ejecutar las instrucciones es LTRclassifier con 70 minutos.

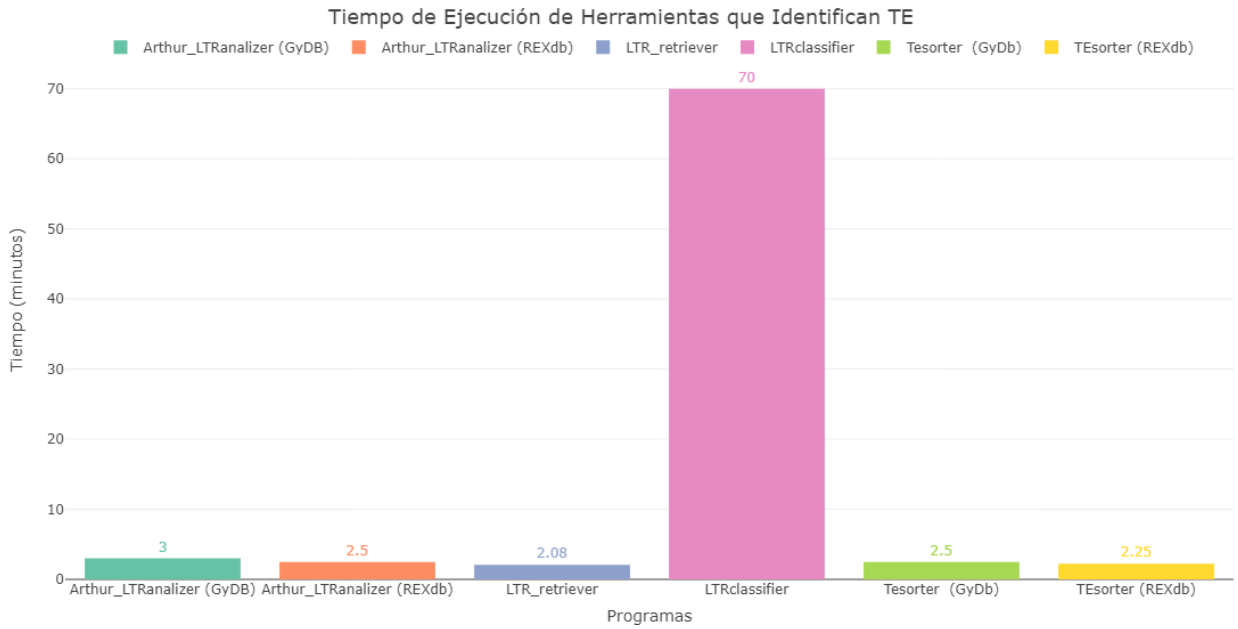

Figura 21. Evaluación del tiempo de ejecución de herramientas que identifican retrotransposones LTR en la librería de TE del arroz.

Los valores obtenidos con la base de datos curada de elementos transponibles del maíz, para las cuatro herramientas, se muestra en la Tabla 9. Al evaluar la sensibilidad de las cuatro herramientas con la base de datos de referencia de elementos transponibles del maíz, se determinó que TEsorter con la librería REXdb obtuvo el mayor porcentaje de sensibilidad con un 88.10 %, seguido de TEsorter con la librería GyDB con 86.90 %, en tercer lugar se ubica Arthur_LTRanalizer con un 86.70 %, en cuarto lugar se encuentra Arthur_LTRanalizer con GyDB con 84.80 %, seguido de LTR_retriever con 84.00 % y la herramienta que presenta la menor sensibilidad fue LTRclassifier, estos valores se muestran en la Figura 22.

Tabla 9. Rendimiento de herramientas que identifican retrotransposones LTR en la librería de TE del Maíz.

Librería TE	Programa	Sensibilidad	Especificidad	Exactitud	Precisión	F1	MCC ⁑	Clados (%) ∗	Tiempo (min)
Maíz	ArthurLTRanalizer (REXdb)	0.867	0.982	0.925	0.980	0.920	0.855	88.654	2.05
	ArthurLTRanalizer (GyDB)	0.848	0.975	0.911	0.971	0.905	0.829	87.237	2.25
	LTRclassifier	0.667	0.911	0.789	0.882	0.760	0.596	NA	65.00
	LTR_retriever (Annotate_TE; Dfam) †	0.840	0.919	0.880	0.912	0.875	0.761	NA	2.00
	TEsorter (REXdb)	0.881	0.989	0.935	0.988	0.931	0.875	91.850	2.02
	TEsorter (GyDB)	0.869	0.990	0.930	0.989	0.925	0.866	90.150	2.13

∗ Porcentaje de elementos que fueron asignados a clados.
† El módulo Annotate_TE del programa LTR_retriever usó la base de referencia de Pfam.
⁑ MCC: Coeficiente de correlación de Matthews.
NA: No aplica.
TE: Elementos transponibles.

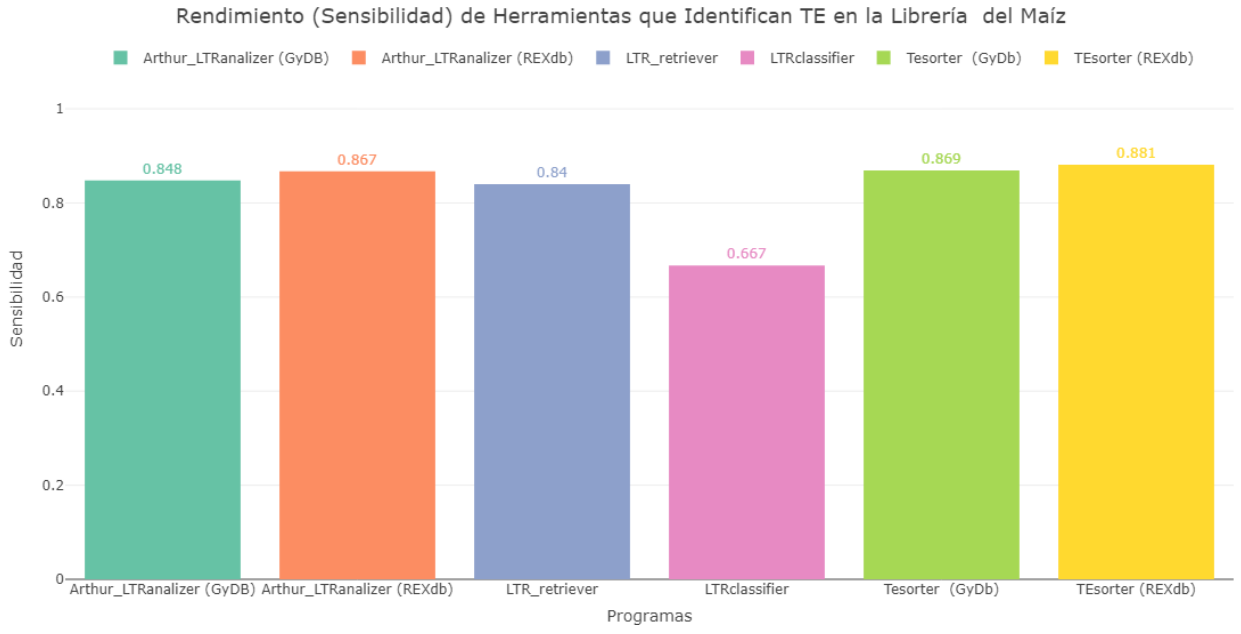

Figura 22. Evaluación de la sensibilidad de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

Se determinó la especificidad de los programas que clasifican elementos retrotransponibles LTR usando la base de referencia de elementos transponibles del Maíz, en la Figura 23 se puede apreciar que TEsorter con la librería GyDB tiene el mayor porcentaje de especificidad con un 99.00 %, en segundo lugar se ubica TEsorter con la librería REXdb con un 98.90 %, en tercer puesto se encuentra Arthur_LTRanalizer con REXdb con un 98.20 %, en cuarto lugar Arthur_LTRanalizer con GyDB con 97.50%, seguido de LTR_retriever con un 91.90 % y el programa que tiene la menor especificidad es LTRclassifier con un 91.10 %.

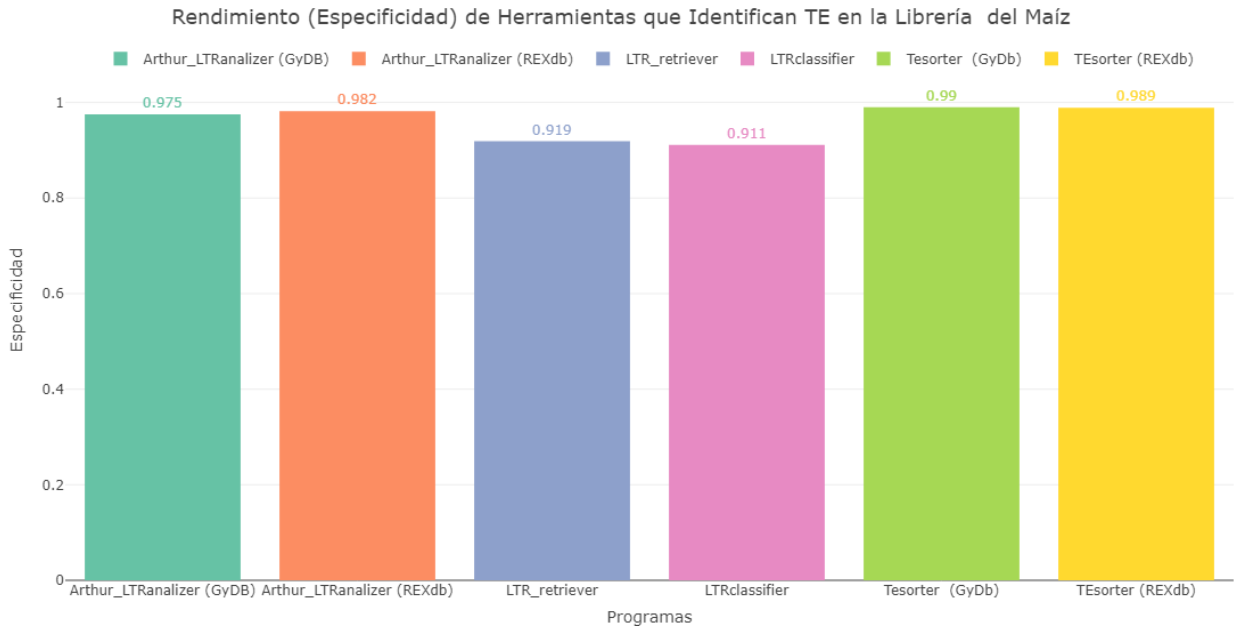

Figura 23. Porcentaje de clados de retrotrasposones LTR identificados por herramientas computacionales en la librería de TE del arroz.

A las cuatro herramientas se evaluó la exactitud usando la base de elementos transponibles del maíz, los valores obtenidos se indican en la Figura 24, TEsorter con la librería REXdb fue la herramienta con mayor exactitud del 93.50 %, seguido por TEsorter con GyDB con 93.00 %, en tercer lugar se ubicó la nueva herramienta Arthur_LTRanalizer con REXdb con un 92.50 %, le sigue Arthur_LTRanalizer con un 91.10 %, seguido por LTR_retriever con un 88.00 % y el programa que tiene la menor exactitud es LTRclassifier con un 78.90 %.

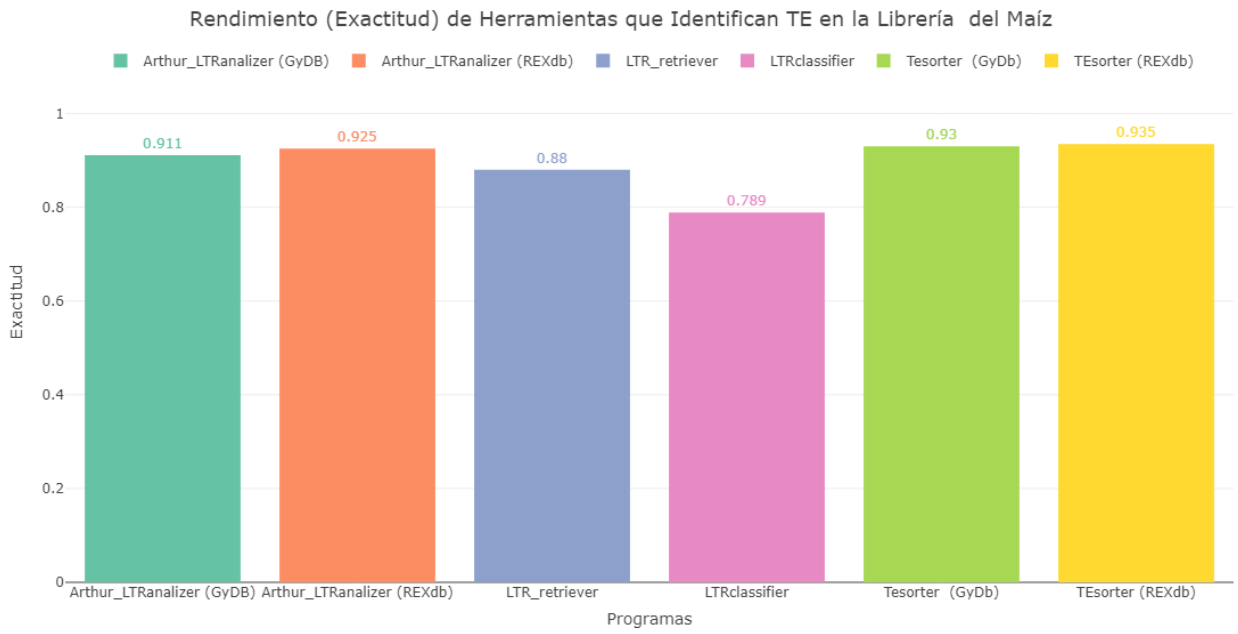

Figura 24. Evaluación de la exactitud de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

Se determinó la precisión de las herramientas con la base de elementos transponibles del Maíz, el programa con mayor precisión fue TEsorter con GyDB con un 98.90 %, seguido de TEsorter con REXdb con 98.80 %, en tercer lugar se ubicó Arthur_LTRanalizer con REXdb con 98.00 %, en cuarto puesto estuvo Arthur_LTRanalizer con 97.10 %, seguido de LTR_retriever con un 91.20 % y el programa con la menor precisión fue LTRclassifier con un 88.20%, se puede apreciar en la Figura 25.

También se calculó el puntaje F1 con la base de referencia de elementos transponibles del Maíz, se encontró que la herramienta con mayor puntaje fue TEsorter con REXdb con un 93.10 %, seguido de TEsorter GyDB con un 92.50 %, en tercer puesto estuvo Arthur_LTRanalizer con la librería de REXdb, le sigue Arthur_LTRanalizer con GyDB, seguido de LTR_retriever con un 87.50% y la herramienta con el menor puntaje F1 fue LTRclassifier, los valores se muestran en la figura Figura 26.

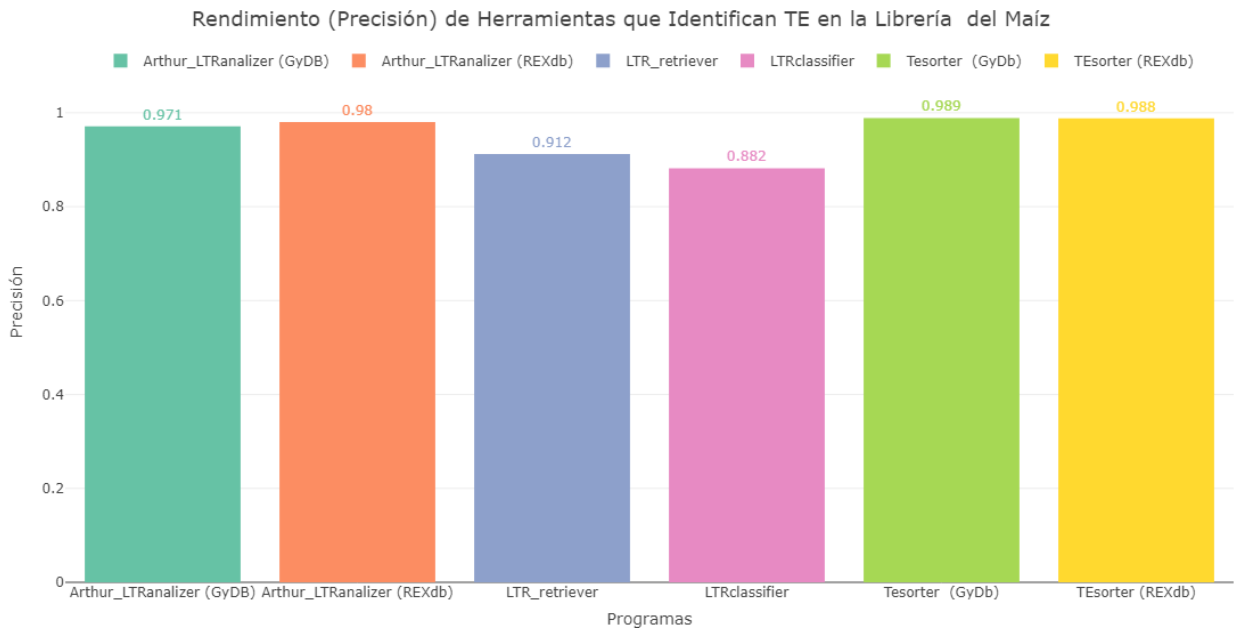

Figura 25. Evaluación de la precisión de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

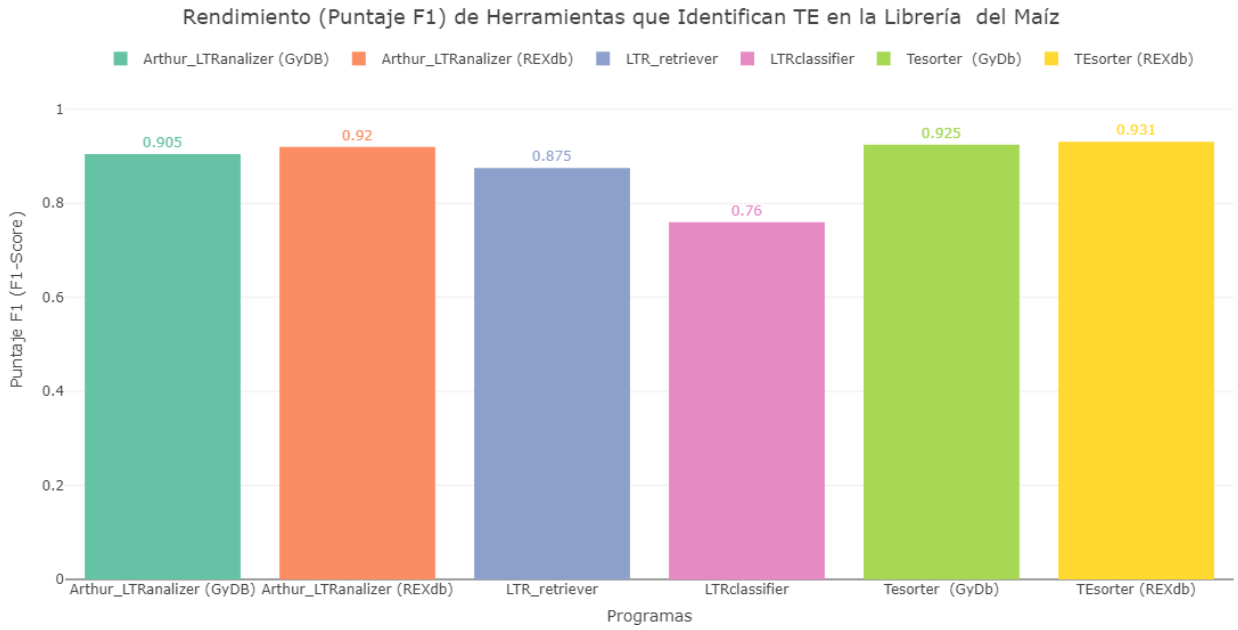

Figura 26. Evaluación del puntaje F1 de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

Se evaluó el Coeficiente de Correlación de Matthews con la base de elementos transponibles del Maíz, se encontró que TEsorter con REXdb fue la herramienta con mayor coeficiente de 87.50 %, le siguió TEsorter con GyDB con un 86.60 %, en tercer

lugar se ubicó Arthur_LTRanalizer con REXdb con un 85.50 %, en cuarto puesto estuvo Arthur_LTRanalizer con GyDB con un 85.50 %, seguido de LTR_retriever con un 76.10 % y LTRclassifier fue la herramienta con menor coeficiente de 59.60 %, estos valores se presentan en la Figura 27.

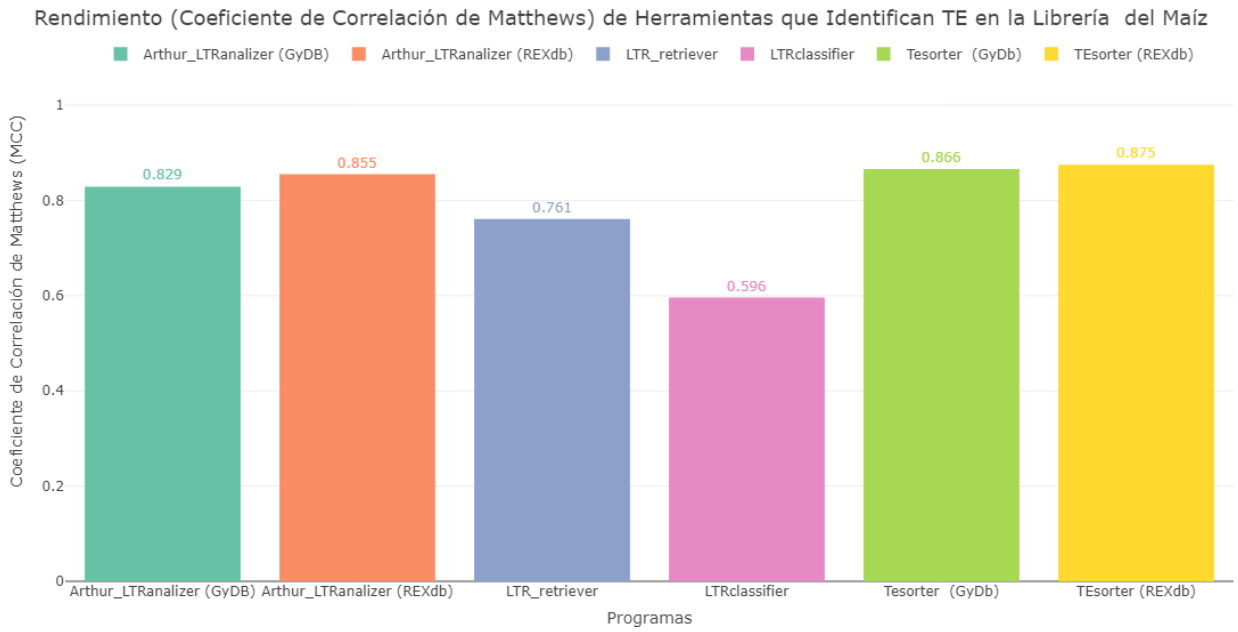

Figura 27. Evaluación del Coeficiente de Correlación de Matthews de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

Adicionalmente, se determinó el porcentaje de clados de retrotransposones LTR identificados por las cuatro herramientas computacionales en la base de referencia de elementos transponibles del maíz, los resultados se aprecian en la Figura 28, TEsorter con REXdb fue el programa que encontró un mayor porcentaje de clados del 91.85 %, seguido de TEsorter con GyDB con un 90.15 %, en tercer puesto estuvo la nueva herramienta Arthur_LTRanalizer con REXdb con un 88.65 % y en tercer lugar se ubicó Arthur_LTRanalizer con GyDB con un 87.24 %, LTRclassifier y LTR_retriever no clasificaron a los retrotransposones LTR a nivel de su linaje.

Se midió el tiempo de ejecución de las herramientas, los resultados se muestran en la Figura 29, la herramienta más rápida fue LTR_retriever que tardo 2 minutos en analizar la base de elementos transponibles del maíz, seguida de TEsorter con REXdb con 2.02 minutos, en tercer lugar se encontró Arthur_LTRanalizer con la librería REXdb con 2.05

minutos, le siguió TEsorter con GyDB con 2.13 minutos, seguido de Arthur_LTRanalizer con GyDB con 2.25 minutos y la herramienta que tardo más tiempo fue LTRclassifier con 60 minutos.

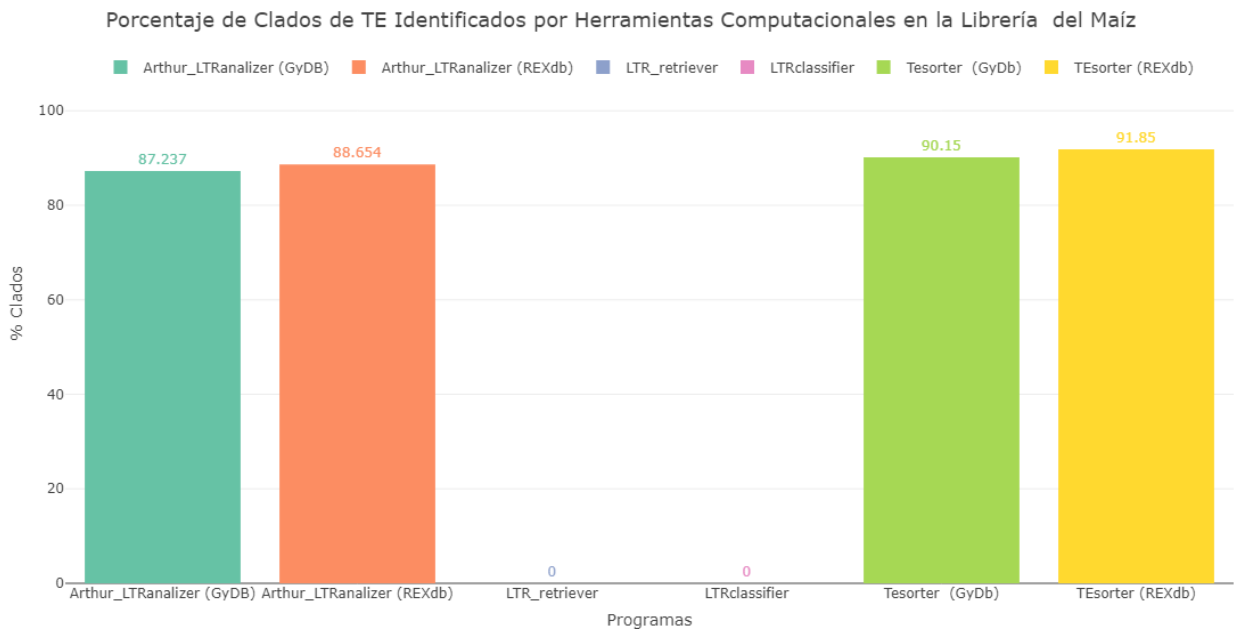

Figura 28. Porcentaje de clados de retrotrasposones LTR identificados por herramientas computacionales en la librería de TE del maíz.

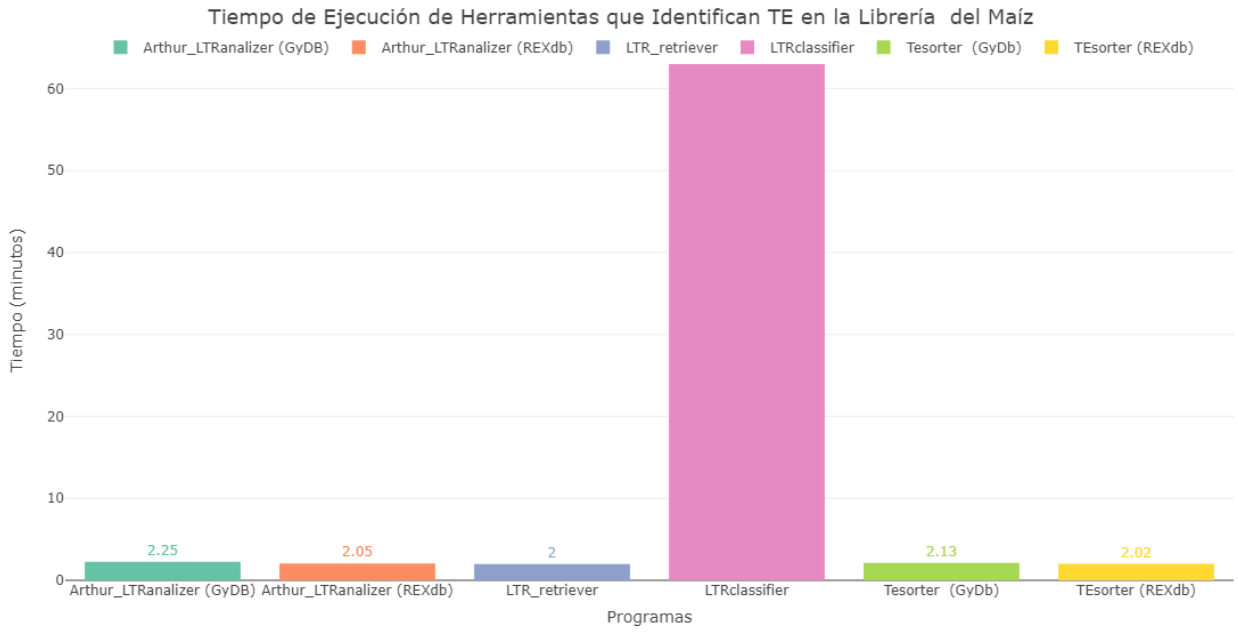

Figura 29. Evaluación del tiempo de ejecución de herramientas que identifican retrotransposones LTR en la librería de TE del maíz.

La validación de Arthur_LTRanalizer, para identificar y clasificar elementos retrotransponibles de secuencias genómicas, se hizo usando la base de datos de secuencias nucleotídicas del arroz, el rendimiento del algoritmo se comparó con las siguientes herramientas: LTRretriever y TEsorter con la base referencial REXdb. Los valores obtenidos con las secuencias genómicas del arroz, para las tres herramientas, se muestra en la Tabla 10.

Tabla 10. Rendimiento de herramientas que identifican retrotransposones LTR en secuencias genómicas del Arroz

Librería TE	Programa	Sensibilidad	Especificidad	Exactitud	Precisión	F1	MCC ‡	Clados (%) *	Tiempo (min)
Arroz	Arthur_LTRanalizer (REXdb)	0.819	0.979	0.899	0.975	0.890	0.808	83.307	30.50
	Arthur_LTRanalizer (GyDB)	0.797	0.977	0.887	0.972	0.876	0.787	82.275	34.00
	LTR_retriever †	0.807	0.964	0.885	0.956	0.875	0.779	NA	17.15
	TEsorter (REXdb)	0.826	0.993	0.910	0.992	0.901	0.831	84.852	25.25

Se determinó la sensibilidad de las tres herramientas con las secuencias genómicas del arroz, los resultados se aprecian en la Figura 30, TEsorter con REXdb fue el programa más sensible con un 82.60 %, en segundo lugar se ubicó la Arthur_LTRanalizer con la librería REXdb con un 81.90 %, a continuación estuvo LTR_retriever con un 80.70 % y en cuarto lugar se ubicó Arthur_LTRanalizer con GyDB con un 79.70 %.

El programa que presentó mayor especificidad al evaluarlo con secuencias genómicas del arroz fue TEsorter con REXdb con un 99.30 %, en segundo lugar se ubicó Arthur_LTRanalizer con la librería REXdb con un 97.90 %, en tercer puesto estuvo Arthur_LTRanalizer GyDB con un 97.70 % y la herramienta que presento menor especificidad fue LTR_retriever con un 96.40 %, como se puede apreciar en la Figura 31.

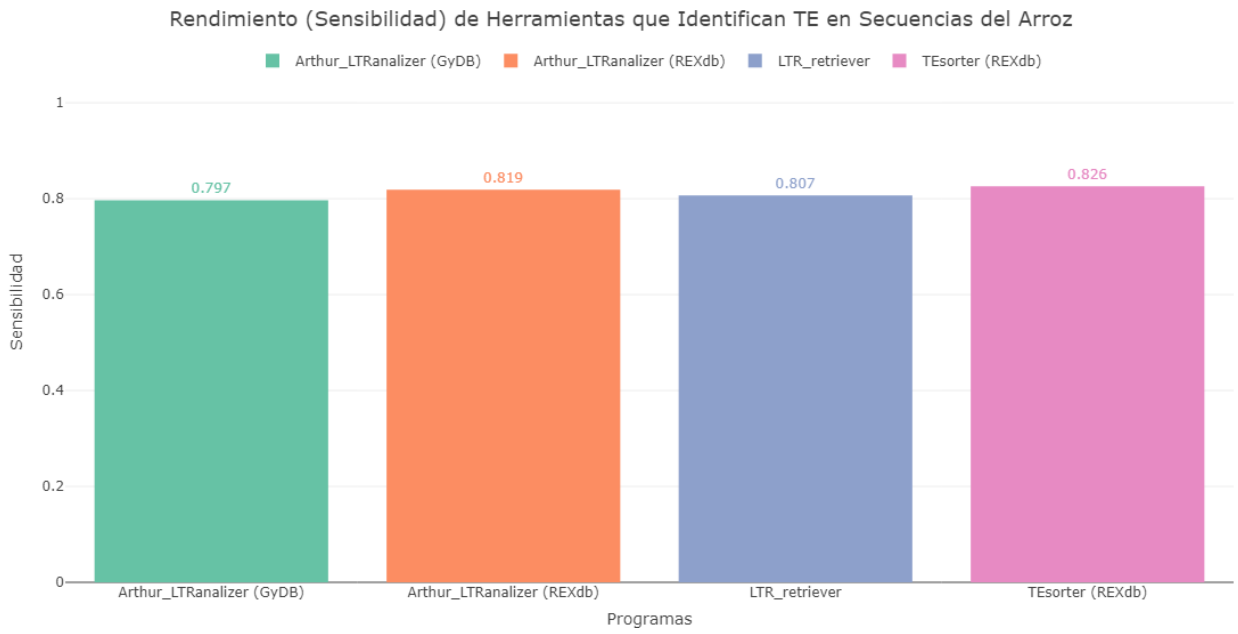

Figura 30. Evaluación de la sensibilidad de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

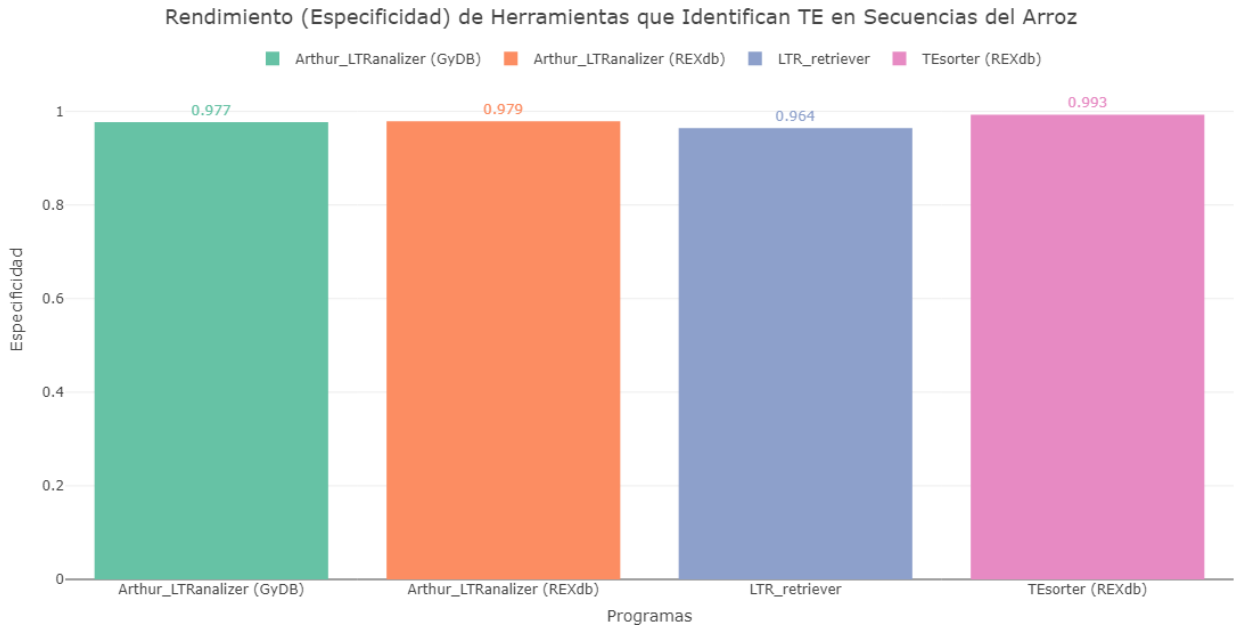

Figura 31. Evaluación de la especificidad de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

La herramienta que tuvo mayor exactitud al evaluarla con secuencias genómicas del arroz fue TEsorter con REXdb con un 91.00 %, en segundo puesto estuvo Arthur_LTRanalizer con REXdb con un 89.90%, en tercer lugar, se encontró Arthur_LTRanalizer con GyDB con un 88.70 % y la herramienta con menor exactitud fue LTR_retriever con un 88.50 %, como se aprecia en la Figura 32.

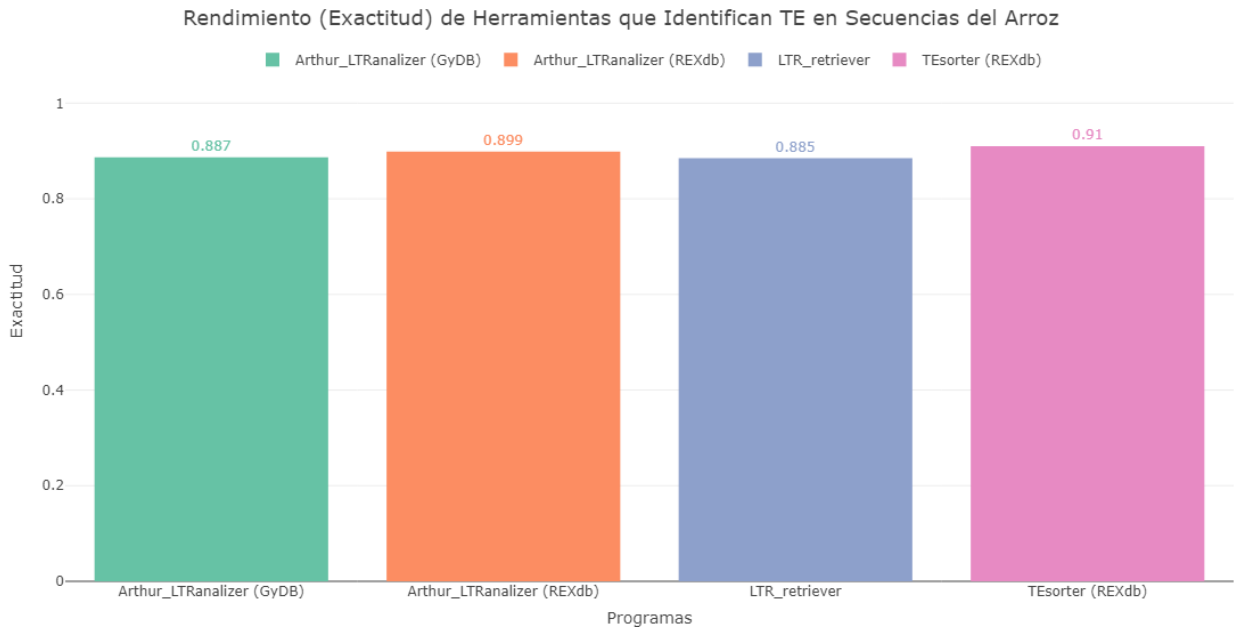

Figura 32. Evaluación de la exactitud de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

Se evaluó la precisión de las tres herramientas, los resultados se indican en la Figura 33, TEsorter fue el programa que presento mayor precisión al analizarlo con secuencias genómicas del arroz con un 99.20 %, en segundo lugar, estuvo Arthur_LTRanalizer con REXdb con un 97.50 %, en tercer puesto se encontró Arthur_LTRanalizer con GyDB con un 97.20 % y el programa menos preciso fue LTR_retriever con un 95.60 %.

Se determinó el puntaje F1, los valores se pueden observar en la Figura 34, TEsorter con REXdb constituyo la herramienta que presento un mayor puntaje del 90.10 %, en segundo lugar, se encontró Arthur_LTRanalizer con REXdb con un 89.00 %, seguido de Arthur_LTRanalizer con GyDB con un 87.60 % y la herramienta con menor puntaje fue LTR_retriever con un 87.50 %.

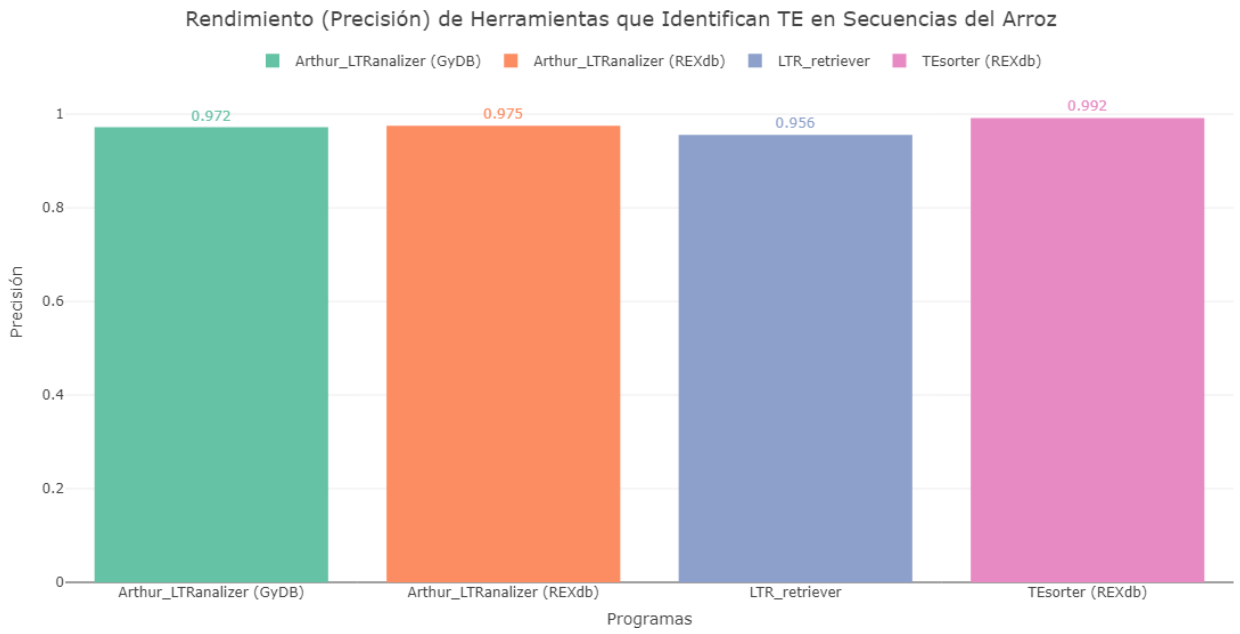

Figura 33. Evaluación de la precisión de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

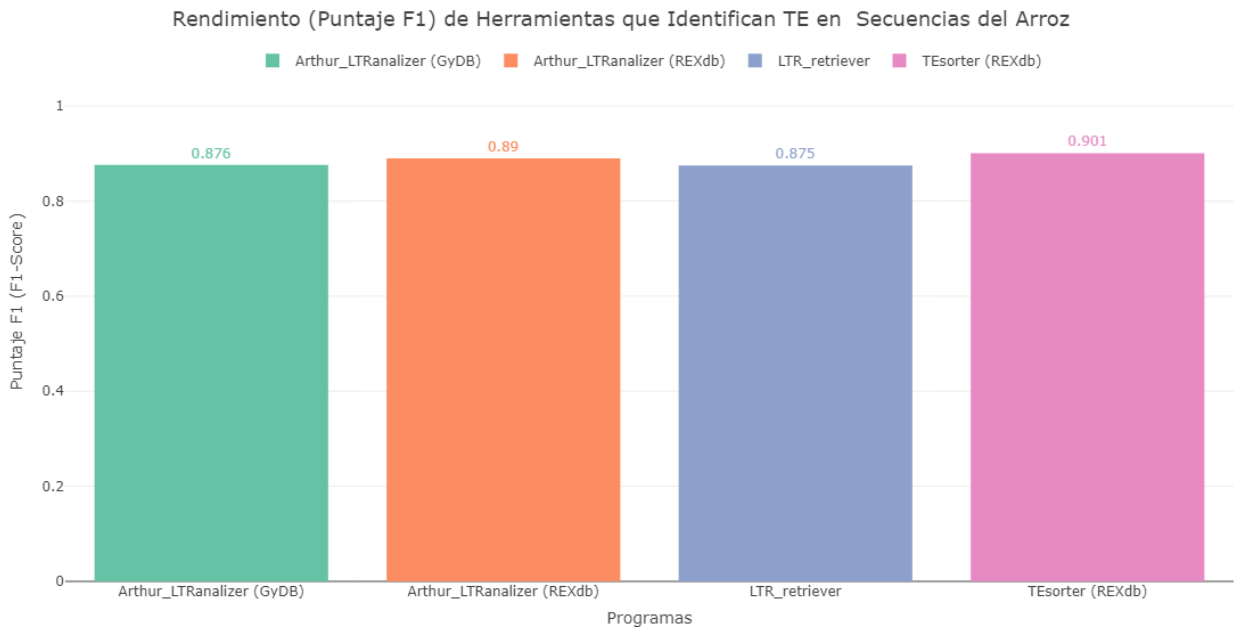

Figura 34. Evaluación del puntaje F1 de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

Se evaluó el Coeficiente de Correlación de Matthews a las tres herramientas utilizando secuencias genómicas del arroz, TEsorter con REXdb presentó el mayor coeficiente con

un 83.10 %, en segundo lugar, estuvo Arthur_LTRanalizer con REXdb con un 80.80 %, seguido de Arthur_LTRanalizer con GyDB con un 78.70 % y el programa que presentó menor coeficiente fue LTR_retriever con un 77.90%, esto se puede observar en la Figura 35.

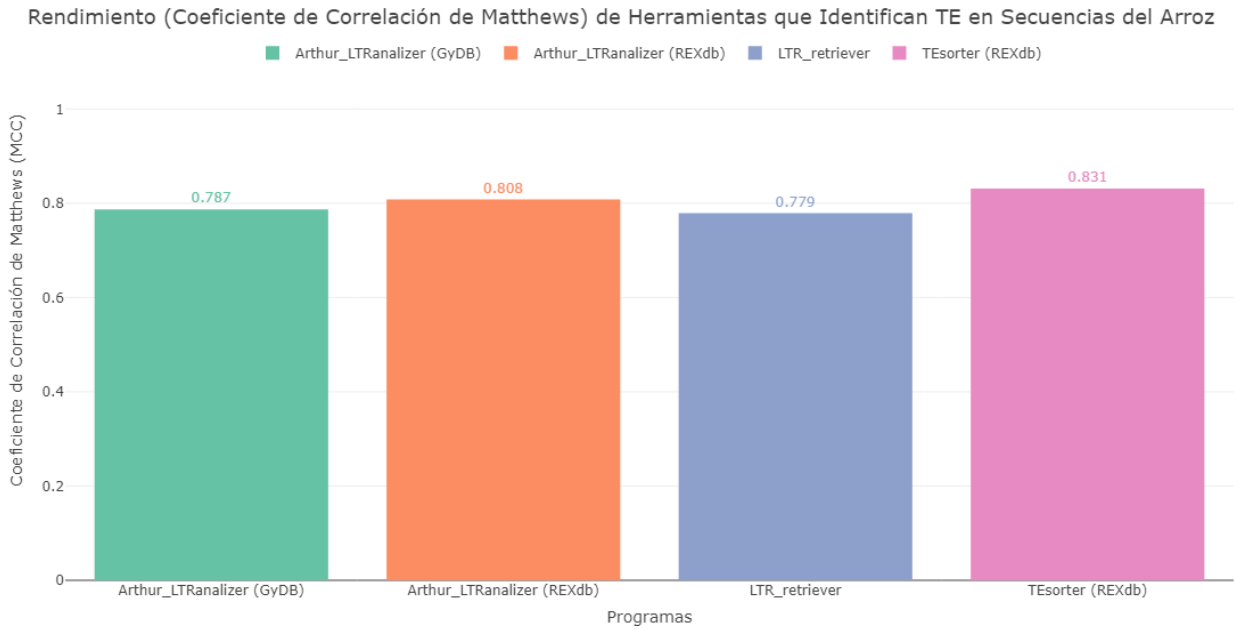

Figura 35. Evaluación del Coeficiente de Correlación de Matthews de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

También se calculó el porcentaje de clados de retrotransposones LTR clasificados por las tres herramientas usando secuencias genómicas del arroz, los resultados se pueden apreciar en la Figura 36, el programa que identificó un mayor porcentaje de linajes fue TEsorter con REXdb con un 84.85 %, en segundo puesto se ubicó Arthur_LTRanalizer con REXdb con un 83.31 %, seguido de Arthur_LTRanalizer con GyDB con un 82.28 % y LTR_retriever no clasifico a los retrotransposones LTR a nivel de linaje.

El programa que clasificó rápidamente a las secuencias genómicas del arroz fue LTR_retriever con un tiempo de 3.29 horas, en segundo lugar se ubicó TEsorter con REXdb que tardó 4.84 horas, en tercer lugar estuvo Arthur_LTRanalizer con REXdb que tardó 5.18 horas, seguido de Arthur_LTRanalizer con GyDB que se demoró 5.37 horas, los resultados se pueden observar en la Figura 37.

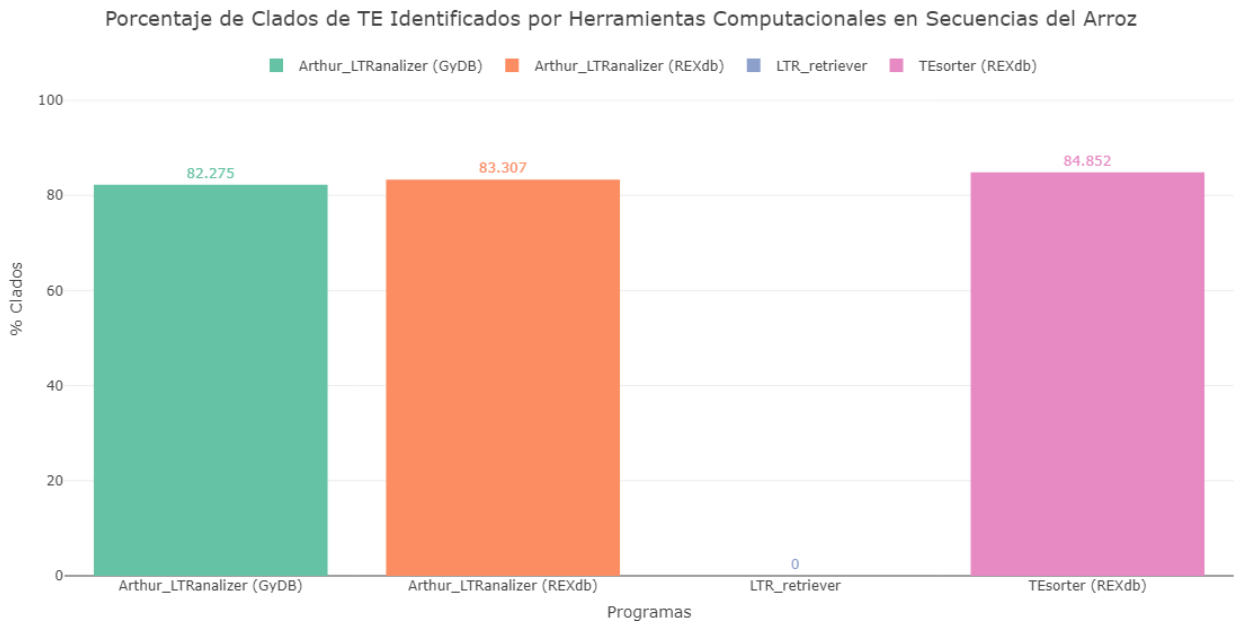

Figura 36. Porcentaje de clados de retrotrasposones LTR identificados por herramientas computacionales en secuencias genómicas del arroz.

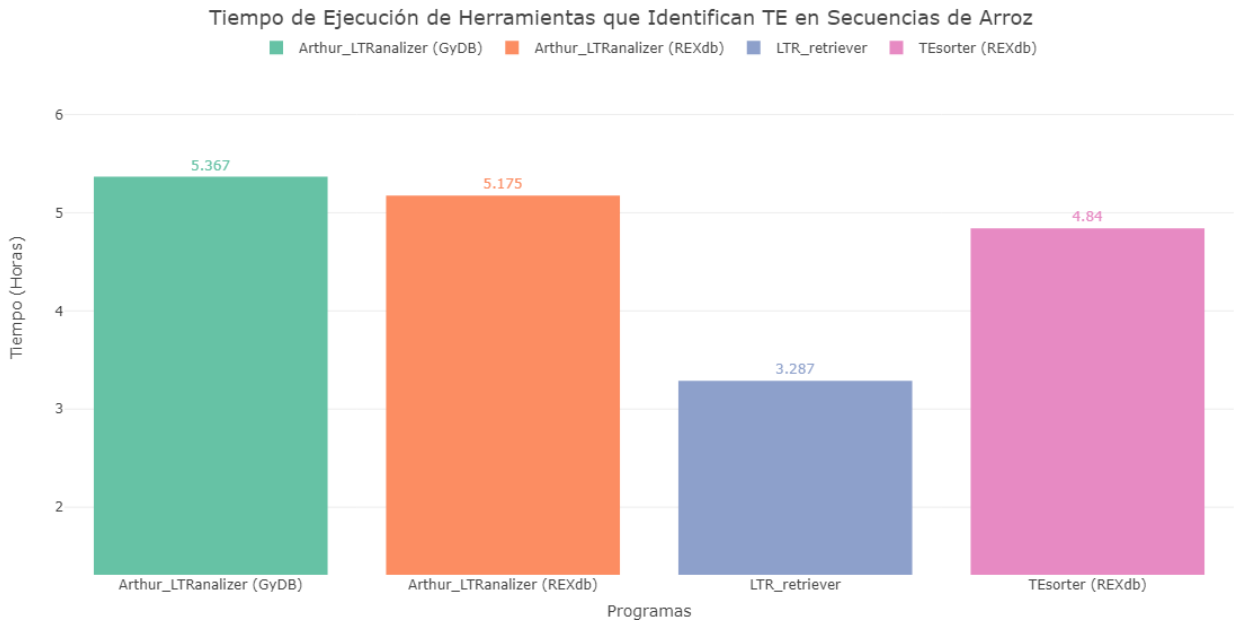

Figura 37. Evaluación del tiempo de ejecución de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

Al evaluar el rendimiento de las cuatro herramientas con las secuencias de elementos transponibles del arroz, se encontró que las herramientas que se caracterizan por tener mayor exactitud y sensibilidad son TEsorter con la librería REXdb junto con

Arthur_LTRanalizer con REXdb, a continuación se ubicó TEsorter con GyDB, Arthur_LTRanalizer con GyDB, LTR_retriever y en último lugar LTRclassifier. Con relación a la especificidad y precisión las tres herramientas con mayor porcentaje fueron TEsorter con REXdb y GyDB y Arthur_LTRanalizer, en tanto que la que estuvo en último puesto fue LTRclassifier, la representación gráfica de estas medidas se puede apreciar en la Figura 38.

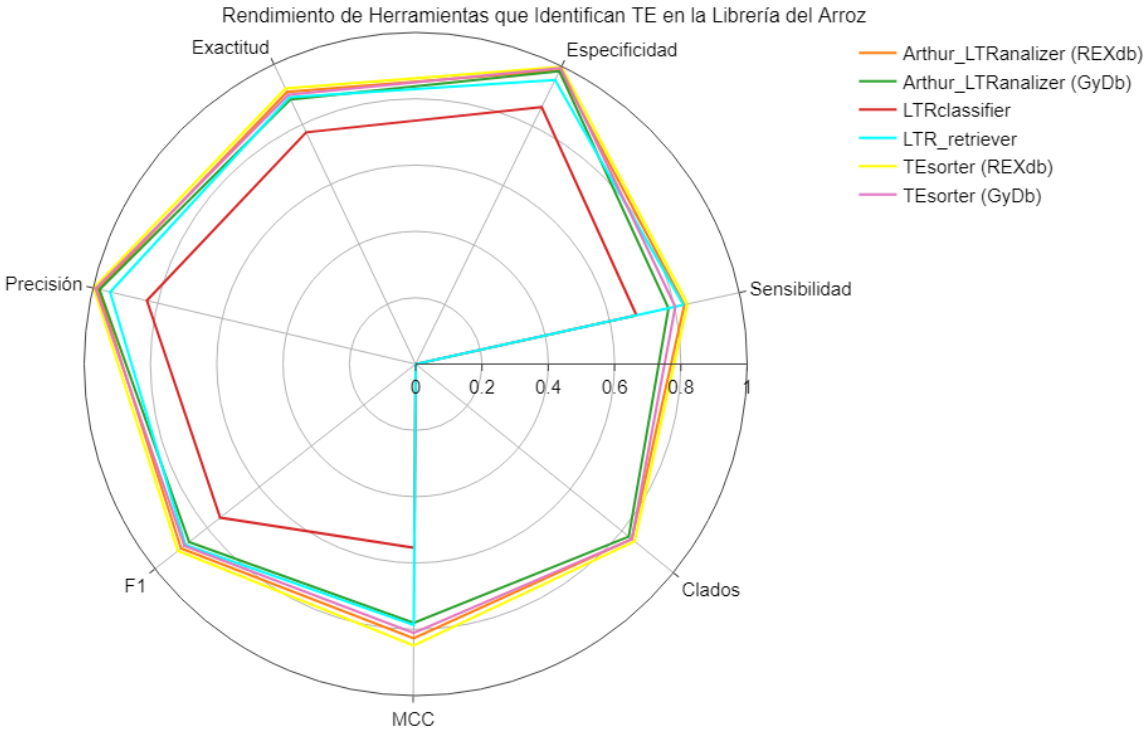

Figura 38. Rendimiento de herramientas que identifican retrotransposones LTR en las secuencias de TE del arroz.

Los dos programas que identifican un mayor porcentaje de clados fueron TEsorter con REXdb y Arthur_LTRanalizer con REXdb, seguidos de TEsorter y Arthur_LTRanalizer con GyDB, mientras que LTR_retriever y LTRclassifier no clasificaron retrotransposones LTR a nivel de linaje. Las herramientas que presentaron una mayor puntuación F1 fueron TEsorter con REXdb y Arthur_LTRanalizer con REXdb y la que obtuvo menor puntaje fue LTRclassifier. Las dos herramientas que obtuvieron un mayor Coeficiente de Correlación de Matthews fueron TEsorter con REXdb junto con Arthur_LTRanalizer con REXdb y la que obtuvo el menor coeficiente fue LTR_classifier. Con relación al tiempo

de ejecución LTR_retriever fue el programa que analizó secuencias rápidamente, a continuación se ubicaron TEsorter con REXdb y Arthur_LTRanalizer con REXdb, seguido de TEsorter con GyDB y Arthur_LTRanalizer con GyDB, el programa que tardó más tiempo en analizar a los elementos fue LTRclassifier.

La evaluación del rendimiento de las cuatro herramientas permitió determinar que TEsorter con REXdb y GyDB obtuvieron los mayores porcentajes de especificidad, exactitud, precisión y sensibilidad en relación a las otras herramientas al evaluarlo con secuencias de elementos transponibles del maíz, a continuación se ubicó el nuevo algoritmo Arthur_LTRanalizer con la librería REXdb y GyDB, mientras que las herramientas que presentaron un menor porcentaje fueron LTR_retriever y LTRclassifier, esto se puede apreciar en la Figura 39.

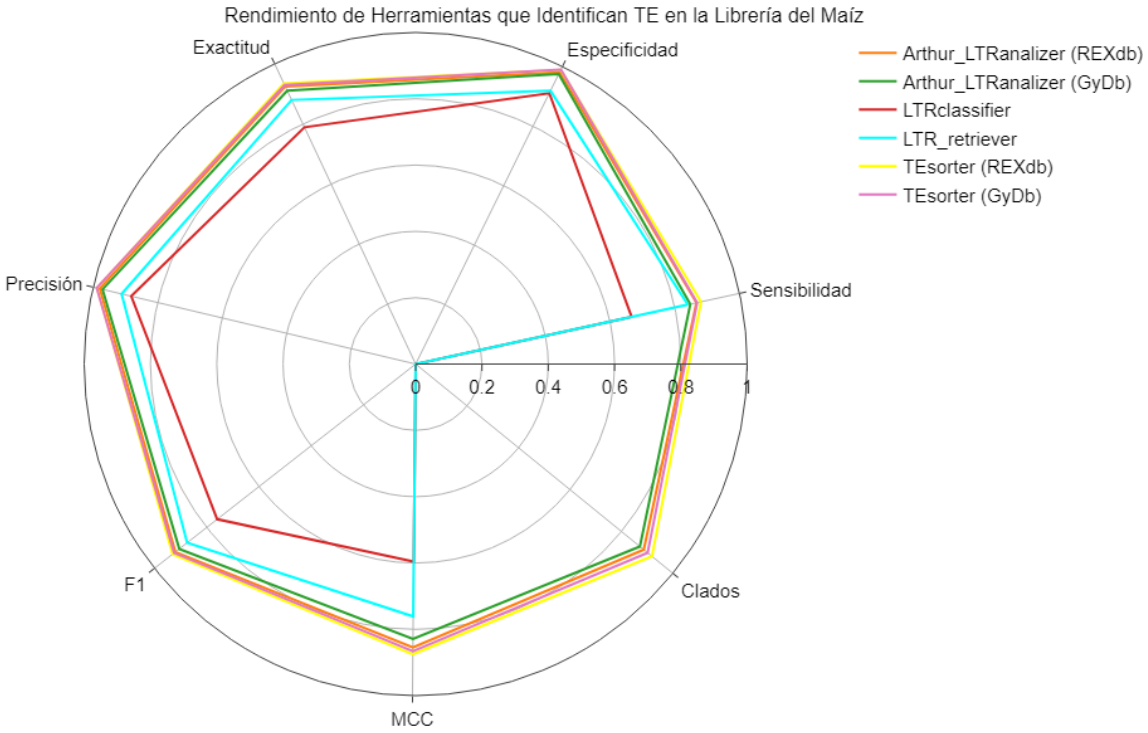

Figura 39. Rendimiento de herramientas que identifican retrotransposones LTR en las secuencias de TE del maíz.

Los programas que clasificaron un mayor porcentaje de retrotransposones LTR a nivel de linaje fueron TEsorter con REXdb y GyDB y Arthur_LTRanalizer con la librería de

REXdb, seguido de Arthur_LTRanalizer con GyDB, en tanto que LTRclassifier y LTR_retriever no clasificaron retrotransposones a nivel de linaje. Con relación al puntaje F1, las dos herramientas que tuvieron mayor porcentaje fueron TEsorter y Arthur_LTRanalizer. En relación al tiempo de ejecución, la herramienta que tardó más tiempo fue LTRclassifier y las más rápida fue LTR_retriever. Los dos programas con mayor Coeficiente de Correlación de Matthews fueron TEsorter y Arthur_LTRanalizer, estas medidas se esquematizan en la Figura 39.

Al evaluar el rendimiento de los tres programas con secuencias genómicas del arroz se obtuvieron resultados similares a los anteriormente indicados con las secuencias de elementos transponibles, como se muestra en la Figura 40. TEsorter con la librería REXdb junto con Arthur_LTRanalizer con REXdb constituyeron las dos herramientas que presentaron mayor exactitud, especificidad, sensibilidad y precisión en comparación a las otras herramientas, a continuación se ubicó Arthur_LTRanalizer con GyDB junto con LTR_Retriever.

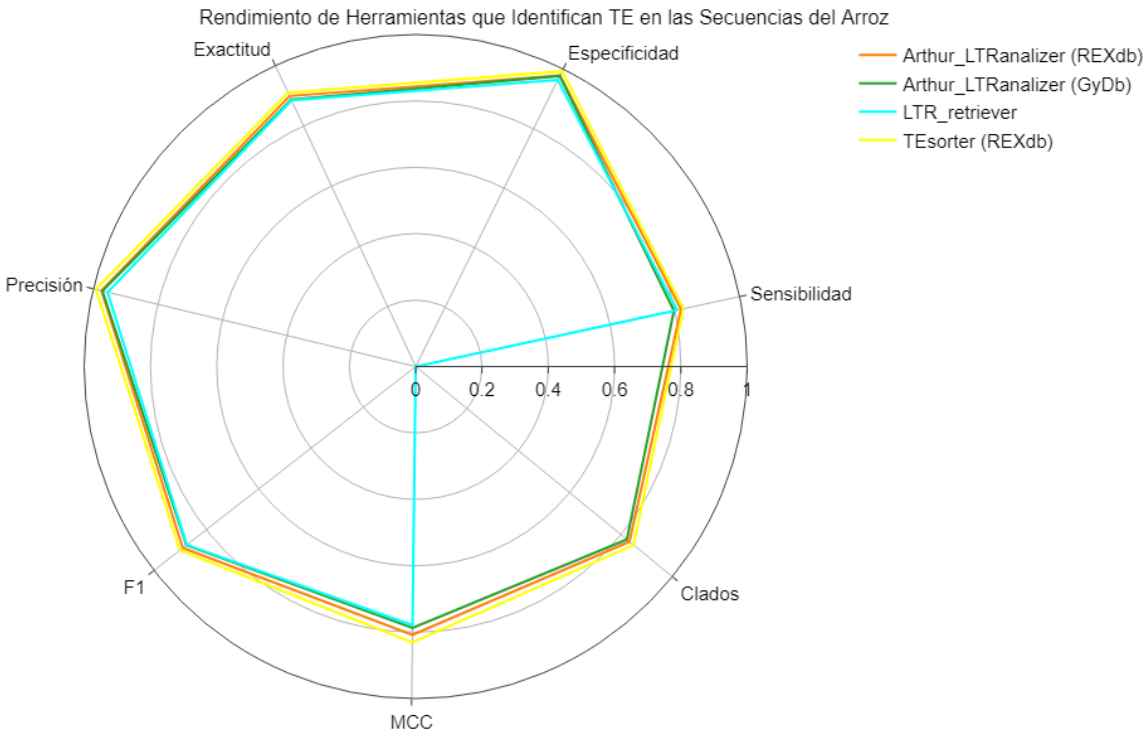

Figura 40. Rendimiento de herramientas que identifican retrotransposones LTR en las secuencias genómicas del arroz.

Con relación al porcentaje de linajes identificados, TEsorter con REXdb y Arthur_LTRanalizer con REXdb clasificaron el mayor porcentaje de elementos retrotransponibles LTR a nivel de su linaje, mientras que LTR_retriever no los clasificó a nivel de linaje. Las dos herramientas que tuvieron mayor Coeficiente de Correlación de Matthews fueron TEsorter y Arthur_LTRanalizer con la REXdb. Con respecto al tiempo de ejecución LTR_retriever tardó menos tiempo, le siguieron TEsorter y Arthur_LTRanalizer con REXdb y en último lugar se ubicó Arthur_LTRanalizer con GyDB.

Adicionalmente, se realizaron las curvas ROC, que relacionan la tasa de falsos positivos con la tasa de verdaderos positivos, en la Figura 41 se puede observar que los dos programas que tienen un mejor rendimiento fueron TEsorter con REXdb y Arthur_LTRanalizer con REXdb, mientras que los programas que tienen una peor estimación fueron LTR_retriever y LTRclassifier.

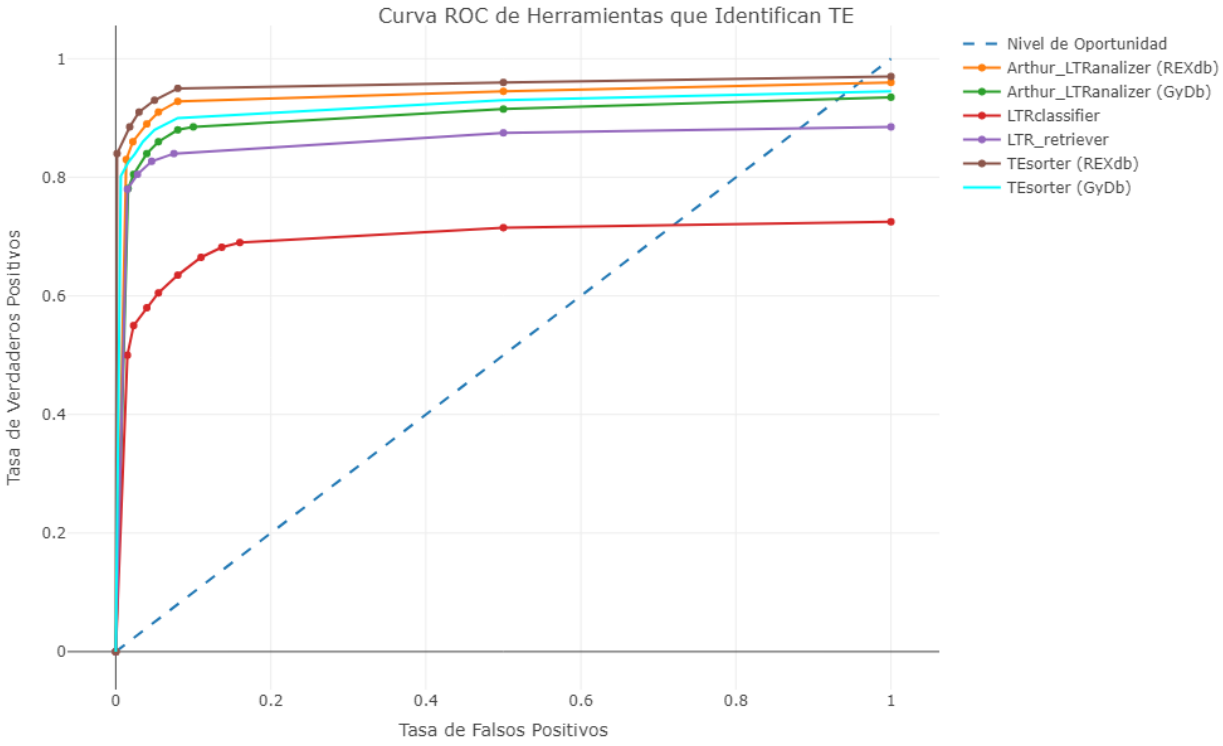

Figura 41. Curvas ROC de herramientas que identifican retrotransposones LTR.

La nueva herramienta computacional diseñada, Arthur_LTRanalizer, toma en cuenta la ventaja de la naturaleza conservada de los dominios proteicos de los retrotransposones.

Usa dos bases de datos de referencia, abiertas y activamente soportadas, como son REXdb y GyDB de perfiles de secuencias de dominios proteicos que están conservados, en lugar de usar bases de datos de elementos representativos.

La evaluación de la nueva herramienta Arthur_LTRanalizer, para identificar y clasificar elementos transponibles provenientes de secuencias de elementos transponibles, se desarrolló tomando como referencia las bases de datos curadas de elementos transponibles del arroz y del maíz, se comparó el rendimiento del nuevo algoritmo con tres herramientas computacionales que identifican y clasifican elementos transponibles como son: el módulo anotate_TE del paquete LTRretriever, LTRclassifier y TEsorter con su base de referencia REXdb y GyDB.

La aplicación implementada en Python cuenta con la función de multiprocesamiento, con lo cual se redujo el tiempo de ejecución, además la interfaz gráfica fue de mucha ayuda para el usuario ya que permitió escoger el archivo de entrada, el tipo de secuencias a analizar, la librería de referencia contra la que se quiere evaluar y la ruta de salida. La herramienta es muy útil para usuarios con poco conocimiento en líneas de comandos, tiene acceso libre y gratuito.

La herramienta Arthur_LTRanalizer, al ser evaluada con dos librerías curadas de elementos transponibles del arroz y del maíz y también con secuencias genómicas del arroz, presentó los siguientes resultados con la librería REXdb: una precisión promedio del 97.97 % que indica que el 2.03% de secuencias LTR fueron falsamente reportadas como LTR, una sensibilidad del 83.87 %, una especificidad del 98.27 %, exactitud de 91.07 %, un puntaje F1 del 90.33 % y un Coeficiente de Correlación de Matthews del 83.00%.

Al comparar las cuatro herramientas computacionales, Arthur_LTRanalizer con la librería de REXdb se ubicó entre los dos primeros lugares, por lo que es una herramienta confiable y eficiente, ya que tuvo un buen desempeño al identificar y clasificar retrotransposones LTR, tiene alta sensibilidad comparable a otras herramientas exitosas como LTR_retriever y TEsorter. La nueva herramienta es fácil de instalar y usar, usa un algoritmo rápido y preciso basado en perfiles de Modelos Ocultos de Markov, presenta

alta precisión y especificidad, tiene una baja tasa de predicción de falsos positivos, produce predicciones exactas en un tiempo razonable y usando memoria disponible en una computadora personal moderna.

El programa retornó archivos con secuencias anotadas de nucleótidos y aminoácidos, los cuales son fáciles de entender y pueden ser usados en análisis comparativos subsecuentes y filogenéticos. La herramienta puede ser fácilmente personalizada para identificar y clasificar cualquier grupo de elementos transponibles que tengan una secuencia de dominios proteicos conservados. Arthur_LTRanalizer puede proporcionar soporte computacional a investigadores, en particular a aquellos que estudian genomas de plantas, ya que reporta características biológicas de los retrotransposones LTR. El programa puede facilitar el trabajo de investigadores interesados en el descubrimiento y análisis detallado de la diversidad y evolución de retrotransposones y de otros elementos transponibles; así como también contribuye al desarrollo de la bioinformática en el Ecuador, ya que constituye una base para posteriores investigaciones en el área de biología.

CONCLUSIONES.

- Se realizó una interfaz gráfica de usuario que permitió escoger los parámetros de entrada para el análisis de las secuencias, los parámetros que se pudieron seleccionar fueron: el archivo de entrada, el tipo de secuencias sean genómicas o de elementos transponibles, nucleotídicas o de aminoácidos, una de las dos bases de referencia de perfiles proteicos de REXdb y GyDB contra las que se realiza la búsqueda de retrotransposones LTR y la ruta en la que se almacenaron los archivos de salida generados por la herramienta. La interfaz diseñada fue eficiente, fácil de usar, de ejecución rápida y amigable con el usuario
- Se elaboraron las instrucciones de programación para el módulo de identificación de retrotransposones LTR empleando el lenguaje Python, que permitieron analizar las secuencias de entrada de aminoácidos contra los perfiles proteicos de REXdb y GyDB. Se empleó el método de perfiles del Modelo Oculto de Markov (HMM) el cual realizó predicciones precisas en diferentes secuencias de plantas. Para eliminar falsos positivos e incrementar la exactitud de la búsqueda, los resultados de coincidencias fueron filtrados tomando en cuenta el valor esperado y la cobertura, se dio prioridad a aquellos elementos que presentaron mayor puntuación.
- Se desarrolló el software para el módulo de clasificación de los retrotransposones LTR usando dominios proteicos de la base de referencia REXdb y GyDB, se clasificó a los elementos tomando en cuenta la presencia y el orden de sus dominios proteicos y se identificó a los elementos completos que estuvieron conformados por todos sus dominios. Arthur_LTRanalizer fue capaz de clasificar en promedio a un 85.30 % de elementos retrotransponibles LTR a nivel de linaje.
- Se evaluó el desempeño del nuevo algoritmo usando medidas de rendimiento como: exactitud, precisión, sensibilidad, puntaje F1, Coeficiente de Correlación de Matthews, curvas ROC, adicionalmente se obtuvo otras medidas como el porcentaje de linajes clasificados y el tiempo de ejecución. También se comparó el rendimiento de la nueva herramienta con otras tres como son: LTR_retriever, LTRclassifier y TEsorter, el resultado de esta evaluación indicó que Arthur_LTRanalizer con la

librería REXdb proporciona resultados muy fiables, se ubicó dentro de las dos mejores herramientas analizadas, presenta una tasa baja de falsos positivos, es una herramienta eficiente, exacta, fácil de instalar y usar y de rápida ejecución.

RECOMENDACIONES.

El programa Arthur_LTRanalizer extrae secuencias de los dominios proteicos correspondientes, las cuales pueden ser fácilmente usadas en posteriores análisis comparativos y filogenéticos para estudiar procesos evolutivos de un grupo particular de elementos transponibles con más profundidad. Por lo que el programa puede personalizarse de acuerdo a las necesidades de los investigadores y contribuir al estudio de la diversidad y evolución de diferentes grupos de elementos transponibles.

En el futuro se puede perfeccionar el programa, mejorando la eficiencia en el tiempo de ejecución de las instrucciones, reducir dependencias, añadir la opción de usar características estructurales de elementos transponibles con filtros para su detección. También se podría determinar elementos autónomos y no autónomos en base al archivo de salida que proporciona Arthur_LTRanalizer y usar un segundo paso de clasificación para aquellos elementos que no fueron clasificados usando la regla 80-80-80 con BLAST. Adicionalmente, se podría extraer las secuencias que cuentan con dominio de reverso transcriptasa de cada elemento identificado, ya que este es el dominio más conservado y es apropiado para análisis filogenéticos.

En un futuro cercano se planea incluir la base de referencia de elementos transponibles de Dfam, la cual es una biblioteca curada que también presenta perfiles HMM, esta base de datos ayudaría a identificar inserciones de transposones en las secuencias de los retrotransposones. También se quiere incluir a Arthur_LTRanalizer como un módulo del programa Inpactor2, ya que este es un nuevo algoritmo que permite detectar y clasificar retrotransposones LTR basado en aprendizaje profundo, el nuevo módulo Arthur_LTRanalizer podría complementar a Inpactor2, lo que contribuiría a confirmar la predicción y/o detección de inserciones de transposones en las secuencias de retrotransposones.

DISCUSIÓN.

La expansión de los datos genómicos crea una necesidad urgente de desarrollar herramientas de software moderno para ayudar a detectar elementos transponibles, particularmente retrotransposones LTR que son los más abundantes en los genomas de plantas, con el objetivo de disminuir las limitaciones de las herramientas actualmente disponibles. La identificación y clasificación de elementos transponibles presenta un reto computacional y algorítmico debido a que las diferentes clases de elementos transponibles tienen características distintivas, las cuales permiten el desarrollo de programas para cada tipo de elementos transponibles.

Cada método usado para identificar y clasificar a estos elementos presenta fortalezas y debilidades, por lo cual es indispensable desarrollar herramientas que involucren diferentes métodos y aproximaciones, uno de los métodos que más se ha usado y que ha entregado resultados precisos es el método de perfiles de Modelos Ocultos de Markov (HMM), mientras que los programas que se basan en la estructura están comúnmente asociados con altas tasas de falsos positivos y este error puede propagarse en análisis posteriores.

En base a los estudios de Newman et al, en los que se clasificó a los retrotransposones LTR a nivel de superfamilia en Copia y Gypsy y más tarde se sub dividieron estas familias en clados o linajes, se han ido desarrollando bases de datos con secuencias de dominios proteicos de elemento retrotransponibles a nivel de linaje, dentro de estas se encuentran REXdb y GyDB, se escogieron estas bases ya que son unas de las más completas y cuyo acceso es gratuito.

Algunas herramientas son difíciles de instalar o están obsoletas, otras tienen dependencias externas con pasos complejos de instalación, no usan la ventaja del multiprocesamiento, pocas fueron diseñadas con una visión en el post procesamiento, producen altas tasas de predicción de falsos positivos y no obtienen todos los LTR conocidos. Mientras que la nueva herramienta Arthur_LTRanalizer es fácil de instalar y usar, la ejecución es con multiprocesamiento tomando ventaja de las arquitecturas

multinúcleo de las computadoras personales, en el que se procesan varios archivos al mismo tiempo en paralelo, corre eficientemente en una computadora personal, tiene parámetros establecidos y generalizados para identificar y clasificar a los elementos.

El programa Arthur_LTRanalizer puede ser fácilmente adaptado para buscar otros elementos transponibles con dominios proteicos en su genoma eucariota, como retroelementos Penelope y DIRS, tomando en cuenta sus dominios RT y tyrosine recombinase, los transposones de ADN podrían identificarse por sus dominios de transposasa como en el caso de helicasas de ADN en Helitrons y polimerasa de ADN en Mavericks.

La identificación automatizada y precisión para clasificar a los retrotransposones LTR contribuye a que investigadores continúen indagando acerca de las capacidades regulatorias de estos elementos y al entendimiento de la evolución de las plantas. Arthur_LTRanalizer es una herramienta atractiva para proyectos futuros de anotación que involucren a elementos transponibles en especial a retrotransposones de terminal largo.

REFERENCIAS BIBLIOGRÁFICAS.

Amselem, J., Cornut, G., Choisne, N., Alaux, M., Alfama-Depauw, F., Jamilloux, V., … Quesneville, H. (2019). RepetDB: A unified resource for transposable element references. *Mobile DNA*, *10*(6), 1–8. https://doi.org/10.1186/s13100-019-0150-y

Amselem, J., Cornut, G., Choisne, N., Alaux, M., Alfama-Depauw, F., Jamilloux, V., … Quesneville, H. (2022a). Urgi: Transposable elements, Arabidopsis. Retrieved October 24, 2022, from https://urgi.versailles.inra.fr/Data/Transposable-elements/Arabidopsis

Amselem, J., Cornut, G., Choisne, N., Alaux, M., Alfama-Depauw, F., Jamilloux, V., … Quesneville, H. (2022b). Urgi: Transposable elements, Maize. Retrieved November 7, 2022, from https://urgi.versailles.inra.fr/Data/Transposable-elements/Maize

Anaconda Inc. (2022). Anaconda. Retrieved September 12, 2022, from https://www.anaconda.com/

Aroh, O., & Halanych, K. M. (2021). Genome-wide characterization of LTR retrotransposons in the non-model deep-sea annelid Lamellibrachia luymesi. *BMC Genomics*, *22*(466), 1–11. https://doi.org/10.1186/s12864-021-07749-1

Baxevanis, A. D., Bader, G. D., & Wishart, D. S. (2020). *Bioinformatics* (Fourth). New Jersey: John Wiley & Sons, Inc.

Bioconda. (2022). Bioconda. Retrieved September 12, 2022, from https://bioconda.github.io/

Biryukov, M., & Ustyantsev, K. (2021). DARTS: An Algorithm for Domain-Associated Retrotransposon Search in Genome Assemblies. *Genes*, *13*(9), 1–8. https://doi.org/10.3390/genes13010009

Bourque, G., Burns, K. H., Gehring, M., Gorbunova, V., Seluanov, A., Hammell, M., … Feschotte, C. (2018). Ten things you should know about transposable elements. *Genome Biology* , *19*(199), 1–12. https://doi.org/10.1186/s13059-018-1577-z

Chang, J., Chapman, B., Friedberg, I., Hamelryck, T., de Hoon, M., Cock, P., … Wilczyński, B. (2023). *Biopython: Tutorial and cookbook*. Retrieved from http://biopython.org/DIST/docs/tutorial/Tutorial-1.81.pdf

Chaparro, C., Gayraud, T., de Souza, R. F., Domingues, D. S., Akaffou, S., Laforga Vanzela, A. L., … Hamon, P. (2015). Terminal-repeat retrotransposons with GAG domain in Plant Genomes: A new testimony on the complex world of transposable elements. *Genome Biology and Evolution*, *7*(2), 493–504. https://doi.org/10.1093/gbe/evv001

Cho, J. (2021). *Plant transposable elements: Methods and protocols* (Vol. 2250). New York: Springer Science. https://doi.org/10.1007/978-1-0716-1134-0

Chuong, E. B., Elde, N. C., & Feschotte, C. (2017). Regulatory activities of transposable elements: From conflicts to benefits. *Nature Reviews Genetics*, *18*(2), 71–86. https://doi.org/10.1038/nrg.2016.139

Cock, P., Antao, T., Chang, J., Chapman, B., Cox, C., Dalke, A., … de Hoon, M. (2009). Biopython: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Curry, E. (2021). *Introduction to bioinformatics with R: A practical guide for biologists*. Florida: Taylor & Francis Group.

Domínguez, M., Dugas, E., Benchouaia, M., Leduque, B., Jiménez-Gómez, J. M., Colot, V., & Quadrana, L. (2020). The impact of transposable elements on tomato diversity. *Nature Communications*, *11*(4058), 1–11. https://doi.org/10.1038/s41467-020-17874-2

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, *14*(9), 755–763. https://doi.org/10.1093/bioinformatics/14.9.755

Eddy, S. R. (2020). *HMMER User's Guide: Biological sequence analysis using profile hidden Markov models*. Cambridge: Howard Hughes Medical Institute.

Eddy, S. R. (2022). HMMER: Biosequence analysis using profile hidden Markov models. Retrieved November 8, 2022, from http://hmmer.org/

Edwards, D. (2022). *Plant bioinformatics: Methods and protocols* (Third). New York: Springer Science. https://doi.org/10.1007/978-1-0716-2067-0

Ferreira, V., Matus, J. T., Pinto-Carnide, O., Carrasco, D., Arroyo-García, R., & Castro, I. (2019). Genetic analysis of a white-to-red berry skin color reversion and its transcriptomic and metabolic consequences in grapevine (Vitis vinifera cv. 'Moscatel

Galego'). *BMC Genomics*, *20*(952), 1–17. https://doi.org/10.1186/S12864-019-6237-5

Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., … Salazar, G. A. (2016). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, *44*, D279–D285. https://doi.org/10.1093/nar/gkv1344

Fischer, C. N., Campos, V. D. A., & Barella, V. H. (2018). On the search for retrotransposons: Alternative protocols to obtain sequences to learn profile hidden Markov models. *Journal of Computational Biology*, *25*(5), 517–527. https://doi.org/10.1089/cmb.2017.0219

Forero, D. A. (2022). *Bioinformatics and human genomics research*. Florida: CRC Press. https://doi.org/10.1201/9781003005926

Gabriel, A. (2023). *Retrotransposons and human disease: L1 retrotransposons as a source of genetic diversity*. New Jersey: World Scientific. https://doi.org/10.1142/9789811249228_fmatter

Galindo-González, L., Mhiri, C., Deyholos, M. K., & Grandbastien, M. A. (2017). LTR-retrotransposons in plants: Engines of evolution. *Gene*, *626*, 14–25. https://doi.org/10.1016/j.gene.2017.04.051

Garcia-Pérez, J. L. (2016). *Transposons and retrotransposons: Methods and protocols* (Vol. 1400). New York: Springer Science. https://doi.org/10.1007/978-1-4939-3372-3

Grüning, B., Dale, R., Sjödin, A., Chapman, B. A., Rowe, J., Tomkins-Tinch, C. H., … Köster, J. (2018). Bioconda: Sustainable and comprehensive software distribution for the life sciences. *Nature Methods*, *15*(7), 475–476. https://doi.org/10.1038/s41592-018-0046-7

Gupta, M. K., & Behera, L. (2021). *Bioinformatics in rice research: Theories and techniques*. Singapore: Springer Nature Singapore Pte Ltd. https://doi.org/10.1007/978-981-16-3993-7

Hasija, Y. (2023). *All about bioinformatics from beginner to expert*. London: Elsevier Inc. https://doi.org/978-0-443-15250-4

Huang, D., Yuan, Y., Tang, Z., Huang, Y., Kang, C., Deng, X., & Xu, Q. (2019). Retrotransposon promoter of Ruby1 controls both light- and cold-induced

accumulation of anthocyanins in blood orange. *Plant Cell and Environment*, *42*(11), 3092–3104. https://doi.org/10.1111/pce.13609

IRGSP. (2022). EnsemblPlants: Oryza sativa Japonica Group (IRGSP-1.0). Retrieved December 6, 2022, from http://plants.ensembl.org/Oryza_sativa/Info/Index

Ismail, H. D. (2022). *Bioinformatics: A practical guide to NCBI databases and sequence alignments*. Florida: CRC Press. https://doi.org/10.1201/9781003226611

Jedlicka, P., Lexa, M., Vanat, I., Hobza, R., & Kejnovsky, E. (2019). Nested plant LTR retrotransposons target specific regions of other elements, while all LTR retrotransposons often target palindromes and nucleosome-occupied regions: In silico study. *Mobile DNA*, *10*(50), 1–14. https://doi.org/10.1186/s13100-019-0186-z

Joldersma, D., & Liu, Z. (2018). The making of virgin fruit: the molecular and genetic basis of parthenocarpy. *Journal of Experimental Botany*, *69*(5), 955–962. https://doi.org/10.1093/JXB/ERX446

Kaufmann, M., Klinger, C., & Savelsbergh, A. (2017). *Functional genomics: Methods and protocols*. New York: Springer . https://doi.org/10.1007/978-1-4939-7231-9

Kong, Q., Siauw, T., & Bayen, A. M. (2021). *Python Programming and Numerical Methods A Guide for Engineers and Scientists*. London: Elsevier Inc.

Krogh, A., Larsson, B., von Heijne, G., & Sonnhammer, E. L. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *Journal of Molecular Biology*, *305*(3), 567–580. https://doi.org/10.1006/jmbi.2000.4315

Liu, Y., Tahir Ul Qamar, M., Feng, J. W., Ding, Y., Wang, S., Wu, G., … Chen, L. L. (2019). Comparative analysis of miniature inverted-repeat transposable elements (MITEs) and long terminal repeat (LTR) retrotransposons in six Citrus species. *BMC Plant Biology*, *19*(140), 1–16. https://doi.org/10.1186/s12870-019-1757-3

Llorens, C., Futami, R., Covelli, L., Domínguez-Escribá, L., Viu, J. M., Tamarit, D., … Moya, A. (2011). The Gypsy Database (GyDB) of mobile genetic elements: Release 2.0. *Nucleic Acids Research* , *39*(suppl_1), D70–D74. https://doi.org/10.1093/nar/gkq1061

Llorens, C., Futami, R., Covelli, L., Dominguez-Escriba, L., Viu, J. M., Tamarit, D., … Moya, A. (2023). GyDB: Gypsy database 2.0. Retrieved January 21, 2023, from https://gydb.org/index.php/Collection_HMM

Lu, Z., Pan, L., Wei, B., Niu, L., Cui, G., Wang, L., … Wang, Z. (2021). Fine mapping of the gene controlling the fruit skin hairiness of prunus persica and its uses for mas in progenies. *Plants*, *10*(1433), 1–10. https://doi.org/10.3390/plants10071433

MacFarland, T. W., & Yates, J. M. (2021). *Using R for biostatistics*. Cham: Springer Nature.

Mastrodomenico, R. (2022). *The python book*. Oxford: John Wiley and Sons Ltd.

MathWorks. (2022). *Bioinformatics Toolbox: User's Guide*. Massachusetts: The MathWorks Inc.

Meador, D. (2022). *Building data science solutions with anaconda: A comprehensive starter guide to building robust and complete models*. Birmingham: Packt Publishing Ltd.

Microsoft. (2022). Visual studio code. Retrieved September 2, 2022, from https://code.visualstudio.com/

Monat, C., Tando, N., Tranchant-Dubreuil, C., & Sabot, F. (2016). LTRclassifier: A website for fast structural LTR retrotransposons classification inplants. *Mobile Genetic Elements*, *6*(6), 1–6. https://doi.org/10.1080/2159256X.2016.1241050

Moore, A. (2021). *Python GUI programming with Tkinter: Design and build functional and user-friendly GUI applications* (Second). Birmingham: Packt Publishing Ltd.

Nakaya, H. (2021). *Bioinformatics*. Brisbane: Exon Publications. https://doi.org/10.36255/exonpublications.bioinformatics.2021

Neumann, P., Novák, P., Hoštáková, N., & Macas, J. (2019). Systematic survey of plant LTR-retrotransposons elucidates phylogenetic relationships of their polyprotein domains and provides a reference for element classification. *Mobile DNA*, *10*(1), 1–17. https://doi.org/10.1186/S13100-018-0144-1

Neumann, P., Novák, P., Hoštáková, N., & Macas, J. (2022). REXdb: A reference database of transposable element protein domains. Retrieved November 3, 2022, from http://repeatexplorer.org/

Novák, P., Neumann, P., Pech, J., Steinhaisl, J., & Macas, J. (2013). RepeatExplorer: A Galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next-generation sequence reads. *Bioinformatics*, *29*(6). https://doi.org/10.1093/bioinformatics/btt054

Orozco-Arias, S., Isaza, G., & Guyot, R. (2019). Retrotransposons in plant genomes: Structure, identification, and classification through bioinformatics and machine learning. *International Journal of Molecular Sciences*, Vol. 20, pp. 1–31. https://doi.org/10.3390/ijms20153837

Orozco-Arias, S., Jaimes, P. A., Candamil, M. S., Jiménez-Varón, C. F., Tabares-Soto, R., Isaza, G., & Guyot, R. (2021). Inpactordb: A classified lineage-level plant LTR retrotransposon reference library for free-alignment methods based on machine learning. *Genes*, *12*(190), 1–17. https://doi.org/10.3390/GENES12020190

Orozco-Arias, S., Jaimes, P., Candamil, M., Jiménez-Varón, C., Tabares-Soto, R., Isaza, G., & Guyot, R. (2022). InpactorDB: A Plant classified lineage-level LTR retrotransposon reference library for free-alignment methods based on Machine Learning. Retrieved November 4, 2022, from https://zenodo.org/record/4453481

Orozco Arias, S., Liu, J., Tabares-Soto, R., Ceballos, D., Domingues, D. S., Garavito, A., … Guyot, R. (2018). Inpactor, integrated and parallel analyzer and classifier of LTR retrotransposons and ITS application for pineapple LTR retrotransposons diversity and dynamics. *Biology*, *7*(32), 1–16. https://doi.org/10.3390/biology7020032

Orozco-Arias, S., Lopez-Murillo, L. H., Candamil-Cortés, M. S., Arias, M., Jaimes, P. A., Rossi Paschoal, A., … Guyot, R. (2023). Inpactor2: A software based on deep learning to identify and classify LTR-retrotransposons in plant genomes. *Briefings in Bioinformatics*, *24*(1), 1–10. https://doi.org/10.1093/bib/bbac511

Orozco Arias, S., Tobon-Orozco, N., Piña, J. S., Jiménez-Varón, C. F., Tabares-Soto, R., & Guyot, R. (2020). Tip_finder: An HPC software to detect transposable element insertion polymorphisms in large genomic datasets. *Biology*, *9*(281), 1–17. https://doi.org/10.3390/biology9090281

Ou, S., & Jiang, N. (2018). LTR_retriever: A highly accurate and sensitive program for identification of long terminal-repeat retrotransposons. *Plant Physiology*, *176*(2), 1410–1422. https://doi.org/10.1104/pp.17.01310

Ou, S., & Jiang, N. (2023). LTR_retriever. Retrieved January 1, 2023, from https://github.com/oushujun/LTR_retriever

Ou, S., Su, W., Liao, Y., Chougule, K., Agda, J. R. A., Hellinga, A. J., … Hufford, M. B. (2019). Benchmarking transposable element annotation methods for creation of a streamlined, comprehensive pipeline. *Genome Biology*, *20*(275), 1–18. https://doi.org/10.1186/s13059-019-1905-y

Ou, S., Su, W., Liao, Y., Chougule, K., Agda, J. R. A., Hellinga, A. J., … Hufford, M. B. (2023). EDTA: Extensive de-novo TE annotator. Retrieved January 1, 2023, from https://github.com/oushujun/EDTA

Ouyang, S., & Buell, C. (2022). Oriza Repeat Database. Retrieved November 28, 2022, from http://rice.uga.edu/annotation_oryza.shtml

Ouyang, Z., Wang, Y., Ma, T., Kanzan, G., Wu, F., & Zhang, J. (2021). Genome-wide identification and development of LTR retrotransposon-based molecular markers for the melilotus genus. *Plantas*, *10*(890), 1–15. https://doi.org/10.3390/plants10050890

Peterson, T. (2013). *Plant transposable elements: Methods and protocols*. New York: Humana Press. https://doi.org/10.1007/978-1-62703-568-2

Potter, S. C., Luciani, A., Eddy, S. R., Park, Y., Lopez, R., & Finn, R. D. (2018). HMMER web server: 2018 update. *Nucleic Acids Research*, *46*(W1), W200–W204. https://doi.org/10.1093/nar/gky448

Ramakrishnan, M., Satish, L., Sharma, A., Kurungara Vinod, K., Emamverdian, A., Zhou, M., & Wei, Q. (2022). Transposable elements in plants: Recent advancements, tools and prospects. *Plant Molecular Biology Reporter*, *40*(4), 628–645. https://doi.org/10.1007/S11105-022-01342-W

Ranganathan, S., Nakai, K., Schönbach, C., & Gribskov, M. (2019). *Encyclopedia of bioinformatics and computational biology*. Cambridge: Elsevier.

Riehl, K., Riccio, C., Miska, E. A., & Hemberg, M. (2022). TransposonUltimate: software for transposon classification, annotation and detection. *Nucleic Acids Research*, *50*(11), 1–13. https://doi.org/10.1093/NAR/GKAC136

Rocha, M., & Ferreira, P. (2018). *Bioinformatics algorithms: Design and implementation in python*. Oxford: Elsevier Inc.

Rodriguez, M., & Makałowski, W. (2022). Software evaluation for de novo detection of transposons. *Mobile DNA*, *13*(14), 1–14. https://doi.org/https://doi.org/10.1186/s13100-022-00266-2

Roseman, M. (2021). *Modern Tkinter for busy Python developers: Quickly learn to create great looking user interfaces for Windows, Mac and Linux using Python's standard GUI toolkit* (Third). Victoria: Late Afternoon Press.

Singh, D. Bukhsh., & Pathak, R. Kumar. (2022). *Bioinformatics : Methods and applications*. Oxford: Elsevier Science.

Sofi, M. Y., Shafi, A., & Masoodi, K. Z. (2022). *Bioinformatics for everyone*. London: Elsevier Inc.

Speight, A. (2021). *Visual studio code for python programmers*. New Jersey: John Wiley & Sons Inc.

Storer, J., Hubley, R., Rosen, J., Wheeler, T. J., & Smit, A. F. (2021). The Dfam community resource of transposable element families, sequence models, and genome annotations. *Mobile DNA*, *12*(2), 1–14. https://doi.org/10.1186/s13100-020-00230-y

Sultana, N., Menzel, G., Seibt, K. M., Garcia, S., Weber, B., Serçe, S., & Heitkam, T. (2022). Genome-wide analysis of long terminal repeat retrotransposons from the cranberry Vaccinium macrocarpon. *Journal of Berry Research*, *12*(2), 165–185. https://doi.org/10.3233/JBR-211515

TAIR. (2023). EnsemblPlants: Arabidopsis thaliana (TAIR10). Retrieved December 5, 2023, from http://plants.ensembl.org/Arabidopsis_thaliana/Info/Index

The Python Software Foundation. (2022). Python. Retrieved October 4, 2022, from https://www.python.org/

Thieme, M., & Bucher, E. (2018). Plant epigenetics coming of age for breeding applications. In M. Mirouze, E. Bucher, & P. Gallusci (Eds.), *Advances in Botanical Research* (Vol. 88, pp. 165–202). Elsevier . https://doi.org/10.1016/bs.abr.2018.09.001

Valencia, J. D., & Girgis, H. Z. (2019). LtrDetector: A tool-suite for detecting long terminal repeat retrotransposons de-novo. *BMC Genomics* , *20*(450), 1–14. https://doi.org/10.1186/s12864-019-5796-9

Vangelisti, A., Mascagni, F., Usai, G., Natali, L., Giordani, T., & Cavallini, A. (2020). Low long terminal repeat (LTR)-retrotransposon expression in leaves of the marine phanerogam Posidonia Oceanica L. *Life*, *10*(30), 1–12. https://doi.org/10.3390/life10030030

Vetaryan, S., Kwan, Y. Y., Namasivayam, P., Ho, C. L., & Syed Alwee, S. S. R. (2018). Isolation and characterisation of oil palm LEAFY transcripts. *Biotechnology & Biotechnological Equipment*, *32*(4), 888–898. https://doi.org/10.1080/13102818.2018.1464949

Vicient, C., & Casacuberta, J. (2020). Additional ORFs in Plant LTR-Retrotransposons. *Frontiers in Plant Science*, *11*(555), 1–5. https://doi.org/10.3389/fpls.2020.00555

Wheeler, T. J., Clements, J., Eddy, S. R., Hubley, R., Jones, T. A., Jurka, J., … Finn, R. D. (2013). Dfam: A database of repetitive DNA based on profile hidden Markov models. *Nucleic Acids Research*, *41*(D1), D70–D82. https://doi.org/10.1093/nar/gks1265

Wheeler, T. J., Clements, J., & Finn, R. D. (2014). Skylign: A tool for creating informative, interactive logos representing sequence alignments and profile hidden Markov models. *BMC Bioinformatics*, *15*(7), 1–9. https://doi.org/10.1186/1471-2105-15-7

Wicker, T., Sabot, F., Hua-Van, A., Bennetzen, J. L., Capy, P., Chalhoub, B., … Schulman, A. H. (2007). A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics* , *8*(12), 973–982. https://doi.org/10.1038/nrg2165

Zhang, L., Hu, J., Han, X., Li, J., Gao, Y., Richards, C. M., … Cong, P. (2019). A high-quality apple genome assembly reveals the association of a retrotransposon and red fruit colour. *Nature Communications*, *10*(1494), 1–13. https://doi.org/10.1038/s41467-019-09518-x

Zhang, R. G., Li, G. Y., Wang, X. L., Dainat, J., Wang, Z. X., Ou, S., & Ma, Y. (2022). TEsorter: An accurate and fast method to classify LTR-retrotransposons in plant genomes. *Horticulture Research*, *9*, 1–4. https://doi.org/10.1093/hr/uhac017

Zhang, R. G., Li, G. Y., Wang, X. L., Dainat, J., Wang, Z. X., Ou, S., & Ma, Y. (2023). TEsorter. Retrieved December 31, 2022, from https://github.com/zhangrengang/TEsorter

Zhou, S. S., Yan, X. M., Zhang, K. F., Liu, H., Xu, J., Nie, S., … Mao, J. F. (2021). A comprehensive annotation dataset of intact LTR retrotransposons of 300 plant genomes. *Scientific Data*, *8*(174), 1–9. https://doi.org/10.1038/s41597-021-00968-x

Zhu, T., Wang, L., Rimbert, H., Rodriguez, J. C., Deal, K. R., de Oliveira, R., … Luo, M. C. (2021). Optical maps refine the bread wheat Triticum aestivum cv. Chinese Spring genome assembly. *Plant Journal*, *107*, 303–314. https://doi.org/10.1111/tpj.15289

ANEXOS.

ANEXO 1.

Mensaje de la Interfaz Gráfica de Inicio del Proceso de Búsqueda.

Mensaje de la Interfaz Gráfica de Finalización del Proceso de Búsqueda.

ANEXO 2.

Archivo Obtenido del Análisis contra Perfiles de Modelos Ocultos de Markov (HMM), de secuencias de Elementos Transponibles del Arroz.

```
#
  --- full sequence --- -------------- this domain -------------   hmm coord   ali coord
env coord
# target name        accession   tlen query name                           accession
qlen   E-value  score  bias   #  of  c-Evalue  i-Evalue  score  bias  from    to  from    to
 from    to  acc description of target
#------------------- ---------- ----- -------------------- ---------- -----
----- --------- ------ ----- --- --- --------- --------- ------ ----- ----- ----- -----
----- ----- ----- ---- ---------------------
Class_I/LTR/Ty3_gypsy/non-chromovirus/OTA/Tat/Retand:Ty3-INT      -            313
Os0030_INT_RIRE2#LTR/Gypsy|aa3 -           3478  2.2e-143  471.0   0.0   1   1  1.4e-143
3.5e-143  470.3   0.0     1   313  1697  2009  1697  2009 0.99 -
Class_I/LTR/Ty3_gypsy/non-chromovirus/OTA/Tat/Retand:Ty3-RT       -            174
Os0030_INT_RIRE2#LTR/Gypsy|aa3 -           3478   1.3e-95  311.6   0.0   1   1   8.3e-96
2.1e-95  310.9   0.0     1   174  1062  1235  1062  1235 1.00 -
Class_I/LTR/Ty3_gypsy/non-chromovirus/OTA/Tat/Ogre:Ty3-INT        -            318
```

Archivo de Salida en Formato FASTA, con Secuencias de Retrotransposones LTR Clasificadas, de Secuencias de Elementos Transponibles del Arroz.

```
>Os0008_INT#LTR/Copia/SIRE#GAG|SIRE Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-GAG
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-GAG;Nombre=SIRE-GAG;Dominio=GAG;Clado
=SIRE;Cobertura=80.6;Valor_e=5e-21;Probabilidad=0.97
LLSGISHSDYDRVAHLQTTHEIWIALSNFHQGTNNIKELRRDLFKKEYIKFEMKPGEALD
DYLSRFNKILSDLRSVL
>Os0008_INT#LTR/Copia/SIRE#PROT|SIRE
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-PROT
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-PROT;Nombre=SIRE-PROT;Dominio=PROT;Cl
ado=SIRE;Cobertura=100.0;Valor_e=1.5e-27;Probabilidad=0.99
WIVDSGCSRHMTGDKNWFSSLKKASKTESIIFGDASTSAVLATSLVKVNEKFELKNVALV
EDLKYNLLS
>Os0008_INT#LTR/Copia/SIRE#INT|SIRE Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-INT
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-INT;Nombre=SIRE-INT;Dominio=INT;Clado
=SIRE;Cobertura=100.0;Valor_e=7.4e-87;Probabilidad=0.98
HRRLGHVGFDHLTRLSGLDLVRGLPKLKKDLDLICTPCRHAKMVASSHTPIVSVMTDAPG
QLLHMDIVGPARVQSVGGKWYVLVIVDDFSRYSWVFFMATKDEAFQHFRGLFLQLEVEFP
GSLKRI*SDNGGEFKNTSFEQFCNERGLEHEFSSPRVPQQNGVVERKNHVLVEMARTMLD
EYKTPRKFWAEAINTACYISN
>Os0008_INT#LTR/Copia/SIRE#RT|SIRE Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RT
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RT;Nombre=SIRE-RT;Dominio=RT;Clado=SI
RE;Cobertura=100.0;Valor_e=1.6e-134;Probabilidad=1.00
WINAMHEELENFERNKDWTLVEPPSGHNIIGTKWVFKNKQNEDGLIVRNKARLVAQGFTQ
VEGLYFDETFAPVARIEAIRLLLAFAASKGFKLYQMDVKSAFLNGFIHEEVYVKQPPGFE
NSDFPNHVFKLSKALYGLKQAPMAWYDRLKNFLLAKGFTMGKVDKTLFVLKHGDNQLFVQ
IYVDDIIFGYSTHALVVDFAENMRREFEMSMMGELSYFLGLQIKQTPQGTFVHQTKYTKD
LLERFKMENCKPISTP
>Os0008_INT#LTR/Copia/SIRE#RH|SIRE Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RH
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RH;Nombre=SIRE-RH;Dominio=RH;Clado=SI
RE;Cobertura=58.7;Valor_e=5.8e-32;Probabilidad=0.98
DADFGGCRIDRKSTSGTCHFLSTSLIAWSSRKQSSVAQSTAESEYGAAASCCSQILWLLS
TLKDYGLTFEKVPLL
```

Archivo de Salida TSV con Elementos Clasificados a Nivel de Linaje, de Secuencias de Elementos Transponibles del Arroz.

```
#TE     Orden   Superfamilia    Clado   Completo        Hebra   Dominio
Os0008_INT#LTR/Copia    LTR     Copia   SIRE    Si      +       GAG|SIRE PROT|SIRE INT|SIRE
RT|SIRE RH|SIRE
Os0016_INT#LTR/Gypsy    LTR     Gypsy   Retand  Si      +       GAG|Retand PROT|Retand
RT|Retand RH|Retand INT|Retand
Os0019_INT#LTR/Gypsy    LTR     Gypsy   Ogre    Si      +       GAG|Ogre PROT|Ogre RT|Ogre
RH|Ogre INT|Ogre
Os0025_INT#LTR/Gypsy    LTR     Gypsy   Athila  No      +       GAG|Athila PROT|Athila
Os0028_INT#LTR/Copia    LTR     Copia   TAR     Si      +       GAG|TAR PROT|TAR INT|TAR
RT|TAR RH|TAR
```

Archivo de Salida TSV con Elementos Clasificados a Nivel de Linaje, de Secuencias de Elementos Transponibles del Arroz.

```
#Id     Longitud        Valor-e Cobertura       Probabilidad    Puntaje
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-GAG 77      5e-21   80.6    0.97    0.74
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-PROT        69      1.5e-27 100.0   0.99
1.31
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-INT 201     7.4e-87 100.0   0.98    1.41
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RT  256     1.6e-134        100.0   1.00
1.72
Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RH  75      5.8e-32 58.7    0.98    0.82
```

Archivo de Salida GFF con Elementos Clasificados a Nivel de Linaje, de Secuencias de Elementos Transponibles del Arroz.

```
Os0008_INT#LTR/Copia    ARTHUR_LTRanalizer      CDS     265     495     0.74    +       0
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-GAG;Nombre=SIRE-GAG;Dominio=GAG;Clado
=SIRE;Cobertura=80.6;Valor_e=5e-21;Probabilidad=0.97
Os0008_INT#LTR/Copia    ARTHUR_LTRanalizer      CDS     2522    2728    1.31    +       1
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-PROT;Nombre=SIRE-PROT;Dominio=PROT;Cl
ado=SIRE;Cobertura=100.0;Valor_e=1.5e-27;Probabilidad=0.99
Os0008_INT#LTR/Copia    ARTHUR_LTRanalizer      CDS     2921    3523    1.41    +       1
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-INT;Nombre=SIRE-INT;Dominio=INT;Clado
=SIRE;Cobertura=100.0;Valor_e=7.4e-87;Probabilidad=0.98
Os0008_INT#LTR/Copia    ARTHUR_LTRanalizer      CDS     4219    4986    1.72    +       0
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RT;Nombre=SIRE-RT;Dominio=RT;Clado=SI
RE;Cobertura=100.0;Valor_e=1.6e-134;Probabilidad=1.00
Os0008_INT#LTR/Copia    ARTHUR_LTRanalizer      CDS     5221    5445    0.82    +       0
ID=Os0008_INT#LTR/Copia|Class_I/LTR/Ty1_copia/SIRE:Ty1-RH;Nombre=SIRE-RH;Dominio=RH;Clado=SI
RE;Cobertura=58.7;Valor_e=5.8e-32;Probabilidad=0.98
```