FACULTAD DE CIENCIAS AGRARIAS

FACULTAD DE CIENCIAS BIOQUÍMICAS Y FARMACÉUTICAS

UNIVERSIDAD NACIONAL DE ROSARIO

Análisis de la distribución de potenciales cuádruplex de Guanina (PQS) en el genoma de tripanosomátidos y su posible relación con el control de la expresión génica.

Lic. Diego Leonardo Andino

TRABAJO FINAL PARA OPTAR AL TITULO DE ESPECIALISTA EN BIOINFORMÁTICA

DIRECTOR: Dra. Pamela Cribb

2017

Análisis de la distribución de potenciales cuádruplex de Guanina (PQS) en el genoma de tripanosomátidos y su posible relación con el control de la expresión génica.

Diego Leonardo Andino

Licenciado en Genética – Universidad Nacional de Misiones

Este Trabajo Final es presentado como parte de los requisitos para optar al grado académico de Especialista en Bioinformática, de la Universidad Nacional de Rosario y no ha sido previamente presentada para la obtención de otro título en ésta u otra Universidad. El mismo contiene los resultados obtenidos en investigaciones llevadas a cabo en FCEQyN-UNLaR, durante el período comprendido entre Julio de 2016 y Agosto de 2017, bajo la dirección de Dra. Pamela Cribb.

Diego Leonardo Andino

Pamela Cribb

Defendida: 14 de Marzo de 2018.

Agradecimientos

A mi directora, por guiarme con paciencia y hacerme sentir libre durante el avance de este trabajo.

A mis amigos que, aun a la distancia, siempre están.

A mi padre y hermanos, por permitirme disfrutar de una buena familia.

A esa persona especial, que me ayuda diariamente con todas mis tribulaciones, me da fuerzas para alcanzar mis metas y, lo mas importante…me hace feliz. Gracias Gallega!!!

A mi madre

Publicaciones y Presentaciones a Congresos

Congreso: Reunión Conjunta de Sociedades de BioCiencias. 13 al 17 de noviembre de 2017, Cuidad Autónoma de Buenos Aires, Argentina.
Comunicación: Poster
Título: Putative Quadruplex Sequence (PQS) disrtribution in Trypanosomatid genomes: a mark for transcription or translation termination?
Autores: Andino Diego, Margarit Ezequiel, Cribb Pamela.

Abreviaciones

CDSs (*coding DNA sequence*): porción del ARNm que es traducida a proteína.

PQS (*putative quadruplex sequence*): secuencia potencialmente formadora de cuádruples de guanina.

PTU (*polycistronic transcription units*): grupo de genes contiguos que son transcritos en una única molécula de ARN policistrónico.

SL (*spliced leader*): secuencia corta que es adicionada al extremo 5’ de los ARNm en tripanosomátidos mediante el fenómeno de *trans-splicing*.

SSR (*strand switch region)*: secuencia que separa dos PTUs adyacentes y en distintas hebras.

UTR 3’ (*untranslated region 3’*): porción 3’ del ARNm que no es traducido a proteína.

UTR 5’ (*untranslated region 5’*): porción 5’ del ARNm que no es traducido a proteína.

Resumen

Los tripanosomátidos son una familia de eucariotas unicelulares con especies causantes de importantes enfermedades en el ser humano. Los genes de estos organismos suelen organizarse en lo que se conoce como unidades de transcripción policistrónica (PTUs), consistente en un grupo de genes (y secuencias intergénicas) adyacentes en la misma hebra, que se transcriben en forma conjunta dando lugar a una única molécula de ARN (ARN policistrónico). Los ARNm maduros son producidos a partir de los ARN policistrónicos mediante un proceso de maduración que incluye el fenómeno de *trans-splicing* y poliadenilación. Actualmente se conocen pocos detalles sobre los mecanismos de regulación de la expresión génica tanto a nivel transcripcional (inicio y terminación de la síntesis de los ARN policistrónicos) como post-transcripcional (maduración, estabilidad y traducción de los ARNm).

Los cuádruples de guanina son estructuras secundarias del ADN o ARN formadas en regiones ricas en guaninas. Se ha descrito la importancia de estas estructuras en la regulación de la expresión génica en diversos organismos.

En el presente trabajo se realizó un análisis bioinformático de la frecuencia y distribución de cuádruples putativos de guanina (PQSs) en los genomas de distintas especies de *Leishmania* y *Trypanosoma*, a fin de evaluar su posible impacto en la regulación de la expresión génica a nivel transcripcional y post-transcripcional.

En *Trypanosoma* se observó una densidad baja de PQSs a nivel genómico con un patrón poco claro entre las distintas especies. Por otro lado, en *Leishmania* se observó una mayor densidad de PQSs a nivel genómico, con una distribución bien definida y similar entre las tres especies analizadas. En este último género, la densidad de PQSs aumenta en la hebra molde de la región adyacente al extremo terminal de las PTUs, por lo cual es posible que los cuádruples de guanina puedan estar vinculados a la terminación de la transcripción. Al analizar la distribución de PQSs dentro de las PTUs de ambos géneros, se observó una diferencia muy marcada en la densidad de PQSs entre las secuencias génicas e intergénicas (con una muy baja densidad de PQSs en las primeras en relación a las segundas). Esto podría ser indicio de algún rol de los cuádruples de guanina en las etapas maduración, estabilidad y/o traducción del ARNm.

Abstract

Trypanosomatids are a family of unicellular eukaryotes including species that cause important diseases in humans. Genes of these organisms are usually organized in polycistronic transcription units (PTUs). PTUs consist on a group of genes (and intergenic sequences) that are located in tandem in the same strand. They are transcribed together to form a single molecule of RNA (polycistronic RNA). Mature mRNAs are produced from polycistronic RNAs through a maturation process that includes the phenomenon of trans-splicing and polyadenylation. Currently, few details are known about the mechanisms of regulation of gene expression both at the transcriptional level (initiation and termination of polycistronic RNA synthesis) and post-transcriptional level (maturation, stability and translation of mRNAs).

Guanine quadruplexes are secondary structures of DNA or RNA which are formed in regions rich in guanines. The importance of these structures in the regulation of gene expression in various organisms has been described.

In order to evaluate the possible impact of guanine quadruplexes on the regulation of gene expression at transcriptional and post-transcriptional level in Trypanosomatids, we performed a bioinformatic analysis of the frequency and distribution of putative guanine quadruplexes (PQSs) in the genomes of different species of *Leishmania* and *Trypanosoma*.

We observed a low density of PQSs in *Trypanosoma* genome, with an unclear pattern among the different species. On the other hand, we identified a higher density of PQSs in *Leishmania* genome with a well defined and similar distribution among the three species analyzed. In *Leishmania* genus, the density of PQSs increases in the region adjacent to the terminal end of the PTUs in the template strand. For that reason, we propose that guanine quadruplexes may be linked to the termination of transcription. When we analyze the distribution of PQSs within the PTUs of both genera, we observed a very marked difference in the density of PQSs between the gene and intergenic sequences (with a very low density of PQSs in the former in relation to the latter). This may indicate a role of guanine quadruplexes in maturation, stability and / or translation of the mRNA.

Índice

Introducción

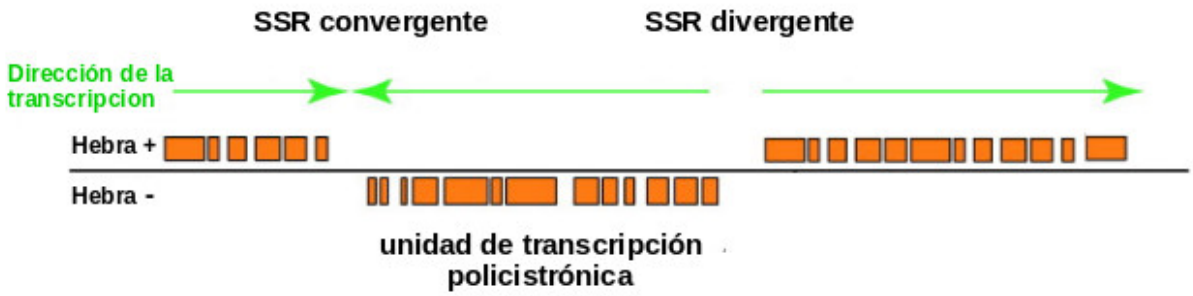
Los tripanosomátidos son un grupo de eucariotas unicelulares, pertenecientes al orden Kinetoplastida, entre los que se reconocen diversos géneros que incluyen parásitos de invertebrados (*Crithidia* y *Leptomonas*), parásitos de plantas e invertebrados (*Phytomonas*), y parásitos de vertebrados e invertebrados (como *Trypanosoma* y *Leishmania*) (Olsen, 1974). Estos últimos géneros son de gran importancia sanitaria, pues incluyen numerosas especies causantes de enfermedades en el ser humano, como la leishmaniasis (*Leishmania* spp.), la enfermedad de Chagas (*Trypanosoma cruzi*) y la enfermedad del sueño (*Trypanosmoma brucei*) (Parsons et al., 2005).

Las primeras secuencias genómicas de tripanosomátidos (*L. major, T. cruzi y T. brucei*) fueron publicadas en el año 2005 (Berriman et al., 2005; El-Sayed et al., 2005; Ivens et al., 2005). Actualmente se cuenta con las secuencias genómicas de decenas de especies de tripanosomátidos. Esta información se encuentra disponible tanto en la base de datos del consorcio internacional (DDBJ, EMBL, GenBank) como en la de TriTrypDB (Recuadro 1) (Aslett et al., 2010).

Recuadro 1
La base de datos TriTrypDB es una base de datos relacional que contiene toda la información disponible acerca de las secuencias genómicas de distintas especies y cepas de *tripanosomátidos* (http://tritrypdb.org/). Se pueden descargar directamente del sitio tanto las secuencias como la información correspondiente a las mismas: nombre o ID, cromosoma, ubicación genómica (sitio de inicio y terminación de la secuencia codificante), tipo de gen (codificante para proteína, rARN, tARN, etc), información acerca de 5´ y 3´UTRs, etc., lo cual constituye una herramienta importante para estudios *in silico*.

Los tripanosomátidos poseen un sistema de expresión génica notablemente diferente al típico eucariota, cuya principal característica es la agrupación de los genes codificantes en varias **Unidades de Transcripción Policistrónicas** (**PTUs**), que son transcritas por la ARN polimerasa II. Las PTUs adyacentes pueden localizarse en la misma o en distintas hebras (separadas por **regiones de cambio de hebra – SSRs**). Estas SSRs pueden ser divergentes (si se encuentran en el extremo de inicio de la transcripción), o convergentes (si se encuentran en el extremo de terminación de la transcripción) (Maree & Patterton, 2014; Myler et al., 2001) (Fig. 1).

Figura 1: Representación de la disposición de los genes (recuadros naranjas) en unidades de transcripción policistrónicas y sus direcciones de transcripción según la hebra (Modificado de Siegel y col., 2011)



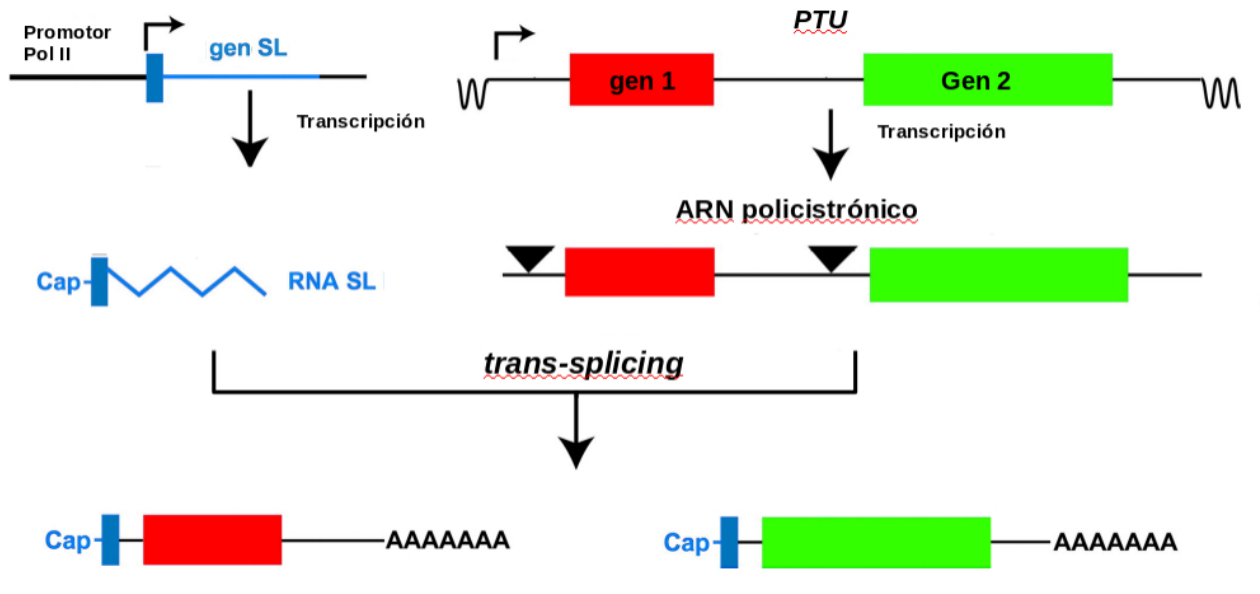
Salvo una excepción (correspondiente al ARN-SL que codifica para la secuencia líder), no se han podido identificar secuencias promotoras para la ARN polimerasa II, por lo cual la naturaleza del inicio de la transcripción de las PTUs aún es motivo de investigación. Diversos estudios se han enfocado en las SSRs divergentes como sitios de inicio de síntesis de los ARN policistrónicos en *Trypanosoma* y *Leishmania*. Si bien existen evidencias de una posible regulación epigenética en estas regiones (presencia de variantes de histonas o modificaciones de las mismas en los sitios de inicio de la transcripción), aún no se ha podido determinar en forma clara ninguna secuencia que actúe como señal para la unión y actividad de la ARN polimerasa II (Maree & Patterton, 2014; Respuela et al., 2008; Siegel et al., 2009).

Por otro lado, la expresión conjunta de un gran número de genes por cada PTU transcrito permite suponer que la regulación fina de la expresión génica se da a nivel postranscripcional (maduración, estabilidad y traducción de los ARNm) (Smircich et al., 2013). La maduración del ARN policistrónico se da por los fenómenos de *trans-splicing*, mediante el cual se adiciona una secuencia líder (SL) de 39-45 nucleótidos en el extremo 5´de cada secuencia codificante para proteínas y de poliadenilación en su extremo 3´ (Kolev et al., 2010)(Fig. 2). Cada ARN codificante maduro está formado por una secuencia lider, regiones UTRs 5’ y 3’ de longitud variable (Dillon et al., 2015), la región CDS y una cola de poli-A.

Los análisis bioinformáticos realizados a fin de determinar patrones de secuencia que puedan estar implicados en la regulación de la expresión génica tanto a nivel transcripcional como postranscripcional han logrado resultados poco concluyentes. A nivel de la transcripción, únicamente se han establecido ciertas características generales de las regiones de cambio de hebra, como su composición en GC y algunos motivos poli-G (Martinez-Calvillo et al., 2003; Siegel et al., 2009). Se ha sugerido que la presencia de estos motivos poli-G podría dar lugar a la formación de

estructuras secundarias particulares (**cuádruples de Guanina**) que permitirían la unión de factores de transcripción y podrían definir la direccionalidad de la transcripción (Siegel et al., 2009). A nivel postranscripcional, se ha visto que las regiones UTRs pueden estar desempeñando roles importantes en el procesamiento del ARN policistrónico, la estabilidad de los ARNm generados y el nivel de traducción (eficiencia traduccional) de los mismos (Martinez-Calvillo et al., 2010; Song et al., 2016).

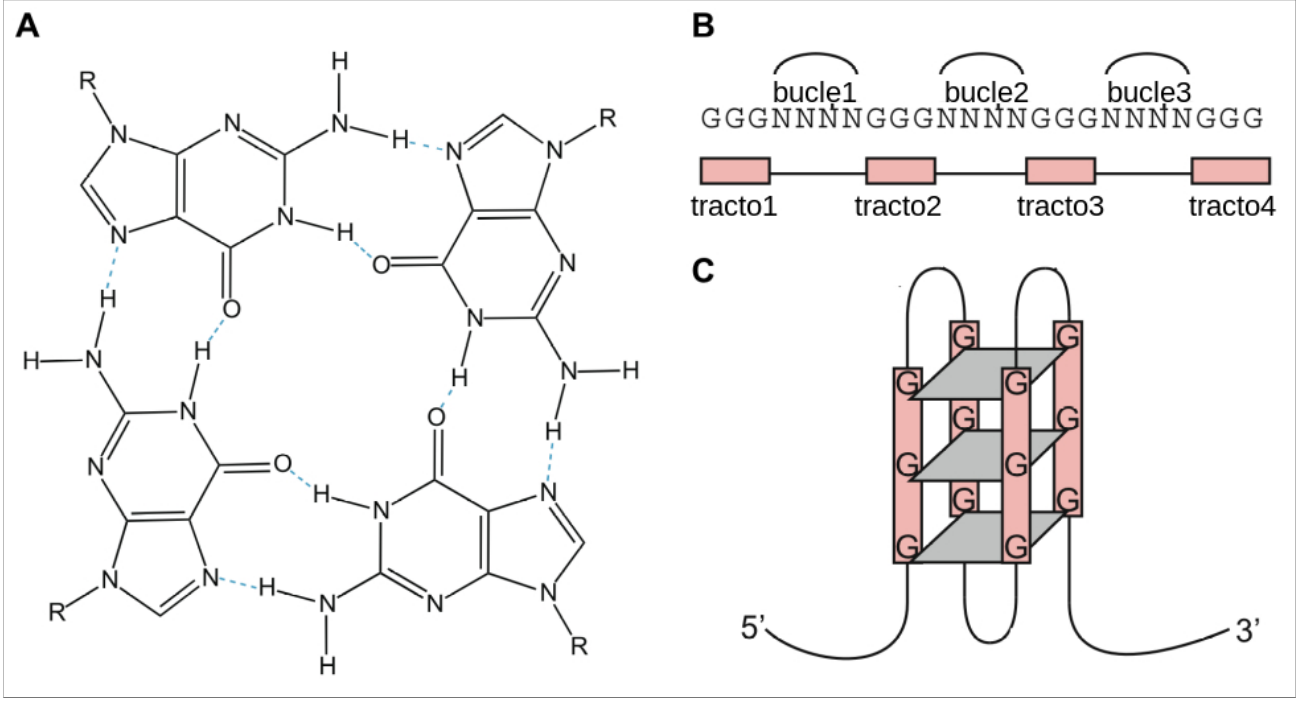
Figura 2: Esquema del proceso de expresión génica en tripanosomátidos. La transcripción de los genes SL (azul) genera RNAs SL que, durante el trans-splicing, transfieren 39nt (rectangulo azul) al extremo 5' de cada gen procesado a partir del ARN policistrónico. Los procesos de poliadenilación y *trans-splicing* se encuentran asociados temporal y espacialmente; la poliadenilación de un gen ocurre simultáneamente al *trans* splicing del gen consecutivo. Los triángulos invertidos indican los puntos de corte del ARN policistrónico (modificado de Landfear, 2003).



Los cuádruples de guanina (G4) son estructuras no convencionales del ADN o ARN en las que cuatro Guaninas (no adyacentes) interaccionan entre sí mediante puentes de Hidrógeno de Hoogsteen (no Watson y Crick) dando lugar a una estructura cuadrangular aplanada denominada tétrada (Fig. 3). Numerosos estudios vinculan estas estructuras con procesos como la replicación, la transcripción, la traducción y la recombinación genética (Frees et al., 2014; Maizels & Gray, 2013; Song et al., 2016). Las secuencias ricas en Guanina que presentan el patrón de bases GxNyGxNyGxNyGx (x>2; y>0) son consideradas secuencias **putativas formadoras de G4** (**PQS**) y se ha visto que están enriquecidas en determinadas regiones del genoma como las regiones

teloméricas, secuencias promotoras, sitios de unión a factores de transcripción y 5´UTR de los ARNm (Du et al., 2008).

Figura 3: (A) Tétrada de guaninas en disposición planar unidas mediante puentes de Hidrógenos tipo Hoogsteen (lineas punteadas azul). (B) Patrón de secuencia donde se representan los tractos de guanina (rectangulos) y los bucles que los separan (lineas). (C) Disposición espacial de los tractos de guanina y los bucles formando un cuádruple de guanina (modificado de Capra et al., 2010).



Pese a que en la actualidad existen diversos algoritmos y herramientas computacionales que permiten identificar probables cuádruples de Guanina en secuencias de ADN (Cammas & Millevoi, 2016), hasta el momento no se ha reportado ningún estudio sistemático de la ocurrencia y distribución de dichos cuádruples en los genomas de tripanosomátidos.

En el presente estudio se analizó la ocurrencia de PQSs en los genomas de 6 especies de tripanosomátidos. Para ello se trabajó con las secuencias genómicas y anotaciones disponibles en la base de datos TriTrypDB. En todos los casos se determinaron las densidades y distribución de PQSs en las PTUs, SSRs convergentes y divergentes, por un lado, y entre secuencias génicas e intergénicas, por el otro, y se analizó su posible relación con mecanismos de regulación de la expresión génica a distintos niveles.

Objetivos

El objetivo general de este trabajo es diseñar las herramientas bioinformáticas que nos permitan realizar un análisis global acerca de la distribución de las posibles secuencias formadoras de cuádruples de guanina en el genoma de tripanosomátidos a fin de determinar si existe una asociación entre estas estructuras y el control de la expresión génica.

Para alcanzar el objetivo general, nos proponemos los siguientes objetivos específicos:

- Identificar las posibles secuencias formadoras de cuádruples de guanina en el genoma de tres especies de *Leishmania* y tres especies de *Trypanosoma* utilizando un predictor de PQSs.
- Diseñar un algoritmo y un *script* que permitan extraer de la base de datos Tritryp la información de las posiciones y las secuencias correspondientes a las regiones génicas e intergénicas.
- Diseñar un algoritmo que permita definir las posiciones y secuencias correspondientes a las PTUs y a las regiones de cambio de hebra (SSR), agrupándolas según sean SSR convergentes o divergentes.
- Analizar si existe una correlación entre el número de posibles cuádruples de Guanina (PQS) y los distintos tipos de secuencias genómicas consideradas.

Metodología

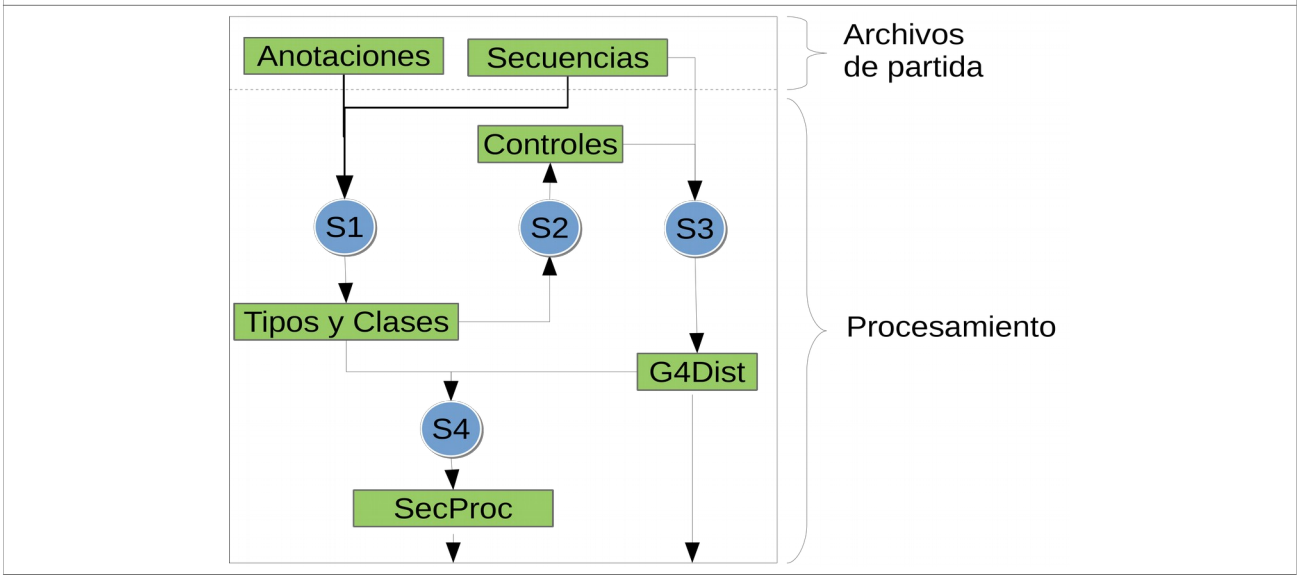
El estudio de distribución de cuádruples de guanina putativos (PQSs) se desarrolló íntegramente mediante el lenguaje de programación R (R Core Team, 2017). Se utilizaron paquetes existentes y se desarrollaron funciones específicas que, mediante el cambio de parámetros, permiten una mayor versatilidad durante el análisis. Todas las funciones creadas fueron almacenadas en un paquete ("TrypR") para facilitar su utilización.

El análisis se realizó en dos etapas sucesivas. En la primera etapa (Figura 4) se llevó a cabo el procesamiento de la información descargada de Tritryp (secuencias genómicas y anotaciones) y se obtuvieron dos archivos de salida: SecProc (conteniendo información de secuencias, contenido de bases y densidad de PQSs) y G4Dist (conteniendo información de posición de PQSs). En la segunda etapa (Figura 5) se utilizaron los archivos de salida de la etapa anterior y mediante la aplicación de 3 *scripts* independientes se obtuvieron los resultados finales (tablas, gráficos y análisis estadísticos).

La totalidad del código empleado se presenta en el Anexo III y puede ser descargado desde: https://github.com/dandinoar/especializacion

Etapa 1: Procesamiento de la información inicial

Figura 4: Diagrama de flujo de la etapa de procesamiento de secuencias. En verde se representan los archivos iniciales (Anotaciones y Secuencias), los archivos intermedios generados (Tipos y Clases, Controles) y los archivos de salida (SecProc y G4Dist). En azul, los s*cripts* utilizados (S1 a S4).



Archivos iniciales:

Se utilizó la última versión (versión 30) de los genomas de *Leishmania major* Friedlin, *L. donovani* BPK282A1, *L. infantum* JPCM5, *Trypanosoma cruzi* CL Brener Esmeraldo-like, *T. cruzi* CL Brener Non Esmeraldo-like, *T. brucei gambiense* DAL972 y *T. evansi* STIB805, disponibles en la página http://tritrypdb.org/common/downloads/release-30/

Se descargaron los archivos ".fasta" con la secuencias genómicas completas (Secuencias) y los archivos ".GFF" con las anotaciones correspondientes (Anotaciones).

Determinación de clases y tipos de secuencias y asignación de contenido de bases (Script S1):

Las secuencias fueron agrupadas en dos niveles jerárquicos, según la siguiente tabla:

Clase	**Tipo**
PTU (*polycistronic transcription unit*)	Génica**
	Intergénicas**
SSR de Inicio (divergentes)	-
SSR de Terminación (convergentes)	-
Telómero de inicio*	-
Telómero de terminación*	-

(*) Las secuencias teloméricas no fueron utilizadas en los análisis posteriores.

(**) Las secuencias génicas coinciden con las CDSs en el caso de los genes codificantes de proteinas. Las UTRs 3' y 5' de cada gen no se encuentran definidas en el archivo de anotaciones, por lo cual no han sido consideradas en el presente trabajo.

Se utilizó la información de posición de las secuencias génicas para establecer la posición, longitud y hebra de distintas clases de secuencias a analizar: PTUs , SSRs divergentes y SSRs convergentes. Cada PTU consiste en una o más secuencias génicas consecutivas en la misma hebra, más las secuencias que las separan (intergénicas). Las SSRs consisten en las regiones que separan dos secuencias génicas adyacentes codificadas en distintas hebras. Las SSR divergentes son aquellas en que el cambio de hebra se da de "-" a "+" y las convergentes aquellas donde el cambio es de "+" a "-" (Ver figura 1, introducción). Adicionalmente se determinó el contenido de bases (G, C, A, T y N) para cada tipo y clase de secuencia y se calculó la longitud efectiva de las mismas (longitud real menos número de bases "N"). Los resultados fueron almacenados en el archivo "Tipos y clases".

Generación de genomas control (Script S2):

A partir de la información anterior (archivo "Tipos y clases"), se generaron genomas artificiales, respetando las longitudes, posiciones, polaridades y contenido de bases de cada tipo de secuencia, pero disponiendo las bases al azar dentro de las mismas. Se generó un genoma control para cada genoma original analizado. Los resultados fueron almacenados en el archivo "Controles".

Determinación de PQSs (Script S3):

Para la determinación de PQSs se analizaron los genomas completos y sus controles mediante el paquete de R "pqsfinder" (Jiri Hon, 2016). Este paquete permite la determinación de PQSs imperfectos (aquellos que poseen pequeñas regiones de no apareamiento en los tractos G), así como también la determinación de PQSs intra- e inter-hebra (Kudlicki, 2016; Mukundan & Phan, 2013). Cada PQS predicho está asociado a un valor de "*score*". En el presente trabajo se determinaron únicamente PQSs intra-hebra, y se utilizaron las opciones por defecto en el algoritmo de predicción. Del total de PQSs predichos, se seleccionaron para los análisis posteriores únicamente aquellos con *score* mayor a 85. Los resultados fueron almacenados en el archivo "G4Dist".

Determinación de número y densidad de PQSs por secuencia (Script S4):

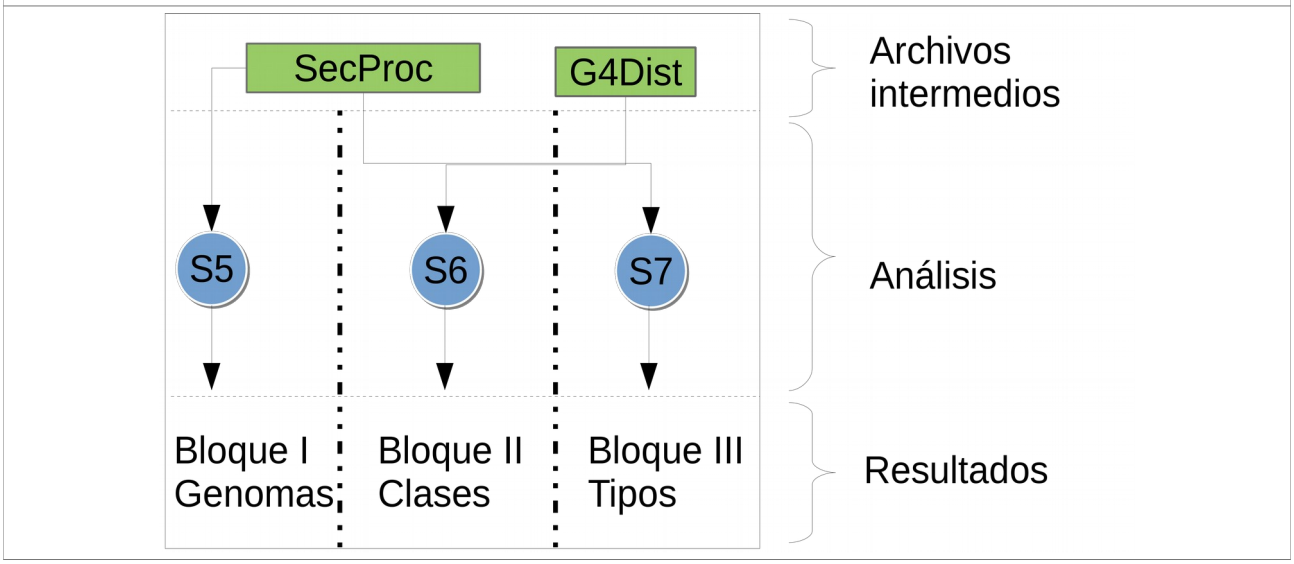
A partir de los datos de clases y tipos de secuencias (salida del *script* S1) y posición de PQSs (salida *script* S3), se determinó el número total de PQSs en cada secuencia. En el caso de secuencias con polaridad definida, también se determinó el número de PQSs para cada una de las hebras (codificante o molde). Luego, teniendo en cuenta la longitud efectiva de cada secuencia, se determinaron las densidades de PQSs (PQSs/Kb ambas hebras y discriminado por hebra). Los resultados fueron almacenados en el archivo "SecProc".

Etapa 2: Análisis y resultados:

El análisis de los datos y presentación de los resultados fue estructurado en tres bloques. En el primer bloque se determinan las características generales de los diferentes genomas. En el segundo bloque se analizan las secuencias a nivel de clases (SSRs y PTUs). En el tercer bloque se

estudian en mayor detalle las características de los PTUs, analizando las secuencias a nivel de tipos (secuencias génicas e intergénicas) y considerando diferencias entre hebras. En todos los bloques se parte de los archivos generados en la primer etapa (SecProc y G4Dist) (Figura 5).

Figura 5: Diagrama de flujo del análisis de datos. En verde se representan los archivos obtenidos en la etapa anterior (SecProc y G4Dist) En azul, los *Scripts* utilizados (S5 a S7). Las líneas discontinuas separan cada uno de los bloques en los que fue dividida esta etapa.



Bloque I: Datos genómicos

I.1) Tamaño genómico, contenido GC y densidad PQS (Script S5)

Se determinó el tamaño genómico, contenido de bases y densidad de PQSs para cada genoma y sus controles correspondientes. Los resultados se presentan en la Tabla 1 (Anexo I).

Bloque II: Análisis interclase (PTUs, SSRs inicio, SSRs terminación) (Script S6)

II.1) Características de las distintas clases de secuencias

Se determinó el número y tamaño de las distintas clases para cada genoma. Los resultados se presentan en las Tablas 2a a 2c (Anexo I) y en la figura II.1 (Anexo II).

II.2) Densidad de PQSs y contenido GC

Se obtuvieron los valores de PQS/Kb y %GC para cada clase. Los resultados se presentan en las Tablas 3a a 3d y 4 (Anexo I) y en la figura II.2 (Anexo II).

II.3) Distribución de PQSs

Se estableció la densidad de PQSs relativa (PQSs/nº secuencias) para cada nucleótido aguas arriba y aguas abajo de los extremos de inicio y terminación de los PTUs. Se consideraron las hebras (codificante y molde) en forma independiente. Los resultados se presentan en las figuras II.3a y II.3b (Anexo II).

Bloque III: Análisis clase PTUs (tipos y hebras) (Script S7)

III.1) Densidad de PQSs y contenido GC por tipo de secuencia

Se obtuvieron los valores de PQS/Kb y %GC para cada tipo de secuencia. Los resultados se presentan en las Tablas 6 a 7 (Anexo I) y en la figura II.4 (Anexo II).

III.2) Densidad de PQS y contenido G por hebra

Se obtuvo el valor promedio de PQSs/Kb y %G en cada hebra de los PTUs. Los resultados se presentan en las Tablas 8 a 9 (Anexo I) y en la figura II.5 (Anexo II).

III.3) Densidad de PQS y contenido G por hebra y tipo de secuencia

Se obtuvieron los valores promedio de PQSs/Kb y %G en cada hebra de las secuencias génicas e intergénicas. Los resultados se presentan en las Tablas 10 a 11 (Anexo I) y en las figuras II.6a y II.6b (Anexo II).

III.4) Distribución de PQSs

Se estableció la densidad de PQSs relativa (PQSs/nº secuencias) para cada nucleótido aguas arriba y aguas abajo de los extremos de inicio y terminación de las secuencias génicas. Se consideraron las hebras (codificante y molde) en forma independiente. Los resultados se presentan en la figura II.7 (Anexo II).

Resultados

Bloque I: Datos genómicos

I.1) Tamaño genómico, contenido GC y densidad PQS

En la tabla 1 se resumen algunas características de los genomas estudiados. Se observan claras diferencias en cuanto al tamaño, el contenido GC y la densidad media de PQSs.

Tabla 1: Datos generales de los genomas. (C) Tamaño genómico, (N.percent) % de bases sin determinar, (GC.percent) % GC, (PQSs) número total de cuádruples, (PQSs.Kb) cuádruples cada 1000 bases, (PQSs.Kb.control) cuádruples cada 1000 bases en los controles (Datos completos en Anexo I: tabla 1).

	C(pb)	N.percent	GC.percent	PQSs.Kb	PQSs.Kb.control
L. donovani	32444968	3,7	59,5	0,755	0,632
L. infantum	32103026	0,1	59,6	0,801	0,706
L. major	32855095	0	59,7	0,874	0,73
T. brucei	22148088	0,2	47,2	0,183	0,095
***T. cruzi* Elike**	32529070	20,6	50,4	0,245	0,254
***T. cruzi* nElike**	32529072	14,7	50,7	0,243	0,266
T. evansi	25432160	0	46,5	0,168	0,088

En relación al tamaño genómico, se pueden definir dos grupos:

-**Genomas grandes:** Las tres especies de *Leishmania* y los dos genomas de *T. cruzi*, con genomas de 32Mpb.

-**Genomas pequeños:** *T. brucei* y *T. evansi*, con genomas entre 22 y 25Mpb.

En relación a contenido GC, también se pueden establecer dos grupos:

-**Alto GC y PQSs:** Las tres especies de *Leishmania*, con %GC en torno a 60 y PQS/Kb en torno a 0,8.

-**Bajo GC y PQSs:** Las cuatro especies de *Trypanosoma*, con %GC entre 47 a 50 y PQS/Kb en torno a 0,2.

La densidad de PQSs en los genomas y en los controles no guarda relación con los tamaños genómicos, pero si coincide con los grupos definidos según el contenido GC. Salvo en los genomas de *T. cruzi**, en el resto de los casos hay una densidad de cuádruples mayor (de casi el doble en *T. evansi* y *T. brucei*) que la esperada según los controles.

*En relación al genoma de *T. cruzi*, es importante mencionar la gran proporción de bases sin determinar (entre 15 y 20%) que podrían afectar los resultados del análisis.

Bloque II: Análisis interclase (PTUs, SSRs inicio, SSRs terminación)

II.1) Procesamiento de secuencias

Se definieron las distintas clases de secuencias para cada genoma y se obtuvo la cantidad y tamaño medio (en pb y número de genes si corresponde) de cada una de ellas. En la tabla 2a se presenta la abundancia de cada clase de secuencia predicha y en la tabla 2b se presenta el tamaño medio en pb de cada una de ellas.

Tabla 2a: Número de clases de secuencias para los distintos genomas. PTUs: Unidades de transcripción policistrónicas; SSR init: regiones de cambio de hebra de inicio (divergentes); SSR term: regiones de cambio de hebra de terminación (convergentes). (Datos completos en Anexo I: tabla 2a)

	L.donovani	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi* Elike	*T.cruzi* nElike	*T.evansi*
SSR init	81	85	89	138	209	203	203
PTUs	187	215	194	282	427	418	422
SSR term	68	68	69	133	177	174	205

Tabla 2b: Resumen tamaño promedio (pb) de cada clase para los distintos genomas. **PTUs**: Unidades de transcripción policistrónicas; **SSR init**: regiones de cambio de hebra de inicio (divergentes); **SSR term**: regiones de cambio de hebra de terminación (convergentes). (Datos completos en Anexo I: tabla 2b)

	L.donovani	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi* Elike	*T.cruzi* nElike	*T.evansi*
SSR init	3109	2943	2481	4049	8571	7327	2481
PTUs	169197	145269	166072	74908	66370	69290	57947
SSR term	3744	3003	2123	3221	5208	3713	1922

Puede observarse que los genomas de las especies de *Leishmania* presentan menor cantidad de PTUs y de mayor tamaño que las especies de *Trypanosoma*. Es importante mencionar que, en este último género, muchos de los PTUs están constituidos por un único gen.

II.2) Densidad de PQSs y contenido GC

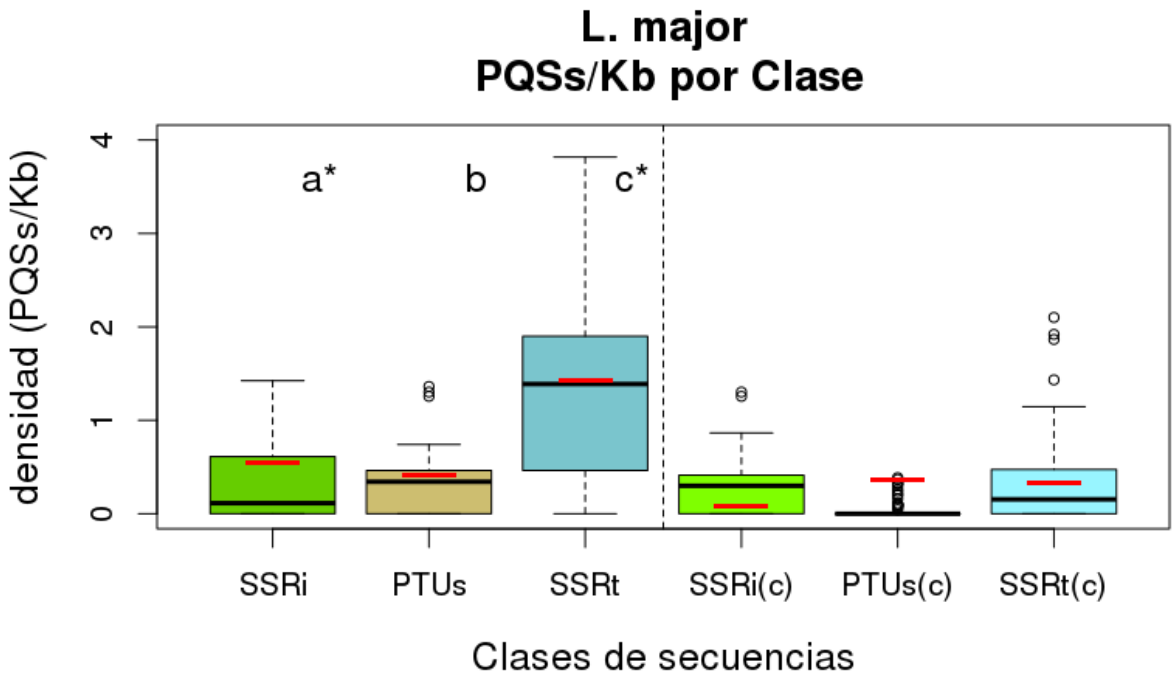
Para cada una de las clases analizadas se estableció la densidad promedio ponderada de PQSs (PQSs/Kb) y contenido GC (Anexo I: tablas 3 y 4; Anexo II: figura 2).

En la tabla 3 se presentan los valores absolutos de PQSs/Kb y la densidad relativa considerando los controles. En la Figura 6 se representan los valores de PQSs/Kb para las distintas clases de secuencias en el genoma de *Leishmania major* y su control.

Tabla 3: densidades medias de PQSs en las distintas clases de secuencias. Promedio ambas hebras. Unidades: PQS/Kb. Entre paréntesis figura el valor de la relación genoma/control.

	L.donovani	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi* Elike	*T.cruzi* nElike	*T.evansi*
SSR init	0,5518 (8,6)	0,5243 (6,4)	0,5435 (7,3)	0,1211 (6,1)	0,0993 (1,2)	0,0939 (1,2)	0,1003 (4)
PTUs	0,3721 (1,2)	0,3685 (1)	0,4199 (1,1)	0,088 (1,8)	0,1171 (0,9)	0,1185 (0,9)	0,0809 (1,8)
SSR term	1,0543 (6,7)	1,2143 (4,1)	1,4199 (4,3)	0,2289 (7,5)	0,1478 (1,4)	0,0992 (0,8)	0,1205 (6,8)

Figura 6: Densidad de PQSs en las tres clases de secuencias en *Leishmania major*. **Izquierda**: genoma original. **Derecha**: control (c). Letras diferentes implican diferencias significativas (test *post hoc* con corrección de holm, p<0.05). Asteriscos indican diferencias significativas (*t-test* p<0.05) de cada clase con su control. *Linea roja*: media ponderada de cada clase. (figuras completas en Anexo II, Figura II.2)



Coincidiendo con la densidad general de PQSs en cada genoma, se observa que todas las clases de secuencias de *Leishmania* presentan un mayor contenido de PQSs que sus equivalentes en *Trypanosoma* (Tabla 3).

Leishmania: en los tres genomas se observa que la densidad absoluta de PQSs en las SSRs de terminación es significativamente mayor al del resto de las secuencias y a sus controles (hasta 6 veces mayor). Las SSRs de inicio también presentan una densidad mucho mayor que sus controles (hasta 8 veces mayor) pero, si se consideran valores absolutos, la diferencia no es tan grande

(Anexo I: Tabla 3a-3d). Los PTUs poseen la menor densidad de PQSs y sus valores se ajustan a los controles.

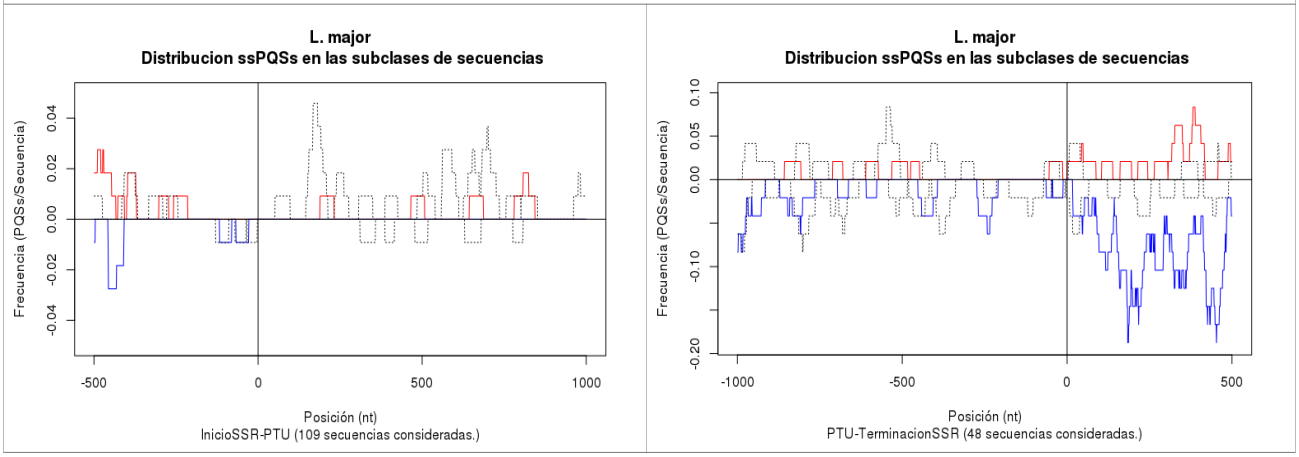
Trypanosoma: en general no hay diferencias estadísticamente significativas entre las distintas clases de secuencias. Si bien en algunos casos se detectan diferencias con los controles, no hay un patrón regular entre los 4 genomas.

La distribución GC entre las SSRs inicio, SSRs terminación y PTUs varía según el genoma considerado. En los tres genomas de *Leishmania*, se observa una clara diferencia en el contenido GC de las secuencias SSR de inicio en comparación con las otras dos secuencias. En los genomas de *Trypanosma* no se da un patrón tan definido en el contenido GC (Anexo I: tabla 4).

II.3) Distribución de PQSs

En la figura 7 se representa la ocurrencia de PQSs a ambos lados de las transiciones SSR inicio-PTU y PTU-SSR terminación en el genoma de *Leishmania major*. La distribución de PQSs en el resto de los genomas se presenta en el anexo II, Figura II.3a).

Figura 7: Distribución relativa de PQSs (PQSs/secuencia) entre clases de secuencias en *Leishmania major*. **Linea roja** PQSs en la hebra codificante. **Linea azul**, PQSs en la hebra molde. **Linea punteada,** PQSs en controles negativos. **Columna izquierda:** transición SSRinicio-PTU. Se representan 500nt del SSRi y 1000nt de la PTU. **Columna derecha:** transición PTU-SSR de terminación. Se representan 1000nt de la PTU y 500nt del SSRt.



-Transcición SSRinicio-PTU: en todos los genomas hay muy pocos PQSs tanto en los PTUs como en las SSR de inicio, coincidiendo con los valores de densidad obtenidos en el apartado II.2. Los controles poseen una densidad ligeramente mayor en los PTUs, principalmente en la hebra codificante.

-Transición PTU-SSRterminación: se observa un patrón distinto entre *Leishmania* y *Trypansoma*. En los primeros, se observan pocos PQSs en el PTU (aunque un poco más que en el extremo de

inicio), seguido de un incremento muy marcado al inicio de la SSR de terminación, principalmente en la hebra molde (>0,15 PQSs/secuencia). Existe una diferencia notoria con los controles, los cuales presentan una mayor cantidad de PQSs en el PTU (sin diferencia marcada entre hebras) y una cantidad menor en la SSR de terminación.

En el caso *Trypanosoma*, el contenido global de PQSs es muy bajo (<0,003 PQSs/secuencia) y no hay un patrón definido en su distribución. Los controles se comportan en forma similar.

En las Figura II.3b del Anexo II se representa la ocurrencia de PQSs en los primeros 2500 pb de la SSR de terminación para las tres especies de *Leishmania*. Se observa que el contenido elevado de PQSs en la hebra molde se mantiene más o menos constante en el rango considerado*.

**Solo se consideran las porciones de la SSRt proximales al PTU.*

Bloque III: Análisis de PTUs

III.1) Densidad de PQSs y contenido GC por tipo de secuencia.

Para cada uno de los tipos de secuencias analizadas (génicas e intergénicas) se estableció la densidad promedio ponderada de PQSs (PQSs/Kb) y contenido GC (Anexo I: tablas 6 y 7; Anexo II: figura II.4).

En la tabla 6 se presentan los valores absolutos de PQSs/Kb y la densidad relativa considerando los controles. En la figura 8 se representan las densidades de PQSs para cada tipo de secuencia en el genoma de *Leishmania major*.

Tabla 6: Densidades medias de PQSs en los distintos tipos de secuencias (génicas e intergénicas) dentro del PTU. Promedio ambas hebras. Unidades: PQS/Kb. Entre paréntesis figura el valor de la relación genoma/control.

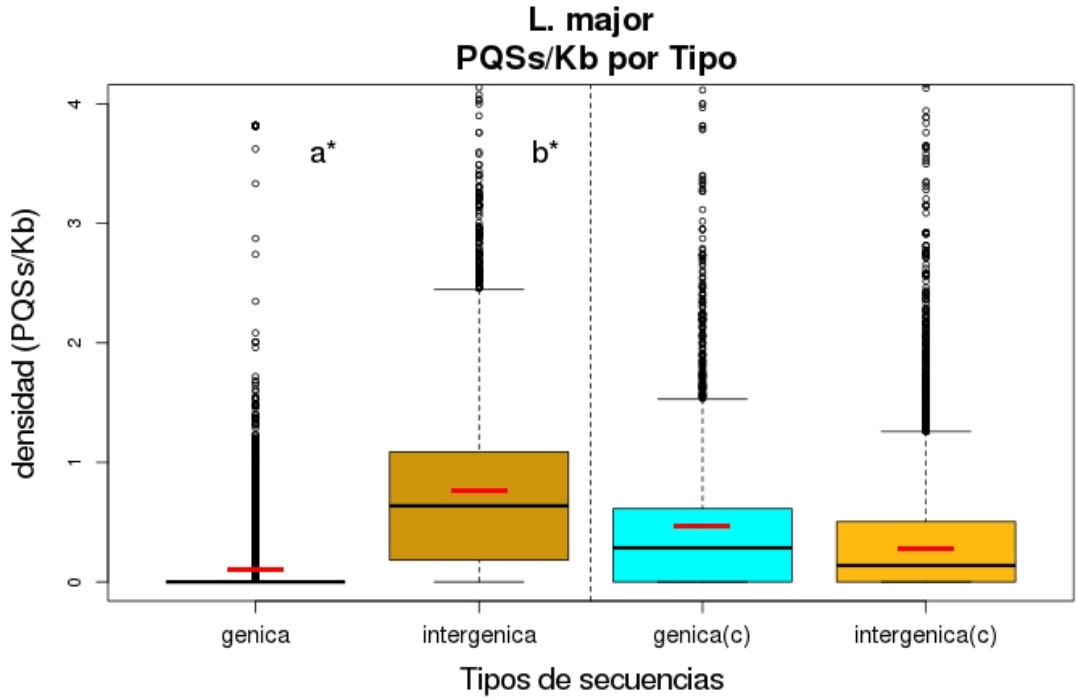
	L.donovani	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi* Elike	*T.cruzi* nElike	*T.evansi*
génica	0,0889 (0,2)	0,0889 (0,2)	0,104 (0,2)	0,0552 (0,8)	0,0709 (0,4)	0,0723 (0,4)	0,0564 (0,8)
intergénica	0,6379 (2,9)	0,6531 (2,5)	0,7583 (2,7)	0,1414 (11,1)	0,205 (3,9)	0,2044 (3,3)	0,1192 (8,6)

Al comparar las regiones génicas con las intergénicas se observa (en todos los genomas) un contenido de PQSs muy superior en estas últimas. Al comparar con los controles, se observa en todos los casos un exceso de PQSs en las secuencias intergénicas y una cantidad inferior a las predichas en las génicas.

Nuevamente, la densidad de PQSs en las secuencias de *Trypanosoma* es inferior a las de *Leishmania*. La principal diferencia entre ambos géneros se da a nivel de las secuencias

intergénicas, con un contenido 4 veces mayor en *Leishmania* respecto a *Trypanosoma*. A nivel de las secuencias génicas, las diferencias entre ambos géneros no son tan marcadas.

Figura 8: Densidad de PQSs para cada tipo de secuencia (génica e intergénica). **Izquierda**: genoma original. **Derecha**: control (c). Letras diferentes indican diferencias estadísticamente significativas entre tipos *(t-test p≤0.05). Asteriscos* indican diferencias significativas *(t-test p≤0.05)* de cada tipo con su control. *Linea roja*: media ponderada de cada tipo.



III.2) Densidad de PQS y contenido G por hebra

Para cada hebra del PTU se estableció la densidad promedio ponderada de PQSs (PQSs/Kb) y contenido G (Anexo I: tablas 8 y 9; Anexo II: Figura II.5).

En la tabla 8 se presentan los valores absolutos de PQSs/Kb y la densidad relativa considerando los controles.

Tabla 8: Densidades medias de PQSs en las distintas hebras de la clase PTU. Unidades: PQS/Kb. Entre paréntesis figura el valor de la relación genoma/control.

	L.donovani	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi* Elike	*T.cruzi* nElike	*T.evansi*
codificante	0,2213 (0,9)	0,2207 (0,8)	0,2593 (0,9)	0,1228 (1,5)	0,1948 (0,9)	0,1954 (0,8)	0,1116 (1,5)
molde	0,5229 (1,3)	0,5163 (1,2)	0,5805 (1,3)	0,0531 (3,8)	0,0395 (1,4)	0,0415 (1,3)	0,0503 (3,7)

En todas las especies de *Leishmania* se observa una mayor cantidad de PQSs en la hebra molde, invirtiéndose este patrón en las especies de *Trypanosoma*. En relación a los controles, se

observan menos PQSs de los esperados en la hebra codificante y más en la molde (salvo en *T. evansi* y *T. brucei*, que poseen más PQSs de los esperados en ambas hebras).

El contenido G es mayor en la hebra molde de *Leishmania* y menor en *Trypanosoma*.

III.3) Densidad de PQS y contenido G por hebra y tipo de secuencia

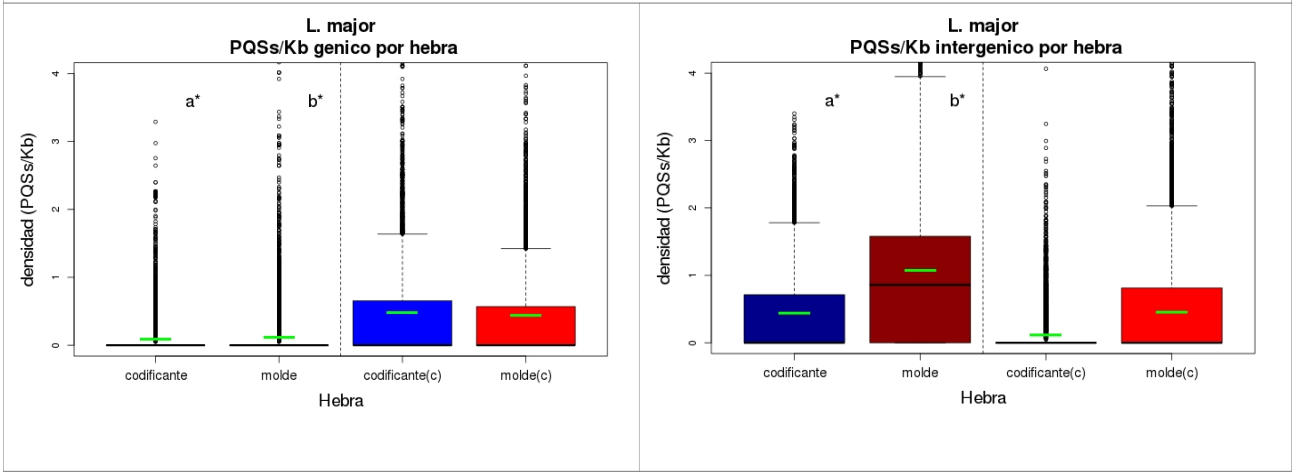
Para cada tipo de secuencia (génica e intergénica) se estableció la densidad de PQSs (PQSs/Kb) y contenido G por hebra (Anexo I: tablas 10 y 11; Anexo II: figuras 6a y 6b).

En la tabla 10 se presentan los valores absolutos de PQSs/Kb y la densidad relativa considerando los controles. En la figura 9 se representan las densidades de PQSs para cada tipo de secuencia y hebra en el genoma de *Leishmania major*.

Tabla 10: Densidades medias de PQSs en las distintas hebras (codificante y molde) de las secuencias génicas e intergénicas Unidades: PQS/Kb. Entre paréntesis figura el valor de la relación genoma/control.

	L.donovani	***L.infantum***	***L.major***	***T.brucei***	***T.cruzi* Elike**	***T.cruzi* nElike**	***T.evansi***
génica codif.	0,0734 (0,2)	0,0739 (0,2)	0,0896 (0,2)	0,0808 (0,6)	0,1211 (0,4)	0,1231 (0,4)	0,0806 (0,7)
génica molde	0,1044 (0,2)	0,1039 (0,2)	0,1183 (0,3)	0,0297 (1,5)	0,0208 (0,6)	0,0216 (0,6)	0,0322 (1,8)
intergénica codif.	0,3621 (4,7)	0,3722 (3,9)	0,4429 (3,9)	0,1922 (9,7)	0,3379 (4,1)	0,3332 (3,4)	0,1614 (8,2)
intergénica molde	0,9138 (2,5)	0,934 (2,2)	1,0736 (2,4)	0,0906 (16,1)	0,0722 (3,29)	0,0757 (2,8)	0,0769 (9,5)

Figura 9: Densidad de PQSs para las distintas hebras de cada tipo de secuencia (génica e intergénica) en Leishmania major. Letras diferentes indican diferencias estadísticamente significativas entre hebras *(paired t-test p≤0.05). Asteriscos* indican diferencias significativas *(t-test p≤0.05)* de cada hebra con su control. *Linea verde*: media ponderada para cada hebra. **Columna izquierda:** PQSs/Kb por hebra para las secuencias génicas y sus controles respectivos (c). **Columna derecha:** PQSs/Kb por hebra para las secuencias intergénicas y sus controles respectivos (c).



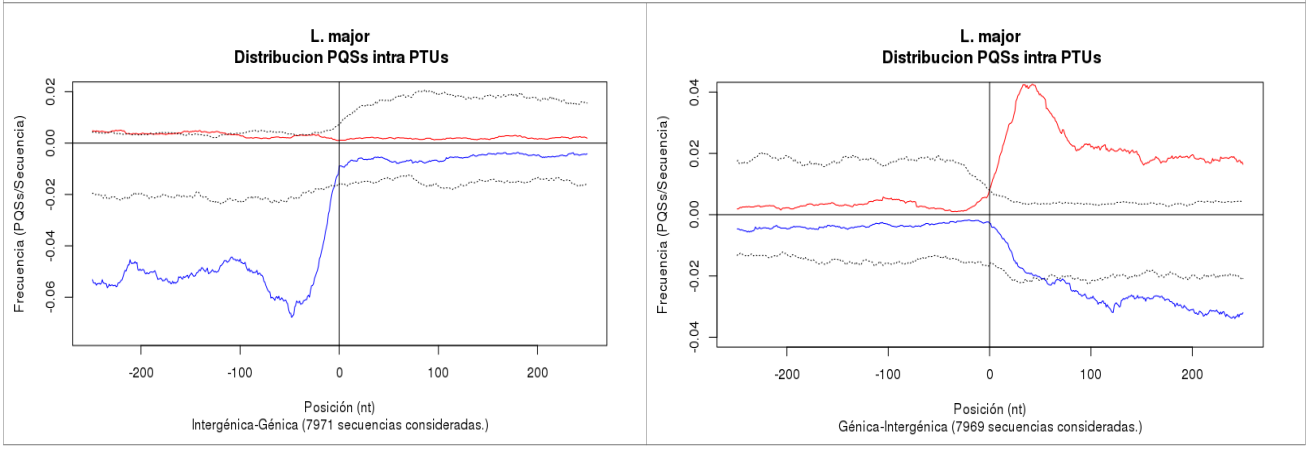
En todas las especies de *Leishmania* se observa una cantidad mayor de PQSs en la hebra molde respecto a la codificante tanto en las secuencias génicas como en las intergénicas (diferencia más evidente en estas últimas). En *Trypanosoma* el comportamiento es inverso para ambos tipos de secuencias (pero con valores absolutos menores).

En *Leishmania* las secuencias génicas poseen valores de PQSs/Kb menores a los controles en ambas hebras y las secuencias intergénicas poseen valores mayores. En valores absolutos, las mayores diferencias con los controles se dan a nivel de la hebra molde intergénica (Anexo I: Tabla 10d). En *Trypanosoma* no se observa un patrón tan definido.

III.4) Distribución de PQSs

En la figura 10 se representa la ocurrencia de PQSs a ambos lados de las transiciones intergénico-génico y génico-intergénico en el genoma de *Leishmania major*. La distribución de PQSs en el resto de los genomas se presenta en el anexo II, Figura II.7).

Figura 10: Distribución relativa de PQSs (PQSs/secuencia) entre tipos de secuencias (génica e intergénica) en Leishmania major. **Linea roja** PQSs en la hebra codificante. **Linea azul**, PQSs en la hebra molde. **Linea punteada,** PQSs en controles negativos. En todos los casos se representan 250nt de cada tipo de secuencia (**Importante**: Las escalas varían entre las Figuras). **Columna izquierda:** transición intergénica-génica. **Columna derecha:** transición génica-intergénica.



Se observan diferencias claras en el patrón de distribución de PQSs entre las regiones génicas e intergénicas en todos los genomas. Las regiones génicas presentan un número de PQSs bajo y relativamente constante (<0,005 PQSs/secuencia) en ambos extremos (250nt). Al comparar con los controles, se observa que en *Leishmania* ambas hebras presentan valores menores a los esperados, mientras que en *Trypanosma* solo la hebra codificante presenta valores menores a los controles. Las regiones intergénicas presentan una mayor cantidad de PQSs, con un patrón que difiere entre los genomas y según el extremo y hebra considerados:

a) Transición intergenica-génica: En *Leishmania* se observa una mayor densidad de PQSs en la hebra molde de la región intergénica (más del doble que en los controles). En *Trypanosoma* no hay un patrón tan definido, pero en general tiende a haber un poco más de densidad de PQSs en la hebra codificante (En ambas hebras hay más PQSs que en los controles).

b) Transición génica-intergénica: en todos los genomas se observa un máximo de PQSs en la hebra codificante de la región intergénica (alrededor de la posición 50). En los controles se observa una clara disminución de la densidad de PQSs en la hebra codificante al inicio de la región intergénica. La densidad de PQSs en la hebra molde no varía en forma tan marcada.

Conclusiones

1.- Los *scripts* desarrollados permitieron definir las distintas clases de secuencias y establecer las densidades y distribución de PQSs en cada una de ellas. El diseño del código permite su aplicación a múltiples genomas en forma simultanea, por lo cual puede ser utilizado para ampliar los análisis a un número mayor de especies.

2.- Ambos géneros analizados presentan grandes diferencias en sus densidades y patrón de distribución de PQSs.

a.-Las especies de *Leishmania* poseen una densidad de PQSs alta y un patrón de distribución bien definido.

b.-Las especies de *Trypanosoma* poseen una densidad de PQSs baja y un patrón de distribución que varía según la especie considerada.

3.- La densidad de PQSs difiere notablemente entre las distintas clases de secuencias. Las diferencias más evidentes se dan en el género *Leishmania*, en el cual se observa que:

a.- Las secuencias SSR de terminación presentan los valores más elevados de PQSs (principalmente en la hebra molde).

b.-Las SSR de inicio y los PTUs presentan valores similares y relativamente bajos.

c.-Tanto las SSR de inicio como las SSR de terminación poseen una densidad de PQSs mayor a la esperada según su contenido GC.

4.- La distribución de PQSs en los PTUs presenta grandes diferencias entre secuencias y hebras.

a.-En todos los genomas, las regiones génicas presentan valores de PQSs/Kb mucho menores que las intergénicas.

b.- La densidad de PQSs siempre es menor que la esperada si la distribución fuera al azar (control) en las regiones génicas y mayor en las intergénicas.

c.- En *Leishmania* la hebra molde de la región intergénica posee la mayor densidad de PQSs (distribuidos en forma más o menos uniforme). En la hebra codificante se observa un máximo de densidad de PQSs al inicio de la misma.

d.-En *Trypanosoma* la mayor densidad de PQSs se da en la hebra codificante de la región intergénica. En general, la distribución de los PQSs en este género no presenta patrones de densidad

y distribución entre secuencias génicas e intergénicas tan evidentes como los observados en *Leishmania*.

Discusión

Las diferencias en el contenido de PQSs totales a nivel genómico entre los géneros *Leishmania* y *Trypanosoma* pueden ser debidas a sus contenidos GC (Huppert et al., 2008; Smargiasso et al., 2009), lo cual concuerda con los valores predichos en el caso de los controles negativos de cada genoma. A pesar de esto, los patrones de distribución de PQSs dentro de cada genoma, bien definidos y regulares entre especies de *Leishmania* y menos definidos entre *Trypanosoma*, podrían indicar que dichas estructuras tienen una mayor implicancia biológica en el primer género.

En *Leishmania*, las grandes diferencias en el patrón de distribución de PQSs entre distintas clases de secuencias podrían ser indicio de la existencia de algún tipo de presión selectiva que tienda a favorecer o impedir la formación de dichas estructuras. De las tres clases de secuencias analizadas, las SSRs de terminación son las que presentan los valores más elevados de PQSs (tanto en relación a las otras clases, como a sus propios controles), lo cual podría estar vinculado a algún mecanismo de terminación de la transcripción. En este sentido, y dado que estas secuencias presentan una gran asimetría en la densidad de PQSs entre hebras, con la mayor parte de los PQSs en su hebra molde (proximal al PTU), se podría plantear que la formación de estructuras de cuádruples de guanina en la hebra molde interferiría con el avance de la ARN polimerasa, provocando o facilitando la terminación de la transcripción del ARN policistrónico (Holder & Hartig, 2014; Agarwal et al., 2014; Smargiasso et al., 2009).

Al analizar la distribución de PQSs entre las regiones génicas (CDSs) e intergénicas, se observa en todos los casos que las primeras contienen las densidades más bajas de PQSs (principalmente en la hebra codificante). Este patrón también ha sido descrito en otras especies y estaría vinculado a evitar impedimentos estéricos tanto al avance de la ARN polimerasa durante la transcripción, como al avance del ribosoma sobre el ARNm durante la traducción (Agarwal et al., 2014; Endoh et al., 2013; Harris & Merrick, 2015). Por otro lado, la mayor densidad de PQSs en las secuencias intergénicas, con un patrón bastante conservado en *Leishmania*, también podría estar vinculado a mecanismos de regulación de la expresión génica. Es importante recordar que las regiones intergénicas consideradas en este trabajo contienen las secuencias UTRs 5’ y 3’ de cada gen (ver métodos) y se ha observado que ambas secuencias desempeñan roles muy importantes en la regulación de la expresión génica a distintos niveles (Beaudoin & Perreault, 2010; Bhartiya et al., 2017; Endoh et al., 2013; Huppert et al., 2008). Por ejemplo, a nivel postranscripcional, se ha

observado que las secuencias UTRs-5’ regulan el inicio de la traducción (Agarwal et al., 2014; Endoh et al., 2013), mientras que las UTRs-3’ impactan en el procesamiento del ARN (Huppert et al., 2008; Song et al., 2016), su estabilidad y eficiencia de traducción. Como se mencionó en la introducción, en tripanosomátidos, no existen los mecanismos clásicos de control transcripcional de eucariotas y el control fino de la expresión de genes codificantes para proteínas se da fundamentalmente a nivel post-transcripcional. En este contexto, es muy probable que la presencia de G4s en las regiones 5´y 3´UTR de los mRNAs estén implicados en estos mecanismos de control (Folgueira et al., 2005; Martinez-Calvillo et al., 2010; McNicoll et al., 2005; Parsons y Miller, 2016).

La ocurrencia de un pico de PQSs en los primeros 50nt del UTR-3’ en las tres especies de *Leishmania* no deja de ser muy llamativo. Las densidades observadas de PQSs en esta región (PQSs/secuencia) permiten estimar que solo unas 400 a 500 secuencias del total contendrían dichas estructuras en la posición considerada. Es posible que estas secuencias correspondan a genes funcionalmente relacionados y que los mismos sufran una regulación conjunta de su expresión, en lo que se conoce como “operón transcripcional” (De Gaudenzi et al., 2011). Mecanismos como este ya han sido descritos en *Plasmodium*, donde se observa una regulación coordinada y dependiente de PQSs de distintos grupos de genes (Bhartiya et al., 2017; Smargiasso et al., 2009).

La mayor parte de los estudios sobre los cuádruples de guanina están enfocados a aquellos formados intra-molecularmente (una hebra participante). Sin embargo, la formación de cuádruples inter-moleculares (dos hebras participantes) también podría ser un evento muy frecuente (Kudlicki, 2016). Si bien en el presente trabajo solo se han considerado los PQSs intra-moleculares, la modificación de algunos parámetros de las funciones utilizadas permitiría repetir el análisis con PQSs inter-moleculares.

Bibliografía

Agarwal, T., Roy, S., Kumar, S., Chakraborty, T. K., & Maiti, S. (2014). In the sense of transcription regulation by G-quadruplexes: asymmetric effects in sense and antisense strands. *Biochemistry*, *53*(23), 3711–3718. https://doi.org/10.1021/bi401451q

Aslett, M., Aurrecoechea, C., Berriman, M., Brestelli, J., Brunk, B. P., Carrington, M., … Wang, H. (2010). TriTrypDB: a functional genomic resource for the Trypanosomatidae. *Nucleic Acids Research*, *38*(October 2009), 457–462. https://doi.org/10.1093/nar/gkp851

Beaudoin, J.-D., & Perreault, J.-P. (2010). 5'-UTR G-quadruplex structures acting as translational repressors. *Nucleic Acids Research*, *38*(20), 7022–7036. https://doi.org/10.1093/nar/gkq557

Berriman, M., Ghedin, E., Hertz-fowler, C., Blandin, G., Renauld, H., Bartholomeu, D. C., … El-Sayed, N. M. (2005). The genome of the African trypanosome Trypanosoma brucei. *Science (New York, N.Y.)*, *309*(5733), 416–422. https://doi.org/10.1126/science.1112642

Bhartiya, D., Chawla, V., Ghosh, S., Shankar, R., & Kumar, N. (2017). Genome-wide regulatory dynamics of G-quadruplexes in human malaria parasite Plasmodium falciparum. *Genomics*, *108*(5–6), 224–231. https://doi.org/10.1016/j.ygeno.2016.10.004

Cammas, A., & Millevoi, S. (2016). SURVEY AND SUMMARY RNA G-quadruplexes: emerging mechanisms in disease, *45*(4), 1584–1595. https://doi.org/10.1093/nar/gkw1280

Cammas, A., & Millevoi, S. (2017). RNA G-quadruplexes: emerging mechanisms in disease. *Nucleic Acids Research*, *45*(4), 1584–1595. https://doi.org/10.1093/nar/gkw1280

Capra, J. A., Paeschke, K., Singh, M., & Zakian, V. A. (2010). G-quadruplex DNA sequences are evolutionarily conserved and associated with distinct genomic features in Saccharomyces cerevisiae. *PLoS Computational Biology*, *6*(7), e1000861. https://doi.org/10.1371/journal.pcbi.1000861

De Gaudenzi, J. G., Noe, G., Campo, V. A., Frasch, A. C., & Cassola, A. (2011). Gene expression regulation in trypanosomatids. *Essays in Biochemistry*, *51*, 31–46. https://doi.org/10.1042/bse0510031

Dillon, L. A. L., Okrah, K., Hughitt, V. K., Suresh, R., Li, Y., Fernandes, M. C., … El-Sayed, N. M. (2015). Transcriptomic profiling of gene expression and RNA processing during Leishmania major differentiation. *Nucleic Acids Research*, *43*(14), 6799–6813. https://doi.org/10.1093/nar/gkv656

Du, Z., Zhao, Y., & Li, N. (2008). Genome-wide analysis reveals regulatory role of G4 DNA in gene transcription. *Genome Research*, *18*(2), 233–241. https://doi.org/10.1101/gr.6905408

Duhagon, M. A., Smircich, P., Forteza, D., Naya, H., Williams, N., & Garat, B. (2011). Comparative genomic analysis of dinucleotide repeats in Tritryps. *Gene*, *487*(1), 29–37. https://doi.org/10.1016/j.gene.2011.07.022

El-Sayed, N. M., Myler, P. J., Bartholomeu, D. C., Nilsson, D., Aggarwal, G., Tran, A.-N., … Andersson, B. (2005). The genome sequence of Trypanosoma cruzi, etiologic agent of Chagas disease. *Science (New York, N.Y.)*, *309*(5733), 409–415. https://doi.org/10.1126/science.1112631

Endoh, T., Kawasaki, Y., & Sugimoto, N. (2013). Suppression of gene expression by G-quadruplexes in open reading frames depends on G-quadruplex stability. *Angewandte Chemie (International Ed. in English)*, *52*(21), 5522–5526. https://doi.org/10.1002/anie.201300058

Folgueira, C., Quijada, L., Soto, M., Abanades, D. R., Alonso, C., & Requena, J. M. (2005). The translational efficiencies of the two Leishmania infantum HSP70 mRNAs, differing in their 3'-untranslated regions, are affected by shifts in the temperature of growth through different mechanisms. *The Journal of Biological Chemistry*, *280*(42), 35172–35183. https://doi.org/10.1074/jbc.M505559200

Frees, S., Menendez, C., Crum, M., & Bagga, P. S. (2014). QGRS-Conserve: a computational method for discovering evolutionarily conserved G-quadruplex motifs. *Human Genomics*, *8*, 8. https://doi.org/10.1186/1479-7364-8-8

Harris, L. M., & Merrick, C. J. (2015). G-quadruplexes in pathogens: a common route to virulence control? *PLoS Pathogens*, *11*(2), e1004562. https://doi.org/10.1371/journal.ppat.1004562

Holder, I. T., & Hartig, J. S. (2014). A matter of location: influence of G-quadruplexes on Escherichia coli gene expression. *Chemistry & Biology*, *21*(11), 1511–1521. https://doi.org/10.1016/j.chembiol.2014.09.014

Huppert, J. L., Bugaut, A., Kumari, S., & Balasubramanian, S. (2008). G-quadruplexes: the beginning and end of UTRs. *Nucleic Acids Research*, *36*(19), 6260–6268. https://doi.org/10.1093/nar/gkn511

Ivens, A. C., Peacock, C. S., Worthey, E. A., Murphy, L., Aggarwal, G., Berriman, M., … Myler, P. J. (2005). The genome of the kinetoplastid parasite, Leishmania major. *Science (New York, N.Y.)*, *309*(5733), 436–442. https://doi.org/10.1126/science.1112680

Kolev, N. G., Franklin, J. B., Carmi, S., Shi, H., Michaeli, S., & Tschudi, C. (2010). The transcriptome of the human pathogen Trypanosoma brucei at single-nucleotide resolution. *PLoS Pathogens*, *6*(9), e1001090. https://doi.org/10.1371/journal.ppat.1001090

Kudlicki, A. S. (2016). G-Quadruplexes Involving Both Strands of Genomic DNA Are Highly Abundant and Colocalize with Functional Sites in the Human Genome. *PloS One*, *11*(1), e0146174. https://doi.org/10.1371/journal.pone.0146174

Landfear, S. M. (2003). Trypanosomatid transcription factors: waiting for Godot. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(1), 7–9. https://doi.org/10.1073/pnas.0337508100

Maizels, N., & Gray, L. T. (2013). The G4 Genome, *9*(4). https://doi.org/10.1371/journal.pgen.1003468

Maree, J. P., & Patterton, H.-G. (2014). The epigenome of Trypanosoma brucei: a regulatory interface to an unconventional transcriptional machine. *Biochimica et Biophysica Acta*, *1839*(9), 743–750. https://doi.org/10.1016/j.bbagrm.2014.05.028

Martinez-Calvillo, S., Yan, S., Nguyen, D., Fox, M., Stuart, K., & Myler, P. J. (2003). Transcription of Leishmania major Friedlin chromosome 1 initiates in both directions within a single region. *Molecular Cell*, *11*(5), 1291–1299.

McNicoll, F., Muller, M., Cloutier, S., Boilard, N., Rochette, A., Dube, M., & Papadopoulou, B. (2005). Distinct 3'-untranslated region elements regulate stage-specific mRNA accumulation and translation in

Leishmania. *The Journal of Biological Chemistry*, *280*(42), 35238–35246. https://doi.org/10.1074/jbc.M507511200

Mukundan, V. T., & Phan, A. T. (2013). Bulges in G-quadruplexes: broadening the definition of G-quadruplex-forming sequences. *Journal of the American Chemical Society*, *135*(13), 5017–5028. https://doi.org/10.1021/ja310251r

Myler, P. J., Beverley, S. M., Cruz, A. K., Dobson, D. E., Ivens, A. C., McDonagh, P. D., … Stuart, K. D. (2001). The Leishmania genome project: new insights into gene organization and function. *Medical Microbiology and Immunology*, *190*(1–2), 9–12.

Olsen, O. W. (1974). *Animal Parasites: Their Life Cycles and Ecology* (6th ed.). Baltimore: University Park Press.

Parsons, M., & Myler, P. J. (2016). Illuminating Parasite Protein Production by Ribosome Profiling. *Trends in Parasitology*, *32*(6), 446–457. https://doi.org/10.1016/j.pt.2016.03.005

Parsons, M., Worthey, E. A., Ward, P. N., & Mottram, J. C. (2005). BMC Genomics, *19*, 1–19. https://doi.org/10.1186/1471-2164-6-127

Respuela, P., Ferella, M., Rada-Iglesias, A., & Aslund, L. (2008). Histone acetylation and methylation at sites initiating divergent polycistronic transcription in Trypanosoma cruzi. *The Journal of Biological Chemistry*, *283*(23), 15884–15892. https://doi.org/10.1074/jbc.M802081200

Siegel, T. N., Hekstra, D. R., Kemp, L. E., Figueiredo, L. M., Lowell, J. E., Fenyo, D., … Cross, G. A. M. (2009). Four histone variants mark the boundaries of polycistronic transcription units in Trypanosoma brucei. *Genes & Development*, *23*(9), 1063–1076. https://doi.org/10.1101/gad.1790409

Siegel, T. N., Hekstra, D. R., Wang, X., Dewell, S., Cross, G. A. M., Gunasekera, K., … Ochsenreiter, T. (2011). Gene expression in Trypanosoma brucei: lessons from high-throughput RNA sequencing. *Trends in Parasitology*, *27*(10), 434–441. https://doi.org/10.1016/j.pt.2011.05.006

Smargiasso, N., Gabelica, V., Damblon, C., Rosu, F., De Pauw, E., Teulade-Fichou, M.-P., … Claessens, A. (2009). Putative DNA G-quadruplex formation within the promoters of Plasmodium falciparum var genes. *BMC Genomics*, *10*, 362. https://doi.org/10.1186/1471-2164-10-362

Smircich, P., Forteza, D., El-sayed, N. M., & Garat, B. (2013b). Genomic analysis of sequence-dependent DNA curvature in Leishmania. *PloS One*, *8*(4), e63068. https://doi.org/10.1371/journal.pone.0063068

Song, J., Perreault, J., Topisirovic, I., & Richard, S. (2016). RNA G-quadruplexes and their potential regulatory roles in translation. *Translation (Austin, Tex.)*, *4*(2), e1244031. https://doi.org/10.1080/21690731.2016.1244031

Especialización en Bioinformática
Trabajo Final

Anexo I: Tablas

Tabla 1: Datos generales de los genomas. (C) Tamaño genómico, (N.percent) % de bases sin determinar, (GC.percent) % GC, (PQSs) número total de cuádruples, (PQSs.Kb) cuádruples cada 1000 bases, (PQSs.Kb.control) cuádruples cada 1000 bases en los controles, (PQSs.Kb/PQSs.Kb.control) contenido PQSs en los genomas en relación a los controles.

1	C(pb)	n	N.percent	GC.percent	PQSs	PQSs.Kb	PQSs.Kb.control	PQSs.Kb/PQSs.Kb.control
Control	840	2	14,3	55,6	4	5,556	5,556	1,000
L.donovani	32444968	36	3,7	59,5	23609	0,755	0,632	1,195
L.infantum	32103026	60	0,1	59,6	25688	0,801	0,706	1,135
L.major	32855095	36	0	59,7	28702	0,874	0,73	1,197
T.brucei	22148088	11	0,2	47,2	4041	0,183	0,095	1,926
***T.cruzi*Elike**	32529070	41	20,6	50,4	6339	0,245	0,254	0,965
***T.cruzin*Elike**	32529072	41	14,7	50,7	6743	0,243	0,266	0,914
T.evansi	25432160	13	0	46,5	4274	0,168	0,088	1,909

Tabla 2a: Número de clases de secuencias para los distintos genomas. PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

2a	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruziElike*	*T.cruzinElike*	*T.evansi*
PTUs	6	187	215	194	282	427	418	422
init-SSR	2	81	85	89	138	209	203	203
init-tel	2	23	43	16	5	9	12	13
term-SSR	2	68	68	69	133	177	174	205
term-tel	2	49	75	56	15	73	70	12

Tabla 2b: Tamaño medio (pb) y desvío estandar para cada clase de secuencia en los distintos genomas. PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

2b	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruziElike*	*T.cruzinElike*	*T.evansi*
mean PTUs	60	169197	145269	166072	74908	66370	69290	57947
mean init-SSR	60	3109	2943	2481	4049	8571	7327	2481
mean init-tel	60	3499	3266	3330	1895	40422	20846	3778
mean term-SSR	60	3744	3003	2123	3221	5208	3713	1922
mean term-tel	60	4453	3671	3865	1835	15237	16891	2653
sd PTUs	0	172773	168477	180077	100793	85518	88261	92981
sd init-SSR	0	3361	3216	2695	3822	40341	39590	2853
sd init-tel	0	2439	3522	1355	2563	74362	43991	4593
sd term-SSR	0	4136	3394	2504	4350	19492	9933	1893
sd term-tel	0	5315	3763	3771	1995	30572	28853	3657

Tabla 2c: media y desvío estándar del número de secuencias codificantes en las PTUs de los distintos genomas.

2c	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruziElike*	*T.cruzinElike*	*T.evansi*
mean seq/PTU	2	43,48	38,88	48,22	28,65	24,77	26,53	23,94
sd seq/PTU	0	41,21	41,62	53,39	38,10	31,54	34,36	37,71

Tabla 3a: densidades medias de PQSs en las distintas clases de secuencias. Promedio ambas hebras. Unidades: PQS/Kb.
PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

3a	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
init-SSR	4,1667	0,5518	0,5243	0,5435	0,1211	0,0993	0,0939	0,1003
init-tel	0	0,3107	2,395	2,252	0	0,038	0,8559	0,2952
PTUs	3,3333	0,3721	0,3685	0,4199	0,088	0,1171	0,1185	0,0809
term-SSR	4,1667	1,0543	1,2143	1,4199	0,2289	0,1478	0,0992	0,1205
term-tel	0	0,4368	2,2806	1,7302	0,0182	1,8439	1,39	1,4137

Tabla 3b: densidades medias de PQSs en las distintas clases de secuencias de los controles. Promedio ambas hebras. Unidades: PQS/Kb.
PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

3b (control)	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
init-SSR	4,1667	0,0638	0,082	0,0747	0,0197	0,08	0,0781	0,0248
init-tel	0	0,0636	0,4063	0,1032	0	0	0,0734	0
PTUs	3,3333	0,3204	0,3543	0,3689	0,0488	0,1286	0,1345	0,0447
term-SSR	4,1667	0,1574	0,2974	0,3277	0,0305	0,1062	0,1205	0,0178
term-tel	0	0,1016	0,4565	0,1548	0	0,043	0,0278	0,0628

Tabla 3c: contenido PQSs en las distintas clases de secuencias en relación a los controles (PQSs genomas/PQSs controles).
PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

3c (genoma/control)	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
init-SSR	1,0000	8,6489	6,3939	7,2758	6,1472	1,2413	1,2023	4,0444
init-tel	NA	4,8852	5,8947	21,8217	NA	NA	11,6608	NA
PTUs	1,0000	1,1614	1,0401	1,1382	1,8033	0,9106	0,8810	1,8098
term-SSR	1,0000	6,6982	4,0831	4,3329	7,5049	1,3917	0,8232	6,7697
term-tel	NA	4,2992	4,9958	11,1770	NA	42,8814	50,0000	22,5111

Tabla 3d: diferencia en el contenido de PQSs entre los genomas y los controles (PQSs genomas – PQSs controles).
PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

3d (genoma-control)	*Control*	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
init-SSR	0,0000	0,4880	0,4423	0,4688	0,1014	0,0193	0,0158	0,0755
init-tel	0,0000	0,2471	1,9887	2,1488	0,0000	0,0380	0,7825	0,2952
PTU	0,0000	0,0517	0,0142	0,0510	0,0392	-0,0115	-0,0160	0,0362
term-SSR	0,0000	0,8969	0,9169	1,0922	0,1984	0,0416	-0,0213	0,1027
term-tel	0,0000	0,3352	1,8241	1,5754	0,0182	1,8009	1,3622	1,3509

Tabla 4: Contenido GC (%) en cada clase de secuencia para los genomas considerados.
PTUs: Unidades de transcripción policistrónicas; init-SSR: regiones de cambio de hebra de inicio (divergentes); term-SSR: regiones de cambio de hebra de terminación (convergentes); init-tel: región telomérica de inicio; term-tel: región telomérica de terminación.

4	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
init-SSR	50	52,9	53,4	52,7	46,4	50,7	51,3	44,8
init-tel	66,7	52	53,4	52,2	44,2	52,4	48,5	39,9
PTUs	53,3	59,6	59,7	59,8	47,2	50,3	50,6	46,6
term-SSR	50	58,9	58,9	60	45,3	53,9	54,1	43,9
term-tel	66,7	57,5	54,7	54,9	45,1	48,6	49,7	39,9

Tabla 5: Número total de secuencias génicas e intergénicas en cada genoma.

5	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	12	8122	8350	9351	8076	10571	11082	10091
intérgenica	6	7943	8145	9102	7795	10150	10671	9678

Tabla 6a: Densidades medias de PQSs en los distintos tipos de secuencias (génicas e intergénicas) dentro del PTU. Promedio ambas hebras. Unidades: PQS/Kb.

6a	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	10	0,0889	0,0889	0,104	0,0552	0,0709	0,0723	0,0564
intergénica	10	0,6379	0,6531	0,7583	0,1414	0,205	0,2044	0,1192

Tabla 6b: Densidades medias de PQSs en los distintos tipos de secuencias (génicas e intergénicas) dentro del PTU para los controles. Promedio ambas hebras. Unidades: PQS/Kb.

6b (control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	10	0,4417	0,4629	0,4611	0,0735	0,1762	0,1816	0,0676
intergénica	10	0,2186	0,2571	0,2833	0,0127	0,0525	0,0622	0,0139

Tabla 6c: contenido PQSs en los distintos tipos de secuencias en relación a los controles (PQSs genomas/PQSs controles).

6c (genoma/con	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	1,0000	0,2013	0,1921	0,2255	0,7510	0,4024	0,3981	0,8343
intergénica	1,0000	2,9181	2,5403	2,6767	11,1339	3,9048	3,2862	8,5755

Tabla 6d: diferencia en el contenido de PQSs entre los genomas y los controles (PQSs genomas – PQSs controles).

6d (genoma-con	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	0	-0,3528	-0,374	-0,3571	-0,0183	-0,1053	-0,1093	-0,0112
intergénica	0	0,4193	0,396	0,475	0,1287	0,1525	0,1422	0,1053

Tabla 7: Contenido GC (%) en cada tipo de secuencia para los genomas considerados.

7	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica	53	62,4	62,5	62,4	50,9	53	53,1	50,4
intergénica	54	57	56,9	57	41,5	45,7	46,6	41,2

Tabla 8a: Densidades medias de PQSs en las distintas hebras de la clase PTU. Unidades: PQS/Kb.

8a	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
codificante	3,3333	0,2213	0,2207	0,2593	0,1228	0,1948	0,1954	0,1116
molde	3,3333	0,5229	0,5163	0,5805	0,0531	0,0395	0,0415	0,0503

Tabla 8b: Densidades medias de PQSs en las distintas hebras de la clase PTU en los controles. Unidades: PQS/Kb.

8b (control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
codificante	3,3333	0,2529	0,2829	0,2982	0,0837	0,2285	0,2375	0,0758
molde	3,3333	0,3878	0,4258	0,4395	0,0140	0,0286	0,0314	0,0136

Tabla 8c: contenido PQSs en las distintas hebras de la clase PTU en relación a los controles (PQSs genomas/PQSs controles).

8c (genoma/control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
codificante	1,0000	0,8750	0,7801	0,8696	1,4671	0,8525	0,8227	1,4723
molde	1,0000	1,3484	1,2125	1,3208	3,7929	1,3811	1,3217	3,6985

Tabla 8d: diferencia en el contenido de PQSs por hebra en la clase PTU entre los genomas y los controles (PQSs genomas – PQSs controles).

8d (genoma-control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
codificante	0,0000	-0,0316	-0,0622	-0,0389	0,0391	-0,0337	-0,0421	0,0358
molde	0,0000	0,1351	0,0905	0,1410	0,0391	0,0109	0,0101	0,0367

Tabla 9: Contenido GC (%) en cada hebra de a clase PTU.

9	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
codificante	26,7	28,7	28,8	28,9	25,5	27,8	28	25,1
molde	26,7	30,9	30,9	30,9	21,7	22,5	22,7	21,5

Tabla 10a: Densidades medias de PQSs en las distintas hebras (codificante y molde) de las secuencias génicas e intergénicas en la clase PTU. Unidades: PQS/Kb.

10a	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica codif.	10,0000	0,0734	0,0739	0,0896	0,0808	0,1211	0,1231	0,0806
génica molde	10,0000	0,1044	0,1039	0,1183	0,0297	0,0208	0,0216	0,0322
intergénica codif.	10,0000	0,3621	0,3722	0,4429	0,1922	0,3379	0,3332	0,1614
intergénica molde	10,0000	0,9138	0,9340	1,0736	0,0906	0,0722	0,0757	0,0769

Tabla 10b: Densidades medias de PQSs en las distintas hebras (codificante y molde) de las secuencias génicas e intergénicas en la clase PTU para los controles. Unidades: PQS/Kb.

10b (controles)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica codif.	10,0000	0,4507	0,4776	0,4766	0,1271	0,3187	0,3276	0,1173
génica molde	10,0000	0,4327	0,4481	0,4456	0,0198	0,0337	0,0356	0,0179
intergénica codif.	10,0000	0,0768	0,0956	0,1148	0,0198	0,0830	0,0979	0,0198
intergénica molde	10,0000	0,3605	0,4187	0,4517	0,0056	0,0219	0,0266	0,0081

Tabla 10c: contenido PQSs en las distintas hebras (codificante y molde) de las secuencias génicas e intergénicas en la clase PTU en relación a los controles (PQSs genomas/PQSs controles).

10c (genoma/control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica codif.	1,0000	0,1629	0,1547	0,1880	0,6357	0,3800	0,3758	0,6871
génica molde	1,0000	0,2413	0,2319	0,2655	1,5000	0,6172	0,6067	1,7989
intergénica codif.	1,0000	4,7148	3,8933	3,8580	9,7071	4,0711	3,4035	8,1515
intergénica molde	1,0000	2,5348	2,2307	2,3768	16,1786	3,2968	2,8459	9,4938

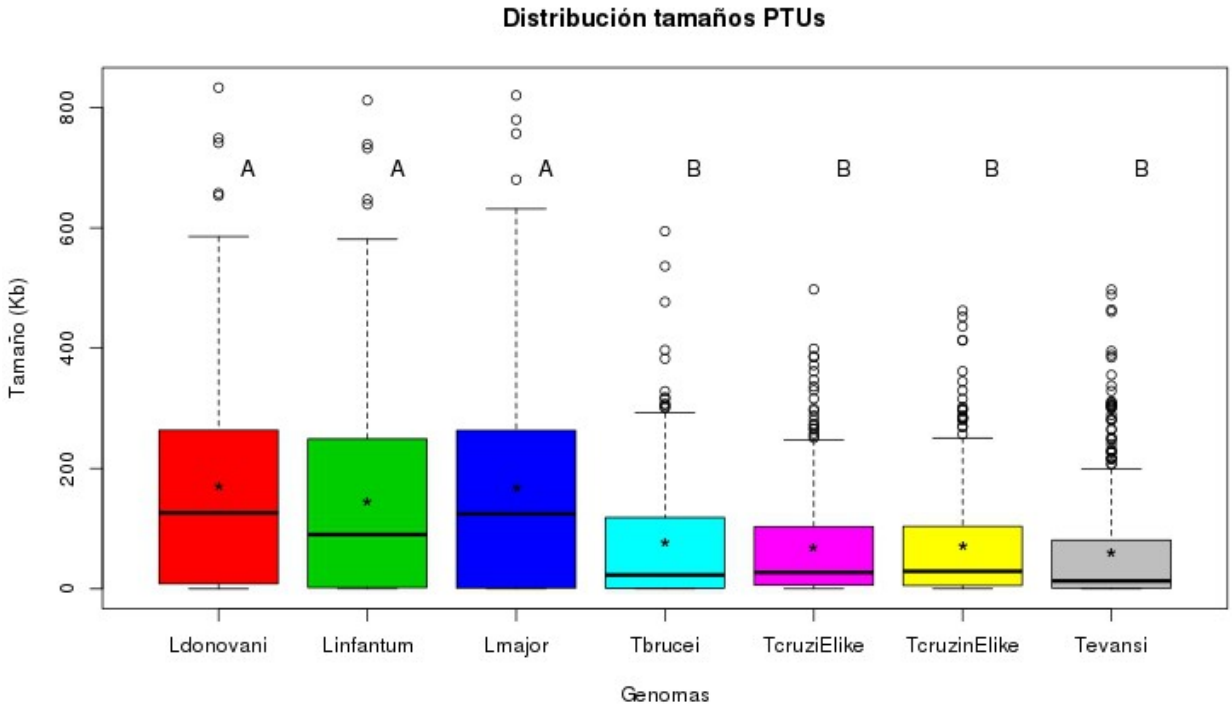
Tabla 10d: diferencia en el contenido de PQSs por hebra y tipo de secuencia en la clase PTU entre los genomas y los controles (PQSs genomas – PQSs controles).

10d (genoma-control)	Control	*L.donovani*	*L.infantum*	*L.major*	*T.brucei*	*T.cruzi*Elike	*T.cruzi*nElike	*T.evansi*
génica codif.	0,0000	-0,3773	-0,4037	-0,3870	-0,0463	-0,1976	-0,2045	-0,0367
génica molde	0,0000	-0,3283	-0,3442	-0,3273	0,0099	-0,0129	-0,0140	0,0143
intergénica codif.	0,0000	0,2853	0,2766	0,3281	0,1724	0,2549	0,2353	0,1416
intergénica molde	0,0000	0,5533	0,5153	0,6219	0,0850	0,0503	0,0491	0,0688

Especialización en Bioinformática
Trabajo Final

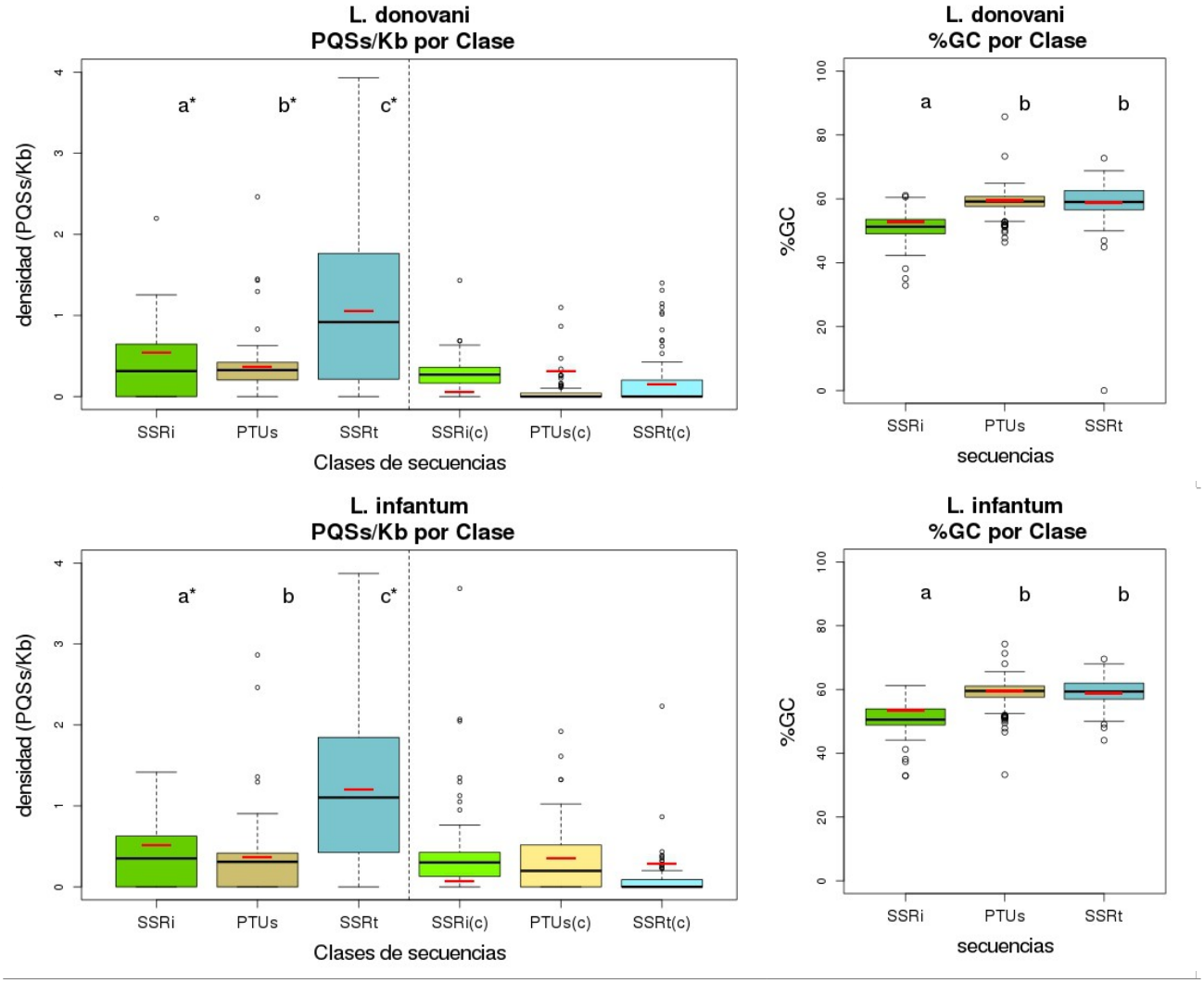
Anexo II: Gráficos

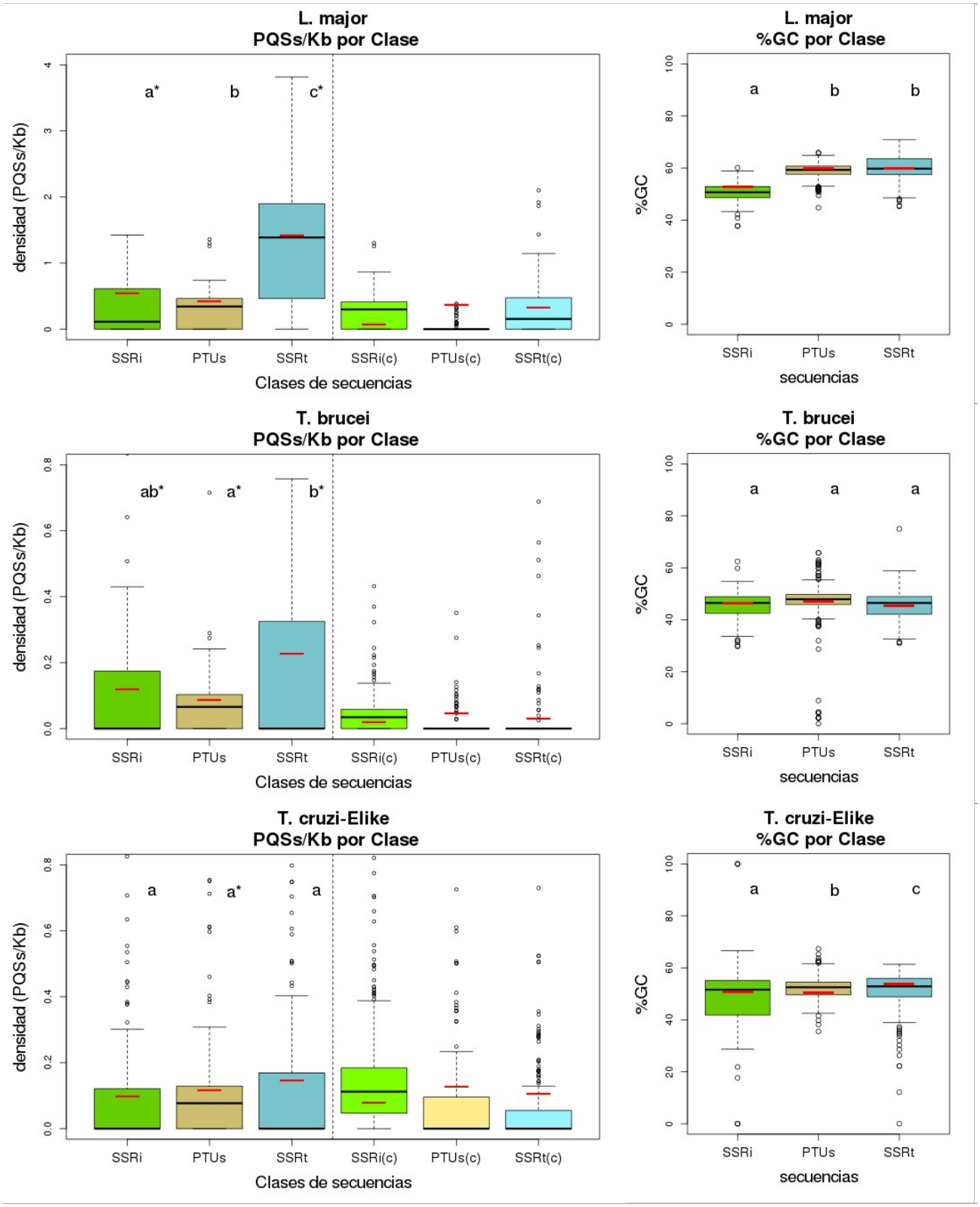
Figura II.1: Distribución de tamaños (Kb) de PTUs en los distintos genomas. Se observan diferencias estadísticamente significativas (test de Welch, $p<0.05$). Letras diferentes implican diferencias significativas (test *post hoc* con corrección de holm, $p<0.05$). *Asteriscos*: medias de tamaño para cada genoma.

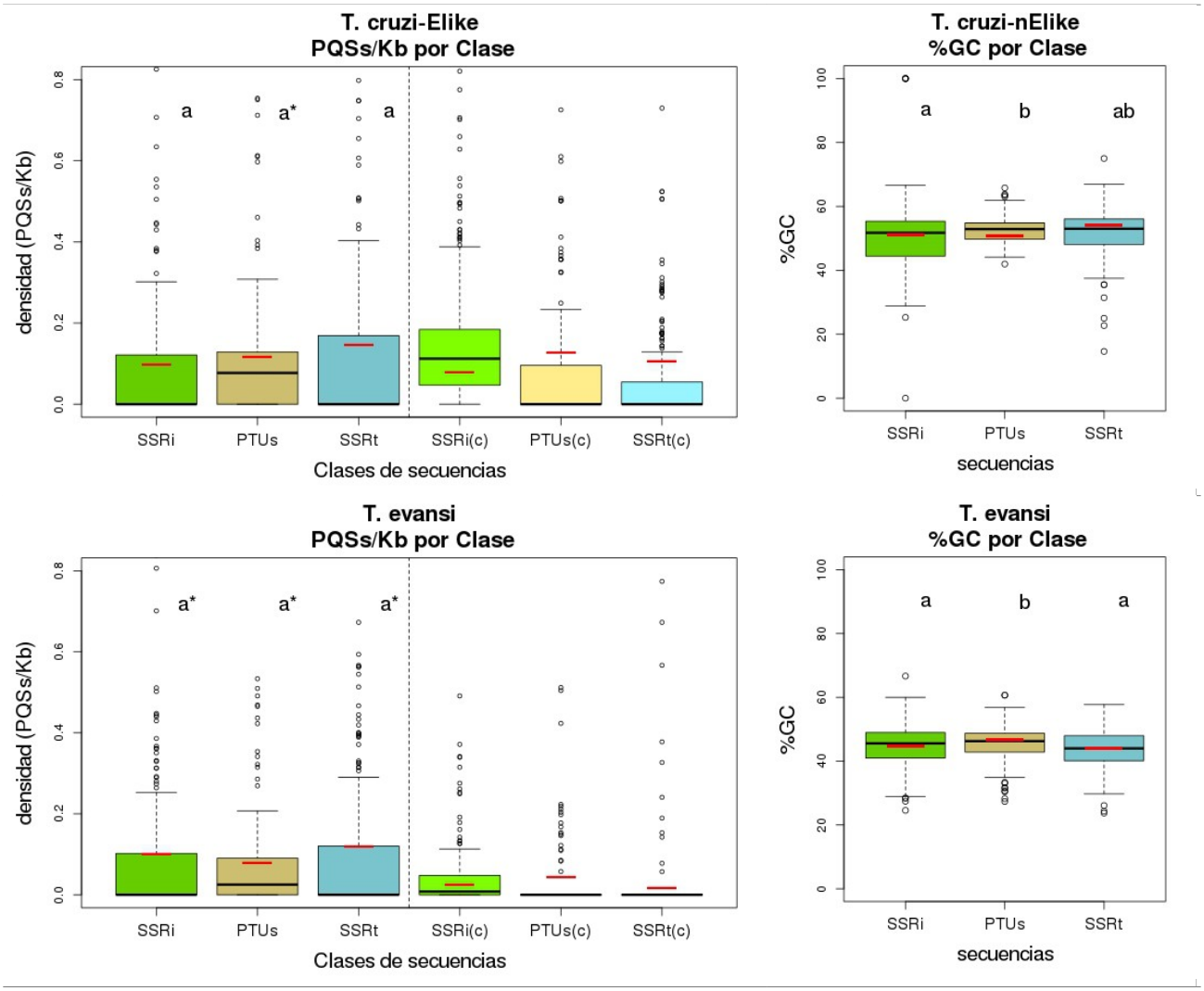


(II.2)

Figura II.2: Densidad de PQSs y contenido GC en genomas y controles. Letras diferentes indican diferencias estadísticamente significativas entre clases (test *post hoc* con corrección de holm, p<0.05). Asteriscos indican diferencias significativas de cada clase con su control (*t-test* p<0.05) . Linea roja: media ponderada de cada clase. **Columna izquierda:** PQSs/Kb para los genomas y sus controles respectivos (c). **Columna derecha:** %GC de las distintas clases de secuencias.




L. major
PQSs/Kb por Clase
densidad (PQSs/Kb)
a* b c*
SSRi PTUs SSRt SSRi(c) PTUs(c) SSRt(c)
Clases de secuencias
L. major
%GC por Clase
%GC
a b b
SSRi PTUs SSRt
secuencias
T. brucei
PQSs/Kb por Clase
densidad (PQSs/Kb)
ab* a* b*
SSRi PTUs SSRt SSRi(c) PTUs(c) SSRt(c)
Clases de secuencias
T. brucei
%GC por Clase
%GC
a a a
SSRi PTUs SSRt
secuencias
T. cruzi-Elike
PQSs/Kb por Clase
densidad (PQSs/Kb)
a a* a
SSRi PTUs SSRt SSRi(c) PTUs(c) SSRt(c)
Clases de secuencias
T. cruzi-Elike
%GC por Clase
%GC
a b c
SSRi PTUs SSRt
secuencias

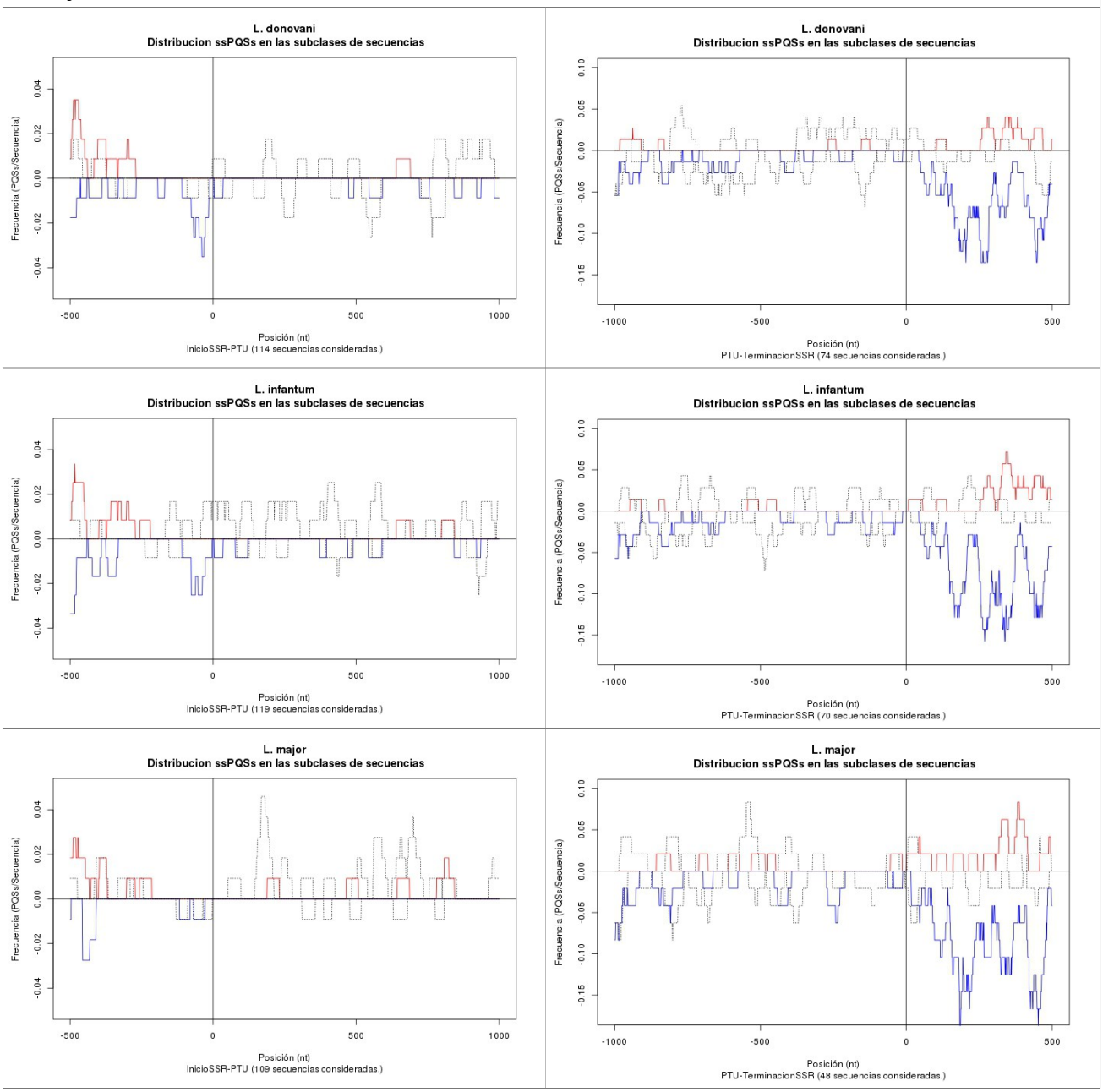

T. cruzi-Elike
PQSs/Kb por Clase
densidad (PQSs/Kb)
a
a*
a
SSRi
PTUs
SSRt
SSRi(c)
PTUs(c)
SSRt(c)
Clases de secuencias
T. cruzi-nElike
%GC por Clase
%GC
a
b
ab
SSRi
PTUs
SSRt
secuencias
T. evansi
PQSs/Kb por Clase
densidad (PQSs/Kb)
a*
a*
a*
SSRi
PTUs
SSRt
SSRi(c)
PTUs(c)
SSRt(c)
Clases de secuencias
T. evansi
%GC por Clase
%GC
a
b
a
SSRi
PTUs
SSRt
secuencias

(II.3)

Se representa la ocurrencia relativa* de PQSs en las transciciones SSR inicio-PTU y PTU-SSR terminación.

(*) En cada posición de las secuencias consideradas, se obtiene el número de PQSs y se divide por el número total de secuencias.

Grafico II.3a: Distribución relativa de PQSs (PQSs/secuencia) entre clases de secuencias. **Linea roja** PQSs en la hebra codificante. **Linea azul**, PQSs en la hebra molde. **Linea punteada,** PQSs en controles negativos (**Importante**: Las escalas varían entre las figuras).
Columna izquierda: transición SSRinicio-PTU. Se representan 500nt del SSRi y 1000nt de la PTU.
Columna derecha: transición ARN policistrónico-SSR de terminación. Se representan 1000nt de la PTU y 500nt del SSRt.



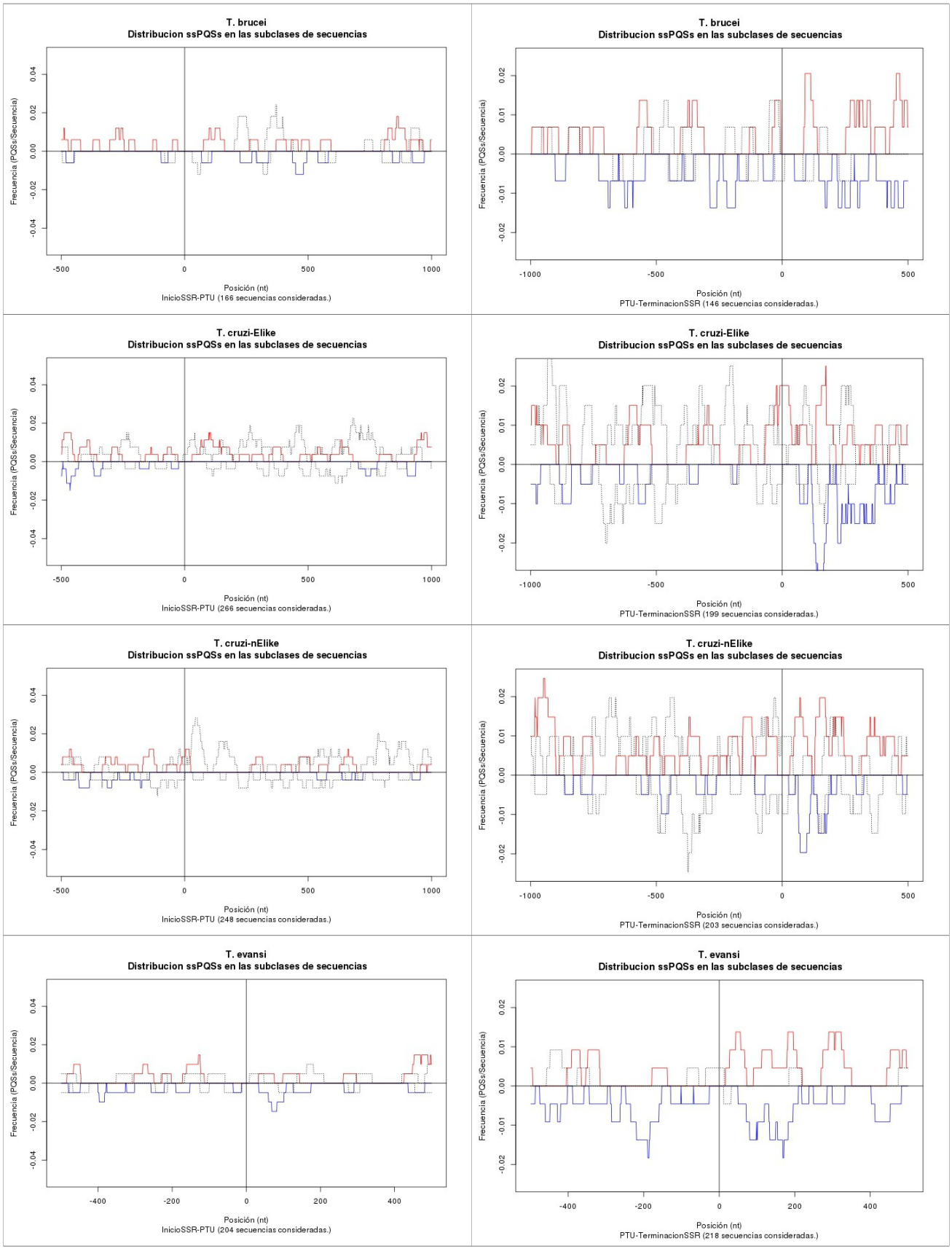
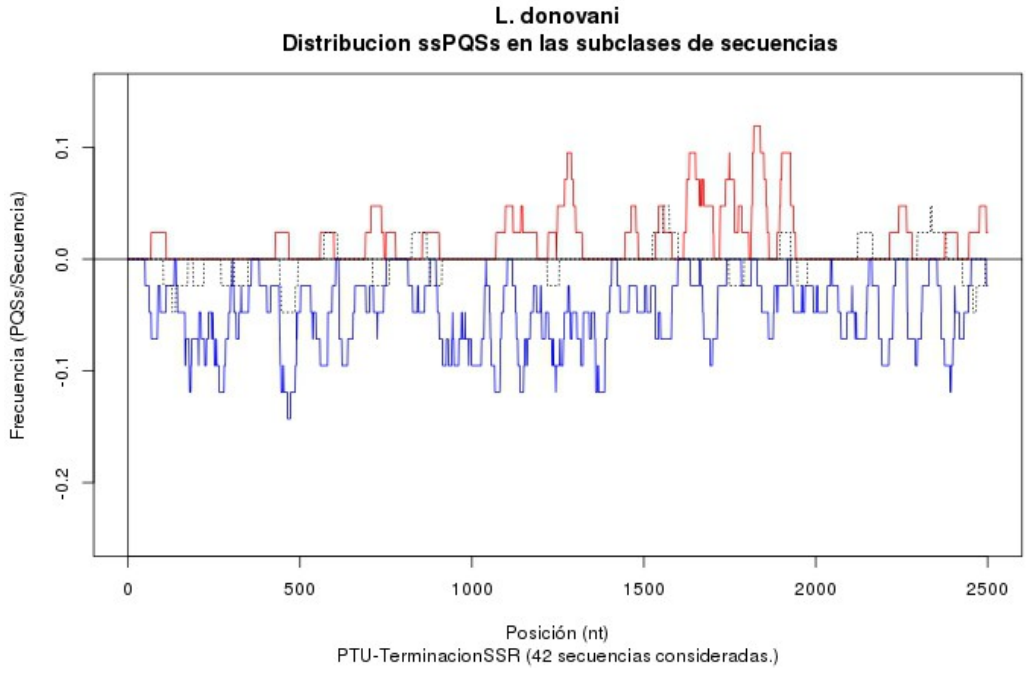
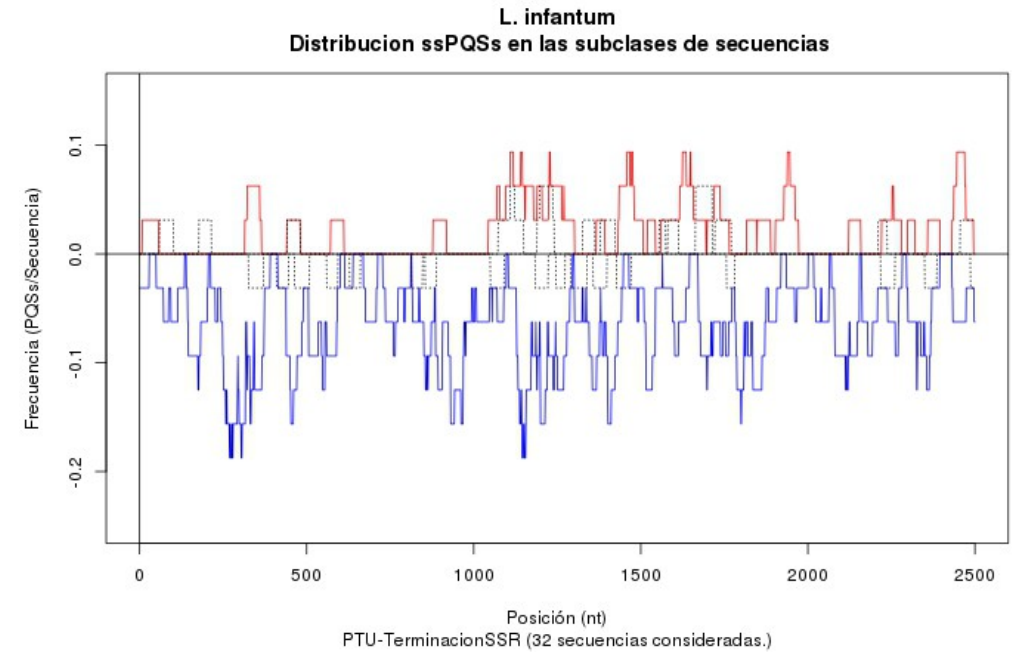
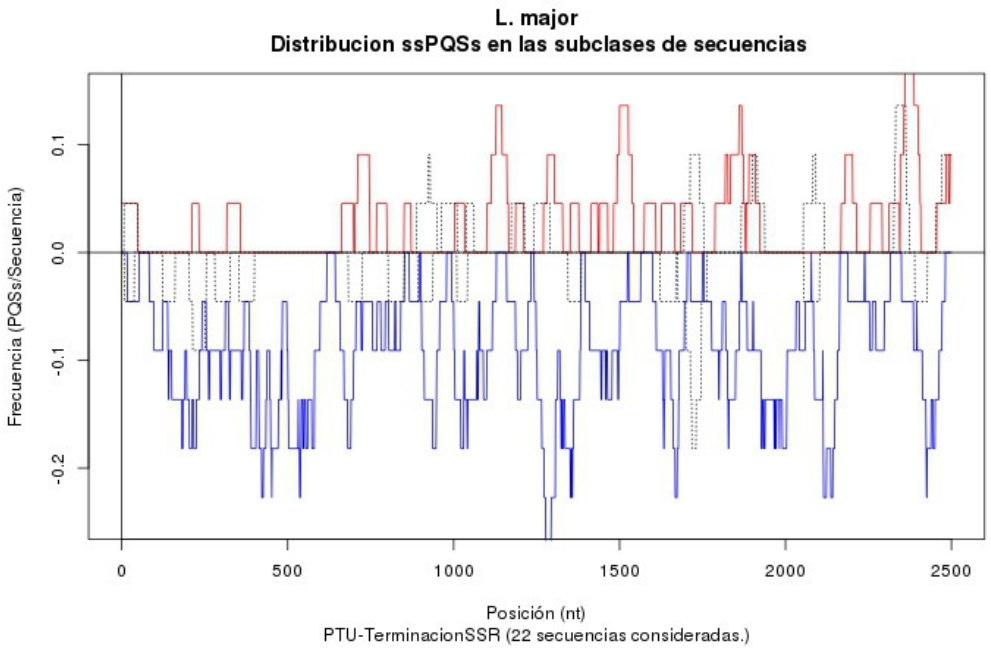

T. brucei
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
InicioSSR-PTU (166 secuencias consideradas.)
T. brucei
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
PTU-TerminacionSSR (146 secuencias consideradas.)
T. cruzi-Elike
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
InicioSSR-PTU (266 secuencias consideradas.)
T. cruzi-Elike
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
PTU-TerminacionSSR (199 secuencias consideradas.)
T. cruzi-nElike
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
InicioSSR-PTU (248 secuencias consideradas.)
T. cruzi-nElike
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
PTU-TerminacionSSR (203 secuencias consideradas.)
T. evansi
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
InicioSSR-PTU (204 secuencias consideradas.)
T. evansi
Distribucion ssPQSs en las subclases de secuencias
Frecuencia (PQSs/Secuencia)
Posición (nt)
PTU-TerminacionSSR (218 secuencias consideradas.)

Figura II.3b: Distribución de PQSs en las primeras 2500 bases del SSR de terminación en *Leishmania*. **Linea roja** PQSs en la hebra codificante. **Linea azul**, PQSs en la hebra molde. **Linea punteada,** PQSs en controles negativos

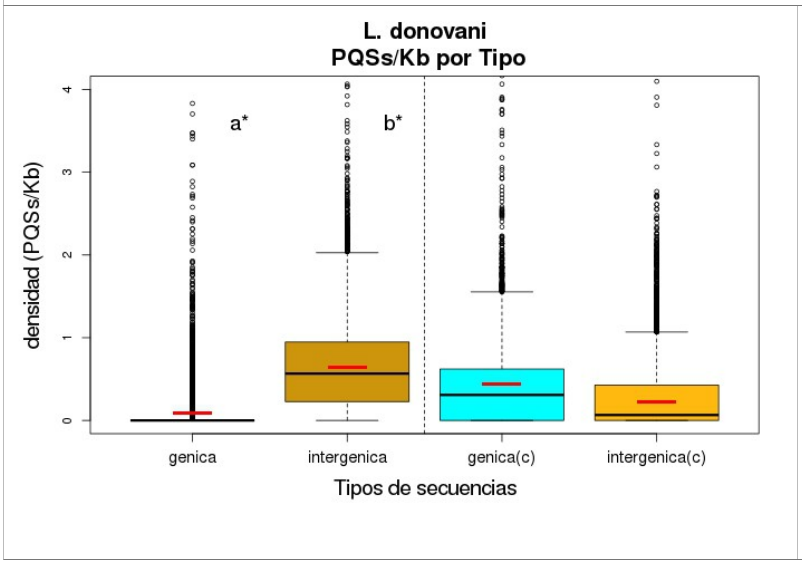


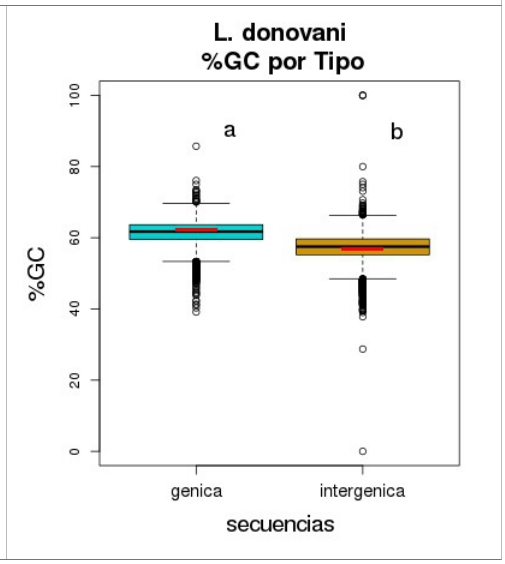


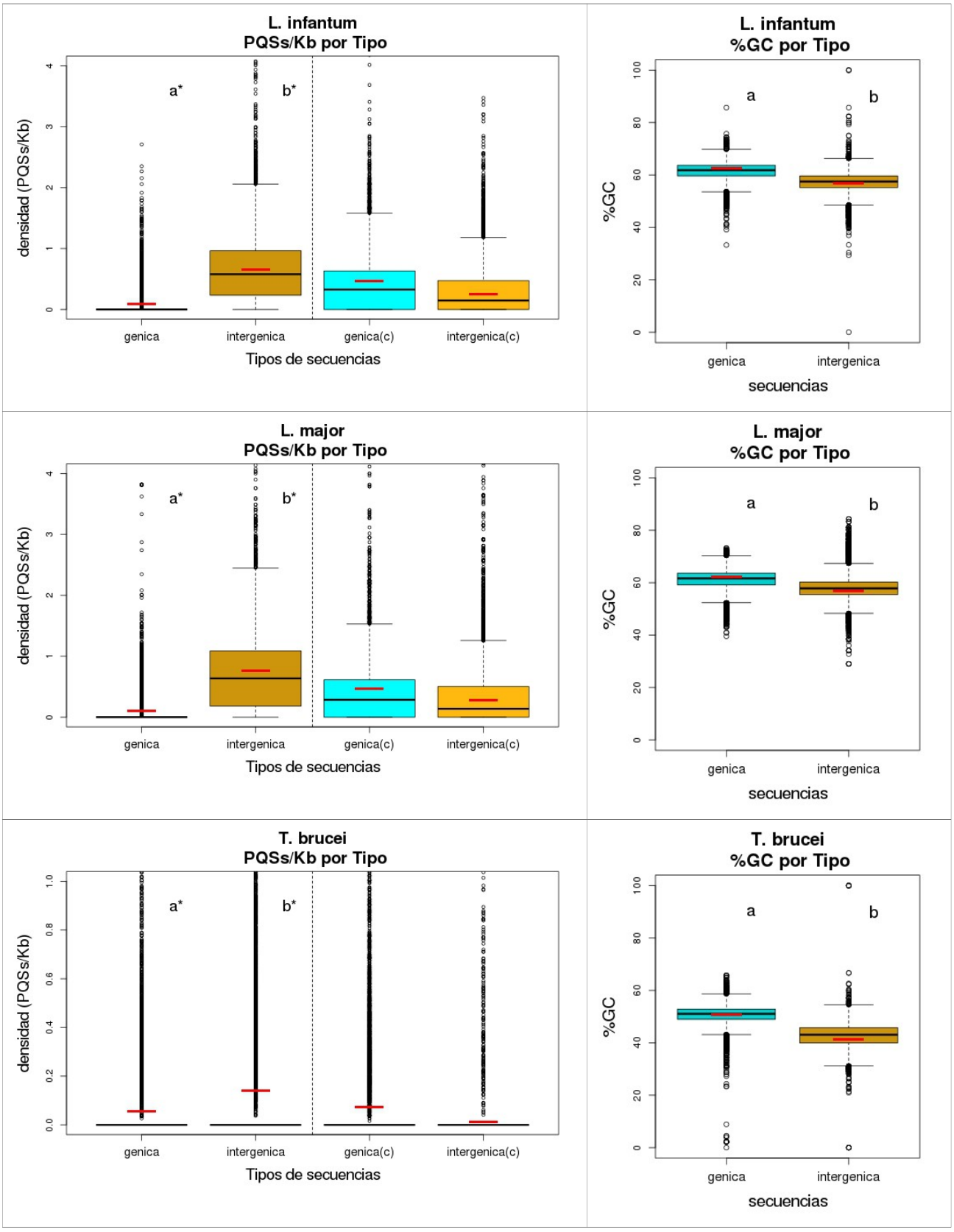


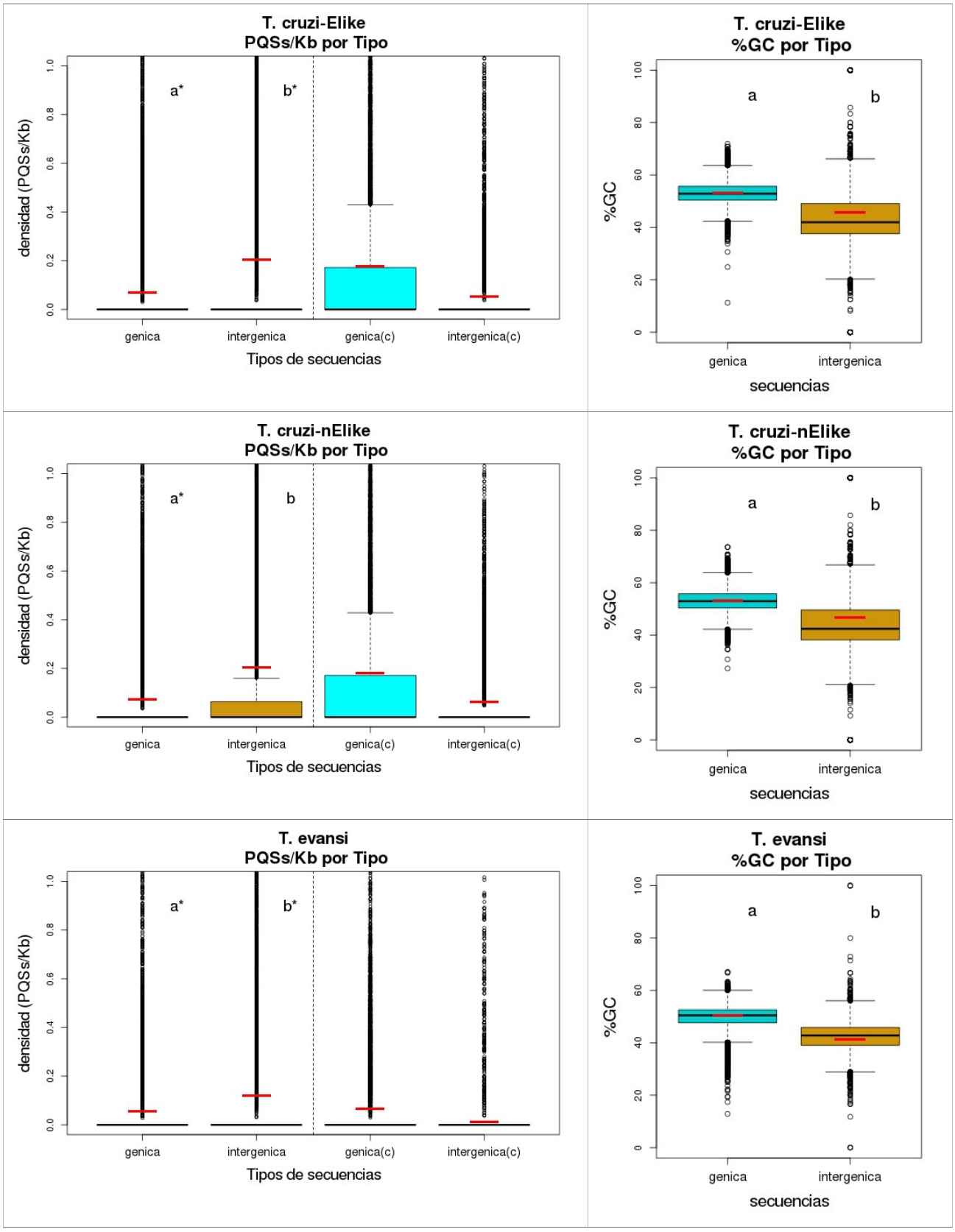
(III.1)

Figura II.4: Densidad de PQSs y contenido GC para cada tipo de secuencia (génica e intergénica). Letras diferentes indican diferencias estadísticamente significativas entre tipos (*t-test*, p<0.05). *Asteriscos* indican diferencias significativas *(t-test, p<0.05)* de cada tipo con su control. *Linea roja*: media ponderada de cada tipo.
Columna izquierda: PQSs/Kb para los genomas y sus controles respectivos (c).
Columna derecha: %GC de los distintos tipos de secuencias.



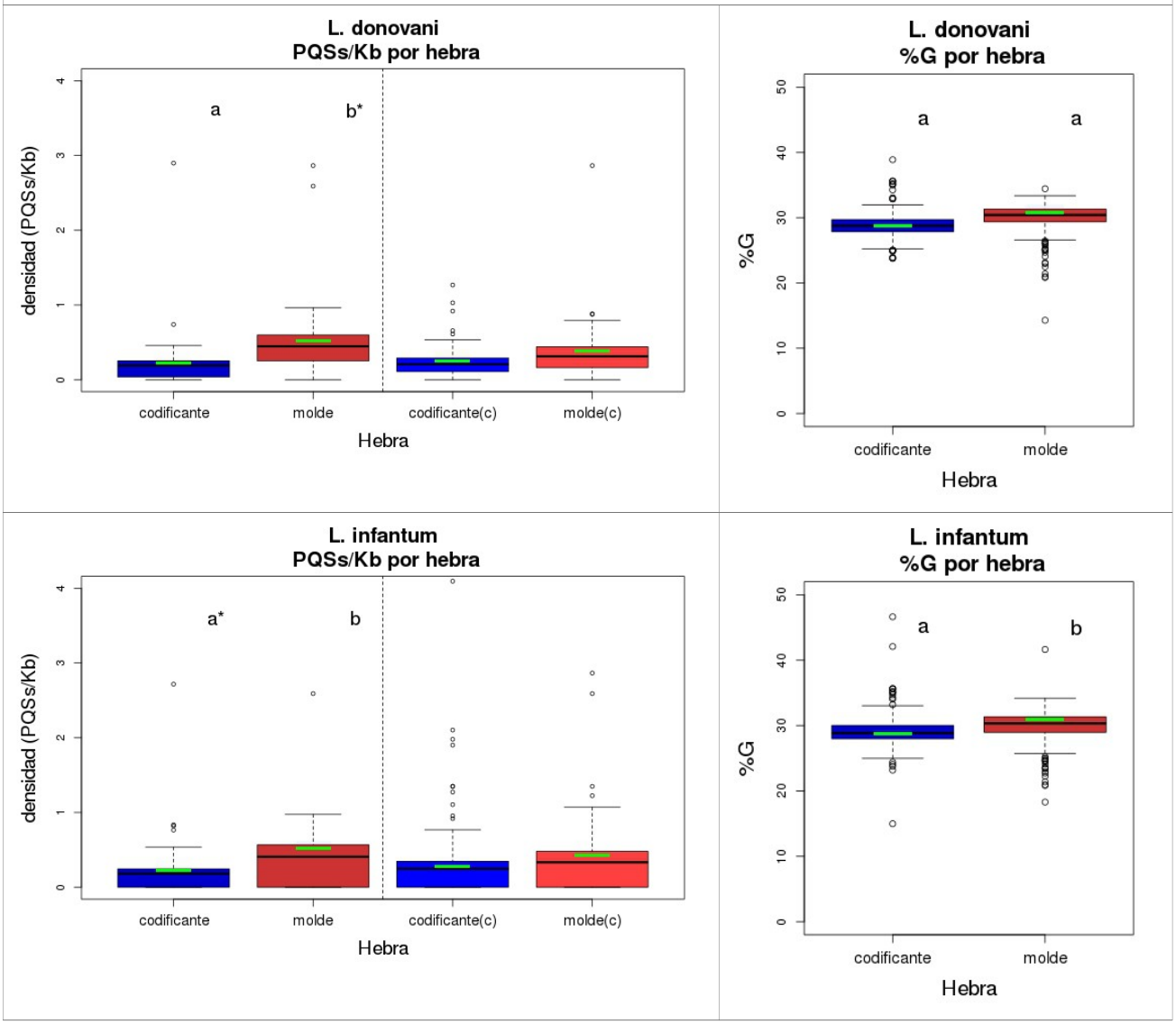


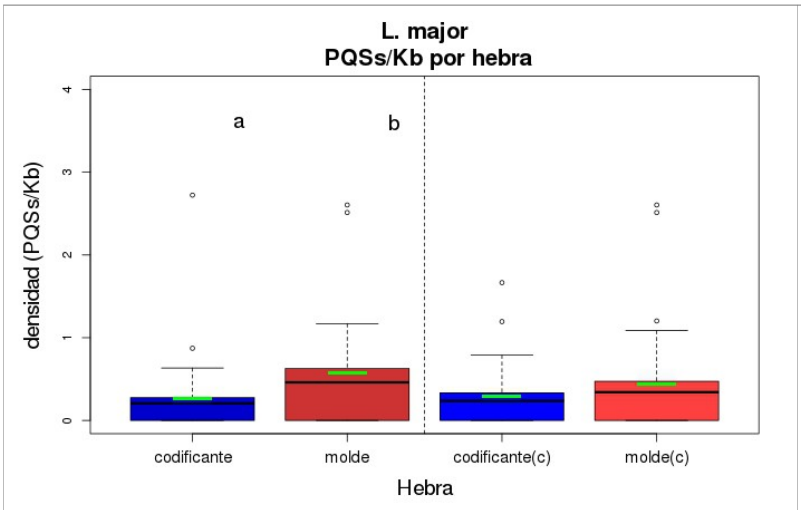

L. infantum
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b*
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
L. infantum
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias
L. major
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b*
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
L. major
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias
T. brucei
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b*
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
T. brucei
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias

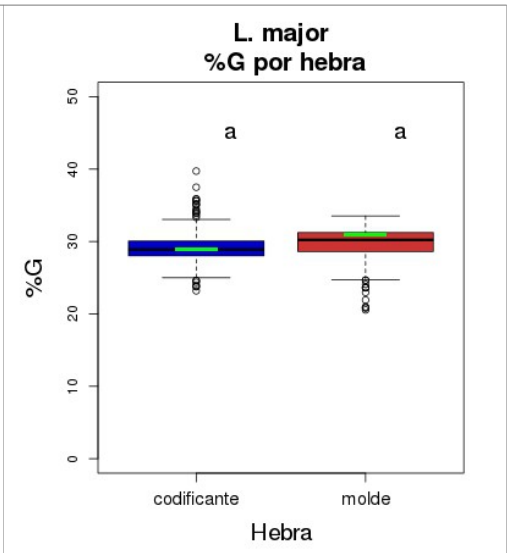

T. cruzi-Elike
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b*
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
T. cruzi-Elike
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias
T. cruzi-nElike
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
T. cruzi-nElike
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias
T. evansi
PQSs/Kb por Tipo
densidad (PQSs/Kb)
a*
b*
genica
intergenica
genica(c)
intergenica(c)
Tipos de secuencias
T. evansi
%GC por Tipo
%GC
a
b
genica
intergenica
secuencias

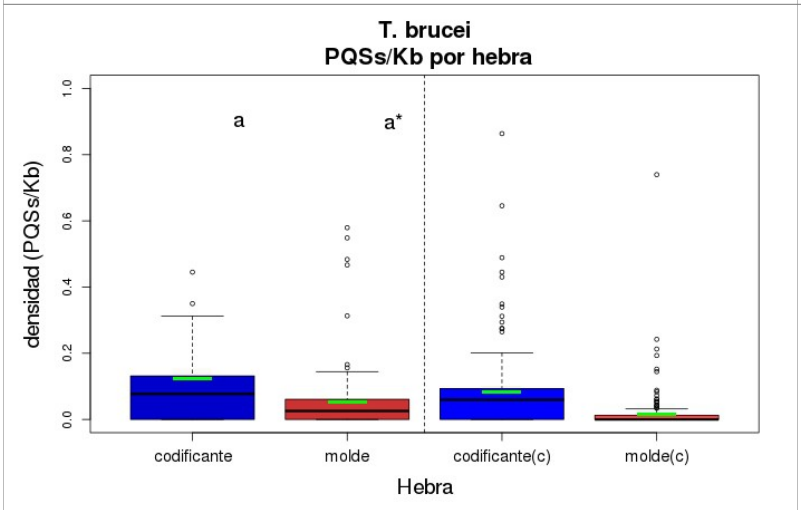
(III.2)

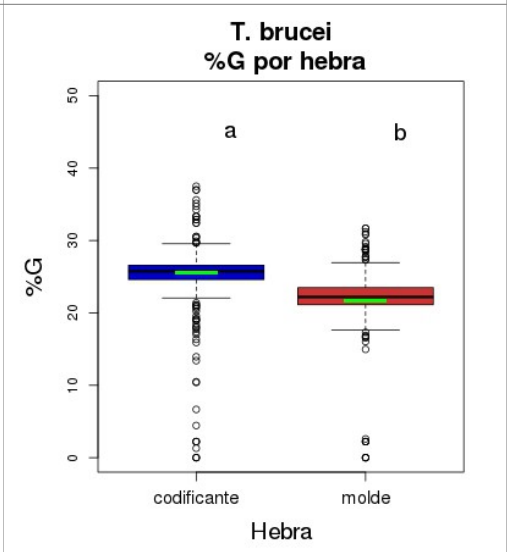
Figura II.5: Densidad de PQSs y contenido G para las distintas hebras de los ARN policistrónicos. Letras diferentes indican diferencias estadísticamente significativas entre tipos (*paired t-test*, p<0.05). *Asteriscos* indican diferencias significativas de cada hebra con su control *(t-test* p<0.05*)*. *Linea verde*: media ponderada para cada hebra.
Columna izquierda: PQSs/Kb para los genomas y sus controles respectivos (c).
Columna derecha: %G en cada hebra del ARN policistrónico.

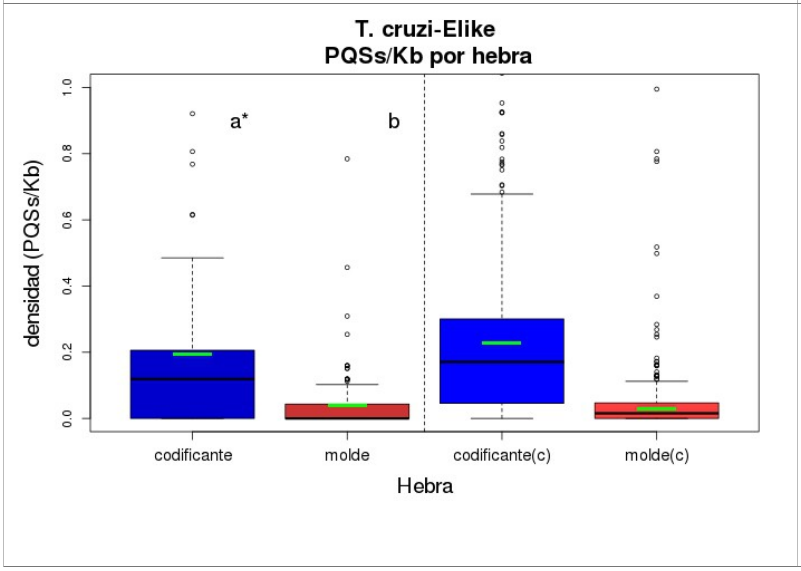


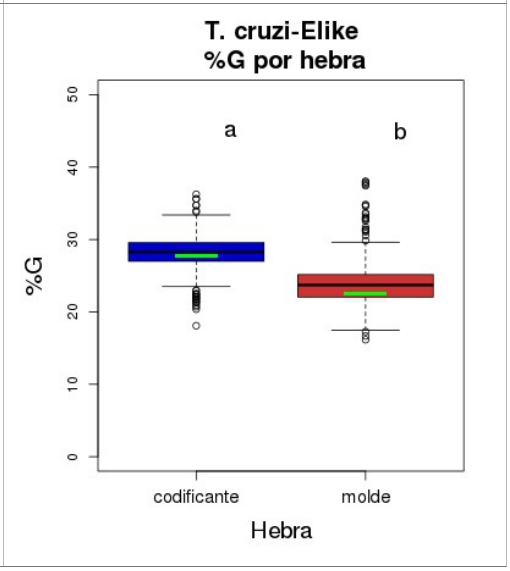

L. major
PQSs/Kb por hebra
densidad (PQSs/Kb)
a
b
codificante
molde
codificante(c)
molde(c)
Hebra


L. major
%G por hebra
%G
a
a
codificante
molde
Hebra


T. brucei
PQSs/Kb por hebra
densidad (PQSs/Kb)
a
a*
codificante
molde
codificante(c)
molde(c)
Hebra


T. brucei
%G por hebra
%G
a
b
codificante
molde
Hebra


T. cruzi-Elike
PQSs/Kb por hebra
densidad (PQSs/Kb)
a*
b
codificante
molde
codificante(c)
molde(c)
Hebra


T. cruzi-Elike
%G por hebra
%G
a
b
codificante
molde
Hebra

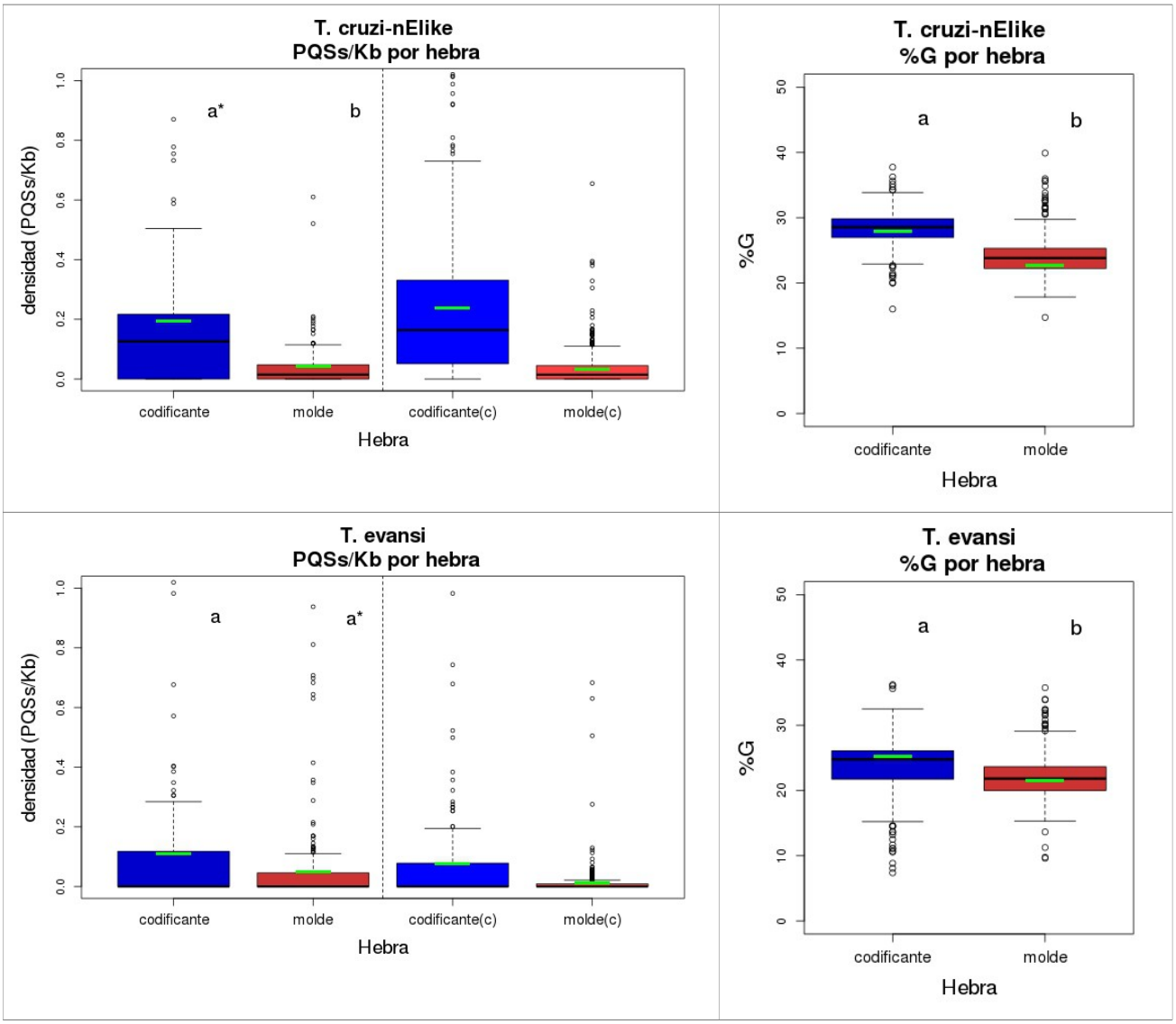
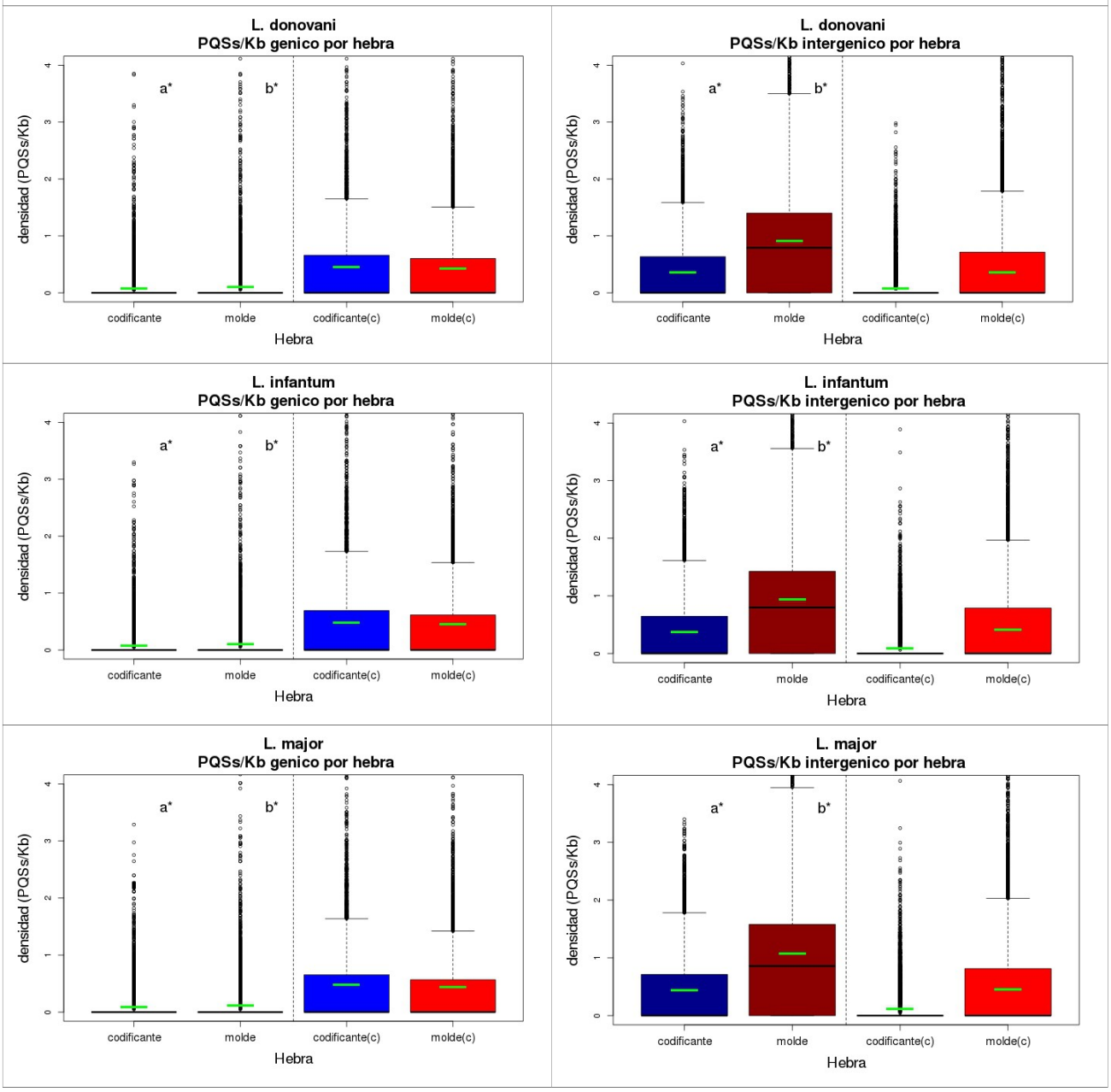

T. cruzi-nElike
PQSs/Kb por hebra
densidad (PQSs/Kb)
a*
b
codificante
molde
codificante(c)
molde(c)
Hebra
T. cruzi-nElike
%G por hebra
%G
a
b
codificante
molde
Hebra
T. evansi
PQSs/Kb por hebra
densidad (PQSs/Kb)
a
a*
codificante
molde
codificante(c)
molde(c)
Hebra
T. evansi
%G por hebra
%G
a
b
codificante
molde
Hebra

Figura II.6a: Densidad de PQSs para las distintas hebras de cada tipo de secuencia (génica e intergénica). Letras diferentes indican diferencias estadísticamente significativas entre hebras *(paired t-test,* p<0.05*). Asteriscos* indican diferencias significativas de cada hebra con su control (*t-test,* p<0.05*). Linea verde*: media ponderada para cada hebra. **Columna izquierda:** PQSs/Kb por hebra para las **secuencias génicas** y sus controles respectivos (c). **Columna derecha:** PQSs/Kb por hebra para las **secuencias intergénicas** y sus controles respectivos (c).



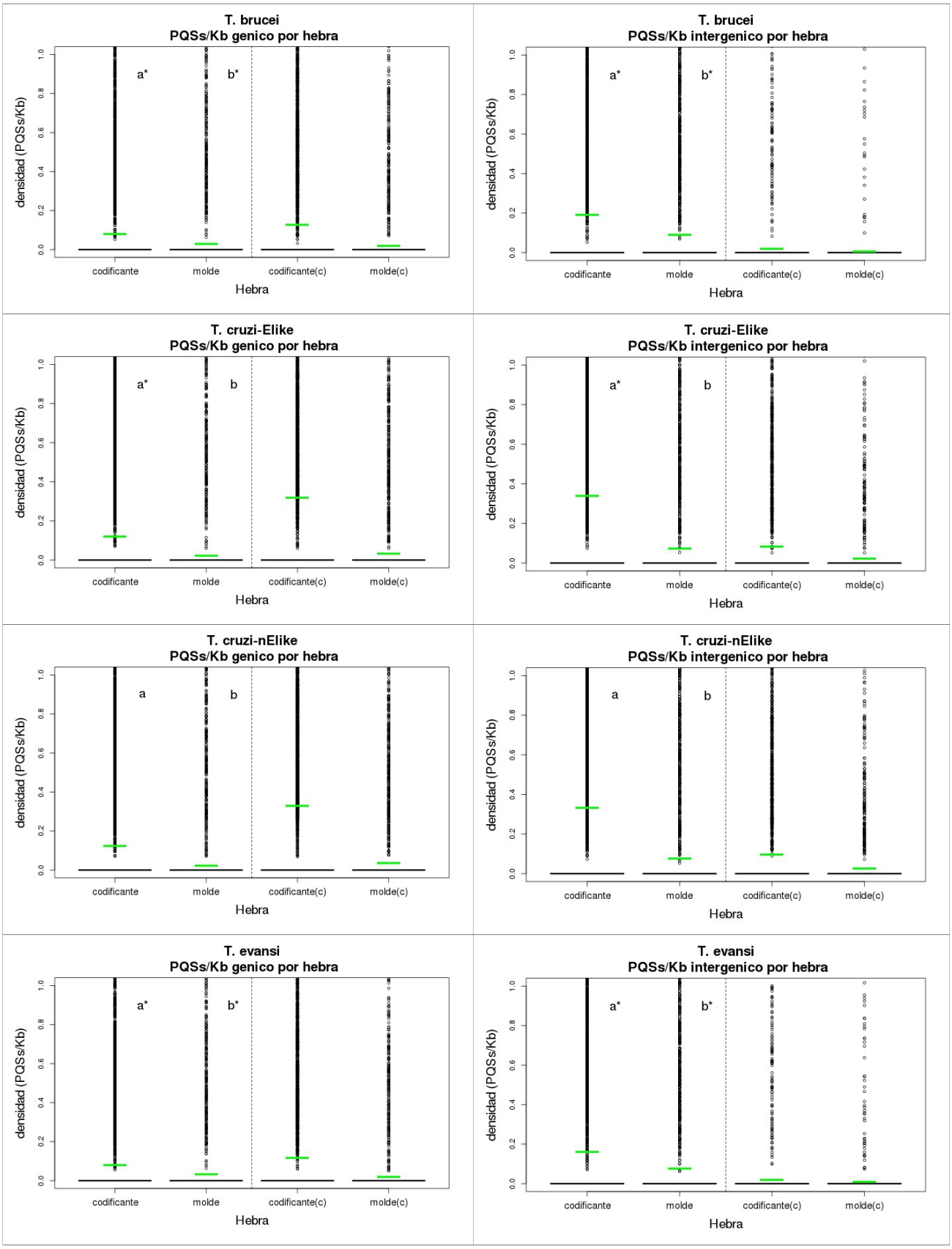
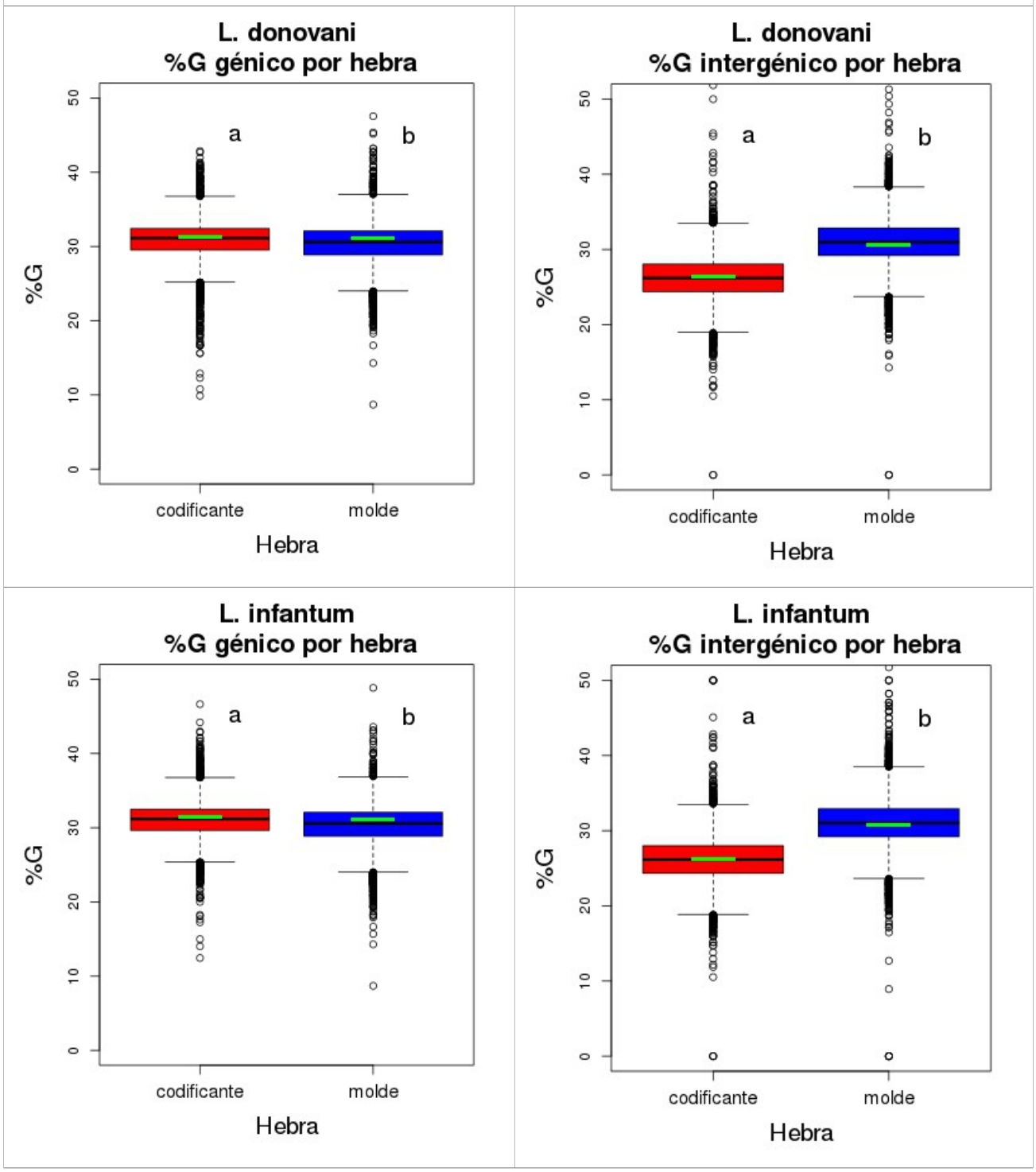
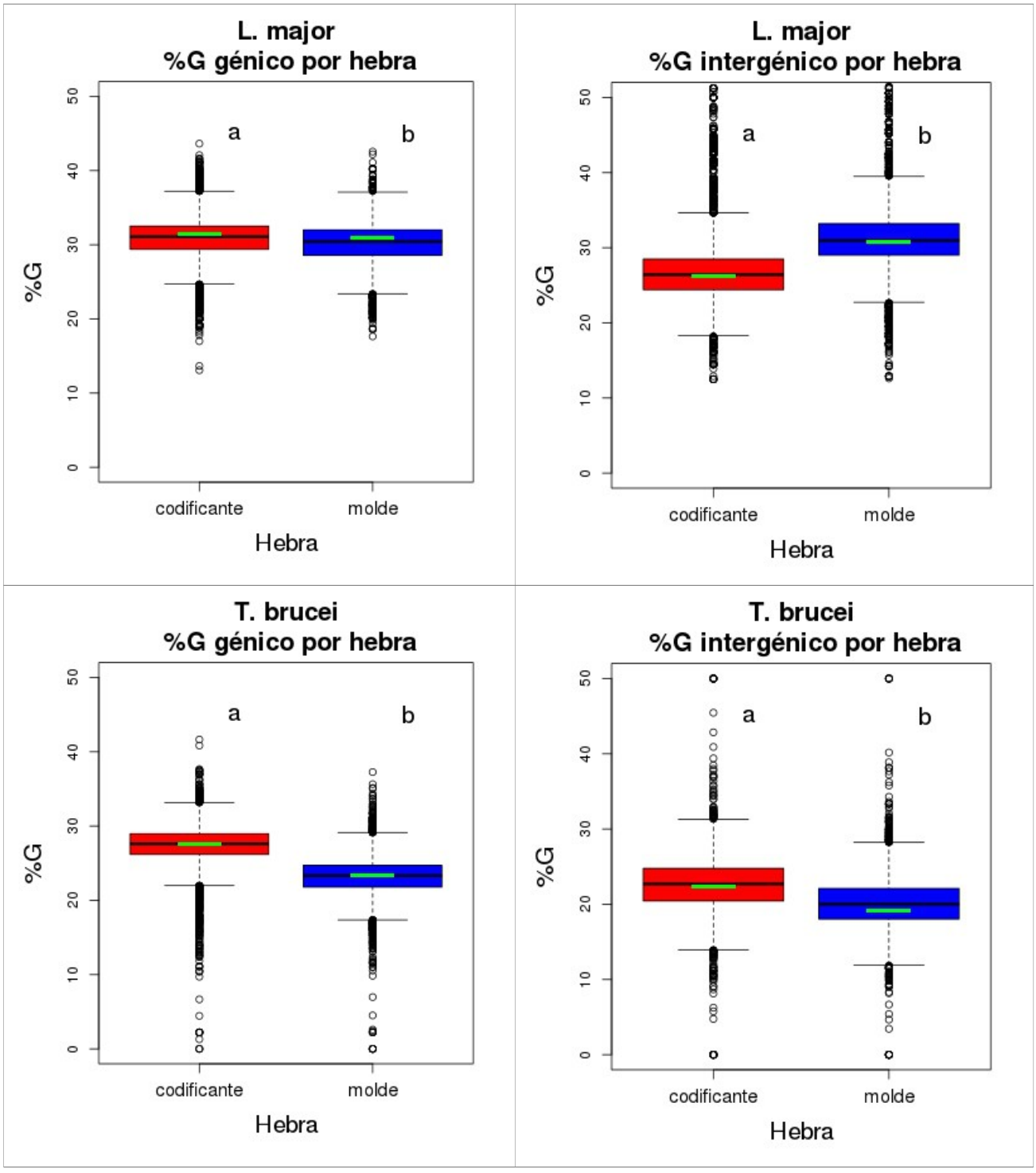
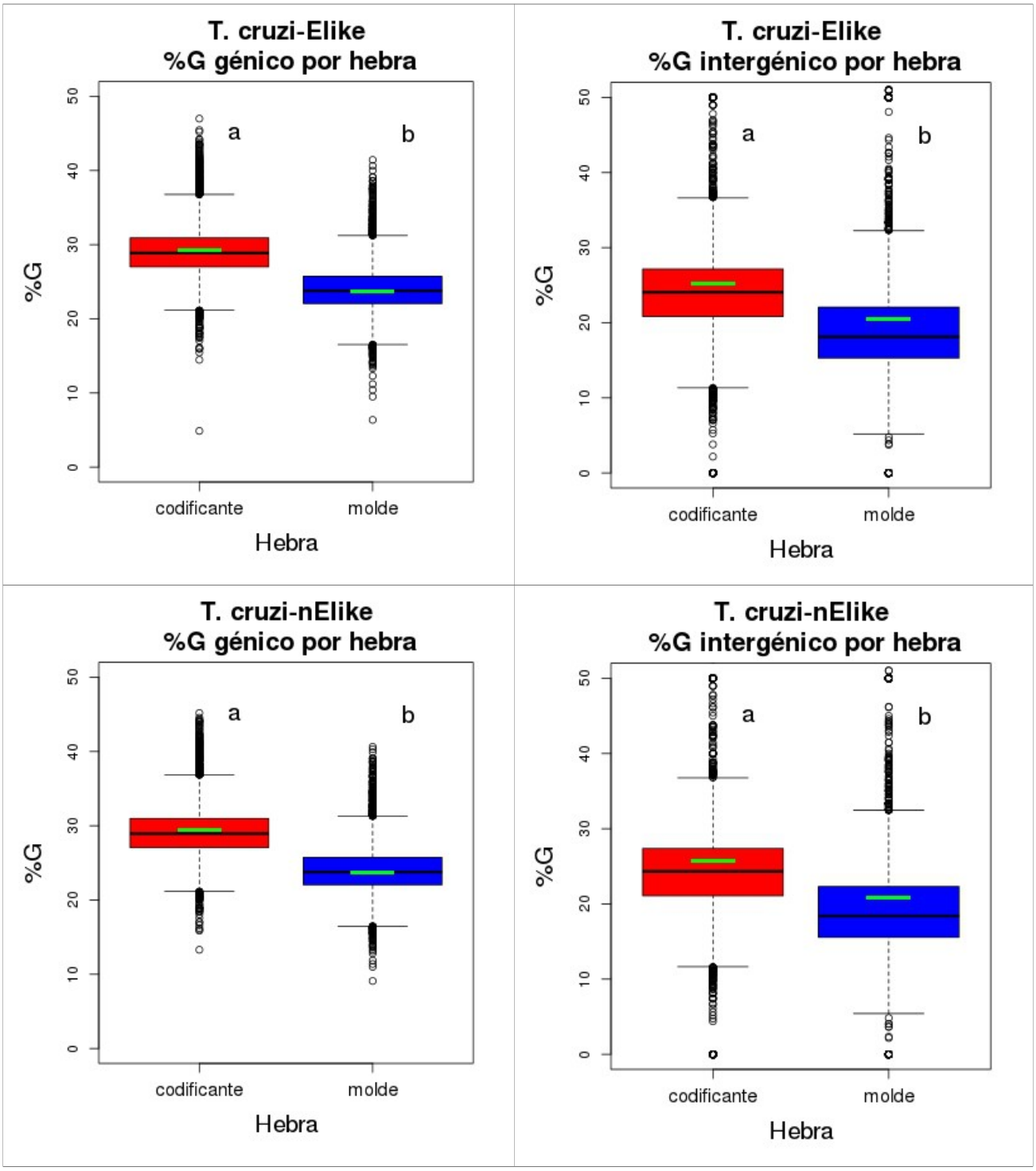

T. brucei
PQSs/Kb genico por hebra
T. brucei
PQSs/Kb intergenico por hebra
T. cruzi-Elike
PQSs/Kb genico por hebra
T. cruzi-Elike
PQSs/Kb intergenico por hebra
T. cruzi-nElike
PQSs/Kb genico por hebra
T. cruzi-nElike
PQSs/Kb intergenico por hebra
T. evansi
PQSs/Kb genico por hebra
T. evansi
PQSs/Kb intergenico por hebra
densidad (PQSs/Kb)
codificante
molde
codificante(c)
molde(c)
Hebra

Figura II.6b: Contenido G para las distintas hebras de cada tipo de secuencia (génica e intergénica). Letras diferentes indican diferencias estadísticamente significativas entre hebras *(paired t-test p≤0.05). Linea verde*: media ponderada para cada hebra. **Columna izquierda:** %G por hebra para las **secuencias génicas** y sus controles respectivos (c). **Columna derecha:** %G por hebra para las **secuencias intergénicas** y sus controles respectivos (c).




L. major
%G génico por hebra
%G
a
b
codificante
molde
Hebra
L. major
%G intergénico por hebra
%G
a
b
codificante
molde
Hebra
T. brucei
%G génico por hebra
%G
a
b
codificante
molde
Hebra
T. brucei
%G intergénico por hebra
%G
a
b
codificante
molde
Hebra


T. cruzi-Elike
%G génico por hebra
T. cruzi-Elike
%G intergénico por hebra
T. cruzi-nElike
%G génico por hebra
T. cruzi-nElike
%G intergénico por hebra
%G
Hebra
codificante
molde
a
b
0
10
20
30
40
50

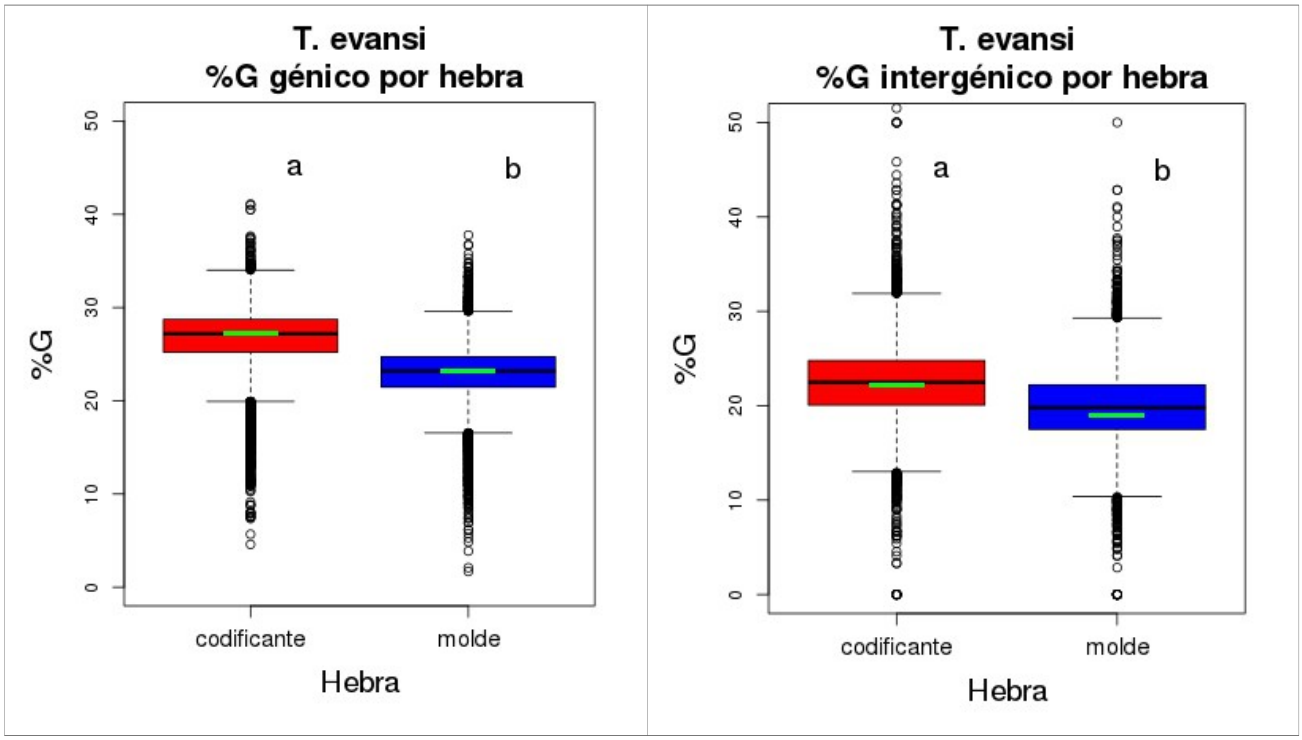
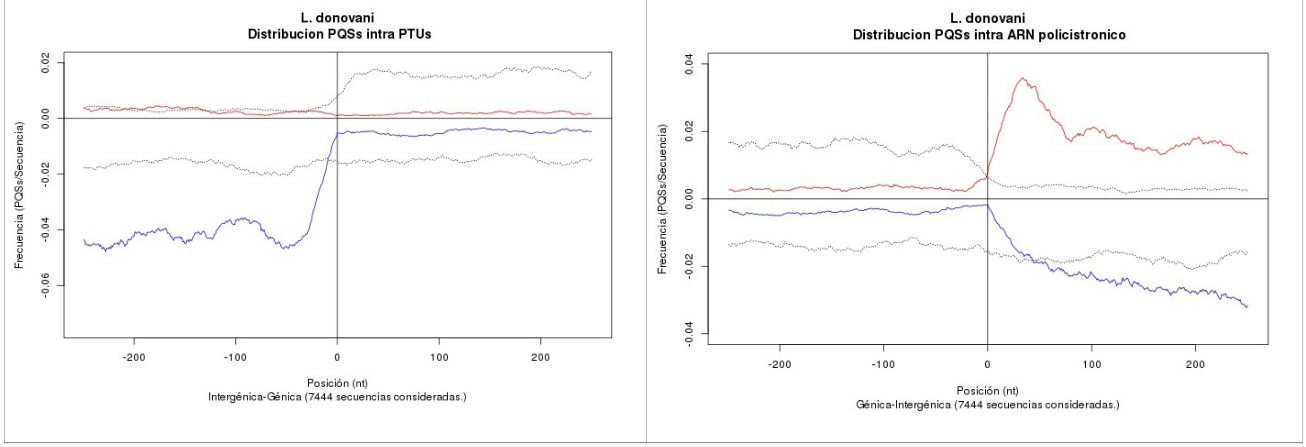
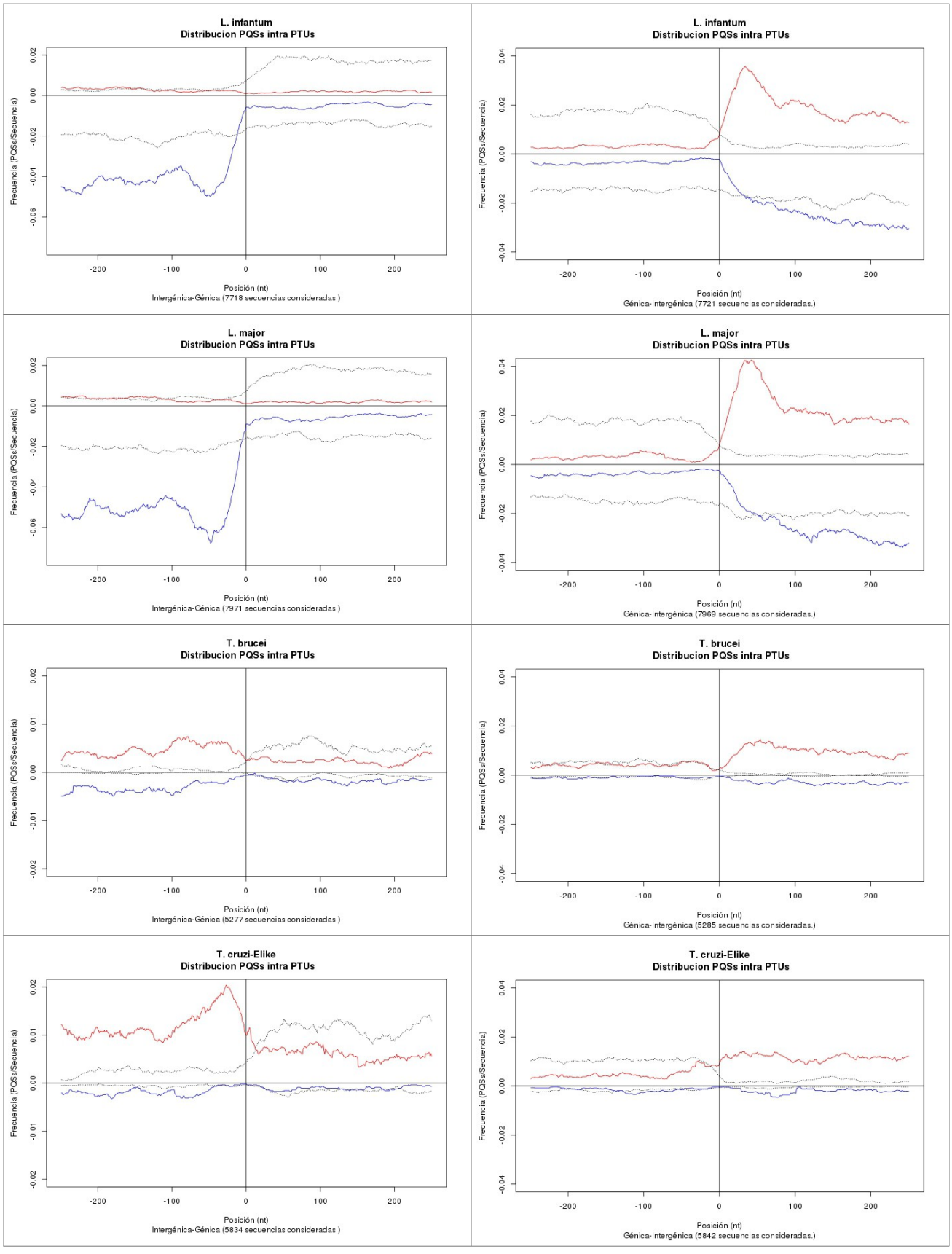
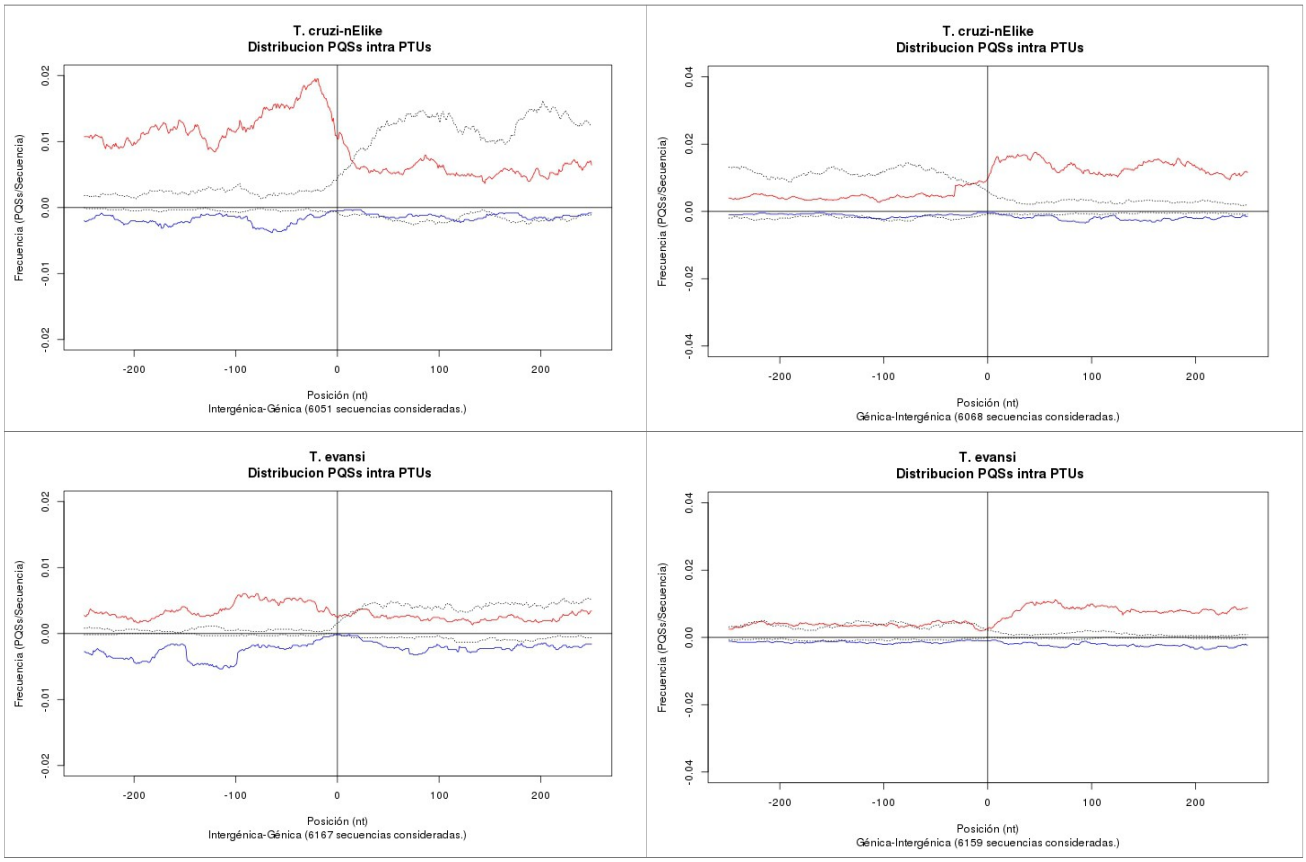


Figura II.7: Distribución relativa de PQSs (PQSs/secuencia) dentro de los PTUs. **Linea roja** PQSs en la hebra codificante. **Linea azul**, PQSs en la hebra molde. **Linea punteada,** PQSs en controles negativos. En todos los casos se representan 250nt de cada tipo de secuencia (**Importante**: Las escalas varían entre figuras). **Columna izquierda:** transición intergénica-génica. **Columna derecha:** transición génica-intergénica.




L. infantum
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (7718 secuencias consideradas.)
L. infantum
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (7721 secuencias consideradas.)
L. major
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (7971 secuencias consideradas.)
L. major
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (7969 secuencias consideradas.)
T. brucei
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (5277 secuencias consideradas.)
T. brucei
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (5285 secuencias consideradas.)
T. cruzi-Elike
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (5834 secuencias consideradas.)
T. cruzi-Elike
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (5842 secuencias consideradas.)


T. cruzi-nElike
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (6051 secuencias consideradas.)
T. cruzi-nElike
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (6068 secuencias consideradas.)
T. evansi
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Intergénica-Génica (6167 secuencias consideradas.)
T. evansi
Distribucion PQSs intra PTUs
Frecuencia (PQSs/Secuencia)
Posición (nt)
Génica-Intergénica (6159 secuencias consideradas.)

Especialización en Bioinformática
Trabajo Final

Anexo IIIa: Código

Script S1: Establecimiento de clases y tipos de secuencias y determinación de composición de bases en cada secuencia.

```
#Script 1: Posición genes, composición de bases
library("TrypR")
##########################################################
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#1)Leo las anotaciones a distintos genomas
setwd("./anotaciones")
nombres<-list.files(pattern="*.gff")
secuencias<-list()
for (i in 1:length(nombres)){
  secuencias[[i]]<-GFF.read(con=nombres[i])#Se necesita conexion a internet
  print(i)
}
rm(i,first_time,nombres)
#####
#2)Abro los archivos con las secuencias genómicas (originales)
setwd("../genomas")
nombres<-list.files(pattern="*.fasta")
genomas<-list()
for (i in 1:length(nombres)){
  genomas[[i]]<-readDNAStringSet(nombres[i], "fasta")
  print(i)
}
#####
#3) Proceso los archivos para obtener las clases y tipos de secuencias
clases<-list()
for (i in 1:length(secuencias)){
  clases[[i]]<-poliRNA(secuencias[[i]])
  print(i)
}
#Obtengo los tipos de secuencias
tipos<-list()
for (i in 1:length(secuencias)){
  tipos[[i]]<-fill.gaps(secuencias[[i]])
  print(i)
}
#####
#4) Asigno valores de GC, AT y N a cada clase y tipo de secuencia
clases.procesadas<-list()
tipos.procesados<-list()
for (i in 1:length(genomas)){
  clases.procesadas[[i]]<-GC.PQS.asign2(secuencias1=clases[[i]],genoma1=genomas[[i]],cuadruplex1=NULL)
  print(paste0("Genoma: ",i," listo (clases)"))
  tipos.procesados[[i]]<-GC.PQS.asign2(secuencias1=tipos[[i]],genoma1=genomas[[i]],cuadruplex1=NULL)
  print(paste0("Genoma: ",i," listo (tipos)"))
}
#####
#5) Guardo imagen en archivos intermedios
setwd("../Archivos intermedios")
#save(tipos.procesados,clases.procesadas,file="Tipos y Clases.RData")
```

Script S2: generación de genomas control

```
#Script 2: generación de genomas control
library("TrypR")
```

```
##########################################################
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#1) Cargo los archivos guardados (salidas script1)
setwd("./Archivos intermedios")
load("Tipos y Clases.RData")
#Solo trabajo con los Tipos
rm(clases.procesadas)
#####
#2) Establezco las secuencias aleatoreas para cada tipo de secuencia, respetando las bases
bases.control<-list()
for (j in 1:length(tipos.procesados)){
  x<-tipos.procesados[[j]]
  G.per<-x$GinStrand
  C.per<-(x$CG-x$GinStrand)
  A.per<-x$AinStrand
  N.per<-x$N
  long<-width(x)
  x$seq<-"N"
  for(i in 1:length(x)){
    mcols(x)[i,]$seq<-seqmix2(A=A.per[i],C=C.per[i],G=G.per[i],N=N.per[i],long=long[i])
    print(paste0(j,":",i))
  }
  #Las bases de las secuencias negativas figuran en forma complementaria en la tabla de referencia,
  #debo complementarlas en la salida para obener las bases correctas.
  mcols(x)[strand(x)=="-",]$seq<-as.character(reverseComplement(DNAStringSet(mcols(x)[strand(x)=="-",]$seq)))
  bases.control[[j]]<-x
}
#####
#3) Establezco los genomas control
bases.control.genomas<-list()
for(k in 1:length(bases.control)){
  y<-bases.control[[k]]
  tabla<-data.frame("seqnames"=seqnames(y),"end"=end(y),"seq"=mcols(y)$seq, stringsAsFactors = F)
  seq<-paste0(tabla$seq, collapse = "")#Pego todas las secuencias en una sola
  #Recorto teniendo en cuenta las longitudes de los cromosomas
  tamaños.cromosomas<-aggregate(tabla$end, by=list(tabla$seqnames), max)
  z<-matrix(data=NA, nrow=nrow(tamaños.cromosomas)+1,ncol = 2)
  z[,1]<-c(1,cumsum(tamaños.cromosomas$x)+1)
  z[,2]<-c(cumsum(tamaños.cromosomas$x),0)
  z<-z[z[,2]!=0,]
  w<-substring(seq,z[,1],z[,2])#Recorto
  w<-DNAStringSet(w)
  #Asigno nombres
  for (i in 1:length(w)){
    names(w)[i]<-paste0(tamaños.cromosomas[i,1]," | organism=",acronimos[k]," (control bases) | version=",date()," |
length=",tamaños.cromosomas[i,2]," | SO=chromosome")
  }
  bases.control.genomas[[k]]<-w
}
#####
#5) Guardo las imagenes en archivos intermedios
setwd("../Archivos intermedios")
# save(bases.control,file="genomas.control.tablas.RData")
# save(bases.control.genomas, file="genomas.control.fasta.RData")
```

Script S3: determinación de PQSs

```
#Script 3: determinación de PQSs
library("TrypR")
##########################################################
#1) Cargo los genomas (originales)
setwd("./genomas")
listagenomas<-list.files(pattern="*.fasta")
```

```
#2)Analizo los genomas de a uno por vez y cargo todo en una lista
g4.original<-list()
for (j in 1:length(listagenomas)){
  #2.1) Convierto los genomas a un objeto de la clase DNAStringSet
  genoma <- readDNAStringSet(listagenomas[j])
  n_cromosomas<-length(genoma)
  #2.2) Analizo los cromosomas de a uno por vez y fusiono en un archivo
  fusion<-c()
  for (i in 1:n_cromosomas){
    chr<-genoma[[i]]
    nombre<-strsplit(names(genoma), " | ")[[i]][1]
    print(nombre)
    #2.2.1) Aplico la función "pqsfinder" a cada cromosoma
    pqs<- pqsfinder(chr, strand = "*")
    GRpqs<-as(pqs, "GRanges")
    seqlevels(GRpqs)<-nombre
    #2.2.2) Cargo los resultados de cada cromosoma en un archivo
    fusion<-c(GRpqs,GRpqs)
    #2.2.3) Filtro PQSs con Score superior a 86
    fusion<-fusion[mcols(fusion)$score>86]
  }
  #2.3) Cargo los resultados de cada genoma en una lista
  g4.original[[j]]<-fusion
}
#####
#3) Cargo los genomas (control)
setwd("../Archivos intermedios")
load("genomas.control.fasta.RData")#2)Analizo los genomas de a uno por vez y cargo todo en una lista
#4)Analizo los genomas de a uno por vez y cargo todo en una lista
g4.control<-list()
for (j in 1:length(bases.control.genomas)){
  genoma <- bases.control.genomas[[j]]
  n_cromosomas<-length(genoma)
  fusion<-c()
  for (i in 1:n_cromosomas){
    chr<-genoma[[i]]
    nombre<-strsplit(names(genoma), " | ")[[i]][1]
    print(nombre)
    pqs<- pqsfinder(chr, strand = "*")
    GRpqs<-as(pqs, "GRanges")
    seqlevels(GRpqs)<-nombre
    fusion<-c(GRpqs,GRpqs)
    fusion<-fusion[mcols(fusion)$score>86]
  }
  g4.control[[j]]<-fusion
}
#####
#5)guardo la lista con datos de PQSs de todos los genomas
#save(g4.original,g4.control,file="g4.RData")
```

Script S4: Fusión tipos-clases de secuencias y PQSs

```
#Script 4: Fusion tipos-clases de secuencias y PQSs
library("TrypR")
#########################################################
#1)Abro los archivos con las secuencias clasificadas (común para genomas y controles)
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#2) Cargo las tablas con las secuencias clasificadas en tipos y clases (salida S1)
setwd("./Archivos intermedios")
load("Tipos y Clases.RData")#2)Analizo los genomas de a uno por vez y cargo todo en una lista
#3) Cargo los datos de posición de los PQSs (salida S3)
load("g4.RData")
#####
```

```
#4)Asigno los PQSs a cada tipo y clase de secuencia
clases<-clases.procesadas
tipos<-tipos.procesados
clases.procesadas.original<-list()
tipos.procesados.original<-list()
clases.procesadas.control<-list()
tipos.procesados.control<-list()
for (i in 1:length(acronimos)){
  clases.procesadas.original[[i]]<-GC.PQS.asign2(secuencias1=clases[[i]],genoma1=NULL,cuadruplex1=g4.original[[i]])
  clases.procesadas.control[[i]]<-GC.PQS.asign2(secuencias1=clases[[i]],genoma1=NULL,cuadruplex1=g4.control[[i]])
  print(paste0("Genoma: ",i," listo (clases)"))
  tipos.procesados.original[[i]]<-GC.PQS.asign2(secuencias1=tipos[[i]],genoma1=NULL,cuadruplex1=g4.original[[i]])
  tipos.procesados.control[[i]]<-GC.PQS.asign2(secuencias1=tipos[[i]],genoma1=NULL,cuadruplex1=g4.control[[i]])
  print(paste0("Genoma: ",i," listo (tipos)"))
}
#####
#5)Paso a formato tabla (¿Necesario?)
procesadas.clases.original<-list()
procesadas.clases.control<-list()
procesadas.tipos.original<-list()
procesadas.tipos.control<-list()
for (i in 1:length(acronimos)){
  a<-as.data.frame(clases.procesadas.original[[i]], row.names = NULL, optional = FALSE)
  b<-data.frame(names=names(clases.procesadas.original[[i]]),a)
  procesadas.clases.original[[i]]<-b
  a<-as.data.frame(clases.procesadas.control[[i]], row.names = NULL, optional = FALSE)
  b<-data.frame(names=names(clases.procesadas.control[[i]]),a)
  procesadas.clases.control[[i]]<-b
  a<-as.data.frame(tipos.procesados.original[[i]], row.names = NULL, optional = FALSE)
  b<-data.frame(names=names(tipos.procesados.original[[i]]),a)
  procesadas.tipos.original[[i]]<-b
  a<-as.data.frame(tipos.procesados.control[[i]], row.names = NULL, optional = FALSE)
  b<-data.frame(names=names(tipos.procesados.control[[i]]),a)
  procesadas.tipos.control[[i]]<-b
}
#####
#6)Guardo en dos archivos .RData
# save(procesadas.clases.original,procesadas.clases.control,file="clases.procesadas.RData")
# save(procesadas.tipos.original,procesadas.tipos.control,file = "tipos.procesados.RData")
```

Script S5: Análisis general genomas

```
#Script 5: Analisis general genomas
library("TrypR")
#########################################################
#1)Abro los archivos con las secuencias clasificadas (común para genomas y controles)
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#2) Cargo la imagen de los clases procesadas de todos los genomas (originales y control)
setwd("./Archivos intermedios")
load("clases.procesadas.RData")
#####
#I.1) Datos generales del genoma
tabla1<-data.frame(matrix(NA, nrow = length(procesadas.clases.original), ncol =7 ))
rownames(tabla1)<-acronimos
for (i in 1:length(procesadas.clases.original)){
  x<-procesadas.clases.original[[i]]#Selecciono el genoma original
  y<-procesadas.clases.control[[i]]#Selecciono el genoma control
  C<-sum(x$width)#Tamaño genómico
  n<-length(levels(x$seqnames))#Número cromosomas
  N.percent<-round(100*sum(x$N)/sum(x$width),digits = 1)#Contenido N (original)
  GC.percent<-round(100*sum(x$CG)/sum(x$LongEf),digits=1)#Contenido GC (corregido)
  PQSs<-sum(x$g4number)#Contenido PQSs (total según score seleccionado)
  PQSs.Kb.o<-round(1000*sum(x$g4number)/sum(x$LongEf), digits = 3)#densidad PQSs/Kb (corregida)
```

```
  PQSs.Kb.c<-round(1000*sum(y$g4number)/sum(y$LongEf), digits = 3)#densidad PQSs/Kb (corregida)

  tabla1[i,1:7]<-c(C,n,N.percent,GC.percent,PQSs,PQSs.Kb.o,PQSs.Kb.c)
}
colnames(tabla1)<-c("C(pb)","n","N.percent","GC.percent","PQSs","PQSs.Kb","PQSs.Kb.control")
#####
#3) Guardo la tabla
write.table(tabla1, file="../resultados/Tabla1.csv", dec = ",", sep="\t")
```

Script S6: Análisis por clases de secuencias

```
#Script 6: Analisis por clases de secuencias
library("TrypR")
#########################################################
#1)Abro los archivos con las secuencias clasificadas (común para genomas y controles)
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#2) Cargo la imagen de los clases procesadas de todos los genomas (originales y control)
setwd("./Archivos intermedios")
load("clases.procesadas.RData")
#####
#II.1) Procesamiento de secuencias
#####Frecuencia de cada subclase para los genomas considerados
Tabla2a<-data.frame(matrix(NA, nrow = 5, ncol = length(procesadas.clases.original)))
rownames(Tabla2a)<-c("poliRNA","init-SSR", "init-tel", "term-SSR", "term-tel")
colnames(Tabla2a)<-acronimos
for (i in 1:length(procesadas.clases.original)){
  a<-sum(procesadas.clases.original[[i]]$Subclass=="poliRNA")
  b<-sum(procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]$Subclass=="SSR")
  c<-sum(procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]$Subclass=="tel")
  d<-sum(procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]$Subclass=="SSR")
  e<-sum(procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]$Subclass=="tel")
  Tabla2a[,i]<-c(a,b,c,d,e)
}
#Guardo la tabla:
write.csv(Tabla2a, "../resultados/Tabla2a.csv")
#limpio
rm(list=c("a","b","c","d","e","i"))
#Media y desvio del tamaño (pb) de cada clase secuencias
Tabla2b<-data.frame(matrix(NA, nrow = 10, ncol = length(procesadas.clases.original)))
rownames(Tabla2b)<-c("mean poliRNA","mean init-SSR", "mean init-tel", "mean term-SSR", "mean term-tel","sd poliRNA","sd
init-SSR", "sd init-tel", "sd term-SSR", "sd term-tel")
colnames(Tabla2b)<-acronimos
Datos.estadisticas.II1<-data.frame()#Armo tabla para enviar a Script Estadísticas-Gráficos
for (i in 1:length(procesadas.clases.original)){
  a<-round(mean(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Subclass=="poliRNA",5]))
  b<-round(mean(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]
$Subclass=="SSR",5]))
  c<-round(mean(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]
$Subclass=="tel",5]))
  d<-round(mean(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]
$Subclass=="SSR",5]))
  e<-round(mean(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]
$Subclass=="tel",5]))
  f<-round(sd(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Subclass=="poliRNA",5]))
  g<-round(sd(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]
$Subclass=="SSR",5]))
  h<-round(sd(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="init" & procesadas.clases.original[[i]]
$Subclass=="tel",5]))
  j<-round(sd(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]
$Subclass=="SSR",5]))
  k<-round(sd(procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="term" & procesadas.clases.original[[i]]
$Subclass=="tel",5]))
  Tabla2b[,i]<-c(a,b,c,d,e,f,g,h,j,k)
```

```
 #Cargo datos para estadisticas-Gráficos
 longitud<-data.frame(genoma=0,longitud=procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Subclass=="poliRNA",5])
 longitud[,1]<-acronimos[i]
 Datos.estadisticas.II1<-rbind(Datos.estadisticas.II1,longitud)
}
#############################################################################################
# EstadísticasII.1 y GráficosII.1
#Elimino controles
Datos.estadisticas.II1<-Datos.estadisticas.II1[Datos.estadisticas.II1$genoma!="Control",]
#Evalúo supuestos:
#a)Independencia: SI
#b)Normalidad: muestras grandes (>>30). Aplico TCL.
#c)Homocedasticidad
bartlett.test(Datos.estadisticas.II1[,2]~Datos.estadisticas.II1[,1])
#No hay homocedasticidad. Número muestral similar (balanceadas)
#Aplico test de welch (Anova heterocedástico)
oneway.test(Datos.estadisticas.II1[,2]~Datos.estadisticas.II1[,1])
#Hay diferencias entre genomas para la longitud de PTUs (p<0.05)
#Analizamos que grupos difieren entre si aplicando métodos post-hoc
#Ajuste de Holm
pairwise.t.test(Datos.estadisticas.II1[,2], Datos.estadisticas.II1[,1], p.adj = "holm")
#La diferencia significativas se dán entre géneros, pero no dentro de cada genero.
jpeg("../resultados/II.1-boxplot.jpg",width = 750, height = 450, quality = 100)
grafico<-Datos.estadisticas.II1[Datos.estadisticas.II1[,1]!="Control",]
levels(grafico[,1])[1]<-"L. donovani"
boxplot((grafico[,2]/1000)~grafico[,1], col=c(2:8), ylab="Tamaño (Kb)", xlab="Genomas", main="Distribución tamaños PTUs")
points(1:7, (Tabla2b[1,c(2:8)])/1000,  pch="*",cex=1.5 ,col = 1)
text(1.2:7.2, 700,c("A","A","A", "B", "B","B","B"), cex=1.1)
dev.off()
#############################################################################################
#Media y desvio de numero de genes en los ARN policistronicos
Tabla2c<-data.frame(matrix(NA, nrow = 2, ncol = length(procesadas.clases.original)))
rownames(Tabla2c)<-c("mean seq/poliRNA","sd seq/poliRNA")
colnames(Tabla2c)<-acronimos
for (i in 1:length(procesadas.clases.original)){
 a<-print(mean((procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Subclass=="poliRNA",9])))
 f<-print(sd((procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Subclass=="poliRNA",9])))
 Tabla2c[,i]<-c(a,f)
}
#Guardo las tablas
write.csv(Tabla2b, "../resultados/Tabla2b.csv")
write.csv(Tabla2c, "../resultados/Tabla2c.csv")
#############################################################################################
#II.2) Densidad de PQSs y contenido GC
#####
#Obtengo gráficos y tablas con promedios GC y PQSs/Kb (PONDERADO por tamaño de secuencia)
#####PQS/Kb y GC en clases
tabla<-data.frame(row.names = c("init-SSR","init-tel","poliRNA","term-SSR","term-tel"))
tabla3<-tabla#ponderada
tabla3.cont<-tabla#ponderada
tabla4<-tabla#ponderada
Datos.estadisticas.II2<-list()#Armo una lista para enviar a Script Estadísticas-Gráficos
Datos.estadisticas.II2.cont<-list()#Armo una lista para enviar a Script Estadísticas-Gráficos
for (i in 1:length(procesadas.clases.original)){
 filtradas<-procesadas.clases.original[[i]][procesadas.clases.original[[i]]$N!=procesadas.clases.original[[i]]$width,]#no considero
las secuencias con 100% de N
 filtradas$Class<-paste0(filtradas$Class,"-",filtradas$Subclass)#Fusiono etiquetas clases-subclases
 filtradas$Class<-gsub("-poliRNA","",filtradas$Class)#Borro texto innecesario
 filtradas.c<-procesadas.clases.control[[i]][procesadas.clases.control[[i]]$N!=procesadas.clases.original[[i]]$width,]#no considero
las secuencias con 100% de N
 filtradas.c$Class<-paste0(filtradas.c$Class,"-",filtradas.c$Subclass)#Fusiono etiquetas clases-subclases
 filtradas.c$Class<-gsub("-poliRNA","",filtradas.c$Class)#Borro texto innecesario
 #Obtengo densidad media PQSs/Kb (PONDERADA)
 #Obtengo los PQSs totales por clase
```

```
  n.PQSs<-aggregate(filtradas$g4number,by=list(filtradas$Class),FUN=sum)
  #Obtengo la longitud total (efectiva) de cada clase
  long.total.clases<-aggregate(filtradas$LongEf,by=list(filtradas$Class),FUN=sum)#suma de longitudes de cada clase
  #Obtengo la densidad PQSs/Kb por clase (multiplico por 500 porque la longitud de clase solo considera una hebra)
  n.PQSs$x<-round((500*n.PQSs$x/long.total.clases$x),digits = 4)
  #Repito para controles
  n.PQSs.c<-aggregate(filtradas.c$g4number,by=list(filtradas.c$Class),FUN=sum)
  n.PQSs.c$x<-round((500*n.PQSs.c$x/long.total.clases$x),digits = 4)
  #Obtengo el porcentaje medio %GC (PONDERADO)
  #Obtengo los GC totales por clase
  n.GCs<-aggregate(filtradas$CG,by=list(filtradas$Class),FUN=sum)
  #Obtengo el %GC por clase
  n.GCs$x<-round((100*n.GCs$x/long.total.clases$x),digit=1)
  # # #Agrego medias a la tabla
  tabla3[1:5,i]<-n.PQSs[,2]
  tabla3.cont[1:5,i]<-n.PQSs.c[,2]
  tabla4[1:5,i]<-n.GCs[,2]
  #Cargo datos para estadisticas-Gráficos
  GC.percent<-100*filtradas[,10]/filtradas[,15]
  salida<-data.frame(genoma=0,filtradas[,c(7,15,16,17,10)],GCpercent=GC.percent, filtradas[,c(13,14)])
  salida[,1]<-acronimos[i]
  Datos.estadisticas.II2[[i]]<-salida
  GC.percent<-100*filtradas.c[,10]/filtradas.c[,15]
  salida<-data.frame(genoma=0,filtradas.c[,c(7,15,16,17,10)],GCpercent=GC.percent, filtradas.c[,c(13,14)])
  salida[,1]<-acronimos[i]
  Datos.estadisticas.II2.cont[[i]]<-salida
}
#############################################################################################################
#EstadísticasII.2 y GráficosII.2
#II.2a) Estadísticas Densidades de PQSs por cada clase de secuencia
#Por un lado evalúo diferencias intragrupo, y por otro diferencias con los controles
#Supuestos:
#a)Independencia: SI
#b)Normalidad: muestras grandes (>>30). Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  bt<-bartlett.test(x[,5]~x[,2])
  print(bt[[3]])
}
#No hay homocedasticidad en ningún genoma. número muestral no similares!!!
#Aplico test de welch (Anova heterocedostico)
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  owt<-oneway.test(x[,5]~x[,2])
  print(owt[[3]])
}
#Obtengo p<0.01 en el genoma 1 a 4 (Leishmania spp y Tb)
#Hay diferencias entre los grupos
#Analizamos que grupos difieren entre si aplicando métodos post-hoc
#Ajuste de Holm
diferencias.interclases<-list()
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  x$Class<-as.factor(x$Class)
  ptt<-pairwise.t.test(x[,5],x[,2], p.adj = "holm")
  print(as.character(x[1,1]))
  print(ptt[[3]])
  diferencias.interclases[[i]]<-ptt[[3]]
}
#En los genomas de Leishmania hay diferencias entre las tres clases (a,b,c)
```

```
#En Tb solo hay diferencias entre poliRNA y term (ab,a,b)
#En el resto de los Trypanosomas no hay diferencias estadísticamente significativas entre las clases (a,a,a)
letras<-matrix(data=c(rep(c("a","b","c"),3),"ab","a","b",rep("a",9)), ncol = 3, byrow = T)
#Analizo diferencias con los controles respectivos
diferencias.clases.control<-matrix(NA,nrow=7,ncol=3)
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  x$Class<-as.factor(x$Class)
  x$trat<-"trat"
  y<-Datos.estadisticas.II2.cont[[i]]
  y<-y[y[,2]!="init-tel"&y[,2]!="term-tel",]#Elimino telómeros
  y$trat<-"cont"
  #y$Class<-paste0("control ",y$Class)
  z<-rbind(x,y)
  z1<-z[z$Class=="init-SSR",]
  diferencias.clases.control[i-1,1]<-round(t.test(z1$g4.Kb~z1$trat)[[3]], digits=4)
  z1<-z[z$Class=="poliRNA",]
  diferencias.clases.control[i-1,2]<-round(t.test(z1$g4.Kb~z1$trat)[[3]], digits=4)
  z1<-z[z$Class=="term-SSR",]
  diferencias.clases.control[i-1,3]<-round(t.test(z1$g4.Kb~z1$trat)[[3]], digits=4)
}
asteriscos<-diferencias.clases.control
asteriscos[diferencias.clases.control<=0.05]<-"*"
asteriscos[diferencias.clases.control>0.05]<-""
significancia<-matrix(paste0(letras,asteriscos), ncol = 3, byrow = F)
#II.2a)Grafico:
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  x$Class<-as.factor(x$Class)
  y<-Datos.estadisticas.II2.cont[[i]]
  y<-y[y[,2]!="init-tel"&y[,2]!="term-tel",]#Elimino telómeros
  y$Class<-paste0("control ",y$Class)
  z<-rbind(x,y)
  if(i<=4){ylim<-4}else(ylim<-0.8)
  colores<-c("chartreuse3","lightgoldenrod3","cadetblue3","chartreuse1","lightgoldenrod1","cadetblue1")
  jpeg(paste0("../resultados/II.2-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  boxplot(z[,5]~z[,2],cex=0.7,xaxt = "n",cex=0.7,ylim=c(0,ylim),col=colores, par(mar=c(5,5,4,2)))
  axis(1,at = 1:6,  labels =c("SSRi","PTUs","SSRt","SSRi(c)","PTUs(c)","SSRt(c)"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n PQSs/Kb por Clase"), cex.main=1.35, xlab=list("Clases de secuencias",par(cex.lab=1.25)),
ylab=list("densidad (PQSs/Kb)",par(cex.lab=1.25)))
  medias.p<-500*aggregate(x[,4],by=list(x[,2]),FUN=sum)[,2]/aggregate(x[,3],by=list(x[,2]),FUN=sum)[,2]#Medias ponderadas
  medias.p.c<-500*aggregate(y[,4],by=list(y[,2]),FUN=sum)[,2]/aggregate(y[,3],by=list(y[,2]),FUN=sum)[,2]#Medias ponderadas
  points(1:6, c(medias.p,medias.p.c),  pch="_",cex=3 ,col = 2)
  abline(v=3.5, lty=2)
  text(c(1.3:3.3),9*ylim/10,significancia[i-1,1:3], cex=1.25)
  dev.off()
}
################################################################################
#II.2b) Estadísticas contenido GC por cada clase de secuencia
#Supuestos:
#a)Independencia: SI
# #b)Normalidad: muestras grandes (>30), entonces aplico directamente TCL.
#c)Homocedasticidad
for (i in 1:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  bt<-bartlett.test(x[,7]~x[,2])
  print(bt[[3]])
}
#No hay homocedasticidad. Número muestral no similares!!!
# Aplico test de welch (Anova heterocedástico)<-?Es correcto?
for (i in 1:8){
```

```
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  owt<-oneway.test(x[,7]~x[,2])
  print(owt[[3]])
}
#Obtengo p<0.01 en todos los genmas, menos en Tb (p>0.05).
#Hay diferencias entre los grupos en ltodos los genomas menos en Tb.
#Analizamos que grupos difieren entre si aplicando métodos post-hoc
#Ajuste de Holm
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  ptt<-pairwise.t.test(x[,7],x[,2], p.adj = "holm")
  print(as.character(x[1,1]))
  print(ptt[[3]])
}
#En los tres Leishmania la diferencia se da entre inicio y el resto de las secuencias (p<0.01)
#En Tb no hay diferencias
#En TcE la diferencia se da entre todas las secuencias (p<0.01)
#En TcnE la diferencia se da entre todas las secuencias (p<0.05)
#En TcnE la diferencia se da entre inicio y poliRNA (p<0.05)
#En Te la diferencia se da entre poliRNA y el resto de las secuencias (p<0.05)
significancia.GC<-matrix(data=c(rep(c("a","b","b"),3),"a","a","a","a","b","c","a","b","ab","a","b","a"), ncol = 3, byrow = T)
#II.2b) Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.II2[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  colores<-c("chartreuse3","lightgoldenrod3","cadetblue3")
  jpeg(paste0("../resultados/II.2b-",i,'-boxplot.jpg'),width = 400, height = 450, quality = 100)
  boxplot(x[,7]~x[,2],ylim=c(0,100),col=colores,xaxt = "n", par(mar=c(5,5,4,2)))
  axis(1, at = c(1,2,3), labels =c("SSRi","PTUs","SSRt"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n %GC por Clase"), cex.main=1.35, xlab=list("secuencias",par(cex.lab=1.25)),
ylab=list("%GC",par(cex.lab=1.25)))
  medias<-aggregate(x[,7],by=list(x[,2]),FUN=mean)#Medias no pnderadas!!!!!!
  medias.p<-100*aggregate(x[,6],by=list(x[,2]),FUN=sum)[,2]/aggregate(x[,3],by=list(x[,2]),FUN=sum)[,2]#Medias ponderadas
  points(1:3, medias.p,  pch="_",cex=4 ,col = 2)
  text(c(1.2:3.2),90,significancia.GC[i-1,1:3], cex=1.25)
  dev.off()
}
##############################################################################################
#Guardo las tablas
colnames(tabla3)<-acronimos
colnames(tabla3.cont)<-acronimos
colnames(tabla4)<-acronimos
write.table(tabla3, file="../resultados/Tabla3.csv", dec = ",", sep="\t")
write.table(tabla3.cont, file="../resultados/Tabla3cont.csv", dec = ",", sep="\t")
write.table(tabla4, file="../resultados/Tabla4.csv", dec = ",", sep="\t")
##############################################################################################
#II.3) Distribución de PQSs entre clases de secuencias
acronimos<-unlist(read.csv("../Acronimos.csv",header = F))
load("clases.procesadas.RData")
load("g4.RData")
#Graficos distribucion
for (i in 2:length(procesadas.clases.original)){
  #Paso la tabla a formato GRange para que funcione la función
  x<-procesadas.clases.original[[i]]
  clases<-GRanges(seqnames = x$seqnames, ranges = IRanges(start=x$start,end = x$end), strand = x$strand)
  mcols(clases)<-x[,c(7,8)]
  #Gráfico
  titulo<-paste0(acronimos[i], "\n Distribucion ssPQSs en las subclases de secuencias")
  a<-PQSdistributionV2(sec=clases, g4=g4.original[[i]], zero="end", upstream=1, downstream = 2500, exclude = "tel", title = titulo,
relative=T, max.score=NULL, min.score=86, ss=T, xlab="Posicion (nt)", ylab="Frecuencia (PQSs/secuencia)", graph=F)
  b<-PQSdistributionV2(sec=clases, g4=g4.control[[i]], zero="end", upstream=1, downstream = 2500, exclude = "tel", title = titulo,
relative=T, max.score=NULL, min.score=86, ss=T, xlab="Posicion (nt)", ylab="Frecuencia (PQSs/secuencia)", graph=F)
```

```
  jpeg(paste0("../resultados/II.3SSRt-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  plot<-plot(a[[1]][,1],a[[1]][,2],type="l", ylim = c(-0.25, 0.15), col="2", main=titulo, xlab="Posición (nt)", ylab="Frecuencia
(PQSs/Secuencia)", sub=a[[3]])
  lines(a[[2]][,1],a[[2]][,2], col="4", lwd=1)
  lines(b[[1]][,1],b[[1]][,2], col="1", lty= 3)
  lines(b[[2]][,1],b[[2]][,2], col="1", lty= 3)
  v<-abline(v=0)
  h<-abline(h=0)
  dev.off()
}
```

Script S7: Análisis por tipos de secuencias

```
#Script 7: Analisis por tipos de secuencias
library("TrypR")
############################################################
#1)Abro los archivos con las secuencias clasificadas (común para genomas y controles)
acronimos<-unlist(read.csv("Acronimos.csv",header = F))
#2) Cargo la imagen de los clases procesadas de todos los genomas (originales y control)
setwd("./Archivos intermedios")
load("clases.procesadas.RData")
load("tipos.procesados.RData")
#####
#III.1) Densidad de PQSs y contenido GC (genomas originales y control)
####PQS/KB y GC en tipos
tabla<-data.frame(row.names = c("genic","intergenic"))
tabla5<-tabla
tabla6<-tabla
tabla7<-tabla
tabla6cont<-tabla
Datos.estadisticas.III1<-list()
Datos.estadisticas.III1.cont<-list()
for (i in 1:length(procesadas.tipos.original)){
  filtradas<-procesadas.tipos.original[[i]][procesadas.tipos.original[[i]]$Class=="poliRNA",]#no considero las secuencias con 100%
de N
  tabla5[1:2,i]<-rbind(nrow(filtradas[filtradas$type=="genic",]),nrow(filtradas[filtradas$type=="intergenic",]))
  filtradas<-filtradas[filtradas$N!=filtradas$width,]#no considero las secuencias con 100% de N
  filtradas<-filtradas[filtradas$type=="genic"|filtradas$type=="intergenic",]
  filtradas.c<-procesadas.tipos.control[[i]][procesadas.tipos.control[[i]]$Class=="poliRNA",]#no considero las secuencias con 100%
de N
  filtradas.c<-filtradas.c[filtradas.c$N!=filtradas.c$width,]#no considero las secuencias con 100% de N
  filtradas.c<-filtradas.c[filtradas.c$type=="genic"|filtradas.c$type=="intergenic",]
  #Obtengo densidad media PQSs/Kb (PONDERADA)
  #Obtengo los PQSs totales por tipo
  n.PQSs<-aggregate(filtradas$g4number,by=list(filtradas$type),FUN=sum)
  n.PQSs.c<-aggregate(filtradas.c$g4number,by=list(filtradas.c$type),FUN=sum)
  #Obtengo la longitud total (efectiva) de cada tipo
  long.total.type<-aggregate(filtradas$LongEf,by=list(filtradas$type),FUN=sum)#suma de longitudes de cada tipo
  #Obtengo la densidad PQSs/Kb por tipo (multiplico por 500 porque la longitud de tipo solo considera una hebra)
  n.PQSs$x<-round(500*n.PQSs$x/long.total.type$x, digits = 4)
  n.PQSs.c$x<-round(500*n.PQSs.c$x/long.total.type$x, digits = 4)
  #Obtengo el porcentaje medio %GC (PONDERADO)
  #Obtengo los GC totales por clase
  n.GCs<-aggregate(filtradas$CG,by=list(filtradas$type),FUN=sum)
  #Obtengo el %GC por tipo
  n.GCs$x<-round(100*n.GCs$x/long.total.type$x, digits = 1)
  #Agrego medias a la tabla
  tabla6[1:2,i]<-n.PQSs[,2]
  tabla7[1:2,i]<-n.GCs[,2]
  tabla6cont[1:2,i]<-n.PQSs.c[,2]
  #Cargo datos para estadisticas-Gráficos
  GC.percent<-100*filtradas[,13]/filtradas[,18]
  salida<-data.frame(genoma=0,filtradas[,c(7,19,20,21,14)],GCpercent=GC.percent, filtradas[,c(17,18)])
```

```
  salida[,2]<-as.factor(as.character(salida[,2]))
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III1[[i]]<-salida
  GC.percent<-100*filtradas.c[,13]/filtradas.c[,18]
  salida<-data.frame(genoma=0,filtradas.c[,c(7,19,20,21,14)],GCpercent=GC.percent, filtradas.c[,c(17,18)])
  salida[,2]<-as.factor(as.character(salida[,2]))
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III1.cont[[i]]<-salida
}
#################################################################################################
# EstadísticasIII.1 y GráficosIII.1
#Supuestos:
#a)Independencia: SI
#b)Normalidad: muestras grandes (>30).Aplico TCL.
# x<-Datos.estadisticas.III1[[4]]
# boxplot(x$g4.Kb~x$type)
# boxplot(log(x$g4.Kb+1)~x$type)
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  vt<-var.test(x[,4]~x[,2])#Para dos grupos
  print(vt[[3]])
}
#No hay homocedasticidad. Número de muestral similares!!!
# Aplico t.test con varianzas no iguales
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  tt<-t.test(x[,4]~x[,2],var.equal=FALSE)
  print(tt[[3]])
}
#Obtengo p<0.01 en tods los genomas (diferencias significativas)
#Analizo diferencias con los controles respectivos
letras<-matrix(rep(c("a","b"),7),nrow=7,ncol=2, byrow = T)
#Contra controles
diferencias.clases.control<-matrix(NA,nrow=7,ncol=2)
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  y<-Datos.estadisticas.III1.cont[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  y<-y[y[,2]!="init-tel"&y[,2]!="term-tel",]#Elimino telómeros
  x$trat<-"trat"
  y$trat<-"cont"
  z<-rbind(x,y)
  z1<-z[z$type=="genic",]
  diferencias.clases.control[i-1,1]<-round(t.test(z1$g4.Kb~z1$trat)[[3]], digits=4)
  z1<-z[z$type=="intergenic",]
  diferencias.clases.control[i-1,2]<-round(t.test(z1$g4.Kb~z1$trat)[[3]], digits=4)
}
asteriscos<-diferencias.clases.control
asteriscos[diferencias.clases.control<=0.05]<-"*"
asteriscos[diferencias.clases.control>0.05]<-""
significancia<-matrix(paste0(letras,asteriscos), ncol = 2, byrow = F)
#III.1a) Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  y<-Datos.estadisticas.III1.cont[[i]]
  y$type<-paste0("control ",y$type)
  z<-rbind(x,y)
  colores<-c("cyan3","darkgoldenrod3","cyan1","darkgoldenrod1")
  if(i<=4){ylim<-4}else(ylim<-1)
  jpeg(paste0("../resultados/III.1-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  boxplot(z[,4]~z[,2],cex=0.7,xaxt = "n",cex=0.7,ylim=c(0,ylim),col=colores, par(mar=c(5,5,4,2)))
  axis(1,at = 1:4,  labels =c("genica","intergenica","genica(c)","intergenica(c)"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n PQSs/Kb por Tipo"), cex.main=1.35, xlab=list("Tipos de secuencias",par(cex.lab=1.25)),
```

```
ylab=list("densidad (PQSs/Kb)",par(cex.lab=1.25)))
  medias.p<-c(tabla6[,i],tabla6cont[,i])#Medias ponderadas
  points(1:4, medias.p, pch="_", cex=4 ,col = 2)
  abline(v=2.5, lty=2)
  text(c(1.3,2.3),9*ylim/10,significancia[i-1,1:2], cex=1.25)
  dev.off()
}
#####################################################
##III.1b) Contenido GC por cada tipo de secuencia
#Supuestos:
#a)Independencia: SI
#b)Normalidad: muestras grandes (>30).Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  vt<-var.test(x[,7]~x[,2])#Para dos grupos
  print(vt[[3]])
}
#No hay homocedasticidad. Número muestral similares.
# Aplico t.test con varianzas no iguales
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  tt<-t.test(x[,7]~x[,2],var.equal=FALSE)
  print(tt[[3]])
}
#Obtengo p<0.01 en tods los genomas
significancia.GC<-matrix(rep(c("a","b"),7),nrow=7,ncol=2, byrow = T)
#III.1b)Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III1[[i]]
  colores<-c("cyan3","darkgoldenrod3")
  jpeg(paste0("../resultados/III.1b-",i,'-boxplot.jpg'),width = 400, height = 450, quality = 100)
  boxplot(x[,7]~x[,2],ylim=c(0,100),col=colores,xaxt = "n", par(mar=c(5,5,4,2)))
  axis(1, at = c(1,2), labels =c("genica","intergenica"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n %GC por Tipo"), cex.main=1.35, xlab=list("secuencias",par(cex.lab=1.25)),
ylab=list("%GC",par(cex.lab=1.25)))
  medias.p<-tabla7[,i]#Medias ponderadas
  points(1:2, medias.p, pch="_", cex=4 ,col = 2)
  text(c(1.2,2.2),90,significancia.GC[i-1,1:2], cex=1.25)
  dev.off()

}
#################################################################################################
colnames(tabla5)<-acronimos
colnames(tabla6)<-acronimos
colnames(tabla7)<-acronimos
colnames(tabla6cont)<-acronimos
write.table(tabla5, file="../resultados/tabla5.csv", dec = ",", sep="\t")
write.table(tabla6, file="../resultados/tabla6.csv", dec = ",", sep="\t")
write.table(tabla7, file="../resultados/tabla7.csv", dec = ",", sep="\t")
write.table(tabla6cont, file="../resultados/tabla6cont.csv", dec = ",", sep="\t")
#################################################################################################
#III.2.a) Densidad de PQS y %G por hebra
tabla<-data.frame(row.names = c("coding","template"))
tabla8<-tabla#Medias contenidos PQSs/Kb por hebra
tabla9<-tabla#Medias contenidos G por hebra
tabla8.cont<-tabla#Medias contenidos PQSs/Kb por hebra
Datos.estadisticas.III2<-list()
Datos.estadisticas.III2.cont<-list()
for (i in 1:length(procesadas.clases.original)){
  filtradas<-procesadas.clases.original[[i]][procesadas.clases.original[[i]]$Class=="poliRNA",]#Trabajo solo con ARNpolicistronicos
  filtradas<-filtradas[filtradas$N!=filtradas$width,]#no considero las secuencias con 100% de N
  #Agrego columnas con número y densidad PQS en la hebra complementaria
  filtradas$g4outStrand<-filtradas$g4number-filtradas$g4inStrand
```

```
  filtradas$g4outStrand.Kb<-1000*filtradas$g4outStrand/filtradas$LongEf
  #Agrego columna de longitudes corregidas
  #Agrego columna de ponderacion de longitud
  filtradas.c<-procesadas.clases.control[[i]][procesadas.clases.control[[i]]$Class=="poliRNA",]#Trabajo solo con
ARNpolicistronicos
  filtradas.c<-filtradas.c[filtradas.c$N!=filtradas.c$width,]#no considero las secuencias con 100% de N
  #Agrego columnas con número y densidad PQS en la hebra complementaria
  filtradas.c$g4outStrand<-filtradas.c$g4number-filtradas.c$g4inStrand
  filtradas.c$g4outStrand.Kb<-1000*filtradas.c$g4outStrand/filtradas.c$LongEf
  #Obengo medias ponderadas de PQSs
  media.instrand.p<-round(1000*sum(filtradas$g4inStrand)/sum(filtradas$LongEf),digits=4)
  media.outstrand.p<-round(1000*sum(filtradas$g4outStrand)/sum(filtradas$LongEf),digits=4)
  media.instrand.p.c<-round(1000*sum(filtradas.c$g4inStrand)/sum(filtradas.c$LongEf),digits=4)
  media.outstrand.p.c<-round(1000*sum(filtradas.c$g4outStrand)/sum(filtradas.c$LongEf),digits=4)
  tabla8[1:2,i]<-rbind(coding=media.instrand.p,template=media.outstrand.p)
  tabla8.cont[1:2,i]<-rbind(coding=media.instrand.p.c,template=media.outstrand.p.c)
  #Obtengo contenido G para ambas hebras
  mediaG.instrand.p<-round(100*sum(filtradas$GinStrand)/sum(filtradas$LongEf),digits=1)
  mediaG.outstrand.p<-round(100*sum(filtradas$CG-filtradas$GinStrand)/sum(filtradas$LongEf),digits=1)
  tabla9[1:2,i]<-rbind(coding=mediaG.instrand.p,template=mediaG.outstrand.p)
  #Cargo datos para estadisticas-Gráficos
  GpercentInStrand<-100*filtradas$GinStrand/filtradas$LongEf
  GoutStrand<-filtradas$CG-filtradas$GinStrand
  GpercentOutStrand<-100*GoutStrand/filtradas$LongEf
  salida<-
data.frame(genoma=0,LongEf=filtradas$LongEf,GinStrand=filtradas$GinStrand,GpercentInStrand,GoutStrand,GpercentOutStrand,f
iltradas[,c(18:21)])
  #salida[,2]<-as.factor(as.character(salida[,2]))
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III2[[i]]<-salida
  #Cargo datos para estadisticas-Gráficos
  GpercentInStrand<-100*filtradas.c$GinStrand/filtradas.c$LongEf
  GoutStrand<-filtradas.c$CG-filtradas.c$GinStrand
  GpercentOutStrand<-100*GoutStrand/filtradas.c$LongEf
  salida<-
data.frame(genoma=0,LongEf=filtradas.c$LongEf,GinStrand=filtradas.c$GinStrand,GpercentInStrand,GoutStrand,GpercentOutStra
nd,filtradas.c[,c(18:21)])
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III2.cont[[i]]<-salida
}
#############################################################################################################
#EstadísticasIII.2 y GráficosII.1
#III.2a) Densidades de PQSs por cada hebra en ARN policistronic
#Supuestos:
#a)Independencia:No (muestras apareadas)
#b)Normalidad: muestras grandes (>30).Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  vt<-var.test(x$g4inStrand.Kb,x$g4outStrand.Kb)
  print(vt[[3]])
}
#No hay homocedasticidad. Número muestral identico (APAREADAS)
#Aplico paired t test, al tratarse de muestras apareadas (dos hebras de una misma secuencia en cada caso)
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  tt<-t.test(x$g4inStrand.Kb,x$g4outStrand.Kb,var.equal=FALSE, paired = T)
  print(tt[[3]])
}
#Tb y Te no presentan diferencias estadísticamente significativas entre las hebras
letras<-matrix(c(rep(c("a","b"),3), "a","a",rep(c("a","b"),2),"a","a"),nrow=7,ncol=2, byrow = T)
#Contra controles
diferencias.hebras.control<-matrix(NA,nrow=7,ncol=2)
for (i in 2:8){
```

```
  x<-Datos.estadisticas.III2[[i]]
  y<-Datos.estadisticas.III2.cont[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  y<-y[y[,2]!="init-tel"&y[,2]!="term-tel",]#Elimino telómeros
  x$trat<-"trat"
  y$trat<-"cont"
  z<-rbind(x,y)
  diferencias.hebras.control[i-1,1]<-round(t.test(z$g4inStrand.Kb~z$trat)[[3]], digits=4)
  diferencias.hebras.control[i-1,2]<-round(t.test(z$g4outStrand.Kb~z$trat)[[3]], digits=4)
}
asteriscos<-diferencias.hebras.control
asteriscos[diferencias.hebras.control<=0.05]<-"*"
asteriscos[diferencias.hebras.control>0.05]<-""
significancia<-matrix(paste0(letras,asteriscos), ncol = 2, byrow = F)
#III.2a)Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  y<-Datos.estadisticas.III2.cont[[i]]
  colores<-c("blue3","brown3","blue1","brown1")
  if(i<=4){ylim<-4}else(ylim<-1)
  jpeg(paste0("../resultados/III.2-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  boxplot(list(x$g4inStrand.Kb,x$g4outStrand.Kb,y$g4inStrand.Kb,y$g4outStrand.Kb),cex=0.7,xaxt =
"n",cex=0.7,ylim=c(0,ylim),col=colores, par(mar=c(5,5,4,2)))
  axis(1,at = 1:4,  labels =c("codificante","molde","codificante(c)","molde(c)"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n PQSs/Kb por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)), ylab=list("densidad
(PQSs/Kb)",par(cex.lab=1.25)))
  medias<-apply(cbind(x$g4inStrand.Kb,x$g4outStrand.Kb,y$g4inStrand.Kb,y$g4outStrand.Kb),2,FUN=mean)#Medias no
pnderadas!!!!!!
  medias.p<-1000*apply(cbind(x$g4inStrand,x$g4outStrand,y$g4inStrand,y$g4outStrand),2,FUN=sum)/sum(x$LongEf)
  points(1:4, medias.p, pch="_", cex=4 ,col = "green")
  abline(v=2.5, lty=2)
  text(c(1.3,2.3),9*ylim/10,significancia[i-1,1:2], cex=1.25)
  dev.off()
}
###########
#III.2b) Contenido de G por cada hebra en ARN policistronico
#Supuestos:
#a)Independencia:No
#b)Normalidad: muestras grandes (>30).Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  vt<-var.test(x$GpercentInStrand,x$GpercentOutStrand)
  print(vt[[3]])
}
#No hay homocedasticidad. Número muestral identico (APAREADAS)
#Aplico paired t test, al tratarse de muestras apareadas (dos hebras de una misma secuencia en cada caso)
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  tt<-t.test(x$GpercentInStrand,x$GpercentOutStrand,var.equal=FALSE, paired = T)
  print(tt[[3]])
}
#Obtengo no diferencias solo en Ld y Lm.
significancia.GC<-matrix(c("a","a","a","b","a","a",rep(c("a","b"),4)),nrow=7,ncol=2, byrow = T)
#III.2b) Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III2[[i]]
  colores<-c("blue3","brown3")
  jpeg(paste0("../resultados/III.2b-",i,'-boxplot.jpg'),width = 400, height = 450, quality = 100)
  boxplot(list(x$GpercentInStrand,x$GpercentOutStrand),ylim=c(0,50),col=colores,xaxt = "n", par(mar=c(5,5,4,2)))
  axis(1, at = c(1,2), labels =c("codificante","molde"), par(cex=1.25)  , tick = TRUE)
  title(main=paste0(x[1,1], "\n %G por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)),
ylab=list("%G",par(cex.lab=1.25)))
  medias.p<-100*apply(cbind(x$GinStrand,x$GoutStrand),2,FUN=sum)/sum(x$LongEf)
```

```
  points(1:2, medias.p, pch="_", cex=4 ,col = "green")
  text(c(1.2,2.2),45,significancia.GC[i-1,1:2], cex=1.25)
  dev.off()
}
#################################################################################################
colnames(tabla8)<-acronimos
colnames(tabla9)<-acronimos
colnames(tabla8.cont)<-acronimos
write.csv(tabla8,"../resultados/tabla8.csv")#Densidad PQSs/Kb por hebra ARNpoli
write.csv(tabla9,"../resultados/tabla9.csv")#Contenido G por hebra ARNpoli
write.csv(tabla8.cont,"../resultados/tabla8.cont.csv")
#################################################################################################
#III.3) Densidad de PQS y %G por hebra y tipo de secuencia (Mismo análisis anterior, pero
#trabajando en forma independiente con las secuencias génicas y las intergenicas)
#####
tabla<-data.frame(row.names = c("genic coding","genic template","intergenic coding","intergenic template"))
tabla10<-tabla#densidad PQSs/Kb por hebra y tipo
tabla10cont<-tabla#densidad PQSs/Kb por hebra y tipo
tabla11<-tabla#Contenido G por hebra y tipo
Datos.estadisticas.III3<-list()
Datos.estadisticas.III3.cont<-list()
for (i in 1:length(procesadas.tipos.original)){
  filtradas<-procesadas.tipos.original[[i]][procesadas.tipos.original[[i]]$Class=="poliRNA",]#Selecciono ARNpoli
  filtradas<-filtradas[filtradas$N!=filtradas$width,]#no considero las secuencias con 100% de N
  filtradas<-filtradas[filtradas$type=="genic"|filtradas$type=="intergenic",]
  #Agrego columnas con número y densidad PQS en la hebra complementaria
  filtradas$g4outStrand<-filtradas$g4number-filtradas$g4inStrand
  filtradas$g4outStrand.Kb<-1000*filtradas$g4outStrand/filtradas$LongEf
  #Separo en génicas e intergénicas
  filtradas.g<-filtradas[filtradas$type=="genic",]
  filtradas.ig<-filtradas[filtradas$type=="intergenic",]
  #Repito para controles
  filtradas.c<-procesadas.tipos.control[[i]][procesadas.tipos.control[[i]]$Class=="poliRNA",]#Selecciono ARNpoli
  filtradas.c<-filtradas.c[filtradas.c$N!=filtradas.c$width,]#no considero las secuencias con 100% de N
  filtradas.c<-filtradas.c[filtradas.c$type=="genic"|filtradas.c$type=="intergenic",]
  #Agrego columnas con número y densidad PQS en la hebra complementaria
  filtradas.c$g4outStrand<-filtradas.c$g4number-filtradas.c$g4inStrand
  filtradas.c$g4outStrand.Kb<-1000*filtradas.c$g4outStrand/filtradas.c$LongEf
  #Separo en génicas e intergénicas
  filtradas.g.c<-filtradas.c[filtradas.c$type=="genic",]
  filtradas.ig.c<-filtradas.c[filtradas.c$type=="intergenic",]
  #Obengo medias ponderadas de PQSs/Kb
  media.instrand.p.g<-round(1000*sum(filtradas.g$g4inStrand)/sum(filtradas.g$LongEf), digits=4)
  media.outstrand.p.g<-round(1000*sum(filtradas.g$g4outStrand)/sum(filtradas.g$LongEf), digits=4)
  media.instrand.p.ig<-round(1000*sum(filtradas.ig$g4inStrand)/sum(filtradas.ig$LongEf), digits=4)
  media.outstrand.p.ig<-round(1000*sum(filtradas.ig$g4outStrand)/sum(filtradas.ig$LongEf), digits=4)
  t<-c(media.instrand.p.g,media.outstrand.p.g,media.instrand.p.ig,media.outstrand.p.ig)
  tabla10[1:4,i]<-t
  #Obengo medias ponderadas de PQSs/Kb(Para controles)
  media.instrand.p.g.c<-round(1000*sum(filtradas.g.c$g4inStrand)/sum(filtradas.g.c$LongEf), digits=4)
  media.outstrand.p.g.c<-round(1000*sum(filtradas.g.c$g4outStrand)/sum(filtradas.g.c$LongEf), digits=4)
  media.instrand.p.ig.c<-round(1000*sum(filtradas.ig.c$g4inStrand)/sum(filtradas.ig.c$LongEf), digits=4)
  media.outstrand.p.ig.c<-round(1000*sum(filtradas.ig.c$g4outStrand)/sum(filtradas.ig.c$LongEf), digits=4)
  t<-c(media.instrand.p.g.c,media.outstrand.p.g.c,media.instrand.p.ig.c,media.outstrand.p.ig.c)
  tabla10cont[1:4,i]<-t
  #Obtengo contenido G para ambas hebras (ponderado)
  mediaG.instrand.p.g<-round(100*sum(filtradas.g$GinStrand)/sum(filtradas.g$LongEf), digits=1)
  mediaG.outstrand.p.g<-round(100*sum(filtradas.g$CG-filtradas.g$GinStrand)/sum(filtradas.g$LongEf), digits=1)
  mediaG.instrand.p.ig<-round(100*sum(filtradas.ig$GinStrand)/sum(filtradas.ig$LongEf), digits=1)
  mediaG.outstrand.p.ig<-round(100*sum(filtradas.ig$CG-filtradas.ig$GinStrand)/sum(filtradas.ig$LongEf), digits=1)
  t<-c(mediaG.instrand.p.g,mediaG.outstrand.p.g,mediaG.instrand.p.ig,mediaG.outstrand.p.ig)
  tabla11[1:4,i]<-t
  #Cargo datos para estadisticas-Gráficos
  GpercentInStrand<-100*filtradas$GinStrand/filtradas$LongEf
```

```
  GoutStrand<-filtradas$CG-filtradas$GinStrand
  GpercentOutStrand<-100*GoutStrand/filtradas$LongEf
  #salida<-
data.frame(genoma=0,type=filtradas$type,LongEf=filtradas$LongEf,GinStrand=filtradas$GinStrand,GpercentInStrand,GoutStrand,
GpercentOutStrand,filtradas[,c(22:25)])
  salida<-
data.frame(genoma=0,type=filtradas$type,LongEf=filtradas$LongEf,GinStrand=filtradas$GinStrand,GpercentInStrand,GoutStrand,
GpercentOutStrand,filtradas[,c(21:24)])
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III3[[i]]<-salida
  GpercentInStrand<-100*filtradas.c$GinStrand/filtradas.c$LongEf
  GoutStrand<-filtradas.c$CG-filtradas.c$GinStrand
  GpercentOutStrand<-100*GoutStrand/filtradas.c$LongEf
  # salida<-
data.frame(genoma=0,type=filtradas.c$type,LongEf=filtradas.c$LongEf,GinStrand=filtradas.c$GinStrand,GpercentInStrand,GoutStr
and,GpercentOutStrand,filtradas.c[,c(22:25)])
  salida<-
data.frame(genoma=0,type=filtradas.c$type,LongEf=filtradas.c$LongEf,GinStrand=filtradas.c$GinStrand,GpercentInStrand,GoutStr
and,GpercentOutStrand,filtradas.c[,c(21:24)])
  salida[,1]<-acronimos[i]
  Datos.estadisticas.III3.cont[[i]]<-salida
}
#############################################################################################################
#EstadísticasIII.3 y GráficosII.3
#III.3a) Densidades de PQSs por cada hebra en genic e intergenic
#Supuestos:
#a)Independencia:No
#b)Normalidad: muestras grandes (>30).Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  vt.g<-var.test(x.g$g4inStrand.Kb,x.g$g4outStrand.Kb)
  vt.ig<-var.test(x.ig$g4inStrand.Kb,x.ig$g4outStrand.Kb)
  print(vt.g[[3]])
  print(vt.ig[[3]])
}
#No hay homocedasticidad. Número muestral muy Similares! (apareadas)
#Aplico t.test con varianzas no iguales (comparo genicas por un lado e intergénicas por el otro)
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  tt.g<-t.test(x.g$g4inStrand.Kb,x.g$g4outStrand.Kb,var.equal=FALSE, paired = T)
  tt.ig<-t.test(x.ig$g4inStrand.Kb,x.ig$g4outStrand.Kb,var.equal=FALSE, paired = T)
  print(as.character(x[1,1]))
  print(tt.g[[3]])
  print(tt.ig[[3]])
}
#Hay diferencia entre hebra codificantes y molde en ambos tipos de secuencias y para todos los genomas
letras<-matrix(rep(c("a","b","a","b"),7),nrow=7,ncol=4, byrow = T)
#Contra controles
diferencias.hebras.control<-matrix(NA,nrow=7,ncol=4)
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  y<-Datos.estadisticas.III3.cont[[i]]
  x<-x[x[,2]!="init-tel"&x[,2]!="term-tel",]#Elimino telómeros
  y<-y[y[,2]!="init-tel"&y[,2]!="term-tel",]#Elimino telómeros
  x$trat<-"trat"
  y$trat<-"cont"
  z<-rbind(x,y)
  z1<-z[z$type=="genic",]
  diferencias.hebras.control[i-1,1]<-round(t.test(z$g4inStrand.Kb~z$trat)[[3]], digits=4)
```

```
  diferencias.hebras.control[i-1,2]<-round(t.test(z$g4outStrand.Kb~z$trat)[[3]], digits=4)
  z1<-z[z$type=="intergenic",]
  diferencias.hebras.control[i-1,3]<-round(t.test(z$g4inStrand.Kb~z$trat)[[3]], digits=4)
  diferencias.hebras.control[i-1,4]<-round(t.test(z$g4outStrand.Kb~z$trat)[[3]], digits=4)
}
asteriscos<-diferencias.hebras.control
asteriscos[diferencias.hebras.control<=0.05]<-"*"
asteriscos[diferencias.hebras.control>0.05]<-""
significancia<-matrix(paste0(letras,asteriscos), ncol = 4, byrow = F)
#III.3a)Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  y<-Datos.estadisticas.III3.cont[[i]]
  y.g<-y[y$type=="genic",]
  y.ig<-y[y$type=="intergenic",]
  colores<-c("blue4","red4","blue1","red1")
  if(i<=4){ylim<-4}else(ylim<-1)
  z.g<-list(coding=x.g$g4inStrand.Kb,temp=x.g$g4outStrand.Kb,coding.cont=y.g$g4inStrand.Kb,temp.cont=y.g$g4outStrand.Kb)
  z.ig<-
list(coding=x.ig$g4inStrand.Kb,temp=x.ig$g4outStrand.Kb,coding.cont=y.ig$g4inStrand.Kb,temp.cont=y.ig$g4outStrand.Kb)
  jpeg(paste0("../resultados/III.3(genic)-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  boxplot(z.g,cex=0.7,xaxt = "n",cex=0.7,ylim=c(0,ylim),col=colores, par(mar=c(5,5,4,2)))
  axis(1,at = 1:4,  labels =c("codificante","molde","codificante(c)","molde(c)"), par(cex=1.25)  , tick = TRUE)
  title( main=paste0(x[1,1], "\n PQSs/Kb genico por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)),
ylab=list("densidad (PQSs/Kb)",par(cex.lab=1.25)))
  medias.g<-1000*apply(cbind(x.g$g4inStrand,x.g$g4outStrand),2,FUN=sum)/sum(x.g$LongEf)
  medias.g[3:4]<-1000*apply(cbind(y.g$g4inStrand,y.g$g4outStrand),2,FUN=sum)/sum(x.g$LongEf)
  points(1:4, medias.g, pch="_", cex=4 ,col = "green")
  abline(v=2.5, lty=2)
  text(c(1.3:2.3),9*ylim/10,significancia[i-1,1:2], cex=1.25)
  dev.off()
  jpeg(paste0("../resultados/III.3(intergenic)-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  boxplot(z.ig,cex=0.7,xaxt = "n",cex=0.7,ylim=c(0,ylim),col=colores, par(mar=c(5,5,4,2)))
  axis(1,at = 1:4,  labels =c("codificante","molde","codificante(c)","molde(c)"), par(cex=1.25)  , tick = TRUE)
  title( main=paste0(x[1,1], "\n PQSs/Kb intergenico por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)),
ylab=list("densidad (PQSs/Kb)",par(cex.lab=1.25)))
  medias.ig<-1000*apply(cbind(x.ig$g4inStrand,x.ig$g4outStrand),2,FUN=sum)/sum(x.ig$LongEf)
  medias.ig[3:4]<-1000*apply(cbind(y.ig$g4inStrand,y.ig$g4outStrand),2,FUN=sum)/sum(x.ig$LongEf)
  points(1:4, medias.ig, pch="_", cex=4 ,col = "green")
  abline(v=2.5, lty=2)
  text(c(1.3:2.3),9*ylim/10,significancia[i-1,3:4], cex=1.25)
  dev.off()
}
##########
#III.3b) Contenido de G por cada hebra de secuencias genicas e intergénicas
#Supuestos:
#a)Independencia: No
#b)Normalidad: muestras grandes (>30).Aplico TCL.
#c)Homocedasticidad
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  vt.g<-var.test(x.g$GpercentInStrand,x.g$GpercentOutStrand)
  vt.ig<-var.test(x.ig$GpercentInStrand,x.ig$GpercentOutStrand)
  print(vt.g[[3]])
  print(vt.ig[[3]])
}
#En algunos casos SI HAY homocedasticidad. Número muestral Similares! (apareadas)
#Aplico t.test para muestras apareadas
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
```

```
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  tt.g<-t.test(x.g$GpercentInStrand,x.g$GpercentOutStrand, paired = T)
  tt.ig<-t.test(x.ig$GpercentInStrand,x.ig$GpercentOutStrand, paired = T)
  print(as.character(x[1,1]))
  print(tt.g[[3]])
  print(tt.ig[[3]])
}
#Diferencias significativas en todos los pares de hebras (genicas e intergéncias)
significancia.GC<-matrix(rep(c("a","b","a","b"),7),nrow=7,ncol=4, byrow = T)
#III.3b) Gráficos
for (i in 2:8){
  x<-Datos.estadisticas.III3[[i]]
  x.g<-x[x$type=="genic",]
  x.ig<-x[x$type=="intergenic",]
  colores<-c("red","blue")
  z.g<-list(coding=x.g$GpercentInStrand,template=x.g$GpercentOutStrand)
  z.ig<-list(coding=x.ig$GpercentInStrand,template=x.ig$GpercentOutStrand)
  jpeg(paste0("../resultados/III.3(genic)b-",i,'-boxplot.jpg'),width = 400, height = 450, quality = 100)
  boxplot(z.g,ylim=c(0,50),col=colores,xaxt = "n", par(mar=c(5,5,4,2)))
  axis(1, at = c(1,2), labels =c("codificante","molde"), par(cex=1.25)  , tick = TRUE)
  title( main=paste0(x[1,1], "\n %G génico por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)),
ylab=list("%G",par(cex.lab=1.25)))
  medias.g<-100*apply(cbind(x.g$GinStrand,x.g$GoutStrand),2,FUN=sum)/sum(x.g$LongEf)
  points(1:2, medias.g, pch="_", cex=4 ,col = "green")
  text(c(1.2:2.2),45,significancia.GC[i-1,1:2], cex=1.25)
  dev.off()
  jpeg(paste0("../resultados/III.3(intergenic)b-",i,'-boxplot.jpg'),width = 400, height = 450, quality = 100)
  boxplot(z.ig,ylim=c(0,50),col=colores,xaxt = "n", par(mar=c(5,5,4,2)))
  axis(1, at = c(1,2), labels =c("codificante","molde"), par(cex=1.25)  , tick = TRUE)
  title( main=paste0(x[1,1], "\n %G intergénico por hebra"), cex.main=1.35, xlab=list("Hebra",par(cex.lab=1.25)),
ylab=list("%G",par(cex.lab=1.25)))
  medias.ig<-100*apply(cbind(x.ig$GinStrand,x.ig$GoutStrand),2,FUN=sum)/sum(x.ig$LongEf)
  points(1:2, medias.ig, pch="_", cex=4 ,col = "green")
  text(c(1.2:2.2),45,significancia.GC[i-1,3:4], cex=1.25)
  dev.off()
}
#############################################################################################
colnames(tabla10)<-acronimos
colnames(tabla10cont)<-acronimos
colnames(tabla11)<-acronimos
write.csv(tabla10,"../resultados/tabla10.csv")
write.csv(tabla10cont,"../resultados/tabla10cont.csv")
write.csv(tabla11,"../resultados/tabla11.csv")
#############################################################################################
#III.4) Distribución de PQSs entre tipos de secuencias
acronimos<-unlist(read.csv("../Acronimos.csv",header = F))
load("tipos.procesados.RData")
load("g4.RData")
#Graficos distribucion
for (i in 2:length(procesadas.tipos.original)){
  #Paso la tabla a formato GRange para que funcione la función
  x<-procesadas.tipos.original[[i]]
  tipo<-GRanges(seqnames = x$seqnames, ranges = IRanges(start=x$start,end = x$end), strand = x$strand)
  mcols(tipo)<-x[,c(7:12)]
  #Gráfico
  titulo<-paste0(acronimos[i],"\n Distribucion PQSs intra PTUs")
  a<-PQSgenicdistributionV2(sec=tipo, g4=g4.original[[i]], zero="start", upstream=250, downstream = 250, exclude = "tel", title =
titulo, relative=T, max.score=NULL, min.score=86, ss=T, xlab="Posicion (nt)", ylab="Frecuencia (PQSs/secuencia)", graph=F)
  b<-PQSgenicdistributionV2(sec=tipo, g4=g4.control[[i]], zero="start", upstream=250, downstream = 250, exclude = "tel", title =
titulo, relative=T, max.score=NULL, min.score=86, ss=T, xlab="Posicion (nt)", ylab="Frecuencia (PQSs/secuencia)", graph=F)
  jpeg(paste0("../resultados/III.4start-",i,'-boxplot.jpg'),width = 700, height = 450, quality = 100)
  plot<-plot(a[[1]][,1],a[[1]][,2],type="l", ylim = c(-0.02, 0.02), col="2", main=titulo, xlab="Posición (nt)", ylab="Frecuencia
(PQSs/Secuencia)", sub=a[[3]])
```

```
  lines(a[[2]][,1],a[[2]][,2], col="4", lwd=1)
  lines(b[[1]][,1],b[[1]][,2], col="1", lty= 3)
  lines(b[[2]][,1],b[[2]][,2], col="1", lty= 3)
  v<-abline(v=0)
  h<-abline(h=0)
  dev.off()
}
```

Anexo IIIb: Funciones desarrolladas

fill.gaps(): Función para establecer todos los tipos de secuencias (génicas e intergénicas)

```
#' Funcion para establecer todos los tipos de secuencias (genicas e intergenicas)
#' @param seq objeto del tipo GRange, conteniendo los rangos a ser procesados.
#' @keywords cats
#' @return objeto GRange con todos los tipos de secuencias determinadas y varias columnas de metadatos.
#' @export
#' @examples
#' fill.gaps()
fill.gaps<-function(seq=NULL){
  #Fusiono secuencias solapadas
  seq<-overlap.resolve(seq)  #Obtengo un objeto grange con las clases de secuencias
  clas<-poliRNA(seq)
  #relleno los gaps
  w<-seq
  strand(w)<-"*"
  gaps<-gaps(w)
  gaps<-gaps[strand(gaps)=="*"]
  #agrego metadatos para poder fusionar los gaps con as secuencias originales
  mcols(gaps)$type<-"intergenic"
  mcols(gaps)$subtype<-"intergenic"
  mcols(gaps)$solapadas<-0
  #fusiono gaps y secuencias
  x<-c(w,gaps)
  x<-sort(x, ignore.strand=T)
  #Busco los pares de solapamientos
  y<-findOverlaps(x,clas)
  #utilizo las posiciones de cada clase para extraer su clasID
  clas_names<-names(clas[y@to])
  #asigno datos de clase a cada secuencia secuencias
  mcols(x)$ClassID<-clas_names
  mcols(x)$Class<-mcols(clas[y@to])$Class
  mcols(x)$Subclass<-mcols(clas[y@to])$Subclass
  strand(x)<-strand(clas[y@to])
  return(x)
}
```

GC.PQS.asign2(): Función para asignar GC y densidad de PQSs a cada secuencia

```
#' Función para asignar GC y densidad de PQSs a cada secuencia
#' @param secuencias1 Objeto GRanges conteniendo las posiciones de secuencias a ser procesadas
#' @param genoma1 Objeto DNAStringSet conteniendo las secuencias genómicas a utilizar
#' @param cuadruplex1 Objeto GRanges conteniendo los cuadrulex a utilizar
#' @keywords
#' @return Objeto GRange con todos los tipos de secuencias determinadas y varias columnas de metadatos.
#' @export
#' @examples
#' GC.PQS.asign2()

GC.PQS.asign2<-function(secuencias1=NULL,genoma1=NULL,cuadruplex1=NULL){
  #Controles
  #####
```

```
  #Recorto las secuencias genómicas
  #####
  salida<-GRanges()#Creo el objeto GRange que será devuelto por la función
  if (!is.null(genoma1)){#Agrego las columnas con datos de bases y longitudes
  for (i in 1:length(genoma1)){#Bucle para trabajar de a un cromosoma e ir concatenando los resultados
    chrname<-strsplit(names(genoma1[i]), " | ")[[1]][1]#limpio el nombre de cada cromosoma
    seq<-paste(genoma1[i], collapse = "")#Obtengo la secuencia(bases) en formato string
    cromosoma<-secuencias1[seqnames(secuencias1)==chrname]#Selecciono las secuencias(rangos) correspondientes a 1
cromosoma
    if(length(cromosoma)>0){#Aseguro que se haya elegido al menos una secuencia
      #Creo vectores con el inicio y fin de cada secuencia
      V1<-start(cromosoma)
      V2<-end(cromosoma)
      #Utilizo los vectores para recortar la secuencia cromosómica
      subsecuencias<-substring(seq,V1,V2)
      #Obtengo la longitud total de cada subsecuencia
      total<-width(subsecuencias)
      #Obtengo vectores con la cantidad de cada tipo de letra en cada subsecuencia
      Gs<-width(gsub("[^G]", "", subsecuencias))
      Cs<-width(gsub("[^C]", "", subsecuencias))
      As<-width(gsub("[^A]", "", subsecuencias))
      Ts<-width(gsub("[^T]", "", subsecuencias))
      Ns<-width(gsub("[^N]", "", subsecuencias))
      #Obtengo la longitud total corregida de cada subsecuencia
      #total.corregido<-total-Ns#total corregido
      #total.corregido[total.corregido==0]<-1#Para evitar indeterminaciones del tipo 0/0
      #Agrego los metadatos al cromosoma
      mcols(cromosoma)$CG<-Cs+Gs
      mcols(cromosoma)$AT<-As+Ts
      mcols(cromosoma)$N<-Ns
      #Agrego valores de G y C discriminados por hebra (solo para secuencias con polaridad)
      mcols(cromosoma)$GinStrand<-Gs
      mcols(cromosoma)$AinStrand<-As
      mcols(cromosoma)[strand(cromosoma)=="-",]$GinStrand<-mcols(cromosoma)[strand(cromosoma)=="-",]$CG-
mcols(cromosoma)[strand(cromosoma)=="-",]$GinStrand
      mcols(cromosoma)[strand(cromosoma)=="-",]$AinStrand<-mcols(cromosoma)[strand(cromosoma)=="-",]$AT-
mcols(cromosoma)[strand(cromosoma)=="-",]$AinStrand
      #Agrego longitud efectiva (longitud real menos bases indeterminadas)
      mcols(cromosoma)$LongEf<-total-Ns#longitud total corregida de cada subsecuencia
      #Concateno los cromosomas en un solo objeto
      salida<-c(salida,cromosoma)
      print(i)
    }else if(length(cromosoma)==0){
      print(paste0("secuencia no considerada: ",chrname ))
    }
  }
  }else{salida<-secuencias1}
  if (!is.null(cuadruplex1)){#Agrego las columnas con cuadruplex
    #realizo el recuento de cuadruplex por secuencia (sin distincion de hebra)
    w<-findOverlaps(cuadruplex1,salida, ignore.strand=T)#identifico los pares de secuencias solapadas
    x<-as.factor(w@to)#identifico las secuencias de salida que poseen cuadruplex
    y<-table(x)#Identifico cuantas veces figura mi secuencia de salida (=nº cuadruplex que posee)
    g4number<-as.data.frame(y)#Creo tabla en la que figure posicion secuencia, nº cuadruplex
    total.secuencias<-rep(0,length(salida))#Creo un vector de ceros
    total.secuencias[as.numeric(as.character(g4number[,1]))]<-g4number[,2]#Relleno el vector utilizando las posiciones de las
secuencias
    mcols(salida)$g4number<-total.secuencias#asigno el número de cuadruplex a una nueva columna
    #Calculo la densidad
    densidad.ds<-salida$g4number*500/(width(salida)-salida$N)
    mcols(salida)$`g4/Kb`<-densidad.ds
    #realizo el recuento de cuadruplex por secuencia (con distincion de hebra)
    w<-findOverlaps(cuadruplex1,salida, ignore.strand=F)
    x<-as.factor(w@to)
    y<-table(x)
```

```
    g4number<-as.data.frame(y)
    total.secuencias<-rep(0,length(salida))
    total.secuencias[as.numeric(as.character(g4number[,1]))]<-g4number[,2]
    mcols(salida)$g4inStrand<-total.secuencias
    #Calculo la densidad
    densidad.ss<-salida$g4inStrand*1000/(width(salida)-salida$N)
    mcols(salida)$`g4inStrand/Kb`<-densidad.ss
  }
  return(sort(salida, ignore.strand=T))
}
```

GFF.read(): Función para abrir archivos de anotaciones de Tritryp

```
#' Funcion para abrir archivos de anotaciones de Tritryp
#' @param con Nombre del archivo GFF con anotaciones.
#' @keywords
#' @return objeto GRange con las posiciones de las secuencias génicas
#' @export
#' @examples
#' GFF.read()
GFF.read<-function(con=NULL){
  a<-import.gff(con)#problemas al trabajar off-line: Quito el enlace en el archivo original.
  #Leo posiciones de genes
  gene<-subset(a, a$type=="gene")
  #Asigno los IDs a cada gen
  names(gene)<-mcols(gene)[5][,1]
  #Leo tamaño de cromosomas
  b<-read.delim(con, header=F, comment.char="")
  b<-b[grep("##sequence-region",b[,1]),1]
  c<-strsplit(as.character(b), split=" ")
  c<-t(as.data.frame(c))
  c<-as.data.frame(c[,2:4], row.names = F)
  c[,2]<-as.numeric(as.character(c[,2]))
  c[,3]<-as.numeric(as.character(c[,3]))
  chromosomes<-GRanges(c[,1],ranges = IRanges(c[,2],c[,3]))
  seqlengths(gene)<-end(chromosomes)
  #borro los metadatos
  mcols(gene)$type<-"genic"
  subtipos<-c("mRNA","ncRNA","rRNA","tRNA","snRNA","snoRNA", "SLRNA")
  mcols(gene)$subtype<-0
  for (i in 1:length(subtipos)){
    mcols(gene)$subtype[grepl(subtipos[i],as.vector(mcols(gene)$Alias))]<-subtipos[i]
  }
  gene2<-gene
  mcols(gene)<-NULL
  mcols(gene)$type<-as.factor(mcols(gene2)$type)
  mcols(gene)$subtype<-as.factor(mcols(gene2)$subtype)
  return(gene)
}
```

poliRNA(): Función para obtener las distintas clases de secuencias (PTUs, SSRs, tel)

```
#' Funcion para obtener las distintas clases de secuencias (ARN policistrónicos, SSRs, tel)
#' @param GR Objeto GRanges conteniendo las posiciones de secuencias a ser procesadas
#' @keywords
#' @return Objeto GRange con las clases de secuencias determinadas y varias columnas de metadatos.
#' @export
#' @examples
#' poliRNA()

poliRNA<-function(GR=NULL){
```

```
 #resuelvo los solapamientos
 GR<-overlap.resolve(GR)
 #Ordeno por posición para cada cromosoma
 gr<-sort(GR, ignore.strand=T)
 #Obtengo las posiciones relativas (dentro de gr) de cada ARN poicistrónico
 for (i in 1:length(gr@seqnames@values)){
  poliRNA<-data.frame()
  chr<-gr[seqnames(gr)==levels(seqnames(gr))[i]]
  str<-strand(chr)
  str<-data.frame((chr@seqnames@values),str@values,str@lengths)
  poliRNA<-rbind(poliRNA,str)
  ###
  poliRNA[,4:5]<-c(cumsum(poliRNA[,3])-poliRNA[,3]+1,cumsum(poliRNA[,3]))
  #Creo un nuevo objeto GRange con las los datos de los ARNpolicistrónicos predichos
  if (i==1){poli_rna<-GRanges(poliRNA[,1],
                        ranges = IRanges(start(chr)[poliRNA[,4]], end(chr)[poliRNA[,5]]),
                        strand = poliRNA[,2]
  )
  mcols(poli_rna)$seqnumber<-poliRNA[,3]#Agrego el numero de secuencias por cada ARNpolicistronico
  }else{
   pr<-GRanges(poliRNA[,1],
            ranges = IRanges(start(chr)[poliRNA[,4]], end(chr)[poliRNA[,5]]),
            strand = poliRNA[,2]
   )
   mcols(pr)$seqnumber<-poliRNA[,3]#Agrego el numero de secuencias por cada ARNpolicistronico
   poli_rna<-c(poli_rna,pr)
  }
 }
 #Busco los gaps que separan cada ARN polcistrónico
 mcols(poli_rna)$class<-"poliRNA"
 seqlengths(poli_rna)<-seqlengths(gr)
 no_poli_rna<-poli_rna
 strand(no_poli_rna)<-"*"
 no_poli_rna<-gaps(no_poli_rna)
 no_poli_rna<-no_poli_rna[strand(no_poli_rna)=="*"]
 mcols(no_poli_rna)$seqnumber<-0
 mcols(no_poli_rna)$class<-"gap"
 #Creo un objeto GRange con los datos de ARN policistrónicos mas los gaps
 Classes<-c(poli_rna,no_poli_rna)
 Classes<-sort(Classes, ignore.strand=T)
 #Agrego metadatos (columnas de clase, subclase)
 z<-data.frame()
 Classes<-relleno(Classes)[-1]
 classes2<-Classes[0]
 for (i in 1:length(Classes@seqinfo@seqnames)){
  chr<-Classes[seqnames(Classes)==levels(seqnames(Classes))[i]]
  l<-length(chr)
  #Corrección porque en un cromosoma el primer transcrito inicia en la posición 1 (Tbgambiensis972, chr 8)
  if((strand(chr[1])=="*")@values){
   init<-2
   ####
   if((strand(chr[init])=="-")@values){
    #x<-data.frame(Class=rep(c("term","poliRNA","init", "poliRNA"),length.out=l),Subclass=c("tel",rep(c("poliRNA","SSR"),
length.out=l-2),"tel"))
    mcols(chr[1])$class<-"term"
   }else if ((strand(chr[init])=="+")@values){
    #x<-data.frame(Class=rep(c("init","poliRNA","term", "poliRNA"),length.out=l),Subclass=c("tel",rep(c("poliRNA","SSR"),
length.out=l-2),"tel"))
    mcols(chr[1])$class<-"init"
   }
   ###
  }#else{init<-1
  ###
  #if((strand(chr[init])=="-")@values){
```

```
   #x<-data.frame(Class=rep(c("poliRNA","init",
"poliRNA","term"),length.out=l),Subclass=c("poliRNA",rep(c("SSR","poliRNA"), length.out=l-2),"tel"))
   #}else if ((strand(chr[init])=="+")@values){
   #x<-data.frame(Class=rep(c("poliRNA","term",
"poliRNA","init"),length.out=l),Subclass=c("poliRNA",rep(c("SSR","poliRNA"), length.out=l-2),"tel"))
   #}
   ###
   #}
   # if (x[l,1]=="poliRNA"){
   #   x[l,2]=x[l,1]
   # }
   #mcols(chr)<-x
   classes2<-c(classes2,chr)
  }
  #classes2[classes2$Subclass=="tel"]
  classes2[classes2$class=="-"]$class<-"init"
  classes2[classes2$class=="+"]$class<-"term"
  names(mcols(classes2))[2]<-"Class"
  #mcols(classes2)<-mcols(Classes)$seqnumber
  mcols(classes2)[1:3]<-mcols(classes2)[c(2,3,1)]
  names(mcols(classes2))[1:3]<-names(mcols(classes2))[c(2,3,1)]
  #Asigno nombre a las clases
  l<-length(classes2)
  names(classes2)<-paste0("ClassID:",seq(1:l))
  #Devuelvo el resultado
  return(classes2)
}
```

relleno() : Función para identificar en forma correcta los SSR y tel de terminación e iniciales

```
#' Funcion para identificar en forma correcta los SSR y tel de terminación e iniciales
#' @param seq objeto GRange con los datos de ARN policistrónicos mas los gaps
#' @keywords
#' @return objeto GRange con los datos de ARN policistrónicos mas los SSRs y Tel caraterizados
#' @export
#' @examples
#' relleno()

relleno<-function(seq=Classes){
  telomerosT<-seq
  salida<-seq[1]
  for (i in 1:length(seq@seqinfo@seqnames)){
    fin<-seqlengths(seq)[i]
    cromosoma<-seq[seqnames(seq)==levels(seqnames(seq))[i]]
    if(length(cromosoma[cromosoma$class=="gap"&end(cromosoma)==fin]$class)==1){
      cromosoma[cromosoma$class=="gap"&end(cromosoma)==fin]$class<-"tel"
    }
    salida<-c(salida,cromosoma)
  }
  telomerosT<-salida
  #Identifico y reemplazo el telomero inicial
  if (length(telomerosT[telomerosT$class=="gap"&start(telomerosT)==1]$class)>=1){
    telomerosT[telomerosT$class=="gap"&start(telomerosT)==1]$class<-"tel"
  }
  #SSRs
  if (length(telomerosT[telomerosT$class=="gap"]$class)>=1){
    telomerosT[telomerosT$class=="gap"]$class<-"SSR"
  }
  #Agrego una columna
  mcols(telomerosT)$Subclass<-mcols(telomerosT)$class
  x<-as.character(strand(telomerosT))
  y<-as.character(telomerosT$class)
  z<-as.data.frame(cbind(x,y))
```

```
  z[,3]<-"A"
  z[2:(nrow(z)),3]<-as.character(z[1:(nrow(z)-1),1])
  z[1:(nrow(z)-1),4]<-as.character(z[2:(nrow(z)),1])
  z[z[,2]=="SSR"&z[,3]=="-",3]<-"init"
  z[z[,2]=="SSR"&z[,3]=="+",3]<-"term"
  z[z[,2]=="poliRNA",3]<-"poliRNA"
  mcols(telomerosT)[,2]<-z[,3]
  return(telomerosT)
}
```

overlap.resolve() : Función de corrección para solapamientos de genes

```
#' Función de corrección para solapamientos de genes:
#' @param object objeto GRange con los datos de ARN policistrónicos mas los gaps
#' @keywords
#' @return objeto GRange sin secuencias solapadas
#' @export
#' @examples
#' overlap.resolve()

overlap.resolve<-function(object=NULL){
  object<-sort(object, ignore.strand=T)
  #Fusiono las secuencias solapadas en la misma hebra
  #Uso la función propia "solapar", para conservar los nombres de las secuencias
  Tb2<-solapar(object)
  #Reasigno los nombres a las secuencias
  a<-reduce(object, drop.empty.ranges=FALSE, min.gapwidth=0L, with.revmap=FALSE,
            with.inframe.attrib=FALSE, ignore.strand=T)
  if (Tb2@elementMetadata@nrows==a@elementMetadata@nrows){
    return(Tb2)
  }else if(Tb2@elementMetadata@nrows!=a@elementMetadata@nrows){
    #Separo las secuencias por hebra
    minus_strand<-Tb2[strand(Tb2)=="-",]
    plus_strand<-Tb2[strand(Tb2)=="+",]
    #Busco los solapamientos entre hebras
    overlaps<-findOverlaps(minus_strand,plus_strand, ignore.strand=T)
    #Extraigo las secuencias solapadas
    overlaping_minus<-minus_strand[overlaps@from]
    overlaping_plus<-plus_strand[overlaps@to]
    #Fusiono las secuencias solapadas y anulo la hebra
    overlaping<-c(overlaping_plus,overlaping_minus)
    strand(overlaping)<-"*"
    overlaping<-solapar(overlaping)
    #overlaping<-reduce(overlaping,ignore.strand=T)
    #Busco las ubicaciones de las secuencias solapadas en el genoma con ambas hebras
    over<-findOverlaps(overlaping,Tb2)
    #Elimino las secuencias solapantes originales
    Tb3<-Tb2[-over@to]
    #Inserto las secuencias solapantes con la hebra indefinida
    Tb4<-sort(c(Tb3,overlaping),ignore.strand=T)
    #Identifico las posiciones de las secuencias solapantes en el genoma
    x<-strand(Tb4)=="*"
    y<-cbind(cumsum(x@lengths)*x@values)
    y<-as.vector(subset (y, y!=0))
    #Extraigo las secuencias solapantes y sus adyacentes
    izq<-as.character(strand(Tb4[y-1,]))
    cent<-as.character(strand(Tb4[y,]))
    der<-as.character(strand(Tb4[y+1,]))
    #Armo una tabla con las secuencias solapantes y sus adyacentes
    hebras<-cbind(izq,cent,der)
    #Utilizo la tabla para reemplazar el valor de hebra en el genoma
    #Ambos extremos misma hebra:
```

```
   strand(Tb4[y[hebras[,1]==hebras[,3]]])<-hebras[hebras[,1]==hebras[,3],3]
   #Extremos con distinta hebra:
   #Creo un nuevo objeto GRange y lo reemplazo en el genoma
   v<-seqnames(Tb4[y[hebras[,1]!=hebras[,3]]])
   s<-start(Tb4[y[hebras[,1]!=hebras[,3]]])
   e<-end((Tb4[y[hebras[,1]!=hebras[,3]]]))
   n<-names((Tb4[y[hebras[,1]!=hebras[,3]]]))
   mc<-mcols((Tb4[y[hebras[,1]!=hebras[,3]]]))
   w<-round(width(Tb4[y[hebras[,1]!=hebras[,3]]])/2)
   dif<-width(Tb4[y[hebras[,1]!=hebras[,3]]])-2*w
   left<-GRanges(v,
                 ranges = IRanges(s,(s+w-1)),
                 strand= hebras[hebras[,1]!=hebras[,3],1])
   rigth<-GRanges(v,
                  ranges = IRanges((e-w+1-dif),e),
                  strand= hebras[hebras[,1]!=hebras[,3],3])
   if (length(left)>0){
     names(left)<-paste0(n,"L")
     mcols(left)<-mc
     names(rigth)<-paste0(n,"R")
     mcols(rigth)<-mc
     #Borro las hebras "*"
     Tb4<-Tb4[-y[hebras[,1]!=hebras[,3]]]
   }
   #Inserto las secuencias modificadas
   Tb5<-sort(c(Tb4,left,rigth),ignore.strand=T)
   return(Tb5)
  }
}
```

solapar() : Función para fusionar secuencias solapadas en la misma hebra

```
#' Función para fusionar secuencias solapadas en la misma hebra:
#' @param object objeto GRange con los datos de ARN policistrónicos
#' @keywords
#' @return Se mantiene el nombre de las secuencias y se indica la cantidad de secuencias solapadas en cada rango
#' @export
#' @examples
#' solapar()

#1)Función para fusionar secuencias solapadas en la misma hebra.
solapar<-function(object=NULL){
  if (ncol(mcols(object))==0){#Si no hay columna de metadatos, creo una
    mcols(object)<-1
  }
  mcols(object)$solapadas<-1#Asigno nombre y valor (uno por defecto) a la primer columna de metadatos
  overlaps<-countOverlaps(object, ignore.strand=F, maxgap = 0, minoverlap = 1)
  solapados<-object[overlaps>=2]
  no_solapados<-object[overlaps==1]
  if (length(solapados)>=1){
    fusionados<-reduce(solapados)
    correspondencia<-findOverlaps(solapados,fusionados)
    for (i in 1:length(fusionados)){
      nombres<-names(solapados)[correspondencia@from[correspondencia@to==i]]
      tipos<-mcols(solapados)$type[correspondencia@from[correspondencia@to==i]]
      subtipos<-mcols(solapados)$subtype[correspondencia@from[correspondencia@to==i]]
      presolapadas<-solapados@elementMetadata@listData$solapadas[correspondencia@from[correspondencia@to==i]]
      l<-sum(presolapadas)
      #l<-length(nombres)
      pegados<-paste0(nombres,"/", collapse ="")
      tipo<-paste0(tipos,"/", collapse ="")
      subtipo<-paste0(subtipos,"/", collapse ="")
```

```
    names(fusionados)[i]<-pegados
    mcols(fusionados)$type[i]<-tipo
    mcols(fusionados)$subtype[i]<-subtipo
    mcols(fusionados)$solapadas[i]<-l
   }
   #mcols(fusionados)<-mcols(fusionados)[,1]
   salida<-sort(c(object[overlaps==1],fusionados),ignore.strand=T)
   return(salida)
  }else{
   return(object)
  }
}
```

PQSdistributionV2(): Función para visualizar la distribución de PQSs entre clases de secuencias

```
#' Función para visualizar la distribución de PQSs entre clases de secuencias
#' @param sec Objeto GRanges conteniendo las posiciones de secuencias a ser procesadas
#' @param g4 Objeto GRanges conteniendo los cuadrulex a utilizar
#' @param zero nucleótido cero (start: inicio PTU, end: terminación PTU)
#' @param upstream nucleótidos aguas arriba a ser considerados
#' @param downstream nucleótidos aguas abajo a ser considerados
#' @param exclude vector con el nombre de secuencias a ser excluida en el análisis (clases y/o tipos)
#' @param relative número de PQSs por nucleótido relativo (T) o aboluto (F)
#' @param title titulo del grafico
#' @param xlab leyenda eje x
#' @param ylab leyenda eje y
#' @param max.score máximo score de PQSs a ser considerado
#' @param min.score mínimo score de PQSs a ser considerados
#' @param ss valores independientes para cada hebra (T) o valor conjunto de ambas hebras (F)
#' @param graph graficar los resultados (T) o presentarlos como tabla (F)
#' @keywords
#' @return Gráfico o tabla con el número o densidad de PQSs en el rango seleccionado
#' @export
#' @examples
#' PQSdistributionV2()

PQSdistributionV2<-function(sec=NULL,g4=NULL, zero="end", upstream=1000, downstream=500, exclude=c("tel"),
                    relative=T, title="",xlab="posicion", ylab="frecuencia PQSs",
                    max.score=NULL, min.score=NULL, ss=T, graph=T){
 #Asigno valores de score máximo y minimo para los cuadruplex a utilizar
 if(!is.null(max.score)){
  g4<-g4[g4$score<=max.score]
 }
 if(!is.null(min.score)){
  g4<-g4[g4$score>=min.score]
 }
 #Excluyo la sublcase especificada (tel o SSR)
 sec2<-sec[mcols(sec)$Subclass!=exclude]
 #Armo leyendas para el grafico según subclase excluida
 filtro<-""
 if (exclude=="tel"){
  filtro<-"SSR"
 }else if (exclude=="SSR"){
  filtro<-"Tel"
 }
 #Trabajo sobre transicion Inicio-PoliRNA
 if (zero=="start"){
  #Selecciono las secuencias poliRNA (nucleo) y las adyacentes
  secpoliRNA<-sec2[sec2$Class=="poliRNA"&width(sec2)>=downstream]
  nonpoliRNA<-sec2[(sec2$Subclass=="SSR"&width(sec2)>=(2*upstream))|(sec2$Subclass=="tel"&width(sec2)>=(upstream))]
  #Defino los pares de secuencias a solapar
  #Creo los intervalos "pegajosos"
  clip<-promoters(secpoliRNA, upstream=1, downstream =1)
```

```
  #pego con las secuencias no-poliRNA
  selector<-findOverlaps(clip,nonpoliRNA, ignore.strand=T)
  #Seleccino unicamente los poliRNA que han pegado en alguna secuencia
  secpoliRNA<-secpoliRNA[selector@from]
  #Defino las ventanas
  ventanas<-promoters(secpoliRNA, upstream=upstream, downstream = downstream)
  #selector<-findOverlaps(ventanas,nonpoliRNA, ignore.strand=T)
  #if(length(selector)!=0){
  print(paste0(length(selector), " secuencias seleccionadas."))
  #ventanas<-ventanas[selector@from]
  #}else if(length(selector)==0){
  if(length(selector)==0){
    return("Ninguna secuencia seleccionada. Intente con una ventana mas pequenia")
  }
  #}
  #Pego los PQSs
  selector2<-findOverlaps(ventanas,g4, ignore.strand=T)
  PQSs<-g4[selector2@to]
  #Calculo las posiciones relativas de los PQSs dentro de cada secuencia
  PQSspos<-PQSs[strand(PQSs)=="+"]
  PQSsneg<-PQSs[strand(PQSs)=="-"]
  ventanaspos<-ventanas[strand(ventanas)=="+"]
  ventanasneg<-ventanas[strand(ventanas)=="-"]
  selector3<-findOverlaps(ventanaspos,PQSspos, ignore.strand=T)
  selector4<-findOverlaps(ventanaspos,PQSsneg, ignore.strand=T)
  selector5<-findOverlaps(ventanasneg,PQSspos, ignore.strand=T)
  selector6<-findOverlaps(ventanasneg,PQSsneg, ignore.strand=T)
  selector7<-findOverlaps(ventanaspos,PQSs, ignore.strand=T)
  selector8<-findOverlaps(ventanasneg,PQSs, ignore.strand=T)
  start3<-start(PQSspos[selector3@to])-start(ventanaspos[selector3@from])#coding
  end3<-end(PQSspos[selector3@to])-start(ventanaspos[selector3@from])#coding
  start4<-start(PQSsneg[selector4@to])-start(ventanaspos[selector4@from])#template
  end4<-end(PQSsneg[selector4@to])-start(ventanaspos[selector4@from])#template
  start5<-(end(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  end5<-(start(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  start6<-(end(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  end6<-(start(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  start7<-start(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  end7<-end(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  start8<-(end(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
  end8<-(start(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
  leyenda<-paste0("Inicio",filtro,"-PTU (", length(selector), " secuencias consideradas.)" )
 }
 else if (zero=="end"){
  secpoliRNA<-sec2[sec2$Class=="poliRNA"&width(sec2)>=upstream]
  nonpoliRNA<-sec2[(sec2$Subclass=="SSR"&width(sec2)>=(2*downstream))|
(sec2$Subclass=="tel"&width(sec2)>=(downstream))]
  #Defino las ventanas
  #Debo invertir las hebras antes de aplicar la funcion "promoters"
  a<-secpoliRNA[strand(secpoliRNA)=="+"]
  b<-secpoliRNA[strand(secpoliRNA)=="-"]
  strand(a)<-"-"
  strand(b)<-"+"
  secpoliRNAinv<-c(a,b)
  #Defino los pares de secuencias a solapar
  #Creo los intervalos "pegajosos"
  clip<-promoters(secpoliRNAinv, upstream=1, downstream =1)
  #pego con las secuencias no-poliRNA
  selector<-findOverlaps(clip,nonpoliRNA, ignore.strand=T)
  #Seleccino unicamente los poliRNA que han pegado en alguna secuencia
  secpoliRNAinv<-secpoliRNAinv[selector@from]
  #seqlengths(secpoliRNAinv)<-NA#Quito el valor de seqlength para evitar errores
  ventanas<-promoters(secpoliRNAinv,upstream=downstream, downstream = upstream)#invierto up y down
  #selectorventanas<-promoters(secpoliRNAinv,upstream=upstream, downstream = 1)
```

```
  #Invierto las hebras de las ventanas para restituir las orientaciones originales
  a<-ventanas[strand(ventanas)=="+"]
  b<-ventanas[strand(ventanas)=="-"]
  strand(a)<-"-"
  strand(b)<-"+"
  ventanas<-c(a,b)
  #selector<-findOverlaps(selectorventanas,nonpoliRNA, ignore.strand=T)
  #if(length(selector)!=0){
  print(paste0(length(selector), " secuencias seleccionadas."))
  #ventanas<-sort(ventanas[selector@from],ignore.strand=T)
  #seqlengths(ventanas)<-seqlengths(sec)#Reasigno el valor de seqlength
  if(length(selector)==0){
    return("Ninguna secuencia seleccionada. Intente con una ventana m?s peque?a")
  }
  #}
  #Pego los PQSs
  selector2<-findOverlaps(ventanas,g4, ignore.strand=T)
  PQSs<-g4[selector2@to]
  #Calculo las posiciones relativas de los PQSs dentro de cada secuencia y hebra
  PQSspos<-PQSs[strand(PQSs)=="+"]
  PQSsneg<-PQSs[strand(PQSs)=="-"]
  ventanaspos<-ventanas[strand(ventanas)=="+"]
  ventanasneg<-ventanas[strand(ventanas)=="-"]
  selector3<-findOverlaps(ventanaspos,PQSspos, ignore.strand=T)
  selector4<-findOverlaps(ventanaspos,PQSsneg, ignore.strand=T)
  selector5<-findOverlaps(ventanasneg,PQSspos, ignore.strand=T)
  selector6<-findOverlaps(ventanasneg,PQSsneg, ignore.strand=T)
  selector7<-findOverlaps(ventanaspos,PQSs, ignore.strand=T)
  selector8<-findOverlaps(ventanasneg,PQSs, ignore.strand=T)
  #Distancias de los extremos del PQS al inicio de su ventana (ventana positiva)
  start3<-(start(PQSspos[selector3@to])-start(ventanaspos[selector3@from]))#coding
  end3<-(end(PQSspos[selector3@to])-start(ventanaspos[selector3@from]))#coding
  start4<-(start(PQSsneg[selector4@to])-start(ventanaspos[selector4@from]))#template
  end4<-(end(PQSsneg[selector4@to])-start(ventanaspos[selector4@from]))#template
  #Distancias de los extremos del PQS al inicio de su ventana (ventana negativa)
  start5<-(end(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  end5<-(start(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  start6<-(end(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  end6<-(start(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  #Distancias de los extremos del PQS al inicio de su ventana (ventanas positiva/negativa)
  start7<-start(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  end7<-end(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  start8<-(end(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
  end8<-(start(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1

  leyenda<-paste0("PTU-Terminacion",filtro," (", length(selector), " secuencias consideradas.)" )
}

if (ss==T){
  #Fusiono los rangos de PQSs para cada ventana (pos/neg) separados por hebra (coding/templ)
  coding<-data.frame(cbind(c(start3,start6),c(end3,end6)))
  template<-data.frame(cbind(c(start4,start5),c(end4,end5)))
  coding[,3]<-"coding"
  template[,3]<-"template"
  # tabla<-rbind(coding,template)
  #Armo una tabla con las densidades de PQSs para cada posición
  #Armo dos tablas (una por hebra) con las posiciones
  codingdensity<-as.data.frame(seq(1:(upstream+downstream)))
  templatedensity<-as.data.frame(seq(1:(upstream+downstream)))
  #Asigno a cada posición su densidad de PQSs
  for (i in 1:(upstream+downstream)){
    codingdensity[i,2]<-sum(coding[,1]<=i & coding[,2]>=i)
    templatedensity[i,2]<-(sum(template[,1]<=i & template[,2]>=i))*-1#Cambio e signo con fines gráficos
  }
```

```
  #Seteo como 0 el punto de transición (Inicio-PoliRNA o PoliRNA-Term)
  codingdensity[,1]<-codingdensity[,1]-upstream
  templatedensity[,1]<- templatedensity[,1]-upstream
  if (relative==T){#Convierto as frecuencias absolutas de PQSs a relativas (PQSs/secuencias)
   codingdensity[,2]<-(codingdensity[,2]/length(selector))
   templatedensity[,2]<-(templatedensity[,2]/length(selector))
  }
  if (graph==T){
   #Grafico!
   plot<-plot(codingdensity[,1],codingdensity[,2],type="l", ylim = c(min(templatedensity[,2]), max(codingdensity[,2])), col="red",
         main=title, xlab=xlab, ylab=ylab, sub=leyenda)
   lines<-lines(templatedensity[,1],templatedensity[,2], col="blue")
   v<-abline(v=0)
   h<-abline(h=0)
   return (c(plot,lines,v,h))
  } else (return(list(codingdensity,templatedensity,leyenda)))
 }else if(ss==F){
  #Fusiono
  coding<-data.frame(cbind(c(start7,start8),c(end7,end8)))
  codingdensity<-as.data.frame(seq(1:(upstream+downstream)))
  for (i in 1:(upstream+downstream)){
   codingdensity[i,2]<-sum(coding[,1]<=i & coding[,2]>=i)
  }
  codingdensity[,1]<-codingdensity[,1]-upstream
  if (relative==T){
   codingdensity[,2]<-(codingdensity[,2]/length(selector))
  }
  if (graph==T){
  plot<-plot(codingdensity[,1],codingdensity[,2],type="l", ylim = c(0, max(codingdensity[,2])), col="brown",
        main=title, xlab=xlab, ylab=ylab, sub=leyenda)
  v<-abline(v=0)
  return (c(plot,v))
  }else (return(list(codingdensity,templatedensity,leyenda)))
 }

}
```

PQSgenicdistributionV2(): Función para visualizar la distribución de PQSs entre tipos de secuencias

```
#' Función para visualizar la distribución de PQSs entre tipos de secuencias
#' @param sec Objeto GRanges conteniendo las posiciones de secuencias a ser procesadas
#' @param g4 Objeto GRanges conteniendo los cuadrulex a utilizar
#' @param zero nucleótido cero (start: inicio PTU, end: terminación PTU)
#' @param upstream nucleótidos aguas arriba a ser considerados
#' @param downstream nucleótidos aguas abajo a ser considerados
#' @param exclude vector con el nombre de secuencias a ser excluida en el análisis (clases y/o tipos)
#' @param relative número de PQSs por nucleótido relativo (T) o aboluto (F)
#' @param title titulo del grafico
#' @param xlab leyenda eje x
#' @param ylab leyenda eje y
#' @param max.score máximo score de PQSs a ser considerado
#' @param min.score mínimo score de PQSs a ser considerados
#' @param ss valores independientes para cada hebra (T) o valor conjunto de ambas hebras (F)
#' @param graph graficar los resultados (T) o presentarlos como tabla (F)
#' @keywords
#' @return Gráfico o tabla con el número o densidad de PQSs en el rango seleccionado
#' @export
#' @examples
#' PQSgenicdistributionV2()

PQSgenicdistributionV2<-function(sec=completa,g4=NULL, zero="start", upstream=1000, downstream=500,
```

```
exclude=c("SLRNA"),
                        relative=F, title="Distribucion PQSs intra-poliRNA",xlab="posicion", ylab="frecuencia PQSs",
                        max.score=NULL, min.score=NULL, ss=T, graph=T){
 if(!is.null(max.score)){
  g4<-g4[g4$score<=max.score]
 }
 if(!is.null(min.score)){
  g4<-g4[g4$score>=min.score]
 }
 #Excluyo las subclases especificadas
 sec2<-sec[sec$Class=="poliRNA"]
 for (i in 1:length(exclude)){
  sec2<-sec2[mcols(sec2)$subtype!=exclude[i]]
 }
 if (zero=="start"){
  #Selecciono las secuencias genica (nucleo) y las adyacentes
  secGenic<-sec2[sec2$type=="genic"&width(sec2)>=downstream]
  nonGenic<-sec2[(sec2$type=="intergenic"&width(sec2)>=(2*upstream))]
  #Defino los pares de secuencias a solapar
  #Creo los intervalos "pegajosos"
  clip<-promoters(secGenic, upstream=1, downstream =1)
  #pego con las secuencias no-genic
  selector<-findOverlaps(clip,nonGenic, ignore.strand=T)
  #Seleccino unicamente los poliRNA que han pegado en alguna secuencia
  secGenic<-secGenic[selector@from]
  #Defino las ventanas
  ventanas<-promoters(secGenic, upstream=upstream, downstream = downstream)
  #selector<-findOverlaps(ventanas,nonGenic, ignore.strand=T)
  #if(length(selector)!=0){
  print(paste0(length(selector), " secuencias seleccionadas."))
  #ventanas<-ventanas[selector@from]
  #}else if(length(selector)==0){
  if(length(selector)==0){
   return("Ninguna secuencia seleccionada. Intente con una ventana mas pequenia")
  }
  #}
  #Pego los PQSs
  selector2<-findOverlaps(ventanas,g4, ignore.strand=T)
  PQSs<-g4[selector2@to]
  #Calculo las posiciones relativas de los PQSs dentro de cada secuencia
  PQSspos<-PQSs[strand(PQSs)=="+"]
  PQSsneg<-PQSs[strand(PQSs)=="-"]
  ventanaspos<-ventanas[strand(ventanas)=="+"]
  ventanasneg<-ventanas[strand(ventanas)=="-"]
  selector3<-findOverlaps(ventanaspos,PQSspos, ignore.strand=T)
  selector4<-findOverlaps(ventanaspos,PQSsneg, ignore.strand=T)
  selector5<-findOverlaps(ventanasneg,PQSspos, ignore.strand=T)
  selector6<-findOverlaps(ventanasneg,PQSsneg, ignore.strand=T)

  selector7<-findOverlaps(ventanaspos,PQSs, ignore.strand=T)
  selector8<-findOverlaps(ventanasneg,PQSs, ignore.strand=T)

  start3<-start(PQSspos[selector3@to])-start(ventanaspos[selector3@from])#coding
  end3<-end(PQSspos[selector3@to])-start(ventanaspos[selector3@from])#coding
  start4<-start(PQSsneg[selector4@to])-start(ventanaspos[selector4@from])#template
  end4<-end(PQSsneg[selector4@to])-start(ventanaspos[selector4@from])#template

  start5<-(end(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  end5<-(start(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  start6<-(end(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  end6<-(start(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  start7<-start(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  end7<-end(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  start8<-(end(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
```

```
  end8<-(start(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
  leyenda<-paste0("Intergénica-Génica (", length(selector), " secuencias consideradas.)" )
 }
 else if (zero=="end"){
  #Selecciono las secuencias genica (nucleo) y las adyacentes
  secGenic<-sec2[sec2$type=="genic"&width(sec2)>=upstream]
  nonGenic<-sec2[(sec2$type=="intergenic"&width(sec2)>=(2*downstream))]
  #Defino las ventanas
  #Debo invertir las hebras antes de aplicar la funcion "promoters"
  a<-secGenic[strand(secGenic)=="+"]
  b<-secGenic[strand(secGenic)=="-"]
  strand(a)<-"-"
  strand(b)<-"+"
  secGenicinv<-c(a,b)
  #Defino los pares de secuencias a solapar
  #Creo los intervalos "pegajosos"
  clip<-promoters(secGenicinv, upstream=1, downstream =1)
  #pego con las secuencias no-poliRNA
  selector<-findOverlaps(clip,nonGenic, ignore.strand=T)
  #Seleccino unicamente los poliRNA que han pegado en alguna secuencia
  secGenicinv<-secGenicinv[selector@from]
  #seqlengths(secGenicinv)<-NA#Quito el valor de seqlength para evitar errores
  ventanas<-promoters(secGenicinv,upstream=downstream, downstream = upstream)#invierto up y down
  #selectorventanas<-promoters(secGenicinv,upstream=upstream, downstream = 1)
  #Invierto las hebras de las ventanas para restituir las orientaciones originales
  a<-ventanas[strand(ventanas)=="+"]
  b<-ventanas[strand(ventanas)=="-"]
  strand(a)<-"-"
  strand(b)<-"+"
  ventanas<-c(a,b)
  #selector<-findOverlaps(selectorventanas,nonGenic, ignore.strand=T)
  #if(length(selector)!=0){
  print(paste0(length(selector), " secuencias seleccionadas."))
  #ventanas<-sort(ventanas[selector@from],ignore.strand=T)
  #seqlengths(ventanas)<-seqlengths(sec)#Reasigno el valor de seqlength
  if(length(selector)==0){
   return("Ninguna secuencia seleccionada. Intente con una ventana mas pequenia")
  }
  #}
  #Pego los PQSs
  selector2<-findOverlaps(ventanas,g4, ignore.strand=T)
  PQSs<-g4[selector2@to]
  #Calculo las posiciones relativas de los PQSs dentro de cada secuencia
  PQSspos<-PQSs[strand(PQSs)=="+"]
  PQSsneg<-PQSs[strand(PQSs)=="-"]
  ventanaspos<-ventanas[strand(ventanas)=="+"]
  ventanasneg<-ventanas[strand(ventanas)=="-"]
  selector3<-findOverlaps(ventanaspos,PQSspos, ignore.strand=T)
  selector4<-findOverlaps(ventanaspos,PQSsneg, ignore.strand=T)
  selector5<-findOverlaps(ventanasneg,PQSspos, ignore.strand=T)
  selector6<-findOverlaps(ventanasneg,PQSsneg, ignore.strand=T)
  selector7<-findOverlaps(ventanaspos,PQSs, ignore.strand=T)
  selector8<-findOverlaps(ventanasneg,PQSs, ignore.strand=T)
  start3<-(start(PQSspos[selector3@to])-start(ventanaspos[selector3@from]))#coding
  end3<-(end(PQSspos[selector3@to])-start(ventanaspos[selector3@from]))#coding
  start4<-(start(PQSsneg[selector4@to])-start(ventanaspos[selector4@from]))#template
  end4<-(end(PQSsneg[selector4@to])-start(ventanaspos[selector4@from]))#template
  start5<-(end(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  end5<-(start(PQSspos[selector5@to])-end(ventanasneg[selector5@from]))*-1#template
  start6<-(end(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  end6<-(start(PQSsneg[selector6@to])-end(ventanasneg[selector6@from]))*-1#coding
  start7<-start(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  end7<-end(PQSs[selector7@to])-start(ventanaspos[selector7@from])
  start8<-(end(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
```

```
  end8<-(start(PQSs[selector8@to])-end(ventanasneg[selector8@from]))*-1
  leyenda<-paste0("Génica-Intergénica (", length(selector), " secuencias consideradas.)" )
 }

 if (ss==T){
  #Fusiono
  coding<-data.frame(cbind(c(start3,start6),c(end3,end6)))
  template<-data.frame(cbind(c(start4,start5),c(end4,end5)))
  coding[,3]<-"coding"
  template[,3]<-"template"
  # tabla<-rbind(coding,template)
  codingdensity<-as.data.frame(seq(1:(upstream+downstream)))
  templatedensity<-as.data.frame(seq(1:(upstream+downstream)))
  for (i in 1:(upstream+downstream)){
   codingdensity[i,2]<-sum(coding[,1]<=i & coding[,2]>=i)
   templatedensity[i,2]<-(sum(template[,1]<=i & template[,2]>=i))*-1
  }
  codingdensity[,1]<-codingdensity[,1]-upstream
  templatedensity[,1]<- templatedensity[,1]-upstream
  if (relative==T){
   codingdensity[,2]<-(codingdensity[,2]/length(selector))
   templatedensity[,2]<-(templatedensity[,2]/length(selector))
  }
  if (graph==T){
   #Grafico!
  plot<-plot(codingdensity[,1],codingdensity[,2],type="l", ylim = c(min(templatedensity[,2]), max(codingdensity[,2])), col="red",
          main=title, xlab=xlab, ylab=ylab, sub=leyenda)
  lines<-lines(templatedensity[,1],templatedensity[,2], col="blue")
  v<-abline(v=0)
  h<-abline(h=0)
  return (c(plot,lines,v,h))
  } else (return(list(codingdensity,templatedensity,leyenda)))
 }else if(ss==F){

  #Fusiono
  coding<-data.frame(cbind(c(start7,start8),c(end7,end8)))

  codingdensity<-as.data.frame(seq(1:(upstream+downstream)))
  for (i in 1:(upstream+downstream)){
   codingdensity[i,2]<-sum(coding[,1]<=i & coding[,2]>=i)
  }
  codingdensity[,1]<-codingdensity[,1]-upstream

  if (relative==T){
   codingdensity[,2]<-(codingdensity[,2]/length(selector))
  }
  if (graph==T){
  plot<-plot(codingdensity[,1],codingdensity[,2],type="l", ylim = c(0, max(codingdensity[,2])), col="brown",
          main=title, xlab=xlab, ylab=ylab, sub=leyenda)
  v<-abline(v=0)
  return (c(plot,v))
  }else (return(list(codingdensity,templatedensity,leyenda)))
 }
}
```

Seqmix2(): Función para armar secuencias de longitud y composición de bases definida

```
#' Función para armar secuencias de longitud y composición de bases definida
#' @param seq Secuencia preexistente donde se insertan las secuencias generadas (no habilitado)
#' @param A numero de Adeninas a incorporar
#' @param C numero de Citocinas a incorporar
#' @param G numero de Guaninas a incorporar
#' @param N numero de nucleótidos sin determinar a incorporar
```

```
#' @param long longitud total de la secuencia a obtener
#' @param distribution tipo de inserción de las secuencias generadas en la secuencia preexistente (no habilitado)
#' @keywords
#' @return Secuencia en formato string
#' @export
#' @examples
#' seqmix2()

seqmix2<-function(seqs=NULL, A=NULL, C=NULL, G=NULL, N=NULL, long=100 ,distribution="random"){
  #Genero la secuencia al azar
  Ts<-long-(A+C+G+N)
  As<-rep("A",A)
  Cs<-rep("C",C)
  Gs<-rep("G",G)
  Ns<-rep("N",N)
  Ts<-rep("T",Ts)
  NT<-c(As,Ts,Gs,Cs,Ns)
  coord<-sample(long,long)
  NT<-cbind(NT,coord)
  string<-NT[order(NT[,2]),1]
  string<-paste0(string, collapse = "")
  if (!is.null(seqs)){
    #seqs<-c("GGG", "CCC", "TTT")
    x<-length(seqs)
    #Genero los puntos de insercion
    #a) al azar
    if (distribution=="random"){
      coord<-sample(long,x)
    }else if (distribution=="uniform"){
    segmento<-round(long/(x+1))
    coord<-seq(segmento, segmento*x, segmento)
    coord<-sample(coord,x)
    }
    seqs<-cbind(seqs,coord)
    coord<-coord[order(coord)]
    V1<-c(1,coord+1)
    V2<-c(coord,long)
    corte<-substring(string,V1,V2)
    corte<-cbind(corte,seq(1,(x*2)+1, by=2))
    seqs<-seqs[order(seqs[,2])]
    inserts<-cbind(seqs,seq(2,x*2, by=2))
    a<-rbind(corte,inserts)
    b<-a[order(a[,2]),1]
    c<-paste0(b, collapse = "")
    return(c)
  }else(return(string))
}
```