Tesis de Maestría en Bioinformática

Programa de Desarrollo de las Ciencias Básicas

Análisis del transcriptoma de Trypanosoma cruzi: estudios de secuenciación profunda en tres estadios del ciclo de vida del parásito

Autora:
Adriana Errico

Orientadores:
Fernando Álvarez
Biomatemática. Facultad de Ciencias

Gustavo Guerberoff
IMERL. Facultad de Ingeniería

Índice

Introducción

1. Tripanosomiasis

Los tripanosomátidos son protozoarios flagelados de amplia distribución, y conforman una de las dos familias del orden *Kinetoplastida* (*Bodonidae* y *Trypanosomatidae*). *Bodonidae* comprende organismos de vida libre con dos flagelos. Los miembros de *Trypanosomatidae* presentan un único flagelo y son parásitos (en principio de insectos, aunque varios grupos se han adaptado a un segundo hospedador, que puede ser un vertebrado o una planta).

Dentro de la familia *Trypanosomatidae* destacamos al género *Trypanosoma*, el cual incluye especies que son parásitos de vertebrados, usualmente transmitidos por vectores, generalmente insectos, ocasionalmente sanguijuelas (en el caso de Tripanosomas anfibios). La mayoría de los Tripanosomas tienen ciclos de vida que alternan entre un hospedero vertebrado y un hospedero invertebrado, en el que se desarrollan en el tracto digestivo. Muchos Tripanosomas permanecen en sus hospederos sin que estos presenten síntomas, en esos casos el tejido se ha adaptado fisiológicamente a la presencia de los parásitos por largos períodos de tiempo. Estos animales actúan como reservorio para esas especies. Pero otros Tripanosomas son patogénicos y las afecciones que producen se les denominan en forma genérica como tripanosomiasis. [Smyth, 1994]

La transmisión entre hospederos vertebrados se da por invertebrados (llamados vectores) que se alimentan de sangre. Poco después que la sangre llega al intestino del insecto, la forma trypomastigota se convierte a epimastigota y luego (directamente o por fisión) dan lugar a otras formas, usualmente llamados tripanosomas metacíclicos los que tienen capacidad infectiva.

Hay dos enfermedades principales en el hombre causadas por especies de tripanosomas [Smyth, 1994]:

Enfermedad del sueño o tripanosomiasis africana, es causada por *Trypanosoma brucei gambiense* y *Trypanosoma brucei rhodesiense*. Ocurre en África tropical y el vector es la mosca tsetse del género *Glossina*.

Enfermedad de Chagas o tripanosomiasis americana, es causada por *Trypanosoma cruzi*. Ocurre en América Central y del Sur y los vectores son insectos de la familia *Reduviidae*.

A pesar de que existen muchas similitudes entre *T. cruzi* y *T. brucei*, tales como que ambos son transmitidos por vectores que son insectos y comparten muchos aspectos de su fisiología y bioquímica básica, hay profundas diferencias en el nivel de la interfaz huésped-parásito que impiden que la información obtenida a partir de uno de los organismos sea útil para el otro. [Barret et al., 2003]

1.1. Historia evolutiva

Comparaciones de secuencias del gen 18S ARNr obtenidas a partir de diferentes especies de tripanosomas de diferentes huéspedes, combinado con otras aproximaciones moleculares, sugieren que el género *Trypanosoma* es monofilético (ver figura 1.1). *T. brucei* y *T. cruzi* comparten un ancestro común hace alrededor de 100 millones de años. Esta datación indica que los seres humanos han estado expuestos a las tripanosomiasis africanas concomitantemente con su evolución. Sin embargo los tripanosomas americanos evolucionaron independientemente del *Homo sapiens*, quien probablemente se unió al rango de huéspedes de *T. cruzi* entre 12.000 y 40.000 años atrás, cuando el hombre llegó al continente Americano. [Barret et al., 2003]

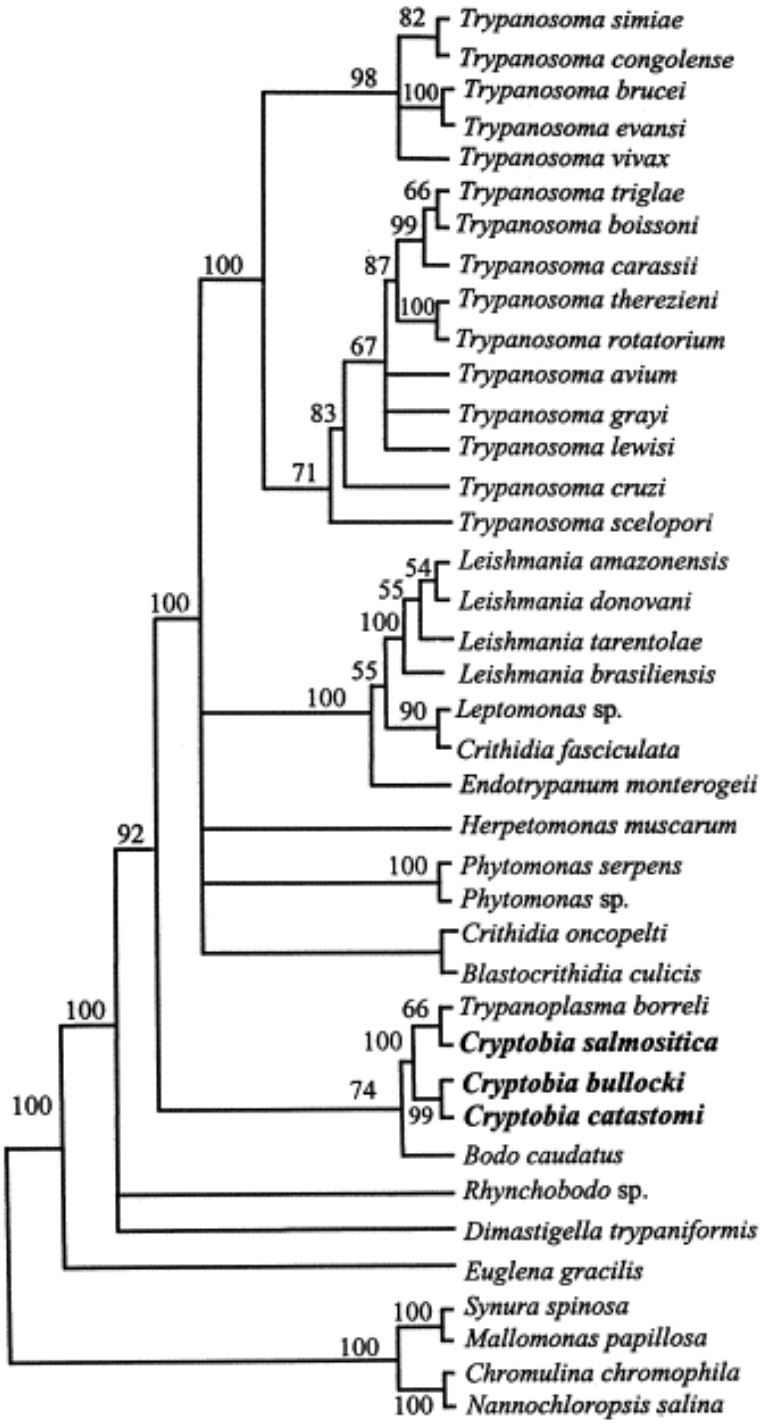


Figura 1.1: filogenia de kinetoplástidos. Puede apreciarse que el género *Trypanosoma* es monofilético. [Wright et al., 1999]

1.2. Biología molecular y genómica de tripanosomátidos

T. cruzi presenta una estructura poblacional clonal, lo que significa que existen diversas cepas o linajes del parásito con características genéticas muy homogéneas internamente; pero las distintas cepas pueden llegar a ser sustancialmente divergentes. *T. cruzi* es un eucariota cuyo genoma es rico en secuencias repetidas, los genes se disponen en arreglos de tipo repetidos en tándem con espacios intergénicos pequeños y pocas secuencias promotoras identificadas. [Gómez y Monteón, 2008]

El proyecto de secuenciación de *T. cruzi* finalizado en el 2005, empleó la cepa CL-Brener. A partir de este análisis se estableció lo siguiente [Gómez y Monteón, 2008]:

- el genoma haploide es de 67 Mb
- presenta alta conservación de sintenia con los restantes tripanosomas, o sea, sus haplotipos presentan un alto grado de estabilidad evolutiva.
- el número haploide de genes se estimó en alrededor de 12.000
- el genoma presenta un alto número de secuencias repetidas, que constituyen hasta el 50% y la mayoría de estas secuencias repetidas son familias de genes que codifican para proteínas de membrana (mucinas, trans-sialidasas, etc.)

La cepa CL-Brener es híbrida, por lo que no constituyó la mejor elección a efectos de tener una cepa de referencia. Esto presenta algunas dificultadas a la hora de usar estos datos como referencia para el ensamblado del genoma de otras cepas.

Las secuencias de repetición se pueden clasificar en dos grupos dentro del genoma: repeticiones agrupadas (*clustered repeats*) y repeticiones dispersas (*dispersed repeats*). Dentro de las primeras se encuentran los microsatélites, minisatélites y los satélites. Dentro de las segundas se tienen los SINE y LINE. Los microsatélites son secuencias de repetición de 2-5 pb organizadas en grupos, con un tamaño promedio que puede alcanzar hasta los 1.000 pb. Los minisatélites son unidades de repetición de cerca de 15 pb y que pueden tener un tamaño muy variable, que va desde 0,5 a 30 kb. El ADN satélite contiene secuencias de repetición de 200 pb organizadas como *clusters* y que están localizados en la región de la heterocromatina de los cromosomas. Los LINE (*Long Interspersed Nucleotide Elements*: elementos nucleotídicos largos interdispersos) son elementos que codifican para enzimas involucradas en su propia movilización, mientras que los SINE (*Short Interspersed Nucleotide Elements*: elementos nucleotídicos cortos interdispersos) son reconocidos y movilizados por la maquinaria de los LINE. [Gómez y Monteón, 2008]

Se reconoce que la regulación de la expresión de genes en tripanosomátidos es básicamente postranscripcional. El control de la expresión se ve afectado por la estabilidad del ARNm. En tripanosomátidos el ARNm es policistrónico, ya que se cotranscriben varios genes en el mismo ARNm. [Gómez y Monteón, 2008]

Los genes están organizados en unidades de transcripción con asimetría o polaridad de hebra. Con frecuencia existe un cambio de hebra en la región intermedia (región de switch transcripcional), de modo que estas unidades de transcripción se alternan en cuanto a la hebra que ocupan. Como producto de la transcripción se generan transcriptos policistrónicos, esto es, grandes moléculas de ARN mensajero que contienen la información codificante para distintas proteínas. Es importante destacar que esto implica que los genes estén organizados en grupos, y todos los genes de uno de estos bloques transcripcionales tienen que estar ubicados en la misma hebra. Esta situación también implica que la regulación de la expresión de los mismos sea de tipo postranscripcional, pues como ya fue mencionado se debe considerar el grupo como una unidad policistrónica. Las distintas unidades policistrónicas alternan a lo largo de los cromosomas. La transición entre dos bloques conlleva un cambio de sentido en la transcripción (switch transcripcional), y la existencia de una región intermedia localizada entre dos bloques consecutivos (ver figura 1.2) [Andersson et al., 1998]

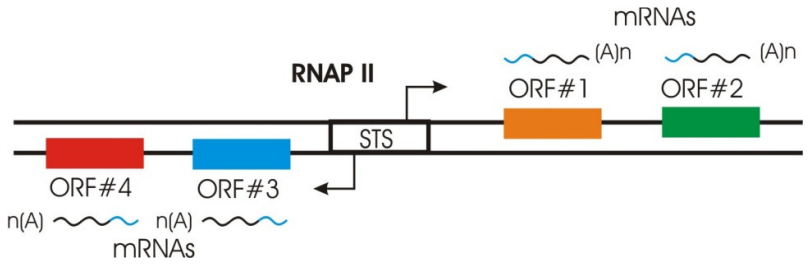


Figura 1.2: ejemplo de una de las regiones strand switch (STS) desde cuyos extremos radian dos grupos de genes.

Las moléculas de ARN mensajero maduro comienzan con una secuencia de 39 nucleótidos denominada *spliced leader* o miniexón, la cual se agrega durante el proceso de maduración del ARNm. A continuación siguen la región 5' (no codificante), el marco abierto de lectura (ORF), la región 3' (no codificante), y la cola de poli-A (que también se agrega durante la maduración). El proceso durante el cual se fragmenta el ARN policistrónico y se agrega el *spliced leader* se denomina *trans-splicing*. En la mayoría de los genes de eucariotas la maduración se produce por un proceso diferente denominado *cis-splicing*, consistente en la eliminación de intrones y empalme de los exones. El

agregado de la cola poli-A en los tripanosomas es similar al del resto de los eucariotas [Liang et al., 2003] [Mandelboim et al., 2003].

Ambos UTR's (3' y 5') de algunos genes han demostrado tener actividad en el nivel postranscripcional. Brandão y Jiang examinaron las regiones UTR de algunos genes de *T. cruzi* y encontraron que en 5' UTR, el 80 % del largo de las muestras era inferior a 120 pb. Las secuencias 3' UTR presentan un rango más amplio en la distribución de largos, variando ente 40 a más de 1000 pb. En 3' UTR, el 80 % de los largos son inferiores a las 450 pb. Esto indica un valor medio del largo para cada UTR (el cociente entre 3' UTR y 5' UTR) de 4,12. Este valor es del orden del encontrado para vertebrados. Dado que en los vertebrados la región 3' UTR está involucrada en ciertos mecanismos en el nivel postranscrpcional y *T. cruzi* también controla la expresión de sus genes a este nivel, Brandão y Jiang concluyen que es esperable, basados en el cociente de UTR's, que la región 3' UTR esté implicada en ese tipo de control. Se ha observado que genes con regiones UTR cortas estás asociados a mayor expresión. [Brandão y Jiang, 2009]

Estos valores de largos para las regiones UTR son considerados más adelante, para realizar el mapeo, ya que se mapeó bajo dos condiciones. Por un lado tomando como referencia exclusivamente las regiones codificantes, lo que se denominó como "genes sin margen" y por otro, considerando además las regiones aledañas a los genes, que se refiere en adelante como "genes con margen". Para determinar el tamaño de cada margen se tomó en consideración los resultados aportados por Brandão y Jiang (2009).

2. *Trypanosoma cruzi* y enfermedad de Chagas

La *tripanosomiasis americana o enfermedad de Chagas* es endémica en muchos países de América del Norte, Central y del Sur predominando desde México hasta Argentina. Se estima que 65 millones de personas viven en áreas de exposición y están en riesgo de contraer la enfermedad, que afecta a unos 8 millones de personas, provocando más de 12.000 muertes al año. [web: página de la Organización Panamericana de la Salud -paho-]

La enfermedad tiene mayor prevalencia en las regiones rurales de América del Sur. El agente etiológico de esta enfermedad es *T. cruzi*, el cual infecta animales salvajes y domésticos, además de seres humanos. [Ramírez Macías, 2012] [Hamilton et al., 2007]

Los tripanosomátidos presentan diferentes estadios a lo largo de su ciclo de vida (ver figura 1.3). Los estadios se clasifican en base a la morfología del

parásito y difieren según la posición del cinetoplasto respecto al núcleo y el flagelo.

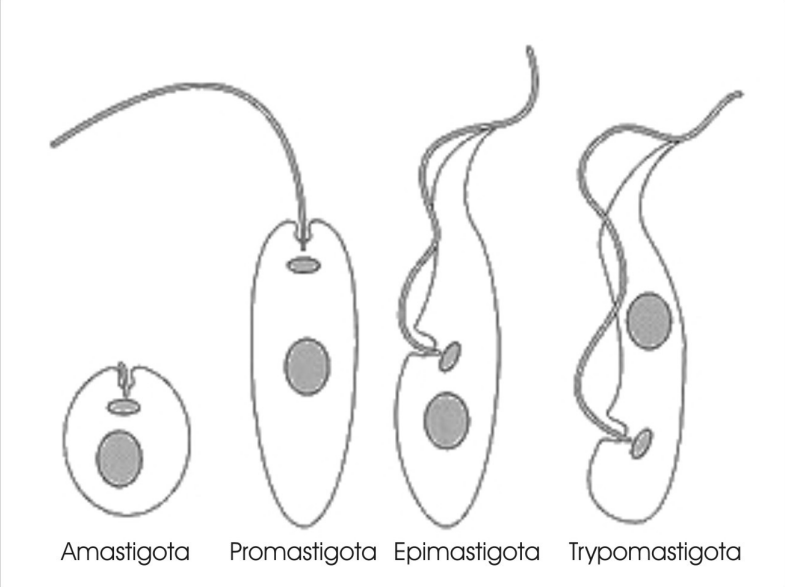


Figura 1.3: formas celulares de los tripanosomátidos. Imagen adaptada de http://en.wikipedia.org/wiki/Trypanosoma_cruzi

En particular, en *T. cruzi* se distinguen cuatro estadios en su ciclo de vida, a saber: amastigota (intracelular), epimastigota (en el tubo digestivo del insecto), trypomastigota (sanguíneos y metacíclicos) según su morfología.

La fuente de transmisión principal para humanos se da a través de sus insectos vectores, los cuales son chinches hematófagas del grupo de los triatomineos, conocidos vulgarmente como vinchucas, pero también puede darse por transfusión de sangre infectada, de la madre al hijo y en raros casos se ha detectado transmisión de la enfermedad por vía de la lactancia. [Freitas et al., 2006]

El ciclo de vida (figura 1.4) se inicia cuando un insecto infectado pica a un ser humano y defeca. Los trypomastigotas metacíclicos se transmiten en las heces y entran al huésped a través de la herida o por las mucosas. Cuando ingresan en una célula humana se convierten en amastigotas y se reproducen por mitosis. Los amastigotas se convierten nuevamente en trypomastigotas y la célula se rompe permitiendo que los trypomastigotas viajen por la sangre e infecten otras células. [web: página del Center for Disease Control and Preventon -CDC-]

Cuando el insecto pica a un huésped infectado, algunos trypomastigotas pasan a él a través de la sangre y se convierten en epimastigotas en el intestino del insecto donde se reproducen por mitosis. Luego los epimastigotas pasan al

recto donde se diferencian en la forma que tiene capacidad infectiva: los trypomastigotas metacíclicos, los cuales son evacuados con las heces pudiendo infectar un nuevo huésped. [web: página del Center for Disease Control and Preventon -CDC-]

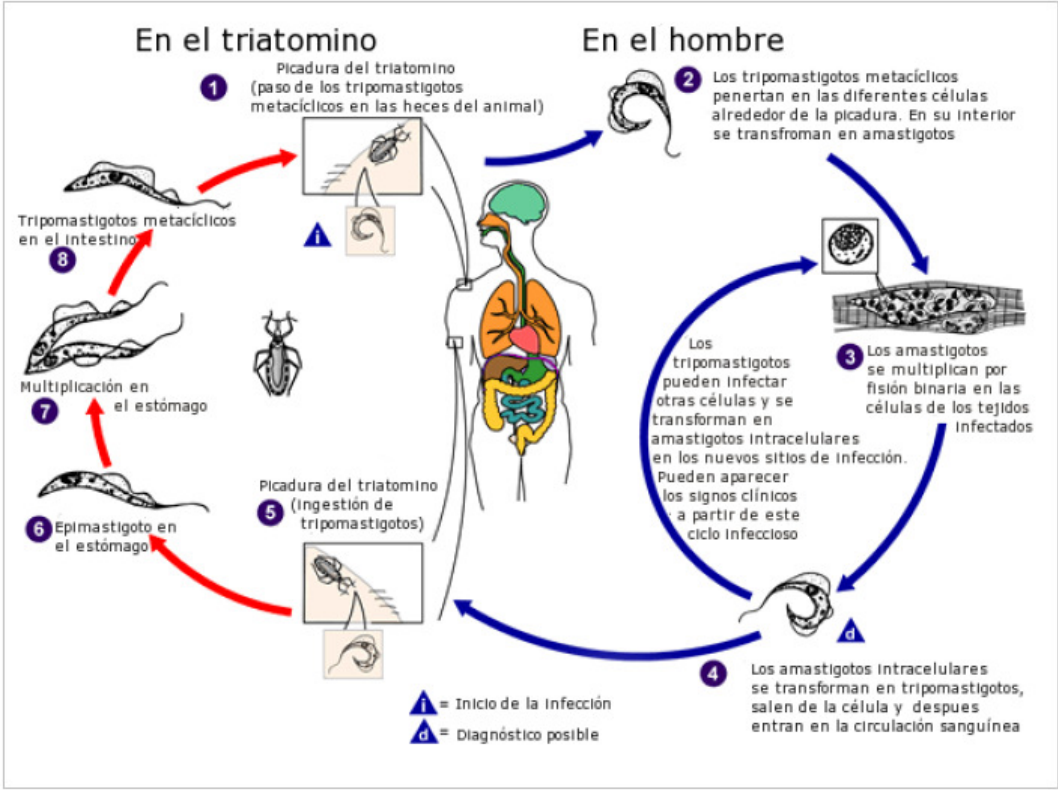


Figura 1.4: ciclo de vida de *T. cruzi*. Imagen adaptada de http://www.cdc.gov/parasites/chagas/biology.html

La enfermedad de Chagas presenta tres formas de contagio. [Jawetz et al., 1992]

- La más extendida se produce por la deposición del insecto vector (la vinchuca) cerca de la herida (peripárpados) que éste produce al alimentarse de una persona y el posterior rascado que lleva las heces hacia la mucosa ocular.

- Por la deposición del insecto que (por ejemplo en el techo de una vivienda) depone, cayendo sus heces sobre mucosas (por ejemplo ojos) del individuo.

- Por ingesta accidental del insecto y pasaje a través de la mucosa gástrica (remanentes de insectos portados en alimentos).

Para los dos primeros casos, en el sitio de entrada del parásito puede formarse un nódulo subcutáneo inflamado (chagoma). La enfermedad de Chagas es frecuente en los lactantes. Especialmente en los niños, la hinchazón unilateral de los párpados es un signo característico del comienzo de la enfermedad. La lesión primaria se acompaña de fiebre, linfadenitis aguda regional y diseminación a la sangre y los tejidos. Los parásitos pueden ser observados luego de una o dos semanas como trypomastigotas en sangre. El desarrollo ulterior depende de los órganos y tejidos más afectados y de la naturaleza de la multiplicación y liberación de las toxinas. Los órganos más gravemente afectados son el sistema nervioso central y el músculo cardíaco. La miocarditis intersticial es el elemento grave más común. Otros órganos afectados son hígado, bazo y médula ósea. La invasión o destrucción tóxica de plexos nerviosos en las paredes de vías digestivas conduce a megaesófago y megacolon. [Jawetz et al., 1992]

La enfermedad de Chagas tiene una fase aguda y otra crónica. Si no se le trata, la infección dura toda la vida. La fase aguda de la enfermedad ocurre inmediatamente después de la infección, puede durar hasta varias semanas o meses y se pueden encontrar los parásitos en la sangre circulante. La infección puede ser leve o asintomática. Puede haber fiebre o inflamación alrededor del sitio de inoculación. La fase crónica es una etapa que es prolongada y en algunos casos asintomática. Durante esta etapa prácticamente no se encuentran parásitos en la sangre. De hecho muchas personas pueden no presentar síntomas durante toda la vida. Sin embargo, se calcula que entre un 20% y un 30% de las personas infectadas presentarán problemas médicos debilitantes y a veces potencialmente mortales a lo largo de la vida. [web: página del Center for Disease Control and Preventon -CDC-]

En cuanto al tratamiento de la enfermedad de Chagas, existen dos drogas. Ambas inducen ocasionalmente serios efectos secundarios y ninguna tiene una alta eficacia contra la etapa crónica de la enfermedad, lo que indica la necesidad de nuevas drogas. [Barret et al., 2003]

Objetivos

1. Objetivos formativos

Familiarizarse con las tecnologías de secuenciación masiva de segunda generación y con el análisis de los datos generados. Se requiere adquirir entrenamiento y formación en técnicas de trabajo en bioinformática para dar respuesta a un problema biológico determinado.

Instruirse en el uso de herramientas para llevar a cabo el análisis bioinformático de datos de secuenciamiento profundo. Ello implica la elección de las herramientas más adecuadas para el análisis de un problema en particular, su utilización de forma correcta y aprender a analizar su salida.

2. Objetivos científicos

Medir el nivel de expresión de los genes en tres estadios del ciclo de vida de *T. cruzi* mediante la realización de análisis bioinformáticos sobre los conjuntos de datos correspondientes a los estadios epimastigota (insecto), amastigota (mamífero) y trypomastigota (mamífero).

Realizar análisis estadísticos sobre los niveles y patrones de expresión. Abordaje a través de varias aproximaciones para determinar la expresión diferencial de los genes y brindar soporte estadístico a la clasificación.

Estudiar la posibilidad de que la expresión en la hebra no codificante incida en la expresión de los genes. Análisis del matcheo de *reads* sobre la hebra no codificante mediante herramientas de mapeo y herramientas gráficas de exploración del genoma en busca de patrones que estén relacionados en la expresión de los genes.

Materiales y métodos

1. Técnicas de secuenciación masiva

Existen diferentes opciones para llevar a cabo una secuenciación. Dependiendo de la tecnología empleada se obtiene mayor o menor cantidad de datos y archivos de salida con distinto formato. A continuación se hará una breve reseña de las técnicas de secuenciación masiva, enfatizando en la que fue empleada para este trabajo.

El método automático de Sanger (basado en la síntesis y posterior separación de los segmentos sintetizados mediante electroforesis en capilares) fue el único método de secuenciación disponible por casi dos décadas correspondientes al desarrollo de la genómica en gran escala. Durante este período se produjeron diversas mejoras técnicas, especialmente en el sentido de lograr un cierto grado de paralelización. Los secuenciadores capilares (basados en el método de Sanger) logran procesar hasta 96 muestras simultáneamente. Sin embargo la secuenciación por el método de Sanger presentaba muchas limitaciones, sobre todo si se trataba de genomas grandes. Los esfuerzos más recientes han apuntado a la generación de nuevos métodos de secuenciación.

El método de Sanger es considerado como una tecnología de primera generación. Métodos más recientes han sido clasificados como NGS (*Next Generation Sequencing*), la segunda generación de estas técnicas. Los más recientes avances, muchos de los cuales están en etapa de prueba y no se comercializan, son considerados como la tercera generación. [Metzker, 2010]

Las técnicas NGS constituyen un salto cualitativo respecto a los métodos utilizados hasta entontes. Esto es debido al paralelismo masivo, o sea, a la posibilidad de tener millones de procesos corriendo simultáneamente. A su vez, esto permitió bajar los costos de secuenciación de forma sustancial.

En cuanto al estudio de la expresión de los genes, los *microarrays* están siendo reemplazados por los métodos de secuenciación de bases, que permiten identificar y cuantificar transcriptos raros sin requerir conocimiento previo de un gen en particular y pueden proveer información con respecto al *splicing* alternativo y variaciones en la secuencia de un gen determinado. [Metzker, 2010]

Los diferentes protocolos distinguen una tecnología de la otra y determinan el tipo de dato producido por cada plataforma. Estas diferencias en los datos de salida presentan un desafío cuando se pretende comparar las plataformas basándose en la calidad y los costos. Aunque los valores de calidad y las estimaciones de precisión son previstas por cada fabricante, no existe consenso en que la “calidad de base” de una plataforma sea equivalente a la de otra. [Metzker, 2010]

1.1. Generalidades de las tecnologías NGS

Estos métodos generalmente implican el corte aleatorio del ADN genómico en pequeños fragmentos a partir de los cuales se crean los segmentos de nucleótidos (*templates*) que serán secuenciados y leídos dando lugar a los *reads* (un *read* es la secuencia que se pudo determinar de un fragmento). Algo que tienen en común las tecnologías NGS es que el *template* es inmovilizado sobre una superficie sólida. La inmovilización de *templates* en sitios espacialmente separados, junto a la posibilidad de realizar un *tracking* de cada uno de ellos permite que cientos de miles o incluso millones de reacciones de secuenciación se lleven a cabo simultáneamente. [Metzker, 2010]

Las distintas empresas que comercializan secuenciadores, implementaron métodos diferentes para la lectura de las bases que forman el *template*. Como salida del proceso de secuenciación se obtienen archivos con la información de los *reads* y sus calidades. [Metzker, 2010]

1.2. Preparación de la librería

Preparar una librería implica la fragmentación del ADN y la colocación de adaptadores a los extremos de los fragmentos. Los adaptadores, que son secuencias conocidas, se usan para fijar los fragmentos al sustrato por complementariedad de bases con oligos previamente adheridos al sustrato.

Si el material genético del que se parte es ARN, entonces se copia a ADN mediante una polimerasa reversa. Ese ADN se seguirá copiando por complementariedad y se obtendrá ADN doble hebra. En este caso es imposible saber a qué hebra de ADN corresponde el material originalmente fragmentado. Pero es posible tener una librería hebra específica, esto significa que se puede identificar la hebra de ADN que corresponde al ARN inicial. Esto se lleva a cabo usando adaptadores diferentes en los extremos 3' y 5' del ADN copia. Que la librería sea hebra específica, resulta de gran importancia en esta tesis ya que, como parte del trabajo, se analizan patrones de matcheo de *reads* sobre la hebra no codificante siendo imprescindible la diferenciación de esta hebra.

1.3. Amplificación

***Templates* amplificados por clones (*Clonally amplified templates*).** La mayoría de los sistemas de análisis de imágenes no tienen la capacidad para detectar eventos fluorescentes aislados (de moléculas individuales), por lo que es necesaria la amplificación de los *templates*. Para realizar esta amplificación los dos métodos son emulsión PCR (emPCR) y la amplificación de fase sólida. [Metzker, 2010]

En emPCR se crea una librería de fragmentos y se ligan adaptadores conteniendo sitios de *priming* universales a los extremos de los fragmentos, lo que permite amplificar genomas complejos con *primers* de PCR comunes. Luego de la ligación, el ADN es separado en simple hebra y capturado en perlas bajo condiciones que favorecen que un único fragmento de ADN se pegue a cada perla. Esta captura se realiza mediante hibridación de los *primers* universales con oligonucleótidos unidos a las perlas (forman un césped sobre la misma) que presentan la secuencia complementaria al *primer*. Un aspecto clave es que cada perla individual debería capturar un solo fragmento, lo cual asegurará la amplificación de ese fragmento de ADN. Esto se logra utilizando una dilución límite de los fragmentos (mucha mayor concentración de perlas que de fragmentos). De esta forma la mayoría se las perlas no se unirán a ningún fragmento, una proporción mucho más baja se unirá sólo a un fragmento y una cantidad prácticamente insignificante se unirá a dos o más fragmentos. Luego de la amplificación, millones de perlas pueden ser

inmovilizadas. La forma en que se inmovilizan las perlas varía según el fabricante pero las opciones son: inmovilización en gel de poliacrilamida sobre un porta objetos de microscopio (Polonator), inmovilización química sobre una superficie de vidrio (Life/APG, Polonator) o inmovilización por deposición de las perlas en pequeños orificios en una placa (Roche/454). Ver figura 3.1. [Metzker, 2010]

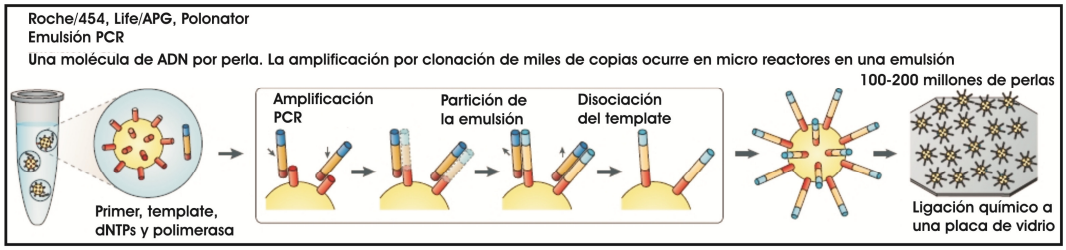


Figura 3.1: en emPCR es creada una emulsión aceite-acuosa para encapsular las perlas con el ADN pegado, en gotas acuosas. La reacción de PCR se lleva a cabo dentro de las gotas de forma de obtener miles de copias de la misma secuencia (*template*). Las perlas pueden ser fijadas a una superficie de vidrio o colocadas en los orificios de una placa. Imagen adaptada de Metzker, 2010

La amplificación de fase sólida también puede ser utilizada para producir una distribución aleatoria de *clusters* de fragmentos clonados en una superficie de vidrio. Una alta densidad de *primers forward* y *reverse* se pegan a la superficie. Este tipo de amplificación puede producir entre 100 y 200 millones de *clusters* de fragmentos espacialmente separados (Illumina/Solexa). En las terminales libres puede hibridar un *primer* universal y así iniciar la reacción NGS. Ver figura 3.2. [Metzker, 2010]

Además de los métodos de amplificación por “micro-clonación” arriba descritos, se vienen desarrollando varios métodos de molécula única que ofrecen ciertas ventajas tales como la menor cantidad de ADN para preparar las muestras, la no introducción de mutaciones durante la fase de PCR y el gran tamaño de los *reads* que facilitan enormemente del ensamblaje. Una descripción detallada de los mismos cae fuera de los propósitos de la presente tesis.

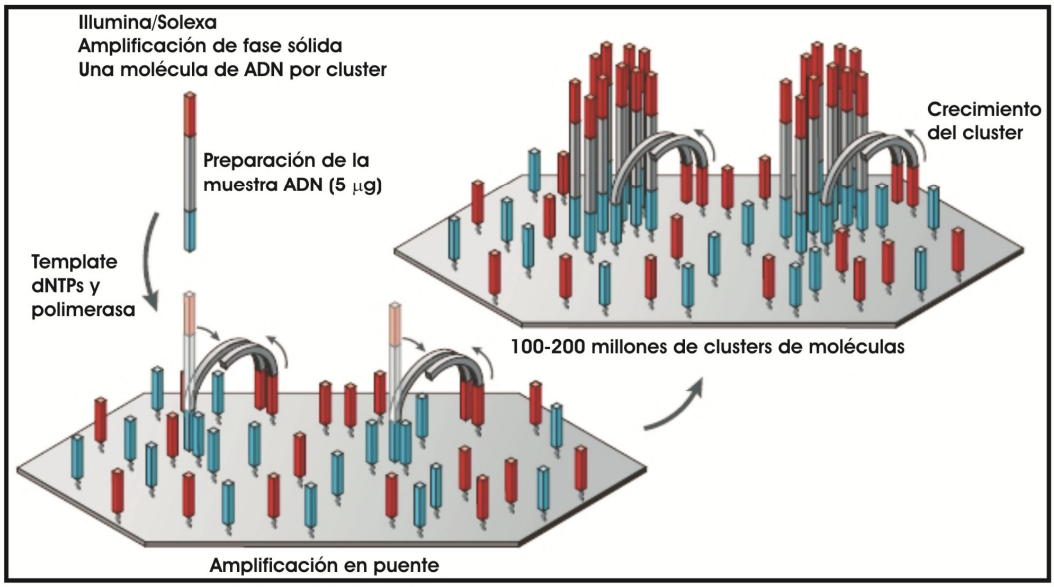


Figura 3.2: la amplificación de fase sólida está compuesta de dos etapas: adición de *primers* y extensión de la hebra simple, *template* de molécula única, y amplificación en puente de los *templates* inmovilizados con *primers* adyacentes para formar *clusters*. Imagen adaptada de Metzker, 2010

1.4. Secuenciación y tratamiento de imágenes

Hay grandes diferencias entre la secuenciación por amplificación de clones y los *templates* de molécula única. La amplificación de clones resulta en una población de *templates* idénticos, cada uno de los cuales ha sido objeto de una reacción de amplificación. En cuanto a las imágenes, la señal observada es un consenso de los nucleótidos adicionados en *templates* idénticos en un ciclo dado. Si un *template* está incompleto dará lugar a un desfasaje. La adición de múltiples nucleótidos en un ciclo también lleva a desfasaje. El desfasaje de la señal incrementa el ruido de fluorescencia causando errores en la asignación de bases y *reads* cortos. El desfasaje no se produce en *templates* de molécula única, no obstante, este método es susceptible a la adición de múltiples nucleótidos en un ciclo dado. [Metzker, 2010]

Terminación reversible cíclica (TRC). La secuenciación se realiza mediante la polimerización de la molécula, la cual se detiene luego de que cada nucleótido es incorporado. La determinación del nucleótido en el sitio donde se detuvo la polimerización generalmente se realiza mediante fluorocromos. Para detener la polimerización luego de la incorporación de cada base se usan los métodos de terminación reversible. Estos consisten en la incorporación de terminadores reversibles que comprenden la incorporación de nucleótidos,

análisis de imágenes con fluorescencia y escisión; y posterior desbloqueo para permitir la continuación de la incorporación de un nuevo nucleótido. En la primera etapa, una ADN polimerasa, ligada al *template* con *primer*, incorpora un solo nucleótido. Este nucleótido está modificado de forma que es capaz de emitir una señal fluorescente que será captada en una imagen. La síntesis de ADN termina luego de la incorporación del nucleótido. Los nucleótidos que no fueron incorporados son quitados mediante lavado. La etapa siguiente, de escisión, elimina el terminador; que fue lo que inhibió la incorporación de más de un nucleótido. Se hace otro lavado previo a iniciar la siguiente etapa de incorporación. Illumina/Solexa emplea un ciclo TRC de cuatro colores, lo que permite agregar los cuatro nucleótidos (identificados debidamente) en un solo ciclo, mientras que Helicos/BioSciences tiene un ciclo TRC de un color, esto implica que un solo tipo de nucleótido es incorporado por vez. Ver figura 3.3 para el caso de Illumina/Solexa. [Metzker, 2010]

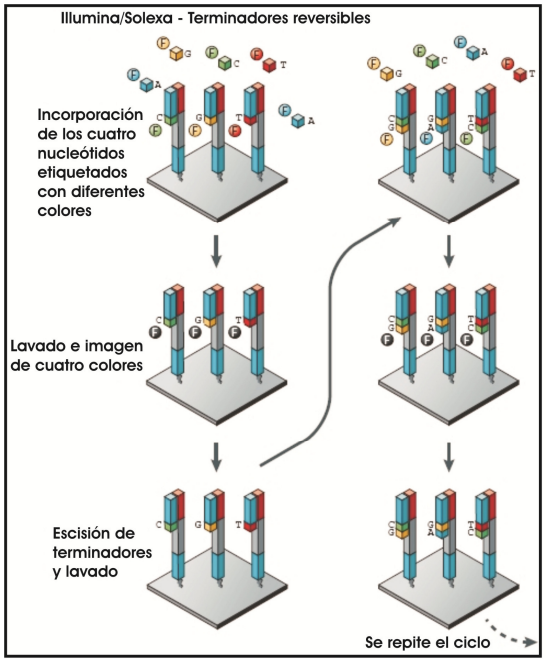


Figura 3.3: Método TRC. Ciclo de cuatro colores utilizado por Illumina/Solexa. Imagen adaptada de Metzker, 2010

La secuencia que se encuentra en una posición dada es leída por ciclos. Ver figura 3.4, correspondiente a la tecnología de Illumina/Solexa. En la posición superior se puede ver el siguiente patrón de colores: verde – azul – rojo – verde – amarillo – rojo. Cada color representa una base según el código de colores que aparece al pie de la figura, dando como resultado la secuencia: CATCGT. Análogamente se obtiene la secuencia de abajo: CCCCCC. [Metzker, 2010]

Figura 3.4: lectura de las secuencias a partir de las imágenes.
Tecnología Illumina/Solexa.

Actualmente Illumina/Solexa es la tecnología más usada para NGS. Usa el método de *templates* amplificados por clones (figura 3.2), acoplado con el ciclo TRC de cuatro colores (figuras 3.3 y 3.4). Una corrida del secuenciador utiliza un *slide* con 8 canales, lo que permite que experimentos diferentes sean secuenciados a la vez. El tipo de error más común son las sustituciones, con una mayor ocurrencia de errores cuando la base previamente incorporada es una G. El analizador de genoma de Illumina/Solexa presenta una subrepresentación de regiones ricas en AT y GC, lo que probablemente se debe a un sesgo en la amplificación durante la preparación de los *templates*. [Metzker, 2010]

Una forma alternativa a la reversible es la pirosecuenciación. Al igual que otros métodos que emplean nucleótidos modificados para determinar la síntesis de ADN, la pirosecuenciación manipula la ADN polimerasa por la adición de dNTP[1] (un tipo por ciclo). Se agregan perlas más pequeñas conteniendo sulfurilasa y luciferasa para facilitar la producción de la señal luminosa. Se agregan luego los dNTPs en un orden determinado y la señal luminosa es captada por una cámara. Por ejemplo, para el caso de la figura 3.5, se agregan C a los pocillos y si en el fragmento existe una G en la posición donde la polimerasa puede incorporar un nucleótido, entonces se da la complementariedad, se establece el enlace químico y es posible medir la señal luminosa que es emitida. La intensidad del pico de la señal es medida y graficada de forma de determinar la cantidad de bases que se incorporan a la vez (ver figura 3.6). [Metzker, 2010]

El secuenciador comercializado por Roche/454 ha tenido algunas mejoras permitiendo obtener *reads* más largos y disminuyendo la cantidad de cruces entre pocillos adyacentes. [Metzker, 2010]

[1] Dioxinuleótido trifosfato

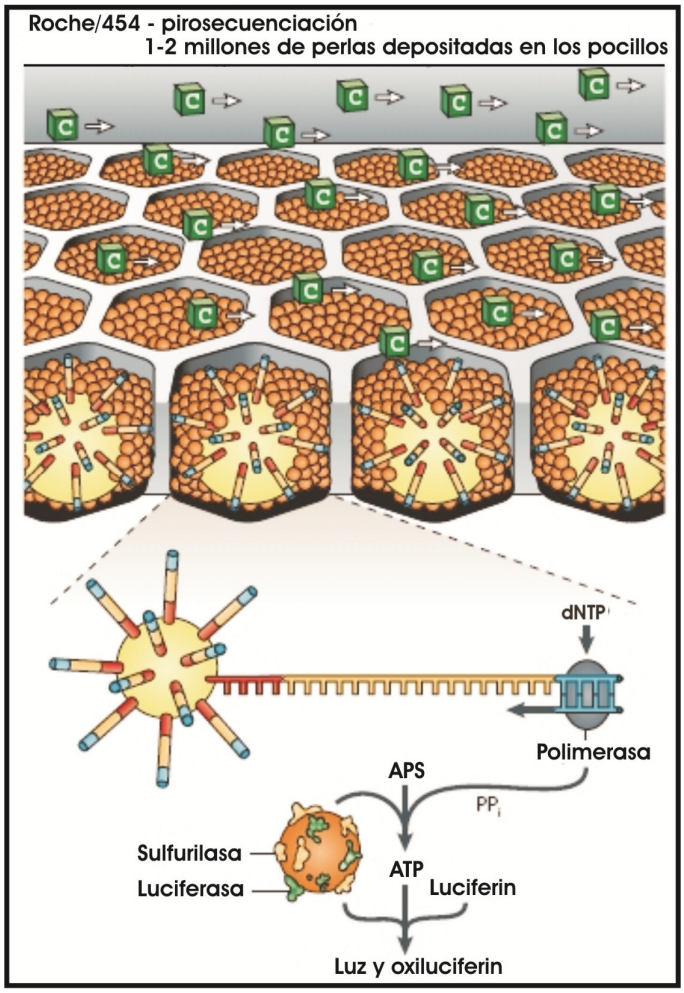


Figura 3.5: pirosecuenciación usando la plataforma de Roche/454 Titanium. Luego que las perlas se depositan en los pocillos, son agregadas otras perlas más pequeñas conteniendo sulfurilasa y luciferasa. Imagen adaptada de Metzker, 2010

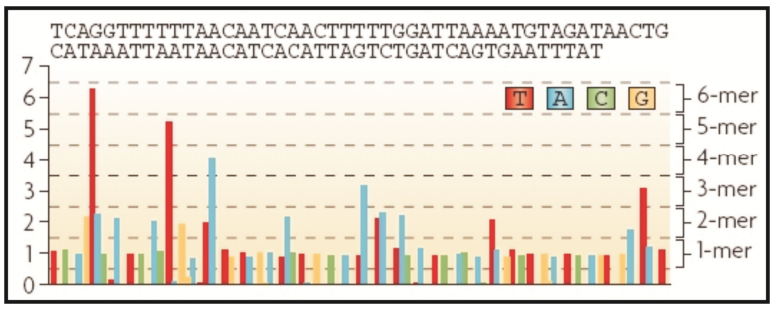


Figura 3.6: gráfica de la intensidad con que es detectada la señal luminosa luego de agregar los dNTPs de un tipo. Si dos bases hibridan (una a continuación de la otra), la señal será de aproximadamente el doble de intensidad que si hibridara una sola base. Este es el caso para la cuarta base de la figura (G), donde el pico amarrillo que le corresponde es el doble de alto que los picos previos (correspondientes a las bases T, C y A). Esto implica que hasta esa incorporación de dNTP, se tiene la secuencia: TCAGG. Imagen adaptada de Metzker, 2010

1.5. La tecnología de secuenciación utilizada

La elección de la tecnología a emplear depende muchas veces del problema biológico a analizar. Los datos utilizados en esta tesis fueron obtenidos por el secuenciador Genome Analyzer IIx comercializado por Illumina/Solexa (cuya producción está actualmente discontinuada). Cabe entonces profundizar en el proceso de secuenciación de esta tecnología dado que fue la empleada para obtener los datos que se analizan en este trabajo.

Existen tres etapas que se llevan a cabo en el secuenciador, a saber: preparación de la librería, generación de *clusters* y secuenciación.

Preparación de la librería. El instrumento realiza la secuenciación paralela y masiva de millones de fragmentos de ADN. Primero se fragmenta el ADN, se adenila y se agregan oligos como adaptadores a ambos extremos del fragmento (figura 3.7). Esos fragmentos se seleccionan por su tamaño.

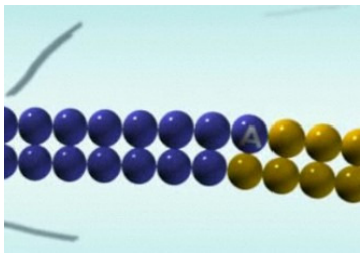


Figura 3.7: el *template* (en azul) adenilado (A) se une al oligo (en amarillo). Imagen adaptada de http://www.youtube.com/watch?v=l99aKKHcxC4&feature=related

Generación de *clusters*. Se realiza sobre superficie sólida. En esta etapa las moléculas individuales son amplificadas previamente a la secuenciación. Los ocho canales de la placa tienen una gran densidad de *oligos* adheridos a su superficie. Estos *oligos* se unen a los adaptadores de los fragmentos que se prepararon en la etapa previa, pues son complementarios (figura 3.8).

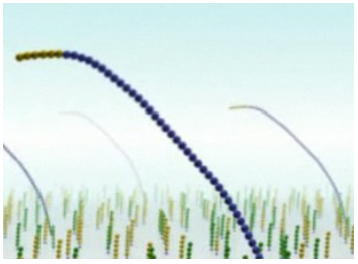


Figura 3.8: los fragmentos con adaptadores se unen a los oligos que están en la placa. Imagen adaptada de http://www.youtube.com/watch?v=l99aKKHcxC4&feature=related

Moléculas de una sola hebra de ADN hibridan con los *oligos* y comienzan las copias. Estas cadenas se doblan de forma que su extremo libre se une a algún *oligo* de la placa formando puentes. El proceso de copia vuelve a empezar.

Este ciclo se repite hasta obtener *clusters* de moléculas idénticas dentro de un mismo *cluster* pero distintas entre diferentes *clusters* (figura 3.9). La hebra complementaria es retirada y los *clusters* están listos para secuenciar.

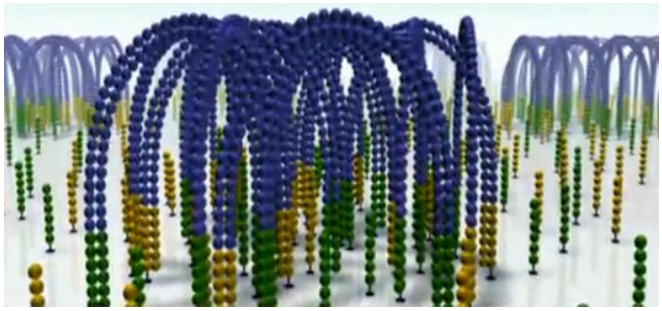


Figura 3.9: amplificación en forma de puente. En primer plano se ve un *cluster* donde todas sus moléculas son iguales, pero distintas a las moléculas que forman otros *clusters*, que se pueden apreciar en segundo plano. Imagen adaptada de http://www.youtube.com/watch?v = l99aKKHcxC4&feature=related

Secuenciación. La secuenciación se realiza por el método de terminación reversible y fluorocromos como se señalo anteriormente. Los *templates* son secuenciados base a base usando cuatro nucleótidos marcados con fluorescencia. Las bases de los cuatro tipos compiten para unirse al *template* pero solo se unirá la complementaria a la base en la hebra que se está copiando (figura 3.10). Los *clusters* son excitados por luz láser, lo que hace que emitan una señal poniendo en evidencia la base de que se trata (figura 3.11). El marcador fluorescente es removido, permitiendo que comience un nuevo ciclo.

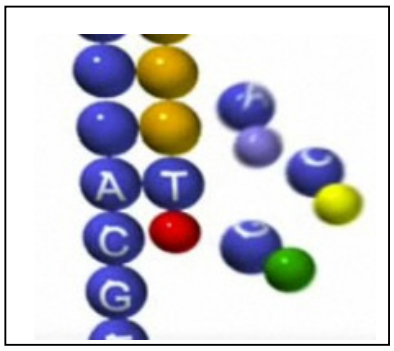


Figura 3.10: hibridación. Los cuatro tipos de nucleótidos compiten por hibridar pero solo lo hará una timina, dado que la base complementaria es una adenina. Imagen adaptada de http://www.youtube.com/watch?v=l99aKKHcxC4&feature=related

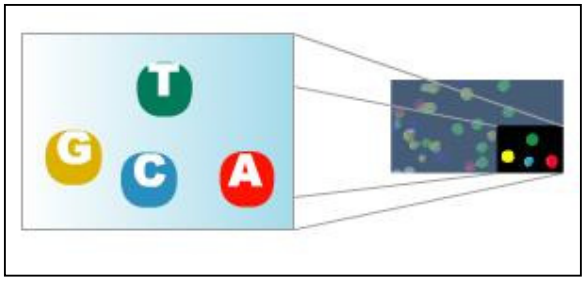


Figura 3.11: lectura de la señal luminosa. Cada *cluster* emite una señal según la base que hibridó. Dependiendo del color de la señal es posible identificar la base. Imagen adaptada de http://www.youtube.com/ watch?v=l99aKKHcxC4&feature=related

1.5. Consideraciones sobre los datos de secuenciación

Como se explicó en la sección previa, la identificación del nucleótido del *template* que es leído se hace a través del reconocimiento de un destello (de color o no) que se registra en una imagen. El volumen de datos correspondiente a estas imágenes, que es generado luego de la secuenciación, es enorme. Por eso las imágenes se procesan de modo que la información queda contenida en archivos de texto plano y las imágenes se eliminan. Aún así, la cantidad de información es tanta que los archivos suelen ser de varios gigas.

Manipular un archivo de ese tamaño implica disponer de mucha memoria RAM. Considerando además que cuando se trata de ensamblados contra un genoma conocido también se requiere trabajar con el archivo de referencia, es aún mayor la cantidad de memoria RAM necesaria para poder llevar a cabo un análisis. Algunos algoritmos han enfrentado este problema dividiendo el archivo en partes y aunando luego la información obtenida de cada análisis parcial.

La lectura de cada nucleótido de un *template* tiene asociada una calidad que indica la probabilidad de que ese nucleótido efectivamente sea el que se está reportando y no otro. La calidad disminuye cuanto más largo es el *template*. Pero este no es el único factor que afecta los datos. También puede suceder que se lea parte del *primer* usado en el experimento, esto dará como resultado la lectura de bases que quedan en el principio del *read* y que son las mismas para todos los *reads*. Otro factor lo constituyen los artefactos del experimento, por ejemplo si en la etapa del preparado de la librería alguna secuencia quedó sobre representada por algún motivo, llevará a un conteo falso cuando se analicen los datos. Uno de los motivos por los que se obtiene esa sobre representación es, para el caso de la amplificación en mini reactores, la existencia de más de una perla por mini reactor. Esto facilita que un *template* sea reproducido en mayor cantidad que si existiera una sola perla, ya que cuenta con el doble de sustrato donde replicarse. Otro motivo para la sobre representación está dado por el contenido GC de los *templates*. Un alto contenido de GC favorece la formación de enlaces más fuertes lo que puede introducir un sesgo en el resultado de la amplificación [Mignon, 2013]. Por estos motivos es fundamental hacer un análisis de los datos en cuanto a su calidad previo a cualquier manipulación. Es necesario descartar los *reads* cuya calidad resulte inapropiada, lo que dependerá de la tolerancia con la que se desea trabajar. En la sección 5 de este capítulo se describe la realización de un análisis de calidad de este tipo.

2. Mapeo

2.1. Técnica de mapeo

La técnica de mapeo consiste en alinear fragmentos pequeños de código (*reads*) con una referencia (para esta tesis, el genoma anotado de *T. cruzi*).

Figura 3.12: mapeo de dos *reads* sobre una referencia

La figura 3.12 muestra cómo dos *reads* (fragmentos pequeños de código) coinciden base a base con partes de una referencia. Los *reads* pueden solaparse en parte permitiendo reconstruir un fragmento mayor de la referencia (figura 3.13).

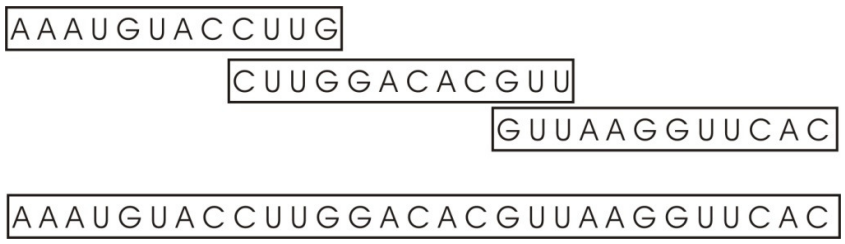


Figura 3.13: mapeo de *reads* que se solapan en parte

También es posible obtener muchos *reads* que mapean con una zona muy pequeña. Lo que genera una distribución como se muestra en la figura 3.14.

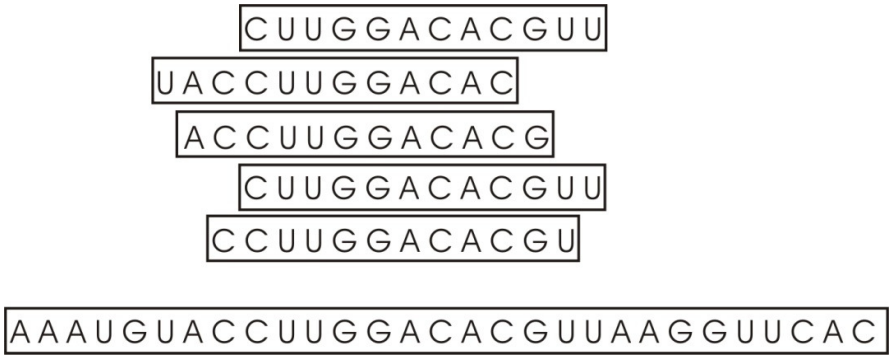


Figura 3.14: zona donde mapean muchos *reads* solapadamente. Este tipo de patrón fue encontrado en la hebra no codificante, en el extremo 5' de un tipo de proteína de membrana, ver más adelante.

2.2. Determinación del nivel de expresión de los genes

Si un gen se expresa, como resultado del proceso de secuenciación de ARN (ARNseq), se obtendrán *reads* que coinciden con la secuencia de bases de ese gen. Cuanto mayor sea el nivel de expresión de un gen, es esperable obtener más *reads* que correspondan a su código, o sea, que mapeen con su secuencia. De esta forma, el conteo de los *reads* que mapean con los genes, sirve para medir su nivel de expresión, pero hay algunas consideraciones a tener en cuenta.

Si un gen (G1) se expresa más que otro gen (G2), entonces es esperable que en el proceso de secuenciación se obtengan más fragmentos de código (*reads*) de G1 que de G2. A los efectos del ensamblado, se obtendrá una mayor cantidad de *reads* que mapean con G1 que los que lo hacen con G2. De esta manera, el conteo de *reads* es una medida de la expresión de los genes. Hay que tener en cuenta dos aspectos fundamentales cuando se hace este conteo. Por un lado, el largo de cada gen. Si G1 es 10 veces más largo que G2 y se trata de genes que se expresan en igual medida, es esperable encontrar un conteo 10 veces mayor de *reads* correspondientes a G1 que el correspondiente a G2, lo que en este caso no implica una sobreexpresión de G1 respecto a G2. Por este motivo es necesario corregir (normalizar) teniendo en cuenta el largo de los genes. Por otro lado hay que considerar las repeticiones, que para el caso de *T. cruzi* se trata de una característica nada despreciable. Si el gen G1 y el G2 son copias del mismo gen, puede darse que se esté expresando solo G1 pero el mapeo asignará *reads* a ambos, dando como resultado que, tanto G1 como G2, se expresan, cuando no es así realmente. Este es el motivo por el que se utilizó la herramienta ERANGE, que se describe más adelante. Existe un aspecto más a tener en cuenta cuando se trata de comparar expresión entre genes entre experimentos. O sea, El gen G1 corresponde al experimento 1 y G2 es el mismo gen pero corresponde al experimento 2. En este caso resulta evidente que el factor de corrección por el largo no aplica ya que se trata del mismo gen, aunque bajo distintas condiciones. Pero puede suceder que un experimento haya resultado más exitoso que el otro en términos de la cantidad de *reads* que se obtuvieron en total por cada experimento. Si por ejemplo, el experimento 1 arrojó el doble de *reads* que el experimento 2, puede darse la situación que se obtenga el doble de *reads* que alinean con el gen G1 que con el G2. Nuevamente, no se trata de sobreexpresión sino que el éxito del experimento 1 respecto al 2 lleva a este resultado para genes que se expresan de forma equivalente en ambas condiciones. Esta situación requiere que se realice otro tipo de corrección que se explicará más adelante.

3. Alineamiento y ensamblado del genoma

Los *reads* deben ser alineados contra un genoma de referencia o ensamblados *de novo.* Esto depende del problema biológico que se está analizando y de la disponibilidad de anotaciones sobre el organismo que se quiere analizar o de otros organismos emparentados.

Es posible alinear *reads* contra una secuencia conocida, por ejemplo un genoma previamente anotado (como en los casos de las figuras 3.12, 3.13 y 3.14). Otra posibilidad es alinearlos entre ellos formando grupos, tratando de formar cadenas de mayor largo (figura 3.15). En el primer caso se trata de una alineamiento contra una referencia y en el segundo, un ensamblado *de novo*.

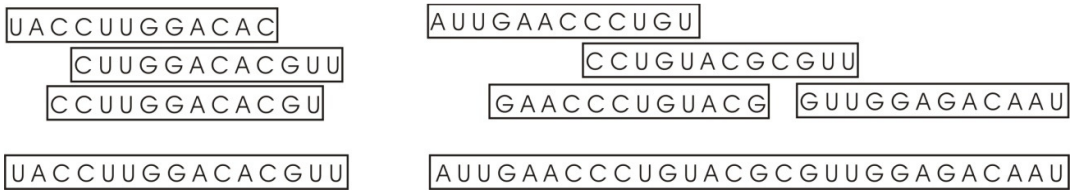


Figura 3.15: ensamblado *de novo*. Los *reads*, que se solapan en parte, pueden agruparse para formar secuencias más largas, que se muestran en la parte inferior de la figura

Cuando se ensambla el genoma *de novo* de un organismo, muchas veces, es posible hacer uso de la información de organismos relacionados para organizar los *reads* en *contigs* o asignarlos a genes, ya que el ordenamiento de genes del organismo nuevo puede ser parecido al de otro con el que está relacionado, es decir tenemos una conservación evolutiva de la sintenia. [Weatherly, 2009]

Alineamiento contra un genoma de referencia. Existen algunas limitaciones a la hora de hacer el alineamiento. Asignar los *reads* sobre regiones repetidas de la referencia resulta un problema complejo. Es posible que el *read* que se intenta alinear coincida con varias regiones en la referencia pero que en realidad haya provenido de una sola o algunas de ellas. También puede darse la existencia de *reads* que no coincidan en ninguna parte de la referencia. Esto es más frecuente cuanto más dudosa es la anotación del genoma que se utiliza como referencia.

Los *reads* de tipo *paired end* (o *mate-pair* si se trata de segmentos más largos) pueden contribuir en la solución del problema de las regiones repetidas. Estos *reads* se componen de dos segmentos que constituyen un par. De cada par se conoce la distancia a la que se encuentran por lo que al alinear deben coincidir los dos y hacerlo a la distancia indicada.

Ensamblado *de novo*. En los ensamblados de este tipo no se cuenta con un genoma de referencia por lo que los *reads* deben ser agrupados, dispuestos en *contigs,* sin contar con información previa y basándose en sus solapamientos. Para mejorar la calidad de estos ensamblados se requiere una gran cantidad de *reads*.

Es posible combinar *reads* provenientes de secuenciadores diferentes. Se ha combinado información de Roche/454 y de Illumina/Solexa para el ensamblado *de novo* en genomas de microbios logrando mejores resultados que los obtenidos con cada plataforma independientemente. [Metzker, 2010]

La combinación de datos de diferentes secuenciadores no parece trivial *a priori*. Se obtienen *reads* de diferente largo con calidades que no siguen la misma norma y los volúmenes de datos que se obtienen con distintas plataformas pueden variar grandemente. La cantidad de *reads* obtenida con las diferentes plataformas es importante a la hora de elegir la herramienta informática para el análisis. Los algoritmos se crean tomando en consideración los datos de entrada, no solo el tipo sino también el volumen de datos. [Metzker, 2010]

3.1. Herramientas de mapeo y algoritmia

En esta sección se describen las herramientas utilizadas en los mapeos llevados a cabo. Corresponde mencionar que existen muchas herramientas para el mapeo pero las que se incluyen a continuación son las que fueron utilizadas en esta tesis. Se detallan las etapas en el procesamiento algorítmico que estas herramientas hacen sobre los datos, lo que fundamenta su elección para realizar el mapeo. BLAST es uno de los alineadores utilizados y fue elegido por tratarse de una herramienta clásica y robusta que ha dado buenos resultados a lo largo de los años que lleva siendo empleado. Su estrategia de alineamiento se basa en el uso de hashs. Bowtie fue elegido por ser una de las implementaciones de la transformada de Burrows-Wheeler (técnica que permite búsquedas rápidas y que se desarrolla en las páginas siguientes) cuya utilización ha sido generalizada debido a la facilidad de uso y buena integración con varios wrappers[2]. Por último la elección en cuanto a la utilización de ERANGE estuvo dada por la dificultad del mapeo cuando el genoma presenta gran cantidad de repeticiones; tal es el caso de *T. cruzi*.

[2] Un wrapper es una pieza de código que permite que dos clases diferentes o incompatibles se entiendan.

BLAST

BLAST (acrónimo de ***B**asic **L**ocal **A**lignment **S**earch **T**ool*) es una herramienta para alinear localmente secuencias (ADN, ARN o proteínas). El algoritmo utilizado es heurístico, por lo que no está garantizado encontrar la mejor solución. Consta de tres etapas, a saber: *seeding* (compilación de una lista de palabras de alta puntuación y búsqueda de *hits* en la base de datos), extensión (extendido a ambos lados de los *hits*) y evaluación (cálculo de un parámetro que permite juzgar los resultados obtenidos). [Altschul et al., 1990]

> *Seeding* – consiste en la búsqueda de palabras cortas en las secuencias de la base de datos, que coincidan con fragmentos de la secuencia problema. Las palabras que se consideran significativas deben tener una puntuación mayor a T (que es un parámetro modificable) y estar a una distancia igual o menor que A de otra palabra (donde A también es un parámetro). Es posible ajustar otro parámetro (W) para considerar el tamaño de las palabras a buscar. Esta primera etapa puede ser llevada a cabo en un tiempo proporcional al largo de la lista de palabras.
>
> Extensión – esta etapa consiste en extender el alineamiento a los lados de cada palabra. La extensión se detiene cuando la puntuación del alineamiento desciende X o más puntos respecto al mayor valor obtenido para el alineamiento. Es la implementación de este punto que impide obtener la mejor solución existente ya que la detención de la extensión del alineamiento dado por X impide seguir extendiendo a lo largo de toda la secuencia; lo que, por otro lado, implicaría demasiado tiempo. Cabe destacar que en este punto se permite la inclusión de gaps en el alineamiento.
>
> Evaluación – cuando termina la extensión de todas las palabras, se evalúa cada alineamiento y se calcula su puntuación considerando la probabilidad que tiene ese alineamiento de haber sido obtenido por azar. La probabilidad que un alineamiento se de por azar depende del tamaño de la base de datos (para nuestro caso, el largo de la referencia contra la que se mapea) y el largo de las palabras (una vez extendidas). Se reportan los alineamientos cuyo valor esperado sea menor a E (que es un parámetro de corte conocido como *e-value*). Para alineamientos más significativos se elegirá un menor valor de E.

BOWTIE

Bowtie es otra herramienta para hacer alineamientos. Para esto indexa el genoma de referencia usando la transformada de Burrows-Wheeler (BWT) y el

índice FM[3]. El método más común de búsqueda en un índice FM es el algoritmo de coincidencia exacta de Ferragina y Manzini pero la coincidencia exacta no es apta ante la variabilidad genética o los errores de secuenciación [Ferragina y Manzini, 2000]. Por este motivo Langmead et al. propusieron dos modificaciones que permiten aplicar el algoritmo al alineamiento de *reads* cortos: un algoritmo de *backtracking* con énfasis en la calidad que permite *mismatches* y un algoritmo de indexación doble que evita el *backtracking* excesivo. [Langmead et al., 2009]

Antes de continuar con la técnica de Bowtie corresponde describir el funcionamiento de la BWT. La transformada se obtiene agregando al final de la secuencia a transformar (típicamente el genoma de referencia), un símbolo lexicográficamente menor a todos los otros y que no esté contenido en la secuencia (por ejemplo, $) y haciendo todas las rotaciones (ver figura 3.16). [Burrows y Wheeler, 1994]

1 BANANA$
2 ANANA$B
3 NANA$BA
4 ANA$BAN
5 NA$BANA
6 A$BANAN
7 $BANANA

Figura 3.16: a partir de la secuencia "BANANA" se obtienen todas sus rotaciones con el agregado del símbolo $. A la izquierda figura el número correspondiente al orden.

Se ordena esta matriz lexicográficamente y se mantiene el número correspondiente (ver figura 3.17).

7 $BANANA
6 A$BANAN
4 ANA$BAN
2 ANANA$B
1 BANANA$
5 NA$BANA
3 NANA$BA

Figura 3.17: matriz de Burrows-Wheeler

Los números, en el orden que aparecen en la matriz de Burrows-Wheeler, constituyen los índices {7,6,4,2,1,5,3} y la última columna es la transformada: BWT (BANANA) = ANNB$AA.

[3] Índice de Ferragina y Manzini, se describe más adelante.

Es fundamental que, a partir de la transformada, se pueda obtener la secuencia original, o sea, que el proceso sea reversible. Existe más de una forma de revertir la transformada; a continuación se detallará la más rápida. Se considera la primera y última columna de la matriz (ver figura 3.18) de forma que la i-ésima ocurrencia de un determinado carácter en una columna corresponde con la i-ésima ocurrencia de ese carácter en la otra columna. Por ejemplo, la segunda "A" de la primer columna se corresponde con la segunda "A" de la segunda columna, lo que se indica mediante una flecha. [Burrows y Wheeler, 1994]

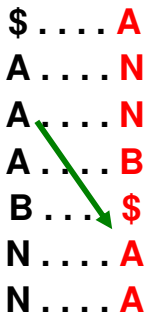


Figura 3.18: correspondencia de caracteres entre la primera y última columna de la matriz.

Para reconstruir la secuencia original se inicia, por ejemplo, con el primer carácter de la segunda columna que es la primera ocurrencia de "A" en esa columna y el siguiente carácter está dado por la primera columna, que es "$". Se busca ese carácter en la segunda columna y va a ser seguido del carácter que está en la primera columna a ese mismo nivel, o sea, "B" (que tiene una única ocurrencia en la secuencia). Se busca "B" en la segunda columna y en la primera corresponde una "A", que es la tercera ocurrencia de ese carácter en esa columna, por lo que debe buscarse la tercera ocurrencia de "A" en la segunda columna, que es la última. El algoritmo continúa de esta forma hasta retornar al carácter de inicio, lo que se puede apreciar en la figura 3.19. [Burrows y Wheeler, 1994]

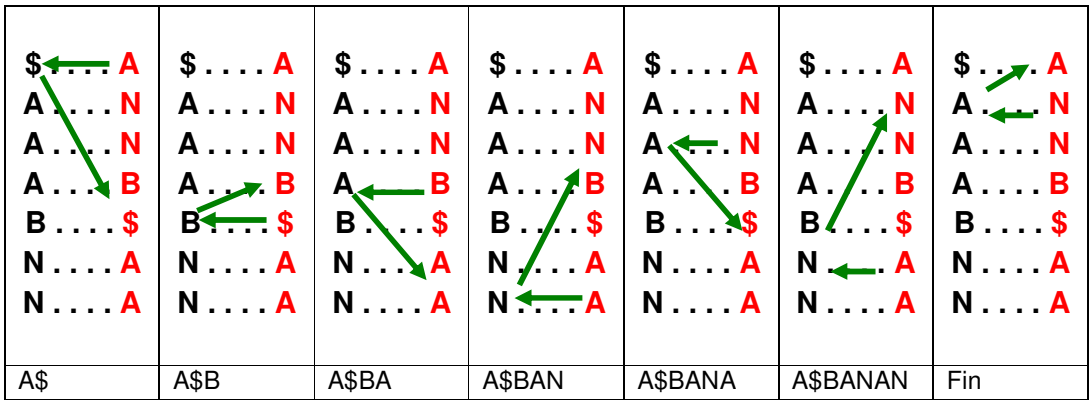


Figura 3.19: Se reconstruye la secuencia original hasta cerrar el ciclo y volver al carácter con que se inició.

Cabe destacar que es indiferente el carácter con que se comience, el ciclo terminará al volver a él. Luego se puede reconstruir la secuencia original comenzando con el símbolo "$", entonces se obtiene: $BANANA. El algoritmo de reconstrucción es lineal en el tiempo y en uso de memoria.

Resta explicar cómo se realiza la búsqueda de un patrón en la secuencia. Por ejemplo, para la secuencia dada (BANANA), buscaremos el patrón "ANA". Se parte de la matriz de la figura 3.17. L(W) es el menor índice k, tal que la k-ésima fila de la matriz empieza con W. U(W) es el mayor índice k, tal que la k-ésima fila de la matriz empieza con W. Aplicando estas definiciones sobre la matriz se tiene: L(ANA)=3 y U(ANA)=4. La diferencia entre L(W) y U(W) se denomina rango, para este ejemplo el rango es (3,4). Anteriormente se había definido el *array* de índices como {7,6,4,2,1,5,3}. Entonces, se debe buscar la posición 3 y 4 en el *array* de índices y obtener de allí las posiciones en la secuencia que contienen el sufijo "ANA". En la posición 3 se tiene un 4, esto quiere decir que existe el patrón "ANA" a partir de la posición 4 de la secuencia BAN**ANA** (aquí marcado en rojo). Análogamente, en la posición 4 del *array* de índices hay un 2, lo que significa que en la posición 2 de la secuencia B**ANA**NA se encontrará el patrón "ANA" (también marcado en rojo).

Debe notarse que la BWT permite la búsqueda de patrones exactos en una secuencia pero el alineamiento exacto de los *reads* en la referencia no es lo deseable ya que los alineamientos pueden contener *mismatches*, lo que puede deberse a errores de secuenciación o diferencias genuinas entre la referencia y los *reads*. Langmead et al. implementaron un algoritmo de alineamiento que implica una búsqueda con *backtracking*. Esto se representa en la figura 3.20, donde se compara una búsqueda exacta (arriba) con una búsqueda con *backtracking* (abajo) para la secuencia GGTA. Los números dentro de las cajas representan el rango de filas de la matriz que comienzan con el sufijo observado hasta ese punto. En el caso de la búsqueda exacta, cuando intenta alinear la segunda G, encuentra un rango nulo, lo que significa que no hay alineamiento total y allí termina la búsqueda. En el caso en que se permiten *mismatches*, al llegar a un rango nulo, se vuelve hacia atrás (*backtracking*) y se modifica una de las bases que matchearon, lo que introduce un *mismatch*. Esto sigue hasta encontrar el primer alineamiento que matchee (con el o los *gaps* que se introdujeron). Dado que no busca todos los alineamientos sino que se detiene al encontrar el primero, no hay garantía de que sea el mejor en términos de número de *mismatches* o de calidad. [Langmead et al., 2009]

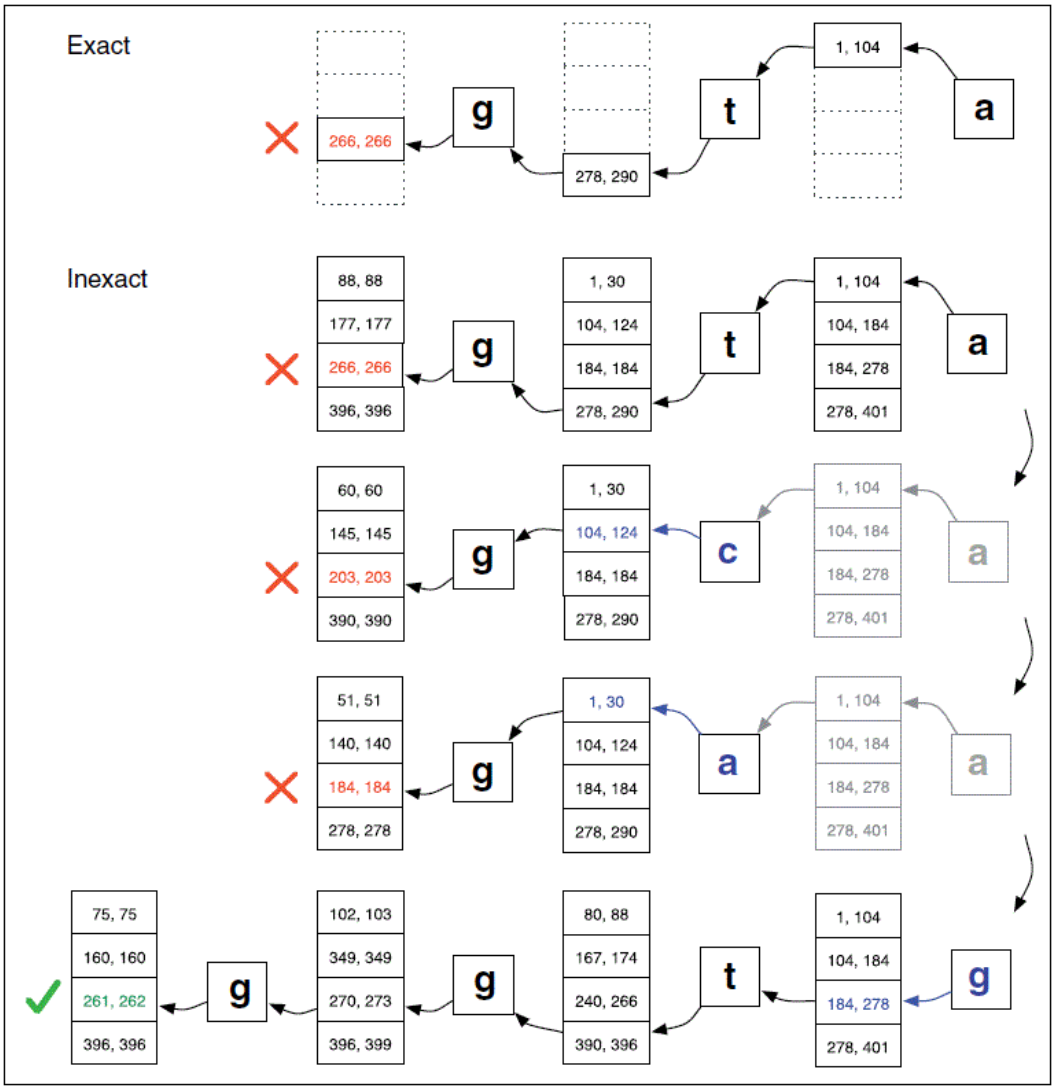


Figura 3.20: comparación entre un alineamiento exacto (arriba) y uno con *mismatches* (abajo). La cruz roja indica que se encontró un rango nulo y el algoritmo aborta para el caso de alineamiento exacto o vuelve hacia atrás para el caso que permite *gaps*. [Langmead et al., 2009]

Por la importancia que tienen los *multimatches* en el genoma de *T. cruzi*, al ser tan repetitivo, cabe destacar el tratamiento que hace Bowtie en estos casos. Es posible establecer diferentes escenarios para el tratamiento de los *multimatches* a través de parámetros. El parámetro -a reportará todos los *multimatches* sin límite en la cantidad, el parámetro -k N, reportará N *multimatches* que se eligen aleatoriamente del total de *multimatches* encontrados. En nuestro caso resultó de interés mantener todos los *multimathces* y resolver su correcta asignación mediante el uso de ERANGE, herramienta que se detalla más adelante.

En cuanto al *backtracking* excesivo, Bowtie usa una técnica de doble indexación. Se crean dos índices del genoma, uno conteniendo la BWT (llamado índice hacia delante) y otro con la BWT con la secuencia reversa (llamado índice espejo). Para ver cómo funciona esto considérese una política de alineamiento que permite un *mismatch*. Un alineamiento válido con un *mismatch* cae en uno de dos casos dependiendo qué mitad del *read* tiene el *mismatch*. Primero trae el índice hacia delante en memoria e invoca el alineador con la restricción de no hacer sustituciones en la mitad derecha del *read*. Luego usa el índice espejo e invoca el alineador sobre el *read* reverso, con la restricción de no hacer sustituciones en la mitad derecha del *read* (la original parte izquierda). Las restricciones sobre el *backtracking* en la mitad derecha previenen el excesivo *backtracking*. Langmead et al. observaron que se da excesivo *backtracking* cuando un *read* tiene varias bases con baja calidad; o cuando el alineamiento contra la referencia es pobre. Para evitar que este tipo de situaciones consuman mucho tiempo se limita la cantidad de veces que puede hacerse *backtracking* (por defecto a 125) [Langmead et al., 2009]

ERANGE (*Enhanced Read Analysis of Gene Expression*)

El paquete ERANGE calcula los niveles de expresión en RPKM a partir de un archivo de entrada que consiste en una referencia y lecturas de RNAseq. El paquete está compuesto por un conjunto de scripts en python que deben ser ejecutados en cierto orden. [Mortazavi et al., 2008]

El RPKM es una medida normalizada y extensamente usada al momento de representar el nivel de expresión de los genes. RPKM corresponde a la sigla en inglés de *number of **r**eads **p**er **k**ilobase per **m**illón reads mapped*. La ecuación que permite calcular el RPKM se incluye a continuación

$$RPKM = \frac{conteo.1000}{largo\ del\ gen}.\frac{1000000}{tamaño\ librería}$$

Por ejemplo, si un gen tiene un conteo de 450, o sea, 450 *reads* matchearon contra ese gen, el largo del gen es de 1200 bases y la librería cuenta con 10.500.000 *reads* (esto quiere decir que se obtuvieron 10.500.000 *reads* en total en ese experimento), entonces el RPKM resulta:

$$RPKM = \frac{450\ .1000}{1200}.\frac{1000000}{10500000} = 35{,}714$$

La idea que encierra el cálculo del RPKM es la de normalizar por el largo del gen a 1000 bases y por el éxito del experimento a 1000000 de *reads*. O sea, el conteo original de *reads* y el valor del RPKM coincidirán sólo si el largo del gen sobre el que matchearon los *reads* es de 1000 y la cantidad total de *reads* del experimento es de 1000000.

Mortazavi et al. describen las funciones de ERANGE de la siguiente manera:

- asignar *reads* que mapean de forma única en el genoma en el sitio de origen y asignar al lugar más probable *reads* que mapean igualmente bien en varios sitios

- detectar *reads splice-crossing*[4] y asignarlos a su gen de origen

- organizar *reads* que pertenecen a un mismo grupo pero que no mapean sobre ningún exón conocido, en exones candidatos o partes de exones

- calcular la prevalencia de transcriptos de cada ARN nuevo o conocido, basándose en conteos normalizados de *reads* únicos, *reads* que corresponden a *splicing* y *multimatches*. Las nuevas regiones ARN candidatas pueden ser provisionalmente agregadas a modelos de genes existentes si cumplen una serie de criterios adicionales. Los candidatos que permanezcan sin ser asignados pueden ser utilizados con otra información confirmada para desarrollar o revisar modelos de genes.

El principal interés en el uso de este software radica en el tratamiento de los *multimatches*. Es habitual el descarte de los *multimatches* al usar alguna de las herramientas informáticas disponibles, principalmente cuando se utilizan con los parámetros por defecto. Como consecuencia el conteo resulta subestimado. También es frecuente la asignación de los *multimatches* en todos los lugares donde se da coincidencia y de esta forma se sobreestima el conteo. En cualquiera de los dos escenarios el conteo no resulta fiel a la realidad. ERANGE evita esta situación distribuyendo los *multimatches* en proporción al número de matcheo de *reads* únicos y de *splice*. [Mortazavi et al., 2008]. La figura 3.20 ilustra la técnica que usa ERANGE para la asignación de los *multimatches*.

[4] Los *reads splice crossing* son aquellos que mapean en parte con una región de la referencia y la otra parte con otra región alejada de la primera. Eso implica la presencia de un intrón en la referencia que no estaba presente en la muestra de origen porque ésta corresponde a ARN.

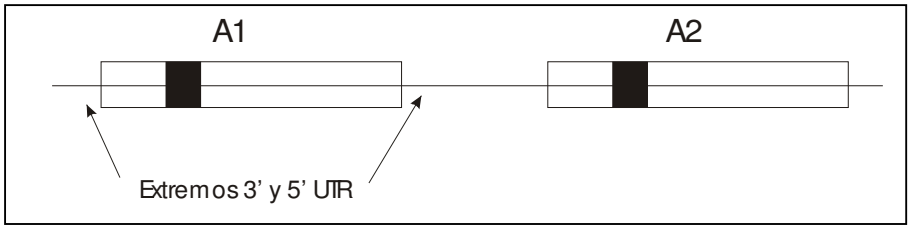


Figura 3.20: El gen A1 y el A2 son copias del mismo gen (o bien tienen la región marcada repetida, aunque no sean copias del mismo gen). ERANGE se fija en los extremos de los genes y cuenta las veces que aparecen para determinar que, por ejemplo, A1 aparece 10 veces y A2, 3. Si hay 26 *reads* que matchean con la zona marcada, 20 serán asignados al gen A1 y 6 al gen A2 (en proporción a la cantidad de veces que aparecen esos genes)

3.2. Los mapeos realizados

Mapeo 1:
Se hizo uso de la herramienta BLAST para este mapeo. Como referencia se utilizó un archivo conteniendo las secuencias de los genes anotados de *T. cruzi*.

Como resulta de interés considerar la zona cercana a los extremos de los genes a efectos de determinar correctamente la expresión, para el segundo mapeo se modificó la referencia.

Mapeo 2:
Nuevamente la herramienta utilizada fue BLAST. La referencia utilizada para este mapeo incluía, además de los genes, sus zonas marginales, tomando 300 bases por fuera de cada gen seguidas al extremo 5' y 200 seguidas al 3'.

Mapeo 3:
Se utilizó Bowtie para este mapeo y la referencia del mapeo 1.

Mapeo 4:
La herramienta usada fue Bowtie con la referencia del mapeo 2.

El formato de la salida de BLAST se ilustra en el registro siguiente:

SOLEXA-1GA-2_0093_FC62HVC:1:14:14292:3309#0/1 NonESM_Tc00.1047053508175.39
100.00 34 0 0 1 34 452 419 2e-12 67.9

Consta de doce columnas cuyo significado se describe a continuación:

Identificador de la secuencia buscada, identificador de la secuencia de referencia, porcentaje de identidad, largo del alineamiento, cantidad de no coincidencias, cantidad de apertura de gaps, inicio de la secuencia buscada, fin de la secuencia buscada, inicio de matcheo en la secuencia de referencia, fin de matcheo en la secuencia de referencia, valor esperado, HSP (par de mayor puntaje).

El formato de la salida de Bowtie se muestra a continuación.

```
SOLEXA-1GA-2_0093_FC62HVC:1:14:14292:3309#0/1  -  ESM_Tc00.1047053506825.140
418  TCATGGCATACCGCTGGAAGTACAAGTTGGGGAC  IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII  1
```

Las columnas de la salida de Bowtie significan lo siguiente:

Nombre del *read* que alinea, hebra (+/-), nombre de la secuencia de referencia, posición en la referencia donde comienza el alineamiento, secuencia del *read* mapeado, matcheo (|) o no matcheo ( ), lista de descriptores de las no coincidencias (separada por comas). Si el alineamiento coincide por completo, la lista está vacía. Los descriptores tienen la forma posición:base de la referencia>base del *read*.

Parseando los archivos de salida de los mapeos es posible identificar los genes sobre los que mapeó algún *read* y contabilizar la cantidad de *reads* mapeados sobre cada uno de los genes. Con esta información se confeccionó una tabla con los conteos de cantidad de *reads* que coincidieron con cada gen. Se normalizó por el total de *reads* obtenidos en cada experimento y por el largo. La normalización que involucra la cantidad de *reads* por experimento es necesaria para dar el mismo tratamiento a los datos que resulten de cada uno de ellos. Si, por ejemplo, el experimento correspondiente al estadio amastigota hubiera arrojado muchos más *reads* que los otros, entonces el conteo apuntará a una alta expresión de genes en ese estadio por sobre los otros, no siendo esto verdadero. La normalización por el largo se hace porque un gen más largo tiene más probabilidad de matcheo con *reads* que uno más corto, suponiendo similar nivel de expresión, esto llevaría a concluir erróneamente que el gen más largo se expresa más que el corto. La tabla confeccionada contiene algo más de 23.200 genes, donde la cantidad de genes que no se expresan en ningún estadio, según este conteo, es del orden del ciento.

4. Origen de los datos de referencia

Se utilizó como referencia el genoma de *Trypanosoma cruzi* disponible en la base de datos TriTrypdb. Se obtuvieron las secuencias correspondientes a los cromosomas y a las regiones codificantes (archivos en formato gff y fasta). La

referencia se divide en tres tipos de archivos de acuerdo al haplotipo de origen, los referentes a Esmeraldo, los no referentes a Esmeraldo y los de *contigs* no asignados.

El genoma de *T. cruzi* que se encuentra anotado y disponible no está completo. Esto se debe a que este parásito presenta algunas particularidades que hacen especialmente difícil su anotación de forma correcta y completa.

Weatherly et al. organizaron los *contigs* de *T. cruzi* en ensamblados del largo de cromosomas. Ellos señalan como dificultades la naturaleza repetitiva de *T. cruzi* y que la cepa de referencia (CL Brener) es un híbrido de dos linajes de *T. cruzi*. Obtuvieron como resultado 32.746 *contigs* parcialmente ensamblados en 638 *scaffolds* correspondientes a cromosomas que no están completos. Utilizaron los cromosomas de *T. brucei* como base para la organización inicial de los cromosomas de *T. cruzi* y los mapas de sintenia para asignar los *contigs* y *scaffolds* basándose en el orden de los genes de *T. cruzi* en el mapa de sintenia de cada cromosoma. Como los mapas de sintenia están basados en el orden de los genes en los cromosomas de *T. brucei*, los genes de *T. cruzi* que corresponden a un mismo *contig*, pueden mapear en los genes de *T. brucei* de diferentes cromosomas. Obtuvieron un genoma de 41 pares de cromosomas de largo entre 78 Kb y 2,4 Mb. Los cromosomas no fueron ensamblados en su totalidad, quedando sin ensamblar los extremos, por este motivo los largos de los cromosomas están subestimados. Coadyuva el hecho que diferentes regiones repetidas colapsan en un solo gen o región durante el ensamblado. Las regiones teloméricas y subteloméricas de los cromosomas no han sido secuenciadas completamente; estas regiones son muy redundantes para ser ensambladas de forma apropiada y muy variables como para ser informativas. [Weatherly, 2009]

Weatherly et al. validaron el ensamblado de forma visual (mediante un browser) y experimental (solapando clones BAC con genes específicos, cuya presencia en los clones era predicha por el propio ensamblado).

Más del 23 % de los genes anotados en el genoma son miembros de grandes familias multigénicas. Se ha sugerido la existencia de más de 20.000 genes adicionales en esas familias que no están presentes en el genoma dado el colapso de los *reads* durante el ensamblado. [Weatherly, 2009]

La complejidad que plantea el análisis de *T. cruzi* se basa en dos hechos fundamentales. Por un lado, la naturaleza repetitiva de su genoma y por otro lado, en que la referencia (*CL Brener*) es un híbrido de dos linajes de *T. cruzi*.

5. Tratamiento de los datos de secuenciación

Se secuenciaron tres estadios del ciclo de vida de *T. cruzi*. La librería preparada fue hebra específica lo cual permite determinar de cuál de las dos hebras de ADN proviene el ARN que se analiza, es decir este tipo de librería permite saber cuál es la hebra que se usa como molde en la transcripción.

El formato de los archivos originales es fastq. Este formato consta de cuatro líneas por *read*, como se muestra a continuación.

@SOLEXA-1GA-2_0093_FC62HVC:2:25:11219:14924#0/1
GATTGCCCCAAAAGCAAAAAGGAGGAATGGGGAGTA
+SOLEXA-1GA-2_0093_FC62HVC:2:25:11219:14924#0/1
ffcdff_fddegggaggfggfcacfccc[fdb`]Xb

La primera línea comienza con "@" y le sigue el identificador del *read*. En la segunda línea se tiene la cadena de nucleótidos que fue leída para ese *template*. La línea siguiente comienza con el carácter "+" seguida del identificador del *read* (que se repite igual que en la primera línea). La última línea contiene la codificación ascii de los valores de calidad para cada base, o sea, el número que se obtiene de restarle 33 al carácter ascii corresponde al valor numérico de calidad. "!" es el carácter que corresponde al menor valor de calidad y "~" al mayor.

El valor de calidad que es asignado a una base está relacionado con la certeza de la lectura de dicha base. Por ejemplo, un valor de calidad de 10 implica que la posibilidad de error es de 1 en 10, o sea, se comete un error de lectura por cada 10 bases que se leen. Para un valor de 20, la posibilidad de error es de 1 en 100 (ver tabla T.3.1). El valor de calidad (*Q*) se puede calcular con la siguiente ecuación, donde p es la probabilidad de error y $\log$ es el logaritmo neperiano:

$$Q = -10.\frac{\log(p)}{\log(10)}$$

Valor de calidad (Q)	Probabilidad de base errónea	Precisión
10	1 en 10	90%
20	1 en 100	99%
30	1 en 1.000	99,9%
40	1 en 10.000	99,99%
50	1 en 100.000	99,999%

Tabla T.3.1: valores de calidad y precisión en la determinación de bases

No sólo es importante la calidad de cada una de las bases sino también la calidad global de todo el *read*. Si la calidad del *read* no es aceptable, según los parámetros que se establezcan de aceptación, entonces ese *read* debe ser descartado. También puede suceder que la calidad de una o más bases en un *read* sean inaceptables, pero la calidad del *read* en general sí lo sea. En ese caso se descartan las bases de baja calidad pero se mantiene el *read*. Para esta situación, descartar las bases implica sustituirlas por N manteniendo sus posiciones, de otra forma se estaría recortando el *read* y alterando la secuencia. Por ejemplo si se tiene el *read* AAGCTAGCT, donde la quinta y octava base son de calidad inaceptable pero el *read* en su totalidad es de buena calidad. Se sustituirán las bases de baja calidad quedando el *read*: AAGCNAGNT, donde las bases sustituidas tienen la capacidad de matchear con cualquier base al momento de hacer el alineamiento; funcionando como comodines.

Utilizando la herramienta FastQC se obtiene información mediante la cual es posible evaluar la calidad de los *reads* [Andrews, 2010]. Esta herramienta requiere como entrada un archivo en formato fastq, por lo que se realiza un análisis por cada estadio del ciclo de vida de *T. cruzi*. La salida está compuesta de once ítems que se muestran como solapas en la interfaz de usuario de la aplicación. Todas contienen una representación gráfica de cada uno de los aspectos evaluados. A la izquierda se ilustra con una imagen de tilde o de una cruz los aspectos que fueron positiva o negativamente evaluados. [web: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/]. Es importante conocer las limitaciones o problemas que presentan los datos con los que se llevará a cabo el análisis. Cabe destacar el resultado obtenido en algunas de las solapas, las que se describen a continuación.

Una utilidad de esta aplicación es que analiza la calidad media de cada base por su posición a través de todos los *reads*. Por ejemplo, para la posición 1, se evalúa la calidad de la base que ocupa esa posición en cada uno de los *reads* y se representa gráficamente. Se hace lo mismo para cada una de las posiciones. Se puede apreciar que la calidad va decayendo hacia el final del *template*, no obstante la calidad en general es buena. Ver figura 3.21.

El motivo de la disminución de calidad hacia el final del *template* se explica por lo siguiente. Si en el proceso de agregado de nucleótidos durante la secuenciación, algunos *templates* se desfasan del resto de los *templates* del *cluster*, la intensidad de la señal en el momento de leer la base, será menor. Por desfasaje debe entenderse una situación como la que se ejemplifica a continuación: en un momento del proceso de secuenciación corresponde la incorporación de una adenina pero un *template* no la adiciona en ese momento y queda desfasado respecto al resto de *templates* de su *cluster*. La probabilidad que esta situación se vuelva a repetir en el mismo u otro *template*

de ese *cluster*, aumenta cuanto más largo sea el *template*. Supongamos que se tiene un 10% del total de *templates* de un *cluster* desfasados, entonces la señal será del entorno del 90% de lo que debería ser si no hubiera *templates* desfasados. Por este motivo se obtienen mejores valores de calidad al principio del *template* que al final y la calidad disminuye en proporción a su largo.

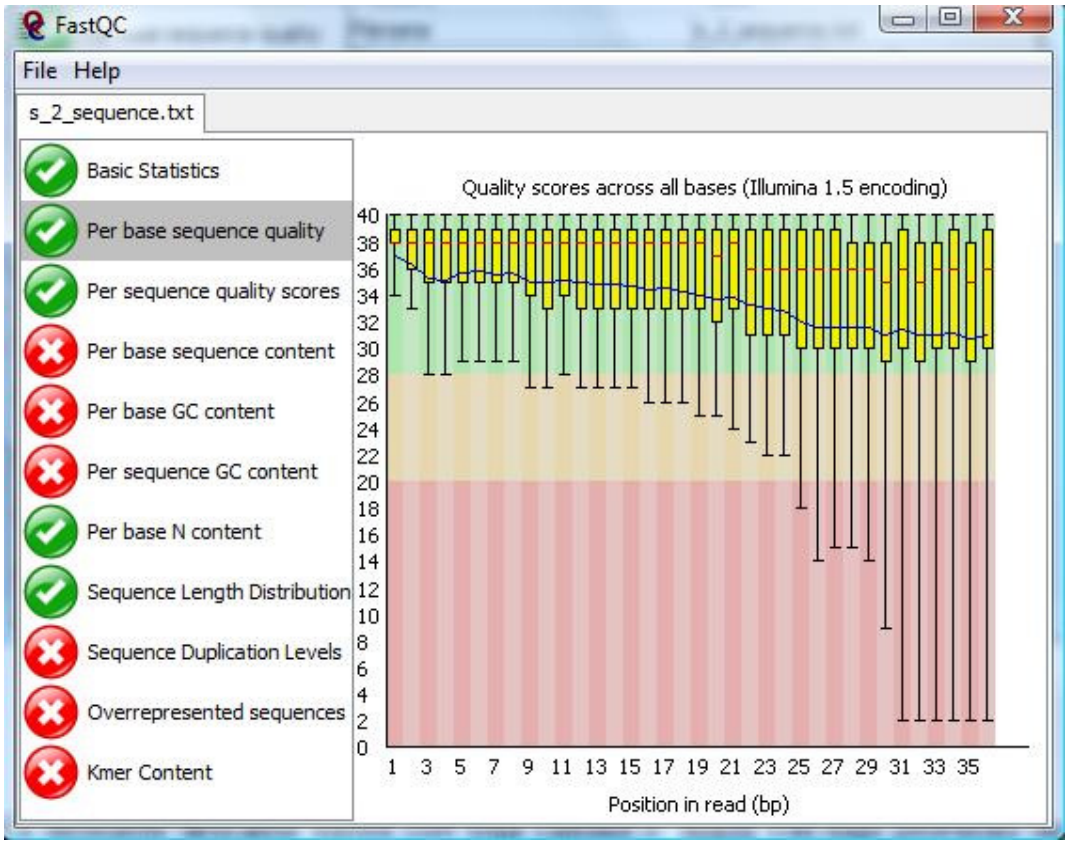


Figura 3.21: valores de calidad de cada base a través de todos los *reads*

Otras utilidades de esta aplicación se relacionan con la detección de sesgos sistemáticos en nuestros datos. La figura 3.22 muestra el porcentaje del contenido de bases tomando todos los *reads*. Los picos que se ven a la izquierda significan que en la posición 1 hay una G para el 100% de los *reads* y en la posición 2, una A, también para el 100% de los *reads*. Esto no es razonable *a priori* ya que significa que todos los *reads* comienzan con las bases GA, lo que lleva a pensar en algún artefacto producto del proceso de secuenciación. Esta situación tuvo su origen en el *primer* utilizado en la secuenciación, que no fue eliminado correctamente, restando de él, el par de bases GA que fueron leídas como parte de cada uno de los *reads*. Estas dos bases que no corresponden en realidad a la información de *T. cruzi*, deben ser recortadas.

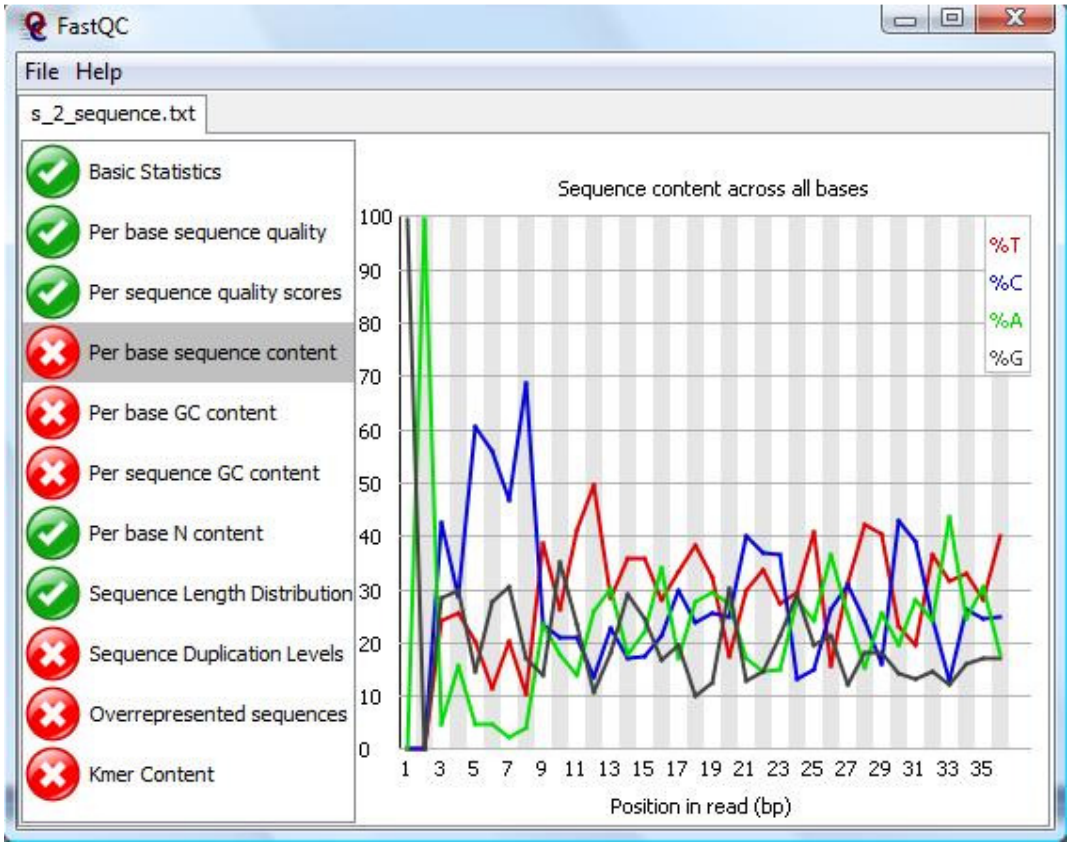


Figura 3.22: porcentaje de contenido de bases de los *reads*

Otra de las solapas muestra la gráfica del nivel de duplicación de los *reads* (figura 3.23). Debe interpretarse de la siguiente manera: el pico de la izquierda significa que el 100% de los *reads* aparecen una vez, lo que resulta obvio, pero aquello que guarda interés son las repeticiones. Por ejemplo, los *reads* que se hallan repetidos dos veces, corresponden al 21% del total, aproximadamente. De esta manera es posible interpretar el resto de la gráfica.

Esto quiere decir que casi todos los *reads* presentan más de una copia en el experimento, habiendo un 40% con 10 copias o más. Esto puede tener tres explicaciones posibles que no son excluyentes. O bien la cobertura de los *reads* para estos experimentos es tan alta que es frecuente encontrar *reads* repetidos o puede deberse a un sesgo producto del PCR previo al proceso de secuenciación o, dado que el genoma de *T. cruzi* es tan repetitivo, se tienen *reads* iguales que tienen origen en diferentes lugares del genoma. Esto último es consistente con lo que sabemos del genoma de *T. cruzi* en cuanto a su naturaleza repetitiva.

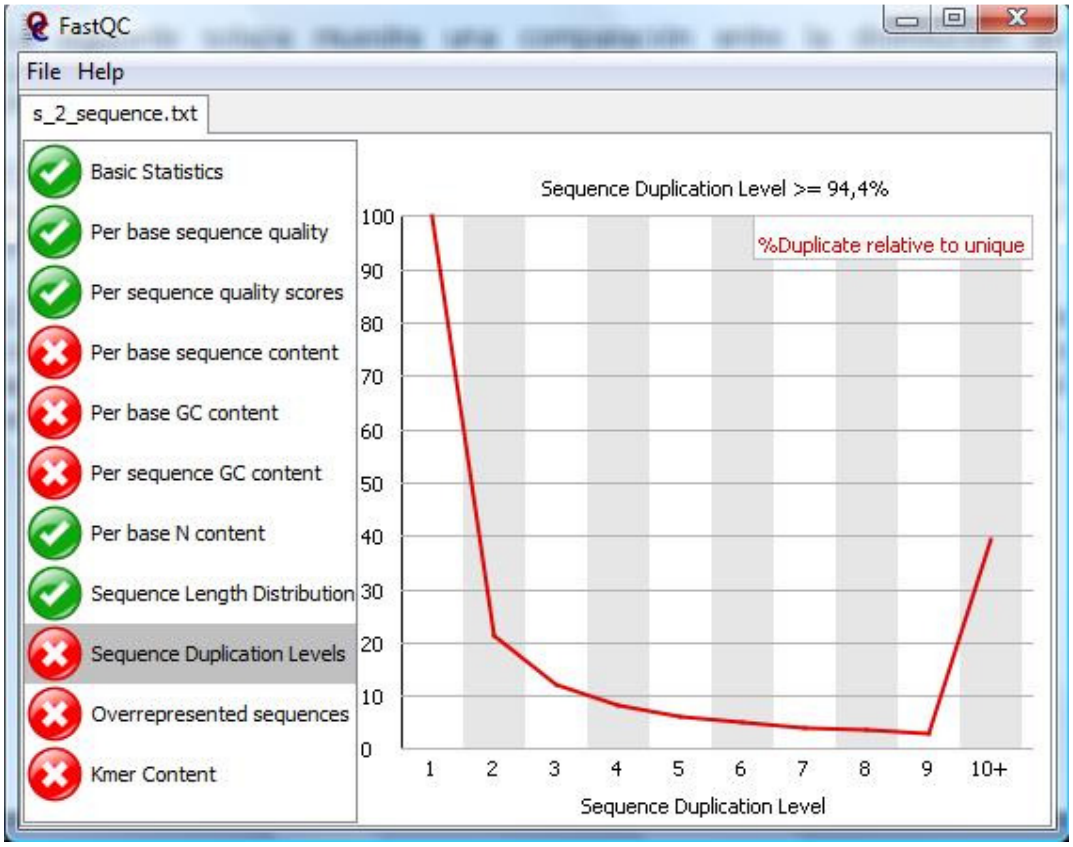


Figura 3.23: nivel de duplicación de los *reads*

Una vez terminado al análisis de los datos crudos con FastQC, se procede al filtrado para obtener el subconjunto de datos que presenten mejor calidad y sobre los cuales se trabaja en adelante. Para esto es deseable descartar bases con baja calidad (*trim* o recorte de los *reads*) o *reads* enteros con bajo promedio de calidad de sus bases. Un valor razonable de corte para la calidad resulta entre 20 y 25.

Se llevó a cabo el procesamiento de los *reads* por calidad y se obtuvieron tres archivos filtrados en formato fastq. Por un lado fue necesario el recorte del par de bases que corresponden al *primer* y por otro lado se filtró por calidad, permitiendo solo los *reads* con valor de calidad promedio mayor o igual a 20.

Para el caso que se ejemplifica en las figuras precedentes (estadio amastigota), los *reads* que persisten luego del filtrado son 15.107.022. Considerando que se tenían originalmente unos 18.500.000 *reads*, la pérdida de datos por el recorte de calidad fue de un 18%.

Algunos aspectos mejoran con el filtrado y la diferencia se puede apreciar comparando el resultado del análisis con FastQC antes y después de aplicar el filtro.

En la figura 3.24 se puede ver la mejora en la calidad de los *reads* filtrados respecto a los datos crudos. La mejora se hace notoria principalmente hacia el final de los *templates* (mitad derecha de la representación gráfica). Comparar con figura 3.21.

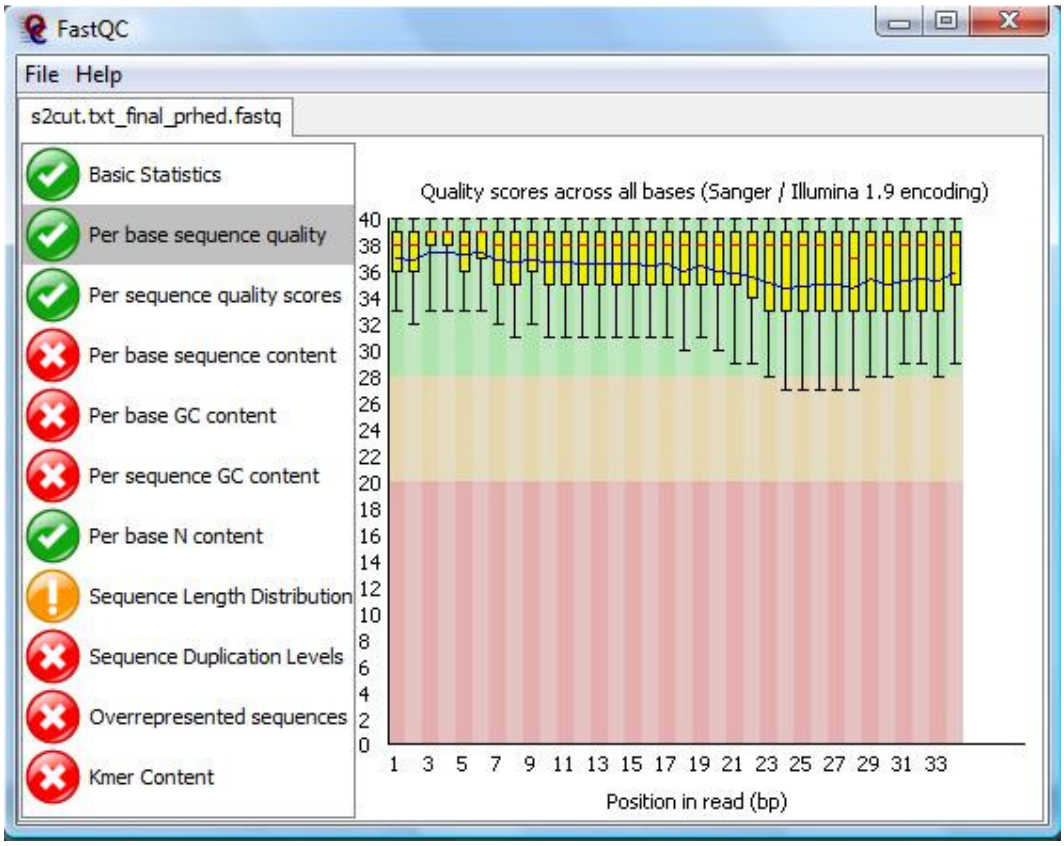


Figura 3.24: valores de calidad de cada base a través de todos los *reads* (datos filtrados)

El mismo análisis fue llevado a cabo para los estadios epimastigota y trypomastigota con resultados similares (que no es incluyen).

A partir de los archivos filtrados se ejecutaron las modificaciones necesarias para obtener archivos en formato fasta, sobre los cuales se hacen los análisis bioinformáticos.

6. Determinación de la expresión diferencial de genes

Para determinar qué genes se expresan diferencialmente, esto es, qué genes presentan una diferencia significativa cuando se compara el valor dado por el conteo de *reads* entre los estadios, es necesario utilizar un test estadístico que permita la diferenciación. En este capítulo se delinearán las bases de las técnicas utilizadas para identificar los genes con expresión diferencial.

6.1. Aplicación del test estadístico según los datos

La aplicabilidad de los distintos tests estadísticos depende de los datos. En nuestro caso se tiene un conteo de *reads* sobre tres estadios, se trata del tipo de datos discreto para tres muestras. Si bien es previsible la tendencia a usar algún test de los que resultan aplicables a tres o más muestras, son de mayor interés las comparaciones 2 a 2.

Es posible considerar que los *reads* obtenidos para cada estadio son generados por un proceso de Poisson, cuya distribución no es normal.

Los datos corresponden al conteo para cada gen en cada estadio, o sea, para un mismo gen se tienen tres valores numéricos, uno por cada estadio. Se considerarán los datos como apareados por tratarse del mismo gen en cada experimento, si bien los individuos (tripanosomas) usados para obtener cada muestra biológica no fueron los mismos. Entonces, para las comparaciones 2 a 2 considerando datos *paired*, no paramétricos, se decidió aplicar el test de Audic y Claverie, por lo que cabe profundizar en sus bases teóricas para su posterior aplicación a los datos.

El test exacto de Fisher y otras técnicas que se suelen utilizar para el análisis de tablas de contingencia de 2 x 2, han sido ampliamente usados, por ejemplo en experimentos donde se tiene una muestra tratada y otra normal. Sin embargo Claverie plantea algunas consideraciones sobre la aplicación del test de Fisher, a saber, que los datos de los genes en un experimento de dos condiciones deben ser escritos en forma un poco artificial; como se muestra a continuación [Claverie, 1999]:

	Condición A	Condición B
Gen 1	g_{1A}	g_{1B}
Todos los demás genes	N_A-g_{1A}	N_B-g_{1B}

Donde g_{1A} y g_{1B} son los conteos asociados al gen 1 y N_A y N_B es el total de *reads* generados en los experimentos A y B respectivamente.

Desde el punto de vista de la teoría, la definición de "todos los demás genes" agregados en una sola categoría es lógicamente inconsistente, ya que implica una comparación de un gen contra la sumatoria de todos los demás. Sin embargo, en la práctica, los valores obtenidos para el test de Audic y Claverie y los obtenidos para el test de Fisher son similares, siendo el último test más conservador. [Claverie, 1999]

Otro test ampliamente usado es t-student que en nuestro caso no es aplicable pues la variabilidad de los datos no puede estimarse porque se carece de réplicas, tanto técnicas como biológicas. Otra complicación en relación al test de la t es que este test se aplica a datos continuos y asume distribución normal, ambas condiciones no son cumplidas por nuestros datos.

6.2. Test estadístico de Audic y Claverie

Asumimos que los *reads* obtenidos en cada experimento de secuenciación obedecen a un proceso de Poisson y que el proceso que los genera, es el mismo para cada experimento. Audic y Claverie denotan por $p(x)$, la probabilidad de observar x *reads* pertenecientes al mismo gen cuando son tomados al azar N *reads*. En otros términos, en un experimento donde se obtienen N *reads*, $p(x)$ es la probabilidad de observar x *reads* que corresponden a un mismo gen. [Audic y Claverie, 1997]

Bajo estas condiciones es razonable suponer como hipótesis nula (H_0) que los genes se expresan de forma no diferenciada en los tres experimentos que se llevaron a cabo (para cada uno de los tres estadios secuenciados del ciclo de vida de *T. cruzi*).

Cuando N es grande (mayor a 1.000), $p(x)$ se aproxima a una distribución de Poisson:

$$p(x) = \frac{e^{-\lambda}\lambda^x}{x!} \quad (1)$$

Donde λ es el número de transcriptos, que es desconocido.

Si se repite el experimento, se observarán y ocurrencias del mismo transcripto en cuanto al conteo de *reads*. Entonces cabe cuestionarse cuánto vale $p(y)$. Una solución aproximada consiste en usar x como estimador de máxima verosimilitud para λ y calcular la probabilidad de y ocurrencias dada una distribución de Poisson con $\lambda = x$:

$$p(y|x) = \frac{e^{-x}x^y}{y!} \quad (2)$$

La ecuación (2) no es simétrica en x e y. Esto es una falla ya que la probabilidad no puede depender de cuál de los valores x o y es observado primero. La igualdad $p(y|x) = p(x|y)$ debe mantener la condición de que un número N de clones se tienen en ambos experimentos. La ecuación (2) no es correcta porque no se ha tomado en cuenta la fluctuación de x alrededor de λ. Para tomar en consideración esto, es necesario integrar $p(y|x)$ en todos los valores posibles de λ, considerando un enfoque Bayesiano:

$$p(y|x) = \int_0^\infty d\lambda\, p(d = \lambda|x)p(y|d = \lambda) \quad (3)$$

$p(d = \lambda|x)$ en la ecuación (3) es la probabilidad de que la actual abundancia de un transcripto determinado sea λ dado que se observaron x *reads* en un experimento. El segundo término en la integral es la probabilidad de tener y ocurrencias, dada una distribución de Poisson de parámetro λ:

$$p(y|d = \lambda) = \frac{e^{-\lambda}\lambda^y}{y!} \quad (4)$$

Usando el teorema de Bayes $p(d = \lambda|x)$ puede escribirse de la siguiente manera:

$$p(d = \lambda|x) = \frac{p(x|d = \lambda)p(d = \lambda)}{p(x)}$$

Por el teorema de la probabilidad total[5], el denominador de esta ecuación se puede escribir como la integral que aparece en la ecuación siguiente:

$$p(d = \lambda|x) = \frac{p(x|d=\lambda)p(d=\lambda)}{\int_0^\infty d\lambda' p(x|d=\lambda')p(d=\lambda')} \quad (5)$$

$p(d = \lambda)$ es desconocido pero es razonable asumir que corresponde a una distribución uniforme entre 0 y N (donde $N \longrightarrow \infty$) y entonces $p(d = \lambda)$ en el numerador se simplifica con $p(d = \lambda')$ en el denominador porque se asume que toman un valor constante.

Sustituyendo además la ec. (1) $p(x|d = \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$ en (5), resulta:

[5] Teorema de la probabilidad total: $p(B) = \sum_{i=1}^n p(B|A_i)p(A_i)$

$$p(d = \lambda|x) = \frac{\lambda^x e^{-\lambda}/x!}{\int_0^\infty d\lambda' \lambda'^x e^{-\lambda'}/x!} \quad (6)$$

$x!$ en el denominador no está afectado por la integral, que es en λ' y se simplifica con $x!$ en el numerador. La integral que queda en el denominador: $\int_0^\infty d\lambda' \lambda'^x e^{-\lambda'}$ es una función $\Gamma(x+1)$[6] y la ecuación (6) resulta:

$$p(d = \lambda|x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (7)$$

Sustituyendo (7) en (3) se tiene:

$$p(y|x) = \int_0^\infty d\lambda \, \frac{\lambda^x e^{-\lambda}}{x!} \frac{e^{-\lambda}\lambda^y}{y!} = \frac{1}{x!y!} \int_0^\infty d\lambda \lambda^x e^{-\lambda} e^{-\lambda} \lambda^y = \frac{1}{x!y!} \int_0^\infty d\lambda \lambda^{(x+y)} e^{-2\lambda} \quad (8)$$

Cambio de variable: $t = 2\lambda \Rightarrow d\lambda = \frac{dt}{2}$

Aplicando el cambio de variable en (8) resulta:

$$p(y|x) = \frac{1}{x!\, y!} \int_0^\infty \frac{dt}{2} \frac{t^{(x+y)}}{2^{(x+y)}} e^{-t} = \frac{1}{x!\, y!} \frac{1}{2^{(x+y+1)}} \int_0^\infty dt \; t^{(x+y)} e^{-t}$$

Donde $\int_0^\infty dt t^{(x+y)} e^{-t}$ es la función $\Gamma(x+y+1)$ que es igual a $(x+y)!$, lo que resulta en:

$$p(y|x) = \frac{1}{x!y!} \frac{1}{2^{(x+y+1)}} (x+y)! \quad (9)$$

Hasta aquí se asumió que los experimentos de donde se extraen los x e y *reads* son iguales en cantidad de transcriptos (N), pero esto no necesariamente sucede. Por eso se considerará que se tienen dos experimentos con cantidades diferentes de tags (N_1 y N_2). En este caso los procesos de Poisson generan con diferente parámetro (λ_1 y λ_2) de forma que la relación entre las Ns es igual que la relación entre las λs. O sea:

[6] Por definición y propiedades de la función Γ ver anexo I.

$$\frac{\lambda_1}{\lambda_2} = \frac{N_1}{N_2} \tag{10}$$

Reescribiendo la ecuación (3) bajo estas condiciones, resulta:

$$p(y|x) = \int_0^\infty d\lambda_1 \, p(d = \lambda_1|x) \, p(y|d = \lambda_2) \tag{11}$$

La ecuación (11) se integrará en λ_1 por lo que es necesario sustituir λ_2 en función de λ_1.

A partir de la ecuación (10) se tiene que: $\lambda_2 = \frac{N_2}{N_1}\lambda_1$

Sustituyendo en (11):

$$p(y|x) = \int_0^\infty d\lambda_1 \, p(d = \lambda_1|x) \, p(y|d = \frac{N_2}{N_1}\lambda_1) \tag{12}$$

Utilizando los resultados del las ecuaciones (4) y (7) se puede reescribir la ecuación (12) de la siguiente manera:

$$p(y|x) = \int_0^\infty d\lambda_1 \frac{{\lambda_1}^x e^{-\lambda_1}}{x!} \frac{(\frac{N_2}{N_1}\lambda_1)^y e^{-\frac{N_2}{N_1}\lambda_1}}{y!} \tag{13}$$

Se hacen algunos cálculos a partir de la ecuación (13):

$$p(y|x) = (\frac{N_2}{N_1})^y \frac{1}{x!\,y!} \int_0^\infty d\lambda_1 \, {\lambda_1}^x e^{-\lambda_1} \, {\lambda_1}^y e^{-\frac{N_2}{N_1}\lambda_1}$$

$$p(y|x) = (\frac{N_2}{N_1})^y \frac{1}{x!\,y!} \int_0^\infty d\lambda_1 \, {\lambda_1}^{(x+y)} \, e^{-\lambda_1(1+\frac{N_2}{N_1})}$$

Cambio de variable: $t = \lambda_1\left(1 + \frac{N_2}{N_1}\right) \Rightarrow dt = d\lambda_1\left(1 + \frac{N_2}{N_1}\right)$

$$p(y|x) = (\frac{N_2}{N_1})^y \frac{1}{x!\,y!} \int_0^\infty \frac{dt}{\left(1 + \frac{N_2}{N_1}\right)} \frac{t^{(x+y)}}{\left(1 + \frac{N_2}{N_1}\right)^{(x+y)}} e^{-t}$$

$$p(y|x) = (\frac{N_2}{N_1})^y \frac{1}{x!\,y!} \frac{1}{\left(1+\frac{N_2}{N_1}\right)^{(x+y+1)}} \int_0^\infty dt\; t^{(x+y)}\, e^{-t}$$

La integral corresponde nuevamente a una función Γ:

$$\Gamma(x+y+1) = (x+y)!$$

Sustituyendo resulta:

$$p(y|x) = \left(\frac{N_2}{N_1}\right)^y \frac{1}{x!y!} \frac{1}{\left(1+\frac{N_2}{N_1}\right)^{(x+y+1)}} (x+y)! \qquad (14)$$

La ecuación (14) constituye el test de Audic y Claverie pero no basta por sí sola. No es simétrica en x e y a diferencia de la ecuación (9) que sí lo era.

Audic y Claverie calculan intervalos de confianza para la aplicación del test, donde para un valor dado de x, conociendo el p-valor se da un intervalo de valores posibles para y. Los p-valores considerados en el *paper* fueron de 0,01 y 0,05. Ver figura 3.25 [Audic y Claverie, 1997]. No se profundizará en la metodología de construcción de estos intervalos ya que nuestra implementación para la aplicación del test empleó otro enfoque.

	$2\epsilon = 0.01$	$2\epsilon = 0.05$
x	y_{min}—y_{max}	y_{min}—y_{max}
0	⋆—7	⋆—5
1	⋆—10	⋆—7
2	⋆—12	⋆—9
3	⋆—14	⋆—11
4	⋆—16	⋆—12
5	⋆—18	0—14
6	⋆—19	0—16
7	0—21	1—17
8	0—23	1—19
9	0—24	2—20
10	1—26	2—22
11	1—27	3—23

Figura 3.25: fragmento de la tabla de intervalos de confianza que se incluye en el *paper* de Audic y Claverie [Audic y Claverie, 1997].

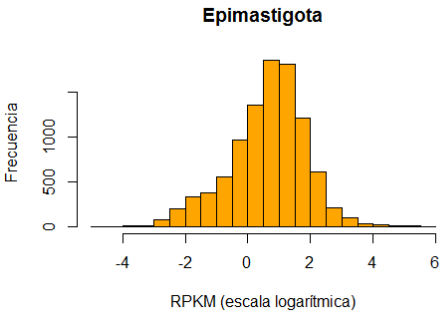
Resultados

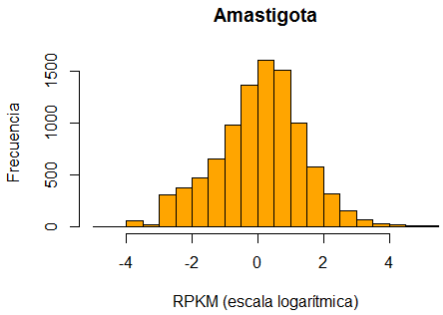
1. Distribución de los datos y análisis estadístico básico

Los alineamientos contra los genes dieron como resultado distintas cantidades de *reads* mapeados para cada estadio, siendo para epimastigota algo más de 470.000, para amastigota del orden de 450.000 y para trypomastigota correspondieron alrededor de 610.000. Resulta notoria la diferencia de trypomastigota respecto a los otros estadios pero se debe recordar que el éxito de ese experimento fue mayor que para los otros dos, lo que en parte justifica la diferencia de *reads* que mapean es ese estadio.

Los datos, que ya habían sido filtrados por calidad, se filtraron nuevamente para remover los *outliers* y se graficó el histograma del conteo de *reads* normalizado (RPKM) para cada estadio, lo que se muestra en la figura 4.1. Para que la visualización sea apropiada, se aplicó el logaritmo sobre los

valores de RPKM. Además se resumen algunos datos estadísticos básicos en la tabla T.4.1.





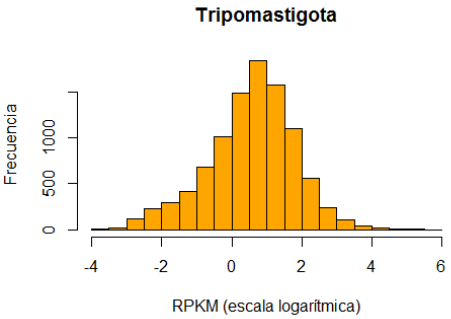


Figura 4.1: histogramas para los tres estadios a partir del RPKM en escala logarítmica

Epimastigota	
Mínimo	0
Máximo	294,21
Promedio	3,74
Desviación estándar	8,738
Amastigota	
Mínimo	0
Máximo	214,92
Promedio	2,483
Desviación estándar	6,812
Trypomastigota	
Mínimo	0
Máximo	360,8
Promedio	3,716
Desviación estándar	9,94

Tabla T.4.1: Valores de RPKM: algunos datos estadísticos básicos por estadio

2. Análisis preliminar a partir de las gráficas de expresión

2.1 Comparaciones entre los distintos estadios del ciclo de vida de *T. cruzi*

Se realizaron gráficas comparando los estadios dos a dos y se obtuvieron las gráficas que se muestran en la figura 4.2 (a), (b) y (c). Se graficaron los conteos correspondientes a los diferentes estadios para los mapeos realizados con margen y sin margen. Se graficaron los conteos crudos en lugar del RPKM, o sea, un punto en la gráfica representa un gen al que le corresponde un valor de conteo de *reads* en ciertas condiciones en las abscisas y bajo otras condiciones en las ordenadas. Ya que se trata de un mismo gen, la corrección por su largo no resulta necesaria.

La gráfica 4.2 (a) corresponde a los estadios epimastigota y amastigota y resulta una distribución bastante uniforme entorno a la diagonal. Esto sugiere *a priori* que el nivel de expresión de los genes es similar para ambos estadios. No sucede lo mismo para las gráficas 4.2 (b) y 4.2 (c) donde es notoria la presencia de un grupo de genes que aparecen por encima de la diagonal, sugiriendo sobreexpresión de esos genes para el estadio trypomastigota respecto al epimastigota y al amastigota.

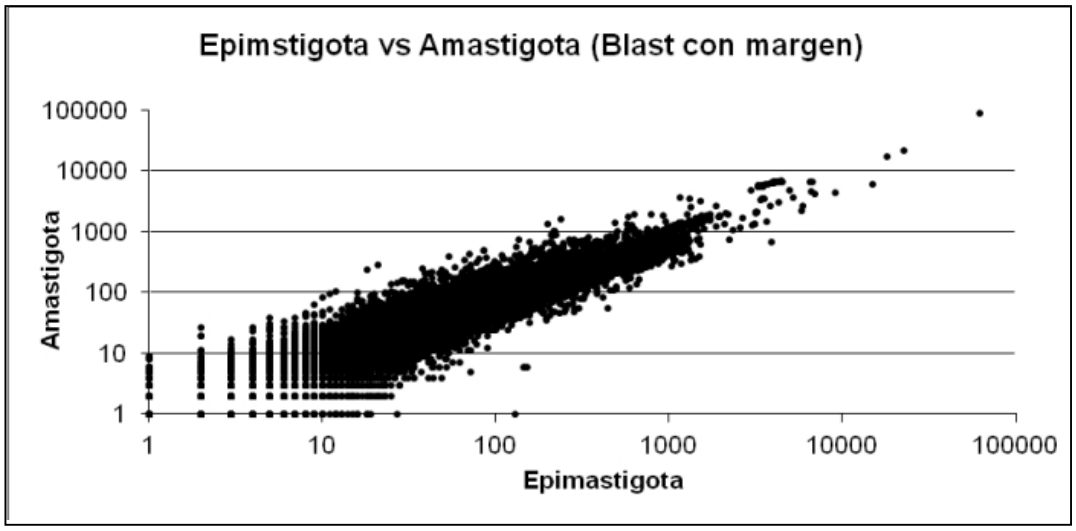


Figura 4.2 (a): Genes para los estadios epimastigota y amastigota (escala doble logarítmica)

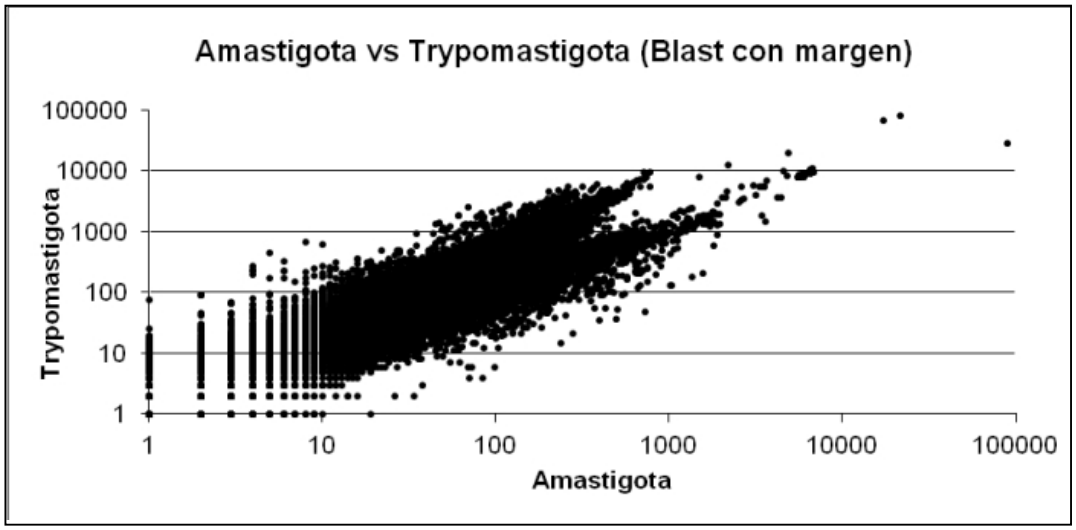


Figura 4.2 (b): Genes para los estadios amastigota y trypomastigota

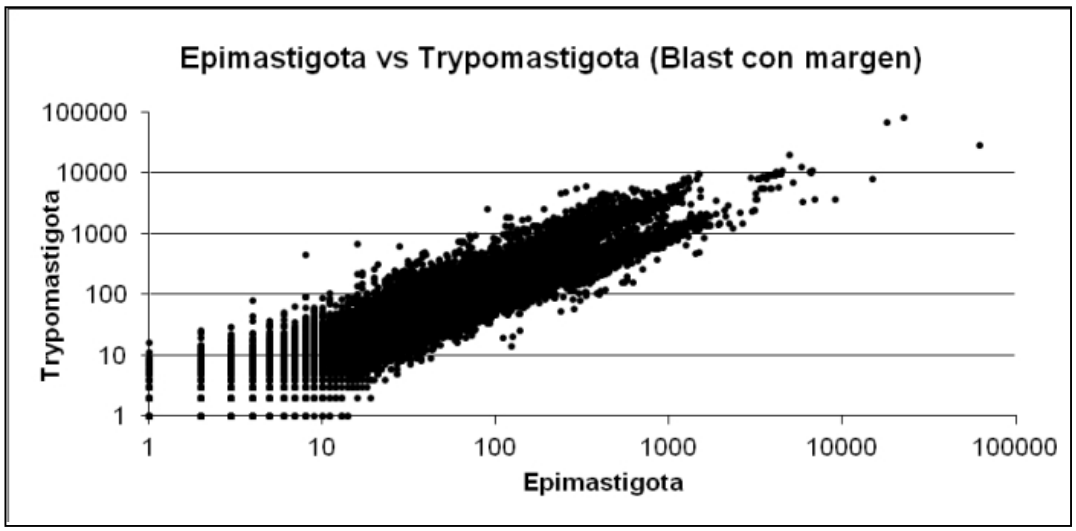


Figura 4.2 (c): Genes para los estadios epimastigota y trypomastigota

Se aislaron los genes de los que se presume posible sobreexpresión y se obtuvieron sus funciones, resultando principalmente en proteínas de membrana (MASPs, mucinas, sialidasas, GP63, etc.) para ambos casos.

2.2. El problema de los *multimatches*

BLAST y Bowtie manejan de forma diferente los *multimatches*. Dada una región repetida en el genoma y un *read* que matchea con ella, BLAST asignará ese *read* a todas las regiones del genoma donde se repita ese patrón, pero Bowtie

asignará en función de los parámetros que se le pasaron. De esta manera, tanto BLAST como Bowtie sobreestiman en el tratamiento de los *multimatches*. Por este motivo se utilizó el paquete de scripts de ERANGE, que considera las repeticiones al momento de asignar los *reads* a cada gen y los asigna de forma proporcional fijándose en los extremos de los genes (zonas UTRs).

3. Aplicación del test estadístico

El test estadístico aplicado fue el de Audic y Claverie [Audic y Claverie, 1997]. Este test resulta indicado, no solo por haber sido ideado de forma específica para datos de secuenciación, sino también por el tipo de datos de que se dispone. Se cuenta con tres muestras de datos con las siguientes características:

- dependientes (*paired*) desde el punto de vista que se trata de los mismos genes
- discretos (porque constituyen el conteo de *reads*)
- distribución no normal
- *two-tailed* (la variación puede darse hacia cualquiera de los extremos).

Se utilizó la herramienta IDEG6 para el cálculo del test y la determinación de la expresión diferencial. Esta herramienta (de uso web) fue implementada por Romualdi et al., quienes además compararon el desempeño de varios tests estadísticos. Sostienen que, aunque el test exacto de Fisher fue ampliamente utilizado en un principio (por ejemplo en el proyecto CGAP -Cancer Genome Anatomy Project-), su aplicabilidad a este tipo de problemas no es adecuada. Implementaron en IDEG6 los siguientes tests: Audic y Claverie, test exacto de Fisher 2x2, Chi-cuadrado, estadístico R y Greller y Tobin. Al aplicar los tests sobre un conjunto de datos notaron que cuando la diferencia de expresión es muy marcada todos ellos tienen éxito hallándola pero cuando la diferencia es sutil, los tests con mejor comportamiento resultaron ser Chi-cuadrado y Audic y Claverie. Aunque Chi-cuadrado tiene mayor sensibilidad, Audic y Claverie es un test especialmente diseñado para problemas de determinación de expresión de genes y está basado en un modelo probabilístico más apropiado. [Romualdi et al., 2001]

El archivo de entrada para esta herramienta debe tener el encabezado:

UNIQID Descr <tamaño librería 1> … <tamaño librería n>

Donde la columna encabezada por UNIQID corresponde al identificador del gen, Descr hace referencia a su función y luego se tienen los conteos. Una columna por cada experimento, en nuestro caso se tienen tres columnas (una por cada estadio secuenciado de *T. cruzi*). Un registro cualquiera sería:

Tc00.1047053397937.10 pumilio%2FPUF+RNA+binding+protein+1%2C+putative 29 28 33

El p-valor elegido fue 0,001. La elección del p-valor se basa en la tabla T.4.2. Esta tabla se confeccionó con los datos de las corridas en IDEG6 para los distintos tests. Por ejemplo para el test de Audic y Claverie, con un p-valor de 0,05, se obtienen 2746 genes diferencialmente expresados. La decisión del p-valor correspondiente a 0,001 se tomó considerando que a partir de ese valor no se obtienen resultados diferentes independientemente del test del que se trate. O sea, para cualquiera de los tests, considerados individualmente, se obtiene igual cantidad de genes diferencialmente expresados si se usa un p-valor de 0,001 o menor.

	p-valor				
Test	0.05	0.01	0.001	0.0001	0.00001
Audic & Claverie	2746	2555	2555	2555	2555
Fisher exact test	2406	2265	2265	2265	2265
Chi-squared 2x2	2513	2332	2332	2332	2332
General Chi-squared	2618	2425	2297	2297	2297

Tabla T.4.2: cantidad de genes diferencialmente expresados por test para los cinco p-valores diferentes ofrecidos por la herramienta IDEG6

Como salida se obtiene un listado de los genes diferencialmente expresados como se muestra en la figura 4.3. Las columnas, de izquierda a derecha, corresponden al identificador del gen, la descripción de la función, el conteo para cada una de las librerías (epimastigota, amastigota, trypomastigota), un valor normalizado del conteo por cada librería y el valor obtenido para el test 2 a 2 (epimastigota-amastigota, epimastigota-trypomastigota, amastigota-trypomastigota). El código de colores se interpreta de la siguiente forma. Para el conteo normalizado (columnas 6, 7 y 8) se pinta el fondo con diferente intensidad de azul para denotar mayor expresión. Con el fondo en amarillo se marcan los estadios entre los que se detecta expresión diferencial. Por ejemplo, para el gen Tc00.1047053400945.10 se puede ver que el conteo normalizado para trypomastigota, de valor 8,7, tiene un fondo azul más intenso que los otros conteos normalizados para ese gen (con valores 2,5 y 2,2). Se

marca en amarillo los valores de test que corresponden a los estadios epimastigota-trypomastigota (columna 10, AC 1-3) y amastigota-trypomastigota (columna 11, AC 2-3). Se debe entonces comparar los valores normalizados para estos estadios para determinar si se trata de sobre o subexpresión. Como el conteo normalizado de trypomastigota es mayor que el de epimastigota se llega a la conclusión que trypomastigota se sobreexpresa respecto a epimastigota en el caso del gen que se tomó de ejemplo. Se hace un razonamiento similar para concluir que trypomastigota se sobreexpresa respecto a amastigota para este mismo gen.

UNIQID	Description	Lib1	Lib2	Lib3	Lib1(norm)	Lib2(norm)	Lib3(norm)	AC 1 2	AC 1 3	AC 2 3
Tc00.1047053397937.10	pumilio%2FPUF+RNA+binding+protein+1%2C+putative	29	28	33	0.4	0.6	0.3	0.040230	0.006492	0.000760
Tc00.1047053397937.5	N-acetylglucosamine-6-phosphate+deacetylase-like+protein%2C+putative	17	17	10	0.3	0.3	0.1	0.056461	0.000798	0.000071
Tc00.1047053398345.10	hypothetical+protein%2C+conserved	34	17	17	0.5	0.3	0.1	0.023350	0.000001	0.001706
Tc00.1047053399033.19	hypothetical+protein%2C+conserved	16	24	14	0.2	0.5	0.1	0.007609	0.006802	0.000004
Tc00.1047053400945.10	trans-sialidase+%28pseudogene%29%2C+putative	162	111	1041	2.5	2.2	8.7	0.019048	0.000000	0.000000
Tc00.1047053401041.10	retrotransposon+hot+spot+protein+%28RHS%2C+pseudogene%29%2C+putative	157	121	111	2.4	2.4	0.9	0.027251	0.000000	0.000000
Tc00.1047053401569.10	trans-sialidase%2C+putative	57	16	177	0.9	0.3	1.5	0.000042	0.000030	0.000000
Tc00.1047053404001.20	hypothetical+protein%2C+conserved	376	292	273	5.8	5.9	2.3	0.017178	0.000000	0.000000
Tc00.1047053404843.20	hypothetical+protein%2C+conserved	63	59	53	0.10	1.2	0.4	0.021580	0.000005	0.000000
Tc00.1047053404975.30	trans-sialidase+%28pseudogene%29%2C+putative	136	24	522	2.1	0.5	4.4	0.000000	0.000000	0.000000

Figura 4.3: fragmento de la salida de la herramienta IDEG6 para el test de Audic y Claverie con p-valor de 0,001

Se desconoce la implementación en detalle del algoritmo de Romualdi [Romualdi et al., 2001] pero resulta claro que el valor dado por el test de Audic y Claverie por sí solo no es suficiente para determinar la expresión diferencial. Es posible que hayan calculado los intervalos de confianza para y dado x y un p-valor, tal como refiere el *paper* de Audic y Claverie [Audic y Claverie, 1997].

Como parte de este trabajo se desarrolló un script en python para la determinación de la expresión diferencial considerando el test de Audic y Claverie. La motivación para este desarrollo fue, por un lado, el entendimiento de un método para el cálculo de la probabilidad acumulada entre un valor determinado y un extremo (en este caso 0 o infinito), el que a la postre será comparado con el p-valor deseado y por otro lado, aumentar la precisión en los valores dados por la herramienta IDEG6.

En principio cabe destacar que la implementación de la ecuación del test de Audic y Claverie requiere de algún cuidado ya que no es posible calcular el factorial de números grandes directamente. Esto llevó a que la implementación de la ecuación (14) de la sección 6.2 del capítulo 3, haya debido ser hecha por partes. Pero el mayor interés a nivel del código le corresponde al cálculo de las integrales para la probabilidad acumulada. Para ilustrar esto se usará un ejemplo donde el conteo para el estadio epimastigota (x) es mayor que el conteo para amastigota (y). Ver figura 4.4. El cálculo de la probabilidad de y

dado x, $p(x|y)$, según la ecuación de Audic y Claverie da un valor que será tomado como referencia. Se calcula la sumatoria de las probabilidades $p(x|n)$ con $n \geq 0$ y $n \leq y$. Esto da la probabilidad que es referida en la figura 4.4 como probabilidad acumulada cero. No existe simetría por lo que un cálculo similar para el otro extremo de la curva, no dará el mismo resultado; así como tampoco la probabilidad acumulada cero puede ser comparada con el p-valor/2. Lo que debe hacerse es calcular las probabilidades acumuladas para ambos extremos y comparar el resultado de su suma con el p-valor. Esto fue implementado en el algoritmo de la siguiente manera. Se iteró en $t = x + 1$ hasta un valor límite que se especifica en el código y que fue fijado en 500.000 (variable top en el código). El valor de tope es intencionalmente exagerado para que no pueda darse la situación de que t no crezca lo suficiente. Se calculó la probabilidad $p(x|t)$ hasta llegar al valor de t para el cual la probabilidad calculada fuera menor o igual a la de referencia, $p(x|y)$. A partir de ese t se calculará la sumatoria a infinito que corresponde a la probabilidad acumulada del extremo derecho de la gráfica. Lo que se hizo para resolver esta sumatoria infinita fue poner dos topes. Por un lado se permite la suma mientras que los términos a sumar sean mayores que un cierto valor que se especifica en el código (en las corridas se trabajó con un tope para términos de $1x10^{-11}$, variable bottom en el código). Como la función converge rápidamente, en algunos cientos de iteraciones se tiene que el nuevo sumando es menor que el límite establecido y se deja de agregar términos a la sumatoria. Por otro lado se verifica que la suma de las probabilidades acumuladas no sea mayor a la probabilidad de referencia, porque de ser así, ya no resulta de interés reportar ese gen porque no se expresa diferencialmente y no tiene sentido seguir sumando términos. A continuación de la figura 4.4 se incluye un fragmento del código que contempla el caso explicado previamente. Además se incluye todo el código en el anexo II.

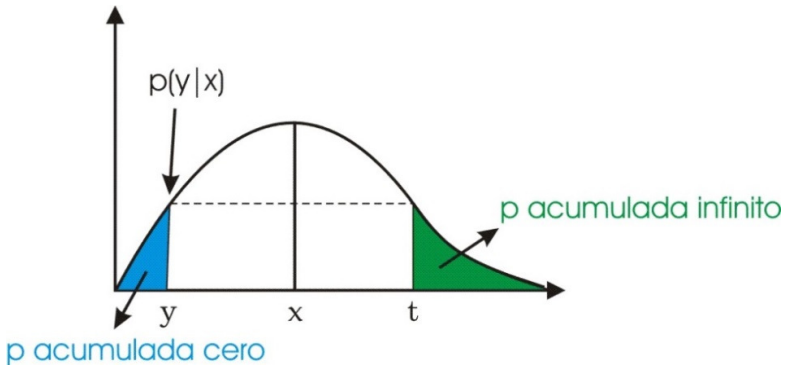


Figura 4.4: cálculo de las probabilidades de los extremos de la curva

acum0 es la probabilidad acumulada cero (el área en azul en la figura 4.4)

acumInf es la probabilidad acumulada infinito (el área en verde en la misma figura)
ac es una función que calcula la probabilidad según la ecuación (14) para el test de Audic y Claverie (ver sección 6.2 del capítulo 3).

Como se mencionó anteriormente las variables top y bottom son determinadas al principio del código y se usaron los siguientes valores:

```
top=500000
bottom=0.00000000001

for n in range(0,y+1):
        acum0=acum0+ac(N1,N2,x,n)
t=x+1
while t<top:
        pControl=ac(N1,N2,x,t)
        if (pControl<=pLimiteXY):
                nuevoY=t
                t=top
        t+=1
m=nuevoY
while m<top:
        acTest=ac(N1,N2,x,m)
        acumInf=acumInf+acTest
        if acTest<bottom or acumInf+acum0>pValor:
                m=top
        m+=1
```

Para determinar si existe expresión diferencial se compara la suma de las probabilidades extremas con el p-valor que el usuario debe introducir como parámetro de entrada al correr el algoritmo, o sea: si acum0 + acumInf < p-valor, entonces hay expresión diferencial.

Hace falta un detalle adicional para poder decir si la expresión diferencial es en más o en menos (sobre o subexpresión). Para eso es necesario comparar los valores de los conteos de los estadios normalizados. No importa cómo se lleva a cabo la normalización pero es imprescindible hacerla. En nuestro código simplemente se dividió el valor del conteo por el tamaño de la librería correspondiente. En el algoritmo de Romualdi se hizo de alguna otra manera y los valores normalizados se muestran en las columnas 6, 7 y 8 de la salida que generan, a lo que le agregan un fondo de color azul de intensidad relativa al valor normalizado. Entonces puede darse el caso donde se tiene que x es mayor que y, pero el valor normalizado de x es menor que el normalizado de y.

Esto no constituye una inconsistencia porque depende del éxito del experimento. En el caso referido puede darse que el experimento que corresponde al segundo estadio (que refiere al conteo en y) haya sido mucho más exitoso que el que corresponde al primer estadio (conteo en x).

Para testear el resultado de nuestro algoritmo contra el algoritmo de Romualdi et al. se utilizó un subconjunto de 25 genes seleccionados al azar. Las salidas de los algoritmos no coinciden totalmente pero los genes que son reportados como diferencialmente expresados por nuestro algoritmo también lo son por el de Romualdi, aunque IDEG6 reporta algunos genes o condiciones adicionales. En este sentido, nuestro algoritmo es más restrictivo.

Los datos de entrada de prueba fueron los siguientes:

Tc00.1047053421173.4	trans-sialidase%2C+putative		429	273	3492			
Tc00.1047053421321.9	tRNA+exportin%2C+putative		47	36	57			
Tc00.1047053421619.10	hypothetical+protein	8	5	7				
Tc00.1047053421959.10	eukaryotic+translation+initiation+factor+4e%2C+putative					31	29	34
Tc00.1047053422041.9	pre-mRNA+splicing+factor+ATP-dependent+RNA+helicase%2C+putative					6	1	6
Tc00.1047053422507.10	glycine+dehydrogenase+%5Bdecarboxylating%5D%2C+putative					36	17	34
Tc00.1047053422843.10	trans-sialidase+%28pseudogene%29%2C+putative				5	2	11	
Tc00.1047053423205.10	trans-sialidase%2C+putative	12	5	86				
Tc00.1047053424195.5	hypothetical+protein%2C+conserved			2	1	2		
Tc00.1047053424195.9	RNA-binding+protein+4	38	38	22				
Tc00.1047053424383.9	trans-sialidase+%28pseudogene%29%2C+putative				35	41	152	
Tc00.1047053424795.10	hypothetical+protein%2C+conserved			6	8	8		
Tc00.1047053424889.10	UDP-Gal+or+UDP-GlcNAc-dependent+glycosyltransferase%2C+putative					14	17	26
Tc00.1047053425311.9	hypothetical+protein%2C+conserved			161	65	123		
Tc00.1047053425435.10	trans-sialidase%2C+putative		293	126	878			
Tc00.1047053425851.9	hypothetical+protein%2C+conserved			73	94	160		
Tc00.1047053426397.9	kinesin%2C+putative	38	60	46				
Tc00.1047053427247.10	serine%2Fthreonine+protein+kinase%2C+putative				198	102	176	
Tc00.1047053427247.4	hypothetical+protein%2C+conserved			6	3	11		
Tc00.1047053427355.10	oligosaccharyl+transferase+subunit%2C+putative				49	52	94	
Tc00.1047053508125.22	mucin+TcMUCII+%28pseudogene%29%2C+putative				35	19	129	
Tc00.1047053506655.20	polyubiquitin+%28pseudogene%29%2C+putative				1837	1222	2129	
Tc00.1047053509013.10	calpain-like+cysteine+peptidase%2C+putative				167	176	93	
Tc00.1047053509965.380	dTDP-glucose+4%2C6-dehydratase%2C+putative				48	29	58	
Tc00.1047053510769.49	40S+ribosomal+protein+S6%2C+putative			22	17	28		

A continuación se presentan las salidas para ambos algoritmos, calculados sobre el mismo conjunto de datos de entrada y para un p-valor de 0,001.

La salida de IDEG6 se muestra en la tabla T.4.3 y corresponde a una adaptación de la salida web de este algoritmo.

Se eliminó la segunda columna, correspondiente a la descripción del gen, para mayor claridad. Esta salida debe ser interpretada de la siguiente manera. Es necesario centrarse en las últimas tres columnas (AC 1-2, AC 1-3 y AC 2-3), que corresponden al resultado del test de Audic y Claverie para los conteos de las librerías (columnas Lib1, Lib2 y Lib3). Por ejemplo, para el segundo gen aparece coloreada la última columna (AC 2-3). Eso significa que ese gen es

reportado por expresión diferencial entre los estadios 2 y 3, o sea, amastigota y trypomastigota. Para saber si se trata de sobre o subexpresión hay que comparar las columnas de las librerías normalizadas para estos estadios (L2n y L3n). Como L2n es mayor que L3n, se puede concluir que existe sobreexpresión del estadio amastigota respecto al trypomastigota o, lo que es equivalente, que se da una subexpresión del estadio trypomastigota respecto a amastigota para este gen.

UNIQID	Lib1	Lib2	Lib3	L1n	L2n	L3n	AC 1-2	AC 1-3	AC 2-3
Tc00.1047053421173.4	429	273	3492	6.6	5.5	29.3	0.001101	0.000000	0.000000
Tc00.1047053421959.10	31	29	34	0.5	0.6	0.3	0.04265	0.004346	0.000625
Tc00.1047053422507.10	36	17	34	0.6	0.3	0.3	0.016439	0.00077	0.028867
Tc00.1047053423205.10	12	5	86	0.2	0.1	0.7	0.059181	0.000000	0.000000
Tc00.1047053424195.9	38	38	22	0.6	0.8	0.2	0.025821	0.000002	0.000000
Tc00.1047053424383.9	35	41	152	0.5	0.8	1.3	0.009064	0.000000	0.00075
Tc00.1047053425311.9	161	65	123	2.5	1.3	1	0.000001	0.000000	0.005612
Tc00.1047053425435.10	293	126	878	4.5	2.5	7.4	0.000000	0.000000	0.000000
Tc00.1047053425851.9	73	94	160	1.1	1.9	1.3	0.000118	0.008611	0.000526
Tc00.1047053426397.9	38	60	46	0.6	1.2	0.4	0.00008	0.005417	0.000000
Tc00.1047053427247.10	198	102	176	3	2.1	1.5	0.000128	0.000000	0.000493
Tc00.1047053508125.22	35	19	129	0.5	0.4	1.1	0.030834	0.000013	0.000000
Tc00.1047053506655.20	1837	1222	2129	28.2	24.6	17.9	0.000008	0.000000	0.000000
Tc00.1047053509013.10	167	176	93	2.6	3.5	0.8	0.000278	0.000000	0.000000

Tabla T.4.3: salida del algoritmo IDEG6 para los datos de entrada de prueba

La salida de nuestro algoritmo para el mismo conjunto de datos de entrada se puede apreciar en la tabla T.4.4. Para comparar las salidas de ambos algoritmos hay que tener en cuenta el siguiente aspecto. Mientras que IDEG6 marca con fondo amarillo cuando hay expresión diferencial, nuestro algoritmo incluye una columna donde se explicita el tipo de expresión diferencial que se da y entre qué estadios sucede (última columna). Por este motivo, si para un gen dado se dan dos tipos de expresión diferencial, como es el caso del gen Tc00.1047053421173.4, IDEG6 pinta de amarillo dos celdas de una misma fila mientras que nuestro algoritmo reporta dos veces el mismo gen, o sea que aparecen dos filas, una por cada vez que se expresa diferencialmente (sobreexpresión de trypomastigota respecto a epimastigota y a amastigota). Por otro lado, nuestro algoritmo incluye una columna con la probabilidad a ser comparada con el p-valor ingresado (penúltima columna). Dado que el p-valor con que se hicieron las corridas fue de 0,001, esta columna reportará valores siempre menores que ese, ya que de ser mayores el gen no es reportado como diferencialmente expresado.

Gen	AC 1-2	AC 1-3	AC 2-3	Probab.	Sobre/Subexpresión
Tc00.1047053421173.4	0.001100633	4.81E-270	3.92E-251	4.81E-270	Sobreexpresion de tripomastigota respecto a epimastigota
Tc00.1047053421173.4	0.001100633	4.81E-270	3.92E-251	3.92E-251	Sobreexpresion de tripomastigota respecto a amastigota
Tc00.1047053423205.10	0.05918075	5.62E-08	2.91E-09	2.11E-07	Sobreexpresion de tripomastigota respecto a epimastigota
Tc00.1047053423205.10	0.05918075	5.62E-08	2.91E-09	1.14E-08	Sobreexpresion de tripomastigota respecto a amastigota
Tc00.1047053424195.9	0.025820667	2.33849E-06	1.26E-08	5.1703E-06	Subexpresion de tripomastigota respecto a epimastigota
Tc00.1047053424195.9	0.025820667	2.33849E-06	1.26E-08	2.55E-08	Subexpresion de tripomastigota respecto a amastigota
Tc00.1047053424383.9	0.009064207	9.46E-08	0.000749616	4.53E-07	Sobreexpresion de tripomastigota respecto a epimastigota
Tc00.1047053425311.9	1.34589E-06	3.96E-14	0.005612117	3.8626E-06	Subexpresion de amastigota respecto a epimastigota
Tc00.1047053425311.9	1.34589E-06	3.96E-14	0.005612117	1.18E-13	Subexpresion de tripomastigota respecto a epimastigota
Tc00.1047053425435.10	4.79E-09	2.80E-15	8.51E-38	1.53E-08	Subexpresion de amastigota respecto a epimastigota
Tc00.1047053425435.10	4.79E-09	2.80E-15	8.51E-38	2.80E-15	Sobreexpresion de tripomastigota respecto a epimastigota
Tc00.1047053425435.10	4.79E-09	2.80E-15	8.51E-38	8.51E-38	Sobreexpresion de tripomastigota respecto a amastigota
Tc00.1047053425851.9	0.00011835	0.008611293	0.000525726	0.00048399	Sobreexpresion de amastigota respecto a epimastigota
Tc00.1047053426397.9	8.03134E-05	0.005417357	1.07E-09	0.00026187	Sobreexpresion de amastigota respecto a epimastigota
Tc00.1047053426397.9	8.03134E-05	0.005417357	1.07E-09	2.69E-09	Subexpresion de tripomastigota respecto a amastigota
Tc00.1047053427247.10	0.000127935	5.05E-13	0.000492679	1.82E-12	Subexpresion de tripomastigota respecto a epimastigota
Tc00.1047053508125.22	0.03083364	1.29812E-05	2.99E-07	7.0269E-05	Sobreexpresion de tripomastigota respecto a epimastigota
Tc00.1047053508125.22	0.03083364	1.29812E-05	2.99E-07	1.5426E-06	Sobreexpresion de tripomastigota respecto a amastigota
Tc00.1047053506655.20	8.30192E-06	8.23E-47	8.78E-20	0.00010176	Subexpresion de amastigota respecto a epimastigota
Tc00.1047053506655.20	8.30192E-06	8.23E-47	8.78E-20	8.23E-47	Subexpresion de tripomastigota respecto a epimastigota
Tc00.1047053506655.20	8.30192E-06	8.23E-47	8.78E-20	8.78E-20	Subexpresion de tripomastigota respecto a amastigota
Tc00.1047053509013.10	0.000278169	7.05E-22	8.29E-35	1.56E-21	Subexpresion de tripomastigota respecto a epimastigota
Tc00.1047053509013.10	0.000278169	7.05E-22	8.29E-35	1.62E-34	Subexpresion de tripomastigota respecto a amastigota

Tabla T.4.4: salida de nuestro algoritmo para los datos de entrada de prueba

Teniendo estas consideraciones en mente se pueden comparar ambas salidas ya que resultan equivalentes. De esta comparación surge que el gen Tc00.1047053397937.10 es reportado por IDEG6 pero no por nuestro algoritmo. Esto sucede porque la probabilidad calculada por nuestro algoritmo es mayor que el p-valor de 0,001. Otra diferencia surge con el gen Tc00.1047053397937.5. Si bien este gen es reportado por ambos algoritmos, para IDEG6 se da subexpresión de trypomastigota respecto los otros dos

estadios mientras que nuestro algoritmo sólo reporta la subexpresión de trypomastigota respecto a amastigota. Estas diferencias, que ningún caso son contradictorias, se deben a que la implementación del test hecha por Romualdi no es igual a la hecha por nosotros y depende de qué se compara con el p-valor. En nuestro caso se trata de la suma de las probabilidades extremas pero en el caso de Romualdi es probable que se haya usado un valor promedio de esas probabilidades. El resto de los genes es reportado de igual manera por ambos algoritmos para el set de datos de prueba.

La ejecución de nuestro algoritmo sobre los 9946 genes da como diferencialmente expresados unos 3686, mientras que IDEG6 reporta 5015 genes para el mismo conjunto de entrada. Los genes reportados por nuestro algoritmo son un subconjunto de aquellos reportados por IDEG6.

4. Análisis de enriquecimiento de términos de ontologías de genes en los grupos con expresión diferencial

La ontología de genes (GO) es un intento de proporcionar un registro uniforme de todos los genes conocidos y sus funciones mediante la anotación de los genes a determinadas categorías de ontología de genes (o términos GO). Al combinar los términos GO se obtiene un grafo directo, acíclico que muestra las relaciones entre los términos. Si un gen anota para un determinado término GO, entonces el gen también anota para sus términos padres (que son menos específicos). Por ejemplo, “regulación de la respuesta inmune” es hijo de “respuesta inmune”, por lo que todos los genes que anotan para lo primero, también anotan para lo segundo. [Conesa et al., 2005]

Hay tres subontologías principales de términos GO: proceso biológico, función molecular y componente celular. La primera subontología, proceso biológico, describe un resultado biológico que por lo general implica una transformación química o física. Algunos ejemplos de los términos que se incluyen en el proceso biológico corresponden al crecimiento y mantenimiento celular. La siguiente subontología, función molecular, describe el componente químico de un proceso biológico, tal como los enlaces específicos a ligandos. Un ejemplo de esta subontología es la actividad catalítica. La última subontología, componente celular, describe dónde tiene lugar el proceso biológico. Los ejemplos incluyen el ribosoma y la membrana nuclear.

La herramienta de anotación de genes basada en ontologías, Blast2GO ofrece la posibilidad de realizar análisis estadísticos sobre la información funcional de los genes. Uno de los análisis es una valoración estadística del enriquecimiento de los términos GO en un grupo de genes de interés al ser comparados con un

grupo de referencia con la finalidad de evaluar las diferencias funcionales. Este análisis se lleva a cabo mediante el test exacto de Fisher combinando con una corrección de FDR[7] (*false discovery rate*). Como resultado se obtiene una lista de términos GO enriquecidos en el grupo que se está testeando y cuya diferencia con el grupo control es estadísticamente significativa. Es posible visualizar gráficamente los resultados, por ejemplo mediante una gráfica de barras. Esta representación muestra, para cada término GO, el porcentaje de secuencias anotadas para ese término. En el eje vertical aparecen los términos GO y en el horizontal, su frecuencia relativa. Las barras azules corresponden a las secuencias del grupo a testear y las rojas, al grupo de referencia (ver figuras 4.5 y 4.9).

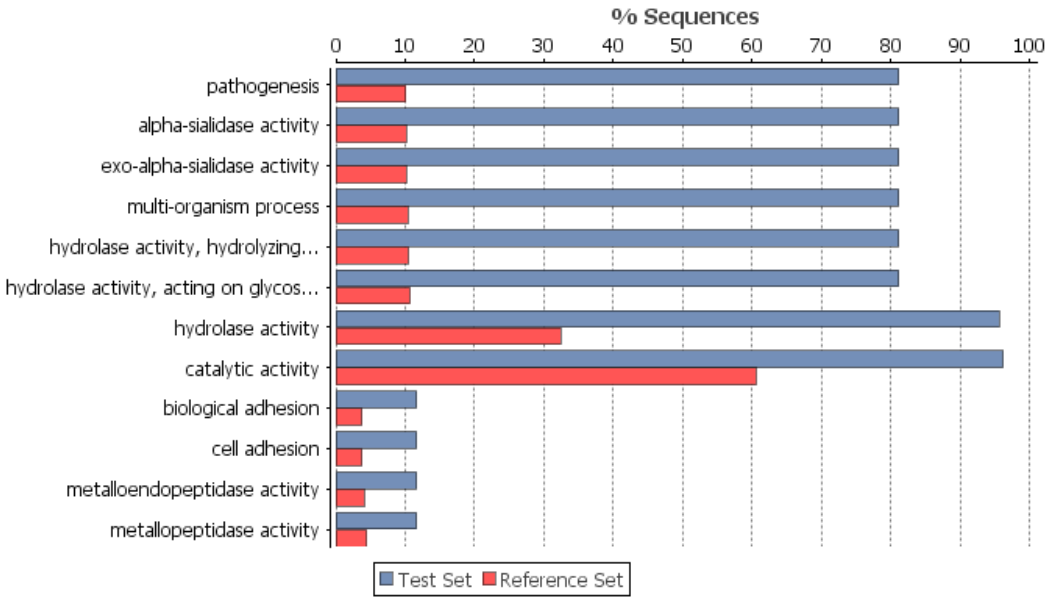


Figura 4.5: términos GO para el análisis de enriquecimiento hecho con el conjunto de genes que se sobre expresan en el estadio trypomastigota respecto a los otros dos estadios

Se llevaron a cabo dos análisis de enriquecimiento de términos GO. En un caso, utilizando como grupo de test los genes sobreexpresados en el estadio trypomastigota respecto a los otros estadios (figura 4.5) y en otro, los genes subexpresados (figura 4.9).

[7] FDR es un método estadístico usado en el testeo de múltiples hipótesis para corregir por comparaciones múltiples. En una lista de hallazgos estadísticamente significativos, FDR es usado para controlar la proporción esperada de rechazos incorrectos a la hipótesis nula.

GO Term	Name	Type	FDR	single test p-Value	# in test group	# in reference group	# non annot test	# non annot reference group	Over/ Under
GO:0009405	pathogenesis	P	1,1E-103	5,0E-107	146	985	34	8760	over
GO:0016997	alpha-sialidase activity	F	1,1E-103	8,6E-107	146	989	34	8756	over
GO:0004308	exo-alpha-sialidase activity	F	1,1E-103	8,6E-107	146	989	34	8756	over
GO:0051704	multi-organism process	P	1,7E-102	1,8E-105	146	1012	34	8733	over
GO:0004553	hydrolase activity, hydrolyzing O-glycosyl compounds	F	3,0E-102	3,9E-105	146	1018	34	8727	over
GO:0016798	hydrolase activity, acting on glycosyl bonds	F	4,8E-101	7,6E-104	146	1041	34	8704	over
GO:0016787	hydrolase activity	F	6,6E-69	1,2E-71	172	3156	8	6589	over
GO:0003824	catalytic activity	F	5,3E-26	1,1E-28	173	5900	7	3845	over
GO:0005488	binding	F	3,9E-24	9,3E-27	30	5434	150	4311	under
GO:0044237	cellular metabolic process	P	1,5E-20	3,9E-23	9	3563	171	6182	under
GO:0097159	organic cyclic compound binding	F	3,3E-18	1,0E-20	8	3222	172	6523	under
GO:1901363	heterocyclic compound binding	F	3,3E-18	1,0E-20	8	3222	172	6523	under
GO:0009987	cellular process	P	1,9E-17	6,6E-20	32	4965	148	4780	under

Figura 4.6: parte de la salida del análisis de enriquecimiento para el conjunto de genes que se sobreexpresan en el estadio trypomastigota respecto a los otros dos estadios

En la figura 4.6 se muestra un fragmento de la tabla de salida del análisis de enriquecimiento donde aparecen, resaltados en rojo, los términos sobre representados y en verde, los sub representados. Los términos resaltados en rojo comparten 146 de los genes que tienen asociados y que corresponden, casi en su totalidad, a trans-sialidasas. Estos términos GO, que figuran como patogénesis, actividad alfa-sialidasa, actividad hidrolasa, etc., presentan vinculaciones, por ejemplo la que se muestra en la figura 4.7, donde se puede ver que la actividad exo-alfa-sialidasa es una actividad alfa-sialidasa, que a su vez es una actividad hidrolasa, etc.

Hay otro grupo de genes (21 en total) que es compartido por varios términos GO (ver figura 4.8) donde aparece la actividad metalloendopeptidasa y metallopeptidasa (función que ha sido mencionada anteriormente como GP63).

La observación de que la sialidasa y GP63 resultan sobre representados en el análisis de enriquecimiento, ratifica la afirmación hecha anteriormente: en el estadio trypomastigota se sobreexpresan genes que codifican para proteínas de membrana y además la afirmación de Barret que las sialidasas son clave para encarar algún tipo de tratamiento farmacológico de la enfermedad de Chagas. [Barret et al., 2003]

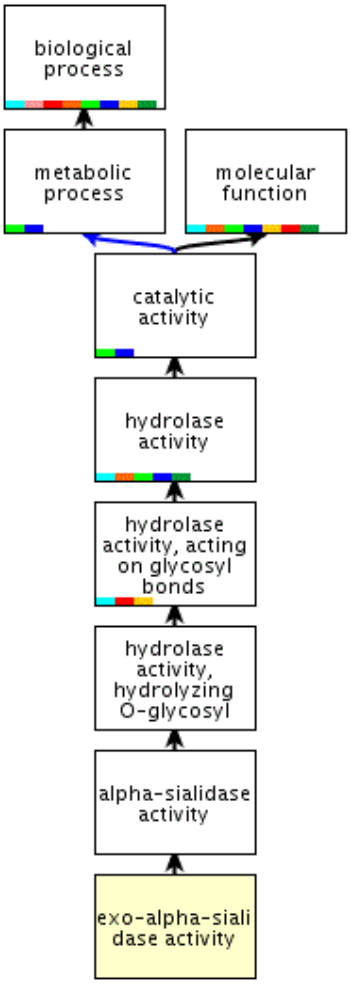


Figura 4.7: vista gráfica de las dependencias del término actividad exo-alfa-sialidasa dado por el browser AmiGO

GO:0022610	biological adhesion	P	3,6E-4	6,4E-6	21	365	159	9380	over
GO:0007155	cell adhesion	P	3,6E-4	6,4E-6	21	365	159	9380	over
GO:0043229	intracellular organelle	C	5,8E-4	1,1E-5	10	1632	170	8113	under
GO:0043226	organelle	C	5,8E-4	1,1E-5	10	1632	170	8113	under
GO:0005737	cytoplasm	C	8,4E-4	1,6E-5	9	1530	171	8215	under
GO:0065007	biological regulation	P	1,1E-3	2,1E-5	1	747	179	8998	under
GO:0016491	oxidoreductase activity	F	1,1E-3	2,2E-5	0	594	180	9151	under
GO:0004222	metalloendopeptidase activity	F	1,2E-3	2,3E-5	21	399	159	9346	over
GO:0016070	RNA metabolic process	P	1,7E-3	3,4E-5	0	579	180	9166	under
GO:0055114	oxidation-reduction process	P	2,5E-3	5,0E-5	0	550	180	9195	under
GO:0090304	nucleic acid metabolic process	P	3,2E-3	6,5E-5	3	892	177	8853	under
GO:0050896	response to stimulus	P	3,4E-3	7,0E-5	1	672	179	9073	under
GO:0016301	kinase activity	F	3,4E-3	7,0E-5	1	673	179	9072	under
GO:0008237	metallopeptidase activity	F	3,4E-3	7,2E-5	21	433	159	9312	over

Figura 4.8: parte de la salida del análisis de enriquecimiento para el conjunto de genes que se sobreexpresan en el estadio trypomastigota respecto a los otros dos estadios

Estos términos, que están sobre representados para el conjunto de genes sobreexpresados en trypomastigota, y que tienen que ver con la capacidad de *T. cruzi* de ser infectivo corresponden al momento del ciclo de vida en que el parásito se encuentra en sangre, que es el estadio en que tiene la capacidad de infectar.

En la figura 4.9 se muestra la gráfica de barras para el análisis de enriquecimiento hecho con el conjunto de genes subexpresados en trypomastigota respecto a los otros dos estadios.

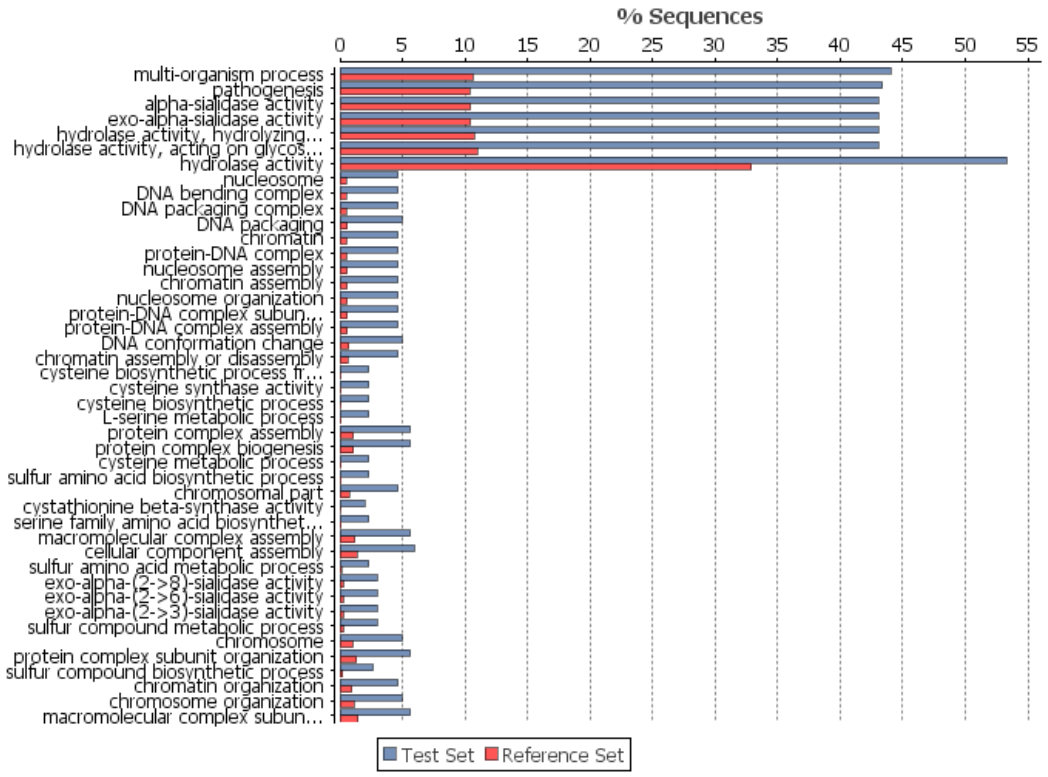


Figura 4.9: términos GO para el análisis de enriquecimiento hecho con el conjunto de genes que se sub expresan en el estadio tripomastigota respecto a los otros dos estadios

Se pude apreciar que se obtienen términos GO que también aparecieron en el análisis hecho con el conjunto de genes sobreexpresados, en particular los vinculados a la actividad sialidasa. Esto significa que existen distintos grupos de genes que, codificando para similar tipo de proteínas, se sobre expresan en el estadio trypomastigota y se subexpresan en los otros.

Por otro lado, figuran términos vinculados con la duplicación del ADN tales como *DNA packaging*, *nucleosome assembly*, *chromatin assembly*, *nucleosome organization*, etc.; que no aparecieron en el análisis de

enriquecimiento correspondiente al conjunto de genes sobreexpresados en trypomastigota. Esto tiene sentido si se considera que *T. cruzi* no se reproduce (no tiene división celular) en el estadio trypomastigota pero sí en los otros estadios. Por esto es esperable que se subexpresen los genes involucrados en la duplicación del ADN en el estadio trypomastigota, o lo que es sinónimo, que se sobreexpresen estos genes en los estadios amastigota y epimastigota.

5. Transcripción en la hebra no codificante

La librería de secuenciación usada en este estudio es hebra específica. Esto se realiza utilizado adaptadores diferentes en los extremos 5'y 3' antes de obtener la primera cadena de cDNA, por lo que luego es posible determinar cuál de las dos hebras de ADN sobre la cual mapean los *reads*, fue usada como molde para la transcripción. El interés en el análisis de la hebra no codificante radica en la posibilidad de hallar indicios de regulación de la expresión génica postranscripcional mediante reguladores *antisense*.

Los NAT's (*natural antisense transcripts*) fueron primeramente descritos en procariotas, donde se encontró que formaban parte de mecanismos generales de control de la expresión génica, regulando a la baja la expresión de los transcriptos *sense*. Sin embargo la existencia de ARN *antisense* en eucariotas, ha sido sugerida anteriormente por varios autores. Los NAT's son transcriptos que tienen secuencias complementarias a transcriptos correspondientes a ARNm y otros tipos de ARN con diversas funciones. La mayoría provienen del mismo locus que los transcriptos *sense* (están en *cis*). Los transcriptos *sense* y *antisense* se solapan al menos parcialmente y presentan complementariedad perfecta. En contraste, los ARN *antisense* en *trans* se originan en diferentes loci y sólo presentan complementariedad parcial con el transcripto *sense*. Los transcriptos *antisense* pueden jugar un rol en la regulación de la expresión de sus contrapartes *sense*. Dada su complementariedad, los transcriptos *antisense* pueden hibridar con los transcriptos *sense* y de esa manera modificar la expresión de estos últimos. [Vanhée-Brossollet y Vaquero,1998]

Con el fin de identificar cuál de las dos hebras fue usada como molde para un *read* particular se realizó el mapeo de *reads* con BLAST y la dirección de matcheo del *read* (en el mismo sentido o inverso de la secuencia) es comparada con el sentido de la hebra codificante sobre esa zona que aparece en la anotación (en el archivo GFF).

Las regiones de mapeo que se identifican de esta forma fueron luego analizadas en detalle utilizando la herramienta Artemis, que es un browser genómico [Rutherford et al., 2000]. Se visualizó la hebra no codificante y los *reads* que alinean en ella y que son exclusivos de esa hebra, o sea, que no

reportan coincidencias con la hebra codificante. Se analizaron los primeros 18 cromosomas y se detectaron algunos patrones para genes codificantes de proteínas MASPs[8] y mucinas en la distribución de los *reads* y en menor medida para RHS[9]. No se hizo extensivo el análisis al resto de los cromosomas porque los patrones encontrados en los 18 primeros se repiten en cada uno de ellos y no era previsible arribar a un hallazgo diferente con un análisis exhaustivo de los cromosomas restantes. Cabe destacar que, si bien resulta interesante el análisis de estos patrones por estadio, los resultados que se presentan a continuación corresponden a los *reads* de los tres estadios. Esto debió ser hecho así para poder identificar los patrones claramente, ya al analizar por estadio se desdibujan las regiones que fueron identificadas como patrones. Contando con más datos y además sabiendo de antemano dónde se hallan los patrones, resulta interesante repetir este análisis a futuro. Por otro lado, en cuanto a la distribución global de los *reads* a lo largo de cada cromosoma no se pudo detectar nada significativo.

A continuación se presenta una descripción de algunos de los genes que presentaron patrones más interesantes.
Las proteínas MASP presentan un patrón de distribución de *reads* en las regiones 5' y 3' UTR como se aprecia en la figura 4.10 (a). Sobre el extremo 5' se acumulan muchos *reads* en forma de columna (ver detalle en la figura 4.10 (b)). Esto significa que los *reads* coinciden en una región muy acotada de la referencia, lo que a su vez conlleva a que se produzcan muchos solapamientos de esos *reads*.

En el otro extremo, hacia el 3', pero por fuera del gen, existe un grupo de *reads* según se ve a la derecha en la figura 4.10 (a), pudiéndose apreciar con más detalle en la figura 4.10 (c). También se puede notar un pequeño grupo de *reads* que alinean sobre el extremo 3' por dentro del gen. Este grupo no está presente para todas las MASPs de los 18 cromosomas analizados, por lo que en principio, no reviste el interés de los otros grupos, cuya presencia sí es una constante para todas MASPs visualizadas con Artemis.

[8] MASP, acrónimo en inglés para mucina asociada a proteína de membrana (*mucin associated surface protein*)

[9] RHS: *retrotransposon hot spot*

Figura 4.10 (a): Patrón de distribución de *reads* para una MASP

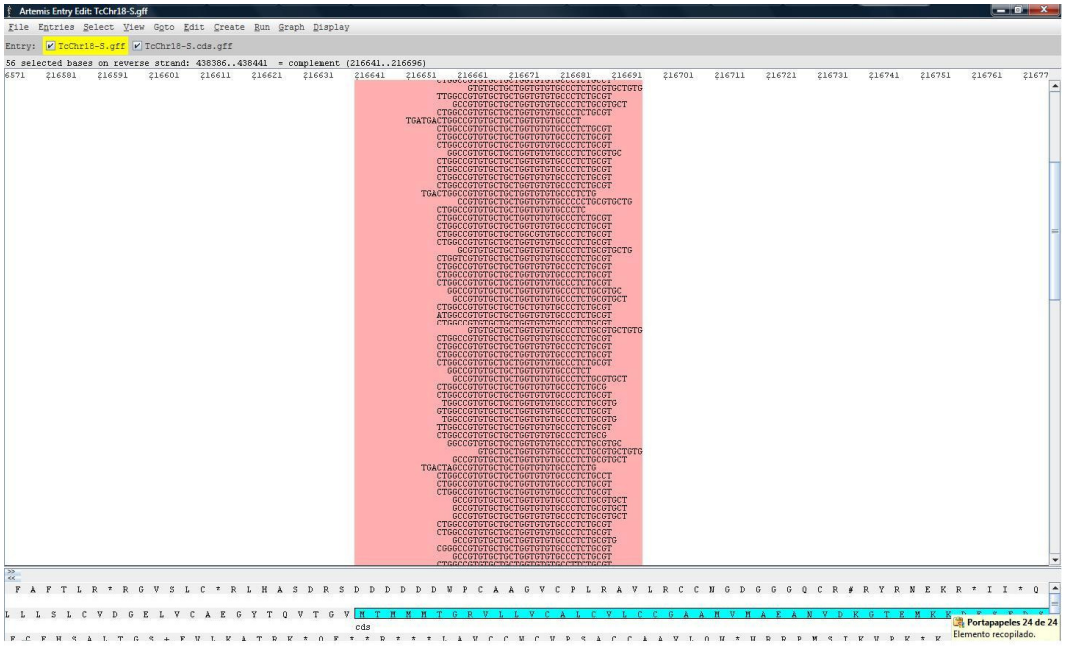


Figura 4.10 (b): Detalle de la región 5' de una MASP, marcado con un círculo en la izquierda de la figura 4.10 (a)

Otro de los grupos de genes cuyos patrones de distribución espacial en la transcripción de la hebra no codificante es peculiar, son los que codifican para las proteínas MucII. Estos genes también presentan un patrón de *reads* que

mapean sobre el extremo 5', existiendo un grupo dentro del gen (de pocos *reads*) y otro fuera (con mayor cantidad de *reads*). Ver figura 4.11 (a), (b) y (c).

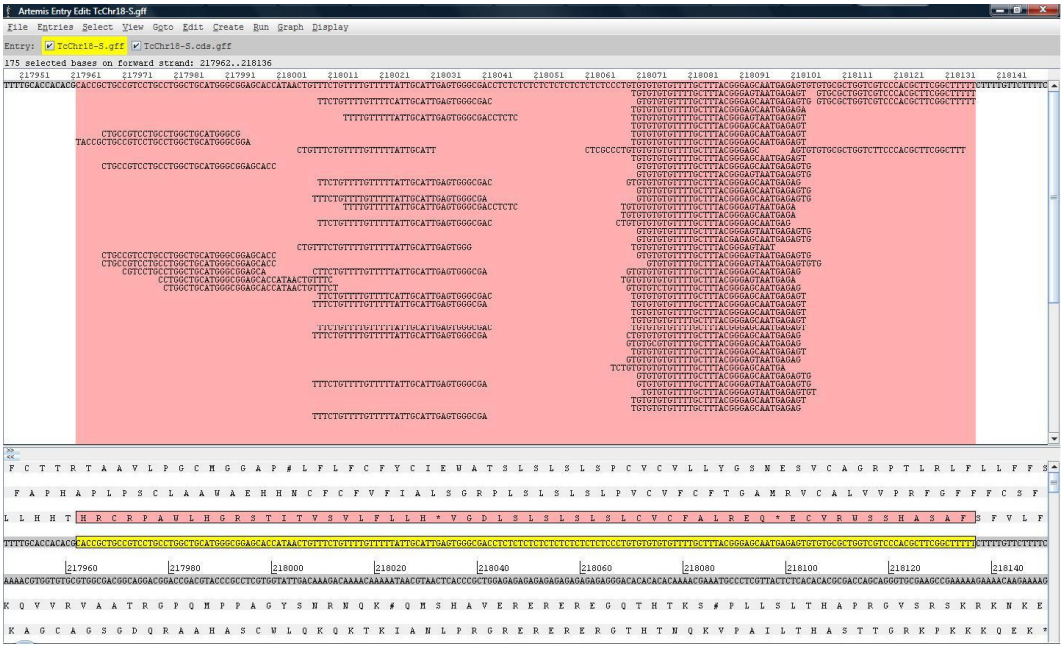


Figura 4.10 (c): Detalle de la región cercana al extremo 3' de una MASP, marcado con un círculo en la derecha de la figura 4.10 (a)

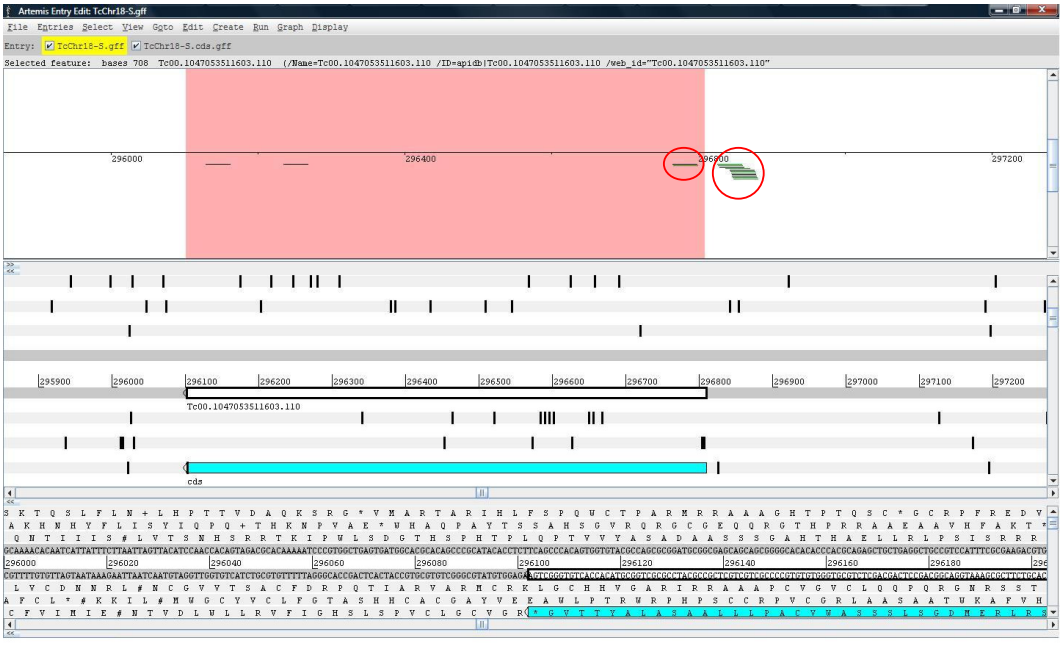


Figura 4.11 (a): Patrón de distribución de *reads* para una MucII

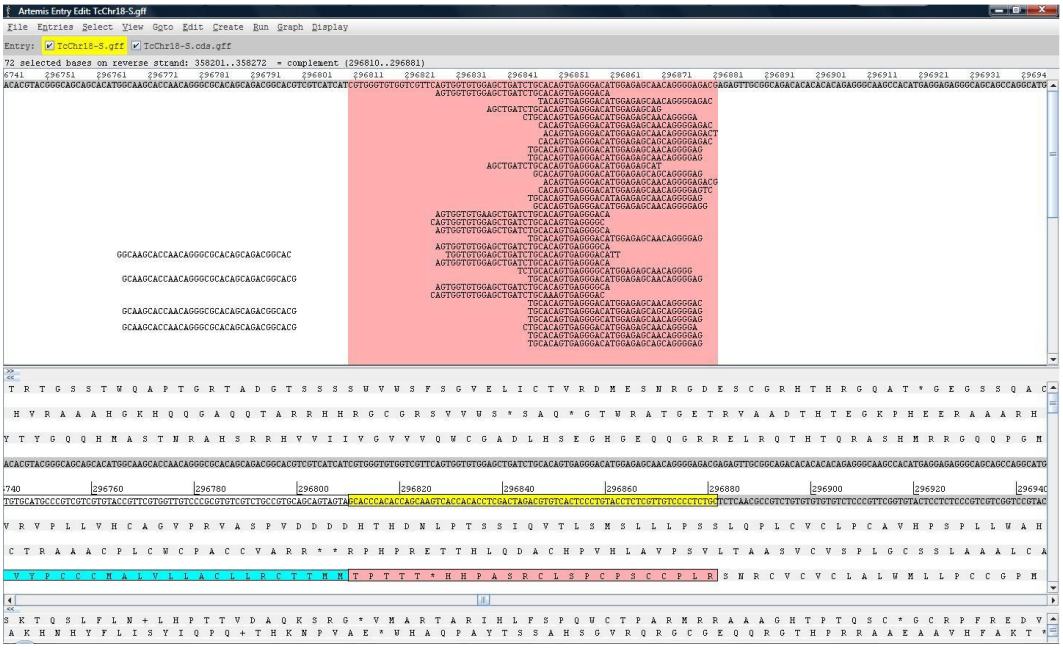


Figura 4.11 (b): Detalle de los *reads* cercanos a la región 5' fuera de la MucII, marcado con el círculo de la derecha en la figura 4.11 (a)

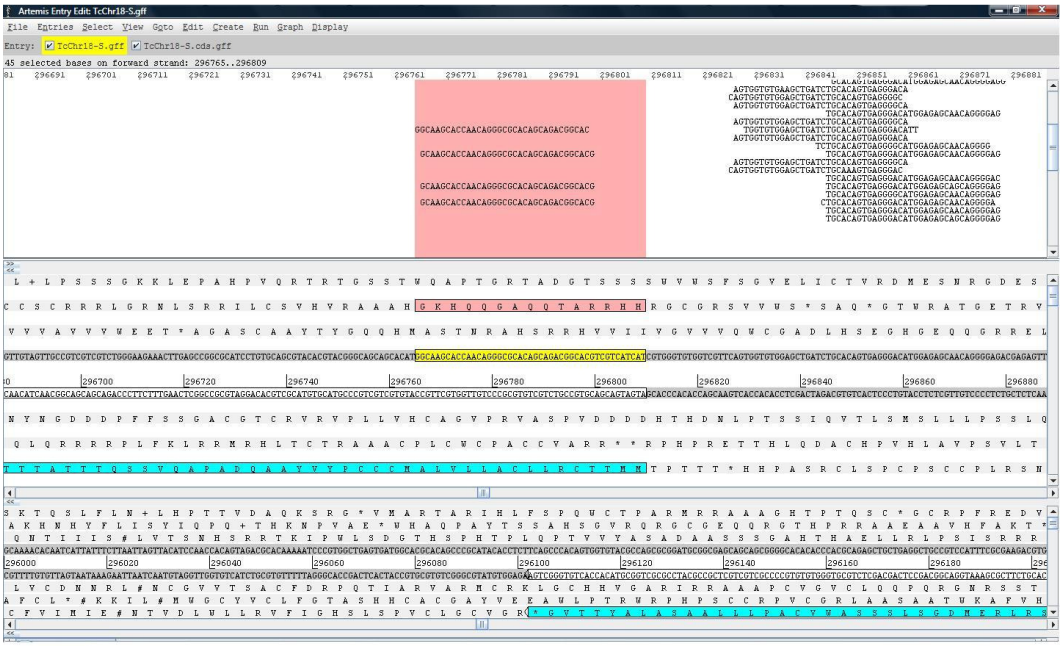


Figura 4.11 (c): Detalle de los *reads* en la región 5' de la MucII, marcado con el círculo de la ozquierda en la figura 4.11 (a)

El último grupo de genes que se describirá corresponde al codificante de las RHS. Estos presentan un patrón de *reads* cercano al extremo 5' (que dista algo más de 1000 bp) y por fuera del gen (figura 4.12 (a) y detalle en figura 4.12

(b)). En este caso el patrón aparece a una distancia considerable del extremo del gen por lo que no resulta tan directo pensar en una influencia de esta región sobre la expresión de este gen.

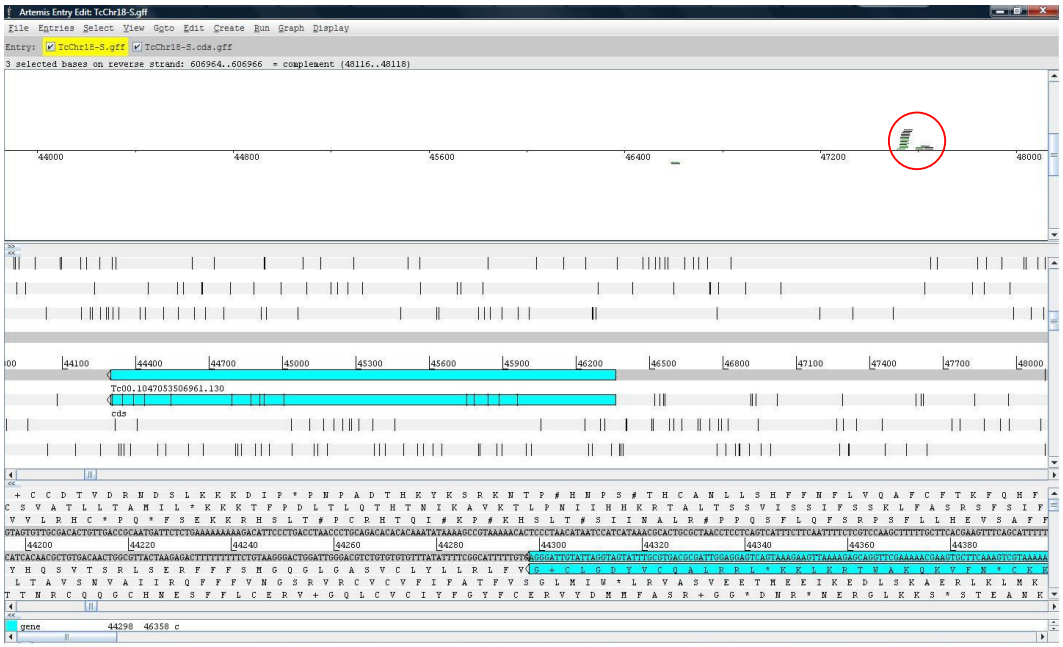


Figura 4.12 (a): Patrón de distribución de *reads* para un RHS

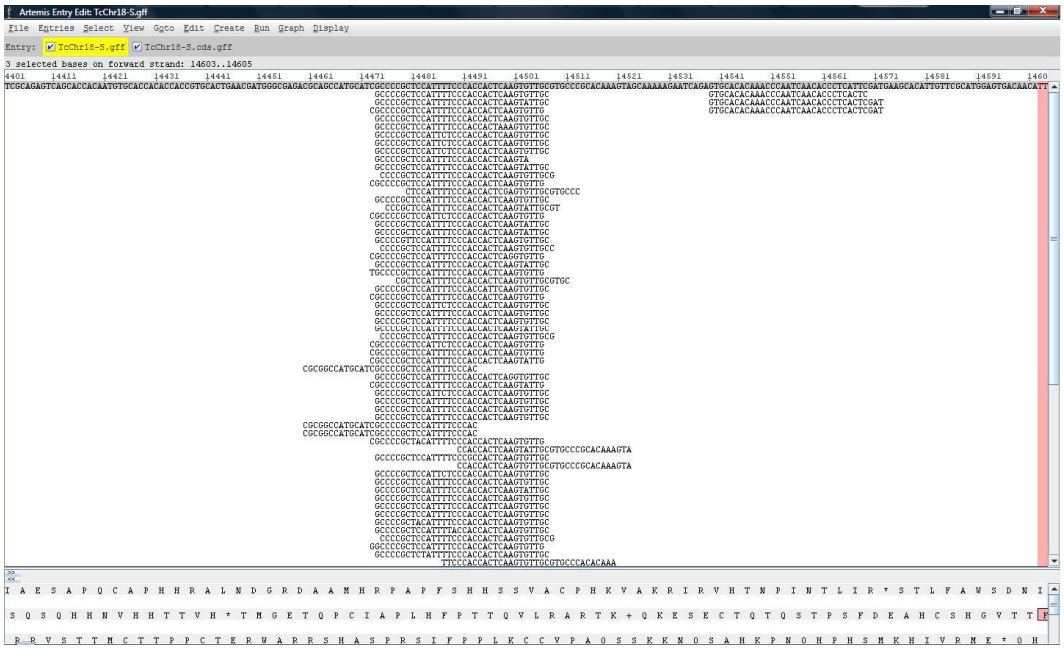


Figura 4.12 (b): Detalle de los *reads* cercanos a la región 5' del RHS, marcados con un círculo en la figura 4.12 (a)

Para analizar estos patrones en más detalle se extrajeron las zonas de los genes que resultaron de interés para determinar si se trata de regiones conservadas. Se analizaron las regiones 3' y 5' para las MASP y MucII en busca de motivos, o sea, secuencias que están muy representadas en el conjunto. Entonces, para las MASPs, se extrajo a partir del extremo 5', 100 bp por fuera y 100 bp por dentro del gen y para su extremo 3' el rango fue de 100 bp por dentro y 400 bp por fuera. Para las MucII se extrajo la región aledaña al extremo 5', tomando 100 bp por fuera y 100 bp por dentro del gen. De esta manera se obtienen las regiones donde matchearon los *reads* con la hebra no codificante, con cierto margen adicional. Para determinar el nivel de conservación de estas regiones se realizó un alineamiento con clustalw y se pudo notar la existencia de grupos en el alineamiento por lo que se hizo una división resultando 6 subgrupos para las MASPs en el extremo 3', 5 para su extremo 5' y 2 subgrupos para MucII en su extremo 5'. Cabe señal aquí que se descartó un subgrupo por cada familia de proteínas por resultar *outsider*, ya que no presentaba coherencia interna en cuanto a sus elementos por tratarse de secuencias muy diferentes. Se separaron las secuencias de las regiones según los subgrupos mencionados previamente y se volvieron a alinear. Esos alineamientos se analizaron con la herramienta JalView [Waterhouse et al., 2009] para determinar el nivel de conservación de cada región por cada subgrupo, obteniéndose secuencias consenso por cada uno. En la figura 4.13 se puede ver un fragmento del logo obtenido con JalView para el subgrupo 1 de MucII. Se puede apreciar que las letras que componen el logo tienen diferente tamaño en relación a la presencia de cada base en una posición dada.

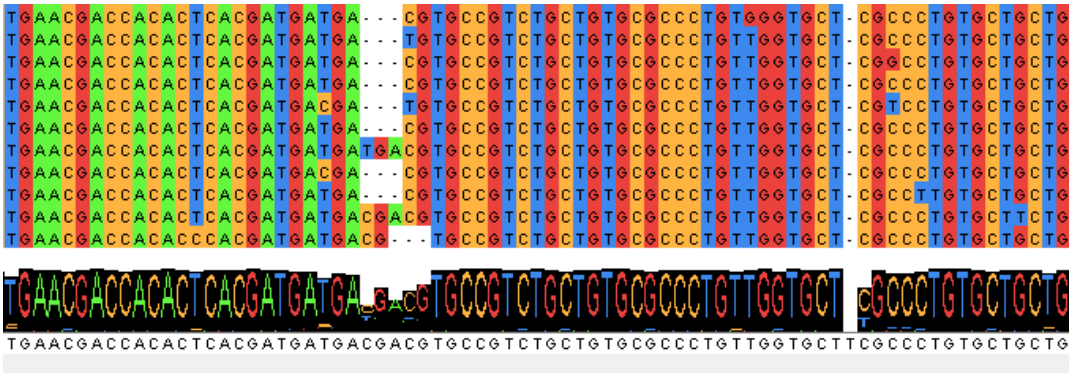


Figura 4.13: Ejemplo de un fragmento de secuencia consenso para el grupo de MucII.

Se obtuvieron secuencias consenso para estas regiones para cada subgrupo. Estos resultados se incluyen como archivos .jvp que pueden ser visualizados usando la versión web de JalView (http://www.jalview.org/node/187). Si bien los consensos presentan cierta variabilidad, puede decirse que son claramente definidos. La excepción la constituye la región 5' de las MucII que está altamente conservada.

Resulta de interés ver cómo alinean los *reads* antisense (hebra no codificante) con las secuencias consenso. Para esto se utilizaron los consensos como referencia para alineamientos que fueron realizados con BLAST, exigiendo un *e-value* de e^{-10}. Hay dos aspectos a analizar a partir de estos resultados. Por un lado, determinar qué tan compartidos resultan estos *reads*, o sea, si los *reads* que alinean para un subgrupo de una proteína, lo hacen también para otros subgrupos de esa misma proteína. Y por otro lado, interesa ver la distribución que presentan esos *reads* al mapear contra los consensos, es decir, si mapean preferentemente en alguna zona en particular.

MASP en su extremo 3' se dividió en 6 subgrupos, según se aprecia en la tabla T.4.5, donde se incluye el número de secuencias que integra cada grupo y la cantidad de *reads* que matchearon en el alineamiento con BLAST.

	Subgrupo 1	Subgrupo 2	Subgrupo 3	Subgrupo 4	Subgrupo 5	Subgrupo 6
Secuencias (genes)	412	16	41	9	12	7
Reads	163	144	102	98	9	66

Tabla T.4.5: Cantidad de secuencias y *reads* que alinean por subgrupo para el extremo 3' de MASP

MASP en su extremo 5', se dividió en 5 subgrupos (ver tabla T.4.6)

	Subgrupo 1	Subgrupo 2	Subgrupo 3	Subgrupo 4	Subgrupo 5
Secuencias (genes)	52	277	124	7	21
Reads	274	220	252	5	23

Tabla T.4.6: Cantidad de secuencias y *reads* que alinean por subgrupo para el extremo 5' de MASP

MucII se dividió en 2 subgrupos según se muestra en la tabla T.4.7.

	Subgrupo 1	Subgrupo 2
Secuencias (genes)	237	35
Reads	112	4

Tabla T.4.7: Cantidad de secuencias y *reads* que alinean por subgrupo para el extremo 5' de MucII

El subgrupo 2 no reviste mayor interés por el bajo número de *reads* que alinean con sus secuencias.

Para visualizar lo concerniente al primer aspecto, o sea, cómo son compartidos los *reads* por los subgrupos, se confeccionaron los diagramas de Venn que se aprecian en las figuras 4.14 y 4.15, para los extremos 3' y 5' de las MASPs respectivamente. No se incluye diagrama para MucII ya que se divide en 2 subgrupos cuya intersección carece de elementos.

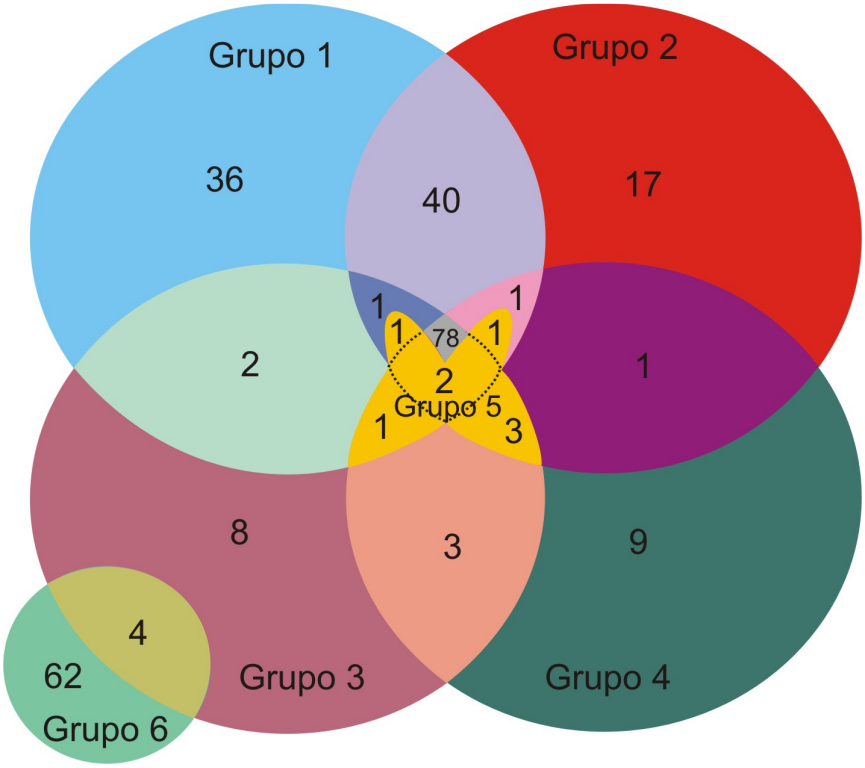


Figura 4.14: Diagrama de Venn para todos los subgrupos correspondientes al extremo 3' de MASP

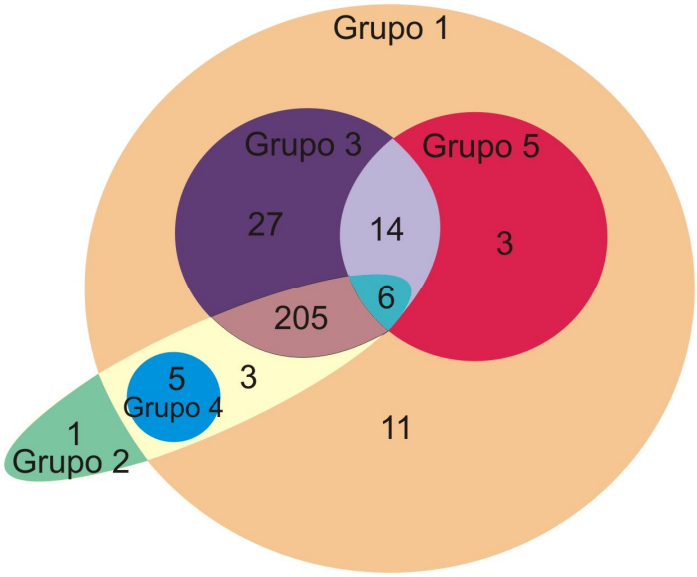


Figura 4.15: Diagrama de Venn para todos los subgrupos correspondientes al extremo 5' de MASP

En cuanto al segundo aspecto, la distribución del mapeo de *reads* (hebra no codificante) con los consensos, se parsearon las salidas de BLAST para obtener la posición inicial, en la referencia, en que alinea cada *read*. Estas posiciones se usaron para obtener los histogramas de la distribución de *reads* para cada subgrupo y proteína. En las figuras 4.16, 4.17 y 4.18 se ven los histogramas correspondientes a un subgrupo por proteína. Los restantes histogramas no se incluyen por presentar características similares o pocos datos para resultar de interés.

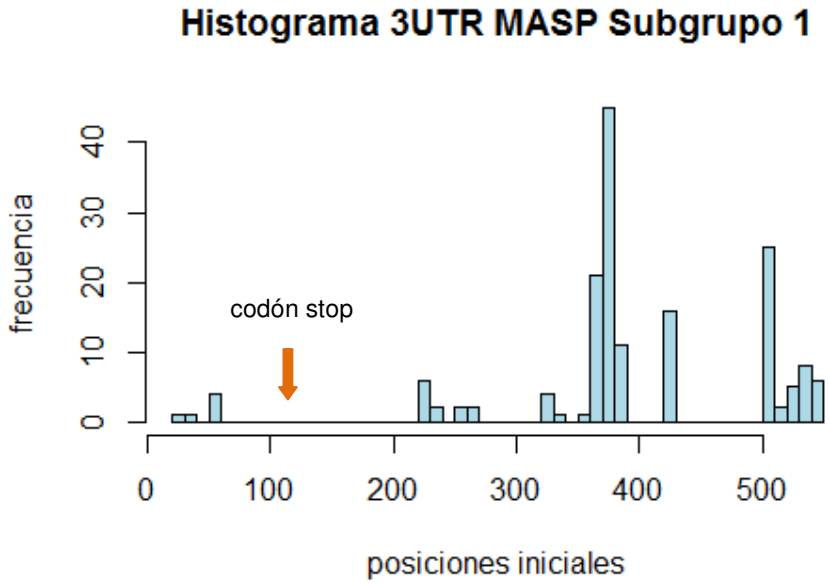


Figura 4.16: Distribución de *reads* para el extremo 3UTR del subgrupo 1 de MASP

En la figura 4.16 correspondiente a la region 3' UTR de MASP, se puede apreciar que la mayor parte de los *reads* mapean sobre el último tercio del consenso, lo que se corresponde con el grupo de *reads* que se puede ver a la derecha en la figura 4.10 (a). Se debe recordar que los fragmentos a partir de los que se obtuvo el consenso tienen largo 500 bases (100 bp dentro del gen y 400 por fuera), por lo que el consenso es un poco más largo, alcanzando las 670 bases en el caso particular de este subgrupo. Considerando esto es posible situar aproximadamente el codón stop pasando la posición 100, como se indica mediante la flecha de la figura 4.16. Una situación análoga es aún más visible para la distribución de la figura 4.17, donde la acumulación de *reads* se da a partir de la segunda mitad del consenso. Esto se corresponde con los muchos *reads* que mapean en forma de columna y que se aprecian a la izquierda en la figura 4.10 (a) en la region 5' UTR del mismo. En este caso el codón de iniciación estaría próximo a la posicón 100, sobre el borde del histograma, considerando que los fragmentos que dieron origen al consenso son de 200 bases de largo y que el consenso en sí tiene 272 bases. Por último, para MucII se puede ver en la figura 4.18 una distribución que, si bien tiene picos, no resulta tan concentrada como para el caso anterior y se corresponde con el mapeo de *reads* alrededor del extremo 5' que puede verse en la figura 4.11 (a) donde hay *reads* que mapean dentro del gen y otros por fuera lo que se aprecia claramente a los lados del codón de iniciación marcado por la flacha que indica una posició cercana a las 100 bases, siendo el consenso de largo 237.

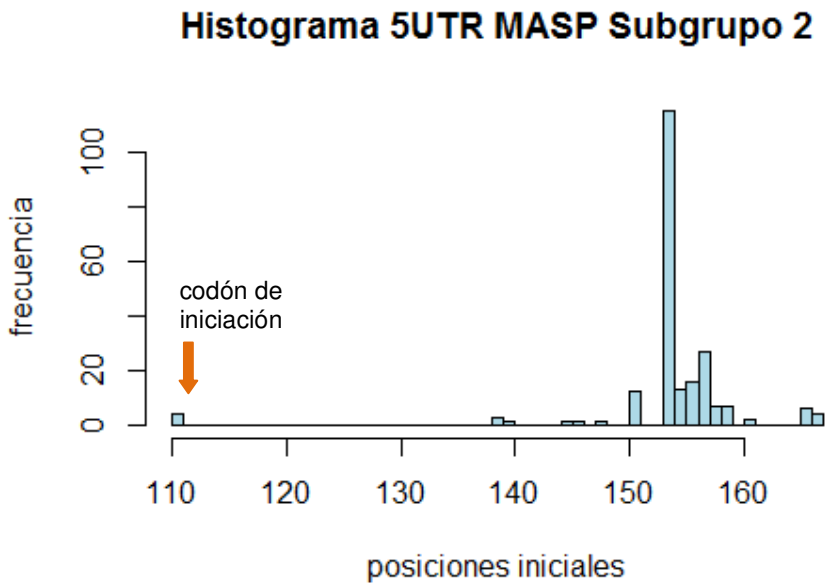


Figura 4.17: Distribución de *reads* para el extremo 5UTR del subgrupo 2 de MASP

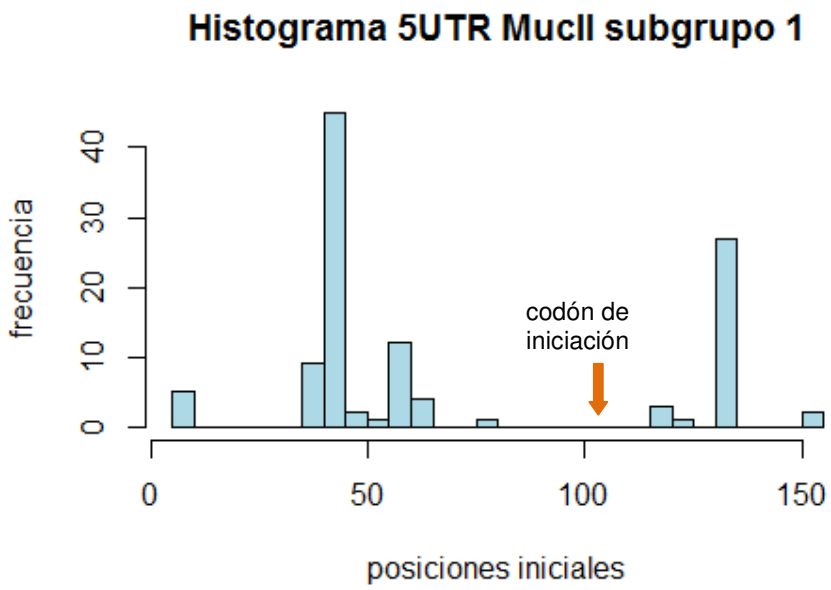


Figura 4.18: Distribución de *reads* para el extremo 5UTR del subgrupo 1 de MucII

Para el caso de la proteína MASP, el subgrupo 6 resulta interesante ya que comparte 4 elementos con el subgrupo 3 y no tiene intersección con ningún otro subgrupo, esto indica que los genes del subgrupo 3 tendrían una modulación independiente del resto de las MASPs.

El hallazgo de estos motivos en los extremos de estos genes indicaría la existencia de algún mecanismo de regulación a nivel postranscripcional.

6. Discusión

En esta tesis se analizan algunos aspectos de la expresión génica durante tres fases del ciclo de vida de *T. cruzi*. Uno de los objetivos de este trabajo fue familiarizarse con este tipo de datos (técnicas de secuenciación), así como las herramientas computacionales y estadísticas para su análisis. Para este fin se usaron datos de secuenciación profunda de RNA (RNAseq). Dichos datos además de permitir abordar la transcripción diferencial de genes (o sobrevida diferencial de los ARNs) son adecuadas para determinar la existencia de transcripción en la hebra no codificante que pudiera actuar como ARNs reguladores *antisense* por el hecho de ser secuenciación hebra específica.

En relación al estudio de la abundancia diferencial de ARNms (“expresión diferencial”) debemos señalar algunos aspectos, que las proteínas de membrana, particularmente las sialidasas, pueden constituir un target para el

tratamiento de la enfermedad de Chagas ya que claramente se sobreexpresan en el estadio infectivo del parásito (trypomastigota). Esto fue confirmado tanto de forma gráfica al comparar los estadios 2 a 2, como por la aplicación del test estadístico de Audic y Claverie; y por el análisis de enriquecimiento de genes hecho con blast2go. Otras proteínas de membrana como GP63 y, en menor medida, MASP y MucII también pueden considerarse de interés a efectos de ser objeto para el tratamiento de la enfermedad, ya que algunos de nuestros análisis las reportan como sobreexpresadas en el estadio trypomastigota, pudiendo requerir un análisis más profundo y centrado en ellas para arribar a evidencia más concluyente. En este sentido es importante recordar que los fármacos existentes para tratar la enfermedad de Chagas son eficaces en la etapa aguda de la enfermedad pero tienen contraindicaciones de importancia que son bastante frecuentes. Durante esta etapa el parásito se encuentra en sangre y los esfuerzos para combatirlo de forma más efectiva pueden enfocarse en las proteínas de membrana que se sobreexpresan en esta etapa. Estos hallazgos apuntan en el mismo sentido que lo concluido por otros investigadores [Barret et al., 2003] al respecto de las sialidasas como posible blanco sobre el que centrar los esfuerzos para tratar la enfermedad de Chagas y también sobe la importancia de las otras proteínas de membrana por hallarse muy repetidas en el genoma. [Gómez y Monteón, 2008]

Respecto al análisis realizado sobre la transcripción de la hebra no codificante cabe destacar que los patrones en que los *reads* mapean con ella sobre MASP, MucII y RHS así como los patrones de conservación de la regiones 5' y 3' de los mismos, llevan a pensar en la posible existencia de algún tipo de regulación postranscripcional basado en un mecanismo regulación tipo *antisense*.

La posibilidad de regulación postranscripcional en tripanosomátidos en general y en *T. cruzi* en particular, ha sido sugerida anteriormente por varios autores [Andersson et al., 1998] [Brandão y Jiang, 2009] [Gómez y Monteón, 2008]. Por este motivo resulta de interés profundizar en este análisis como parte de un trabajo a futuro, contando con mayor cantidad de datos, ya que la evidencia de patrones tan claros lleva a pensar que el proceso implicado tras ellos está lejos de ser azaroso. Por otro lado, la posibilidad de que exista regulación a nivel postranscripcional de la expresión de los genes que codifican algunas proteínas de membrana, a efectos de combatir la enfermedad, apunta a desarrollos que pueden ser aplicados en la etapa crónica de la enfermedad, al igual que los fármacos actualmente en uso.

Anexo I

Función Γ (gamma)

Si $s > 0$, la integral $\int_{0^+}^{\infty} e^{-t} t^{s-1}$ converge y se denota por $\Gamma(s)$. Tiene algunas propiedades de interés que se enuncian a continuación.

1. $\Gamma(1) = 1$
2. $\Gamma(n+1) = n\Gamma(n)$
3. $\forall\, n \in \mathbb{N}, \Gamma(n+1) = n!$
4. $\Gamma(n)\Gamma\left(n + \frac{1}{2}\right) = 2^{1-2z}\sqrt{\pi}\Gamma(2n)$
5. $\Gamma(1/2) = \sqrt{\pi}$
6. $\Gamma(1-n)\Gamma(n) = \frac{\pi}{\sin(\pi n)}$

La propiedad utilizada por Audic y Claverie [Audic y Claverie, 1997] es la 3. A continuación se probará esa propiedad, para lo que es necesario probara además las dos anteriores.

Demostración 1.

$$\Gamma(1) = \int_{0^+}^{\infty} dt\; e^{-t} t^{1-1} = \int_{0^+}^{\infty} dt\; e^{-t} t^{0} = \int_{0^+}^{\infty} dt\; e^{-t} = -e^{-t}|_{0^+}^{\infty} = -e^{-\infty} + e^{0} = 1$$

Demostración 2.

$$\Gamma(n+1) = \int_{0^+}^{\infty} dt\; e^{-t} t^{n+1-1} = \int_{0^+}^{\infty} dt\; e^{-t} t^{n}$$

Aplicando integración por partes: $\int_a^b u\, dv = uv|_a^b - \int_a^b v\, du$

Elegimos u y dv:

$$u = t^n \;\Rightarrow du = n\, t^{n-1} dt$$
$$dv = e^{-t} dt \;\Rightarrow v = -e^{-t}$$

Sustituyendo:

$$\Gamma(n+1) = t^n(-e^{-t})\,|_{0^+}^{\infty} - \int_{0^+}^{\infty}(-e^{-t})\, n\; t^{n-1} dt = \lim_{t\to\infty} -t^n\, e^{-t} + n \int_{0^+}^{\infty} e^{-t}\, t^{n-1} dt$$

El primer término es una indeterminación del tipo $\frac{\infty}{\infty}$ donde el denominador es de mayor orden que el numerador por lo que el límite tiende a 0. El segundo término corresponde a $\Gamma(n)$. Entonces:

$$\Gamma(n+1) = n\,\Gamma(n)$$

Demostración 3.

Se demostrará por inducción.
Por la propiedad 1 se tiene que $\Gamma(1) = 1$
Asumimos que se cumple para k, entero mayor que 0: $\Gamma(k) = (k-1)!$
Si multiplicamos por k a ambos lados de la ecuación anterior, se tiene:

$$k\,\Gamma(k) = k(k-1)! = k!$$

Por la propiedad 2 tenemos que $k\,\Gamma(k) = \Gamma(k+1)$

Entonces: $\Gamma(k+1) = k!$

Anexo II

Código implementado para el cálculo del test de Audic y Claverie y para la determinación de la expresión diferencial de genes.

```
#!/usr/bin/env python
# coding=utf-8

#Modo de uso: python ac.py <archivo_de_entrada> <p-valor> <nombre_archivo_salida>
#Script para calcular test de Audic y Claverie
#Input: archivo con los conteos de reads
#Output: archivo con el cálculo del test, se debe pasar el nombre al invocar el script

import sys
import math, decimal, time

arch = open(sys.argv[1], 'r')
pValor =float(sys.argv[2])
out = open(sys.argv[3], 'w')
primeraLinea=0

def ac(n1,n2,X,Y):
	N2N1=n2/n1
	acPN=decimal.Decimal(math.pow(N2N1,Y/12)) #(N2/N1)^y/12
	acFN=decimal.Decimal(math.factorial(X+Y)) #(x+y)!
	acFD=math.factorial(X)*math.factorial(Y) #x!y!
	acPDx=decimal.Decimal(math.pow(1+N2N1,X/15)) #(1+N2/N1)^x/15
	acPDz=decimal.Decimal(math.pow(1+N2N1,Y/25)) #(1+N2/N1)^y/25
	acI=acFN/acFD
	acII=decimal.Decimal(acI/(acPDx*acPDx*acPDx*acPDx*acPDx*acPDx*acPDx*acPDx*
	acPDx*acPDx*acPDx*acPDx*acPDx*acPDx*acPDx))
	acIII=decimal.Decimal(acII/(acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz
	*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acPDz*acP
	Dz*acPDz*acPDz*acPDz*acPDz*acPDz)/decimal.Decimal(1+N2N1))
	acVI=decimal.Decimal(decimal.Decimal(acPN*acPN*acPN*acPN*acPN*acPN*acPN*ac
	PN*acPN*acPN*acPN*acPN*acIII))
	return acVI

def imprimoLinea(Gen, Desc,n1,n2,n3,X,Y,Z,acum):
	out.write(Gen)
	out.write('\t')
	out.write(Desc)
	out.write('\t')
	out.write(str(ac(n1,n2,X,Y)))
	out.write('\t\t')
	out.write(str(ac(n1,n3,X,Z)))
	out.write('\t\t')
	out.write(str(ac(n2,n3,Y,Z)))
	out.write('\t\t')
	out.write(str(acum))
```

```
top=500000      #topea la cantidad de iteraciones cuando se calcula una integral con límite
infinito
bottom=0.00000000001 #límite en el valor de los términos de una suma
out.write('\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t AC 1-2 \t\t\t\t\t\t\t AC 1-3 \t\t\t\t\t\t\t AC 2-3 \n')
for line in arch:
        if primeraLinea==0:
                N1=int(line.split()[0])
                N2=int(line.split()[1])
                N3=int(line.split()[2])
                primeraLinea=1
        else:
                gen=line.split()[0]
                desc=line.split()[1]
                x=int(line.split()[2])
                y=int(line.split()[3])
                z=int(line.split()[4])
                #Normalización por el éxito del experimento
                xn=x/N1
                yn=y/N2
                zn=z/N3

                pLimiteXY=ac(N1,N2,x,y)
                pLimiteXZ=ac(N1,N3,x,z)
                pLimiteYZ=ac(N2,N3,y,z)
                if (x==y):
                        acum0=0
                        acumInf=0
                        for n in range(0,y+1):
                                acum0=acum0+ac(N1,N2,x,n)
                        m=y
                        while m<top:
                                acTest=ac(N1,N2,x,m)
                                acumInf=acumInf+acTest
                                if acTest<bottom:
                                        m=top
                                m+=1
                        if acum0<acumInf:
                                min=acum0
                        else:
                                min=acumInf
                        if (xn>yn):
                                if (min<pValor): #subexpresion de amastigota respecto a
                                epimastigota = sobreexpresion de epi respecto de ama
                                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                                        out.write ('\t\t Subexpresion de amastigota respecto a
                                        epimastigota \n')
                        else:
                                if (min<pValor): #sobreexpresión de amastigota respecto a
                                epimastigota = subexpresión de epi respecto a ama
                                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                                        out.write ('\t\t Sobreexpresion de amastigota respecto a
                                        epimastigota \n')
                elif (x>y):
```

```
                acum0=0
                acumInf=0
                for n in range(0,y+1):
                        acum0=acum0+ac(N1,N2,x,n)
                t=x+1
                while t<top:
                        pControl=ac(N1,N2,x,t)
                        if (pControl<=pLimiteXY):
                                nuevoY=t #tambien puede ser nuevoY=t-1
                                t=top
                        t+=1
                m=nuevoY
                while m<top:
                        acTest=ac(N1,N2,x,m)
                        acumInf=acumInf+acTest
                        if acTest<bottom or acumInf+acum0>pValor:
                                m=top
                        m+=1
                if (acum0+acumInf<pValor): #subexpresion de amastigota respecto a
                epimastigota = sobreexpresion de epi respecto de ama
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
                        if (xn>yn):
                                out.write ('\t\t Subexpresion de amastigota respecto a
                                epimastigota \n')
                        else:
                                out.write ('\t\t Sobreexpresion de amastigota respecto a
                                epimastigota \n')
        else: #x<=y
                acum0=0
                acumInf=0
                w=0
                while w<x:
                        acAux=ac(N1,N2,w,y)
                        if acAux>=pLimiteXY:
                                w=x+2
                        else:
                                acum0=acum0+acAux
                        w+=1
                m=y
                while m<top:
                        acTest=ac(N1,N2,x,m)
                        acumInf=acumInf+acTest
                        if acTest<bottom or acumInf+acum0>pValor:
                                m=top
                        m+=1
                if (acumInf+acum0<pValor): #sobreexpresión de amastigota respecto a
                epimastigota = subexpresión de epi respecto a ama
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
                        if (xn<yn):
                                out.write ('\t\t Sobreexpresion de amastigota respecto a
                                epimastigota \n')
                        else:
```

```
                        out.write ('\t\t Subexpresion de amastigota respecto a
                        epimastigota \n')

if (x==z):
        acum0=0
        acumInf=0
        for n in range(0,z+1):
                        acum0=acum0+ac(N1,N3,x,n)
        m=z
        while m<top:
                acTest=ac(N1,N3,x,m)
                acumInf=acumInf+acTest
                if acTest<bottom:
                        m=top
                m+=1
        if acum0<acumInf:
                min=acum0
        else:
                min=acumInf
        if (xn>zn):
                if (min<pValor): #subexpresion de tripomastigota respecto a
                epimastigota = sobreexpresion de epi respecto de tripo
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                        out.write ('\t\t Subexpresion de tripomastigota respecto
                        a epimastigota \n')
        else:
                if (min<pValor): #sobreexpresión de tripomastigota respecto a
                epimastigota = subexpresión de epi respecto a tripo
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                        out.write ('\t\t Sobreexpresion de tripomastigota
                        respecto a epimastigota \n')
elif (x>z):
        acum0=0
        acumInf=0
        for n in range(0,z+1):
                acum0=acum0+ac(N1,N3,x,n)
        t=x+1
        while t<top:
                pControl=ac(N1,N3,x,t)
                if (pControl<=pLimiteXZ):
                        nuevoZ=t #tambien puede ser nuevoZ=t-1
                        t=top
                t+=1
        m=nuevoZ
        while m<top:
                acTest=ac(N1,N3,x,m)
                acumInf=acumInf+acTest
                if acTest<bottom or acumInf+acum0>pValor:
                        m=top
                m+=1
        if (acum0+acumInf<pValor): #subexpresion de amastigota respecto a
        epimastigota = sobreexpresion de epi respecto de ama
                imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
```

```
                if (xn>zn):
                        out.write ('\t\t Subexpresion de tripomastigota respecto
                        a epimastigota \n')
                else:
                        out.write ('\t\t Sobreexpresion de tripomastigota
                        respecto a epimastigota \n')
else: #x<=z
        acum0=0
        acumInf=0
        w=0
        while w<x:
                acAux=ac(N1,N3,w,z)
                if acAux>=pLimiteXZ:
                        w=x+2
                else:
                        acum0=acum0+acAux
                w+=1
        m=z
        while m<top:
                acTest=ac(N1,N3,x,m)
                acumInf=acumInf+acTest
                if acTest<bottom or acumInf+acum0>pValor:
                        m=top
                m+=1
        if (acum0+acumInf<pValor): #sobreexpresión de amastigota respecto a
        epimastigota = subexpresión de epi respecto a ama
                imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
                if (xn<zn):
                        out.write ('\t\t Sobreexpresion de tripomastigota
                        respecto a epimastigota \n')
                else:
                        out.write ('\t\t Subexpresion de tripomastigota respecto
                        a epimastigota \n')

if (y==z):
        acum0=0
        acumInf=0
        for n in range(0,z+1):
                        acum0=acum0+ac(N2,N3,y,n)
        m=z
        while m<top:
                acTest=ac(N2,N3,y,m)
                acumInf=acumInf+acTest
                if acTest<bottom:
                        m=top
                m+=1
        if acum0<acumInf:
                min=acum0
        else:
                min=acumInf
        if (yn>zn):
                if (min<pValor): #subexpresion de tripomastigota respecto a
                amastigota = sobreexpresion de ama respecto de tripo
```

```
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                        out.write ('\t\t Subexpresion de tripomastigota respecto
                        a amastigota \n')
        else:
                if (min<pValor): #sobreexpresión de tripomastigota respecto a
                amastigota = subexpresión de ama respecto a tripo
                        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,min)
                        out.write ('\t\t Sobreexpresion de tripomastigota
                        respecto a amastigota \n')
elif (y>z):
        acum0=0
        acumInf=0
        for n in range(0,z+1):
                acum0=acum0+ac(N2,N3,y,n)
        t=y+1
        while t<top:
                pControl=ac(N2,N3,y,t)
                if (pControl<=pLimiteYZ):
                        nuevoZ=t #tambien puede ser nuevoZ=t-1
                        t=top
                t+=1
        m=nuevoZ
        while m<top:
                acTest=ac(N2,N3,y,m)
                acumInf=acumInf+acTest
                if acTest<bottom or acumInf+acum0>pValor:
                        m=top
                m+=1
        if (acum0+acumInf<pValor): #subexpresion de tripomastigota respecto
        a amastigota = sobreexpresion de ama respecto de tripo
                imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
                if (yn>zn):
                        out.write ('\t\t Subexpresion de tripomastigota respecto
                        a amastigota \n')
                else:
                        out.write ('\t\t Sobreexpresion de tripomastigota
                        respecto a amastigota \n')
else: #y<=z
        acum0=0
        acumInf=0
        w=0
        while w<y:
                acAux=ac(N2,N3,w,z)
                if acAux>=pLimiteYZ:
                        w=y+2
                else:
                        acum0=acum0+acAux
                w+=1
        m=z
        while m<top:
                acTest=ac(N2,N3,y,m)
                acumInf=acumInf+acTest
                if acTest<bottom or acumInf+acum0>pValor:
```

```
                m=top
        m+=1
if (acumInf+acum0<pValor): #sobreexpresión de tripomastigota
respecto a amastigota = subexpresión de ama respecto a tripo

        imprimoLinea(gen,desc,N1,N2,N3,x,y,z,acum0+acumInf)
        if (yn<zn):
                out.write ('\t\t Sobreexpresion de tripomastigota
                respecto a amastigota \n')
        else:
                out.write ('\t\t Subexpresion de tripomastigota respecto
                a amastigota \n')
```

Glosario

Cinetoplasto: también llamado kinetoplasto, es una masa de ADN circular extranuclear que contiene numerosas copias del genoma mitocondrial.

Cromatina: las unidades básicas de la cromatina son los nucleosomas, que son secuencias del orden de 150 pb de largo asociadas a un complejo específico de 8 histonas (octámero de histonas).

Haplotipo: constitución alélica de múltiples loci para un mismo cromosoma.

Heterocromatina: regiones de la cromatina más compactas, condensadas, empaquetadas que se tiñen fuertemente con coloraciones para ADN

Kinetoplástidos: grupo de protistas flagelados que incluye a varios parásitos responsables de enfermedades en seres humanos y otros animales.

Linfadenitis: inflamación de los ganglios linfáticos

Miocarditis intersticial: inflamación del miocardio y tejido conjuntivo de soporte que no tiene función orgánica, sino que contiene los vasos sanguíneos que nutren al parénquima y las terminaciones nerviosas. Buscar una definición más decorosa.

Policistrónico: cuando un ARNm codifica más de una proteína.

Sintenia: la sintenia describe la co-localización física de loci genéticos en el mismo cromosoma dentro de un individuo o especie.

Referencias

Altschul, Stephen F.; Gish, Warren; Miller, Webb; Myers, Eugene W.; Lipman, David J.; *Basic Local Alignment Search Tool*; Journal of Molecular Biology; 1990

Andersson, Björn; Åslund, Lena; Tammi, Martti; Tran, Anh-Nhi; Hoheisel, Jörg D.; Pettersson, Ulf; *Complete Sequence of a 93.4-kb Contig from Chromosome 3 of Trypanosoma cruzi Containing a Strand-Switch Region*; Genome Research; Vol. 8; pp. 809-816; 1998; doi. 10.1101/gr.8.8.809

Andrews, Simon; *FASTQC. A quality control tool for high throughput sequence data*; URL http://www. bioinformatics. babraham. ac. uk/projects/fastqc; 2010

Audic, Stéphane; Claverie, Jean-Michel; *The Significance of Digital Gene Expression Profiles*; Genome Research; pp. 986-995; 1997

Barret, Michel P.; Burchmore, Richard J. S.; Stich August; Lazzari Julio O.; Frasch, Alberto Carlos; Cazzulo, Juan José; Krishna, Sanjeev; *The trypanosomiases*; The Lancet; Vol. 362; November 1, 2003

Brandão, Adeilton; Jiang, Taijiao; *The composition of untranslated regions in Trypanosoma cruzi genes*; Parasitology International; 2009

Burrows, Michael; Wheeler, David J.; *A block-sorting lossless data compression algorithm*; 1994

Claverie, Jean-Michel; *Computational methods for the identification of differential and coordinated gene expression*; Human molecular genetics; Vol 8; pp 1821-1832; 1999

Conesa, Ana; Götz, Stefan; Garcia-Gomez, Juan Miguel; Terol, Javier; Talon, Manuel; Robles, Montserrat; *Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research*; Bioinformatics; Vol. 21; pp. 3674-3676; 2005

Ferragina, Paolo; Manzini, Giovanni; *Opportunistic data structures with applications*; Foundations of Computer Science; 2000

Freitas, Jorge; Augusto-Pinto, Luis; Pimenta, Juliana; Bastos-Rodrigues, Luciana; Gonçalves, Vanessa; Teixeira, Santuza; Chiari, Egler; Junqueira, Ângela; Fernandes, Octavio; Macedo, Andréa; Machado, Carlos; Pena, Sérgio;

Ancestral Genomes, Sex, and the Population Structure of Trypanosoma cruzi; PLoS Pathogens; Vol. 2; 2006

Gómez Gutiérrez, Alberto; Monteón Padilla, Victor M.; *Algunos aspectos de la organización y regulación genética de Trypanosoma cruzi: El agente etiológico de la enfermedad de Chagas*; Revista Latinoamericana de Microbiología; Vol 50; Nos 3 y 4; Julio-Setiembre 2008, Octubre-Diciembre 2008

Hamilton, Patrick; Gibson, Wendy; Stevens, Jaime; *Patterns of co-evolution between trypanosomes and their hosts deduced from ribosomal RNA and protein-coding gene phylogenies*; Molecular Phylogenetics and Evolution; Vol. 44; pp. 15-25; 2007

Jawetz, Ernest; Melnick, Joseph L.; Adelberg, Edward A.; Microbiología Médica; 14ª edición; México; 1992

Langmead, Ben; Trapnell, Cole; Pop, Mihai; Salzberg, Steven L.; *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome*; Genome Biology 10; No. 3; 2009

Liang, Xue-hai; Haritan, Asaf; Uliel, Shai; Michaeli, Shulamit; *Trans and cis Splicing in Trypanosomatids Mechanism, Factors, and Regulation*; Eukaryotic Cell; Vol. 2; No. 5; pp.830-840; 2003; doi. 10.1128/EC.2.5.830-840.2003

Mandelboim, Michal; Barth, Sarit; Biton, Moshe; Liang, Xue-hai; Michaeli, Schilamit; *Silencing of Sm proteins in Trypanosoma brucei by RNA interference captured a novel cytoplasmic intermediate in spliced leader RNA biogenesis*; Journal of Biological Chemistry; Vol. 278; pp. 51469-51478; 2003; doi: 10.1074/jbc.M308997200.

Metzker, Michael L.; *Sequencing technologies – the next generation*; Nature Reviews; volumen 11, January 2010

Mignon Balzer, Susanne; *Characteristics of Pyrosequencing Data Analysis, Methods, and Tools*; PhD thesis, University of Berger, 2013

Mortazavi, Ali; Williams, Brian A.; McCue, Kenneth; Schaeffer, Lorian; Wold, Barbara; *Mapping and quantifying mammalian transcriptomes by RNA-Seq*; Nature Methods 5, 621 – 628; 2008; doi:10.1038/nmeth.1226

Ramírez Macías, María Inmaculada; *Búsqueda de nuevos fármacos de actividad antiparasitaria frente a Leishmania spp y Trypanosoma cruzi*; Tesis doctoral; Universidad de Granada; 2012.

Romualdi, C.; Bertoluzzi, S.; Danieli, G.A.; *Detecting differentially expressed genes in multiple tag sampling experiments: comparative evaluation of statistical tests*; Human molecular genetics; vol. 10; no. 19; pp. 2133-41; 2001

Rutherford, K.; Parkhill, J.; Crook, J.; Horsnell, T.; Rice, P.; Rajandream, M.A.; Barrell, B.; Artemis: sequence visualization and annotation; Bioinformatics (Oxford, England); Vol. 16; pp. 944-945; 2000

Smyth, J. D.; *Introduction to animal parasitology*; Cambidge University Press; 3era edición; 1994

Vanhée-Brossollet, C.; Vaquero, C.; *Do natural antisense transcripts make sense in eukaryotes?*; Gene 211, no. 1;pp. 1-9; 1998

Waterhouse, A.M.; Procter, J.B.; Martin, D.M.A; Clamp, M.; Barton, G. J.; *Jalview Version 2 - a multiple sequence alignment editor and analysis workbench*; Bioinformatics 25; no. 9; pp.1189-1191; 2009; doi: 10.1093/bioinformatics/btp033

Weatherly, D. Brent; Boehlke, Courtney; Tarleton, Rick L.; Chromosome level *assembly of the hybrid Trypanosoma cruzi genome*; BMC Genomics; junio 2009

Wright, André-Denis; Li, Sen; Feng, Shujuan; Martin, Donald S.; Lynn, Denis H.; *Phylogenetic position of the kinetoplastids, Cryptobia bullocki, Cryptobia catostomi, and Cryptobia salmositica and monophyly of the genus Trypanosoma inferred from small subunit ribosomal RNA sequences*; Molecular and Biochemical Parasitology, 1999

Referencias web

http://454.com/products/gs-flx-system/index.asp

http://cran.r-project.org

http://es.wikipedia.org/wiki/Trypanosoma_cruzi

https://products.appliedbiosystems.com/ab/en/US/adirect/ab?cmd=catNavigate2&catID=607061&tab=DetailInfo

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

http://www.blast2go.com/b2ghome

http://www.cdc.gov/parasites/chagas/biology.html

http://www.cdc.gov/parasites/chagas/es/enfermedad.html

http://www.illumina.com/systems/sequencing.ilmn

https://www.nanoporetech.com/

http://www.paho.org/hq/index.php?option=com_content&view=category&id=3591&layout=blog&Itemid=1051&lang=es

http://www.wellcome.ac.uk/Education-resources/Teaching-and-education/Animations/DNA/WTDV026689.htm

http://www.youtube.com/watch?v=l99aKKHcxC4&feature=related