Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación

Título: *Herramientas computacionales para la comparación de genomas y detección de genes ortólogos con un enfoque de grafo bipartido.*

Tesis en opción al grado de Máster en Bioinformática y Biología Computacional

Autor: *Miguel Ángel Fernández Marín.*

Tutores: *Dr. Ricardo Grau Ábalo.*

MSc. Deborah Galpert Cañizares.

MSc. Maikel Yelandi Leyva Vázquez.

Villa Clara, abril 2012
“Año del 54 de la Revolución”

"El hombre que pone corazón en lo que hace, consigue recursos donde los incapaces se dan por vencidos"

Simón Bolívar.

AGRADECIMIENTOS

Agradezco:

Al Comandante en Jefe Fidel Castro Ruz y a la Revolución por haber transformado sueños y esperanzas en realidades. A nuestros queridos profesores de la maestría que han contribuido en nuestra formación profesional. A la Universidad de las Ciencias Informáticas por confiar tantos recursos en pos del conocimiento.

Gracias a mis tutores Deborah, Maikel y Ricardo Grau y al estudiante de 5to año de la carrera Ciencias de la Computación Reinier Millo Sánchez, por el apoyo y dedicación incondicional brindado. Por su paciencia en horas de desesperación y circunstancias extremas.

A mi querida esposa Débora González Tolmo que ha contribuido de una forma especial en la realización de esta investigación, su apoyo y su presencia han hecho de la desesperanza una infinita perseverancia. Gracias mi amor por ser como eres.

A mis tres grandes amores, Elidianys, Blanca Rosa y Cecilia por haber confiado en mí y haber apoyado cada paso que he dado en la vida, las amo y doy gracias por tenerlas siempre.

DEDICATORIA

Dedico la presente investigación a mi familia, en especial a Blanca Rosa y a Cecilia por enseñarme el significado de la vida.

A mi querida hermana Elidianys.

A mi esposa Débora González Tolmo.

RESUMEN

En los estudios de comparación de genomas, específicamente el problema de la detección de genes ortólogos, se toman en cuenta las mutaciones de nucleótidos y los reordenamientos globales en los genomas. Los algoritmos consultados en la bibliografía que abordan este problema muestran aproximadamente un 90% de precisión siendo éste un problema latente.

El presente trabajo plantea como objetivo diseñar una herramienta computacional para la detección de genes ortólogos, basado en el enfoque de grafos bipartidos, que combine, mediante el operador de agregación "Ordered Weighted Average operator" (OWA) y la media aritmética, rasgos como la homología de los genes, la longitud de las secuencias, la relación evolutiva según el modelo "The Five Model" (SG2009) y la pertenencia a regiones conservadas teniendo en cuenta los reordenamientos globales en los genomas y las mutaciones. La fase de agrupamiento del algoritmo implementa la técnica de particionamiento de grafos BUS, que incluye la búsqueda de bloques con un orden conservado, la eliminación de ambigüedades y la selección de los mejores subconjuntos uno a uno.

Se tomó como referencia para validar la clasificación el algoritmo Inparanoid 7.0, aunque el mismo no se ha reportado con un 100% de exactitud en la clasificación. El algoritmo y la experimentación utilizando los genomas *Saccharomyces Cervisiae* y *Schizosaccharomyces Pombe* fueron implementados en Matlab 7.10.0. La validación muestra una coincidencia en la clasificación de un 85.24%.

Palabras Claves: Genes ortólogos, alineamiento, reordenamientos globales en los genomas, precisión de algoritmos, segmentos conservados, grafo bipartido, particionamiento de grafo BUS.

ABSTRACT

In comparative genomic studies, specifically in the ortholog detection problem, nucleotides mutations and global genome rearrangements are taking into account. The ortholog detection algorithms consulted in literature show about a 90% of accuracy, hence, precision is a latent problem.

The present work has the main goal of designing a computational tool for the detection of ortholog genes based on the bipartite graph approach to combine with the Ordered Weighted Average operator (OWA) and the arithmetic media, the homology of the genes, the length of the sequences, the evolutionary relationship under The Five Model (SG2009) and the membership to conserved regions considering global genome rearrangements and mutations. The clustering step implements the BUS partitioning technique that includes the building of the synteny blocks, the deletion of ambiguities and the selection of the best "one to one" subsets.

The classification of the Inparanoid 7.0 algorithm was taken as a reference to validate the classification of the proposed algorithm, even though Inparanoid´s classification has not been reported with a 100% of accuracy. The algorithm and the experiments with *Saccharomyces Cervisiae* and *Schizosaccharomyces Pombe* genomes were implemented in Matlab 7.10.0. The validation process has shown 85.24% of coincidences.

Keywords: Ortholog genes, alignment, global genome rearrangements, algorithm accuracy, conserved synteny, bipartite graphs, BUS graph partitioning.

TABLA DE CONTENIDOS

Tabla de Contenidos

INTRODUCCIÓN

El genoma constituye para todo ser vivo, las características genéticas y hereditarias, que determina las diferencias exclusivas entre ellos. Su estudio, ha permitido desarrollar importantes avances y mejoras en la calidad de vida de los humanos y de otras especies en general. El mismo se encuentra en el ADN o ácido desoxirribonucleico y se caracteriza por ser único e irrepetible. El ADN está presente en todas las células que componen a un organismo, determinando los rasgos físicos y biológicos de estos. Además, al ser el ADN la molécula universal de la herencia, las investigaciones se enfocan en su mayoría hacia su estudio. El conocimiento molecular, puede ser la clave de muchos fenómenos que actualmente se entienden a niveles menos profundos, descritos por otras ciencias biológicas como la fisiología, biología celular y bioquímica.

El estudio preciso de los genomas, para lograr el conocimiento exacto de la secuencia, constituye un punto de arranque para nuevos descubrimientos en las ciencias biomédicas. El mismo, posibilita obtener respuestas cada vez más acertadas, sobre la expresión de genes, la regulación genética, la interacción de las células con su entorno y la homología entre secuencias de distintas especies; con el objetivo de comprender la evolución. En consecuencia, la genómica comparativa, es la ciencia que estudia las semejanzas y diferencias entre genomas de diferentes organismos. Es un intento de beneficiarse de la información proporcionada por las firmas de la selección natural, para entender la función y los procesos evolutivos que actúan sobre los genomas. Aunque es todavía un campo reciente, promete adquirir nuevas percepciones sobre muchos aspectos de los cambios en los seres vivos a través del tiempo (López et al., 2005).

Cuando se comparan genomas, se puede detectar homologías en la composición genética entre distintas especies, lo que posibilita el reconocimiento de funciones comunes en distintos organismos, como es el caso de los genes ortólogos. La detección de genes ortólogos es un área dentro de la Bioinformática, que aunque ha sido ampliamente trabajada, aún requiere de algoritmos más precisos (Stevens et al., 2009) por su valor en la predicción de la función de las proteínas. Esta disyuntiva se presenta en la bibliografía, como un problema de aprendizaje no supervisado, específicamente, como el agrupamiento de genes ortólogos.

Los genes ortólogos son aquellos que evolucionan a partir de un ancestro común en un proceso de especiación. Tienen como contraparte los genes parálogos, que también se asemejan en su secuencia pero que han resultado de la duplicación. Es tarea del proceso para la detección de ortólogos, distinguir los genes que son ortólogos a partir de los homólogos en cuanto a la secuencia. Otros genes son ortólogos pues sus productos proteicos preservan su función aunque no conservan la similitud de la secuencia (López et al., 2005).

Varios algoritmos de detección de ortólogos, generalmente, se basan en el enfoque de árbol filogenético para la clasificación de los genes. Inicialmente, seleccionan los homólogos, generan los alineamientos múltiples para cada grupo de dominio de homólogos y construyen el árbol filogenético a partir de estos. Finalmente, extraen los ortólogos de estos árboles. Los métodos de árbol típicamente reconcilian el árbol de genes con el de especies para distinguir los nodos de duplicación y especiación. Una alternativa a este último enfoque es el basado en grafo o de comparación par a par de genes. Estos algoritmos parten de construir un grafo bipartito con la similitud par a par de secuencias de los genes de dos genomas en comparación y luego aplican heurísticas como: "Reciprocall Best Hits" (RBH), "Reciprocall Smallest Distance" (RSD), para podar el grafo antes del agrupamiento (Kuzniar et al., 2008).

Diversas bases de datos han sido construidas a partir de la predicción basada en grafo como por ejemplo: euKaryotic Orthologous Groupdatabase (KOG) (Tatusov et al., 2003), INPARANOID (O'Brien et al., 2005), OrthoMCL (Li et al., 2003), OrthoMCL_DB (Chen et al., 2006), MultiParanoid (Alexeyenko et al., 2006), Eukaryotic Gene Orthologues database (EGO) (Lee et al., 2002) y más recientemente INPARANOID 7.0 (Östlund et al., 2010). Otros algoritmos como SOAR (Chen et al., 2005), MSOAR (Fu et al., 2007) toman en cuenta la similitud entre las secuencias, los reordenamientos globales de genes y los bloques de genes que conservan el orden para estimar la distancia evolutiva.

El reporte de una prueba de referencia reciente (Salichos and Rokas, 2011), muestra que aquellos algoritmos que integran información sobre la búsqueda de similitud, la filogenética y las regiones de los genomas que conservan el orden deben ser una mejor opción para la genómica evolutiva y los estudios funcionales. Con esta motivación, en la Universidad Central de Las Villas, se ha desarrollado un conjunto de investigaciones encaminadas a la comparación de genomas utilizando la combinación de rasgos. Muestra de esto fue el trabajo de Diploma "Herramientas Computacionales para la Comparación de Genomas" (Blay, 2009), en la cual definen rasgos proponiendo un enfoque de función local-global para la comparación de genes entre dos genomas de eucariotas cercanos en la evolución. Se trabajó el enfoque de grafo, construyendo el grafo bipartito a partir del alineamiento par a par de secuencias, utilizando el algoritmo de Needleman Wunsch (Needleman and Wunsch, 1970) implementado en Matlab 7.4, sin aplicar técnicas de paralelizar procesos. Sobre la base del esquema del algoritmo BUS (Kamvysselis, 2003), se combinó la similitud de secuencias con la información de los bloques de orden conservado. Se aplicaron políticas de poda y agrupamiento. Se validaron los resultados con el cromosoma 5 de *Saccharomyces Cerevisiae* y el genoma completo de *Saccharomyces Bayanus* con resultados prometedores.

Posteriormente, teniendo en cuenta el enfoque local-global anteriormente mencionado, surgió un nuevo algoritmo que desarrolla técnicas para paralelizar procesos apoyado por el software Matlab 9.0, para la comparación par a par de genes, utilizando las estructuras algebraicas de código genético definidas en el propio Laboratorio (Sánchez and Grau, 2009). Se realizó la prueba de este algoritmo con el genoma *Saccharomyces Cerevisiae* y el genoma *Schizosaccharomyces Pombe*. Se validaron los resultados con la lista de ortólogos curada manualmente. Los resultados obtenidos sugirieron una mejora en la comparación de genes, a partir de la distancia evolutiva entre estos y en cuanto a la fase de poda y agrupamiento del algoritmo general de detección de ortólogos. Adicionalmente, el descubrimiento de algunas imprecisiones entre los datos de prueba y de referencia mostró la necesidad de una mejor preparación de los datos para la validación de los algoritmos.

En el propio laboratorio, se ha definido un nuevo modelo evolutivo denominado "TheFive Bases Model" (SG2009) y nuevas métricas de comparación de genes (Sánchez and Grau, 2009), que pueden ser efectivas en la detección de ortólogos. Además, la medida de disimilitud local-global se pueden conseguir mediante una agregación de distancias entre genes, que a su vez cumpla con las propiedades de una distancia como el "Ordered Weighted Average operator" (OWA) (Yager, 1988) y la media aritmética. Por último, debido a que se han realizado algoritmos basados en Best Unambiguos Subset (BUS) como el que propone (Kamvysselis, 2003), que logran la identificación para más del 90% de los genes, aplicar esta técnica puede favorecer la obtención de las relaciones de ortólogos entre pares de genes. Por todo lo anteriormente expuesto, surge el siguiente ***Problema Científico:***

Elevar la precisión de las herramientas computacionales para la detección de genes ortólogos basadas en el enfoque de grafo bipartito, teniendo en cuenta distintos rasgos de los genes, constituye en la actualidad un problema latente. El uso de los rasgos más analizados por la literatura como la homología y la pertenencia a regiones conservadas en conjunto con otros rasgos como la longitud de las secuencias, y la relación evolutiva según el modelo SG2009, pudieran contribuir al análisis de nuevas funciones de distancias entre genes. El estudio y la aplicación del operador OWA y la media aritmética favorecerían la combinación de rasgos. Luego, la aplicación de una estrategia de poda, propiciaría la eliminación de las relaciones entre genes evolutivamente lejanos y por lo tanto menos propensos a ser ortólogos. Por último, la aplicación del BUS como algoritmo de agrupamiento pudiera favorecer la obtención de las relaciones de ortólogos entre pares de genes.

Se define como ***objeto de estudio:*** Las herramientas computacionales para la detección de genes ortólogos basados en el enfoque de grafo bipartido. El cual se enmarca en el ***campo de acción:*** Aplicación de la teoría de grafos bipartidos en herramientas computacionales para la detección de genes ortólogos. Por lo que se propone como ***objetivo general:***

Diseñar una herramienta computacional para la detección de genes ortólogos, basado en el enfoque de grafos bipartidos, que combine la homología de los genes, la longitud de las secuencias, la pertenencia a regiones conservadas y la relación evolutiva entre ellos según el modelo SG2009; esto permitiría incrementar las posibilidades de determinar funciones desconocidas en genes a partir del conocimiento de las funciones en los ortólogos.

Para dar cumplimiento al objetivo general se definieron como **objetivos específicos**:

- ✓ Revisar el estado del arte sobre algoritmos existentes para la detección de genes ortólogos que utilicen la teoría de grafos bipartidos.
- ✓ Desarrollar un algoritmo que posibilite la extracción de los datos seleccionados de los ficheros de los genomas hacia una estructura de datos de fácil manejo.
- ✓ Crear un algoritmo que permita la conformación de rasgos entre genes de dos genomas, con información de la homología, la longitud de las secuencias, la pertenencia a regiones verdaderamente homólogas y la relación evolutiva entre ellos.
- ✓ Construir un algoritmo que a partir de la agregación de rasgos mediante el operador OWA o la media aritmética y el particionamiento sobre grafos permita conformar los grupos de genes ortólogos.
- ✓ Validar los resultados a través de experimentos con datos de *Saccharomyces Cerevisiae* y *Schizosaccharomyces Pombe* utilizando la clasificación del algoritmo INPARANOID 7.0.

Para alcanzar el objetivo propuesto y teniendo como base el problema a resolver, se formula la siguiente **hipótesis**: Si se construye una herramienta computacional, basada en la teoría de grafo bipartido, combinando la información sobre la homología, el tamaño entre las secuencias, la pertenencia a regiones verdaderamente homólogas y la relación evolutiva entre los genes, entonces se puede tener precisión en la detección de genes ortólogos.

Como **aporte científico** se espera la obtención de un algoritmo de detección de ortólogos basados en un enfoque de grafo bipartito, combinando la información sobre la homología y el tamaño entre las secuencias, la pertenencia a regiones verdaderamente homólogas y la relación evolutiva entre los genes, así como la aplicación de estrategias de poda y uso del BUS como métodos de agrupamiento sobre grafos. Como **aporte práctico** se ofrece una librería de algoritmos, implementados sobre el lenguaje de computación técnica Matlab en su versión 7.10.0 (R2010a), con el propósito de detectar genes ortólogos entre dos especies.

Este documento de la investigación se encuentra estructurado como se presenta a continuación:

Capítulo I: Fundamentación Teórica.

El presente capítulo, aclara conceptos de homología, ortología y paralogía fundamentales para entender el trabajo realizado. Además, realiza una revisión bibliográfica acerca de los algoritmos de detección de ortólogos con un enfoque de grafo. Igualmente, se hace un análisis de diferentes medidas de distancia local, con el propósito de ponderar las conexiones entre nodos de grafos bipartitos. También, se hace un estudio del operador OWA y la media aritmética como vías de combinar la información de rasgos; además de algoritmo de agrupamiento de grafos, BUS.

Capítulo II: Descripción de la solución propuesta.

En el capítulo, se identifican las herramientas informáticas y el lenguaje de programación a utilizar. Se describe de forma gráfica, utilizando diagramas de procesos, y en lenguaje natural, la solución informática propuesta para la detección de genes ortólogos.

Capítulo III: Validación de los resultados.

En este capítulo se realiza una comparación de los resultados obtenidos a partir de la investigación y los del algoritmo Inparanoid 7, que constituye una herramienta para la detección de genes ortólogos entre diferentes especies. Además, se analizan y discuten los experimentos obtenidos empleando la prueba estadísticas McNemar.

Justificación de la investigación

El presente trabajo es parte de los proyectos de investigación del Laboratorio de Bioinformática de la UCLV:

Título del proyecto: Métodos estadísticos y de Inteligencia Artificial para el análisis de secuencias de ADN.

Jefe del Proyecto: Dra. Gladys Casas Cardoso

Título del proyecto: Desarrollo y Aplicación de la Arquitectura de los genomas mediante la aplicación de las estructuras algebraicas

Jefe del Proyecto: Dr. Robersy Sánchez Rodríguez

Además el trabajo que se propone forma parte de las investigaciones para optar por el grado de doctor de la tutora en el tema aprobado por el CITMA en marzo del 2010 “Métodos Computacionales para la Comparación de Genomas e Identificación de Genes Ortólogos”.

CAPÍTULO 1
FUNDAMENTACIÓN TEÓRICA

El presente capítulo, abarca conceptos de homología, ortología y paralogía fundamentales para entender la investigación realizada. Además, aborda aspectos relacionados con las medidas de similitud utilizadas en la comparación de genes y comprende una revisión bibliográfica acerca de los algoritmos de detección de ortólogos con un enfoque de grafo. Se proponen diferentes medidas de distancia local, con el propósito de ponderar las conexiones entre nodos de grafos bipartitos en un nuevo algoritmo de detección de ortólogos basado en grafo. Se hace un estudio del operador OWA en la noción de distancias y la media aritmética como vías de combinar la información de las distancias locales. Incluye la descripción del algoritmo de agrupamiento sobre grafos, BUS.

1.1 Comparación y clasificación de genes

La rápida acumulación de los datos de genomas, es un desafío importante para los investigadores que intentan extraer la información funcional y evolutiva de estos. Para evitar el desbordamiento de la información, producto de la constante influencia de nuevas secuencias de genomas, se requiere de una clasificación evolutiva de los genes. Tales clasificaciones, se basan en la ortología y la paralogía, que describen los dos tipos fundamentales de relaciones de homología entre los genes.

1.1.1 Homología

La homología es una hipótesis comprobable, en la que los caracteres entre las diferentes especies comparten una similitud de secuencia significativa (al menos del 30 –35% como regla de oro para las secuencias de proteínas) descendiendo de un carácter ancestral común. Las secuencias que están relacionadas evolutivamente son conocidas como homólogas (Webber, 2004).

Las comparaciones entre genes, proteínas o genomas, son claves para obtener descubrimientos exitosos en la genómica comparativa. La forma más común de comparar dos genes, es realizar alineamientos de sus secuencias, lo que facilita determinar si poseen suficiente similitud como para poder justificar la existencia de homología entre ellos.

Existe una marcada diferencia entre el término similitud y homología. El primero es un concepto cuantificable, que puede medirse y expresarse como un porcentaje de identidad entre los datos. El segundo, se refiere a una conclusión obtenida de estos, e indica si están relacionadas o comparten una historia evolutiva. Los genes son o no son

homólogos, pero no existen grados de homología. Un alto grado de similitud entre secuencias no necesariamente se debe a un origen común evolutivo, solo que se utiliza el porcentaje de identidad o valores estadísticos relacionados con este para predecir la homología junto a la similitud estructural (Webber, 2004).

La similitud y el alineamiento, primeramente definidos por (Needleman and Wunsch, 1970), son globales; es decir, el alineamiento global se realiza sobre las secuencias completas. Más adelante, el problema de alineamiento local fue estudiado por (Smith and Waterman, 1981). Los algoritmos correspondientes utilizan matrices de puntuación para asignar diferentes pesos a la sustitución de los pares de caracteres en el cálculo del costo de cada paso. Los "huecos" o "gaps" pueden ser referidos como "indels" (inserciones/eliminaciones) y son tratados por los algoritmos con la substracción de una puntuación de penalidad $\gamma(s)$ que depende de la longitud s del "hueco". Generalmente, las penalidades lineales tienen la forma $\gamma(s) = sU$, asignando un costo de U unidades por carácter del "hueco" y las penalidades de afinidad tienen la forma $\gamma(s) = sU + W$ donde W es un costo de apertura de "hueco" y U es un costo de "hueco" extendido por residuo en el "hueco".

En particular el algoritmo de (Smith and Waterman, 1981), compara subcadenas de todas las posibles longitudes, calculando las puntuaciones del alineamiento basado en la noción de distancia de edición de Levenshtein del año 1965 (Deza, 2006),es decir, calculando el costo mínimo de transformación de una secuencia en la otra. El algoritmo posee una penalidad de apertura de "hueco" y otra de hueco extendido. Su tiempo de ejecución es de orden cúbico, en dependencia de la longitud de las secuencias que se comparan. Se reporta en la literatura como uno de los más usados en la comparación de proteínas. Sólo otras implementaciones más rápidas han ocupado su lugar en esta comparación, el algoritmo heurístico FASTA (Pearson, 1990) y el BLAST (Altschul, 1997) que brindan aproximaciones de la puntuación de alineamiento local.

El algoritmo de Smith-Waterman, no pone restricciones en términos de una tabla de similitud, en el alineamiento que reporta junto al cálculo de una puntuación positiva. En cambio, BLAST y FASTA ponen restricciones adicionales sobre los alineamientos en función de la velocidad: solo las secuencias por encima de cierto umbral de similitud son reportadas. Debido a esto, el algoritmo de Smith-Waterman es más sensitivo que BLAST y FASTA (Hulsen et al., 2006b). Sin embargo, BLAST y FASTA son más populares, debido a que el tiempo de ejecución es menor; además, por la adición de un valor de significación estadística *E-value*, que describe el número de coincidencias probables cuando se busca una secuencia en una base de datos de cierto tamaño. Representando el estimado de la probabilidad de encontrar este grado de similitud con una secuencia aleatoria. Siempre que E sea cercano a cero, más confiable

será la predicción de la homología (algunos consideran los alineamientos con *E-value* menor que 10^{-3}como suficiente evidencia de la homología (Webber, 2004).

1.1.2 Ortología y paralogía

La relación de ortología entre dos genes ocurre, cuando son homólogos de diferentes especies y se derivan de un gen en el último ancestro común, presentando las mismas funciones biológicas. De ahí, la importancia de identificar ortólogos, con un alto grado de fiabilidad, para la transferencia de información funcional entre los genes de diferentes organismos.

Los genes ortólogos son secuencias homólogas derivadas de un evento de especiación a partir de una sola secuencia ancestral en el último ancestro común de las especies que se comparan. Los ortólogos normalmente realizan funciones equivalentes en especies estrechamente relacionadas.

La relación de ortología se expresa en forma de grupos de ortólogos (relaciones uno-uno, uno-muchos o muchos-muchos). Además encontramos el concepto de co-ortólogos: una o más secuencias en un linaje que son colectivamente ortólogas con una o más en otro linaje debido a duplicaciones específicas (Kuzniar et al., 2008).

Por otra parte, los parálogos son secuencias homólogas derivadas por una duplicación a partir de una única secuencia. Las relaciones de parálogos pueden ocurrir en un mismo genoma. Los parálogos pueden contener funciones nuevas y es probable que tengan mecanismos distintos, pero biológicamente tienen funciones relacionadas. Los genes son parálogos, cuando la similitud se produce dentro del mismo genoma por duplicación de un gen. Los genes parálogos que preceden a la división de especies, son llamados “outparalogs”, los que proceden de la división de especies son llamados “inparalogs”(Remm et al., 2001).

Inparalogs: Parálogos que resultan de un linaje específico de duplicación después de un evento de especiación dado (a veces son llamados parálogos "recientes"). Es probable que hayan tenido funciones similares dentro de una especie.

Outparalogs: Parálogos resultantes de la duplicación antes de un evento dado de especiación (a veces llamado parálogos "antiguos"). Es probable que ellos tengan diferentes funciones.

A continuación se muestra un gráfico (**figura 1**), tomado de (Chen et al., 2005), que representa las relaciones de ortólogos y parálogos después de dos eventos de especiación y dos de duplicación. El árbol representa genomas que son obtenidos mediante estos eventos, los cuales son G_1= (A_1), G_2= (B_1, C_1) y G_3= (B_2, C_2, C_3). Todos los genes en G_2 y G_3 son co-ortólogos al gen A_1. Los genes B_1 y C_1 son “outparalogs” con respecto a G_3 (ejemplo, segunda

especiación) y son inparalogs con respecto a G_1 (ejemplo, la primera especiación). El gen C_2 es un descendiente directo (ejemplar verdadero) del gen C antecesor, mientras que C_3 no lo es si es duplicado desde C_2.

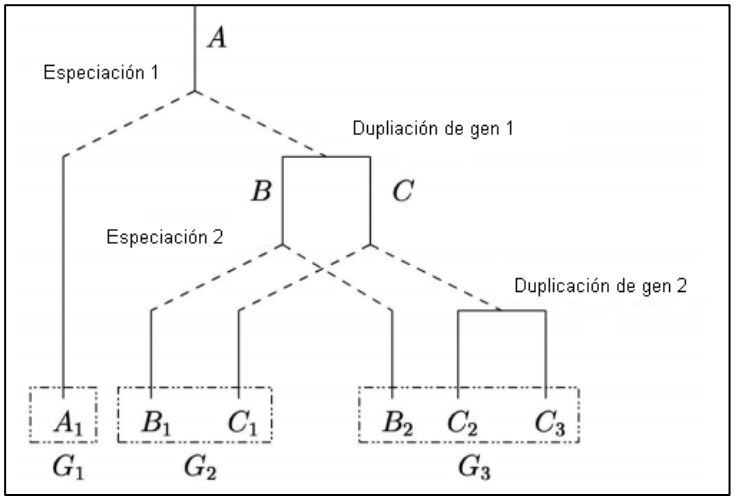


Figura 1: Relación de ortología y paralogía entre 3 especies.

Muchas son las técnicas informáticas creadas para dar respuesta a estudios sobre la ortología. Aún, los científicos consideran la informática como una vía rápida y menos costosa de reproducir experimentos e interpretar sus resultados desde una perspectiva funcional y evolutiva. En los últimos años, se ha incrementado el número de herramientas a favor del desarrollo de la bioinformática, determinadas por variados enfoques para la detección de ortólogos, como los métodos basados en árboles, en grafos y los híbridos.

1.2 Métodos de detección de ortólogos basados en grafos

Como los genes ortólogos están relacionados con la historia evolutiva, la detección de estos, está frecuentemente relacionada con los árboles filogenéticos. Su construcción demanda grandes recursos computacionales. Este enfoque requiere de la obtención de homólogos, la generación de múltiples alineamientos correctos para cada grupo de dominios homólogos, la construcción de un árbol filogenético por cada grupo y finalmente la extracción de ortólogos (Remm et al., 2001).

En cambio, los métodos basados en grafos son adecuados para la inferencia de ortología de uno o muchos genomas. A diferencia de los métodos basados en árboles, no construyen alineamientos múltiples de secuencias, ni árboles filogenéticos. Se basan principalmente, en el cálculo de similitud entre pares de secuencias (conocida como comparación par a par o "all versus all") y las definiciones operacionales de ortología. Realizan un preprocesamiento

de los datos podando los grafos bipartitos creados antes de pasar a la fase de agrupamiento del grafo para encontrar los grupos de ortólogos.

Tanto Towfic (Towfic et al., 2009), Kamvysselis (Kamvysselis, 2003) como Ozer (Ozer et al., 2004), aportan métodos híbridos que utilizan árboles y grafos para combinar la similitud de las secuencias con información sobre la interacción entre las proteínas, la información de "synteny" y los dominios de proteínas, respectivamente. Específicamente, el término "synteny" abarca la información de "conserved synteny" (bloques de orden conservados), es decir, las regiones genómicas ortólogas que contienen genes ortólogos en el mismo orden colinear.

La información de los "conserved synteny", sólo es aplicable a especies cercanas en la evolución. A pesar de esto, puede influir en la búsqueda de ortólogos cuando la inferencia de la homología está dañada por una baja similitud de de las secuencias, o en la distinción de verdaderos ortólogos a partir de parálogos ("outparalogs") en la presencia de pérdidas recíprocas de genes (Kuzniar et al., 2008). Este autor también acota, que la mayoría de los ortólogos tienden a ser encontrados en los bloques de orden conservado, específicamente, si la razón de reordenamientos genómicos es baja.

En esta investigación, se analiza la semejanza a nivel de aminoácidos de las secuencias de genes de dos genomas de distintas especies, tratando de combinarla con la información de bloques conservados y la relación evolutiva entre estos genes, con el objetivo de inferir la relación de ortología. Se utilizó el enfoque basado en grafo por las facilidades que brinda para representar la información de similitud entre los genes. A continuación se exponen las ideas fundamentales de los métodos basados en este enfoque consultados en la bibliografía.

1.2.1 Nearest neighbour

El término "Nearest neighbour", se utiliza para la designación colectiva de todos los enfoques que aplican una definición operacional de ortología como "Best Hit" (BeT), "Reciprocal Best Hit" (RBH), "Symmetrical Best Hit" (SymBeT) y "Reciprocal Smallest Distance" (RSD) para podar el grafo antes de aplicar técnicas de particionado. La definición operacional es comúnmente el primer paso, para encontrar posibles ortólogos, teniendo en cuenta algunas de las mejores coincidencias entre genomas completos de dos especies, aún cuando no garanticen la proximidad filogenética. Algunos métodos basados en grafos, usan la técnica de agrupamiento para extender "nearest – neighbour" a más de dos especies y construir grupos de ortólogos multi-especies (OGs). Estos enfoques usan la definición de ortología liberalmente, porque los ortólogos y los parálogos están a menudo contenidos en un mismo OG (Kuzniar et al., 2008).

BeT: es la mejor correspondencia de BLAST ("Best BLAST hit"). Los pares de secuencias seleccionadas por esta definición son aquellas con mejor puntuación obtenida del alineamiento según BLAST (Tatusov et al., 1997).

BBH: Significa mejores correspondencias bidireccionales. El autor en (Overbeek et al., 1999) presenta este método para la detección de grupos conservados de genes sobre la base de la definición de "run". Un conjunto de genes en un cromosoma se considera como un "run" si y solo si todos estos genes se encuentran en la misma hebra ("strand") y los "huecos" entre genes adyacentes son de 300 pares de bases o menos. Cada par de gen en un "run" se denomina "close". Dos genes x_a y x_b de dos genomas *A* y *B*, x_a y x_b son un BBH, si y sólo si existe una similitud reconocida entre ellos (ejemplo: scores de FASTA menores que 1.0×10^{-5}), y no existe un gen z_b en *B* que sea más similar que x_b a x_a y no existe un gen z_a en *A* que sea más similar que x_a a x_b. Los genes (x_a, y_a) de *A* y (x_b,y_b) de *B* forman un par de "close bidirectional best hits" (mejores correspondencias bidireccionales cercanas) si y sólo si, x_a y y_a forman un "close", x_b y y_b forman un "close", x_a y x_b son un BBH y y_a y y_b son un BBH.

BBH busca por pares de genes los que mejores se relacionan y los marca como ortólogos. Como defecto tiene que en el caso de una reciente duplicación de genes, éstos son igualmente marcados como ortólogos, sin señalar la presencia de homólogos adicionales. De esta forma, no se dan garantía de que las coincidencias uno a uno representen relaciones de ortólogos, estableciéndose incorrectas relaciones entre pares de genes (Kamvysselis, 2003).

RBH: constituye una mejor correspondencia recíproca (Hirsh and Fraser, 2001). La proteína *x* en el genoma *A* es una mejor correspondencia recíproca de la proteína *y* en el genoma *B*, si la búsqueda hacia delante del genoma *B* con la proteína *x* produce como correspondencia tope la proteína *y*, y la consulta recíproca del *A* con la proteína *y* produce como correspondencia tope la proteína *x*. En (Wall et al., 2003), los autores comentan un fallo potencial de RBH en el sentido de que si el BLAST hacia delante produce una mejor correspondencia de parálogos, sin tener en cuenta si el BLAST recíproco corrige el error recuperando un ortólogo real, entonces ambos pares serán excluidos. De esta forma, RBH evita la admisión de pares de parálogos en el conjunto de proteínas para las cuales se estima la razón evolutiva, sin embargo, pudiera excluir erróneamente un par de ortólogos auténtico. A pesar de esta posible limitación, varias herramientas de detección de ortólogos reportados en la literatura, que aparecen referenciados en las siguientes secciones, usan este método.

RSD: representa a la menor distancia recíproca (Wall et al., 2003). El algoritmo emplea BLAST como primer paso, comenzando por un genoma *B*, y una secuencia de consulta *x*, perteneciente al genoma *A*. Se obtiene un conjunto de correspondencias *H*, que exceden un umbral de significación predefinido (por ejemplo $E < 10^{-20}$). Luego se realiza un alineamiento por separado de cada secuencia de proteínas en *H* con la secuencia de consulta original *x*, usando un

programa de alineamiento múltiple. Si cada región alienable de las dos secuencias excede una fracción del umbral de la longitud total del alineamiento, (valor de corte de 0.8), se obtiene un estimado de máxima verosimilitud del número de sustituciones de aminoácidos que separan las dos secuencias de proteínas dada una matriz de razón de sustitución de aminoácidos empírica.

El modelo bajo el cual se obtiene un estimado de máxima verosimilitud, puede incluir una variación en la razón evolutiva entre sitios de las proteínas. Para comparaciones más distantes, asumen generalmente una distribución gamma con parámetro de forma $\alpha = 1.53$. Entre todas las secuencias en *H* para las cuales se estiman la distancia evolutiva, sólo se retiene *y*, la secuencia que produce la menor distancia. Entonces, *y* se utiliza para un BLAST recíproco contra el genoma *A*, recuperando un conjunto de correspondencias de alta puntuación *L*. Si una correspondencia en *L* es la secuencia de consulta original *x*, la distancia entre *x* e *y* se recupera del conjunto de mejores distancias calculadas previamente. Las restantes correspondencias de *L* son alineadas por separado y con *y* se calcula un estimado de la distancia de máxima verosimilitud para estos pares. Si la secuencia de proteína *L*, que produce la menor distancia hacia *y*, es la secuencia de consulta original *x*, se asume que se ha encontrado un par de ortólogos verdadero, se retiene *y* al igual que su distancia evolutiva.

1.2.2 Inparanoid

La primera versión del programa (Remm et al., 2001), identifica ortólogos e "inparalogs" entre dos genomas, pero no reporta los "outparalogs". La metodología puede ser vista como una extensión de la técnica "all versus all", pero con reglas especiales para el análisis del agrupamiento, con el fin de extraer todos los "inparalogs". Este algoritmo ha sufrido modificaciones de parámetros con el propósito de obtener algoritmos más fuertes y específicos. De esta manera se desarrolla, la versión 7 del algoritmo, la cual utiliza un filtro de baja complejidad, para enmascarar sólo durante la siembra, pero no durante la extensión, constituyendo un enmascaramiento suave. Esto restringe el filtrado de baja complejidad, permitiendo bajar el umbral de puntuación de 50 a 40 bits. (Östlund et al., 2010).

Sin embargo, las coincidencias aceptadas en el primer paso se han reajustado utilizando BLAST con ajuste, antes de que los criterios de solapamiento sean aplicados para evitar el efecto de alineamientos cortos. La distancia entre el primero y el último residuo alineado debe ser igual o superior al 50% de la longitud de la secuencia. Además, la suma de las longitudes de las regiones alineadas de la secuencia debe ser igual o superior al 25% de la longitud de la secuencia. Cuando hay múltiples pares de segmentos de alta puntuación, el algoritmo exige que mantengan el mismo orden relativo en ambas secuencias, y que no se solapen por más del 5% (Östlund et al., 2010).

El programa requiere la entrada de dos ficheros con secuencias de proteínas de genomas *A* y *B* en formato FASTA, opcionalmente, permite utilizar un tercer genoma como un grupo externo y así detectar las secuencias que faltan y mejorar la detección de ortólogos. La búsqueda BLAST, comienza con el cálculo de las puntuaciones de similitud entre todas las secuencias. Estas son calculadas en cuatro pasos: *A* contra *B*, *B* contra *A*, *A* contra *A* y *B* contra *B*.

Si un grupo C es utilizado como externo, se calculan las puntuaciones de similitud por pares de *A* contra *C* y *B* contra *C*. El programa BLAST reporta ocasionalmente, puntuaciones asimétricas entre pares de secuencias *x-y* e *y-x*. Para evitar los problemas que los resultados asimétricos podrían causar en los pasos posteriores, todos los puntajes se promedian por parejas. Se aplican dos valores de cortes por cada coincidencia por pares. El valor de corte de puntuaciones, es necesario para separar las puntuaciones significativas de falsas coincidencias, reduciendo la cantidad de datos y el valor de corte de solapamiento se aplica para evitar pequeñas coincidencias a nivel de dominio de proteínas (Remm et al., 2001).

Inparanoid 7, tiene en cuenta las relaciones de uno a muchos y de muchos a muchos entre dos genomas. Los valores de confianza son asignados a "inparalogs" individuales y grupos de ortólogos como un todo. El programa y la base de datos se pueden descargar gratis. Se limita a la comparación de genomas completos en pareja y no permite los grupos solapados en presencia de una proteína híbrida (Kuzniar et al., 2008).

1.2.3 La base de datos Clusters of Orthologous Groups (COG)

El método BeT, propone que un mejor éxito en el genoma destino significa que la proteína en este genoma es más similar a la proteína dada de otro genoma. En comparaciones de múltiples genomas, pares potenciales de ortólogos son identificados mediante BeTs, que pueden unirse para formar grupos de ortólogos representados en todo o en un subconjunto del genoma analizado (Tatusov et al., 1997). Este enfoque posee dos complicaciones. Primeramente, muchas proteínas han evolucionado mediante la duplicación después de la divergencia de las especies comparadas. En estos casos, descifrar relaciones de co-ortólogos puede resultar una tarea difícil y los grupos de ortólogos que incluyen ampliaciones deben ser tratados de forma particular. La segunda complicación es causada por el hecho de que muchas proteínas existen en forma de multidominio, codificada por un solo gen en varias especies y como producto de dos o más genes independientes en otras (Tatusov et al., 1997).

El enfoque para identificar conjuntos de proteínas ortólogas basado en una consistente agrupación mediante BeTs, ha sido implementado en la colección de proteínas COGs. El protocolo de construcción COG incluye un procedimiento automático para la detección de grupos de ortólogos candidatos, la división manual de proteínas multidominio en los dominios de componentes, y la posterior preservación manual y anotación. COGs comienza con seis genomas

procariotas y un genoma eucariota unicelular (Tatusov et al., 1997). La actualización realizada en el 2001, incrementa el número de genomas procariotas a 43 (Tatusov et al., 2001).

La actualización del sistema desarrollado para la delimitación de clústeres de grupos de ortólogos de proteínas (COG), a partir de genomas procariotas secuenciados y eucariotas unicelulares, realizada en el 2003, incluye la construcción de grupos de ortólogos previsto para genomas eucariotas llamado KOGs (grupos de ortólogos eucariotas). Para la construcción de éstos, se emplea el mismo procedimiento utilizado en los genomas procariotas (Tatusov et al., 2003).

El enfoque COG, no hace diferenciación entre "inparalogs" y "outparalogs" automáticamente. El procedimiento automático de agrupamiento crea grupos exclusivos, en consecuencia, las proteínas multidominio deben ser manejadas manualmente (Kuzniar et al., 2008). Es incapaz de distinguir los eventos de duplicaciones recientes, de los eventos anteriores de duplicación, por lo que no es aplicable debido a que el genoma S. cerevisiae, contiene cientos de pares de genes que se duplicaron antes de las divergencias de las especies. Tampoco distingue entre dos copias de genes antiguos duplicados por lo que muchas relaciones de ortólogos no son detectadas (Kamvysselis, 2003).

1.2.4 MultiMSOAR 2.0

MCL: Algoritmo de agrupamiento de Markov, que encuentra grupos a través de rondas iterativas de expansión e inflación y que promueve las regiones fuertemente conectadas, debilitando las regiones escasamente conectadas (Hwang et al., 2006).

MSOAR: Se basa en la reorganización del genoma, es un sistema de asignación de ortólogos de alto rendimiento y muestra como se corresponden cada uno en el camino evolutivo de especiación después de la duplicación (Shi et al., 2011).

MultiMSOAR: Se basa en árboles para la asignación de ortólogos entre múltiples genomas, usando el método de agrupamiento simple basado en los resultados por pares del MSOAR. Sólo considera los grupos de ortólogos que no tienen pérdidas de genes en alguna especie. Es aceptado para especies estrechamente relacionadas, pero es muy restrictivo cuando se consideran más especies, debido a que deben tener nuevos genes y pérdidas de genes, así cómo duplicaciones de su historia evolutiva (Shi et al., 2011).

MultiMSOAR 2.0 se utiliza para identificar grupos de ortólogos de múltiples genomas, es una extensión del MSOAR 2.0 y presenta un nuevo aprovechamiento combinatorio para la construcción de grupos de ortólogos. En comparación

con el MultiMSOAR (versión anterior), permite la pérdida de genes dentro de grupos de ortólogos y los grupos de ortólogos contienen genes sólo de un subconjunto de los genomas. Minimiza el número de nuevos genes, pérdidas de genes y duplicaciones dentro de una familia de genes en la asignación de los grupos de ortólogos (Shi et al., 2011).

Además, construye familias de genes para todos los primeros genomas mediante la búsqueda de similitud de secuencia y el algoritmo MCL. Luego, aplica MSOAR 2.0 para encontrar pares de ortólogos entre todos los pares de genomas. Construye un grafo ponderado usando la información de los pares de ortólogos y la similitud de la secuencia entre cada par de ortólogos, e intenta encontrar el peso máximo para cada familia de genes. Luego, divide cada conjunto en subconjuntos disjuntos de genes ortólogos (SOGs). Cada SOG puede consistir en muchos grupos de ortólogos. Las etiquetas del MultiMSOAR 2.0 del árbol de especies indican 1 ó 0 en dependencia de si el SOG contiene un gen de las especies correspondientes o no (Shi et al., 2011).

El árbol resultante es llamado árbol de grupos ortólogos (TOGs). Luego se utilizan dos algoritmos diferentes (NodeCentric y el TreeCentric) para etiquetar los nodos internos de cada TOG sobre la base del principio de parsimonia y contrastes biológicos. Tiene la ventaja de que provee más información sobre los nacimientos de genes y que es una herramienta para identificar grupos de ortólogos de múltiples genomas que están relativamente cercanos. Sin embargo, cuando se incrementa el número de las duplicaciones de los genes, disminuye la precisión de la predicción, siendo esta su desventaja principal (Shi et al., 2011).

1.2.5 OrthoInspector

OrthoInspector (Linard et al., 2011), incorpora un algoritmo original para la detección rápida de las relaciones de ortólogos e "inparalogs" entre las diferentes especies. En comparación con los métodos existentes, mejora la sensibilidad de la detección, con una pérdida mínima de especificidad. Además, incorpora varias herramientas de visualización para facilitar los estudios en profundidad sobre la base de estas predicciones. El algoritmo está divido en tres pasos fundamentales. Primero, los resultados "all versus all" de BLAST, son proporcionados por el usuario y se analizan para encontrar todos los mejores resultados de BLAST de cada proteína, con el propósito de crear los grupos de "inparalogs". Segundo, los grupos de "inparalogs" son comparados en cada organismo por pares para definir el potencial de los ortólogos y/o "inparalogs". Tercero, con los mejores resultados se verifica el potencial de la ortología.

Orthoinspector se utiliza como una herramienta de partida para inferir las relaciones de ortología, ya que su sensibilidad y especificidad están bien equilibrados, también la inferencia de ortología es menos intensiva computacionalmente que Ensembl Compara, el único otro método que logra resultados similares (Linard et al.,

2011). Está codificado en Java 1.6.x, requiere de un bajo respaldo de base de datos para manejar la enorme cantidad de datos producidos por el análisis "all versus all". Proporciona dos interfaces de usuario diferentes, un cliente de líneas de comandos y una interfaz gráfica. Sin embargo, necesita tener actualizada la base de datos, mejorar la tasa de secuenciación e incorporar el proceso de actualización incremental.

1.2.6 OrthoList

OrthoMCL: Proporciona un método escalable para la construcción de grupos de ortólogos de eucariotas, utilizando el algoritmo de agrupamiento MCL (Li et al., 2003).

HomoloGene: Es un sistema para la detección automatizada de homólogos entre los genes anotados de varios genomas eucariotas completamente secuenciados (NCBI, 2011).

modENCODE: Es una herramienta computacional, dirigida a identificar y analizar todos los elementos funcionales que comprende el genoma en los organismos modelos: Drosophila y C. elegancia (Elements, 2012).

OrthoList constituye una lista de ortólogos C.elegans de genes humanos y es el resultado de un meta-análisis de la comparación de 4 métodos (Inparanoid, OrthoMCL, HomoloGene y Ensembl Compara), donde se obtuvo una lista de 7.663 genes de codificación única. La lista representa un ~ 38% de 20 250 de la predicción de la codificación de proteínas de genes en C. elegans (Shaye and Grenwald, 2011). Es sensible y específica en la detección de ortólogos. Es eficaz en la predicción de genomas, lo cual depende de la exactitud de los modelos de genes de los genomas bajo una anotación del mismo, tanto en C. elegans como en los seres humanos, es un proceso continuo que se basa en diversos criterios. Entre los métodos de consulta para el meta-análisis, utiliza versiones más recientes de la secuencia del genoma como parte de la información que ofrece modENCODE (Shaye and Grenwald, 2011).

OrthoList es una lista que necesita ser actualizada, a pesar de que las secuencias más conservadas ya están representadas en el genoma. Genera una lista de productos sensibles y específicos de los genes C. elegans con genes humanos homólogos para agilizar las pantallas funcionales genéticas hacia adelante. (Shaye and Grenwald, 2011).

1.2.7 BUS

El algoritmo de detección de ortólogos BUS (Kamvysselis, 2003) se basa en las debilidades de BBH y COG para resolver la correspondencia del cruce de genes a través de las especies. Parte de la construcción de un grafo bipartido completo pesado a partir de la similitud obtenida de la comparación par a par entre genes de dos genomas. Representa la mejor coincidencia de cada gen como un conjunto de genes, en lugar de un único mejor éxito, que hace que sea más robusto a ligeras diferencias en la similitud de secuencia.

En la etapa de preprocesamiento de los datos, se eliminan todos los arcos de un nodo con menos del 80% del peso máximo de los arcos que conectan a este, esta poda es realizada al grafo que caracteriza la similitud de la secuencia en cuanto a la homología. Sobre la base de las similitudes sin ambigüedades que resultan en esta etapa, se construyen bloques de orden conservados (synteny blocks) de genes, cuando los genes vecinos en una especie tienen coincidencia de uno a uno con los genes vecinos en las otras especies. Se utilizan estos bloques de orden conservado para resolver ambigüedades adicionales, preferentemente manteniendo las coincidencias.

En la fase de agrupamiento se separa este grafo en subgrafos cada vez más pequeños hasta que las únicas coincidencias conectan los verdaderos ortólogos. Usa una técnica de particionado de grafo donde se divide el grafo en partes aproximadamente del mismo tamaño, de manera que prevalezcan pocas conexiones entre los grupos creados. En el caso de un grafo pesado, la partición se encamina a minimizar la suma de los pesos de los arcos implicados en el corte. Busca los subconjuntos de genes que son óptimos a nivel local, de tal manera que todas las mejores coincidencias de los genes dentro del grupo están contenidas dentro del grupo, y no fuera de este. Estos mejores subconjuntos no ambiguos (BUS), aseguran que el grafo bipartido sea separable máximamente, mientras que mantiene todas las posibles relaciones de ortólogos. Finalmente, el algoritmo BUS proporciona una buena solución para determinar las correspondencias de genes entre dos genomas, que funciona bien en un rango de distancias evolutivas.

1.3 La detección de ortólogos basada en grafo con agregación de medidas de similitud.

Bergmann define el significado de medida de similitud, como una función que mide la similitud y es expresada como un valor numérico (Bergmann, 2002). Este autor también define el principio local-global describiendo una medida global en todos los casos y una medida local a nivel de atributo. Plantea además, que la medida global debe tener un carácter pragmático y reflejar la importancia, la relevancia y la utilidad de los aspectos de la similitud, mientras que la local se encarga de los detalles de carácter técnico y de dominio, y es una tarea independiente.

En general, la detección de ortólogos entre dos genomas con enfoque de grafo, comienza con el cálculo de todos los valores de similitud de las secuencias por pares de genes de diferentes especies, a partir del alineamiento par a par. Luego se aplica una definición operacional de ortología con una estrategia de poda, seguidamente se aplica un algoritmo de agrupamiento como el algoritmo de agrupamiento de Markov, la mínima partición común, la descomposición de ciclo máximo (Chen et al., 2005) o el algoritmo BUS propuesto por (Kamvysselis, 2003).

Como alternativas de uso para una definición operacional de ortología en la solución propuesta en esta investigación, se define un método basado en la similitud, con medidas de distancia aplicadas a genes de diferentes genomas. Las

mismas caracterizan la homología entre secuencias, la longitud de las secuencias, la pertenencia a los bloques colineales locales y la relación evolutiva basada en el modelo "SG2009" de (Sánchez and Grau, 2009).

Para modelar la disimilitud entre dos genes x e y, ésta se divide en funciones de distancias locales para cada una de las características anteriormente mencionadas. Luego, la información de todas las distancias halladas, son agregadas en una función de distancia global. A continuación se explica cada una de las distancias aplicadas en la presente investigación.

A partir de un gen x perteneciente al genoma A y un gen y perteneciente al genoma B, se define una distancia local entre estos genes como sigue:

$d_i: AXB \rightarrow \mathcal{R}^+$

Específicamente:

$d_i: AXB \rightarrow [0,1]$

Que cumple con las siguientes propiedades:

La distancia entre dos puntos es nula si y solo si los dos puntos son coincidentes.

$d_i(x, y) = 0 \Leftrightarrow x = y$

La distancia entre dos puntos cumple la propiedad de simetría. Esto significa, que la distancia entre los puntos x e y es igual a la distancia entre y e x.

$d_i(x, y) = d_i(y, x)$

Las distancias entre tres puntos cumplen con la "desigualdad triangular".

$d_i(x, y) + d_i(y, z) \geq d_i(x, z)$

1.3.1 Función de distancia local que describe la homología entre pares de secuencias

Tomando en cuenta las razones expuestas en la sección 1.1.1, se considera que el algoritmo de SW es una buena elección para el cálculo del alineamiento entre pares de secuencias con vistas a medir su grado de similitud. Luego se define el grafo $G(V, E)$ bipartido completo no dirigido, pesado por los "bitscore" generados por el algoritmo SW, donde el conjunto V tiene orden $n = n_1 + n_2$ siendo n_1 el total de genes en el genoma A y n_2 el total de genes en el genoma B. Este grafo se poda teniendo en cuenta el peso de la mejor conexión de cada nodo, quitando aquellas relaciones de similitud menores que el 80% del peso de la mejor conexión.

A partir de la puntuación óptima en bits del alineamiento local, se define un coeficiente de asociación ca para la función local de disimilitud d_1, donde a y b constituyen las secuencias de aminoácidos de los genes x e y respectivamente.

sw: es la puntuación óptima en bits del alineamiento local entre las secuencias de aminoácidos a y b.

n: Cantidad de genes del genoma A.

m: Cantidad de genes del genoma B.

$$ca(a,b) = \frac{sw(a,b)}{\text{maximo}(sw(a_i,a_j))}\ , i = 1,\dots,n, j = 1,\dots,m \quad (1)$$

$$d_1(a,b) = \begin{cases} 1 - ca(a,b) & si\ \ ca(a,b) > 0 \\ 1 & si\ \ ca(a,b) \leq 0' \end{cases} \quad (2)$$

1.3.2 Función de distancia local que describe la longitud entre pares de secuencias

Se define la longitud entre secuencias, como una función de distancia local d_2, la cual es una diferencia normalizada (Duch., 2000).

$longitud$: es el tamaño de una secuencia de aminoácidos.

$maxima\_longitud$: Es la longitud de la cadena de aminoácidos de mayor tamaño.

$minima\_longitud$: Es la longitud de la cadena de aminoácidos de menor tamaño.

x_p: Proteína o secuencia de aminoácidos del gen x perteneciente al genoma A.

y_p: Proteína o secuencia de aminoácidos del gen y perteneciente al genoma B.

$$d_2(x_p, y_p) = \frac{|longitud(x) - longitud(y)|}{maxima\_longitud(z_i) - \ minima\_longitud(z_i)} \quad (3)$$

1.3.3 Función de distancia local que describe la pertenencia a los "Locally Collinear Blocks" (LCB).

Dado que la recombinación genética puede producir reordenamientos del genoma en la evolución, las regiones de ortólogos se pueden reordenar o invertir en relación con otros genomas. Debido a esto, se utiliza el software MAUVE en su versión 2.3.1, que permite identificar segmentos conservados en las secuencias que no parecen estar alterados por reordenamientos del genoma. En el conjunto de LCB inicial seleccionado, algunas serán correspondencias aleatorias que aparecen generalmente como pequeños reordenamientos de genes que son

descartados en el alineamiento final. Los LCB en el alineamiento final, estarán compuestos por aquellas regiones consideradas como verdaderamente homólogas (Darling et al., 2004).

En la investigación propuesta se considera, que los genes que pertenecen al mismo LCB, son más probables a ser ortólogos. A partir de un estudio realizado por el autor, con los datos de los genomas que se usan en la presente investigación, se estimó un promedio de la cantidad de bases que comparten un gen y un LCB para que el primero pertenezca al segundo, como resultado se determinó, que el promedio de bases que tiene un gen en un LCB es uno, seleccionando este valor para la clasificación de pertenencia de un gen a un LCB. En consecuencia, se define la función de distancia d_3, que constituye un rasgo binario entre dos genes de diferentes genomas, que mide la disimilitud entre conjuntos. Según (Hubalek, 1981), el conjunto de medidas que generalmente brindan buenos resultados para conjuntos de datos binarios está formado por: el Coeficiente de Dice-Sorencen, el Coeficiente de Kulczynski, el Coeficiente de Driver-Kroeber-Ochiai y el Coeficiente de Jaccard, siendo este último el más utilizado por los investigadores. En ese trabajo se elige el Coeficiente de Jaccard (Jaccard, 1901) para la comparación de los genes en cuanto a la pertenencia a los *LCB*.

$matLCB: A \times B \rightarrow \{0,1\}^j$,

j: $1, \dots, k$ (Cantidad total de LCB)

i: $1, \dots, n+m$ (Es la cantidad de genes de ambos genomas)

z: es un gen que puede ser del genoma A o del B. Los primeros n elementos pertenecen al genoma A. Los elementos de $n+1$ hasta m pertenecen al genoma B.

LCB: representa los LCB obtenidos.

$$matLCB(i,j) = \begin{cases} 0 & si \quad z_i \notin LCB_j \\ 1 & si \quad z_i \in LCB_j \end{cases} \quad (4)$$

$$d_3(x,y) = 1 - CJaccard(x,y,matLCB) \quad (5)$$

El valor del coeficiente de Jaccard oscila entre 0 y 1, y es igual que el número de bits en el correspondiente par de vectores binarios en $matLCB$, se divide por el número de bits de cualquiera de los vectores. Esta medida de distancia es llamada la distancia de Soergel y satisface la desigualdad triangular (Lipkus, 1999). Además, representa el porcentaje de coordenadas distintas de cero que difieren en vectores de genes x e y en $matLCB$.

A partir del estudio del artículo (Sánchez and Grau, 2009) y en conjunto con estos investigadores, se definieron dos variantes de distancias genéticas evolutivas, la primera tiene en cuenta la secuencia de aminoácidos y la segunda la secuencia de nucleótidos. Los próximo dos epígrafes caracteriza las dos medidas de distancia definidas.

1.3.4 Función de distancia genética evolutiva teniendo en cuenta la secuencia de aminoácidos de genes

$a_p = (a_1, a_2, \dots, a_L)$
$b_p = (b_1, b_2, \dots, b_L)$

a_p y b_p son las cadenas resultantes en el alineamiento de la proteína de los genes x e y respectivamente.

Como ambas secuencias alineadas tienen la misma longitud se tiene que:

$$\mathrm{L} = len(a_p) \quad (6)$$

C_{a_i}
C_{b_i}

C_{a_i} es un conjunto cuyos elementos son todos los codones que producen al aminoácido a_i, C_{b_i} es un conjunto cuyos elementos son todos los codones que producen al aminoácido $\mathrm{b_i}$.

La variable x_{n_i} representa un nucleótido i perteneciente al gen x del genoma A, y_{n_i} representa un el nucleótido i perteneciente al gen y del genoma B. La distancia entre dos bases de tripletes extendidos (dos codones) $x_{n_1}x_{n_2}x_{n_3}$ e $y_{n_1}y_{n_2}y_{n_3}$ es calculada mediante la fórmula:

$$d_c\left((x_{n_1}, x_{n_2}, x_{n_3}), (y_{n_1}, y_{n_2}, y_{n_3})\right) = \frac{|x_{n_1} - y_{n_1}|}{5} + |x_{n_2} - y_{n_2}| + \frac{|x_{n_3} - y_{n_3}|}{25} \quad (7)$$

Donde x_{n_i}, y_{n_i} pertenecen al alfabeto extendido del ADN, siendo D una o más bases alternativas o "gaps" (Smith and Waterman, 1981 G, U), A (adenina), C (citocina), G (guanina) y U (uracilo). Se realiza una transformación que propone (Sánchez et al. 2009) en su artículo como sigue:

$D = 0, A = 1, C = 2, G = 3, U = 4.$

La distancia entre dos aminoácidos, incluyendo los gaps, se calcula mediante el promedio de las distancias d_c entre cada par de codones que la producen, siguiendo el conjunto extendido de nucleótidos.

$$d_a(a_i, b_i) = \frac{1}{z_1 * z_2} \sum_{j=1}^{z_1} \sum_{k=1}^{z_2} d_c\left(C_{a_{i_j}}, C_{b_{i_k}}\right) \quad (8)$$

$$z_1 = |C_{a_i}| \quad (9)$$

$$z_2 = \left|C_{b_i}\right| \quad (10)$$

$$C_{a_{i_j}} \in C_{a_i}$$

$$C_{b_{i_k}} \in C_{b_i}$$

$$d_4(a_p, b_p) = \frac{1}{L}\sum_{k=1}^{L} d_a(a_{p_k}, b_{p_k}) \quad (11)$$

1.3.5 Función de distancia genética evolutiva teniendo en cuenta la secuencia de nucleótidos

$a_p = (a_1, a_2, \dots, a_L)$

$b_p = (b_1, b_2, \dots, b_L)$

a_p y b_p son las cadenas resultantes en el alineamiento de la proteína de los genes x e y respectivamente.

La traducción de un aminoácido a un codón es una relación de uno a muchos, al reverso, si constituye una relación de uno a uno. Sin embargo, no se parte de la secuencia de nucleótidos debido a que algunos genes no tienen longitud múltiplo de 3 afectando la selección de los codones. Por tal motivo, se parte de la secuencia de aminoácidos del gen y se obtiene la secuencia de nucleótidos a partir de la función **aa2nt** implementada en Matlab (Matlab, 2010). En la secuencia de aminoácidos donde hay un "gap", aparecen tres "gaps" en la secuencia de nucleótidos. Esto quiere decir, que las dos secuencias de aminoácidos se convierten a secuencias de nucleótidos. Siendo x_n o y_n las secuencias de nucleótidos alineados obtenidos en la operación anterior y f la cantidad de tripletes o codones.

$$d_5(x_n, y_n) = \frac{1}{f}\sum_{i=1}^{f} d_c\left(\left(x_{i_1}, x_{i_2}, x_{i_3}\right), \left(y_{i_1}, y_{i_2}, y_{i_3}\right)\right) \quad (12)$$

Donde d_c se calcula según la fórmula (7). Se tiene en cuenta la misma transformación, del alfabeto de 6 letras a un alfabeto de 6 números, anteriormente mencionada.

Por cada distancia d_i, definidas anteriormente, se construye un grafo $G_i(V, E)$ bipartido completo no dirigido, donde cada arista de E es pesada por las distancias obtenidas entre cada par de genes y el conjunto de vértices V tiene $n = n_1 + n_2$ elementos, siendo n_1 los genes del genoma A y n_2 los genes del genoma B. Cada arista conecta sólo los vértices del conjunto A con los vértices del conjunto B.

Ya analizadas las 5 funciones de distancias y los grafos bipartidos completos pesados, que representan las relaciones de distancias entre cada par de genes, se proponen dos variantes para lograr la combinación de estos rasgos.

1.3.6 Variante 1 de agregación de distancias con Operador OWA.

En los modelos tradicionales de distancias entre objetos, se suele utilizar ponderaciones simples o basadas en medias ponderadas que reflejan la importancia de cada característica en el problema. En muchos casos, puede que esta ponderación no sea suficiente, debido a que se necesite otro tipo de medida que permita englobar a un mayor número de casos. Esto se puede conseguir con la utilización de los operadores OWA, que facilitan la obtención de una amplia gama de distancias parametrizadas entre la distancia mínima y la distancia máxima del problema. De esta forma, en función de los intereses del análisis, se le podrá dar una mayor importancia a las distancias menores o mayores del problema según el tipo de ponderación utilizado (Lindahl, 2008).

El operador OWA fue introducido por Yager (Yager, 1988), con el objetivo de agregar información. El autor lo define a partir del vector de información $\mathrm{V_1} = (\mathrm{a_1}, \mathrm{a_2}, \dots, \mathrm{a_n})$, donde $a_i \in [0,1]$ y $1 \leq i \leq n$, y el vector peso $W = (w_1, w_2, \dots, w_n)$. Después de reordenar descendentemente los elementos en V_1, se obtiene el vector $V_2 = (b_1, b_2, \dots, b_n)$, donde b_j es el j-ésimo más grande de V_1. Finalmente, define el operador OWA como una función de asignación F:

$F: I^n \to I, I = [0,1]$.

La función F se defino como:

$$\mathrm{F}(\mathrm{a_1}, \mathrm{a_{2,}} \dots, \mathrm{a_n}) = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{w_i b_i} = \mathrm{w_1 b_1} + \mathrm{w_2 b_2} + \cdots + \mathrm{w_n\, b_n}. \quad (13)$$

Donde $w_i \in [0,1]\ y\ \sum_{i=1}^{n} w_i = 1$. El peso w_i no está asociado con un argumento a_1 en particular, pero si con un particular orden de posición i de los argumentos. Esto quiere decir que w_i es el peso asociado con el i-ésimo argumento más grande (Yager, 1988).

Determinar los pesos del operador OWA se ha vuelto una tarea crítica. Muchas variantes han surgido para realizar el cálculo de los vectores de peso, una de las más sencillas de utilizar es el método funcional introducido por Yager (Yager, 1996a) para los operadores OWA, resultando de gran utilidad para obtener las ponderaciones. El autor lo define como la función:

$Q: [0,1] \to [0,1]$, $Q(0) = Q(1)$ y $Q(x) \geq Q(y)$ para $x > y$.

Esta función se conoce como "basic unit interval monotonic function" (BUM). Utilizando esta función se puede obtener las ponderaciones w_j para $j = 1 \dots n$ de la siguiente forma:

$$w_j = Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \quad (14).$$

Este método garantiza que la suma de las ponderaciones es siempre 1, esto quiere decir que $w_1 + w_2 + \cdots + w_n = 1$ y que $w_j \in [0,1]$.

Yager (Yager, 1996b) sugiere un enfoque para hallar el vector de peso a través de un cuantificador regular no decreciente. Esto quiere decir que Q se considera un cuantificador regular monótono creciente (Regular Increasing Monotone, RIM por sus siglas en inglés). En Singh (Singh, 2011) se considera una familia de cuantificadores RIM, tomando a $\alpha = 2$, como:

$$Q_\alpha(r) = r^\alpha, \alpha \geq 0 \quad (15).$$

Finalmente la distancia global utilizando OWA para agregar, la presente investigación la define:

$$dg_1 = \sum_{j=1}^{n} w_j d_{o_j} \quad (16)$$

d: vector de información de distancia, donde cada componente se calcula por las fórmulas 2, 3, 5, 11 o 12.

d_{o_i}: es el elemento i-ésimo del vector d ordenado descendentemente.

w_j: es calculado mediante la fórmula (14).

n: total de distancias a agregar.

Donde el vector de pesos es hallado mediante la fórmula (14), escogiendo la familia RIM Q_α representada por la fórmula (15), con $\alpha = 2$.

1.3.7 Variante 2 de agregación de distancias con la media aritmética.

Si $x_1, x_2, \ldots, x_n$ son números positivos, la media aritmética de estos se calcula como

$$a = \frac{x_1 + x_2 + \cdots + x_n}{n} \quad (17)$$

La distancia global utilizando la media se define:

$$dg_2 = \frac{1}{n}\sum_{j=1}^{n} d_j \quad (18)$$

d_j: distancia j-ésima que puede ser calculada por las fórmulas 2, 3, 5, 11 o 12.

n: total de distancias a agregar.

Para cada agregación que se realice, se construye el grafo dirigido $G_d(V, E)$. Tiene como particularidad, que para las aristas podadas entre cualesquiera dos vértices en el grafo $G(V, E)$, los arcos que conectan a estos vértices del grafo dirigido siguen estando podados. Luego los dos arcos que conectan a dos nodos mantienen el mismo peso. Estos pesos son calculados mediante la función de agregación que se escoja de las anteriormente definidas.

1.3.8 Particionamiento del grafo

A partir del grafo $G_d(V, E)$ se aplica la misma fase de agrupamiento del algoritmo BUS descrito en la sección 1.2.7. Los procedimientos implementados se encuentran descritos en el capítulo II.

Conclusiones

El presente capítulo aborda las técnicas de comparación de genes y describe los algoritmos de detección de ortólogos con un enfoque de grafo. Igualmente, se detalla las cinco medidas de distancia local a utilizar en el algoritmo que propone esta investigación, con el propósito de ponderar las conexiones entre nodos de grafos bipartitos. Se determina las técnicas de combinación de rasgos como el operador OWA y la media aritmética.

CAPÍTULO 2
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el este capítulo, se presenta un conjunto de herramientas informáticas y lenguajes de programación, que son utilizados para resolver el problema planteado en la investigación. Además se describe de forma gráfica, utilizando diagramas de procesos, y en lenguaje natural, la solución propuesta para la detección de genes ortólogos.

2.1 Herramientas utilizadas y lenguajes empleados.

2.1.1 MATLAB 7.10.0 (R2010a) y lenguaje MATLAB ®.

Para la implementación de los algoritmos, se empleó la herramienta MATLAB 7.10.0 (R2010a) y como lenguaje de programación MATLAB ®, pues constituye un lenguaje de alto desempeño diseñado para realizar cálculos técnicos. Elizondo (2002) opina que esta herramienta, integra la visualización y la programación en un ambiente fácil de utilizar, donde los problemas y las soluciones se expresan en una notación matemática. Se caracteriza por ser un sistema interactivo, cuyo elemento básico de datos es el arreglo, que no requiere de dimensionamiento previo. Esto permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran.

MATLAB ® también, presenta una familia de soluciones a aplicaciones específicas de acoplamiento rápido llamadas "ToolBoxes", uno de ellos es el de Bioinformática, que ofrece a los científicos un entorno abierto y extensible, a través de un gran número de funciones especializadas, para explorar ideas, crear prototipos de nuevos algoritmos y aplicaciones para la investigación farmacéutica, la ingeniería genética, y otros proyectos acerca de la genómica y la proteómica. Además, proporciona acceso a los formatos de datos genómicos y proteómicos, técnicas de análisis y visualizaciones especializadas para la secuencia genómica y proteómica y análisis de microarrays. La mayoría de sus funciones se implementan en el lenguaje abierto MATLAB ®, lo que permite personalizar los algoritmos o desarrollar otros(Matlab, 2010).

Para la presente investigación fue de gran utilidad esta herramienta, pues con la ayuda del "ToolBoxes" de Bioinformática, se pudo manipular con mayor flexibilidad el formato gbk (extensión de ficheros provenientes del Genbank) mediante la lectura del mismo por la función *genbankread*. Además, se realizó los alineamientos pertinentes entre dos proteínas, sin necesidad de apoyo de otra herramienta. La conversión de cadenas de proteínas a

nucleótidos y viceversa mediante las funciones aa2nt y nt2aa respectivamente. Además, permitió crear un conjunto de funcionalidades nuevas que automatiza la solución propuesta para el problema de la investigación dado.

2.1.2 Mauve en su versión 2.3.1

A partir de la revisión del manual de usuario de Mauve en Internet (Mauve, 2010), se pudo conocer que este software presenta una simple interfaz de usuario para el alineamiento de genomas mediante dos algoritmos: el algoritmo original y el progresivo. El primer algoritmo utiliza una técnica de alineamiento anclada que permite agilizar el proceso de alinear secuencias. Luego reconfigura en cada genoma el orden de las anclas alineadas e identifica los genomas reordenados, mediante el cual obtiene los "Locally Collinear Blocks" (LCB). Mauve completa el alineamiento global con "gaps" para cada LCB, aplicando el algoritmo progresivo. Este funciona mejor para organismos estrechamente cercano y el peso mínimo de los LCB debe ser estimado manualmente para una mejor evaluación de los reordenamientos. No puede alinear regiones conservadas a través de los subconjuntos de los genomas comparados.

El algoritmo progresivo, emplea una metodología diferente para evaluar alineamiento de segmentos conservados, que le permite ser aplicado a un número mayor de genomas que el algoritmo original. El progresivo puede alinear genomas más divergentes que el algoritmo original. El ajuste manual de los parámetros de evaluación para el alineamiento no es usualmente necesario. Es más lento que el algoritmo original y consume mucho más memoria. En el estudio presente, se están comparando dos genomas cercanos en la cadena evolutiva, por tanto, es más práctico usar el algoritmo original para la detección de LCBs en estos genomas.

Este, no funciona bien para grandes números de categorías, porque no puede alinear regiones conservadas a través de los subconjuntos de los genomas comparados. En cambio, el Mauve progresivo, emplea una metodología algorítmica diferente para evaluar alineamientos de segmentos conservados a través de los subconjuntos de categorías. Lo que permite ser aplicado a un número mayor de genomas que al algoritmo original. Puede alinear genomas más divergentes que el algoritmo original. El ajuste manual de los parámetros de evaluación para el alineamiento no es usualmente necesario. Aunque es más lento que el Mauve original y consume mucho más memoria. En el estudio presente, se está comparando dos genomas que son estrechamente cercanos en la cadena evolutiva, por tanto, es más práctico usar el Algoritmo Original de Mauve para la detección de LCB en estos genomas.

A continuación se muestra la (figura 2), que detalla los parámetros utilizados para la búsqueda de los LCB y luego una descripción de sus parámetros, tomados de (Mauve, 2010):

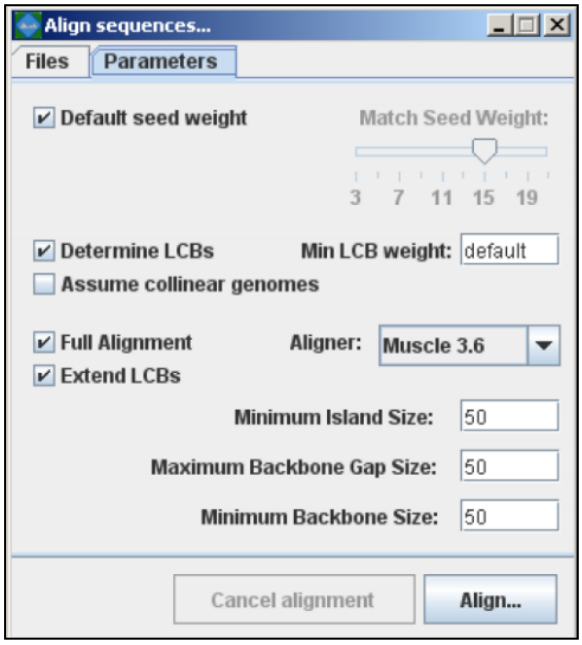


Figura 2: Parámetros de alineamiento y valores por defecto del algoritmo original de Mauve, foto del software *(Mauve, 2010)*.

Match seed weight: fija el peso mínimo del patrón de semilla para calcular los alinemientos múltiples locales durante la primera pasada de del alineamiento de anclaje. Cuando se están alineando dos genomas divergentes o múltiples genomas simultáneamente, un valor pequeño de esta semilla puede proveer mayor sensibilidad. Sin embargo, como el Mauve requiere que las semillas que hagan correspondencias sean únicas en cada genoma, un valor muy bajo puede reducir la sensibilidad.

Default seed weight: garantiza que el Mauve seleccione una semilla inicial para el peso de las correspondencias (Match seed weight) apropiado para la longitud de las secuencias a ser alineadas. Esta semilla por defecto para genomas de 1MB es típicamente cercana a 11, cercana a 15 para genomas de 5MB, y continúa creciendo con el tamaño de los genomas que están siendo alineados. Estos valores por defecto pueden ser conservadores (demasiado grandes), especialmente cuando se están alineando genomas divergentes. Por otro lado, valores altos de esta semilla reducen el ruido en la correspondencia y pueden conducir a mejores alineamientos en algunos casos.

Min LCB weight: fija el número mínimo de correspondencias entre nucleótidos identificados en una región colineal, para que esa región se considere como de verdadera homología y no una semejanza aleatoria.

Determine LCBs: si esta opción es deshabilitada, simplemente se identificarán las correspondencias (alineamientos múltiples locales) a lo largo de los genomas.

ExtendLCBs: controla si el algoritmo extenderá el rango de los LCB existentes y buscará LCB adicionales.

Assume collinear genomes: determina si no hay reordenamientos a lo largo de los genomas comparados.

Aligner: selecciona el algoritmo que Mauve usará para calcular el alineamiento global.

Island and Backbone sizes: una isla es una región del alineamiento donde uno o varios genomas comparados tienen un elemento único de secuencia. Un "backbone" es una región conservada entre todos los genomas comparados. Este parámetro fija el tamaño de los "gaps" usados para calcular los segmentos islas y "backbones".

Full alignment: si esta opción no se selecciona, el Mauve identificará los LCB pero no hará una búsqueda recursiva de las anclas de alineamiento o un alineamiento progresivo.

Los parámetros que se utilizan son los que implementa por defecto, ya que se prestan para alinear genomas estrechamente cercanos, con una cantidad moderadamente alta de reordenamientos.

Después de realizar el alineamiento de los genomas con el Mauve, se obtiene un fichero con información sobre cada LCB detectado, a partir de este fichero, se calcula la posición inicial y final de cada LCB relativa a los genomas. Teniendo los límites de cada LCB en los genomas, se puede verificar si un gen determinado pertenece o no aun LCB. El fichero antes mencionado, contiene la representación del alineamiento de los genomas. En lugar de incluir cada nucleótido alineado, este formato almacena las coordenadas de las regiones que se conservan para salvar espacio en disco. El mismo, comienza con una línea declarando la versión del formato, seguido de varias líneas describiendo las secuencias que fueron alineadas (dos líneas por genoma). La última línea contiene la cantidad de intervalos detectados (IntervalCount), en este caso se detectaron 1883 intervalos. La (Figura 3) muestra un ejemplo del inicio de este fichero.

```
FormatVersion     4
SequenceCount     2
Sequence0File     D:\Inputs\genomas\SCerevisiae\SCerevisiae_Completo.gbk
Sequence0Length   12071326
Sequence1File     D:\Inputs\genomas\SPombe\Pombe genoma completo.gbk
Sequence1Length   12571820
IntervalCount     1883
```

Figura 3: Parte inicial del fichero de alineamiento de Mauve para S. Cerevisiae y S. Pombe.

El resto del fichero, contiene la definición de cada uno de los intervalos. En conjunto, estos intervalos realizan un alineamiento completo de los genomas con reordenamiento. En la (figura 4) se muestra un ejemplo de la definición de un intervalo:

```
Interval 2
23    75446 1475359
GappedAlignment
85    75469 1475382
cacagtggcatcacacctgaaggcaatctcagcgctatttattctgtactcgctcataacggctctgat
aatggtggcatcacaacgaaaagcaatgtcagcagagttaacacgatattcgctgcaaacagcacgaat
23    75554 1475467
GappedAlignment
190   75577 1475490
gatggaaataccacctttctctttttttttgatcaagatatctggcggagttttattcagacgaatccc
gatattgataccaccacgttctttctttttgaatacaatgttgggggggttccttatttaatcttatacc
53    75767 1475680
```

Figura 4: Definición de un intervalo en el fichero de alineamiento de Mauve.

Cada intervalo comienza con la palabra "Interval", seguida de un número que especifica el número del intervalo. Cuando se construye el alineamiento, Mauve escoge un conjunto de multi-MUM (regiones que alinean exactamente y que están presentes en cada genoma alineado) por medio del cual ancla su alineamiento. Cada definición de un intervalo almacena la posición de los multi-MUM que lo componen, en conjunto con los alineamientos de las regiones entre las anclas que son calculados usando ClustalW. En el ejemplo, la línea 23 75446 1475359 representa un multi-MUM de longitud 23, con posición inicial 75446 en el primer genoma y en el segundo con posición inicial en 1475359.

Las posiciones válidas para los multi-MUM en cada genoma, son un número mayor que uno y menor que la longitud de la secuencia del genoma, los multi-MUM con posición cero, se consideran una isla. Los intervalos que tengan al menos un multi-MUM que no sea una isla se consideran un LCB. El ejemplo de la (figura 4) corresponde a un LCB, sin embargo, los intervalos de la siguiente figura no constituyen LCB.

```
Interval 788
24474 1531788     0

Interval 789
64600 0     6217386

Interval 790
2860  1557245     0

Interval 791
1856  0     5961685
```

Figura 5: Intervalos que no se consideran un LCB.

A partir de la explicación de la salida que brinda el Mauve, se aprovecha para obtener un fichero integrador, donde almacena un listado de los LCB detectados, partiendo de que para que exista un LCB debe haber al menos un multi-

MUM que no sea una isla. Para ello, se implementa un conjunto de funcionalidades que como fichero de salida tiene las siguientes características:

```
#seq0_leftend seq0_rightend seq1_leftend seq1_rightend
31576 32274 -11539037 -11539735
71921 72748 -3845560 -3846387
75446 75820 1475359 1475733
77633 78874 1130991 1132232
```

Figura 6: Muestra del fichero con la información de los límites de cada LCB

La primera línea explica que las dos primeras columnas, significan los índices de inicio y fin de cada LCB detectado en el genoma de referencia, para este caso es la de S. cerevisiae, y las dos últimas significan los índices de inicio y fin de los LCB detectado en el segundo genoma, para este caso es el S. pombe.

2.1.3 Lenguaje natural y diagramas de procesos para la descripción de algoritmos

El lenguaje natural, es una forma sencilla de describir un algoritmo, la ventaja fundamental es la facilidad de comprensión por cualquier persona que domine el idioma utilizado. Y el diagrama de procesos o de flujo, constituyen técnicas apropiadas para la descripción de algoritmos de forma gráfica, dado que utiliza un conjunto de estereotipos definidos universalmente. También utiliza lenguaje natural para describir que sucede en cada paso del algoritmo. A continuación se muestra los estereotipos que brinda el Visual Paradigm 8.0 para la realización de diagramas de procesos o flujos, así como su significado:

	Indica que un evento va a comenzar.
	Indica que un evento va a culminar.
------->	Indica que un dato es creado, consultado o modificado.
	Indica el paso de un evento a otro.
	Indica una tarea simple a realizar por el algoritmo.

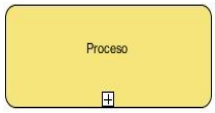
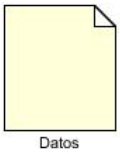
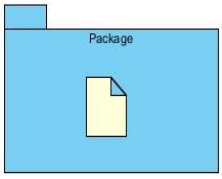
	Indica que para realizarse completamente el proceso, internamente debe realizar más de una tarea simple.
	Indica que existe una condición y dos posibles caminos a seguir, teniendo en cuenta el cumplimiento o no de esta condición.
	Indica un conjunto de datos que pueden ser creados, consultados o modificados por el algoritmo.
	Indica el agrupamiento de un conjunto de datos que pueden ser creados, consultados o modificados por el algoritmo.

Tabla 1: Estereotipos de la descripción gráfica.

2.1.4 Visual Paradigm 8.0.

La herramienta CASE Visual Paradigm for UML Enterprise Edition, utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado. Está diseñada para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientada a la creación de diseños para las diferentes etapas de desarrollo de un software.Se utilizó, para modelar los diagramas de procesos correspondientes a los algoritmos realizados.

2.2 Datos de los genomas

Kamvysselis (Kamvysselis, 2003) plantea, que las levaduras presentan organismos ideales para procedimientos computacionales en vías de desarrollo, encaminados a realizar análisis comparativos de genomas. Son organismos muy estudiados, y el amplio conocimiento funcional existente permite validar los resultados encontrados contra trabajos previos. Su tamaño de genoma pequeño (250 veces más pequeño que el humano) posibilita el análisis de las secuencias con un costo razonable en cuanto a tiempo y espacio. Por todas estas consideraciones, se decidió utilizar un genoma de levadura y un genoma de Schizosaccharomyces en el experimento.

Se realizó un estudio de los datos anotados, más relevantes, de estos dos genomas. El primero es la levadura de cerveza (Saccharomyces cerevisiae), que es un hongo unicelular utilizado industrialmente en la fabricación de pan, cerveza y vino. Constituye un modelo adecuado para el estudio de problemas biológicos. Se utilizaron los 16 cromosomas del mismo y un total de 5861 genes. Su genoma fue descargado en ficheros con formato gbk (ficheros del genbank) de la dirección ftp://ftp.ncbi.nih.gov/genomes/Fungi/Saccharomyces_cerevisiae_uid128/. El segundo genoma es la levadura Schizosaccharomyces pombe o S. pombe, que como el anterior, es utilizada como organismo modelo en biología molecular y biología celular. Se utilizaron los tres cromosomas del mismo y un total de 5006 genes. Su genoma fue descargado en ficheros con formato gbk (ficheros del genbank) de la dirección ftp://ftp.ncbi.nih.gov/genomes/Fungi/Schizosaccharomyces_pombe_uid127/. Para la selección de los genes se tuvo en cuenta las coincidencias de los genes anotados por el Inparanoid, debido a que este constituye el algoritmo modelo para comparar los resultados de la presente investigación.

Del primer genoma, no se incluyeron dos genes debido a que no aparecían en los genes anotados por el Inparanoid en los ficheros fasta. Estos son, según el identificador locus_tag tanto de los icheros FASTA como de los ficheros gbk, "YCL058C" y "YEL033W". Del segundo genoma no se incluyeron cuatro por la misma y son "SPAC1952.04c", "SPBP35G2.16c", "SPBC8E4.12c" y "SPCC1235.16".

2.3 Datos de entrada para el algoritmo

Como ambos genomas lo componen tantos ficheros como cromosomas tienen, se realiza una lectura organizada de los mismos, comenzando por el primer cromosoma, luego el segundo y así hasta llegar al último fichero que coincide con el último cromosoma, facilitando el análisis de genomas con múltiples cromosomas y de genomas incompletos formados por varias secuencias de "contings" organizados.

2.4 Descripción general de la solución propuesta

El algoritmo general que se propone **(ver diagrama de procesos 1)** se describe en 6 pasos como sigue:

Paso 1: Se parte de los arreglos SC (genes del genoma SCerevisiae) y SP (genes del genoma SPombe), que constituyen las estructuras de datos creadas con la información de los genomas, para crear a partir de un alineamiento local todos contra todos, la **Matriz de Bitscore** (creada a partir de la puntuación obtenida del alineamiento de un gen contra otro), la **Matriz de Score** (que se obtiene a partir de la anterior, pero cada Bitscore, de la **Matriz de Bitscore**, es normalizado dividiendo por la máxima puntuación obtenida de los alineamientos en esta matriz) y la **Matriz de Alineamiento** (se obtiene a partir del alineamiento de un gen contra otro, cada elemento almacenado en esta matriz constituye un arreglo de longitud dos, donde el primer y segundo elemento coinciden con las cadenas obtenidas a partir del alineamiento de un gen de S. cerevisiae y uno de S. pombe respectivamente).

Paso 2: Calcular las 5 funciones de distancias locales entre los dos genomas. Para el cálculo de las distancias, es necesario respetar la poda realizada a la matriz de similitud. Esta información se almacena en 5 ficheros que representan 5 grafos bipartidos podados en su forma matricial.

Paso 3: Aplicar el operador OWA y la media aritmética, como técnicas de combinación de rasgos, alternando los dos últimos, a partir de la entrada de los grafos obtenidos en el paso 2. Cada agregación realizada crea una matriz que se guarda en un fichero, representando el grafo bipartido podado de la relación entre genes.

Paso 4: Eliminar las ambigüedades en cada uno de los grafos podados obtenidos a partir de la agregación de rasgos. Luego, se obtiene y almacena la información en ficheros, de los grafos bipartidos podados y sin ambigüedades.

Paso 5: Aplicar el algoritmo de agrupamiento BUS, a los grafos bipartidos podados y sin ambigüedades obtenidos en el paso 4. Luego se obtienen y almacenan en ficheros distintos, las listas de genes ortólogos, una por cada grafo dirigido podado y sin ambigüedades.

A continuación se muestra la (figura 7), que es una descripción general del algoritmo, en forma de diagrama de procesos:

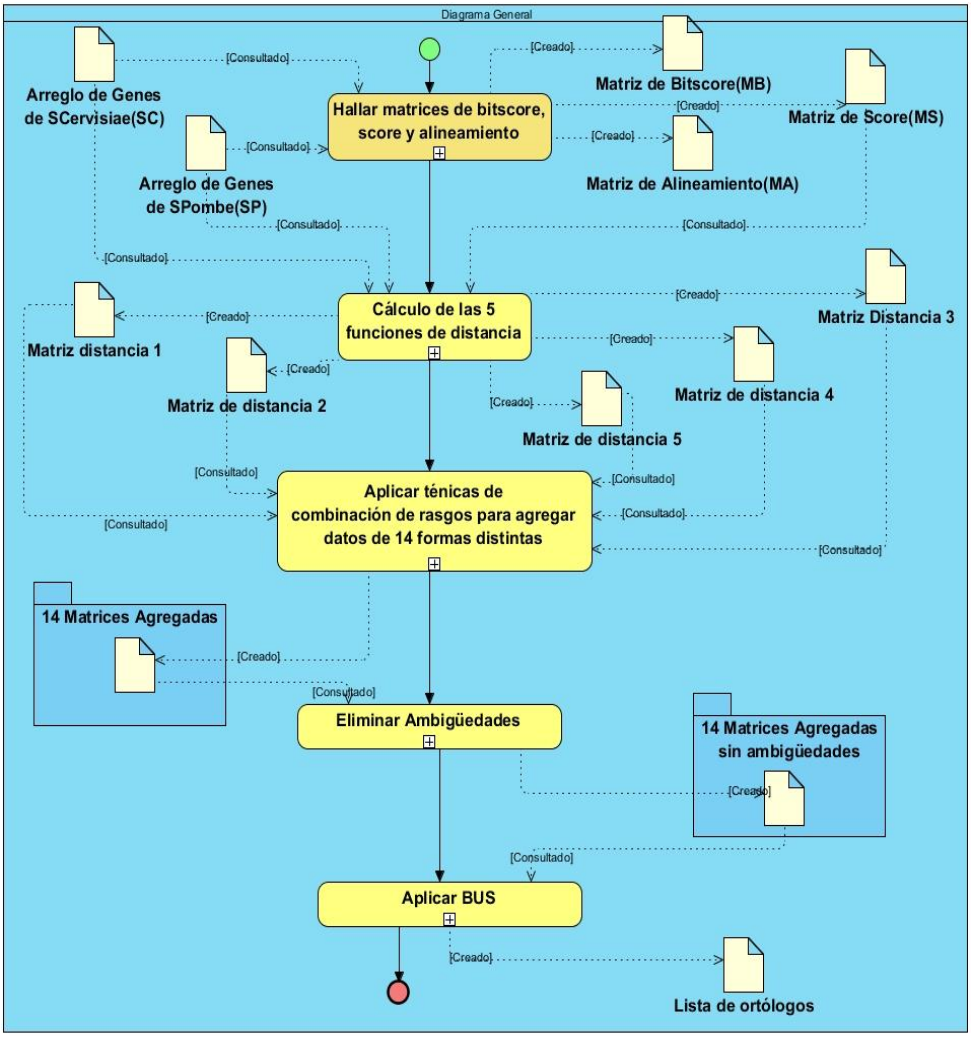


Figura 7: Diagrama de procesos del Algoritmo general.

2.4.1 Descripción del paso 1: Hallar matrices de score, alineamiento y bitscore.

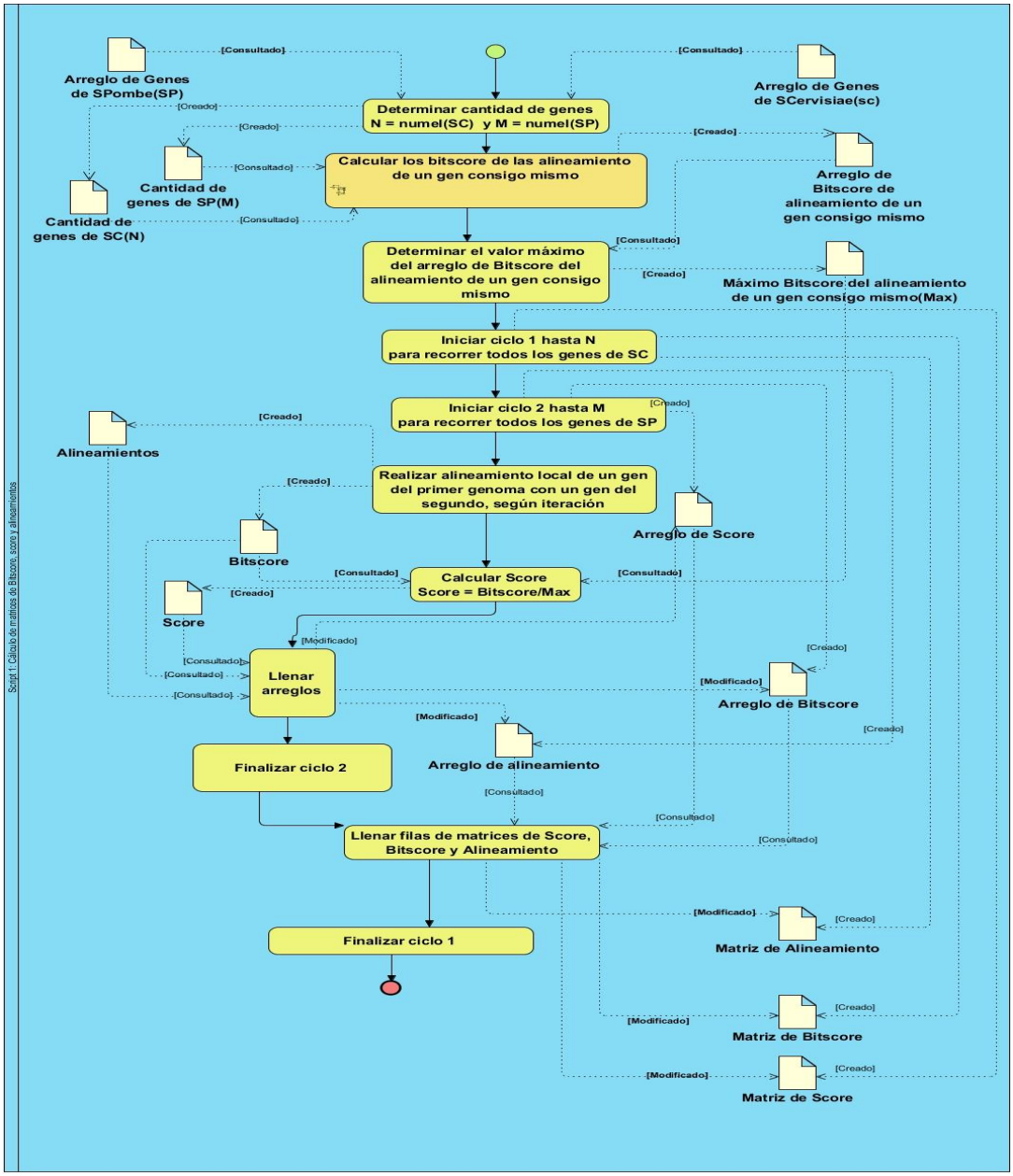


Figura 8: Diagrama de procesos para el algoritmo que crea la Matriz de Score, de Bitscore y de Alineamiento.

Uno de los enfoques más comunes para la clasificación de genes, está basado en la búsqueda de homologías de secuencias. La solución propuesta representada en la (figura 8), toma en cuenta 2 secuencias como entrada y provee una medida de similitud entre ellas. Mientras más similitud halla, más pequeña es la medida. El uso de la función *swaling* implementada en MATLAB, garantiza el alineamiento local de dos secuencias de proteínas mediante el

algoritmo Smith-Waterman. Como parámetros se seleccionaron los valores por defecto, la matriz de puntuación 'BLOSUM50', que incluye una estructura con un factor de escala capaz de convertir las unidades de la puntuación de salida a los bits, con una penalización por gaps de 8. A partir del alineamiento de todos los genes del primer genoma, contra todos los genes del segundo genoma se calculan tres matrices, que su tamaño es de 5861 filas por 5006 columnas, representando las filas a los genes de Saccharomyces cerevisiae y las columnas a los de Schizosaccharomyces pombe. A continuación se describen las matrices:

Nota: $i = 1,2,\dots,5861; j = 1,2,\dots,5006$.

La **Matriz de Bitscore** guarda en la posición i, j, la puntuación obtenida a partir del alineamiento del gen i del primer genoma con el gen j del segundo genoma.

La **Matriz de Alineamiento** guarda en la posición i, j un arreglo de tamaño dos. El primer y segundo elemento son las cadenas resultantes del alineamiento del gen i del primer genoma contra el gen j del segundo genoma.

La **Matriz de Score** almacena en la posición i, j, el cálculo obtenido a partir de dividir la puntuación almacenada en la posición i, j de la **Matriz de Bitscore** por el máximo valor de esta matriz.

2.4.2 Descripción del paso 2: Cálculo de las 5 funciones de distancia.

El objetivo de este paso es calcular 5 matrices de distancia local, teniendo en cuenta 4 rasgos, entre pares de genes de distintos genomas (ver figuras 9, 10, 11, 12, 13). El Capítulo 1 (sección 1.3) se definen las 5 funciones de distancias locales. La primera distancia define la homología, la segunda tiene en cuenta la longitud de la secuencia, la tercera halla la pertenecía a los LCB, la cuarta y la quinta caracterizan a un mismo rasgo, la distancia evolutiva entre pares de genes de distintos genomas, teniendo en cuenta el modelo SG2009 de (Sánchez and Grau, 2009), estas dos últimas difieren en que la primera es la distancia evolutiva teniendo en cuenta las proteínas de los genes y la segunda, la distancia evolutiva teniendo en cuenta secuencias de nucleótidos de los genes.

2.4.2.1 Descripción de la implementación del rasgo que caracteriza la homología de la secuencia.

La función de distancia implementada, según la homología entre secuencias, persigue como objetivo crear un matriz con la información sobre la homología entre cada par de genes. Esto se logra, consultando la Matriz de Bitscore (MB) y la Matriz Score (MS) obtenida en el paso 1, descrito en el epígrafe 2.4. La matriz MB almacena los valores que caracteriza la similitud de todos los genes contra todos y MS muestra estos valores normalizados (entre 0 y 1).
En la investigación realizada por Kamvysselis (Kamvysselis, 2003), la matriz de similitud es podada para eliminar aquellos relaciones entre genes débiles (baja similitud) según el análisis realizado sobre sus secuencias de proteínas, esta misma idea fue aplicada en la presente investigación. La poda se realiza teniendo en cuenta, que es una matriz de

similitud cuyas filas y columnas representan dos genomas de distintas especies. Para ello es necesario recorrer cada fila de la matriz, buscando por cada una, la columna que posee el máximo valor (relación más similar), luego, todos los valores que estén por debajo del rango entre el 80% del máximo y el máximo se eliminan de la matriz MS, colocando un valor infinito, que en este caso es 2. En consecuencia, la matriz MS contiene a parte de la información de la relación de similitud normalizada entre dos genomas, también la información de aquellos genes no similares representados por el número 2, lo que significa que los genes no tienen relación. A la hora de representar el grafo bipartido, el 2 significa que no hay aristas entre los nodos.

Finalmente, si se buscan aquellos genes menos distantes a lo largo de las especies en la matriz MS, entonces los valores de similitud que interesan son los positivos, que indican una buen alineamiento, aquellas puntuaciones negativas se obvian en la fórmula 2 definida en el Capítulo 1, además se obvian los elementos con relación 2 debido a que se mantiene la poda realizada a MB. La distribución de estas puntuaciones es afectada por la longitud de las secuencias, con secuencias muy largas se tienen valores más altos y más bajos que con secuencias más cortas. La normalización de los valores de MB se realiza dividiendo por el mejor alineamiento posible a partir del análisis de todos contra todos. Luego, sustituyendo los valores negativos por cero y dividiendo todos los elementos por el mejor alineamiento de un gen consigo mismo, se obtiene la matriz esperada.

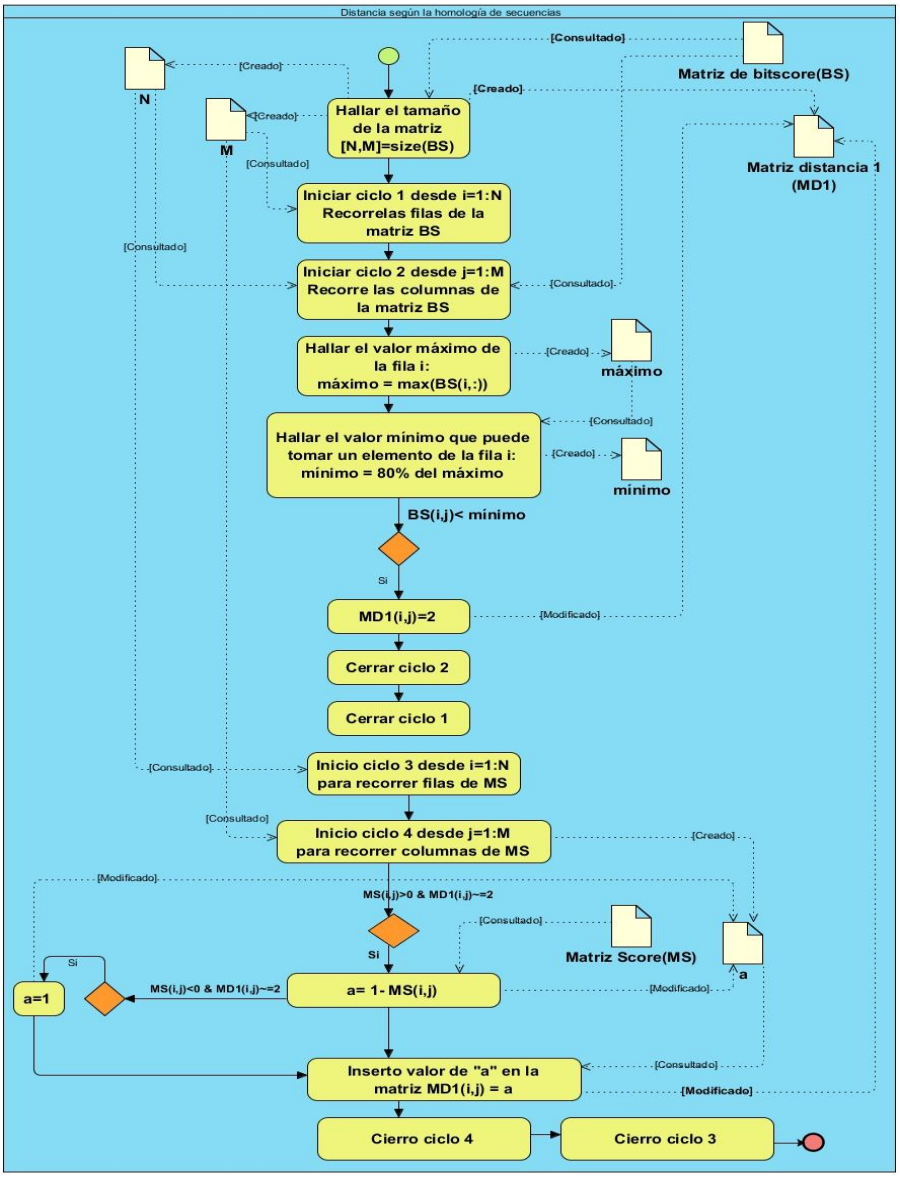


Figura 9: Diagrama de procesos de la implementación de la función de distancia según la homología entre secuencias.

2.4.2.2 Descripción de la implementación del rasgo que caracteriza la longitud de la secuencia.

Se define la longitud de una secuencia como la cantidad de aminoácidos que contiene su secuencia. La función de distancia implementada, persigue como objetivo crear una matriz, que sus elementos representen la relación entre dos genes, tomando como atributo la longitud de sus secuencias. Para ello es necesario consultar los arreglos de genes creados, calcular la cantidad de genes por cada uno. Continúa, seleccionando las cadenas a comparar y calcula

sus longitudes. Es necesario calcular también el gen más largo y el gen más corto. Luego, mediante la fórmula 3 descrita en el **capítulo 1** se calcula la relación entre dos secuencias y se inserta de forma organizada en la nueva matriz.

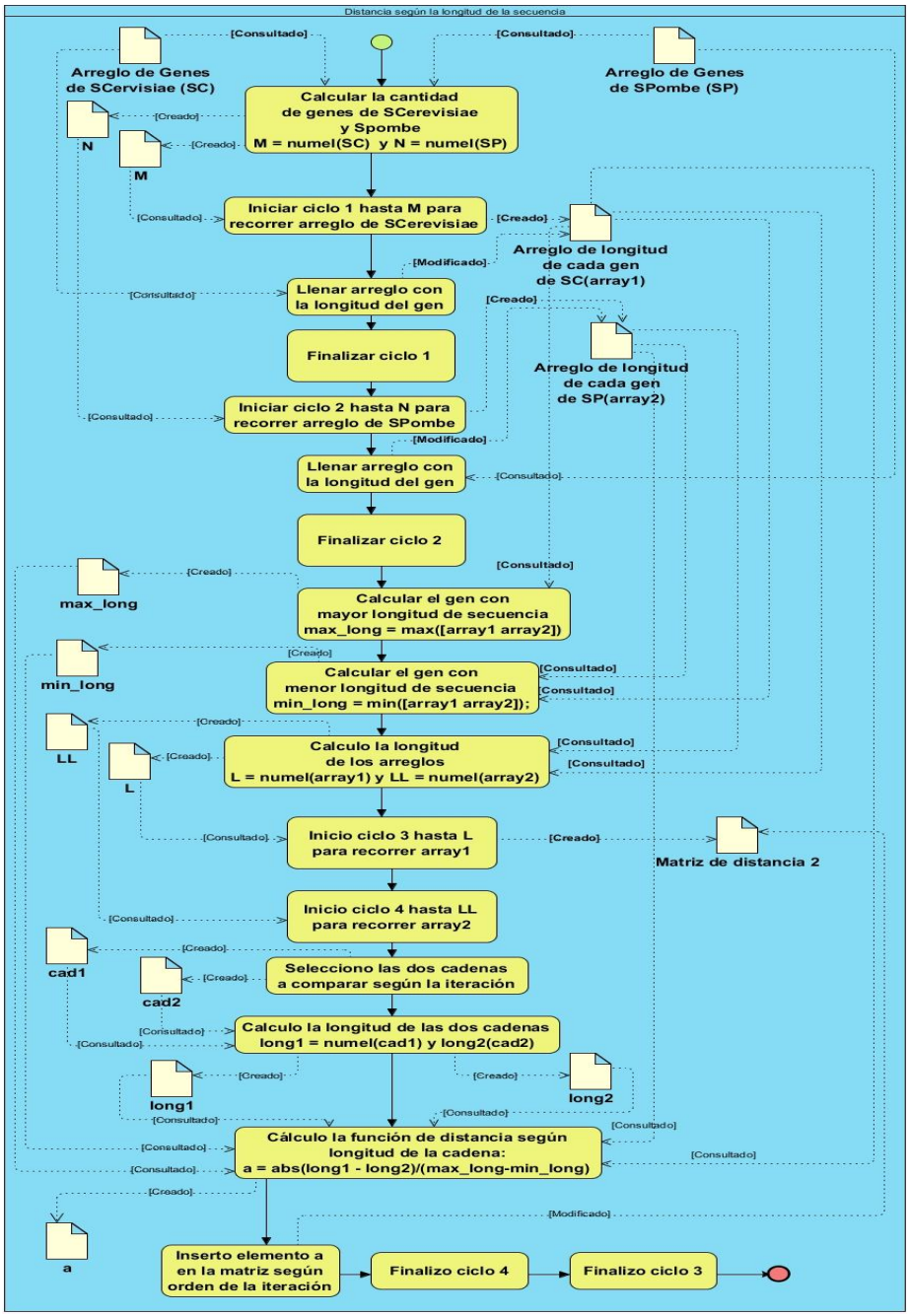


Figura 10: Diagrama de procesos de la implementación de la función de distancia según la longitud entre secuencias.

2.4.2.3 Descripción de la implementación del rasgo que caracteriza la pertenencia a los LCB.

Se define que un gen pertenece a un LCB, si al menos tienen en común una base. El objetivo que persigue la función de distancia según la pertenencia a los bloques de orden conservado (LCB), es obtener una matriz cuyos datos son 0 y 1, por lo que dos genes pertenecen a un mismo LCB si contiene el valor 1, 0 lo contrario. Para esto es necesario consultar las estructuras de datos que contienen ambos genomas, y las lista de LCB con su posición de inicio y fin. Esta lista es calculada, después de realizar el alineamiento de los dos genomas con el Mauve, ya que el mismo ofrece un fichero en su salida con información sobre cada LCB detectado. Estos ficheros están descritos en las figuras 3, 4 y 5.

Teniendo los límites de cada LCB en cada genoma, se verifica si los genes del primer genoma pertenece o no a un LCB, lo mismo para los genes del segundo genoma. Luego se obtiene una matriz de tamaño $(N + M) * num\_lcb$, donde N son los genes de S. cerevisiae, M los de S. pombe y num_lcb la cantidad de LCB detectados. Esta matriz en la posición i, j, contiene 1 si el gen pertenece al LCB y con 0 si ocurre lo contrario. Debe señalarse que en esta matriz las primeras N filas son los genes de S. cerevisiae y las próximas $N + 1$ hasta $N + M$ son los genes de S. pombe. Luego, con el uso de la función pdist implementada en MATLAB, se calcula la distancia de Jaccard como se explica en el Capítulo 1, fórmula 5, entre pares de objetos. En este caso, los objetos son los genes y los LCB. Según (Matlab, 2010), a *pdist* se le pasa como parámetro la matriz anteriormente mencionada y devuelve un arreglo de longitud $\frac{(N+M)*(N+M-1)}{2}$, el mismo indica las relaciones entre filas como sigue (fila 2, fila 1), (fila 3, fila 1), …,(fila M+N, fila 1),(fila 3, fila 2),…,(fila M+N, fila M+N-1). Este vector es cómodamente utilizado como matriz de dismilitud. Seguidamente, es convertido a una matriz cuadrada mediante la función *squareform* implementada en MATLAB, de modo que el elemento i, j en la matriz donde $i < j$, corresponde a la distancia entre el objeto i y j en el conjunto de datos originales. Luego, la matriz deseada es hallada teniendo en cuenta esta regla. A continuación se explican gráficamente los procesos implementados.

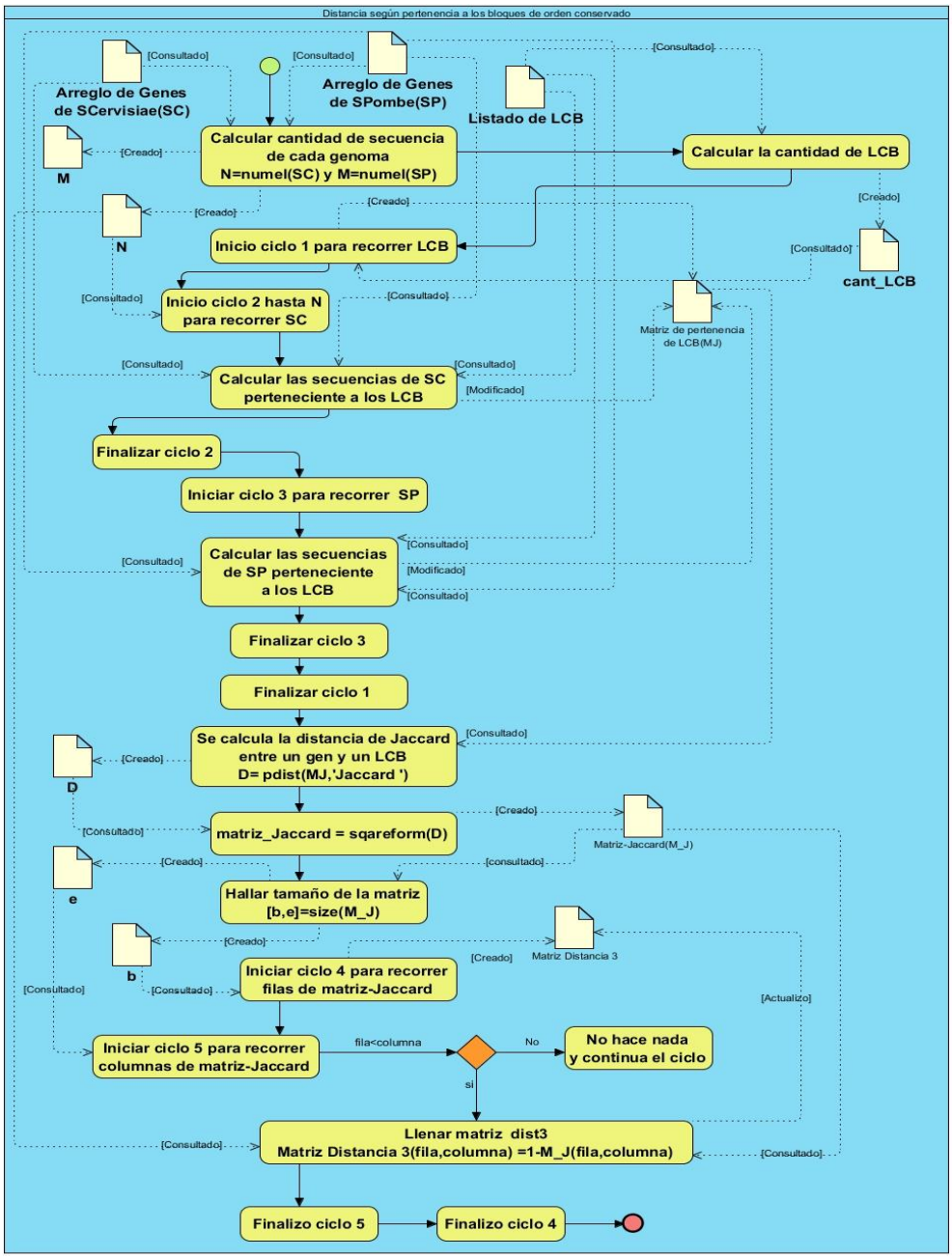


Figura 11: Diagrama de procesos de la implementación de la función de distancia según la pertenencia a los LCB.

2.4.2.4 Descripción del algoritmo de la función de distancia evolutiva para proteínas de genes

El algoritmo trabaja con la **Matriz de Alineamiento** obtenida en el **paso 1**. El objetivo es hallar la distancia entre dos secuencias de aminoácidos alineadas. Para esto, es necesario recorrer todos los elementos de la matriz e ir

seleccionando los alineamientos entre dos genes. Luego, es necesario ir hallando las distancias entre pares de aminoácidos, por lo que en la posición i,j, se verifica numéricamente la extensión de las cadenas alineadas (las cadenas alineadas tiene siempre el mismo tamaño). Dichas cadenas son tratadas en MATLAB como arreglos de 'char', por lo que se puede acceder a cada carácter de la misma forma que se accede a los elementos de un arreglo y se realiza el análisis del elemento i (un aminoácido) de la primera con el elemento j (otro aminoácido) de la segunda. Para estos, se construye dos arreglos A y B cuyos elementos son los posibles codones (cadenas de longitud tres) que lo codifican respectivamente.

Estos arreglos no son más que conjuntos, donde los elementos son codones, aplicando la teoría de relaciones binarias, se obtiene la relación $A \times B = \{(a,b), a \in A \; y \; b \in B\}$. Donde a y b se relacionan a partir de una puntuación que representa la distancia entre cadenas de codones descrita en la (figura 10). Posteriormente, se hallan todas las puntuaciones de los elementos de $A \times B$, se suman y se dividen por $|A \times B|$, que no es más que la cardinalidad del conjunto $A \times B$ (cantidad de elementos que contiene este conjunto). Esta operación determina la distancia entre dos aminoácidos partiendo de los codones que la codifican.

Halladas todas las distancias entre pares de aminoácidos, se suman y se dividen por la cantidad de aminoácidos que contiene las cadenas alineadas, obteniéndose el promedio de distancias entre las cadenas alineadas almacenadas en i,j. Luego, este valor es almacenado en una nueva matriz en la posición i,j. Esta nueva matriz contiene la información hallada a partir de la distancia antes descrita que relaciona los alineamientos de las proteínas entre dos genes de diferentes genomas.

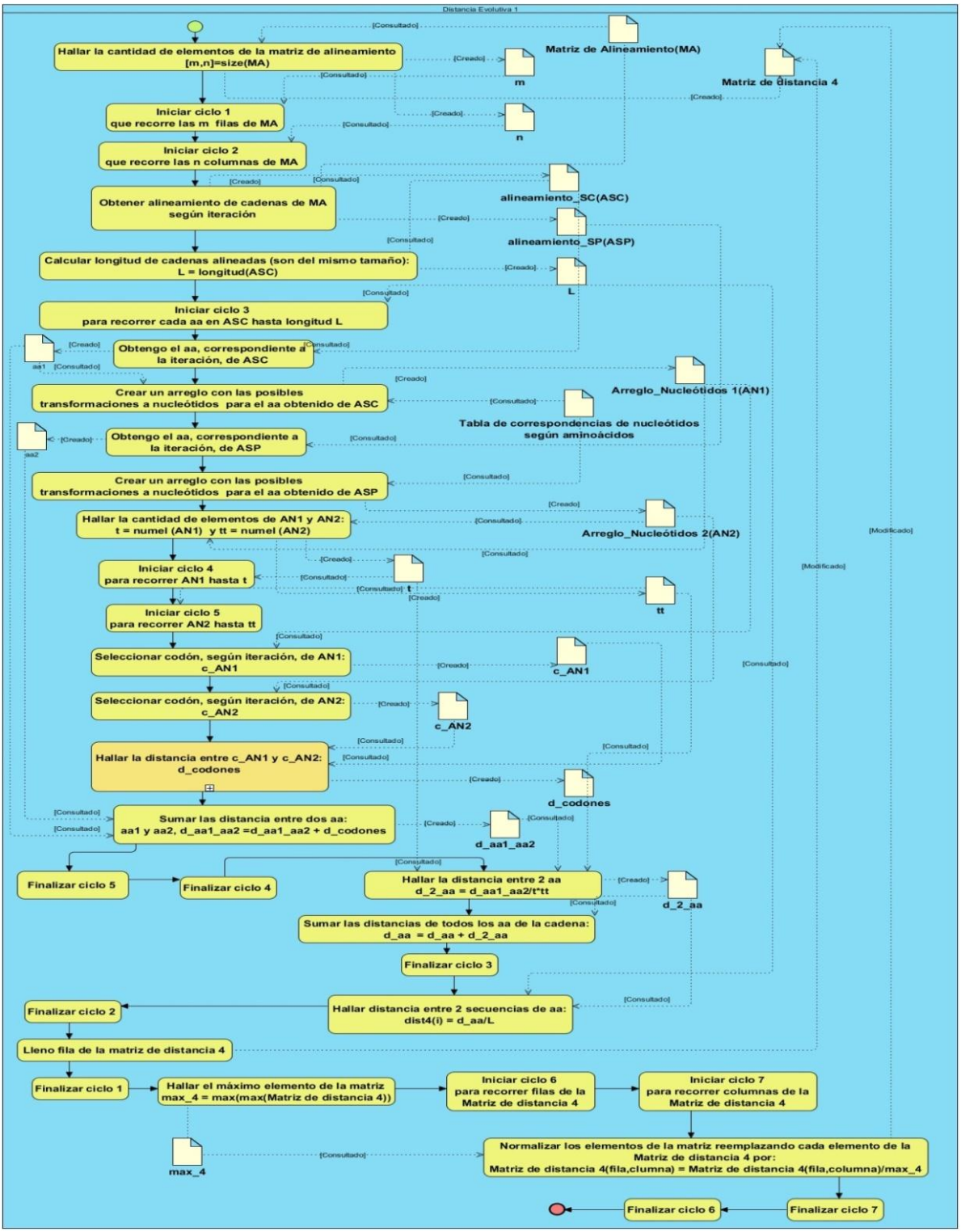


Figura 12: Diagrama de procesos de la implementación de la función de distancia evolutiva para proteínas según modelo SG2009.

2.4.2.5 Descripción del algoritmo de la función de distancia evolutiva para secuencias de nucleótidos de genes

El objetivo de esta función de distancia implementada, es construir una matriz que caracterice la relación evolutiva entre cada par de genes, a través de su secuencia de nucleótidos. Como consecuencia, también necesita de la **Matriz de Alineamiento** obtenida en el **paso 1**, para ir obteniendo las secuencias de aminoácidos en la posición i, j. Luego, las secuencias de aminoácidos alineadas se traducen a secuencias de nucleótidos igualmente alineadas, mediante la función *aa2nt* desarrollada por **MATLAB**, esta función, acepta como parámetros una secuencia de aminoácidos y la especificación de si es una secuencia de ADN o de ARN, en este caso se especifica que es de ADN.

Obtenida las conversiones antes mencionadas, se calcula la longitud de estas secuencias, que al ser divididas por tres, representa la cantidad de codones que contienen (los codones son tres nucleótidos consecutivos). Seguidamente, transforma el alfabeto de nucleótidos, en un alfabeto de números como se declara en el **Capítulo 1**. Procede hallando la distancia entre dos codones, el primero del genoma de referencia y el segundo del otro genoma, teniendo en cuenta la fórmula (7) del anterior capítulo. Los valores arrojados por la aplicación de la función de distancia a todos los codones, se suman y se divide por la cantidad de codones existentes en las secuencias. Estos valores son introducidos en la matriz de distancia 4, luego los valores de la nueva matriz son reemplazados por ellos mismos divididos por el valor máximo de los elementos de la matriz de distancia 4, esto se realiza con el fin de que la matriz resultante tenga sus datos normalizados. A continuación se muestra los diagramas de procesos, de las dos funciones de distancias evolutivas anteriormente descritas, obtenidas a partir del modelo FB, manteniendo el mismo orden que en la descripción:

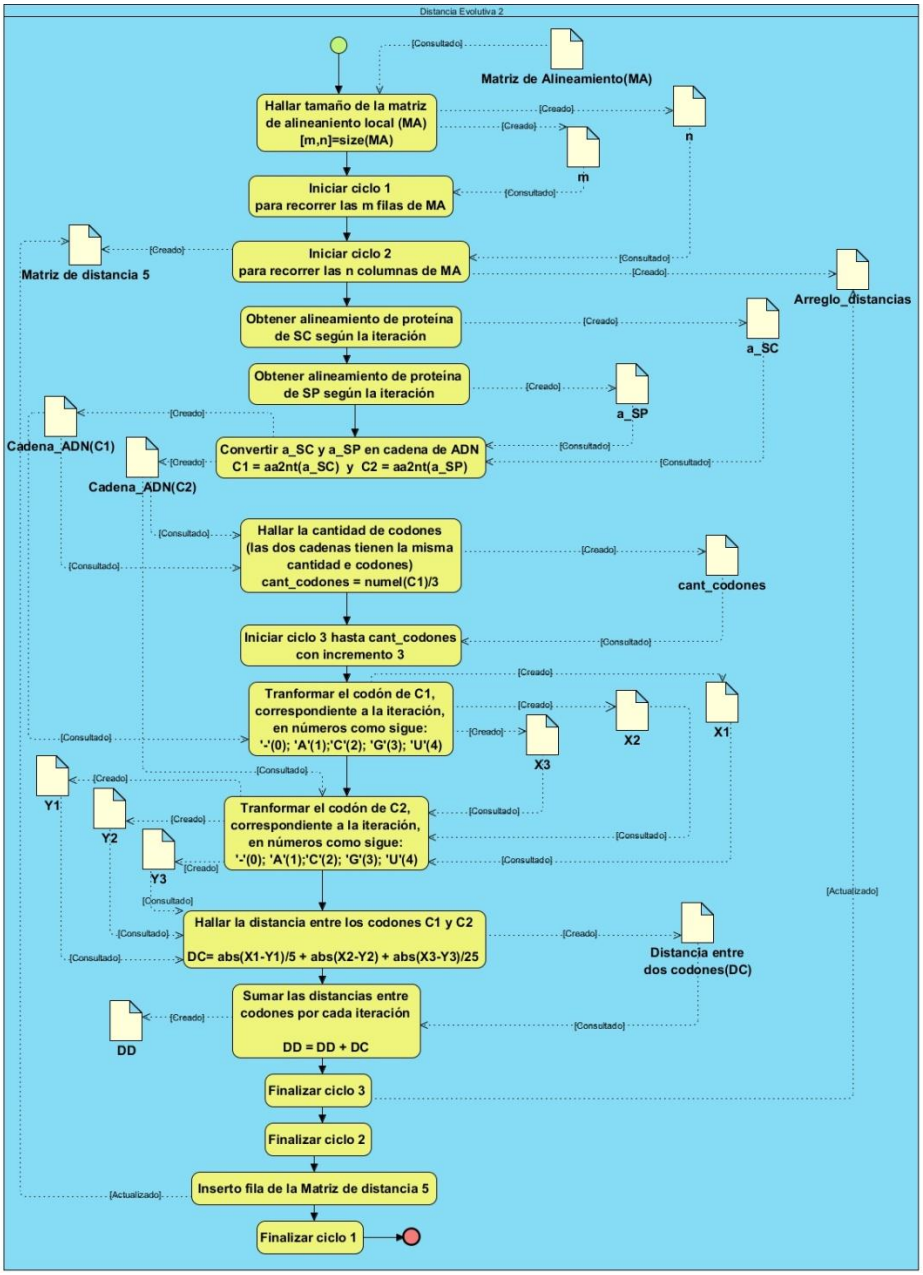


Figura 13: Diagrama de procesos de la implementación de la función de distancia evolutiva para ADN según el modelo SG2009.

2.4.3 Descripción del paso 3: Aplicar técnicas de combinación de rasgos.

En este paso (descrito en las figuras 14 y 15) se trabaja con 4 rasgos, el primero caracteriza la homología entre pares de secuencias, el segundo la longitud entre pares de secuencias, el tercero el grado de pertenencia a los LCB y el cuarto la distancia evolutiva. Para este último se definió e implementó en la presente investigación dos distancias,

sólo la que tiene en cuenta las cadenas de nucleótidos de los genes está presente a partir de este paso hasta el paso 5. Estos 4 rasgos representados en forma de matrices se agregan para obtener 14 matrices de distancia, que representan las matrices de adyacencia de 14 grafos bipartidos, conteniendo la información mezclada de varios rasgos predefinidos. La combinación de rasgo para cada matriz será explicada en los experimentos del **Capítulo 3**. La agregación entre matrices se realiza mediante el operador OWA y la media aritmética, siempre teniendo en cuenta que el rasgo 1 debe ir en la agregación, debido a que tiene implícito la poda inicial realizada a la matriz de similitud que caracteriza la homología entre pares de secuencias. La agregación se organizó de tres formas: primero combinando los cuatro rasgos, segundo combinando tres rasgos y tercero combinando dos rasgos.

Después de hallar las 14 matrices de costo, se tienen caracterizados a 14 grafos bipartidos, que no son completos, debido a que se realizó una poda inicial partiendo de la similitud entre genes. Estos grafos tienen en común, el conjunto de nodos V_1 y V_2, constituyendo los genes del primero y segundo genoma respectivamente.

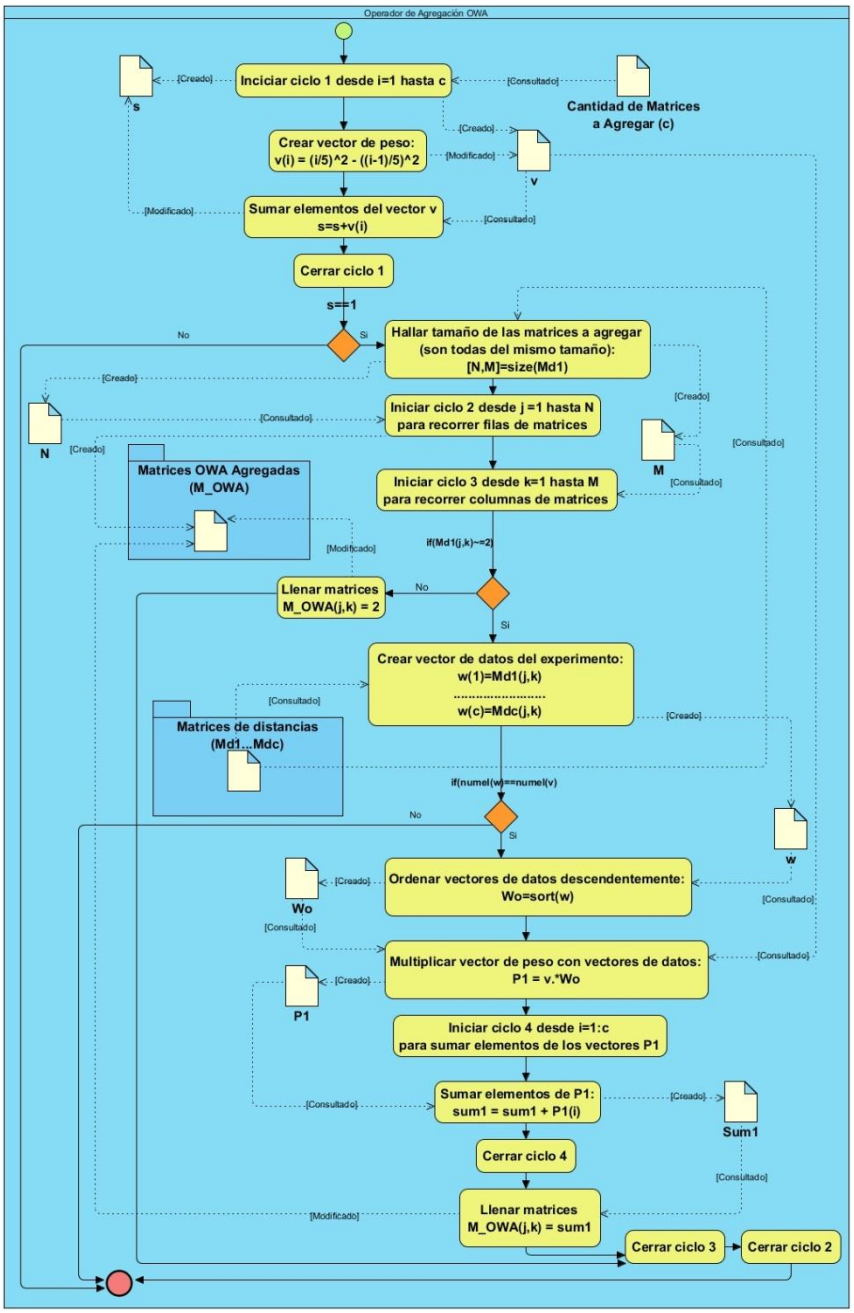


Figura 14: Diagrama de procesos para agregar datos mediante el operador OWA.

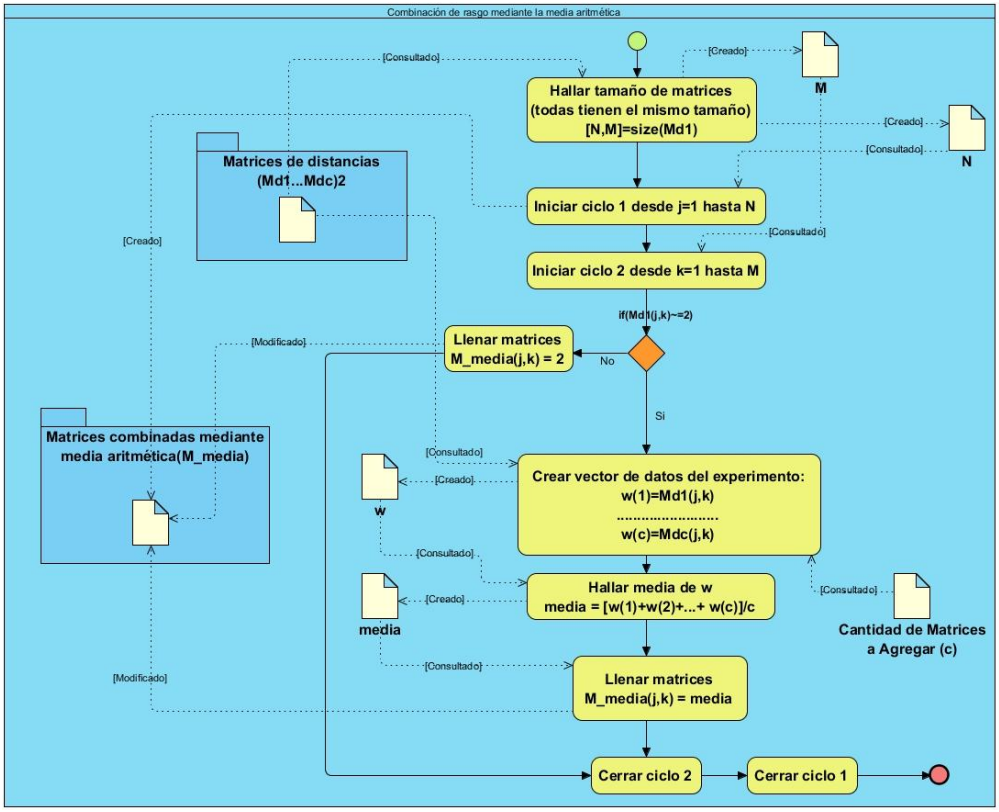


Figura 15: Diagrama de procesos para agregar datos mediante la media aritmética.

2.4.4 Descripción del paso 4: Eliminar las ambigüedades.

De los grafos bipartidos obtenidos, existen numerosos subgrafos de dos elementos entre proteínas que no tienen estrechas relaciones. Estos son utilizados para construir bloques de orden conservado basados en la distancia física entre genes consecutivos conectados.

En general, para realizar la eliminación de ambigüedades en cualquiera de los grafos, primero hay que construir los bloques de orden conservado, para ello se cuenta, con las posiciones de los genes dentro de la secuencia genómica descrita en los campos Indice_inicio e Indice_final, dispuestos en las estructuras de datos implementados para guardar la información de los dos genomas. Antes de este paso, es necesario hallar las correspondencias uno a uno sin ambigüedades, buscando subgrafos de dos vértices dentro del grafo bipartido.

Los nodo S $v \in V_1$ y $w \in V_2$, tienen una correspondencia uno a uno si se cumple, que existe una única arista de salida desde v a w en el grafo dirigido, que relaciona los vértices de V_1 con los de V_2; además, existe una única arista de salida desde w a v en el grafo dirigido, que relaciona los vértices de V_2 con los de V_1. La lista de correspondencias

uno a uno, es un arreglo de estructuras, donde cada elemento contiene los índices de dos genes del primer y segundo genoma respectivamente, los cuales forman la correspondencia sin ambigüedades.

Para la construcción de los bloques de orden conservado, hay que analizar que las parejas de genes sin ambigüedades sean físicamente cercanas. Se toma en cuenta la definición brindada por (Kamvysselis, 2003), que un gen es físicamente cercano a otro, si la distancia d entre sus posiciones, no supera los 20kb, que corresponde aproximadamente a 10 genes, lo que quiere decir, es que el valor absoluto de la diferencia entre las posiciones de ambos genes en la secuencia genómica es menor o igual que 20 000. En consecuencia, si a un nivel genómico ocurren muchos reordenamientos desde la separación de las especies, o si las secuencias comparadas son cortas, los segmentos conservados serán cortos. En este caso, fijar d a valores pequeños puede perjudicar el funcionamiento del algoritmo. Por otro lado, si el número de genes sin ambigüedades es demasiado pequeño al principio de este paso, los genes usados como bases estarán distantes, y no será posible construir bloques de orden conservado para valores pequeños de d.

Además, hay que tener en cuenta a la hora de crear los bloques, que conserven el orden de los genes, para esto, se define el siguiente procedimiento: dos conexiones uno a uno sin ambigüedades (x_1, y_1) y (x_2, y_2) tal que x_1 es físicamente cercano a x_2 y y_1 es físicamente cercano a y_2, se construye un bloque de orden conservado $B = (\{x_1, x_2\}, \{y_1, y_2\})$. Después, para todo gen x_3 que es próximo a $\{x_1, x_2\}$, si existe una arista de salida (x_3, y_3) tal que y_3 es próximo a $\{y_1, y_2\}$, se ignorará cualquier otra arista de salida (x_3, y')si y'no es próximo a $\{y_1, y_2\}$.

Sin este paso, los genes duplicados en las especies comparadas permanecerán en grupos de homología de dos a dos (Kamvysselis, 2003). Este paso juega un importante rol cuando las especies comparadas están muy separadas a través del proceso evolutivo y la similitud entre las secuencias no es suficiente para resolver todas las ambigüedades. Sólo se tomará en cuenta, los bloques de orden conservado que tengan como mínimo tres genes para resolver las ambigüedades, de esta forma se evita la confusión por el reordenamiento de genes aislados.

Finalmente, para eliminar las ambigüedades, es necesario saber si un gen es físicamente cercano a un bloque de orden conservado. Esto se puede saber teniendo en cuenta tres elementos: el primero, saber el genoma al cual pertenece el gen, el segundo, conocer el índice del gen y tercero tener la lista de índices de genes del mismo genoma que pertenecen al bloque de orden conservado. El procedimiento a seguir es: conocido el gen a comparar, buscar dentro del bloque de orden conservado al menos un gen que sea físicamente cercano a este. De esta forma, el gen y el bloque de orden conservado se consideran físicamente cercanos. Posteriormente, las aristas que conectan a genes cercanos a bloques de orden conservado con genes alejados de estos bloques serán ignoradas, como aristas que no representan relaciones entre ortólogos.

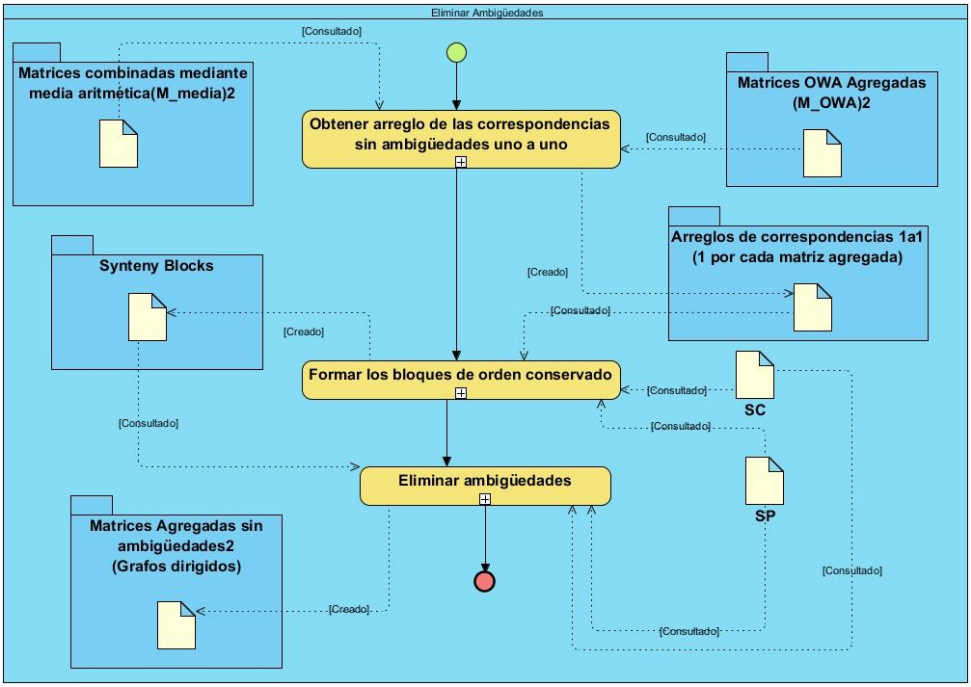


Figura 16: Diagrama de procesos de la implementación de la eliminación de ambigüedades.

2.4.5 Descripción del paso 5: Aplicar BUS.

El algoritmo de agrupamiento BUS, se encarga de separar del grafo, los subgrafos conectados a este por aristas no óptimas. De manera que, la mejor conexión de un nodo del subconjunto esté contenido dentro de él y no en otro; además, no existe un nodo fuera del subconjunto que tenga su mejor conexión dentro del subconjunto. Estas propiedades aseguran la optimización del subconjunto y que no contengan ambigüedades.

Este algoritmo es definido por Kamvysselis (Kamvysselis, 2003) como el subconjunto$S$ de nodos, tal que $\forall x: x \in S \Leftrightarrow best(x) \subseteq S$, $best(x)$ constituye los nodos conectados por las aristas de menor peso desde el nodo x. En consecuencia, se construye un grafo como el subgrafo del grafo podado y sin ambigüedades obtenido del paso anterior, que contiene solo las mejores aristas de salida de cada nodo. Notar que en el trabajo se consideran múltiples mejores aristas de salida. Para construir un BUS, se comienza con un nodo no visitado. Seguidamente a este subconjunto se adiciona los nuevos nodos encontrados dentro del recorrido de las mejores aristas hacia adelante y hacia atrás. Se incluyen nodos adicionales si su mejor conexión está dentro del subconjunto, o si él es la mejor conexión de un nodo dentro del subconjunto. Esto asegura que al separar los subconjuntos no se quede ningún nodo aislado, y no se deseche ninguna mejor conexión. Cuando no es posible adicionar un nuevo nodo, la condición BUS

está satisfecha. Cada BUS encontrado forma un grupo de homología, en especial aquellos BUS compuestos por solo dos elementos forman pares de ortólogos.

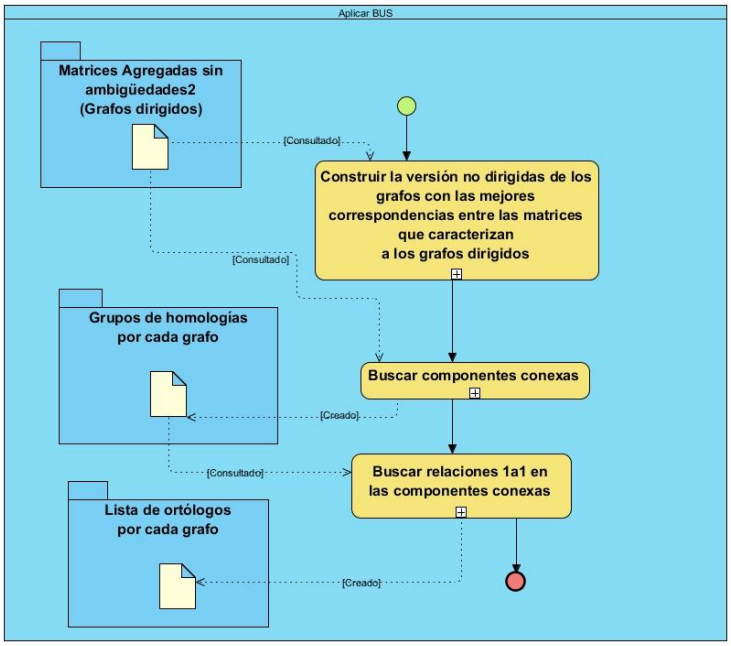


Figura 17: Diagrama de procesos de la implementación del BUS.

Conclusiones

El capítulo describe las herramientas utilizadas en la presente investigación, enfatizando el uso que se les dio en el trabajo. Además explica las características de los datos de los genomas utilizados así como las estructuras creadas para un mejor aprovechamiento de los mismos en las distintas entradas del algoritmo. También, describe la solución informática propuesta, a través de diagramas de procesos y en lenguaje natural.

CAPÍTULO 3
VALIDACIÓN DE LOS RESULTADOS

El presente capítulo realiza la prueba no paramétrica horizontal de "McNemar" para analizar los resultados de 17 experimentos diseñados. Además se hace un análisis gráfico de los índices de coincidencias entre el algoritmo Inparanoid 7 y el diseñado por la presente investigación.

3.1 Descripción de los experimentos

Para el análisis de los datos, se realizaron 14 experimentos, resultando 14 listas de ortólogos. Estos experimentos fueron pensados para ir combinando rasgos, a través del operador de agregación OWA y de la media aritmética. Los rasgos a combinar fueron: la homología entre pares de secuencias, la longitud entre pares de secuencias, la pertenencia a los LCB y la distancia evolutiva.

La muestra utilizada para probar los experimentos, fue seleccionada, de forma tal que quedara balanceada con respecto a los ortólogos que resultan del software Inparanoid 7, que resulta la clasificación de referencia, y sólo el 0.0037% de todos los demás pares de genes que no fueron analizados. A continuación se explica la estructura de cada experimento y sus resultados según la prueba no paramétrica de "McNemar":

3.1.1 Experimentos realizados agregando rasgos con el operador OWA

Experimento 1: Se tiene en cuenta los 4 rasgos utilizados. Este experimento arroja una lista con 3253 ortólogos. De ellos sólo fueron clasificados correctamente 2010 y 1797 pares fueron clasificados como no ortólogos y si lo son según la clasificación utilizada como referencia, esto quiere decir que la cantidad de falsos positivos es muy elevada. Este experimento no clasifica como verdaderos negativos ningún elemento de la muestra.

Inparanoid & Experimento 1

	Experimento 1	
Inparanoid	0	1
0	3786	0
1	1797	2010

Figura 18: Resultados de la prueba de McNemar realizada al experimento 1.

Test Statistics[b]

	Inparanoid & Experimento 1
N	7593
Chi-Square[a]	1795.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 19: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 1.

Experimento 2: Se tiene en cuenta sólo tres rasgos: la homología de la secuencia, la longitud de las secuencias y la pertenencia a los LCB. Este experimento arroja una lista de 3212 pares de ortólogos. De ellos sólo tienen una correcta clasificación 1974 pares. Aunque los clasificados como verdaderos positivos disminuye con respecto al experimento anterior, se puede constatar que aumenta la clasificación de falsos positivos a 1833.

Inparanoid & Experimento 2

	Experimento 2	
Inparanoid	0	1
0	3786	0
1	1833	1974

Figura 20: Resultados de la prueba de McNemar realizada al experimento 2.

Test Statistics[b]

	Inparanoid & Experimento 2
N	7593
Chi-Square[a]	1831.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 21: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 2.

Experimento 3: Se tiene en cuenta sólo tres rasgos: la homología de la secuencia, la longitud de las secuencias y la distancia evolutiva. Este experimento proporciona una lista de 3183 pares de ortólogos. De ellos sólo tienen una correcta clasificación 1932 pares. La clasificación de falsos positivos sigue siendo elevada, son clasificados como tal

1875 pares. Se puede apreciar con respecto a los dos experimentos anteriores, que la clasificación de verdaderos positivos disminuyó y aumentó la clasificación de falsos positivos.

Inparanoid & Experimento 3

	Experimento 3	
Inparanoid	0	1
0	3786	0
1	1875	1932

Figura 22: Resultados de la prueba de McNemar realizada al experimento 3.

Test Statistics[b]

	Inparanoid & Experimento 3
N	7593
Chi-Square[a]	1873.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 23: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 3.

Experimento 4: Se tiene en cuenta sólo tres rasgos: la homología de la secuencia, la pertenencia a los LCB y la distancia evolutiva. Este experimento reporta una lista de 3185 pares de ortólogos. De ellos se clasifican como verdaderos ortólogos 2020. Sigue teniendo una elevada clasificación de falsos positivos de 1787 pares. Este experimento clasifica mejor que los anteriores mencionados, aumenta los verdaderos positivos y disminuyen los falsos positivos.

Inparanoid & Experimento 4

	Experimento 4	
Inparanoid	0	1
0	3786	0
1	1787	2020

Figura 24: Resultados de la prueba de McNemar realizada al experimento 4.

Test Statistics[b]

	Inparanoid & Experimento 4
N	7593
Chi-Square[a]	1785.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 25: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 4.

Experimento 5: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la longitud de la secuencia. Este experimento proporciona una lista de 3160 pares de ortólogos. De ellos sólo tienen una clasificación como verdaderos ortólogos 1916 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1891. Comparado con el experimento 4, la clasificación empeora, pues disminuye la cantidad de clasificados como verdaderos positivos y aumenta los falsos positivos. Hasta el momento se considera este experimento como el peor de los ya analizados.

Inparanoid & Experimento 5

	Experimento 5	
Inparanoid	0	1
0	3786	0
1	1891	1916

Figura 26: Resultados de la prueba de McNemar realizada al experimento 5.

Test Statistics[b]

	Inparanoid & Experimento 5
N	7593
Chi-Square[a]	1889.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 27: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 5.

Experimento 6: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la pertenencia a los LCB. Este experimento proporciona una lista de 3179 pares de ortólogos. De ellos son clasificados correctamente 2034 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1773. Comparado con el experimento 4, que hasta el momento es el de mejor resultado, tiene una mejor clasificación, debido a que aumenta la clasificación de verdaderos positivos y disminuye la de falsos positivos.

Inparanoid & Experimento 6

	Experimento 6	
Inparanoid	0	1
0	3786	0
1	1773	2034

Figura 28: Resultados de la prueba de McNemar realizada al experimento 6.

Test Statistics[b]

	Inparanoid & Experimento 6
N	7593
Chi-Square[a]	1771.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 29: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 6.

Experimento 7: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la distancia evolutiva. Este experimento proporciona una lista de 3157 pares de ortólogos. De ellos sólo tienen una correcta clasificación 1949 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1858. Este experimento empeora la clasificación comparado con los anteriores experimentos, exceptuando el experimento 5.

Inparanoid & Experimento 7

	Experimento 7	
Inparanoid	0	1
0	3786	0
1	1858	1949

Figura 30: Resultados de la prueba de McNemar realizada al experimento 7.

Test Statistics[b]

	Inparanoid & Experimento 7
N	7593
Chi-Square[a]	1856.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 31: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 7.

3.1.2 Experimentos realizados agregando rasgos con la media aritmética

Experimento 8: Se tiene en cuenta los 4 rasgos utilizados. Este experimento arroja una lista con 3246 ortólogos. De ellos sólo fueron clasificados correctamente 1996 pares. La clasificación de falsos positivos detecta 1811 pares. Comparado con el experimento 6, que hasta el momento es el mejor experimento, empeora la clasificación debido a que disminuye los clasificados como verdaderos positivos y aumenta los clasificados como falsos positivos. Comparado con el experimento 5, que es el peor experimento hasta el momento, clasifica mejor ya que aumenta los verdaderos positivos y disminuye los falsos positivos.

Inparanoid & Experimento 8

	Experimento 8	
Inparanoid	0	1
0	3786	0
1	1811	1996

Figura 32: Resultados de la prueba de McNemar realizada al experimento 8.

Test Statistics[b]

	Inparanoid & Experimento 8
N	7593
Chi-Square[a]	1809.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 33: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 8.

Experimento 9: Se tiene en cuenta tres rasgos: la homología de la secuencia, la longitud de las secuencias y la pertenencia a los LCB. Este experimento arroja una lista de 3213 pares de ortólogos. De ellos sólo tienen una correcta clasificación el 1981. Sigue teniendo una elevada clasificación de falsos positivos (no ortólogos que si son ortólogos con respecto a la clase) de 1826 pares. Con respecto al experimento 8, empeora la clasificación pues disminuye los clasificados como verdaderos positivos y aumenta los falsos positivos.

Inparanoid & Experimento 9

	Experimento 9	
Inparanoid	0	1
0	3786	0
1	1826	1981

Figura 34: Resultados de la prueba de McNemar realizada al experimento 9.

Test Statistics[b]

	Inparanoid & Experimento 9
N	7593
Chi-Square[a]	1824.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 35: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 9.

Experimento 10: Se tiene en cuenta sólo tres rasgos: la homología de la secuencia, la longitud de las secuencias y la distancia evolutiva. Este experimento proporciona una lista de 3182 pares de ortólogos. De ellos sólo tienen una correcta clasificación, según la clase, 1938 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1869 pares de ortólogos. Comparado con el experimento 6, la clasificación empeora debido a que la clasificación de verdaderos positivos disminuye y las de falsos positivos aumenta. Comparado con el experimento 5, que hasta el momento es el peor experimento, clasifica mejor debido a que aumenta ligeramente la clasificación de verdaderos positivos y disminuye los falsos positivos.

Inparanoid & Experimento 10

Inparanoid	Experimento 10	
	0	1
0	3786	0
1	1869	1938

Figura 36: Resultados de la prueba de McNemar realizada al experimento 10.

Test Statistics[b]

	Inparanoid & Experimento 10
N	7593
Chi-Square[a]	1867.001
Asymp. Sig.	.000

a. Continuity Corrected
b. McNemar Test

Figura 37: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 10.

Experimento 11: Se tiene en cuenta sólo tres rasgos: la homología de la secuencia, la pertenencia a los LCB y la distancia evolutiva. Este experimento proporciona una lista de 3184 pares de ortólogos. De ellos clasifica como verdaderos ortólogos 2020 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1787. Comparado con en el experimento 4, se puede percatar que clasifica exactamente igual que este.

Inparanoid & Experimento 11

Inparanoid	Experimento 11	
	0	1
0	3786	0
1	1787	2020

Figura 38: Resultados de la prueba de McNemar realizada al experimento 11.

Test Statistics[b]

	Inparanoid & Experimento 11
N	7593
Chi-Square[a]	1785.001
Asymp. Sig.	.000

a. Continuity Corrected
b. McNemar Test

Figura 39: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 11.

Experimento 12: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la longitud de la secuencia. Este experimento proporciona una lista de 3161 pares de ortólogos. De ellos sólo tienen una correcta clasificación el 1929 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1878. Este experimento con respecto a los anteriores tiene una peor clasificación, exceptuando el experimento 5 y el 3. Este experimento hasta el momento constituye el tercero más malo teniendo en cuenta su clasificación.

Inparanoid & Experimento 12

	Experimento 12	
Inparanoid	0	1
0	3786	0
1	1878	1929

Figura 40: Resultados de la prueba de McNemar realizada al experimento 12.

Test Statistics[b]

	Inparanoid & Experimento 12
N	7593
Chi-Square[a]	1876.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 41: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 12.

Experimento 13: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la pertenencia a los LCB. Este experimento proporciona una lista de 3179 pares de ortólogos. De ellos sólo tienen una correcta clasificación el 2033. Sigue teniendo una elevada clasificación de falsos positivos de 1774 pares. Este experimento, después del experimento 6, constituye el mejor clasificador de ortólogos.

Inparanoid & Experimento 13

	Experimento 13	
Inparanoid	0	1
0	3786	0
1	1774	2033

Figura 42: Resultados de la prueba de McNemar realizada al experimento 13.

Test Statistics[b]

	Inparanoid & Experimento 13
N	7593
Chi-Square[a]	1772.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 43: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 13.

Experimento 14: Se tiene en cuenta sólo dos rasgos: la homología de la secuencia y la distancia evolutiva. Este experimento proporciona una lista de 3108 pares de ortólogos. De ellos sólo tienen una correcta clasificación, según la clase, 1944 pares. Sigue teniendo una elevada clasificación de falsos positivos de 1863. Comparado con el mejor experimento empeora la clasificación debido a que disminuye la selección de verdaderos positivos y aumentan los falsos positivos. Comparado con experimento 5, hasta el momento es el peor que clasifica, evidencia una mejoría en la clasificación de verdaderos positivos y una disminución de los falsos positivos.

Inparanoid & Experimento 14

	Experimento 14	
Inparanoid	0	1
0	3786	0
1	1863	1944

Figura 44: Resultados de la prueba de McNemar realizada al experimento 14.

Test Statistics[b]

	Inparanoid & Experimento 14
N	7593
Chi-Square[a]	1861.001
Asymp. Sig.	.000

a. Continuity Corrected

b. McNemar Test

Figura 45: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 14.

3.1.3 Análisis del índice de coincidencia entre el algoritmo de referencia y el algoritmo diseñado por la presente investigación para los 14 primeros experimentos

A partir de la prueba no paramétrica para datos dicotómicos realizada mediante el "Test de McNemar", se realiza un análisis del índice de coincidencia de la clasificación proporcionada por el algoritmo Inparanoid 7 contra el algoritmo implementado por la presente investigación, este último con 14 corridas del mismo variando el parámetro de selección de rasgos para la agregación. Todos los experimentos evidencian un porciento similar de coincidencia de clasificación, pero los más altos son el experimento 6 y el 13 con un 76.5% y un 76.4% de coincidencia respectivamente.

Experimento	Índice de coincidencia
1	76.33%
2	75.86%
3	75.31%
4	76.47%
5	75.10%
6	76.65%
7	75.53%
8	76.15%
9	75.95%
10	75.39%
11	76.47%
12	75.27%
13	76.64%
14	75.46%

Tabla 2: Índice de coincidencia de la clasificación con el Inparanoid 7.

La tabla 2 relaciona los verdaderos positivos (los casos comunes clasificados como ortólogos) y los falsos negativos (los casos comunes clasificados como no ortólogos) de la siguiente forma:

Índice de coincidencia = $\frac{FN+VP}{TC}$

FN: falsos negativos.

VP: verdaderos positivos.

TC: total de elementos muestreados.

La figura 46 muestra la gráfica que resume el anterior análisis:

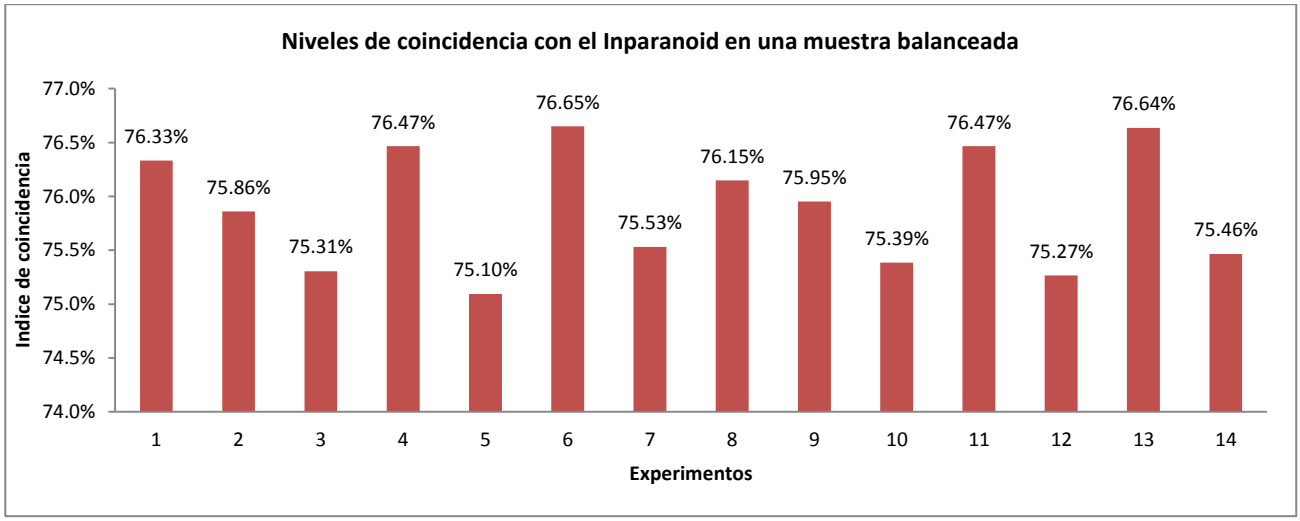


Figura 46: Niveles de coincidencia con el Inparanoid en una muestra balanceada.

A continuación se muestra la tabla 3 que analiza el índice de coincidencia teniendo en cuenta dos clases, los clasificados correctamente como verdaderos ortólogos (verdaderos positivos) y los clasificados correctamente como no ortólogos (falsos negativos). Se puede evidenciar que clasifica correctamente todos los no ortólogos para la muestra dada, y que la mejor clasificación para ortólogos es la arrojada por los experimentos 6 y 13.

	Indice de coincidencia	
Experimento	No Ortólogos	Ortólogos
1	100.00%	52.80%
2	100.00%	51.85%
3	100.00%	50.75%
4	100.00%	53.06%
5	100.00%	50.33%
6	100.00%	53.43%
7	100.00%	51.20%
8	100.00%	52.43%
9	100.00%	52.04%
10	100.00%	50.91%
11	100.00%	53.06%
12	100.00%	50.67%
13	100.00%	53.40%
14	100.00%	51.06%

Tabla 3: Índice de coincidencia de la clasificación con el Inparanoid 7 teniendo en cuenta dos clases.

La tabla 3 divide por clases el análisis donde se obtiene un índice de coincidencia en la clasificación correcta de pares ortólogos y la clasificación correcta de los pares no ortólogos como sigue:

Índice de coincidencia de pares de ortólogos $= \frac{VP}{TC}$

Índice de coincidencia de pares no ortólogos $= \frac{FP}{TC}$

FN: falsos negativos.

VP: verdaderos positivos.

TC: total de elementos muestreados.

En la figura 47 se evidencia los anteriores resultados en una gráfica:

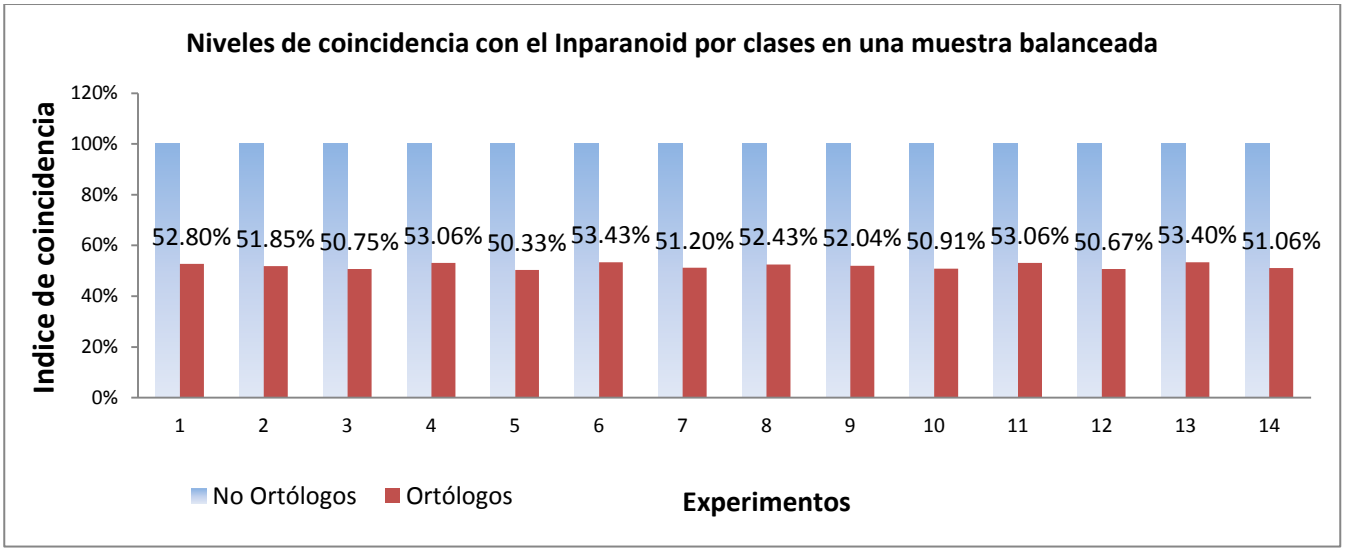


Figura 47 Niveles de coincidencia con el Inparanoid por clases en una muestra balanceada.

3.1.4 Experimentos realizados después de haber ejecutado el algoritmo solo con los datos de la muestra balanceada

Experimento 15: Para visualizar el comportamiento de la clasificación, se decide ejecutar nuevamente el algoritmo, pero sólo con los datos escogidos en la muestra balanceada y con el experimento 6, que es el que mejor clasificador de ortólogos comprobado por la presente investigación. Los resultados arrojados muestran una clasificación similar al experimento 6 diferenciándose en los verdaderos positivos, actualmente no clasifica 3 y los incrementa en los falsos positivos.

Inparanoid y 1a1

	1a1	
Inparanoid	0	1
0	3786	0
1	1776	2031

Figura 48: Resultados de la prueba de McNemar realizada al experimento 15.

Estadísticos de contraste[b]

	Inparanoid y 1a1
N	7593
Chi-cuadrado[a]	1774.001
Sig. asintót.	.000

a. Corregido por continuidad
b. Prueba de McNemar

Figura 49: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 15.

Como el software Inparanoid 7, detecta las relaciones de ortólogos de uno a uno, uno a muchos, muchos a muchos y el algoritmo de la presente investigación implementa sólo las relaciones uno a uno, se realizó una variante de BUS que permitiera clasificar como ortólogos aquellas relaciones uno a muchos y muchos a muchos teniendo como requisito que perteneciera a una misma hebra (strand). A continuación se muestran los resultados que evidencia una mejora en la clasificación.

Experimento 16: Para el experimento teniendo en cuenta las relaciones uno a uno y uno a muchos a partir de los grupos de homología hallados por el BUS reporta coincidencias de verdaderos ortólogos de 2194 pares. Disminuyendo la cantidad de falsos positivos.

Inparanoid y 1a1_1aM

	1a1_1aM	
Inparanoid	0	1
0	3786	0
1	1613	2194

Figura 50: Resultados de la prueba de McNemar realizada al experimento 16.

Estadísticos de contraste[b]

	Inparanoid y 1a1_1aM
N	7593
Chi-cuadrado[a]	1611.001
Sig. asintót.	.000

a. Corregido por continuidad
b. Prueba de McNemar

Figura 51: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 16.

Experimento 17: Este experimento tiene en cuenta las relaciones uno a uno, uno a muchos y muchos a muchos, reporta coincidencias de verdaderos ortólogos de 2688 pares. Disminuyendo la cantidad de falsos positivos.

Inparanoid y 1a1_1aM_MaM

	1a1_1aM_MaM	
Inparanoid	0	1
0	3784	2
1	1119	2688

Figura 52: Resultados de la prueba de McNemar realizada al experimento 17.

Estadísticos de contraste[b]

	Inparanoid y 1a1_1aM_MaM
N	7593
Chi-cuadrado[a]	1111.022
Sig. asintót.	.000

a. Corregido por continuidad
b. Prueba de McNemar

Figura 53: Estadísticos de contraste arrojados por la prueba de McNemar realizada al experimento 17.

3.1.5 Análisis del índice de coincidencia entre el algoritmo de referencia y el algoritmo diseñado por la presente investigación del experimento 15 al 17

A partir de la prueba no paramétrica para datos dicotómicos realizada mediante el "Test de McNemar", se realiza un análisis del índice de coincidencia de la clasificación proporcionada por el algoritmo Inparanoid 7 contra el algoritmo implementado por la presente investigación, incorporándole el análisis de la selección de las relaciones uno a muchos y muchos a muchos, este último con 3 corridas del mismo variando el parámetro de selección de relaciones a clasificar. El mejor resultado se alcanza para cuando se escoge la relación uno a uno, uno a muchos y muchos a muchos con un 85.24% de coincidencia. La tabla 4 fue realiza siguiendo el mismo criterio de la tabla 3.

Experimento	Índice de coincidencia
11	76.61%
11-1M	78.76%
11-1M-MM	85.24%

Tabla 4: Índice de coincidencia de la clasificación con el Inparanoid.

La figura 54 muestra la gráfica que resume el anterior análisis:

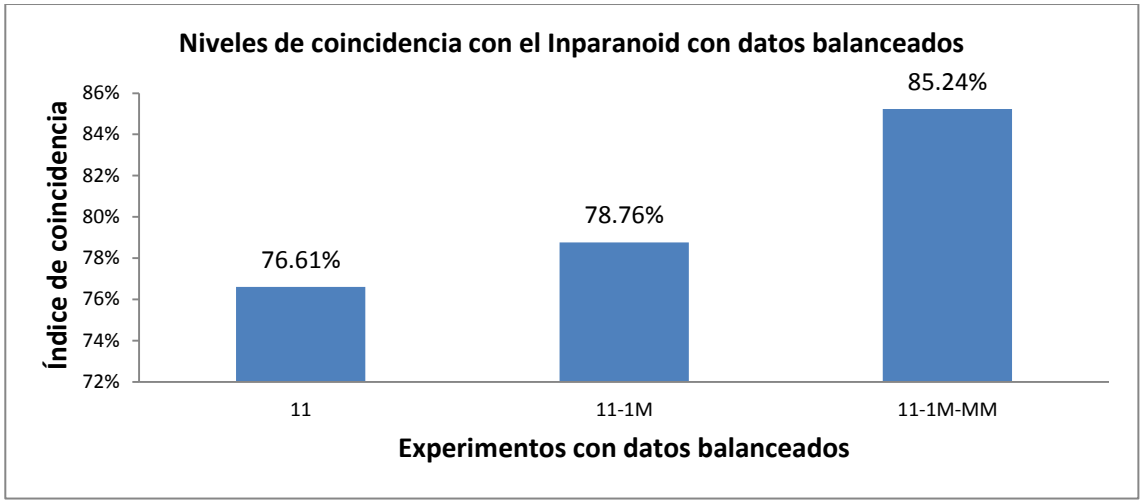


Figura 54: Niveles de coincidencia con el Inparanoid con datos balanceados.

A continuación se muestra la tabla 5 que analiza el índice de coincidencia teniendo en cuenta dos clases, los clasificados correctamente como verdaderos ortólogos (verdaderos positivos) y los clasificados correctamente como no ortólogos (falsos negativos). Se puede evidenciar que clasifica correctamente todos los no ortólogos para la muestra dada, y que la mejor clasificación para ortólogos es la arrojada por los experimentos 6 y 13.

	Índice de coincidencia	
Experimento	No Ortólogos	Ortólogos
11	100.00%	53.35%
11-1M	100.00%	57.63%
11-1M-MM	99.95%	70.61%

Tabla 5: Índice de coincidencia de la clasificación con el Inparanoid teniendo en cuenta dos clases.

En la siguiente figura 55 se evidencia los anteriores resultados en una gráfica:

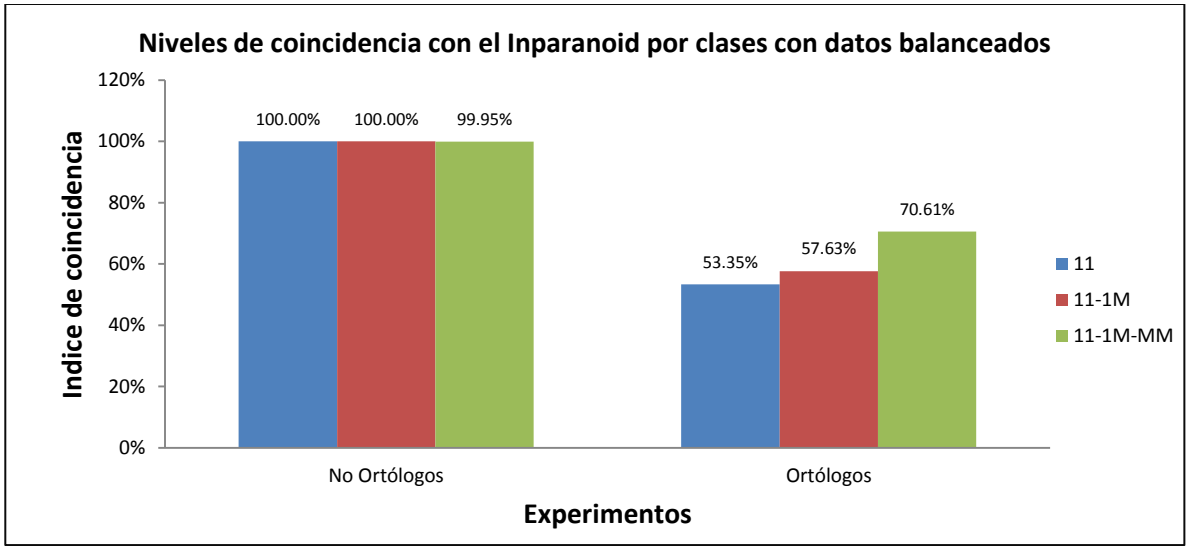


Figura 55: Niveles de coincidencia con el Inparanoid por clases con datos balanceados.

Conclusiones

El presente capítulo muestra un análisis estadístico realizado mediante el "Test de McNemar" con el propósito de comparar los resultados de las clasificaciones arrojadas por las distintas corridas del algoritmo propuesto en la presente investigación y el Inparanoid 7. Se pudo constatar que existen diferencias significativas, por lo que no hay una clasificación similar entre los dos algoritmos comparados. Las tablas de contingencias, que resultan de aplicar el "test" realizado, permitieron calcular el índice de coincidencia entre ambos algoritmos, teniendo en cuenta los falsos negativos y los verdaderos positivos. El experimento 17 arrojó el mejor resultado (85.4% de coincidencias), luego de haber transformado el algoritmo para el particionado de modo que se tengan en cuenta no sólo las relaciones de uno a uno, sino también las relaciones de uno a muchos y las de muchos a muchos.

CONCLUSIONES

Se construyó un algoritmo de detección de genes ortólogos combinando rasgos de los genes de dos genomas como la información de la homología, la longitud de las secuencias, la pertenencia a regiones conservadas teniendo en cuenta los reordenamientos globales de los genomas y la relación evolutiva entre los genes. Esta combinación de rasgos se logró utilizando el operador OWA y la media aritmética como métodos de agregación para el cálculo de las matrices de distancia. Además se implementó la técnica de particionado de grafos BUS.

En la validación del algoritmo con relación a la clasificación del algoritmo Inparanoid 7.0 para los genomas *Saccharomyces Cervisiae* y *Schizosaccharomyces Pombe*, se pudo constatar que aún persiste un alto número de falsos positivos, aunque a partir de la transformación del algoritmo BUS para permitir la clasificación incluyendo las relaciones de uno a muchos y de muchos a muchos, se logró un incremento de los verdaderos positivos y una disminución de los falsos positivos, logrando un nivel de coincidencia del 85.4 %.

Los mejores resultados de clasificación se obtuvieron combinando los rasgos de la homología de las secuencias y la pertenencia a los bloques conservados.

El por ciento de falsos positivos resultó ser bajo lo cual evitaría asignar funciones de genes ortólogos a genes con funciones desconocidas.

RECOMENDACIONES

1. Utilizar la distancia 4 para la ejecución de nuevos experimentos.
2. Utilizar el software “Orthocluster” para el cálculo de los bloques de orden conservado (“synteny blocks”).
3. Probar otros algoritmos de particionamiento de grafos en la fase de agrupamiento.
4. Mejorar la selección de los valores de los parámetros del alineamiento para elevar la precisión de la puntuación de similitud entre las secuencias.
5. Optimizar el vector de pesos utilizado en el operador de agregación OWA.
6. Estudiar otros rasgos que caractericen a los genes con vistas a mejorar la precisión de los algoritmos.

Bibliografía

BIBLIOGRAFÍA

ALEXEYENKO, A., TAMAS, I., LIU, G. & SONNHAMMER, E. L. L. 2006. Automatic clustering of orthologs and inparalogs shared by multiple proteomes. *Bioinformatics,* 22**,** e9-e15.

ALTSCHUL, S. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research,* 25**,** 3389–3402.

BERGMANN, R. 2002. *Experience management: foundations, development methodology, and Internet-based applications,* Hildesheim, Germany.

BLAY, M. E. 2009. *Herramientas Computacionales para la Comparación de Genomas* Licenciatura en Ciencia de la Computación, Universidad Central "Marta Abreu" de Las Villas. .

CHEN, F., MACKEY, A. J., JR., C. J. S. & ROOS, D. S. 2006. OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Research,* 34**,** 363-368.

CHEN, X., ZHENG, J., FU, Z., NAN, P., ZHONG, Y. & LONARDI, S. 2005. Assignment of Orthologous Genes via Genome Rearrangement. *IEEE/ACM Trans. Comput. Biology Bioinform,* 2**,** 302-315.

DARLING, A. C. E., MAU, B. & BLATTNER, F. R. 2004. MAUVE: Multiple Alignment of Conserved Genomic Sequence With Rearrangements. *Genome Res.,* 14**,** 1394-1403

DEZA, E. 2006. *Dictionary of Distances,* Elsevier.

DUCH., W. 2000. Similarity-based methods: a general framework for classification, approximation and association. *Control and Cybernetics,* 29**,** 1-30.

ELEMENTS, T. N. H. G. R. I. M. O. E. O. D. 2012. *modENCODE* [Online]. Available: http://www.modencode.org/ [Accessed 10 de Marzo 2012].

FU, Z., CHEN, X., VACIC, V., NAN, P., ZHONG, Y. & JIANG, A. T. 2007. MSOAR: A High-Throughput Ortholog Assignment System Based on Genome Rearrangement. *COMPUTATIONAL BIOLOGY,* 14**,** 1160–1175.

HIRSH, A. E. & FRASER, H. B. 2001. Protein dispensability and rate of evolution. *Nature,* 411**,** 1046-1049.

HUBALEK, Z. 1981. Coefficients of association and similarity, based on binary (presence-absence) data: An evaluation. . *Prague, Czechoslovak Academy of Science, Institute of Parasitology.* .

HULSEN, T., VLIEG, J. D., LEUNISSEN, J. A. & GROENEN, P. M. 2006b. Testing statistical significance scores of sequence comparison methods with structure similiraty. *BMC Bioinformatics,* 7**,** 1-13.

Bibliografía

HWANG, W., CHO, Y.-R., ZHANG, A. & RAMANATHAN, M. 2006. A novel functional module detection algorithm for protein-protein interaction networks. *Algorithms Mol Biol,* 1.

JACCARD, P. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles.,* 37**,** 547-579.

KAMVYSSELIS, M. K. 2003. *Computational comparative genomics: genes, regulation, evolution.* Doctor of Philosophy in Computer Science, Massachusetts Institute of Technology

KUZNIAR, A., VAN HAM, R. C. H. J., PONGOR, S. & LEUNISSEN, J. A. M. 2008. The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics,* 30**,** 1-13.

LEE, Y., SULTANA, R., PERTEA, G., CHO, J., KARAMYCHEVA, S., TSAI, J. & 2002 Cross-Referencing Eukaryotic Genomes: Tigr Orthologous Gene Alignments (Toga). *Genome Res.,* 12**,** 493-502.

LI, L., STOECKERT, C. J. J. & ROOS, D. S. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.,* 13**,** 2178-2189.

LINARD, B., THOMPSON, J. D., POCH, O. & LECOMPTE, O. 2011. OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics,* 12**,** 1471-2105.

LINDAHL, J. M. M. 2008. *Nuevas extensiones a los operadores OWA y su aplicación a los métodos de decisión.* Doctorado, Universidad de Barcelona.

LIPKUS, A. H. 1999. A proof of the triangle inequality for the Tanimoto distance. *Journal of Mathematical Chemistry,* 26**,** 263-265.

LÓPEZ, M. L., GUTIÉRRES, A. U. L., ESPUÑEZ, T. D. R. S. & TORRES, A. M. R. 2005. ¿Que sabe usted acerca de ... genómica? *Revista Mexinaca de Ciencias Farmacéuticas,* 36**,** 42-44.

MATLAB. 2010. *Matlab R2010a Help* [Online]. Available: http://www.mathworks.com [Accessed].

MAUVE. 2010. *Mauve User Guide* [Online]. Wisconsin-Madison: Genome Center of Wisconsin. Available: http://gel.ahabs.wisc.edu/mauve/mauve-user-guide [Accessed].

NCBI. 2011. *HomoloGene* [Online]. Available: http://www.ncbi.nlm.nih.gov/homologene [Accessed 10 de Marzo 2012].

NEEDLEMAN, S. B. & WUNSCH, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.,* 48**,** 443-453.

Bibliografía

O'BRIEN, K. P., REMM, M. & SONNHAMMER, E. L. L. 2005. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Research,* 33**,** D476-D480.

ÖSTLUND, G., SCHMITT, T., FORSLUND, K., KÖSTLER, T., MESSINA, D. N., ROOPRA, S., FRINGS, O. & SONNHAMMER, E. L. L. 2010. InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. *Nucleic Acids Research,* 38**,** D196–D203.

OVERBEEK, R., FONSTEIN, M., D'SOUZA, M., PUSCH, G. D. & MALTSEV, N. 1999. The use of gene clusters to infer functional coupling. *Proc. Natl. Acad. Sci. USA,* 96**,** 2896–2901.

OZER, H. G., C.J., ZHANG, F. & Y., B. 2004. Clustering of Eukaryotic Orthologs Based on Sequence and Domain Similarities Using the Markov Graph-Flow Algorithm. *Advances in Bioinformatics and its Applications*.

PEARSON, W. R. 1990. Rapid and Sentive Sequence Comparison with PASTP and FASTA. *Methods Enzymol,* 183**,** 63-98.

REMM, M., STORM, C. E. V. & SONNHAMMER, E. L. L. 2001. Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons. *J. Mol. Biol.,* 314**,** 1041-1052.

SALICHOS, L. & ROKAS, A. 2011. Evaluating Ortholog Prediction Algorithms in a Yeast Model Clade. *PLoS ONE,* 6**,** 1-11.

SÁNCHEZ, R. & GRAU, R. 2009. An algebraic hypothesis about the primeval genetic code architecture. *Mathematical Biosciences,* 221**,** 60-76.

SHAYE, D. D. & GRENWALD, I. 2011. OrthoList: A Compendium of C. elegans Genes with Human Orthologs. *PLoS One,* 6**,** 1-11.

SHI, G., PENG, M.-C. & JIANG, T. 2011. MultiMSOAR 2.0: An Accurate Tool to Identify Ortholog Groups among Multiple Genomes. *PLoS ONE,* 6**,** 1-9.

SINGH, A. 2011. *Architecture value mapping using fuzzy cognitive maps as a reasoning mechanism for multi-criteria conceptual design evaluation.* Doctor, Missouri University of Science And Technology.

SMITH, T. F. & WATERMAN, M. S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.,* 147**,** 195–197.

STEVENS, R., HILLER, E., DÖRFLINGER, M., GABALDON, T., SCHWARZMÜLLER, T., KUCHLER, K. & RUPP, S. 2009. Comprehensive gene deletion study to identify cell wall organisation and structure in Candida glabrata. *International journal of medical microbiology*, 299.

Bibliografía

TATUSOV, R. L., FEDOROVA, N. D., JACKSON, J. D., JACOBS, A. R., KIRYUTIN, B. & KOONIN, E. V. 2003. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 4**,** 1-14.

TATUSOV, R. L., KOONIN, E. V. & LIPMAN, D. J. 1997. A Genomic Perspective on Protein Families. *SCiencie,* 278**,** 631-637.

TATUSOV, R. L., NATALE, D. A., V.GARKAVTSEV, I., TATUSOVA, T. A., T.SHANKAVARAM, U. & KIRYUTIN, B. 2001. The COG database: new developments in phylogenetic classification of protein from complete genomes. *Nucleic Acids Research,* 29**,** 22-28.

TOWFIC, F., G.W.H.M & HONAVAR, V. Year. Detection of Gene Orthology Based On Protein-Protein Interaction Networks. *In:* IEEE International Conference on Bioinformatics and Biomedicine, BIBM, 2009 Washington DC, USA.

WALL, D. P., FRASER, H. B. & HIRSH, A. E. 2003. Detecting putative orthologs. *Bioinformatics,* 19**,** 1710–1711.

WEBBER, C. A. P., CHRIS P. 2004. Genes and Homology. *Current Biology,* 14**,** R332.

YAGER, R. R. 1988. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics,* 18**,** 183–190.

YAGER, R. R. 1996a. Constrained OWA agregation. *Fuzzy sets and systems,* 1**,** 89-101.

YAGER, R. R. 1996b. Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems,* 11**,** 49-73.