Análisis de la tasa de mutación en secuencias de unión a factores de transcripción en *Escherichia coli*.

Gonzalo Jiménez Foronda
Máster en Bioinformática y Bioestadística
Genómica Comparativa

Ivan Erill Sagales
Carles Ventura Royo
02/01/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis de la tasa de mutación en secuencias de unión a factores de transcripción en Escherichia coli.*
Nombre del autor:	*Gonzalo Jiménez Foronda*
Nombre del consultor/a:	*Ivan Erill Sagales*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	01/2018
Titulación::	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Genómica Comparativa*
Idioma del trabajo:	*Castellano*
Palabras clave	*Secuencias de unión a factores de trascripción, Tasa de mutación, Genómica comparativa.*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Finalidad: Determinar si las tasas de mutación de secuencias de unión a factores de transcripción difieren significativamente de las tasas de mutación de sus regiones colindantes, en distintas cepas de Escherichia coli.

Contexto de aplicación: Este proyecto puede constituir la base para un estudio más amplio sobre mecanismos evolutivos de introducción de variabilidad en las redes transcripcionales bacterianas, con el efecto que esto tiene en nuestros conocimientos sobre la adaptabilidad de las bacterias.

Metodología: Se ha seguido una metodología plenamente bioinformática. Se han alineado secuencias de unión a factores de transcripción, junto con sus secuencias adyacentes, a los genomas completos de las cepas de *E.coli*. Posteriormente, se han filtrado los alineamientos siguiendo criterios de calidad y se ha realizado el conteo de las sustituciones, diferenciando los sitios de unión a factores de transcripción de sus secuencias adyacentes. Por último, se han comparado las tasas de sustitución empleando tests no paramétricos.

Resultados: Los resultados constan de un gráfico descriptivo y un contraste de hipótesis, para cada uno de los casos analizados.

Conclusiones: La conclusión principal obtenida es que, en *E.coli*, las tasas

de mutación son menores en las secuencias de unión a factores de transcripción que en sus secuencias adyacentes, indistintamente de si, al unirse a un determinado factor de transcripción, la secuencia activa o reprime la transcripción. De cara a futuras líneas de investigación sería interesante realizar un análisis más amplio, en el que se incluyesen distintos grupos bacterianos.

Abstract (in English, 250 words or less):

Aim: To determine if the mutation rates of transcription factor binding sequences differ significantly from the mutation rates of their adjacent regions, in different strains of Escherichia coli.

Context: This project can be the startpoint for a broader study on evolutionary mechanisms for the introduction of variability in bacterial transcriptional networks, with the effect that this has on our knowledge about the adaptability of bacteria.

Methodology: A fully bioinformatic methodology has been followed. Transcription factor binding sequences, together with their adjacent sequences, have been aligned to the complete genomes of *E. coli* strains. Subsequently, the alignments were filtered following quality criteria and the count of the substitutions was carried out, differentiating the transcription factor binding sequences from their adjacent sequences. Finally, substitution rates have been compared using nonparametric tests.

Results: The results consist of a descriptive graph and a hypothesis test for each of the analyzed cases.

Conclusions: The main conclusion obtained is that, in E.coli, the mutation rates are lower in the transcription factor binding sequences than in their adjacent sequences, regardless of whether, when binding to a certain transcription factor, the sequence activates or represses transcription. In the face of future lines of research it would be interesting to carry out a broader analysis, in which different bacterial groups were included.

Índice

1. Introducción 1
1.1 Contexto y justificación del Trabajo 1
1.2 Objetivos del Trabajo 2
1.2.1 Objetivos generales: 2
1.2.2 Objetivos específicos: 3
1.3 Enfoque y método seguido 3
1.4 Planificación del Trabajo 4
1.5 Breve sumario de productos obtenidos 7
1.6 Breve descripción de los otros capítulos de la memoria 7
2. Materiales y Métodos 9
2.1 Secuencias de unión a factores de transcripción 9
2.2 Genomas completos de las cepas analizadas 9
2.3 Obtención de las secuencias adyacentes 10
2.4 Alineamiento de las secuencias 11
2.5 Filtrado de alineamientos 12
2.6 Conteo de sustituciones 12
2.7 Análisis estadístico 12
3. Resultados 13
4. Conclusiones 16
4. Glosario 18
5. Bibliografía 19
6. Anexos 22
6.1 Secuencias de unión a factores de transcripción 22
6.2 Genomas completos de las cepas analizadas 22
6.3 Obtención de las secuencias adyacentes 25
6.4 Alineamiento de las secuencias 25
6.5 Filtrado de alineamientos 26
6.6 Conteo de sustituciones 27
6.7 Análisis estadístico 33

Lista de figuras

Figura 1. Calendario de planificación del TFM durante octubre. 5
Figura 2. Calendario de planificación del TFM durante noviembre. 5
Figura 3. Calendario de planificación del TFM durante diciembre..... 6
Figura 4. Se muestra el calendario de las pruebas de evaluación continua, así como los hitos parciales alcanzados en las mismas. 6
Figura 5. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción y en sus secuencias adyacentes. 13
Figura 6. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción activadoras y en sus secuencias adyacentes..... 14
Figura 7. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción represoras y en sus secuencias adyacentes..... 15

1. Introducción

1.1 Contexto y justificación del Trabajo

Desde hace unos años se conoce la importancia de la organización de la cromatina, la accesibilidad del DNA y la sincronización de la replicación del DNA en la variación de la tasa de mutación somática a lo largo del genoma de células cancerosas (Schuster-Böckler & Lehner, 2012[1]; Lawrence et al., 2013[2]; Polak et al., 2014[3], 2015[4]). Según Supek y Lehner (2015)[5], la reparación desigual del DNA es, en última instancia, la causante de la variabilidad en las tasas de mutación a lo largo del genoma humano. Otros estudios, como los de Polak et al. (2014)[3] también parecen apuntar a que la accesibilidad de la maquinaria de reparación constituye un factor determinante en la tasa de mutación.

Recientemente han surgido dos artículos que relacionan directamente elevadas tasas de mutación con secuencias de unión a factores de transcripción, tanto en genomas de levaduras (Reijns et al., 2015)[6] como en células de tumores colorrectales (Katainen et al., 2015)[7]. Mientras que Reijns et al. (2015)[6] interpretaron esta relación como un bloqueo parcial de la maquinaria de reparación, provocado por la unión del factor de transcripción a su secuencia de unión, Katainen et al. (2015)[7] lo atribuyeron a una replicación defectuosa del DNA debido a condiciones anormales.

A raíz de estos trabajos Sabarinathan, Mularoni, Deu-Pons, Gonzalez-Perez & Lopez-Bigas (2016)[8] trataron de elucidar el impacto de la unión DNA-proteínas sobre la reparación del DNA. En primer lugar, analizaron la tasa de mutación de secuencias de unión a factores de transcripción en 38 muestras de melanoma primario, obteniendo resultados concordantes con los de otros trabajos. La tasa de mutación era significativamente superior en secuencias de unión a factores de transcripción que en el resto de secuencias, exceptuando otras secuencias asociadas de forma importante con la unión a proteínas, como son aquellas en las que se ensambla el nucleosoma. Este aumento en la tasa de mutación no era observable en secuencias de unión a factores de transcripción inactivas, por lo que se podía descartar una explicación basada únicamente en las secuencias implicadas y quedaba patente que era necesaria la unión DNA-proteínas para poder observar un aumento en la tasa de mutación.

El siguiente paso fue comprobar cuál era la causa de dicho aumento en la tasa de mutación. Alexandrov et al.(2013)[9] habían constatado que la principal causa de las mutaciones somáticas en meloncitos se debía a su exposición a la radiación ultravioleta. Esto llevó a Sabarinathan et al. (2016)[8] a proponer dos hipótesis para explicar el incremento observado en la tasa de mutación. Por una parte, podría deberse a la reparación defectuosa del DNA en estos sitios, o podría estar causada por una mayor probabilidad de lesiones inducidas por

radiación ultravioleta en sitios de unión DNA-proteínas. Para comprobarlo llevaron a cavo un mapeo de la actividad de la maquinaria de reparación por escisión de nucleótidos (NER) en secuencias de fibroblastos epiteliales con mutaciones inducidas por radiación ultravioleta. Los resultados obtenidos mostraban claramente una disminución en la actividad de la maquinaria de reparación en los sitios de unión de factores de transcripción en comparación con las secuencias adyacentes. Por lo tanto, se puede concluir que la accesibilidad de la maquinaria de reparación determina la tasa de mutación a nivel de nucleótidos y que la unión DNA-proteínas está dificultando la actividad de la maquinaria de reparación.

Teniendo en cuenta que en procariotas se produce una regulación transcripcional, mediante factores de transcripción (Van Hijum, Medema & Kuipers, 2009)[10] equivalente a la que ocurre en eucariotas, y que, del mismo modo, existe equivalencia (aunque no homología) entre los mecanismos de reparación por escisión de nucleótidos en eucariotas y bacterias (Petit & Sancar, 1999)[11]. Parece razonable proponer que la unión DNA-proteínas también dificulta la actividad de la maquinaria de reparación de DNA en bacterias y, en consecuencia, existe un incremento de la tasa de mutación en secuencias de unión a factores de transcripción de estos organismos. De ser así, podríamos estar hablando de un importante mecanismo evolutivo de introducción de variabilidad en las redes transcripcionales bacterianas, con el consecuente efecto que esto tiene en la enorme adaptabilidad que presentan las bacterias.

El presente trabajo se propone analizar y comparar las tasas de mutación en los sitios de unión de factores de transcripción y en sus regiones colindantes, en uno de los organismos modelo bacterianos en los que se posee más información sobre regulación transcripcional, *Escherichia coli* (Thieffry et al., 1998[12]; Shen-Orr et al., 2002[13]; Ma et al., 2004[14]; Faith et al., 2007[15]; Balaji et al., 2007[16]; Balleza et al., 2009[17]). De este modo, se pretende comprobar si la disminución de la tasa de mutación en las secuencias de unión a factores de transcripción con respecto a sus regiones colindantes, observadas en eucariotas, se extiende también a organismos procariotas, lo cual supondría la base para realizar estudios más extensos, en los que se incluyesen múltiples grupos bacterianos.

1.2 Objetivos del Trabajo

1.2.1 Objetivos generales:

1. Determinar si las tasas de mutación de secuencias de unión a factores de transcripción difieren significativamente de las tasas de mutación de sus regiones colindantes, en distintas cepas de Escherichia coli.

1.2.2 Objetivos específicos:

1.1 Obtener las secuencias de unión a factores de transcripción en el genoma de referencia, así como las secuencias de los genomas de las cepas de *E.coli* empleadas en el análisis.

1.2 Obtener las regiones colindantes de las secuencias de unión a factores de transcripción.

1.3 Encontrar y alinear las secuencias homólogas en los distintos organismo utilizados.

1.4 Determinar el número de sustituciones en las secuencias homólogas y regiones colindantes.

1.5 Analizar y comparar las tasas de mutación de las secuencias de unión a factores de transcripción y sus regiones colindantes.

1.3 Enfoque y método seguido

Existen dos aproximaciones posibles para este trabajo:

-Bioinformática: Se utiliza la información disponible en bases de datos de referencia. Para determinar las tasas de mutación de las secuencias, se consideran las sustituciones de bases presentes en las secuencias homólogas de los distintos serotipos de cada bacteria estudiada. Por lo tanto, metodológicamente, lo que se persigue en esta aproximación es obtener alineamientos de calidad entre secuencias homólogas para poder analizar las mutaciones presentes en las mismas.

-Experimental: Se trabaja directamente con las especies estudiadas para obtener información de primera mano, en lugar de emplear información ya disponible en bases de datos. En este caso, para determinar las tasas de mutación, se inducen las mutaciones directamente en los organismos del estudio, empleando para ello distintos agentes físico-químicos (principalmente radiación ultravioleta), y analizando a posteriori que secuencias se han visto más afectadas, las de unión a factores de transcripción o sus regiones colindantes. Metodológicamente, se trata de una aproximación mucho más compleja y que requiere un control más exhaustivo de las variables que intervienen en el estudio.

Para poder alcanzar los objetivos propuestos en este estudio, se ha elegido seguir una aproximación bioinformática. De este modo se reduce significativamente la dificultad del estudio (el número de variables a tener en cuenta, la complejidad del diseño experimental…etc) y los riesgos asociados al mismo, pudiendo obtener unas conclusiones igual de significativas que con un enfoque experimental. Por otra parte, para satisfacer los objetivos del estudio no es necesaria la obtención de información de primera mano, y sí que es muy interesante la utilización de la gran cantidad de información que ya se posee sobre los genomas de muchísimas especies, permitiendo de este modo

economizar enormemente en los recursos necesarios para realizar este proyecto. Además, con esta aproximación se favorece una adquisición de competencias por parte del alumno más acordes al contenido del máster.

1.4 Planificación del Trabajo

1.1.a) Buscar en la base de datos de referencia RegulonDB (Gama-Castro et al., 2016)[18] y descargar los datos correspondientes a las secuencias de unión a factores de transcripción de Escherichia coli correspondientes a la cepa K-12 MG1655.

1.1.b) Descargar los genomas de las cepas de *E.coli* empleadas en el análisis.

1.1.c) Familiarizarse con los datos y ajustar su formato y contenido si fuese necesario.

1.2.a) Mapear las secuencias de unión a factores de transcripción en el genoma de referencia y guardar las secuencias colindantes, en torno a 250 pares de bases en cada dirección, a las secuencias de unión a factores de transcripción.

1.3.a) Buscar secuencias homólogas en los distintos serotipos de *E.coli* mediante la herramienta BLASTN, incluida en la suite BLAST+ (Camacho et al., 2009)[19].

1.3.b) Establecer un criterio de calidad de los alineamientos y descartar los alineamientos poco fiables.

1.4.a) Contar el número de sustituciones en las secuencias homólogas, teniendo en cuenta cuales de ellas se producen en secuencias de unión a factores de transcripción y cuales en las regiones colindantes. Se tendrá en cuenta la modalidad de regulación de la secuencia de unión a factores de transcripción (inducible o reprimible), para evaluar separadamente ambos casos.

1.5.a) Utilizar estadística descriptiva para examinar las diferencias entre las secuencias de unión a factores de transcripción y sus regiones colindantes.

1.5.b) Emplear tests no paramétricos para determinar si las diferencias entre las secuencias de unión a factores de transcripción y sus regiones colindantes son significativas.

A continuación se muestra el calendario seguido para el desarrollo del proyecto (Figura 1, Figura 2 y Figura 3), así como los hitos parciales alcanzados en las pruebas de evaluación continua (Figura 4).

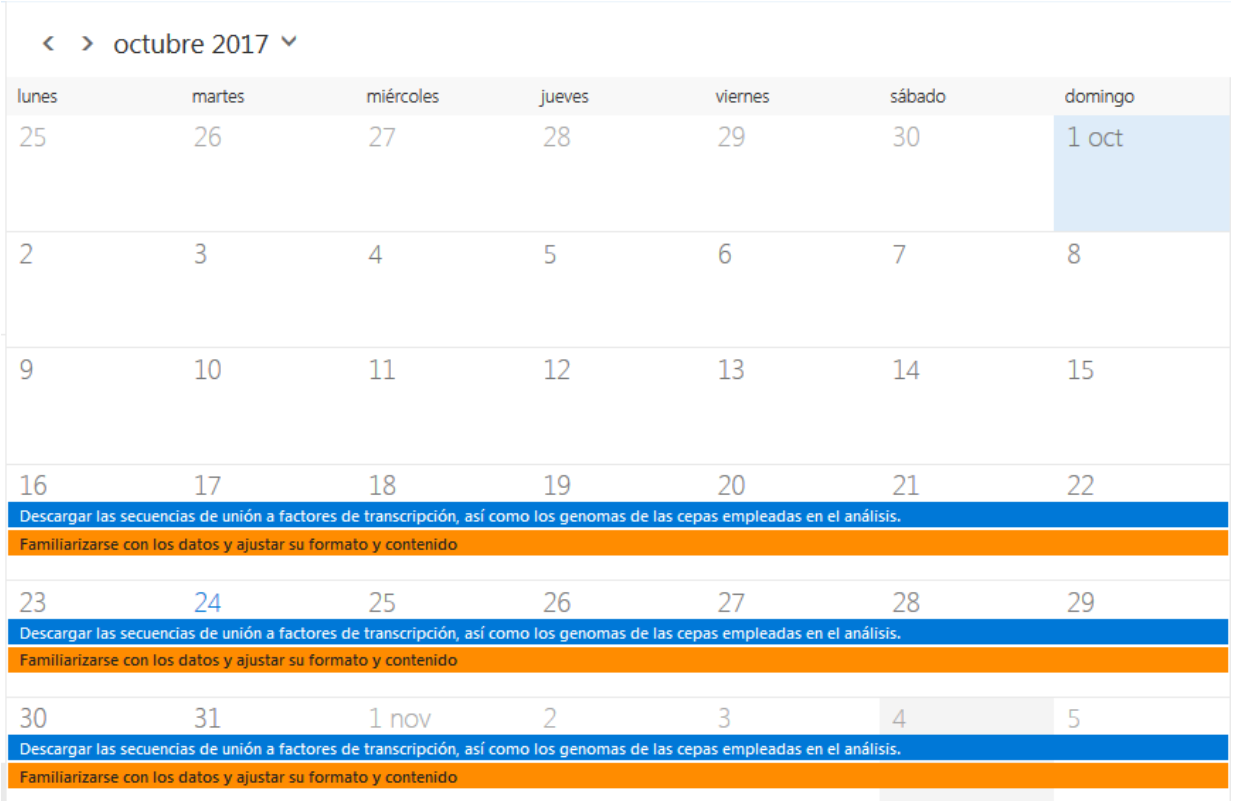

Figura 1. Calendario de planificación del TFM durante octubre.

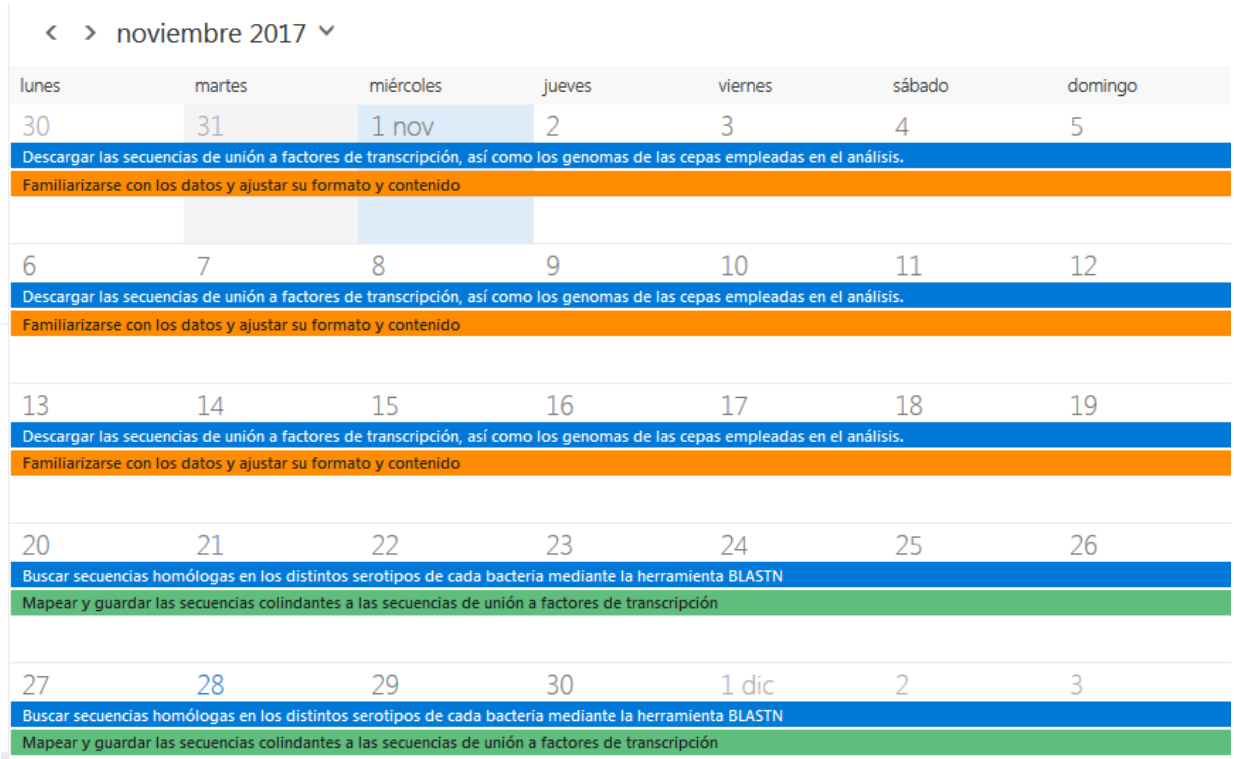

Figura 2. Calendario de planificación del TFM durante noviembre.

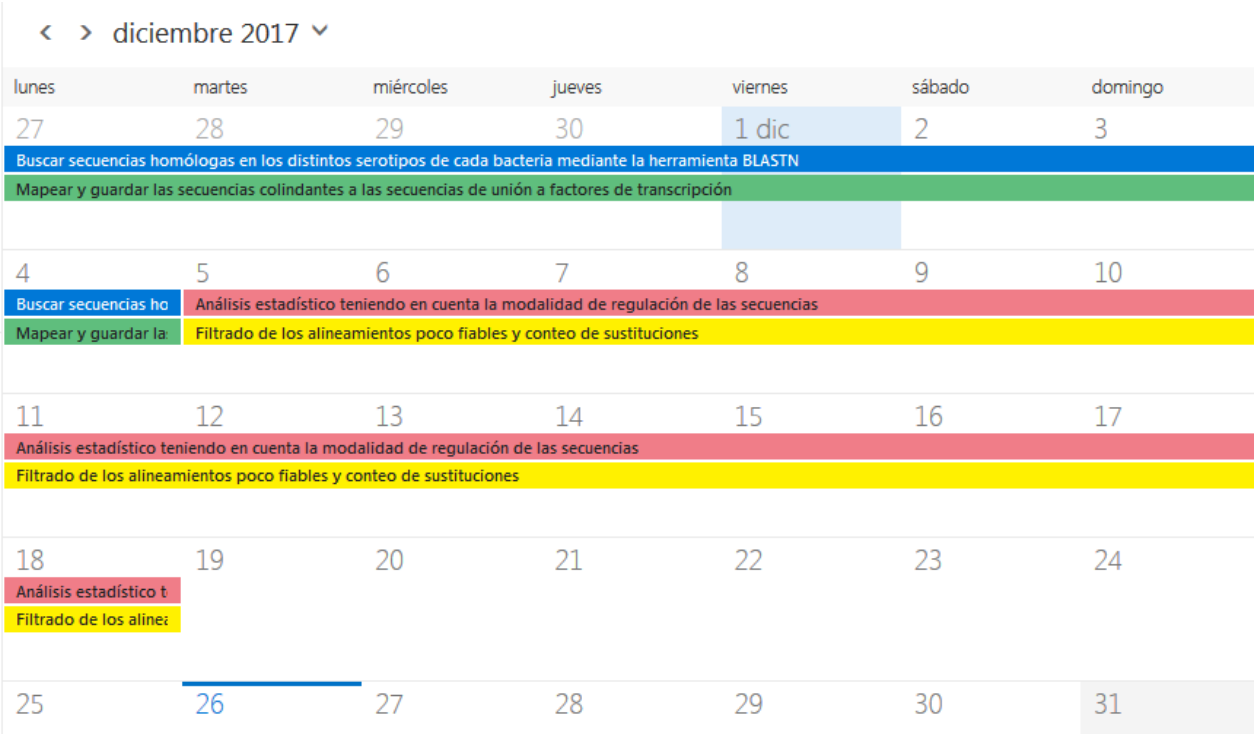

Figura 3. Calendario de planificación del TFM durante diciembre.

Nombre	Inicio	Entrega	Hitos
PEC0 - Definición de los contenidos del trabajo	20/09/2017	02/10/2017	Definición de la temática del proyecto.
PEC1 - Plan de trabajo	03/10/2017	16/10/2017	Planificación del trabajo.
PEC2 - Desarrollo del trabajo - Fase 1	17/10/2017	20/11/2017	Obtención de las secuencias de trabajo para cada organismo.
PEC3 - Desarrollo del trabajo - Fase 2	21/11/2017	18/12/2017	Disponer de suficientes alineamientos de calidad. ---------- Análisis estadístico satisfactorio y concluyente.
PEC4 - Redacción de la memoria	19/12/2017	02/01/2018	Redacción de la memoria.
PEC5a - Elaboración de la presentación	03/01/2018	10/01/2018	Elaboración de la presentación.
PEC5b - Defensa pública	11/01/2018	22/01/2018	Defensa pública.

Figura 4. Se muestra el calendario de las pruebas de evaluación continua, así como los hitos parciales alcanzados en las mismas.

1.5 Breve sumario de productos obtenidos

-'DiccionarioTFM.json': Lista de diccionarios que contienen las secuencias de unión a factores de transcripción, así como información relevante sobre cada secuencia.

-'dataset.json': Lista de diccionarios con la información de cada genoma, de las cepas de *E.coli*, a descargar.

-‘TFBSs.fasta’: Archivo FASTA con las secuencias de unión a factores de transcripción y su secuencias adyacentes, correspondientes al genoma de referencia *E.coli* K-12 MG1655.

-‘merged.fasta’: Archivo FASTA con todos los genomas de las cepas de *E.coli* descargados.

-‘blast_output.xml’: Archivo con los alineamientos resultantes del blastn.

-‘dictt_final.json’: Archivo con las tasas de sustitución de todos los alineamientos.

-‘dic+_final.json’: Archivo con las tasas de sustitución de los alineamientos cuyas secuencias de unión a factores de transcripción son activadoras.

-‘dic-_final.json’: Archivo con las tasas de sustitución de los alineamientos cuyas secuencias de unión a factores de transcripción son represoras.

-‘ R_TFM.Rmd’: Archivo con los análisis estadísticos resultantes.

1.6 Breve descripción de los otros capítulos de la memoria

A continuación se presenta un breve resumen del resto de capítulos de la memoria:

2. Materiales y Métodos

Tras una breve descripción de los principales recursos empleados a lo largo del proyecto, se exponen de forma concisa cada uno de los apartados que se han desarrollado para poder obtener los resultados buscados. Cada uno de estos apartados se correlacionan con los objetivos y tareas establecidos en al planificación del trabajo.

2.1 Secuencias de unión a factores de transcripción

En este apartado se describe la obtención de las secuencias de unión a factores de transcripción, correspondientes a la cepa de referencia *E.coli* K-12, junto con información necesaria para la obtención de sus secuencias adyacentes.

2.2 Genomas completos de las cepas analizadas

Se expone de forma concisa los pasos necesarios para descargar los genomas empleados en el alineamiento, así como las bases de datos empleadas para dicho fin.

2.3 Obtención de las secuencias adyacentes

En este apartado se explica cómo se han obtenido las secuencias adyacentes a las secuencias de unión a factores de transcripción, mediante la información obtenida en el apartado 2.1 sobre las posiciones de las secuencias y el genoma de referencia *E.coli* K-12 MG1655, descargado en el apartado 2.2.

2.4 Alineamiento de las secuencias

Se detallan los pasos seguidos para generar una base de datos a partir de los genomas descargados en el apartado 2.2 , así como los archivos y código empleados para el alineamiento de las secuencias mediante un BLAST local.

2.5 Filtrado de alineamientos

Se establecen dos criterios de calidad para los alineamientos y se describe su filtrado según dichos criterios.

2.6 Conteo de sustituciones

Se detalla el proceso seguido para programar el conteo automático de las sustituciones de bases de los alineamientos, así como la obtención de las tasas de sustitución. Se especifica, además, la división de los conteos en tres casos separados: todos los alineamientos, alineamientos cuyas secuencias de unión a factores de trascripción sean activadoras y alineamientos cuyas secuencias de unión a factores de trascripción sean represoras.

2.7 Análisis estadístico

Se detallan los gráficos generados y las pruebas no paramétricas realizadas sobre los conteos de los tres casos separados obtenidos en el punto 2.6.

3. Resultados

En este apartado se presentan y describen tanto los gráficos como los resultados de las pruebas no paramétricas resultantes de los análisis estadísticos realizados en el punto 2.7.

4. Conclusiones

Se exponen las conclusiones obtenidas sobre los resultados, así como una valoración sobre las limitaciones del trabajo, el seguimiento de la planificación y el cumplimiento de los objetivos.

4. Glosario

5. Bibliografía

6. Anexos

Se incluye el código generado en el resto de apartados con comentarios.

2. Materiales y Métodos

A lo largo del proyecto se han empleado:

-Python (Sanner, 1999)[20] en Jupyter (Ragan-Kelley et al., 2014)[21], incluidos en la distribución Anaconda, para la obtención de secuencias, el filtrado de alineamientos y el conteo de sustituciones.

-La suite de herramientas BLAST+ (Camacho et al., 2009)[19], para el alineamiento de las secuencias.

-R (RCTeam, 2000)[22] en la plataforma Rstudio (Racine, 2012)[23], para los análisis estadísticos.

2.1 Secuencias de unión a factores de transcripción

Las secuencias de unión a factores de transcripción se han descargado directamente desde RegulonDB (Gama-Castro et al., 2016)[18], específicamente desde los sets de datos respaldados por evidencia experimental en la literatura científica. Las secuencias se pueden consultar en:

http://132.248.248.120/menu/download/datasets/files/BindingSiteSet.txt

Estas secuencias corresponden a la cepa de referencia *E.coli* K-12 e incluyen información relevante sobre las mismas (Identificador de la secuencia, nombre del factor de transcripción, sitio de inicio y final de la secuencia…etc.). Empleando python (Sanner, 1999)[20] se ha modificado ligeramente el formato de la información contenida en el set de datos y se ha creado una lista de diccionarios ('DiccionarioTFM.json'), cada uno de los cuales incluye la información correspondiente a una secuencia de unión a factores de transcripción (Código comentado en el Anexo).

2.2 Genomas completos de las cepas analizadas

Para descargar los genomas completos de todas las cepas de *E.coli* disponibles en NCBI (NCBI Resource Coordinators, 2016)[24], en primer lugar se ha descargado una tabla con información precisa sobre los genomas de cada cepa (genomes_proks.txt), incluyendo su código de acceso para distintas bases de datos. Dicha tabla se encuentra disponible en la base de datos

Genome de NCBI (NCBI Resource Coordinators, 2016)[24] y se puede consultar en: https://www.ncbi.nlm.nih.gov/genome/genomes/167

Empleando python (Sanner, 1999)[20] se modificó ligeramente el formato de los datos descargados y se generó una lista de diccionarios, asegurando el filtrado de aquellos genomas que no se encontrasen completos ('dataset.json'). El código empleado puede consultarse en los anexos.

El siguiente paso fue descargar los genomas completos de todas las cepas, para lo cual se empleó Entrez (Maglott et al., 2005)[25], que permite el acceso de los usuarios a múltiples bases de datos de NCBI (NCBI Resource Coordinators, 2016[24]). Dicho acceso se puede realizar mediante un navegador, pero en nuestro caso, al necesitar descargar gran cantidad de secuencias, fue necesario acceder programáticamente. Para ello, se empleó el módulo Bio.*Entrez* de Biopython (Cock et al., 2009)[26], que permite acceder de forma remota a Entrez (Maglott et al., 2005)[25] desde un entorno de programación python (Sanner, 1999)[20].

Se barajaron dos opciones, descargar los genomas desde la base de datos RefSeq (Pruitt et al., 2006)[27], con información altamente contrastada para eliminar redundancias, o descargar los genomas desde GenBank (Benson et al,. 1993)[28], que incluye todas las secuencias públicamente disponibles. Finalmente se optó por la última opción, puesto que las posiciones de los sitios de unión a factores de transcripción diferían entre RegulonDB (Gama-Castro et al., 2016)[18] y RefSeq (Pruitt et al., 2006)[27] para la cepa de referencia *E.coli* K-12, lo cual, como se verá más adelante, sería un problema importante a la hora de recuperar las regiones adyacentes a las secuencias de unión a factores de transcripción.

Por lo tanto, se programó la descarga de los genomas desde Entrez (Maglott et al., 2005)[25] utilizando los códigos de acceso de los genomas en GenBank (Benson et al,. 1993)[28] contenidos en el archivo 'dataset.json' y con una latencia entre descargas de 2 segundos para evitar sobrecargas del servidor. Como resultado, se obtuvo un archivo en formato genbank por cada genoma descargado. Todo el código empleado en esta sección puede consultarse en el Anexo.

2.3 Obtención de las secuencias adyacentes

En última instancia, se buscaba obtener las secuencias de unión a factores de transcripción más 500 pares de bases de secuencias adyacentes, 250 pares de bases por la izquierda y otros 250 pares de bases desde la derecha.

Mediante Bio.SeqIO, incluido en Biopython (Cock et al., 2009)[26], se ha cargado el genoma de referencia *E.coli* K-12 MG1655, incluido en la descarga de los genomas completos del apartado anterior, en un entono python (Sanner, 1999)[20]. A continuación, se utilizan la posiciones de inicio y final de cada secuencia de unión a factores de transcripción contenidas en el archivo 'DiccionarioTFM.json' para localizar dichas secuencias en el genoma de referencia. En lugar de emplear las posiciones originales a la posición de inicio

se le resta 250 pares de bases y al posición final se le suma 250 pares de bases. De este modo se obtienen las secuencias de unión a factores de transcripción junto con sus secuencias adyacentes. Dichas secuencias se guardaron en el archivo en formato FASTA 'TFBSs.fasta' para los alineamientos posteriores. Todo el código empleado en este apartado puede encontrarse en el Anexo.

2.4 Alineamiento de las secuencias

Para realizar los alineamientos se descargó e instaló localmente BLAST+ (Camacho et al., 2009)[19] desde:

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

En primer lugar, se generó una base de datos con las secuencias sobre las que se realizarán los alineamientos, en nuestro caso sobre todos los genomas completos descargados. Para ello, se cambió el formato de todos los genomas descargados al formato FASTA mediante python (Sanner, 1999)[20] y se unificaron en un único archivo FASTA mediante el comando del entorno Windows:

```
C:\Users\Gonzalo\TFM\fasta_sequences\ copy *.fasta merged.fasta
```

De este modo, se obtuvo el archivo 'merged.fasta' que contiene todos los genomas analizados en formato FASTA. La base de datos se creó mediante el siguiente comando mínimo:

```
C:\Users\Gonzalo\TFM\fasta_sequences\ makeblastdb –in merged.fasta –dbtype nucl –parse_seqids
```

El resultado fue la creación de 6 ficheros para la base de datos: 'merged.fasta.nhr', 'merged.fasta.nin', 'merged.fasta.nog', 'merged.fasta.nsd', 'merged.fasta.nsi' y 'merged.fasta.nsq'.

Las secuencias para el alineamiento se encuentran en el archivo 'TFBSs.fasta' y corresponden a las secuencias de unión a factores de transcripción, y sus regiones colindantes, del genoma de referencia *E.coli* K-12 MG1655. La base de datos sobre la que se realizarán los alineamientos, 'merged.fasta', contiene todos los genomas completos de las cepas de *E.coli*. El último paso es realizar los alineamientos:

```
C:\Users\Gonzalo\TFM\fasta_sequences\ blastn –db merged.fasta –query TFBSs.fasta –outfmt 5 –out blast_output.xml
```

Obtenemos el archivo en formato .xml, 'blast_output.xml', que contiene todos los alineamientos. Se ha elegido este formato por su conveniencia a la hora de trabajar en python (Sanner, 1999)[20], como se verá en el siguiente apartado. Todo el código correspondiente a este apartado se puede consultar en el Anexo.

2.5 Filtrado de alineamientos

Tras cargar en python (Sanner, 1999)[20] los alineamientos mediante el paquete SearchIO incluido en Biopython (Cock et al., 2009)[26], establecemos un filtraje siguiendo dos criterios de calidad para los alineamientos:

-El E-value o valor esperado (E), que indica el "ruido" de fondo de un alineamiento y que, por lo tanto, se puede interpretar como un indicador de como de significativo es dicho alineamiento. En nuestro caso, consideramos significativos los alineamientos con un E-value inferior a $1 \cdot 10^{-10}$.

-La cobertura del alineamiento, que indica, para cada alineamiento, que porcentaje de la secuencia a alinear se encuentra realmente alineada. En nuestro caso, consideramos significativos los alineamientos con una cobertura superior al 70%.

Por ser un paso sencillo, se empleó directamente el diccionario creado a partir de los alineamiento filtrados para el conteo de las sustituciones, sin crear un nuevo archivo independiente.

2.6 Conteo de sustituciones

El conteo de sustituciones se realiza directamente sobre el diccionario de alineamientos filtrados. Se han programado en python (Sanner, 1999)[20] dos contadores para cada alineamiento, uno para la secuencia de unión a factores de transcripción y otro para sus regiones adyacentes. Los contadores solo cuentan las sustituciones de bases y están programados para tener en cuenta la posición de inicio del alineamiento y la longitud de la secuencia a alinear, evitando espacios.

Por otro lado, se ha dividido el número de sustituciones en cada contador por el número de nucleótidos en el que se han contado dichas sustituciones, con el objetivo de hacer comparables dichos valores entre las secuencias de unión a factores de transcripción y sus regiones adyacentes.

Además se han realizado tres conteos independientes, uno con todos los alineamientos, otro que incluye únicamente los alineamientos cuyos sitios de unión a factores de transcripción sean activadores (al unirse a un factor de transcripción concreto) y el último que incluye únicamente los alineamientos cuyos sitios de unión a factores de transcripción sean represores (al unirse a un factor de transcripción concreto). Por lo tanto, se han obtenido tres archivos independientes con los conteos, 'dictt_final.json', 'dic+_final.json' y 'dic-_final.json'. Todo el código empleado en este apartado puede consultarse en el anexo.

2.7 Análisis estadístico

Mediante el paquete jsonlite (Ooms, 2014)[29], se cargaron los archivos JSON con las tasas de sustitución en R (RCTeam, 2000)[22]. Siempre distinguiendo entre las tasas de sustitución de las secuencias de unión a factores de

transcripción y de sus secuencias adyacentes, se han calculado sus respectivas medias y medianas. Además, se han generado histogramas representativos de ambos casos para poder realizar una primera comparación de los datos.

Como los datos no siguen una distribución normal, se ha empleado el test no paramétrico U de Mann-Whitney (Mann y Whitney, 1947)[30] para compararlos. Consideramos los datos como pareados, puesto que las secuencias de unión a factores de transcripción comparten el mismo 'background' genético que sus respectivas secuencias adyacentes. La hipótesis nula elegida para el contraste es:

H0: La tasa se sustitución de las secuencias de unión a factores de transcripción no es menor que la de sus secuencias adyacentes.

Los análisis estadísticos se han realizado sobre cada uno de los tres casos analizados: todos los alineamientos, alineamientos con sitios de unión a factores de transcripción activadores y alineamientos con sitios de unión a factores de transcripción represores. Todo el código empleado en este apartado puede encontrarse en el anexo.

3. Resultados

En primer lugar, veamos los resultados de los análisis para todos los alineamientos (Figura 5):

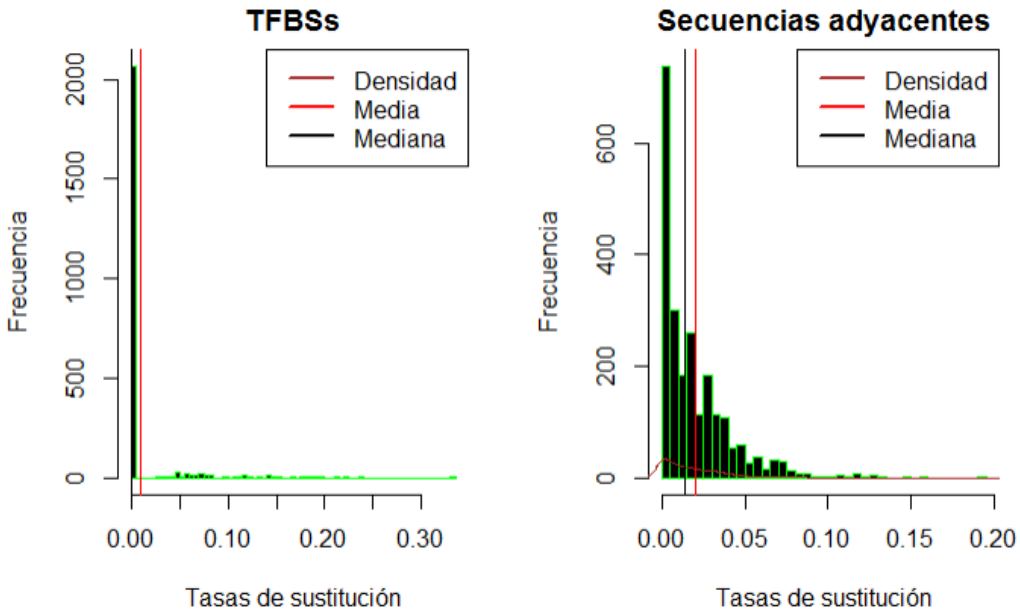

Figura 5. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción y en sus secuencias adyacentes.

Se pueden apreciar claramente las diferencias en las tasas de sustitución. En las secuencias de unión a factores de transcripción, las tasas de sustitución son predominantemente cero, la mediana es cero y la media es muy pequeña. Por otra parte, en la secuencias adyacentes nos encontramos mayores

frecuencias de tasas de sustitución distintas a cero, y la media y mediana son mayores que en el caso de las secuencias de unión a factores de transcripción. Es por esta diferencia tan aparente que se ha elegido la hipótesis nula para el contraste:

H0: La tasa se sustitución de las secuencias de unión a factores de transcripción no es menor que la de sus secuencias adyacentes.

El resultado del contraste es el siguiente:

```r
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE, alternative = "less")
```

```
        Wilcoxon signed rank test with continuity correction

data:  mutations$X2 and mutations$X4
V = 312150, p-value < 2.2e-16
alternative hypothesis: true location shift is less than 0
```

Para un nivel de significación α=0,01 podemos rechazar la hipótesis nula y, por lo tanto, confirmamos que la tasa se sustitución de las secuencias de unión a factores de transcripción es menor que la de sus secuencias adyacentes.

Comparemos los resultados con los obtenidos para los alineamientos con secuencias de unión a factores de transcripción activadoras (Figura 6).

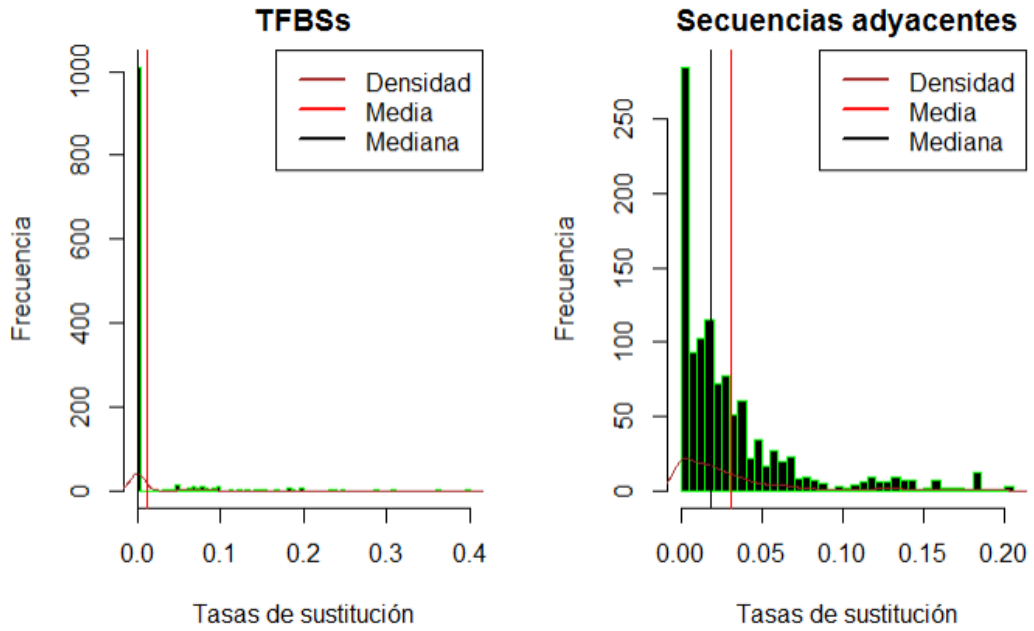

Figura 6. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción activadoras y en sus secuencias adyacentes.

Como podemos ver, los histogramas son similares a los obtenidos para las tasas de sustitución de todos los alineamientos. Se aprecia un incremento notable en la frecuencia de tasas de sustitución distintas a cero en secuencias

adyacentes, con el correspondiente incremento de la media y la mediana, con respecto a las secuencias de unión a factores de transcripción.

De nuevo, la hipótesis nula para el contraste es:

H0: La tasa se sustitución de las secuencias de unión a factores de transcripción no es menor que la de sus secuencias adyacentes.

Realizamos el contraste:

```r
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE, alternative = "less")
```

```
        Wilcoxon signed rank test with continuity correction

data:  mutations$X2 and mutations$X4
V = 60354, p-value < 2.2e-16
alternative hypothesis: true location shift is less than 0
```

El resultado del contraste es idéntico al realizado para las tasas de sustitución de todos los alineamientos. Para un nivel de significación α=0,01 podemos rechazar la hipótesis nula y, por lo tanto, confirmamos que la tasa de sustitución de las secuencias de unión a factores de transcripción activadoras es menor que la de sus secuencias adyacentes.

Por último, veamos los resultados obtenidos para los alineamientos con secuencias de unión a factores de transcripción represoras (Figura 7).

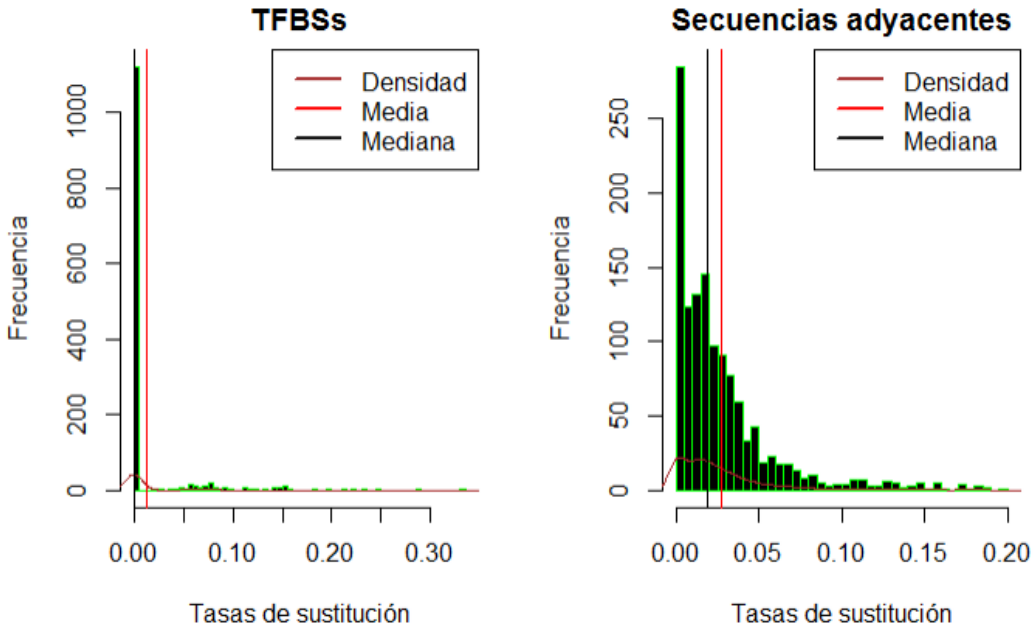

Figura 7. Se representa la frecuencia con la que aparece cada tasa de sustitución en las secuencias de unión a factores de transcripción represoras y en sus secuencias adyacentes.

De nuevo obtenemos gráficos similares a los ya presentados, observamos un incremento notable en la frecuencia de tasas de sustitución distintas a cero en secuencias adyacentes, con respecto a las secuencias de unión a factores de transcripción, cuya frecuencia de tasas de sustitución es predominantemente cero.

Por lo tanto, podemos emplear la misma hipótesis nula que en los casos anteriores .

H0: La tasa se sustitución de las secuencias de unión a factores de transcripción no es menor que la de sus secuencias adyacentes.

El resultado del contraste es el siguiente:

```{r}
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE, alternative = "less")
```

```
        Wilcoxon signed rank test with continuity correction

data:  mutations$X2 and mutations$X4
V = 101300, p-value < 2.2e-16
alternative hypothesis: true location shift is less than 0
```

Al igual que en los casos anteriores, podemos rechazar la hipótesis nula con un nivel de significación α=0,01, confirmando que la tasa de sustitución de las secuencias de unión a factores de transcripción represoras es menor que la de sus secuencias adyacentes.

4. Conclusiones

Vistos los resultados, podemos concluir que, en *E.coli*, las tasas de mutación (teniendo en cuenta tan solo las sustituciones), son menores en las secuencias de unión a factores de transcripción que en sus secuencias adyacentes, indistintamente de si, al unirse a un determinado factor de transcripción, la secuencia activa o reprime la transcripción. Dicha conclusión da respuesta al objetivo general del trabajo, determinar si las tasas de mutación de secuencias de unión a factores de transcripción difieren significativamente de las tasas de mutación de sus regiones colindantes, en distintas cepas de Escherichia coli, y ha sido posible alcanzarla gracias a la consecución de los objetivos parciales establecidos en la planificación.

Los resultados obtenidos no concuerdan con las investigaciones realizadas en células eucariotas (Reijns et al., 2015[6]; Katainen et al., 2015[7]), a pesar de que en procariotas se produce una regulación transcripcional (Van Hijum, Medema & Kuipers, 2009)[10] equivalente a la que ocurre en eucariotas, y de que sus mecanismos de reparación por escisión de nucleótidos sean similares (Petit & Sancar, 1999)[11]. Si se hubiesen reproducido los resultados obtenidos en eucariotas, que muestran tasas de mutación superiores en secuencias de unión a factores de transcripción que en sus secuencias adyacentes (Reijns et al.,

2015[6]; Katainen et al., 2015[7]), podríamos estar hablando de un importante mecanismo evolutivo de introducción de variabilidad en las redes transcripcionales bacterianas. Por el contrario, nuestros resultados parecen apoyar una visión más conservativa de las secuencias de unión a factores de transcripción. Desde este punto de vista, se entienden las secuencias de unión a factores de transcripción como unidades fundamentales para el mantenimiento de las redes transcripcionales bacterianas, y, debido a la importancia que esto tiene para el organismo, parece razonable que sean secuencias poco variables. Sin embargo, las evidencias presentadas en este trabajo no son suficientes como para poder alcanzar ningún tipo de conclusión general al respecto, siendo necesarias futuras investigaciones para esclarecer este punto.

Es necesario tener en cuenta los límites de los resultados obtenidos. El estudio se ha realizado únicamente sobre diversas cepas de *E.coli* y es posible que la variabilidad entre las distintas cepas sea baja, siendo este un posible factor importante que afectaría a nuestros resultados. Además, hay que tener en cuenta la posibilidad de que los resultados no sean extrapolables a otros grupos bacterianos. De cara a futuras líneas de investigación sería interesante realizar un análisis más amplio, en el que se incluyesen distintos grupos bacterianos. De este modo, la variabilidad entre secuencias sería más significativa y se podrían obtener conclusiones más generales y extrapolables.

El seguimiento de la planificación del trabajo ha sido un tanto ajustada debido, principalmente, al tiempo disponible para su realización. En un primer momento se barajó la inclusión de distintos grupos de Enterobacterias en el análisis, lo cual finalmente no fue posible, quedando pendiente para futuros análisis. La metodología prevista ha sido la adecuada para este proyecto, pues se basa en la generación de código para una resolución adaptada específicamente a nuestro problema. La consecución de este proyecto ha supuesto un reto personal por la continuas adaptaciones necesarias (especialmente en el código empleado) y la cantidad de factores a tener en cuenta en cada uno de sus apartados.

4. Glosario

Tasa de mutación: Número de mutaciones por unidad de tiempo o espacio (pares de bases, genes…etc.).

Tasa de sustitución: Número de sustituciones de nucleótidos por unidad de tiempo o espacio (pares de bases, genes…etc.).

NER: nucleotide excision repair (reparación por escisión de nucleótidos).

TFBS: *transcription factor binding sites* (sitios de unión a factores de transcripción).

Secuencias colindantes o adyacentes: Se refieren a secuencias contiguas entre sí.

FASTA: Se trata de un formato de fichero informático, basado en el texto, para la representación de secuencias y que posee una única línea descriptiva para cada secuencia.

JSON: *JavaScript Object Notation*. Se trata de un formato de texto ligero para el intercambio de datos.

5. Bibliografía

[1] Schuster-Böckler, B., & Lehner, B. (2012). Chromatin organization is a major influence on regional mutation rates in human cancer cells. Nature, 488(7412), 504-507. doi:10.1038/nature11273

[2] Lawrence, M. S., Stojanov, P., Polak, P., Kryukov, G. V., Cibulskis, K., Sivachenko, A., … Getz, G. (2013). Mutational heterogeneity in cancer and the search for new cancer genes. Nature, 499(7457), 214–218. http://doi.org/10.1038/nature12213

[3] Polak, P., Lawrence, M. S., Haugen, E., Stoletzki, N., Stojanov, P., Thurman, R. E., … Sunyaev, S. R. (2014). Reduced local mutation density in regulatory DNA of cancer genomes is linked to DNA repair. Nature Biotechnology, 32(1), 71–75. http://doi.org/10.1038/nbt.2778

[4] Polak, P., Karlić, R., Koren, A., Thurman, R., Sandstrom, R., Lawrence, M., … Sunyaev, S. R. (2015). Cell-of-origin chromatin organization shapes the mutational landscape of cancer. Nature, 518(7539), 360–364. http://doi.org/10.1038/nature14221

[5] Supek, F., & Lehner, B. (2015). Differential DNA mismatch repair underlies mutation rate variation across the human genome. Nature, 521(7550), 81–84. http://doi.org/10.1038/nature14173

[6] Reijns, M. A. M., Kemp, H., Ding, J., de Procé, S. M., Jackson, A. P., & Taylor, M. S. (2015). Lagging strand replication shapes the mutational landscape of the genome. Nature, 518(7540), 502–506. http://doi.org/10.1038/nature14183

[7] Katainen, R., Dave, K., Pitkänen, E., Palin, K., Kivioja, T., Välimäki, N., . . . Aaltonen, L. A. (2015). CTCF/cohesin-binding sites are frequently mutated in cancer. Nature Genetics, 47(7), 818-821. doi:10.1038/ng.3335

[8] Sabarinathan, R., Mularoni, L., Deu-Pons, J., Gonzalez-Perez, A., & López-Bigas, N. (2016). Nucleotide excision repair is impaired by binding of transcription factors to DNA. Nature, 532(7598), 264-267. doi:10.1038/nature17661

[9] Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Aparicio, S. A. J. R., Behjati, S., Biankin, A. V., … Stratton, M. R. (2013). Signatures of mutational processes in human cancer. Nature, 500(7463), 415–421. http://doi.org/10.1038/nature12477

[10] Van Hijum, S. A. F. T., Medema, M. H., & Kuipers, O. P. (2009). Mechanisms and Evolution of Control Logic in Prokaryotic Transcriptional Regulation. Microbiology and Molecular Biology Reviews : MMBR, 73(3), 481–509. http://doi.org/10.1128/MMBR.00037-08

[11] Petit, C., & Sancar, A. (1999). Nucleotide excision repair: From E. coli to man. Biochimie, 81(1-2), 15-25. doi:10.1016/s0300-9084(99)80034-0

[12] Thieffry, D., Huerta, A. M., Pérez-Rueda, E. and Collado-Vides, J. (1998), From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli. Bioessays, 20: 433–440. doi:10.1002/(SICI)1521-1878(199805)20:5<433::AID-BIES10>3.0.CO;2-2

[13] Shen-Orr S., Milo R., Mangan S., Alon U. (2002). Network motifs in the transcriptional regulation network of Escherichiacoli. Nat. Genet.. 31. 1061-4036

[14] Ma, H.-W., Kumar, B., Ditges, U., Gunzer, F., Buer, J., & Zeng, A.-P. (2004). An extended transcriptional regulatory network of Escherichia coli and analysis of its hierarchical structure and network motifs. Nucleic Acids Research, 32(22), 6643–6649. http://doi.org/10.1093/nar/gkh1009

[15] Faith JJ., Hayete B., Thaden JT., Mogno I., Wierzbowski J., et al. (2007) Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. PLOS Biology 5(1): e8. https://doi.org/10.1371/journal.pbio.0050008

[16] Balaji, S., Babu, M. M., & Aravind, L. (2007). Interplay between network structures, regulatory modes and sensing mechanisms of transcription factors in the transcriptional regulatory network of E. coli. Journal of Molecular Biology, 372(4), 1108–1122. http://doi.org/10.1016/j.jmb.2007.06.084

[17] Balleza, E., López-Bojorquez, L. N., Martínez-Antonio, A., Resendis-Antonio, O., Lozada-Chávez, I., Balderas-Martínez, Y. I., … Collado-Vides, J. (2009). Regulation by transcription factors in bacteria: beyond description. Fems Microbiology Reviews, 33(1), 133–151. http://doi.org/10.1111/j.1574-6976.2008.00145.x

[18] Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., Ledezma-Tejeida, D., Muñiz-Rascado, L., García-Sotelo, J. S., ... & Medina-Rivera, A. (2016). RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. Nucleic Acids Research, 44(D1), D133-D143.

[19] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: architecture and applications. BMC Bioinformatics, 10, 421. http://doi.org/10.1186/1471-2105-10-421

[20] Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model*, *17*(1), 57-61.

[21] Ragan-Kelley, M., Perez, F., Granger, B., Kluyver, T., Ivanov, P., Frederic, J., & Bussonnier, M. (2014, December). The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication. In AGU Fall Meeting Abstracts.

[22] RCTeam (2000). R language definition. Vienna, Austria: R foundation for statistical computing.

[23] Racine, J. S. (2012). RStudio: A Platform-Independent IDE for R and Sweave. Journal of Applied Econometrics, 27(1), 167-172.

[24] NCBI Resource Coordinators. (2016). Database resources of the National Center for Biotechnology Information. Nucleic Acids Research, 44(Database issue), D7–D19. http://doi.org/10.1093/nar/gkv1290

[25] Maglott, D., Ostell, J., Pruitt, K. D., & Tatusova, T. (2005). Entrez Gene: gene-centered information at NCBI. Nucleic acids research, 33(suppl_1), D54-D58.

[26] Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., ... & De Hoon, M. J. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25(11), 1422-1423.

[27] Pruitt, K. D., Tatusova, T., & Maglott, D. R. (2006). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic acids research, 35(suppl_1), D61-D65.

[28] Benson, D., Lipman, D. J., & Ostell, J. (1993). GenBank. Nucleic Acids Research, 21(13), 2963-2965.

[29] Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects. arXiv preprint arXiv:1403.2805.

[30] Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics, 50-60.

6. Anexos

6.1 Secuencias de unión a factores de transcripción

```
import csv
#Cargamos el documento descargado desde RegulonDB
with open('BindingSiteSet.txt') as f:
    #Modificamos ligeramente el formato de cada línea del documento
    f = [x.replace(',', '').replace('\t', ',') for x in f]
    #Creamos una lista de diccionarios, en la que cada
    #diccionario contiene la información de un sitio de
    #unión a factores de transcripción
    a = [{k: str(v) for k,v in row.items()}
         for row in csv.DictReader(f, skipinitialspace=True)]

import json
#Guardamos la lista de diccionarios en un archivo JSON
#para poder utilizarla entre distintas sesiones
with open('DiccionarioTFM.json', 'w') as outfile:
    json.dump(a, outfile, indent=4, sort_keys=True, separators=(',',
':'))
```

6.2 Genomas completos de las cepas analizadas

```
import json
import csv
#Cargamos el documento descargado desde NCBI
with open('genomes_proks.txt') as g:
    #Modificamos ligeramente el formato de cada linea del documento
    g = [x.replace(',', '').replace('\t', ',') for x in g]
    #Creamos una lista de diccionarios, en la que cada
    #diccionario contiene la información de un genoma
    b = [{k: v for k, v in row.items()} for row in csv.DictReader(g,
skipinitialspace=True)]

#Creamos una lista vacía
c=[]
#Para cada diccionario de la lista
for n in b:
    #si el diccionario contiene un genoma completo
    if n["Level"]=="Complete Genome":
        #se guarda en la lista creada previamente
        c.append(n)
#Guardamos la lista de diccionarios en un archivo JSON
with open('dataset.json', 'w') as outfile:
    json.dump(c, outfile, indent=4, sort_keys=True, separators=(',',
':'))

#importamos json
 import json
```

```
#importamos time
import time

#importamos listdir
from os import listdir

#importamos Entrez (incluido en biopython)
from Bio import Entrez
Entrez.email ="ggarfield@hotmail.es"

#Establecemos el directorio de los genomas
gnome_path = 'genomes_GenBank'

#Establecemos el directorio de datos
data_path = 'src_data'

#Iniciamos el contador de los genomas descargados
N=0

#Creamos la lista vacía que contendrá los códigos
#de acceso a Genbank
l_acces=[]

#Cargamos la lista de diccionarios con la información
#sobre los genomas a descargar
with open('dataset.json') as json_file:
    dataset = json.load(json_file)

#Para cada genoma se descarga y guarda la secuencia
#utilizando el código de acceso a GenBank

#Para cada uno de los diccionarios con la información de los genomas
for orgn in dataset:
    updatejson=True
    new_access=[]
    #se obtienen los códigos de acceso a distintas bases de datos
    l_access=orgn['Replicons']
    #se modifica el formato para que sea más sencillo acceder al
    #código de acceso de GenBank más adelante
    l_access=l_access.replace('-','_')
    l_access=l_access.replace(' ','_')
    l_access=l_access.replace('/',':')
    l_access=l_access.replace(';',':')
    l_access=l_access.split(':')

    #se muestran los códigos de acceso del genoma
    print(l_access)

#obtenemos los nombres de los archivos presentes en el directorio  de
genomas
    dir_contents=listdir('genomes_GenBank')

    #para cada número de acceso
    for acc in l_access:
```

```
        #filtramos los números de acceso que no pertenezcan a GenBank
        if "NC" not in acc and "chromosome" not in acc and "plasmid"
not in acc and "NZ" not in acc and "_" not in acc:
            N=N+1
            print(N)
            print("Processing: " + acc)
            accs, numb=acc.split('.')
       #comprobamos que el genoma no se encuentre ya en el directorio
            if not((acc+'.gb') in dir_contents):
                print("--> Downloading: " + acc)
       #descargamos el genoma
                try:
                    net_handle =
Entrez.efetch(db="nuccore",id=acc,rettype='gbwithparts',retmode="txt"
)
                    gnome_record=net_handle.read()
       #obtenemos el número de acceso del propio archivo
                    start=gnome_record.find('VERSION')+9
                    end=gnome_record.find('\n',start)
                    accession_line=gnome_record[start:end].strip()
                    end=accession_line.find(' ')
                    accession = accession_line[0:end]
        #eliminamos el número la versión del número de acceso
                    accession, number = accession.split('.')
                    print("Accession: " + accession)
        #guardamos el genoma en un archivo "número de acceso".gb
                    out_handle = open(gnome_path+'/'+accession+".gb",
"w")
                    out_handle.write(gnome_record)
        #mantenemos una latencia de 2 segundos para no sobrecargar
        #el servidor
                    time.sleep(2)
        #actualizamos la lista de números de acceso
                    new_access.append(accession)
                    out_handle.close()
                    net_handle.close()
        #si ocurre un error, la descarga continua con el siguiente
        #número de acceso
                except Exception:
                    pass
        #si el número de acceso ya estaba en la lista no se actualiza
            else:
                updatejson=False
                print("Accession: " + acc)
        #si el número de acceso se ha actualizado
        if updatejson:
        #lo actualizamos en su entrada en el diccionario
            orgn['Replicons']=new_access

    #lo guardamos en un archivo JSON
        with open('dataset_final.json', 'w') as outfile:
            json.dump(dataset, outfile, indent=4, sort_keys=True,
separators=(',', ':'))
```

6.3 Obtención de las secuencias adyacentes

```
#Importamos SeqIO desde biopython
from Bio import SeqIO

#Cargamos el genoma de referencia Escherichia coli K-12 MG1655
record = SeqIO.read("genomes_GenBank/U00096.gb", "genbank")


#Recuperamos las secuencias de unión a factores de transcripción
#en el genoma de referencia

#Creamos una lista vacía
sub_record=[]

#Para cada una de las posiciones de la lista de diccionarios
for i in range(len(a)):
#Si las posiciones de la secuencia son cero no se utiliza la secuencia
    if int(a[i]["TF-bs left end position in the genome "])==0 and
int(a[i]["TF-bs right end position in the genome"])==0:
        pass
#De lo contrario
    else:
#se utilizan las posiciones de la secuencia ±250 pares de bases
#para localizar las secuencias de interés en el genoma de referencia
        sub_records=record[(int(a[i]["TF-bs left end position in the
genome "]))-250:(int(
            a[i]["TF-bs right end position in the genome"]))+250]
#guardamos cada secuencia en un diccionario
        sub_records.id=a[i]['TF binding site (TF-bs) identifier
assigned by RegulonDB ']
        sub_records.name=a[i]['TF name']
        sub_records.description='id=' + a[i]['TF binding site (TF-bs)
identifier assigned by RegulonDB '] +' '+ 'TF name=' + a[i]['TF
name']+ ' ' + 'position='+a[i]["TF-bs left end position in the genome
"]+'-'+a[i]["TF-bs right end position in the genome"]
#añadimos cada diccionario a la lista previamente creada
        sub_record.append(sub_records)
#generamos un archivo FASTA con todas las secuencias
        SeqIO.write(sub_record, "TFBSs.fasta", "fasta")
```

6.4 Alineamiento de las secuencias

```
#Generamos una lista con los nombres de los archivos
#que contienen los genomas descargados

#Importamos os
import os

#Establecemos el directorio donde están los archivos
gnome_path = 'genomes_GenBank'
```

```
#Creamos la lista vacía
file_list = []

#Para cada archivo del directorio
for file in os.listdir("./genomes_GenBank/"):
    #si el archivo tiene la terminación .gb
    if file.endswith(".gb"):
        #se añade a la lista
        file_list.append(file)

#Establecemos el directorio donde estarán los archivos FASTA
fasta_sequences="fasta_sequences"

#Importamos SeqIO desde biopython
from Bio import SeqIO

#Para cada nombre en la lista de archivos
for file in file_list:
    #eliminamos la terminación .gb
    file_name=file.replace('.gb','')
    #convertimos cada archivo en un archivo FASTA
    SeqIO.convert("genomes_GenBank"'/'+file, "genbank",
fasta_sequences+'/'+file_name+".fasta", "fasta")
```

Se unifican las secuencias en un único archivo FASTA mediante el comando del entorno Windows:

C:\Users\Gonzalo\TFM\fasta_sequences\ copy *.fasta merged.fasta

De este modo, se obtiene el archivo 'merged.fasta', que contiene todos los genomas analizados en formato FASTA. La base de datos se genera mediante el siguiente comando mínimo:

C:\Users\Gonzalo\TFM\fasta_sequences\ makeblastdb –in merged.fasta –dbtype nucl –parse_seqids

Las secuencias para el alineamiento se encuentran en el archivo 'TFBSs.fasta' y corresponden a las secuencias de unión a factores de transcripción, y sus regiones colindantes, del genoma de referencia *E.coli* K-12 MG1655. La base de datos sobre la que se realizan los alineamientos, 'merged.fasta', contiene todos los genomas completos de las cepas de *E.coli*. El último paso es realizar los alineamientos:

C:\Users\Gonzalo\TFM\fasta_sequences\ blastn –db merged.fasta –query TFBSs.fasta –outfmt 5 –out blast_output.xml

6.5 Filtrado de alineamientos

```
#Importamos SearchIO desde biopython
from Bio import SearchIO
```

```
#Cargamos los alineamientos resultantes de BLASTN
qresults = SearchIO.parse('fasta_sequences/blast_output.xml', 'blast-
xml')

#Filtramos los alineamientos siguiendo dos criterios

#Creamos el diccionario vacío donde irán los alineamientos
filtered_dicc={}

#Establecemos el criterio del E-value
evalue_filter = lambda hsp: hsp.evalue < 1e-10

#Establecemos el criterio de la cobertura
coverage_filter = lambda hsp: (hsp.aln_span/hsp.query_span)*100 > 70

#Para cada alineamiento
for qresult in qresults:
    #filtramos según el criterio de E-value
    filtered_qresult=qresult.hsp_filter(evalue_filter)
    #filtramos según el criterio de covertura
    filtered_qresults = filtered_qresult.hsp_filter(coverage_filter)
    #guardamos los alineamientos filtrados en el diccionario
    filtered_dicc[filtered_qresults.id] = filtered_qresults
```

6.6 Conteo de sustituciones

```
#Para el conteo de las sustituciones

#Creamos un diccionario vacio que contendrá la información de las
sustituciones
mutations={}

#Para cada acceso del diccionario de alineamientos
for f in filtered_dicc.keys():
    #para cada Hit del diccionario
    for e in filtered_dicc[f]:
        #para cada alineamiento de cada Hit
        for t in e:
        #ponemos los contadores a cero
            TFBSs=0
            External=0
            Total=0
            j=0
            p=0
            v=0
    #la longitud de la secuencia de unión a factores de transcripción
    #es la longitud total del query - 500 (±250 por cada lado)
            n=t.query_span-500
#para los nucleótidos de la secuencia a alinear (query) y la alineada
(hit)
            for i,r in zip(t.query.seq, t.hit.seq):
```

```
    #la longitud de la secuencia adyacente por la izquierda
    #es 250 menos el lugar de inicio del alineamiento
                l=(250-t.query_range[0])
                #sin contar los espacios
                if i !='-':
                #comenzamos el contador de nucleótidos general
                    j=j+1
             #si el contador se encuentra en la posición de la secuencia
             #de unión a factores de transcripción
                if j>l and j<(l+n):
                    #y no hay un espacio
                    if i !='-':
              #comenzamos el contador de nucleótidos de
              #la secuencia de unión a factores de transcripción
                        p=p+1
            #si los nucleótidos de las secuencias alineadas no coinciden
                    if i!=r and r!='-' and i!='-':
            #sumamos uno al conteo de sustituciones de
            #la secuencia de unión a factores de transcripción
                        TFBSs+=1
            #y al conteo general de sustituciones
                        Total+=1
#si el contador se encuentra en la posición de las secuencias
adyacentes
                else:
            #sin contar los espacios
                    if i !='-':
            #comenzamos el contador de nucleótidos de las secuencias
            #adyacentes
                        v=v+1
            #si los nucleótidos de las secuencias alineadas no coinciden
                    if i!=r and r!='-' and i!='-':
            #sumamos uno al conteo de sustituciones de las secuencias
            #adyacentes
                        External+=1
            #y al conteo general de sustituciones
                        Total+=1

              #si el conteo de sustituciones no es cero
              #calculamos las tasas de sustitución dividiendo
              #el conteo de sustituciones entre el número de nucleótidos
              #en el que se han contado
              if TFBSs!=0:
                  TFBSs_n=TFBSs/p
              else:
                  TFBSs_n=0
              if External!=0:
                  External_n=External/v
              else:
                  External_n=0
              if Total!=0:
                  Total_n=Total/(v+p)
              else:
```

```
                Total_n=0

            #guardamos los datos en el diccionario previamente creado
            mutations[f]=(TFBSs, TFBSs_n, External, External_n, Total,
Total_n )

#Guardamos el diccionario como un archivo JSON
import json
with open('dictt_final.json', 'w') as outfile:
    json.dump(mutations, outfile, indent=4, sort_keys=True,
separators=(',', ':'))



#Para el conteo de las sustituciones en alineamientos con secuencias
de unión a factores de transcripción activadoras


#Creamos un diccionario vacio que contendrá la información de las
sustituciones
mutationsA={}

#Para cada acceso del diccionario de alineamientos
for f in filtered_dicc.keys():
    #para los códigos del diccionario
    for r in a:
   #empleamos el código si se corresponde con una secuencia activadora
        if r['Gene expression effect caused by the TF bound to the TF-
bs ']=='+':
            if f==r["TF binding site (TF-bs) identifier assigned by
RegulonDB "]:
                #para cada Hit del diccionario
                for e in filtered_dicc[f]:
                    #para cada alineamiento de cada Hit
                    for t in e:
                        #ponemos los contadores a cero
                        TFBSs=0
                        External=0
                        Total=0
                        j=0
                        p=0
                        v=0
    #la longitud de la secuencia de unión a factores de transcripción
    #es la longitud total del query - 500 (±250 por cada lado)
                        n=t.query_span-500
#para los nucleótidos de la secuencia a alinear (query) y la alineada
(hit)
                        for i,r in zip(t.query.seq, t.hit.seq):
    #la longitud de la secuencia adyacente por la izquierda
    #es 250 menos el lugar de inicio del alineamiento
                            l=(250-t.query_range[0])
    #sin contar los espacios
                            if i !='-':
    #comenzamos el contador de nucleótidos general
                                j=j+1
```

```
            #si el contador se encuentra en la posición de la secuencia
            #de unión a factores de transcripción
                            if j>l and j<(l+n):
            #y no hay un espacio
                                if i !='-':
          #comenzamos el contador de nucleótidos de
          #la secuencia de unión a factores de transcripción
                                    p=p+1
          #si los nucleótidos de las secuencias alineadas no coinciden
                                if i!=r and r!='-' and i!='-':
          #sumamos uno al conteo de sustituciones de
          #la secuencia de unión a factores de transcripción
                                    TFBSs+=1
          #y al conteo general de sustituciones
                                    Total+=1
#si el contador se encuentra en la posición de las secuencias
adyacentes
                            else:
          #sin contar los espacios
                                if i !='-':
          #comenzamos el contador de nucleótidos de las secuencias
          #adyacentes
                                    v=v+1
          #si los nucleótidos de las secuencias alineadas no coinciden
                                if i!=r and r!='-' and i!='-':
          #sumamos uno al conteo de sustituciones de las secuencias
          #adyacentes
                                    External+=1
          #y al conteo general de sustituciones
                                    Total+=1

          #si el conteo de sustituciones no es cero
          #calculamos las tasas de sustitución dividiendo
          #el conteo de sustituciones entre el número de nucleótidos
          #en el que se han contado
                        if TFBSs!=0:
                            TFBSs_n=TFBSs/p
                        else:
                            TFBSs_n=0
                        if External!=0:
                            External_n=External/v
                        else:
                            External_n=0
                        if Total!=0:
                            Total_n=Total/(v+p)
                        else:
                            Total_n=0

          #guardamos los datos en el diccionario previamente creado
                        mutationsA[f]=(TFBSs, TFBSs_n, External,
External_n, Total, Total_n )
```

```
#Guardamos el diccionario como un archivo JSON
import json
with open('dic+_final.json', 'w') as outfile:
    json.dump(mutationsA, outfile, indent=4, sort_keys=True,
separators=(',', ':'))



#Para el conteo de las sustituciones en alineamientos con secuencias
de unión a factores de transcripción represoras

#Creamos un diccionario vacio que contendrá la información de las
sustituciones
mutationsR={}

#Para cada acceso del diccionario de alineamientos
for f in filtered_dicc.keys():
    #para los códigos del diccionario
    for r in a:
    #empleamos el código si se corresponde con una secuencia represora
        if r['Gene expression effect caused by the TF bound to the TF-
bs ']=='-':
            if f==r["TF binding site (TF-bs) identifier assigned by
RegulonDB "]:
                #para cada Hit del diccionario
                for e in filtered_dicc[f]:
                    #para cada alineamiento de cada Hit
                    for t in e:
                        #ponemos los contadores a cero
                        TFBSs=0
                        External=0
                        Total=0
                        j=0
                        p=0
                        v=0
     #la longitud de la secuencia de unión a factores de transcripción
     #es la longitud total del query - 500 (±250 por cada lado)
                        n=t.query_span-500
#para los nucleótidos de la secuencia a alinear (query) y la alineada
(hit)
                        for i,r in zip(t.query.seq, t.hit.seq):
           #la longitud de la secuencia adyacente por la izquierda
           #es 250 menos el lugar de inicio del alineamiento
                            l=(250-t.query_range[0])
           #sin contar los espacios
                            if i !='-':
           #comenzamos el contador de nucleótidos general
                                j=j+1
           #si el contador se encuentra en la posición de la secuencia
           #de unión a factores de transcripción
                            if j>l and j<(l+n):
           #y no hay un espacio
                                if i !='-':
           #comenzamos el contador de nucleótidos de
```

```
          #la secuencia de unión a factores de transcripción
                                    p=p+1
          #si los nucleótidos de las secuencias alineadas no coinciden
                                if i!=r and r!='-' and i!='-':
          #sumamos uno al conteo de sustituciones de
          #la secuencia de unión a factores de transcripción
                                    TFBSs+=1
          #y al conteo general de sustituciones
                                    Total+=1
#si el contador se encuentra en la posición de las secuencias
adyacentes
                               else:
          #sin contar los espacios
                                if i !='-':
          #comenzamos el contador de nucleótidos de las secuencias
          #adyacentes
                                    v=v+1
          #si los nucleótidos de las secuencias alineadas no coinciden
                                if i!=r and r!='-' and i!='-':
          #sumamos uno al conteo de sustituciones de las secuencias
          #adyacentes
                                    External+=1
          #y al conteo general de sustituciones
                                    Total+=1

          #si el conteo de sustituciones no es cero
          #calculamos las tasas de sustitución dividiendo
          #el conteo de sustituciones entre el número de nucleótidos
          #en el que se han contado
                        if TFBSs!=0:
                            TFBSs_n=TFBSs/p
                        else:
                            TFBSs_n=0
                        if External!=0:
                            External_n=External/v
                        else:
                            External_n=0
                        if Total!=0:
                            Total_n=Total/(v+p)
                        else:
                            Total_n=0

           #guardamos los datos en el diccionario previamente creado
                        mutationsR[f]=(TFBSs, TFBSs_n, External,
External_n, Total, Total_n )


#Guardamos el diccionario como un archivo JSON
import json
with open('dic-_final.json', 'w') as outfile:
    json.dump(mutationsR, outfile, indent=4, sort_keys=True,
separators=(',', ':'))
```

6.7 Análisis estadístico

```
#Importamos la librería jsonlite
library(jsonlite)

#Cargamos las tasas de sustitución para todos los alineamientos
mutations <- fromJSON("dictt_final.json")

#Ajustamos el formato
mutations<-data.frame(mutations)
mutations<-t(mutations)
mutations<-data.frame(mutations)

#Calculamos las medias y medianas para los sitios de unión
#a factores de transcripción
mean = mean(mutations$X2)
median= median(mutations$X2)


#Calculamos las medias y medianas para las secuencias adyacentes
mean1 = mean(mutations$X4)
median1 = median(mutations$X4)

#Disponemos los gráficos en un mismo output
attach(mtcars)
layout(matrix(c(1,2), 1, 2, byrow = TRUE))

#Generamos el primer histograma
hist(mutations$X2,breaks = 60,main = "TFBSs",xlab = "Tasas de
sustitución",ylab = "Frecuencia",col = "black",border = "green")

#Añadimos las lineas de la media y la mediana
abline(v=mean, col="red")
abline(v=median, col="black")

#Añadimos la leyenda
legend(x = "topright",
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Generamos el segundo histograma
hist(mutations$X4,breaks = 60,main = "Secuencias adyacentes" ,xlab =
"Tasas de sustitución",ylab = "Frecuencia",col = "black", border =
"green")

#Añadimos las lineas de la media y la mediana
lines(density(mutations$X4),col="brown")
abline(v=mean1, col="red")
abline(v=median1, col="black")

#Añadimos la leyenda
legend(x = "topright",
```

```
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Realizamos el test de Mann-Whitney
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE,
alternative = "less")

-------------------------------------

#Cargamos las tasas de sustitución para los alineamientos
#con secuencias de unión a factores de transcripción activadoras
mutations <- fromJSON("dic+_final.json")

#Ajustamos el formato
mutations<-data.frame(mutations)
mutations<-t(mutations)
mutations<-data.frame(mutations)

#Calculamos las medias y medianas para los sitios de unión
#a factores de transcripción
mean = mean(mutations$X2)
median= median(mutations$X2)

#Calculamos las medias y medianas para las secuencias adyacentes
mean1 = mean(mutations$X4)
median1 = median(mutations$X4)

#Disponemos los gráficos en un mismo output
attach(mtcars)

layout(matrix(c(1,2), 1, 2, byrow = TRUE))

#Generamos el primer histograma
hist(mutations$X2,breaks = 60,main = "TFBSs",xlab = "Tasas de
sustitución",ylab = "Frecuencia",col = "black", border = "green")

#Añadimos las lineas de la media y la mediana
lines(density(mutations$X2),col="brown")
abline(v=mean, col="red")
abline(v=median, col="black")

#Añadimos la leyenda
legend(x = "topright",
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Generamos el segundo histograma
hist(mutations$X4,breaks = 60,main = "Secuencias adyacentes",xlab =
"Tasas de sustitución",ylab = "Frecuencia",col = "black", border =
"green")

#Añadimos las lineas de la media y la mediana
lines(density(mutations$X4),col="brown")
```

```
abline(v=mean1, col="red")
abline(v=median1, col="black")

#Añadimos la leyenda
legend(x = "topright",
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Realizamos el test de Mann-Whitney
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE,
alternative = "less")

---------------------------------------

#Cargamos las tasas de sustitución para los alineamientos
#con secuencias de unión a factores de transcripción activadoras
mutations <- fromJSON("dic-_final.json")

#Ajustamos el formato
mutations<-data.frame(mutations)
mutations<-t(mutations)
mutations<-data.frame(mutations)

#Calculamos las medias y medianas para los sitios de unión
#a factores de transcripción
mean = mean(mutations$X2)
median= median(mutations$X2)

#Calculamos las medias y medianas para las secuencias adyacentes
mean1 = mean(mutations$X4)
median1 = median(mutations$X4)

#Disponemos los gráficos en un mismo output
attach(mtcars)

layout(matrix(c(1,2), 1, 2, byrow = TRUE))

#Generamos el primer histograma
hist(mutations$X2,breaks = 60,main = "TFBSs",xlab = "Tasas de
sustitución",ylab = "Frecuencia",col = "black", border = "green")

#Añadimos las lineas de la media y la mediana
lines(density(mutations$X2),col="brown")
abline(v=mean, col="red")
abline(v=median, col="black")

#Añadimos la leyenda
legend(x = "topright",
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Generamos el segundo histograma
hist(mutations$X4,breaks = 60,main = "Secuencias adyacentes",xlab =
```

```
"Tasas de sustitución",ylab = "Frecuencia",col = "black", border =
"green")

#Añadimos las lineas de la media y la mediana
lines(density(mutations$X4),col="brown")
abline(v=mean1, col="red")
abline(v=median1, col="black")

#Añadimos la leyenda
legend(x = "topright",
 c("Densidad",'Media', "Mediana"),
 col = c("brown", "red", "black"),
 lwd = c(2, 2, 2))

#Realizamos el test de Mann-Whitney
wilcox.test(mutations$X2,mutations$X4,data=mutations,paired=TRUE,
alternative = "less")
```