BacDupWeb: Desarrollo de una aplicación web para la visualización dinámica de duplicaciones génicas en bacterias

Máster Universitario en Bioinformática y Bioestadística
Biología Molecular y Genética

Autor: **Daniel Ibáñez Maldonado**
Profesor colaborador: **José Francisco Sánchez Herrero**
Responsable Asignatura: **Laura Calvet Liñán**

2 de Junio de 2022

Agraïments

A Alba Moya por haberme tendido una alfombra roja en mi parte del desarrollo y a Jose por su paciencia en intentar mantenerme dentro de los parámetros de un buen científico y no de un tertuliano de bar. Gracias por tu seguimiento en todo lo necesario a pesar de torturarte por slack entre niños y traslado.

A la Marta, el Carles, l'Alba, el Javier, el Dani i la Marianne per aguantar les meves teories sobre el canvi climàtic pels Pirineus i per l'Empordà. Ja aviso que després d'aquest TFM en faré un altre sobre el CO2.

Al Joan, l'Enric, l'Oriol i l'Albert, per compartir sopars, excursions, passejades, partits de pàdel i converses íntimes encara que alguna se'ns escapi de les mans. Especial agraïment per tot el que m'ha ensenyat el número 23 del món i la seva ajuda per entendre Darwin. Prometo aviat un Bages. Ara sí. I fins i tot un Arenys.

Als amics de Santa Margarita per ensenyar-me a no saber discutir amb ordre i sense cridar. A la Maria Gràcia, l'Olívia i el Miquel Àngel per ser capaços de discutir absolutament de tot. I és que si no s'ha descobert, no existeix, oi? Un plaer compartir Prenafeta, Talaixà, Torre de la Mora i Cabo de Gata. Bé, Cabo de Gata no. Si ho penso podria fer un TFM sobre les nostres trobades i sortides. No s'acabaria mai!!!!

Al Dalmau i la Montse per preguntar sempre, què? Com vas? I què és el que fas exactament. Estàs estudiant? Has acabat? I allà han estat, fent un plat de pasta, un arròs, tenint cura de les tomaqueres o fent una podrida amb els néts. Montse, mil gràcies per ajudar-nos tant amb els fills, especialment amb el Martí quan era mooolt petit. I mira ara, aviat 22!!!

Als meus germans que estimo tant, David, Ferran i Cristina per fer fàcils les relacions familiars i saber-nos estimar amb les nostres diferències, tot i compartir feina amb dos i al Josep, la Marta i la Pilar per les sobretaules tant agradables a l'Empordà, Cabrils o el Montseny. I els nebots, la Clàudia, la Júlia, el Guillem i la Lara, i també l'Adrià, la Clara, la Laia i l'Anna. Ara que he acabat el Màster, Tiana no pot esperar. Una especial menció al David per tocar la pera amb Darwin, la genètica, la intel·ligència artificial i les dentrites. En bona part escric aquí per culpa teva.

Als meus pares per tot el seu amor i suport incondicional. Jo també us estimo moltíssim. Gràcies mama per fer-ho tot tant fàcil, no ja només ara, si no tota la vida, per cuidar dels nets quan ha calgut i pel fricandó i les truites de patates. I gràcies papa per compartir tot amb nosaltres i per transmetre la permanent

curiositat per tot, des dels 747, els ovnis o el Zilog Z80, fins els arduinos i Joana la Boja. No crec que arribi ni a historiador ni a tocar la bateria, però el camí està traçat per si m'animo.

Hana, Pau i Martí, us estimo molt. Algunes coses no han estat fàcils però ens n'hem sortit i ara ve el millor. M'heu acompanyat en aquest màster gairebé cada dia sense saber ben bé per què ho feia i aguantant les meves teories i la meva cara de broma. Finalment, gràcies al màster, he vist i he entès, que els gossos són moooolt intel·ligents i això és el més important. La Mandi especialment. I un record per la Greta.

I hola Laura. Moltes gràcies per ser al meu costat, per acompanyar-me en aquest camí, per sortir menys, per moltes nits adormint-te sola al sofà mentre jo anava al límit amb una PEC, com avui. Gràcies per aprendre amb mi sobre zigots, gens, rna missatger i proteïnes. Gràcies també per portar aquests dos pipiolos i aquesta pipiola més enllà dels 18 anys. Crec que encara els tindrem un temps per aquí però... bona feina! Ara ens toca a nosaltres. T'estimo.

Ficha del trabajo final

Título del TFM:	**BacDupWeb:** Desarrollo de una aplicación web para la visualización dinámica de duplicaciones génicas en bacterias.
Autor:	Daniel Ibáñez Maldonado
Consultor/a:	José Francisco Sánchez Herrero
PRA:	Laura Calvet Liñán
Fecha entrega:	2 de Junio de 2022
Titulación:	Máster de Bioinformática y Bioestadística
Área del Trabajo Final:	Biología Molecular y Genética
Idioma del trabajo:	Español
Número de créditos:	15
Palabras clave	*bacterias, Shiny, aplicación web, duplicación génica, infección nosocomial*

Índice

1 Introducción ---------- 3
1.1 Contexto y justificación del Trabajo ---------- 3
1.1.1 Contexto biológico de partida 3
1.1.2 El proyecto BacDup 4
1.1.3 BacDupWeb como continuación del proyecto BacDup 6
1.1.4 Justificación del trabajo. Porqué BacDupWeb 9
1.2 Objetivos del Trabajo ---------- 11
1.2.1 Objetivos previos al desarrollo: 11
1.2.2 Objetivos de ejecución del desarrollo 11
1.2.3 Objetivos de entrega 11
1.3 Enfoque y método seguido ---------- 12
1.4 Planificación del Trabajo ---------- 12
1.4.1 Planificación: Tareas previas 12
1.4.2 Planificación: Desarrollo Fase 1 13
1.4.3 Planificación: Desarrollo Fase 2 13
1.4.4 Planificación: Entregas 13
1.5 Breve sumario de contribuciones y productos obtenidos ---------- 14
1.6 Breve descripción de los otros capítulos de la memoria ---------- 15
2 Metodología ---------- 16
2.1 Entorno tecnológico de trabajo ---------- 16
2.2 Estudio previo de los archivos de datos a visualizar ---------- 17
2.3 Elección de los lenguajes de desarrollo ---------- 19
2.4 Breve descripción de Shiny ---------- 21
2.5 Relación de inputs de la aplicación ---------- 22
2.6 Elementos a mostrar como resultados ---------- 23
2.7 Diagrama de bloques de BacDupWeb ---------- 24
3 Resultados ---------- 25
3.1 Descripción general de la aplicación ---------- 25
3.2 Relación de apartados y subapartados de la aplicación: ---------- 25
3.3 Descripción detallada de los apartados desarrollados ---------- 26
3.3.1 Bloque selector: Carga de archivos 26
3.3.2 Bloque selector: Carga de filtros 27
3.3.3 Bloque selector: Carga de campos de visualización 28
3.3.4 Bloque Resultados: Datos generales y funciones básicas 28
3.3.5 Bloque Resultados: Tabla dinámica de genes duplicados 29
3.3.6 Bloque Resultados: Gráfico Circos de genes duplicados 30
3.3.7 Bloque Resultados: Tabla taxonómica y de cromosomas 32

3.3.8 Bloque Resultados: Frecuencia de genes duplicados 34
3.3.9 Bloque in/out: Funciones de intercambio de datos 35
3.4 Puesta en producción de BacDupWeb ---------- 36
4 Discusión ---------- 37
4.1 Reflexión sobre BacDup y BacDupWeb ---------- 37
4.2 Velocidad reactiva y estética de Shiny ---------- 38
4.3 Líneas de futuro ---------- 38
5 Valoración económica ---------- 39
6 Conclusiones ---------- 40
7 Glosario ---------- 42
8 Bibliografía ---------- 44
9 Anexo: Repositorio GitHub ---------- 46

Lista de figuras

Figura 1: Workflow de BacDup 5
Figura 2:Biocircos de Enterococcus Faecalis V583 desde BacDup..... 6
Figura 3: Cabecera de la aplicación web 7
Figura 4: Punto de partida de BacDupWeb 8
Figura 5: Cronograma del desarrollo de BacDupWeb 14
Figura 6: : Ejemplo de archivo de duplicados dup_Annot 17
Figura 7: Detalle de campos del archivo dup_annot 18
Figura 8: Archivo de secuencias con las longitudes en bp (length.csv) 18
Figura 9: detalle de archivo info.csv..... 19
Figura 10: Ejemplo básico e inicial del paquete Shiny 21
Figura 11: Diagrama de bloques de BacDupWeb..... 24
Figura 12: BacDupWeb carga archivos 26
Figura 13:BacDupWeb filtros 27
Figura 14: BacDupWeb selector de campos..... 28
Figura 15: BacDupWeb: tablas de totales y datos filtrados..... 29
Figura 16: Tabla principal de BacDupWeb 30
Figura 17: Gráfico Circos dinámico..... 31
Figura 18: Biocircos de genes con 7 copias 32
Figura 19: BacDupWeb apartado datos cepa 34
Figura 20: BacDupWeb frecuencia del número de duplicados 34
Figura 21: link a ncbi RefSeq / GenBank 35
Figura 22: Tabla de precios del servidor Shiny Apps en junio de 2022 39

Resumen

Conocer la biología y la genética de las bacterias y otros patógenos, es de vital importancia para conocer mejor los mecanismos que influyen en el desarrollo de las infecciones en humanos. En concreto, son de una gran preocupación de salud pública, las infecciones bacterianas en entornos hospitalarios, infecciones que, en muchos casos, presentan gran virulencia. Uno de estos trabajos es el proyecto **BacDup** que centra sus esfuerzos en realizar un mapa preciso de las duplicaciones génicas a partir de las secuencias originales de cepas bacterianas, ya fueran obtenidas *de novo* u obtenidas de repositorios como **RefSeq** o **GenBank**. Disponer de un mapa de duplicaciones se considera de gran interés para ayudar a encontrar mecanismos de defensa a esta virulencia. El proyecto **BacDupWeb** que se presenta en esta memoria, es el complemento de **BacDup** que permite la visualización, manipulación, filtrado y estudio de los datos de duplicaciones. **BacDupWeb** ofrece la necesaria facilidad y velocidad de acceso a toda esta información para agilizar las posibles respuestas a estas infecciones.

Abstract

Knowing the biology and genetics of bacteria and other pathogens is of vital importance to better understand the mechanisms that influence the development of infections in humans. Specifically, bacterial infections in hospital environments are of great public health concern, infections that, in many cases, are highly virulent. One of these works is the **BacDup** project, which focuses its efforts on making a precise map of gene duplications from the original sequences of bacterial strains, whether they were obtained de novo or obtained from repositories such as **RefSeq** or **GenBank**. Having a map of duplications is considered of great interest to help find defense mechanisms against this virulence. The **BacDupWeb** project presented in this report is the **BacDup** complement that allows the visualization, manipulation, filtering and study of duplication data. **BacDupWeb** offers the necessary easy and speed of access to all this information to speed up possible responses to these infections.

1 Introducción

1.1 Contexto y justificación del Trabajo

1.1.1 Contexto biológico de partida

De todas las preocupaciones que se generan alrededor de la salud pública, una de las que más creciente interés suscitan es la de las infecciones de origen nosocomial. En efecto, una de las causas que generan más complicaciones y muertes, es la de las bacterias en entornos hospitalarios, cada vez más resistentes a los antibióticos y tratamientos habituales. Este problema ocupa, en muchas ocasiones, camas y recursos hospitalarios durante más días de los causados por la propia enfermedad de ingreso y un coste económico de importante magnitud [1]–[3] .

Muchas de las complicaciones graves causadas por infecciones hospitalarias son causadas por bacterias especialmente activas, virulentas y resistentes a la mayoría de antibióticos. Se ha publicado un estudio muy reciente y exhaustivo sobre los patógenos presentes en mas de 200 países [4]. Se trata de patógenos que aprovechan la situación de debilidad de los pacientes, ya sea porque son pacientes inmunodeprimidos por los propios tratamientos, ya sea porque su sistema inmunitario no funciona a pleno rendimiento por su patología de ingreso. Cuando estos patógenos son especialmente virulentos, la agresividad y la velocidad de expansión hacen imposibles los esfuerzos de los equipos médicos que no tienen tiempo suficiente para analizar el patógeno concreto y su posible tratamiento.

Una de las vías de lucha contra estos patógenos es el diseño de fármacos y terapias que tienen sus principios activos basados en características genéticas y proteínas sintetizadas. Y es que cada vez se buscan mas las respuestas del funcionamiento de las infecciones en la estructura del genoma, siendo uno de los mecanismos posibles, el de las duplicaciones génicas, es decir genes que presentan múltiples copias y que podrían ser el origen de una mayor virulencia en las infecciones [5]–[8]. Disponer de mayor conocimiento de estas duplicidades génicas podría aportar nuevas líneas de investigación y elementos de estudio para una futura fabricación de fármacos para combatirlas.

En este contexto biológico, este TFM versa sobre cómo ofrecer al investigador, herramientas útiles para acceder a los datos genómicos de estas bacterias. Dada la virulencia en muchas de estas infecciones, es de gran interés que las herramientas, como la que se presenta en esta memoria, sean ágiles, intuitivas y de fácil acceso. En este caso se trata de acceder rápidamente a los datos de genes duplicados y a los productos codificados.

1.1.2 El proyecto BacDup

Con el objetivo de mejorar el conocimiento de estas infecciones de origen nosocomial, en 2016 se inició un trabajo de varios profesionales del ámbito de la salud, eso es, microbiólogos, bioinformáticos, biólogos, sobre algunas características de estas bacterias. En concreto este trabajo consistió en localizar en el genoma bacteriano, aquellas proteínas que podían estar sintetizándose a partir de genes distintos, genes duplicados o con varias copias en zonas distintas del genoma. Este trabajo desembocó en dos publicaciones que se pueden encontrar citadas al final del documento [9]–[11]

Aunque es de gran interés el contenido y las conclusiones biológicas de dichos trabajos, en este apartado no se van a considerar con detalle las conclusiones si no los procesos informáticos que permitieron llegar a dichos resultados. En concreto estamos hablando de la parte del estudio consistente en la lectura y filtrado informático de datos. Un equipo formado por José Francisco Sánchez Herrero y Alba Moya Garcés consiguió crear un pipeline que partía de los archivos Fasta secuenciados para terminar localizando los genes duplicados y visualizándolos en un gráfico en forma de genoma circular. Este pipeline utiliza llamadas a la herramienta **Blast** [12]para realizar comparaciones y alineamiento de secuencias con el objetivo de conseguir encontrar duplicados o genes con *n* copias, para finalmente usar el paquete Circos de R para su visualización en formato circular y con varias capas concéntricas de datos.

Fue el trabajo de final de máster de Alba Moya Garcés que llevó todo este proceso a un pipeline cuyo *workflow* se muestra en la figura 1.

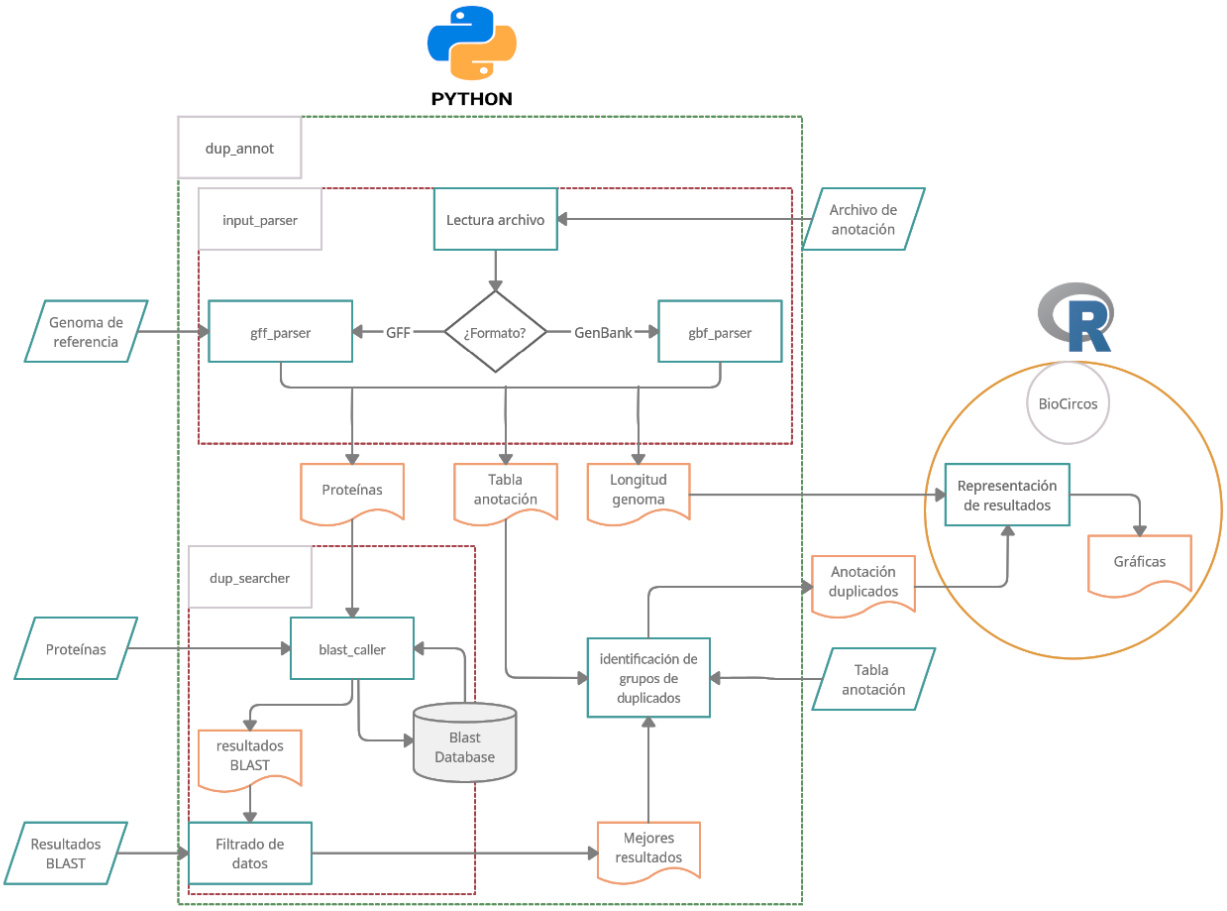


Figura 1: Workflow de BacDup

Toda esta tarea informática se puede localizar en un repositorio de GitHub citado en la bibliografía [13], [14] y desemboca en la representación gráfica del genoma del patógeno y de los genes duplicados usando el paquete Biocircos. En la figura 2 se muestra como ejemplo el gráfico Biocircos generado por **BacDup** del microorganismo ***Enterococcus faecalis V583 (firmicutes)*** que tiene como referencia el código **GCF_000007785.1** en **RefSeq**[15].

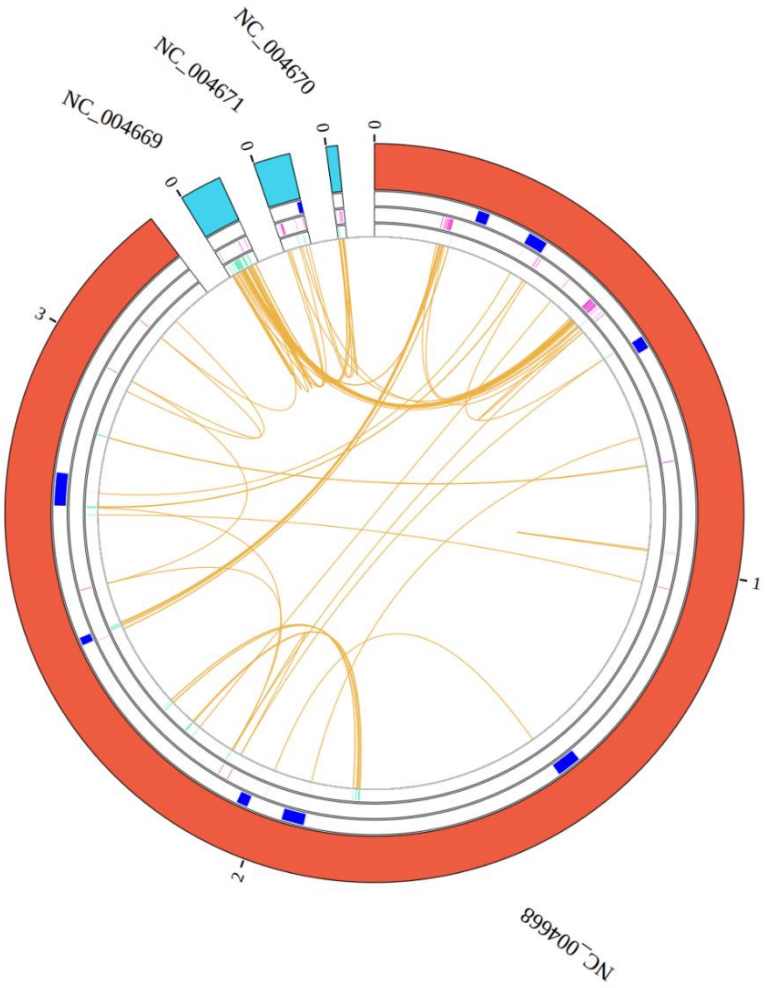


Figura 2:Biocircos de Enterococcus Faecalis V583 desde BacDup.

El arco rojo es el cromosoma principal de la bacteria, mientras los tramos azul claro son los plásmidos. Las líneas naranjas delgadas unen los grupos de genes con múltiples copias según su locus en el genoma. En arcos concéntricos después del genoma, aparecen los pseudogenes y los genes según se ubican en hebra positiva o negativa.

1.1.3 BacDupWeb como continuación del proyecto BacDup

BacDup es un proyecto de gran alcance, con un pipeline capaz de generar un mapa de duplicados pero que tiene una limitación, y es que se requiere crear un informe cada vez que queremos acceder a una cepa distinta o si deseamos filtrar los datos de un genoma por algún concepto como por ejemplo que incluya o no pseudogenes. Es evidente que **BacDup** es muy potente pero también es cierto que tiene sus limitaciones en cuanto a la agilidad de acceso a los datos y de manipulación de los mismos al final del proceso.

En la propia lectura del TFM de Alba Moya [13] ya se apuntaba a la mejora de esta parte de visualización e interpretación de los datos como un necesario complemento final a **BacDup**. Es este pues, el origen de **BacDupWeb**, como la herramienta necesaria para cerrar el círculo del análisis de duplicidades. En un primer momento se pensaba en una continuación del pipeline, pero aunque se obtuvieran resultados interesantes, se estaría dentro del mismo procedimiento cerrado y poco accesible por parte del investigador. Fue aquí donde surgió la idea de crear una aplicación web, fácil de usar, ágil e intuitiva que permitiera acceder a dichos datos, como utilidad independiente del proceso de anotación. Y de esa primera idea nació **BacDupWeb**, una herramienta que ofrece la parte final visual al pipeline de **BacDup** pero desde un entorno distinto e independiente.

Figura 3: Cabecera de la aplicación web

BacDupWeb nace pues, con la idea inicial de mostrar los datos de los genes duplicados de forma ágil pudiendo modificar determinadas características actuando como filtro para seleccionar los datos que le interesen potencialmente al investigador sin tener que generar un informe cada vez con datos distintos. De este modo, y como se puede comprobar posteriormente en los resultados, dada una cepa secuenciada *de novo* o ya conocida, y pasada por **BacDup** para obtener el archivo de genes duplicados, se pueden visualizar todos los productos duplicados y filtrar por una gran variedad de campos, clasificar por cualquier campo y configurar la tabla y los gráficos según nuestro estudio.

Nótese que, mientras el proceso de generación de los archivos de anotación es un único proceso, con el que se obtienen los archivos de análisis, **BacDupWeb** ofrece la posibilidad de infinitos estudios distintos sobre dichos datos. Por lo tanto no debe considerarse un proceso lineal a partir de **BacDup**, si no un punto de partida para cualquier estudio sobre cualquier cepa de la que se hayan obtenido los datos de duplicidades. Para tener una visión más clara del punto de partida, en la Figura 4 se muestra dónde se sitúa **BacDupWeb** partiendo del workflow de **BacDup**. Más adelante, en el apartado de resultados se explica con todo detalle las funcionalidades que conforman esta aplicación web.

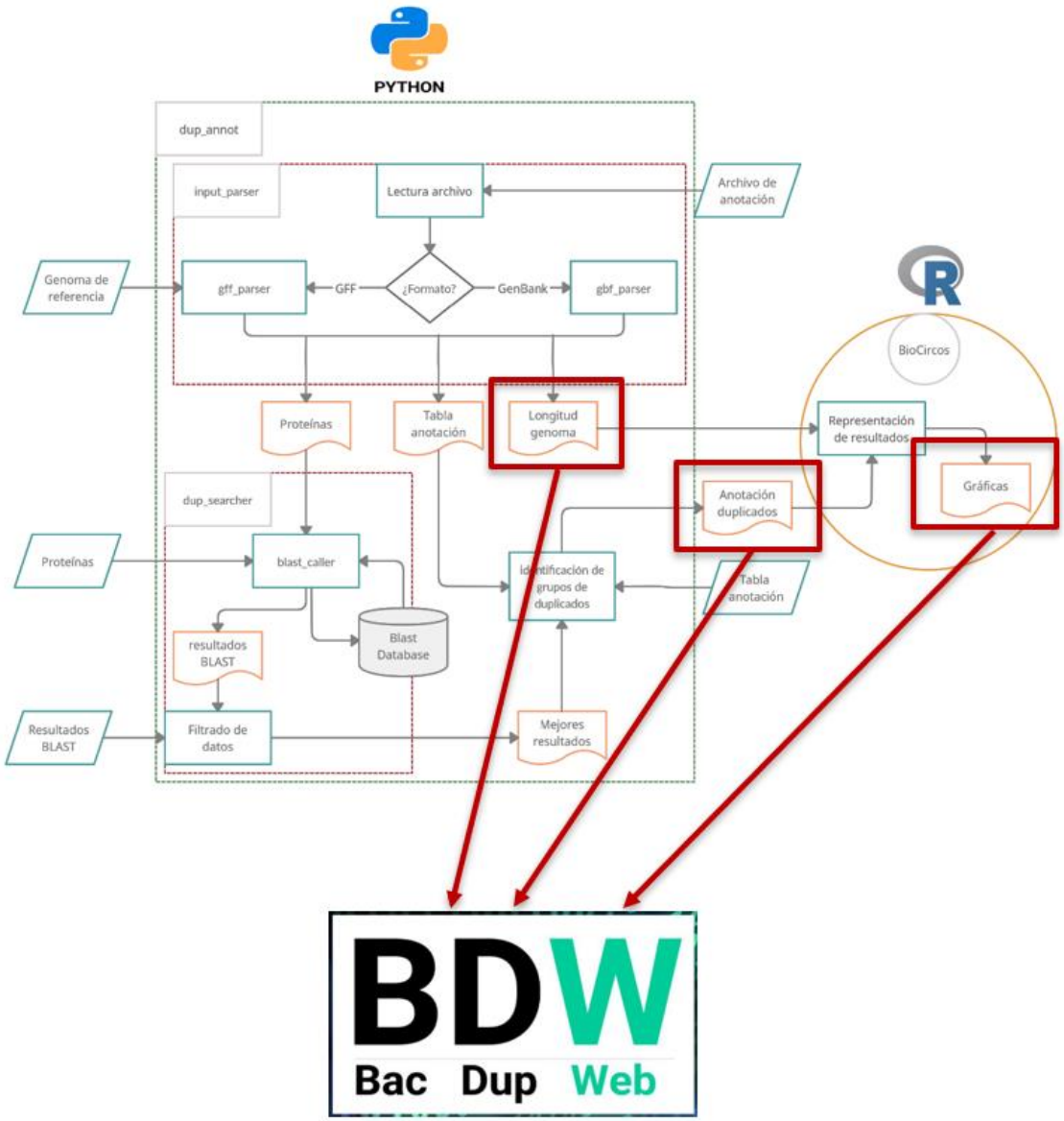


Figura 4: Punto de partida de BacDupWeb

En definitiva, la nueva herramienta informática **BacDupWeb** pretende agilizar la consulta de los datos por parte del investigador, haciendo mucho más práctico el trabajo de carga de archivos, filtrado y visualización.

1.1.4 Justificación del trabajo. Porqué BacDupWeb.

En biología y bioinformática hay un formato de estudios en el que uno debe definir una serie de funciones, unos archivos de origen y unos parámetros, y a partir de ahí se genera un informe de resultados en pdf, Markdown o html. Estos informes pueden ser muy completos e interesantes pero en determinados casos tienen un problema y es que siguen un pipeline muy rígido que tiene un inicio y un final. Si el estudio es muy concreto y los datos y parámetros son únicos, se genera el informe y puede considerarse una conclusión final válida.

El problema surge cuando el estudio en sí requiere ir realizando pruebas o consultas con distintos datos o variando filtros o parámetros. Es en esa situación que, tener que pasar cada vez por el mismo proceso ejecutando todo el pipeline, puede ser complejo y poco práctico. En este sentido, la visualización de resultados con posibilidad de cargar fácilmente archivos distintos de salida y manipular variables, lleva a la necesidad de soluciones distintas, más dinámicas y es aquí donde las aplicaciones web son ideales.

En líneas generales el trabajo con un pipeline clásico tiene las siguientes características:

- Hay que instalar aplicaciones en local en los ordenadores personales.
- La velocidad de los resultados dependen del ordenador local.
- Para cada archivo y para cada valor de parámetros, debes generar otra vez el informe, realizando todos los cálculos de nuevo y el formato gráfico del informe.
- Conseguir que otro usuario pueda utilizar estas funciones es realmente complicado, tanto por el motivo inicial que hay que instalar en local todo lo necesario, si no que además hay que acompañar el informe de un manual de instrucciones muy personalizado.

Toda esta filosofía, si bien sirve para determinados estudios, podríamos decir con un inicio y un final marcados, pueden no ser prácticos cuando se desea realizar muchas pruebas con distintos datos, y filtrándolos según sus características. Además determinadas herramientas puede ser interesante que sean usadas por varias personas a la vez con distintas fuentes de datos.

Para resolver este tipo de estudios más variables o con muchos posibles resultados, siempre que sea factible, la opción de una aplicación web es la mejor opción porque resuelve los siguientes problemas:

- Se encuentra fácilmente en Internet desde cualquier sitio y ordenador
- No se requiere instalar aplicaciones ni librerías en local.
- Podemos comprobar resultados de forma reactiva en tiempo real cambiando los parámetros.
- Aunque se puede acompañar de un manual, las aplicaciones suelen ser intuitivas, al menos mucho más que la generación de informes.
- Podemos optar al uso de procesadores en la nube mucho más potentes que nuestro ordenador.
- La aplicación es accesible al investigador alejado de la informática y la programación.

Este es el motivo de elegir esta forma de representación para los datos de **BacDup** y de ahí el nombre **BacDupWeb**.

1.2 Objetivos del Trabajo

1.2.1 Objetivos previos al desarrollo:

- Adquirir conocimientos sobre la estructura del ADN Bacteriano.
- Elaborar una relación de componentes genómicos interesantes para el investigador.
- Adquirir conocimientos sobre las herramientas actuales de búsqueda y comparación de secuencias
- Dominar con profundidad todos los campos del archivo de datos de duplicados generado por BacDup
- Elaborar una relación de funcionalidades concretas que deberá realizar nuestra aplicación.
- Obtener una relación de datos objetivos con los que deberá lidiar la aplicación
- Encontrar las herramientas informáticas adecuadas a los datos y funcionalidades estudiadas.

1.2.2 Objetivos de ejecución del desarrollo

- Conseguir un diseño moderno de la aplicación
- Definir unas funcionalidades de filtrado útiles, intuitivas y prácticas.
- Desarrollar un resultado visual de datos en tablas claras y dinámicas
- Crear una aplicación con un resultado visual gráfico claro y dinámico

1.2.3 Objetivos de entrega

- Redactar una buena memoria
- Crear una buena presentación, clara y útil
- Realizar una buena presentación y defensa pública
- Mostrar el resultado de una aplicación web de consulta de genomas bacterianos clara, útil y práctica más allá del TFM.

1.3 Enfoque y método seguido

El enfoque principal del trabajo es la de obtener agilidad en la consulta de la información. En este caso, el trabajo del bioinformático consideramos que no es tanto el de sacar conclusiones biológicas con un informe al final de un pipeline rígido, si no el de dotar al investigador de una herramienta de consulta rápida de genes duplicados y de sus productos. Y eso para cualquier cepa de estudio.

Por lo tanto, no se está planteando un script o una utilidad para obtener unos resultados concretos, si no que se ha diseñado una potente aplicación web, dinámica, para que el usuario de la misma obtenga el máximo de resultados y conclusiones distintas en función de unos archivos originales de estudio y cambiando una serie de parámetros, modificables en tiempo real.

El enfoque que se prioriza es el de la agilidad en la consulta por encima incluso del de disponer de más funcionalidades. Por lo tanto hay que definir un equilibrio entre prestaciones y agilidad en las consultas al aplicar los filtros. A partir de aquí se busca el entorno de programación que priorice esta característica.

1.4 Planificación del Trabajo

El trabajo se ha organizado según 4 Fases que se citan a continuación en distintos sucapítulos.

1.4.1 Planificación: Tareas previas

En este apartado se han realizado tareas para conocer los motivos biológicos que han llevado al interés en desarrollar una aplicación como BacDupWeb y al análisis previo de los archivos que hay que manipular, así como los objetivos finales a nivel técnico. En este apartado se incluye la toma de decisiones respecto las herramientas informáticas a utilizar.

Respecto el TFM, después de esta fase de estudio se hace entrega de la PEC1.

1.4.2 Planificación: Desarrollo Fase 1

La primera parte de la fase 1 ha consistido en el diseño de la aplicación a nivel estructural, teniendo en cuenta la carga de archivos, los filtros la parte de datos directos y la parte de datos variables. En esta fase se ha iniciado el desarrollo de la parte estructural mediante Shiny.

La segunda parte de la fase 1 ha consistitdo en realizar ya la programación de las funciones básicas de carga de archivos, filtros y aproximadamente un 70% de las funcionalidades de la aplicación.

Respecto el TFM, después de esta fase 1 de desarollo se hace entrega de la PEC2.

1.4.3 Planificación: Desarrollo Fase 2

En la primera parte de esta fase se ha complementado la aplicación con el gráfico circos, el link a NCBI y la posibilidad de bajarse un informe del estudio realizado, además de capturar datos de un repositorio de github.

En la segunda parte de esta fase se han revisado todas las funciones mediante la carga de distintos archivos, se ha mejorado la estética con CSS y JS y se han ido terminando los distintos apartados del código con comentarios.

Respecto el TFM, después de esta fase 2 de desarollo se hace entrega de la PEC3.

1.4.4 Planificación: Entregas

Este bloque consiste en la elaboración de la memoria final del TFM además de la presentación y la defensa pública del mismo. Además este bloque incluye la finalización de la aplicación en sí misma, tanto en el trabajo mediante control de versiones en GitHub como en la puesta en producción de la aplicación de Shiny en shinyapps.io

Al final de la fase final se ha entregado la presente memoria que equivale a la PEC4. Esta entrega es previa a la presentación y defensa pública que forman parte de toda la fase final del proyecto.

En la Figura 5 se adjunta el cronograma de la planificación por semanas con el detalle de tareas según la estructura del trabajo de TFM encajado en el semestre correspondiente.

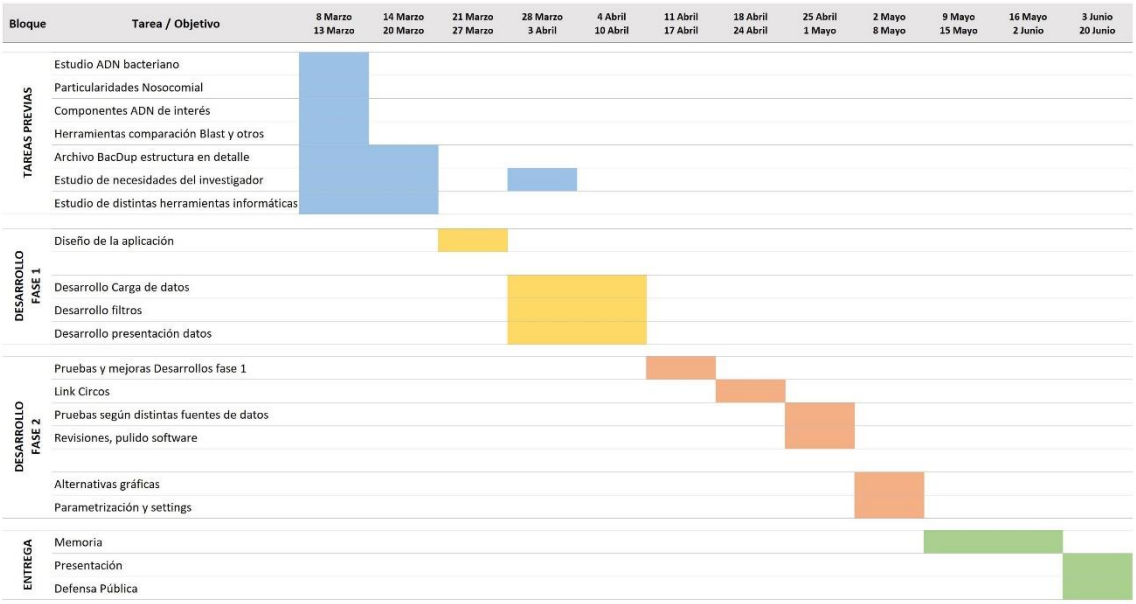


Figura 5: Cronograma del desarrollo de BacDupWeb

1.5 Breve sumario de contribuciones y productos obtenidos

El presente Trabajo Final de Máster ofrece, como resultado **BacDupWeb**, una aplicación web para el análisis de genes duplicados en bacterias teniendo, como punto de partida, los archivos resultantes de **BacDup**.

Además de la aplicación, se hace entrega de la memoria del TFM, una memoria descriptiva que justifica dicho desarrollo, además de explicar los antecedentes, l diseño, la planificación, la ejecución y las conclusiones finales.

Toda esta información y aplicaciones se pueden encontrar recopiladas en el siguiente repositorio de GitHub: https://github.com/dibanezmal/BACDUPWEB

Y la propia aplicación en: https://bacdupweb.shinyapps.io/BacDupWeb8/

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos que vamos a encontrar en la presente memoria son los siguientes:

Metodología: En esta capítulo se explica los criterios elegidos para ejecutar el proyecto, incluyendo algunos cambios que hayan podido surgir en el camino.

Resultados: Este es el bloque principal del proyecto donde se explica con detalle la aplicación desarrollada con todas sus funcionalidades.

Discusión: Capítulo dedicado al análisis posterior del resultado obtenido en base a algunos apartados, funcionalidades, detalles, aspectos positivos y aspectos negativos.

Valoración económica: se hace una breve reflexión sobre los posibles costes de una aplicación web en general y de BacDupWeb en particular.

Conclusiones: Conclusiones finales sobre el resultado obtenido.

Glosario: Términos técnicos, acrónimos y palabras singulares de este proyecto que puedan ayudar al lector no especializado.

Bibliografía: Referencias de consulta y de soporte a las argumentaciones y al desarrollo del proyecto.

2 Metodología

Para el desarrollo de la aplicación se ha tenido que realizar, antes de todo, un planteamiento de donde se partía y a donde se quería llegar. Posteriormente realizar los pasos intermedios necesarios. Con esta filosofía de trabajo salieron una serie de puntos a tratar que aquí quedan agrupados :

- Datos de partida: donde se tiene la información, en qué formato y cómo se puede obtener.
- Qué se desea obtener: Tablas y Gráficos que serían de utilidad.
- Qué parámetros y campos de los archivos seria interesante poder variar en tiempo real.
- Diseño del entorno de programación: qué lenguajes de programación permiten realizar este desarrollo y qué bases de datos usar. En definitiva abordar el clásico proyecto web con los tres apartados: frontend, backend y bbdd.
- Qué editor de texto usar para el código.
- Software para control de versiones.
- Como realizar pruebas. Servidor localhost y/o público.

En los siguientes apartados de este capítulo se van explicando los anteriores puntos.

2.1 Entorno tecnológico de trabajo

Para elaborar este trabajo se ha utilizado un ordenador portátil Lenovo Yoga con sistema operativo Windows 10, procesador intel Core i7, con 8 Gb de RAM y disco SSD de 500 GB.

Como editor de textos se empezó usando Visual Studio Code al trabajar inicialmente con Django pero al final se optó por el editor de R Studio configurando el entorno gráfico y haciéndolo similar a VS con fondo oscuro y con el código en colores distintos para las funciones, variables y comentarios.

Como lenguaje de programación se ha usado el lenguaje R mediante el framework Shiny que permite facilitar la programación en HTML.

Para modificar el aspecto gráfico estándar de Shiny se han usado scripts de CSS, HTML y JS.

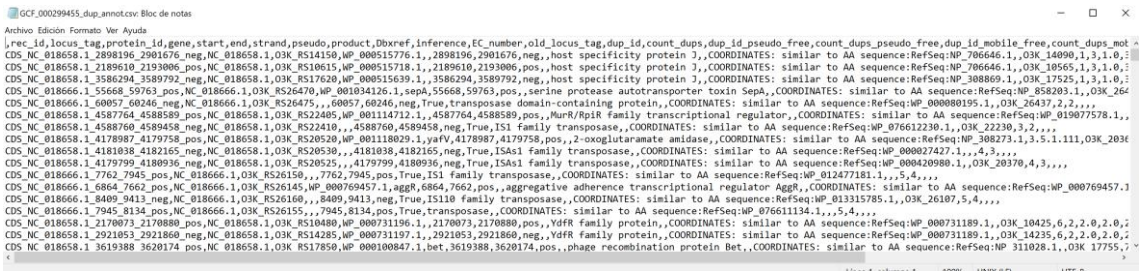
Se ha trabajado mediante control de versiones usando GitHub. El repositorio donde se ubica la aplicación es público y es el siguiente:

https://github.com/dibanezmal/BACDUPWEB

2.2 Estudio previo de los archivos de datos a visualizar

Al margen de la visión general del objetivo, de los motivos biológicos y de la lectura del trabajo **BacDup** del que parte **BacDupWeb**, la primera parte del desarrollo consistió en conocer técnicamente de dónde partimos. En efecto, BacDupWeb tiene que visualizar unos datos, pero cómo son estos datos, están todos en un archivo o en varios archivos, qué campos contienen? qué significa cada campo? Qué tamaño tienen los archivos? Son manejables?.

La parte principal del proyecto parte del archivo dup_Annot en formato csv. Este archivo es una anotación de todas las proteínas duplicadas detectadas en base al uso de Blast con unos parámetros, en este caso, bastante exigentes a la hora de considerar un gen o proteína duplicados de otro.


GCF_000299455_dup_annot.csv: Bloc de notas

Archivo Edición Formato Ver Ayuda

,rec_id,locus_tag,protein_id,gene,start,end,strand,pseudo,product,Dbxref,inference,EC_number,old_locus_tag,dup_id,count_dups,dup_id_pseudo_free,count_dups_pseudo_free,dup_id_mobile_free,count_dups_mob

CDS_NC_018658.1_2898196_2901676_neg,NC_018658.1,O3K_RS14150,WP_000515776.1,,2898196,2901676,neg,,host specificity protein J,,COORDINATES: similar to AA sequence:RefSeq:NP_706646.1,,O3K_14090,1,3,1.0,3

CDS_NC_018658.1_2189610_2193006_pos,NC_018658.1,O3K_RS10615,WP_000515718.1,,2189610,2193006,pos,,host specificity protein J,,COORDINATES: similar to AA sequence:RefSeq:NP_706646.1,,O3K_10565,1,3,1.0,3

CDS_NC_018658.1_3586294_3589792_neg,NC_018658.1,O3K_RS17620,WP_000515639.1,,3586294,3589792,neg,,host specificity protein J,,COORDINATES: similar to AA sequence:RefSeq:NP_308869.1,,O3K_17525,1,3,1.0,3

CDS_NC_018666.1_55668_59763_pos,NC_018666.1,O3K_RS26470,WP_001034126.1,sepA,55668,59763,pos,,serine protease autotransporter toxin SepA,,COORDINATES: similar to AA sequence:RefSeq:NP_858203.1,,O3K_264

CDS_NC_018666.1_60057_60246_neg,NC_018666.1,O3K_RS26475,,,60057,60246,neg,True,transposase domain-containing protein,,COORDINATES: similar to AA sequence:RefSeq:WP_000080195.1,,O3K_26437,2,2,,,,

CDS_NC_018658.1_4587764_4588589_pos,NC_018658.1,O3K_RS22405,WP_001114712.1,,4587764,4588589,pos,,MurR/RpiR family transcriptional regulator,,COORDINATES: similar to AA sequence:RefSeq:WP_019077578.1,,

CDS_NC_018658.1_4588760_4589458_neg,NC_018658.1,O3K_RS22410,,,4588760,4589458,neg,True,IS1 family transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_076612230.1,,O3K_22230,3,2,,,,

CDS_NC_018658.1_4178987_4179758_pos,NC_018658.1,O3K_RS20520,WP_001118029.1,yafV,4178987,4179758,pos,,2-oxoglutaramate amidase,,COORDINATES: similar to AA sequence:RefSeq:NP_308273.1,3.5.1.111,O3K_203

CDS_NC_018658.1_4181038_4182165_neg,NC_018658.1,O3K_RS20530,,,4181038,4182165,neg,True,ISAs1 family transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_000027427.1,,,4,3,,,,

CDS_NC_018658.1_4179799_4180936_neg,NC_018658.1,O3K_RS20525,,,4179799,4180936,neg,True,ISAs1 family transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_000420980.1,,O3K_20370,4,3,,,,

CDS_NC_018666.1_7762_7945_pos,NC_018666.1,O3K_RS26150,,,7762,7945,pos,True,IS1 family transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_012477181.1,,,5,4,,,,

CDS_NC_018666.1_6864_7662_pos,NC_018666.1,O3K_RS26145,WP_000769457.1,aggR,6864,7662,pos,,aggregative adherence transcriptional regulator AggR,,COORDINATES: similar to AA sequence:RefSeq:WP_000769457.1

CDS_NC_018666.1_8409_9413_neg,NC_018666.1,O3K_RS26160,,,8409,9413,neg,True,IS110 family transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_013315785.1,,O3K_26107,5,4,,,,

CDS_NC_018666.1_7945_8134_pos,NC_018666.1,O3K_RS26155,,,7945,8134,pos,True,transposase,,COORDINATES: similar to AA sequence:RefSeq:WP_076611134.1,,,5,4,,,,

CDS_NC_018658.1_2170073_2170880_pos,NC_018658.1,O3K_RS10480,WP_000731196.1,,2170073,2170880,pos,,YdfR family protein,,COORDINATES: similar to AA sequence:RefSeq:WP_000731189.1,,O3K_10425,6,2,2.0,2.0,2

CDS_NC_018658.1_2921053_2921860_neg,NC_018658.1,O3K_RS14285,WP_000731197.1,,2921053,2921860,neg,,YdfR family protein,,COORDINATES: similar to AA sequence:RefSeq:WP_000731189.1,,O3K_14235,6,2,2.0,2.0,2

CDS_NC_018658.1_3619388_3620174_pos,NC_018658.1,O3K_RS17850,WP_000100847.1,bet,3619388,3620174,pos,,phage recombination protein Bet,,COORDINATES: similar to AA sequence:RefSeq:NP_311028.1,,O3K_17755,7

Figura 6: : Ejemplo de archivo de duplicados dup_Annot

Un registro de este archivo puede ser el siguiente:

```
CDS_NC_018666.1_60057_60246_neg,NC_018666.1,O3K_RS26475,,,60057,
60246,neg,True,transposase domain-containing
protein,,COORDINATES: similar to AA
sequence:RefSeq:WP_000080195.1,,O3K_26437,2,2,,,,
```

Los campos de este archivo son los mostrados en la Figura 7:

Column Name	Description
rec_id	Contig or chromosome ID
locus_tag	Unique gene ID provided by Genbank annotation
protein_id	Protein ID provided by Genbank annotation
gene	Gene name (if available)
start	Coordinate Start position (in rec_id)
end	Coordinate End position (in rec_id)
strand	Strand value (+/-)
pseudo	Pseudogene True/False
product	Gene description (if available)
Dbxref	External reference to third party database (if available)
inference	Inference details (if available)
EC_number	Enzyme family unique ID (if available)
old_locus_tag	Former Unique gene ID provided by Genbank annotation

Figura 7: Detalle de campos del archivo dup_annot

Por otra parte tenemos el archivo de longitudes de los cromosomas y plásmidos. Por regla general, en bacterias solo hay un cromosoma de pocas Megabases de longitud y luego varios plásmidos relativamente cortos que complementan el genoma. Un archivo length tiene un contenido como el de la Figura 8

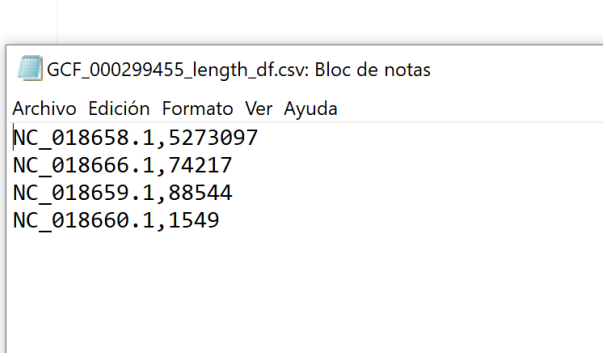


Figura 8: Archivo de secuencias con las longitudes en bp (length.csv)

Finalmente disponemos de un archivo info.csv que se puede obtener directamente de **RefSeq/GenBank** [16] cuando no se trata de secuenciación de novo, e incluye información general, por ejemplo de la taxonomía de la cepa y por tanto de la especie de patógeno. Siguiendo este mismo ejemplo con el código GCF_000299455.1, podemos ver el archivo info.csv en la Figura 9.

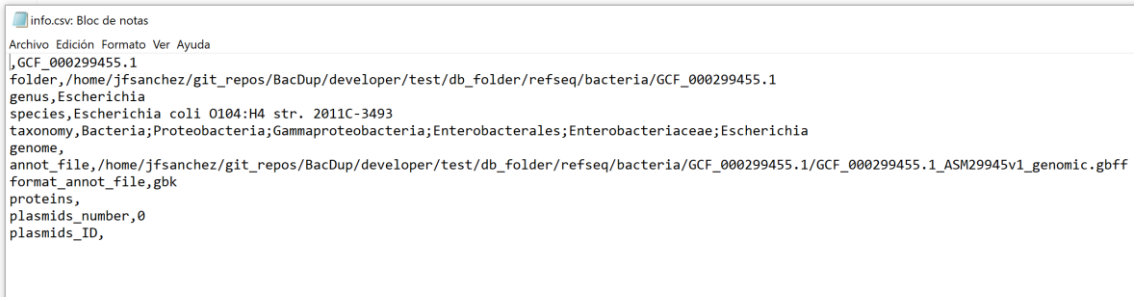


Figura 9: detalle de archivo info.csv

2.3 Elección de los lenguajes de desarrollo

Una vez se ha realizado el análisis de los tres archivos de partida y de los datos que hay que visualizar, el siguiente paso es elegir las herramientas adecuadas para el desarrollo. En esta fase es necesario adentrarse en el análisis de la velocidad de cada lenguaje para las consultas y en los aspectos positivos o negativos de cada opción para cada apartado de la aplicación.

Para el diseño de la aplicación, como en todas las de desarrollo web, se suele plantear en tres bloques: la parte de *frontend*, la parte de *backend* y la parte de *bbdd (bases de datos)*. En un primer momento se planteó la posibilidad de trabajar con Mysql como gestor de bases de datos, trabajar con HTML y CSS para *frontend* y trabajar con Python para *backend*.

Si nos centramos inicialmente sobre la elección del motor de bases de datos a utilizar, una vez analizados los archivos para varias cepas, se detectó que los archivos eran “pequeños” con unos 1000 registros. Incluso aunque hubiera archivos con 10.000 registros, seguirían siendo poco pesados a efectos informáticos. Además, los archivos se iban a manipular pero no a escribir sobre ellos. Por este motivo se consideró que no tenía sentido trabajar con estructuras de bases de datos complejas y la aplicación podía cargar los datos en memoria y trabajar sobre ellos sin necesidad de un motor de bases de datos.

Es interesante destacar que aplicaciones como Galaxy o como Google Colab, usadas durante el máster, trabajan en servidores web con aplicaciones donde el usuario carga los archivos de estudio. **BacDupWeb** podía trabajar del mismo modo.

En cuanto a la programación, en un primer momento, se plantearon varias opciones para realizar el desarrollo. A priori Shiny se prejuzgó como un sistema de programación sencillo, de entornos académicos, para realizar pequeñas aplicaciones web. No era inicialmente el objetivo ya que no se deseaba tener limitaciones, al menos ya de inicio. Se deseaba abordar el trabajo de forma teóricamente más profesional y con lenguajes de mayor alcance y diseñados específicamente para realizar páginas web.

Desde un inicio se descartó el uso de **html, css y Javascript** como único sistema ya que el conocimiento de estos lenguajes eran solo los del propio máster. Con el tiempo disponible para el TFM, no era posible aprender en profundidad el lenguaje y crear después el código.

Así que se eligió **Python**, dado el mayor conocimiento y por ser un lenguaje atractivo, más fácil de interpretar y rápido en avanzar. Para abordar el proyecto se debía trabajar con **Django** o **Flask**, dos *framework* [17]para el desarrollo de aplicaciones web en **Python**. Se optó por Django por ser más reciente y sin limitaciones. Así que se realizó un curso intensivo de más de 30 horas [18]en la que se realizaron distintos ejemplos. No obstante, aunque no fue evidente en un primer momento, la filosofía de intentar dominar un lenguaje nuevo en tan poco tiempo, era poco recomendable.

Con una cierta preocupación por el hecho objetivo de empezar de cero y haber perdido mucho tiempo, se retomó la idea inicial de **Shiny.** Esta decisión se tomó no sin realizar previamente una investigación en profundidad de **Shiny** mediante tutoriales y artículos de expertos que incluso comparaban la potencia de **PowerBi** y **Shiny.**

Hay que decir que pronto se pudo comprobar que la velocidad con la que **Shiny** nos permite obtener los primeros resultados y de forma ágil e intuitiva, es sorprendente. En dos jornadas intensivas de programación ya se pudieron empezar a mostrar datos de la tabla **dup_annot** con algunos filtros, reaccionando a los cambios de forma inmediata. A partir de aquí, no se tuvieron más dudas sobre el lenguaje de programación a seguir. En resumen, no se usan motores de bases de datos tipo **Mysql** o similares, se usa html a través de Shiny para la parte de *frontend* y se programa mediante R en la parte *backend*, un lenguaje mucho más trabajado y con mucha información en Internet [19].

2.4 Breve descripción de Shiny

Como se ha comentado previamente, Shiny es un framework de R para realizar aplicaciones web. Un framework es un entorno de trabajo que consiste, básicamente, en una serie de librerías y funciones que nos permiten construir partes de la aplicación web de forma más directa, sin tener que escribir todo el código html.

Es especialmente sorprendente el caso de Shiny respecto otros frameworks como Django o Flask que serian los equivalentes para Python. Y es que Shiny trae de inicio, ejemplos de aplicaciones como por ejemplo la de la Figura 10 en la que se muestra un histograma en el que podemos visualizar los cambios en el gráfico al cambiar el numero de breaks o bins en tiempo real.

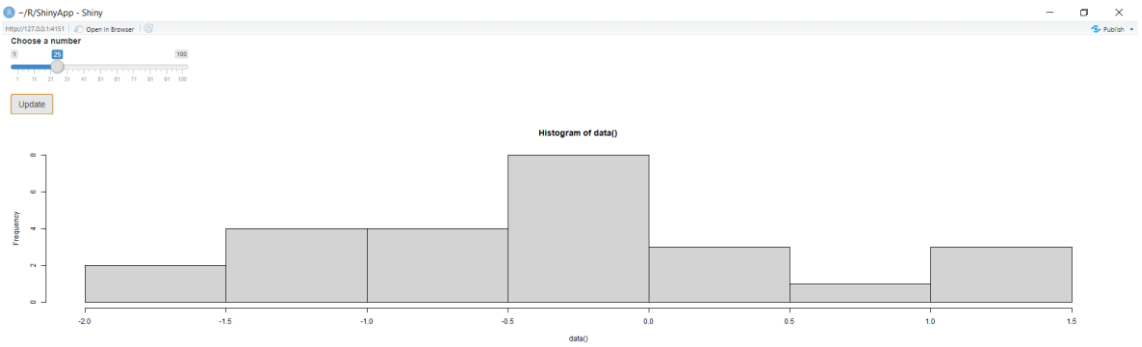


Figura 10: Ejemplo básico e inicial del paquete Shiny

Consideramos que no entra dentro de los objetivos de este trabajo analizar en profundidad como funciona internamente Shiny pero si es importante destacar que, un ejemplo como el de la Figura 10, se crea con menos de 10 líneas de código y eso es muy destacable en programación. En cuanto a funcionamiento general, Shiny trabaja con dos bloques de código:

- **UI:** User Interface, Interfaz de usuario
- **SERVER:** Manipulación de datos, programación en R

La interfaz de usuario (UI) está formada por todo lo que visualiza el usuario y tiene dos partes, una parte consiste en los parámetros que queremos modificar, filtros que queremos aplicar, carga de archivos, etc. Son los **inputs**. La segunda parte son las tablas, los gráficos y, en general, los resultados de nuestra selección. Son los **outputs**. Toda esta parte, inputs y outputs, visibles para el usuario, es lo que en términos de programación se llama **frontend**.

En la parte no visible (**backend**) de la aplicación está el bloque Server que consiste en la parte de los cálculos, la manipulación de datos, la aplicación de los filtros, el código en sí mismo. Normalmente se trata de scripts de R que reciben los inputs del bloque UI y trabajan con ellos para devolverlos modificados en forma de tabla filtrada o de gráfico. Estas funciones vuelven como outputs a la parte frontend para ser renderizados (montados) en pantalla.

Al final de cualquier aplicación desarrollada con el framework **Shiny**, existe una función **ShinyApp()** que ejecuta la comunicación entre el bloque ui y el bloque server. Trabajar con esta base es sencillo como concepto pero enseguida la programación se va complicando, el manejo de variables entre la parte ui y la parte server no es evidente y requiere de una buena organización del código y no dejar de incluir comentarios.

Como observaciones adicionales, Shiny permite modificar su código frontend añadiendo características estéticas de css, html y js. De hecho, Shiny puede trabajar en su parte UI con una página externa index.html creada aparte [20].

2.5 Relación de inputs de la aplicación

Conociendo ya el lenguaje de programación y los archivos de salida hay que intentar realizar una lista de funcionalidades en cuanto a variables modificables del sistema. En esta lista se relacionan las que a priori se desean implementar:

Carga de archivos: En un principio se decide cargar directamente los archivos dup_annot y length, los necesarios para el paquete circos.

Secuencias: Es interesante poder seleccionar si trabajamos con el cromosoma principal, con un plásmido o con todo el genoma. Por lo tanto interesante un selector de secuencias. Importante destacar que esta información depende de cada archivo dup_annot.

Hebra: Es importante en los estudios genéticos conocer si un gen o una proteína procede de la hebra positiva o negativa.

Pseudogenes y fagos: Los archivos dup_annot contienen genes pero también pseudogenes y elementos móviles. En cualquier estudio será interesante poder discriminar entre incluir o no estos elementos.

Número de duplicados: En estudios de virulencia puede ser interesante concentrar los esfuerzos en aquellos genes o proteínas o elementos genéticos que presentan muchas copias. En este sentido disponer de un selector de número mínimo de copias es interesante. Por ejemplo genes o proteínas con más de 5 copias. Es importante destacar que este selector depende de datos del propio archivo cargado.

Campos de la tabla: Interesante poder configurar la tabla de los datos en función de nuestro estudio. Poder seleccionar exactamente los campos a mostrar, será de gran ayuda. En un caso la proteína será lo más importante junto al número de duplicados y en otro caso conocer el locus inicial y final, la posición en el genoma, será clave para el estudio.

Usuario y sesión: Aunque finalmente se explica que no se ha implementado en el tiempo de TFM, la aplicación está pensada para que puedan acceder varios usuarios simultáneos, cada uno con su nombre y *password* y poder incluso registrar en un pequeño archivo los datos de la cuenta y parámetros de configuración básicos.

2.6 Elementos a mostrar como resultados

En el apartado anterior se han definido los elementos input que podremos modificar. En este se relacionan los resultados en pantalla, es decir, aquellas tablas, gráficos y datos de interés para el investigador.

Tabla principal de datos: Tabla de genes o proteínas del archivo dup_annot que cumplen con las condiciones de los filtros y que se muestran en la tabla según los campos seleccionados.

Gráfico circos: Gráfico circos que varie según los parámetros elegidos.

Tabla de secuencias: Tabla de cromosomas y plásmidos:

Taxonomia: Información taxonómica de la cepa de estudio obtenida de GenBank / NCBI.

Detalle de duplicados: Información de los duplicados según los filtros aplicados.

Histograma: Gráfico con la distribución frecuencial de duplicados.

Además debe ser posible emitir un informe de los resultados estudiados y también acceder online a la información de la cepa a GenBank si esta está registrada.

2.7 Diagrama de bloques de BacDupWeb

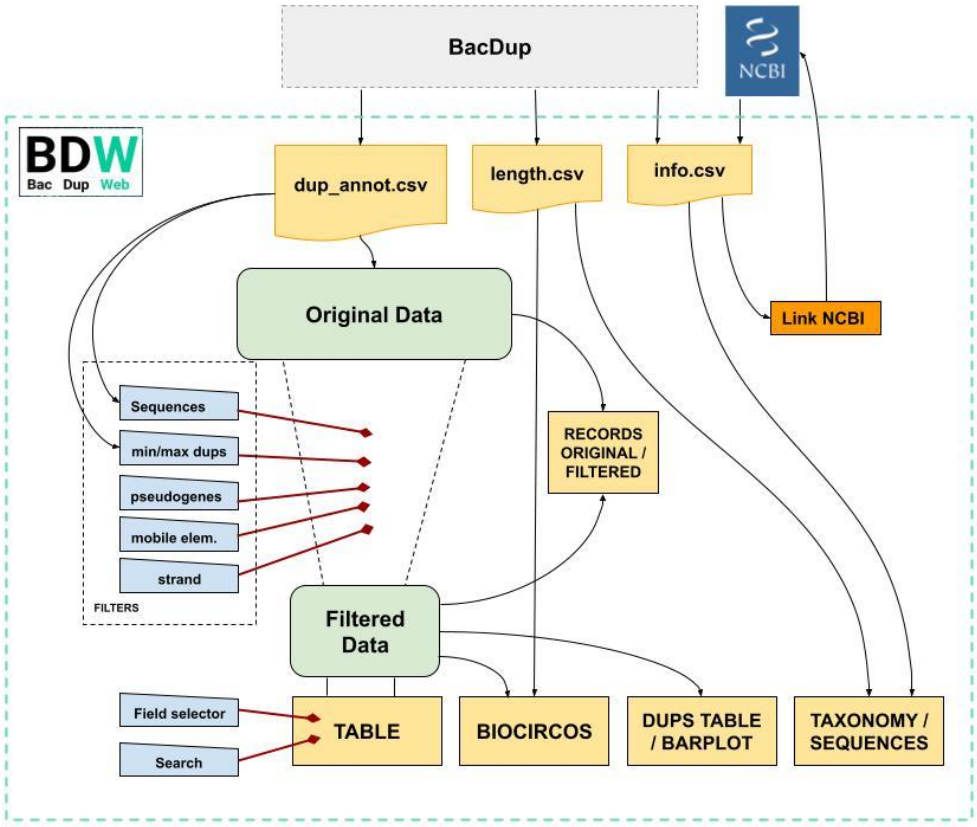


Figura 11: Diagrama de bloques de BacDupWeb

Los rectángulos amarillos inferiores corresponden a los resultados de la aplicación en sus 5 apartados. Las tablas original y filtrada aparecen en verde y todos los filtros en azul. En la parte superior, también en amarillo, los tres archivos de entrada a BacDupWeb.

3 Resultados

3.1 Descripción general de la aplicación

El resultado del trabajo ha consistido en el desarrollo de una aplicación web que consta de 5 apartados de resultados que van variando en función de los archivos cargados y los filtros aplicados. A continuación se realiza un listado de dichos apartados y subapartados para posteriormente explicar la información que se obtiene en cada uno de ellos.

3.2 Relación de apartados y subapartados de la aplicación:

BLOQUE SELECTOR

- Carga de archivos
- Carga de selectores automáticos
- Carga de selectores condicionales

BLOQUE RESULTADOS

- Datos generales y funciones básicas
- Tabla dinámica de genes duplicados
- Gráfico Circos de genes duplicados
- Tabla taxonómica y de cromosomas
- Datos frecuenciales de genes duplicados

BLOQUE IN/OUT

- Link a datos info en NCBI
- Download report

3.3 Descripción detallada de los apartados desarrollados

3.3.1 Bloque selector: Carga de archivos

La primera función imprescindible para trabajar con la aplicación es la carga de archivos. Como hemos explicado en el apartado anterior, **BacDupWeb** necesita alimentarse de 2 archivos:

Archivo anotación duplicados: GCF_nnnnnnnnn_dup_annot.csv

Archivo información de genomas: GCF_nnnnnnnnn_length.csv

Donde el indicador de 9 caracteres nnnnnnnnn identifica la cepa según la identificación en **GenBank**. Aunque podría realizarse una carga más amplia de los archivos iniciales, esta versión de **BacDupWeb** requiere de esta nomenclatura de archivos ya que tiene implicaciones en el funcionamiento de la aplicación. Si los archivos tienen la estructura correcta pero no se llaman así, no se podrán realizar llamadas a **GenBank**.

Si observamos la aplicación, en la parte izquierda encontramos la carga de los dos archivos. Una vez cargados podemos accionar el botón **Go** que activará todas las funcionalidades de consulta sobre dicha cepa.

Lógicamente los dos archivos deben pertenecer a la misma cepa. En este caso la cepa es la GCF_000299455. La aplicación tiene algunos controles para no avanzar si no se cargan los archivos pero hay que incluir funciones sobre el tipo de archivos sobre los que ahora mismo no hay control.

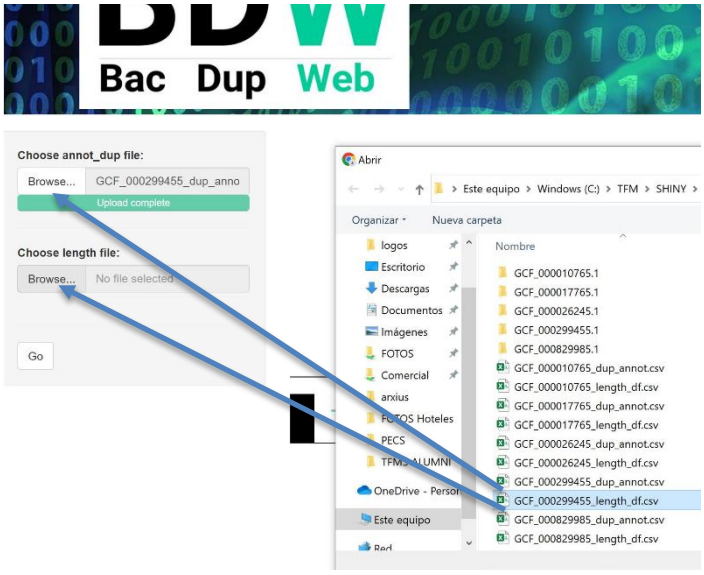


Figura 12: BacDupWeb carga archivos

3.3.2 Bloque selector: Carga de filtros

Con el objetivo de visualizar los datos de duplicaciones génicas, por defecto las tablas y los gráficos aparecen con todos los registros activos. Es a partir del uso de la aplicación que podemos actuar sobre una serie de filtros. De estos filtros hay de dos tipos, filtros fijos que no dependen de la cepa y filtros que sí dependen de los datos de la cepa, es decir, de los archivos cargados.

En la figura de la derecha podemos ver, los filtros que aparecen en la zona izquierda de la aplicación y que permiten seleccionar datos de la tabla. Empezando por los tres filtros de la parte inferior, podemos filtrar según la hebra sea positiva, negativa o ambas, según se incluyan o no los pseudogenes en nuestro estudio o bien incluyamos o no los elementos móviles (fagos).

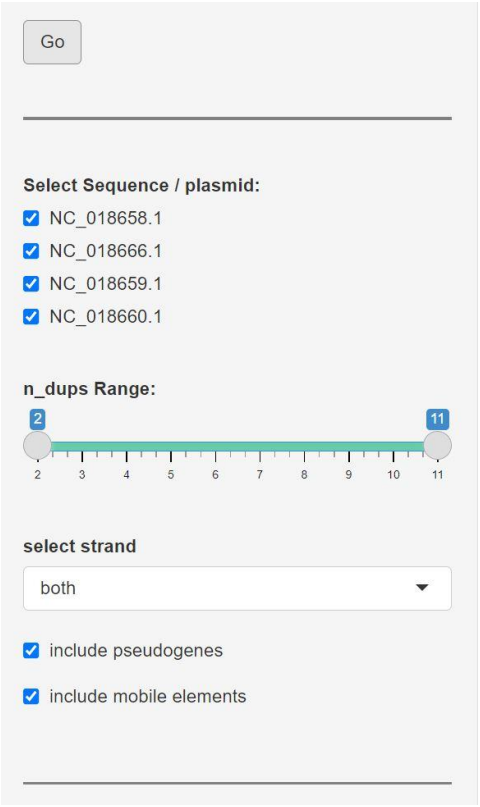


Figura 13:BacDupWeb filtros

Además de los filtros generales iguales para todas las cepas, la aplicación es capaz de cargar también como filtros, elementos de los propios archivos. Hay dos filtros que dependen de los archivos cargados. El primero es la relación de cromosomas o plásmidos que contiene el genoma de la cepa. De este modo, mediante *checkbox*, podremos decidir si mostramos en tablas y gráficos todas las secuencias, solo el cromosoma principal o solo un plásmido concreto. El segundo filtro se corresponde con los genes duplicados en función del número de copias. Mediante un selector slider, podemos seleccionar los genes con solo 2 copias o al revés, genes con 5 o más copias por ejemplo, que podrían ser susceptibles de causar más virulencia al mostrar más carga. En cualquier caso, el filtro nos permite realizar dicha selección y el mínimo y el máximo respecto al número de copias, lo determina el fichero cargado. En el caso de la Figura 13 vemos que la cepa dispone de 1 cromosoma principal y 3 plásmidos por los que podremos filtrar y además disponemos de genes con 2 copias y hasta un máximo de 11 copias.

3.3.3 Bloque selector: Carga de campos de visualización

Como complemento de los filtros, disponemos de un selector que más que actuar como filtro de los datos, permite seleccionar los campos de la tabla a visualizar. Este selector actua solo sobre la tabla principal y permite definir qué campos queremos visualizar de la tabla.

Por defecto **BacDupWeb** nos propone un conjunto de 7 campos a visualizar como se muestra en la Figura 14, a priori los más relevantes pero el usuario podrá seleccionar los que desee en tiempo real. Esta es una de las maravillas de las aplicaciones web dinámicas, y es que podremos configurar la tabla no al incio del estudio, no mediante un parámetro que pasamos al informe, si no que se mostrarán los campos que seleccionemos en tiempo real sobre los datos filtrados. La potencia de filtrar y configurar la tabla de forma dinámica, con actualización inmediata es francamente útil, en general y en este caso en particular.

Como configuración adicional, el *checkbox compact view* nos permite optar por mostrar todos los campos si lo deseleccionamos o si volvemos a marcarlo, volveremos a los campos de la selección.

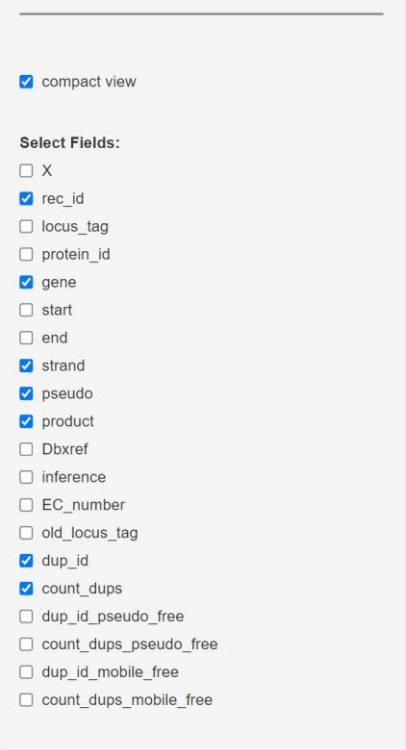


Figura 14: BacDupWeb selector de campos

3.3.4 Bloque Resultados: Datos generales y funciones básicas

Una vez descritas la carga de archivos, el selector de campos y los filtros, entramos de lleno en los resultados, en la aplicación en sí misma.

Como primera parte tenemos dos tablas en la parte superior, indicadas con un recuadro rojo en la Figura 15 que nos indica los registros de la tabla y el máximo y mínimo número de copias para un gen. La tabla de la izquierda nos ofrecerá datos siempre del archivo original completo **Original Data** y la tabla de la derecha **Filtered Data**, nos mostrará el total de datos filtrados.

Aquí ya se puede comprobar la agilidad en la actualización de datos. En milisegundos, cualquier actuación sobre los filtros de la banda izquierda, nos actualizan la tabla **Filtered Data** con el total de genes y copias mínima y máxima que contiene la tabla filtrada, pudiéndola comparar con los totales originales en todo momento.

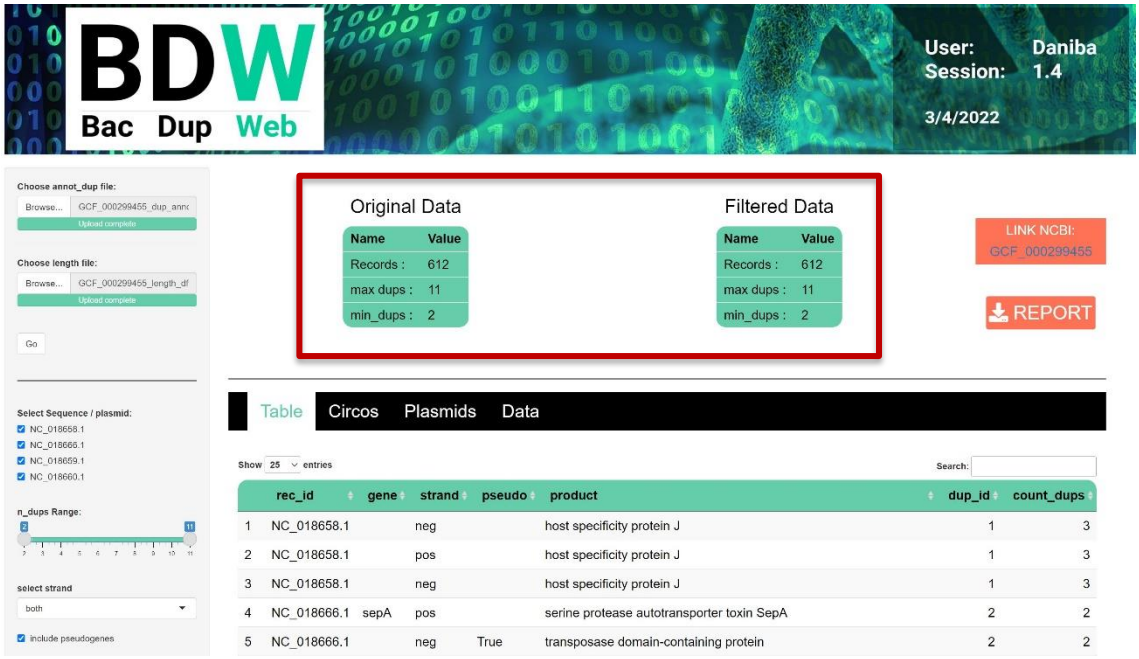


Figura 15: BacDupWeb: tablas de totales y datos filtrados

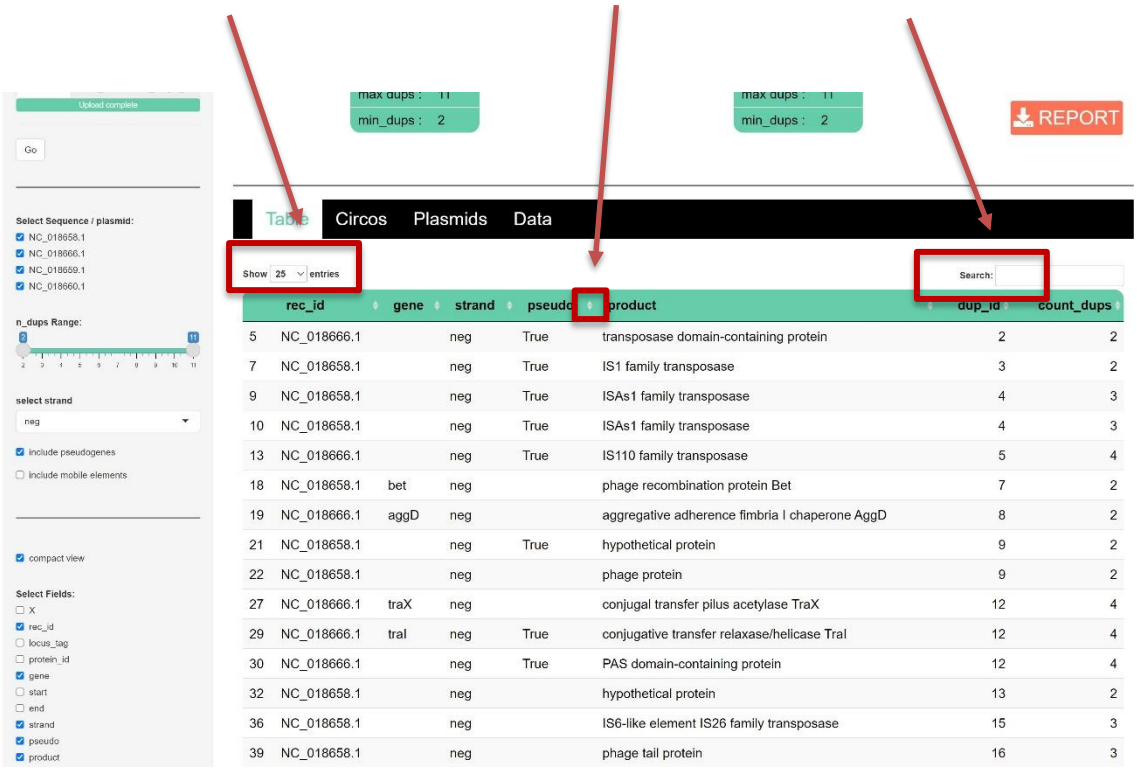
3.3.5 Bloque Resultados: Tabla dinámica de genes duplicados

Esta parte es el resultado principal del proyecto y consiste en mostrar los registros de la tabla original dup_annot, que cumple con los filtros activados. La aplicación está dividida en su parte principal, en un menú de tipo horizontal con 4 opciones. En este apartado comentamos la primera pestaña de la aplicación **Table**.

En la Figura 16 podemos observar la tabla con los campos seleccionados por defecto y que ya hemos comentado que podemos modificar en tiempo real. Además tenemos un selector más que es la cantidad de registros por página, en indicada en la misma figura como **show=25** y por otra parte disponemos de la posibilidad de clasificar la tabla por cualquier campo.

Una vez seleccionados unos registros mediante los filtros, la funcionalidad que permite el máximo de selección es la de buscar un texto directamente en el

recuadro search. Filtrar por texto ofrece una potencia elevada en los estudios para buscar proteínas concretas entre cientos o miles.



	rec_id	gene	strand	pseudo	product	dup_id	count_dups
5	NC_018666.1		neg	True	transposase domain-containing protein	2	2
7	NC_018658.1		neg	True	IS1 family transposase	3	2
9	NC_018658.1		neg	True	ISAs1 family transposase	4	3
10	NC_018658.1		neg	True	ISAs1 family transposase	4	3
13	NC_018666.1		neg	True	IS110 family transposase	5	4
18	NC_018658.1	bet	neg		phage recombination protein Bet	7	2
19	NC_018666.1	aggD	neg		aggregative adherence fimbria I chaperone AggD	8	2
21	NC_018658.1		neg	True	hypothetical protein	9	2
22	NC_018658.1		neg		phage protein	9	2
27	NC_018666.1	traX	neg		conjugal transfer pilus acetylase TraX	12	4
29	NC_018666.1	traI	neg	True	conjugative transfer relaxase/helicase TraI	12	4
30	NC_018666.1		neg	True	PAS domain-containing protein	12	4
32	NC_018658.1		neg		hypothetical protein	13	2
36	NC_018658.1		neg		IS6-like element IS26 family transposase	15	3
39	NC_018658.1		neg		phage tail protein	16	3

Figura 16: Tabla principal de BacDupWeb

En el ejemplo de la Figura 16, vemos en concreto que mostramos 25 registros de la tabla y que tenemos el filtro de hebra fijada a negativa además de no incluir fagos y elementos móviles.

La agilidad e inmediatez con la que la tabla se actualiza y con la que podemos clasificar o buscar conceptos en la caja *search*, consideramos que demuestran la enorme utilidad de **BacDupWeb** y la acercan al objetivo buscado.

3.3.6 Bloque Resultados: Gráfico Circos de genes duplicados

En este apartado nos situamos en la segunda pestaña del menú, el gráfico circos. En efecto, se trata de la posibilidad de mostrar un gráfico circos tal como ya aparece en **BacDup** pero con la particularidad que podemos mostrarlo inmediatamente según los archivos cargados pero, además, se actualiza según los filtros seleccionados, lo que hace de esta función un **Biocircos** dinámico.

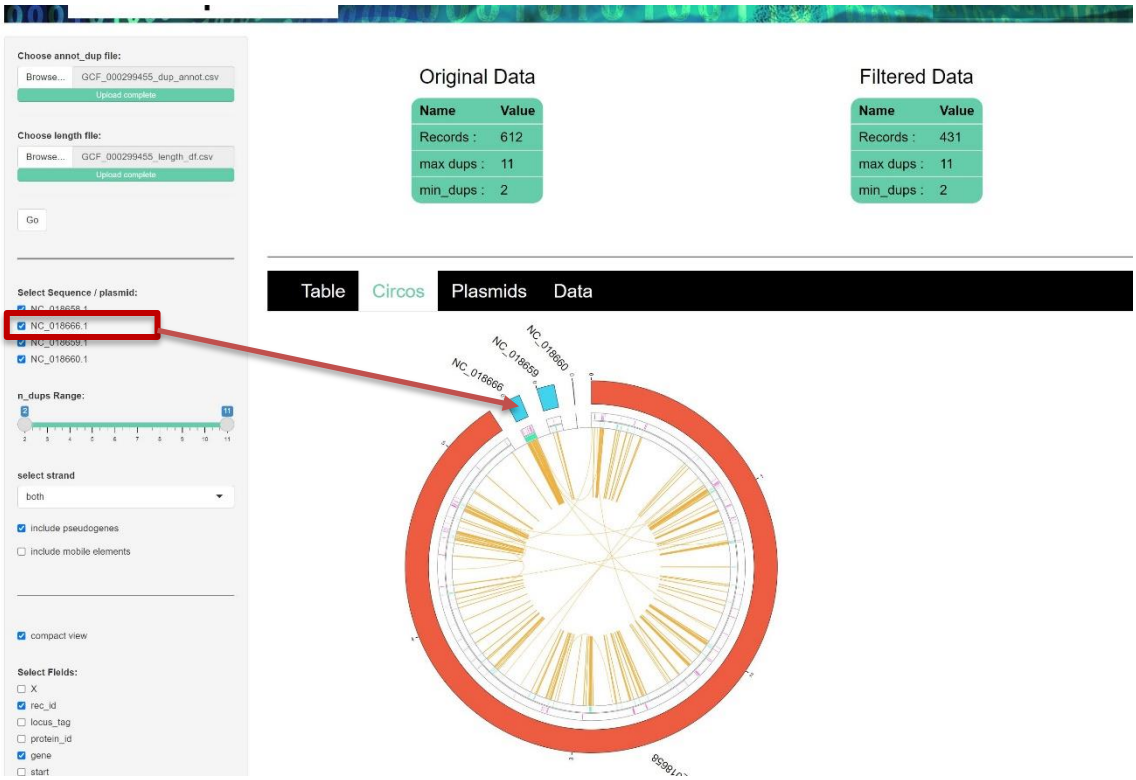


Figura 17: Gráfico Circos dinámico

Es importante informar que el script de este gráfico se ha obtenido de BacDup e incluido en esta aplicación mediante un archivo externo llamado **function_ext.R** que es llamado al principio de la aplicación. Con pequeñas adaptaciones y haciendo la llamada adecuada, conseguimos crear el gráfico circos en función de la cepa de estudio.

Para realizar el gráfico circos son necesarios los dos archivos cargados, dup_annot y length. Mediante el archivo length se define la estructura principal del arco, con el cromosoma principal en color naranja y los plásmidos en azul claro. A partir de ahí, el archivo dup_annot indica todas las relaciones de duplicados mediante líneas que enlazan una copia con la otra de cada gen duplicado.

Si actuamos sobre los filtros del panel lateral, podemos llegar a ver, por ejemplo, el gráfico circos con un solo gen y sus cinco copias, y donde están ubicadas. BacDupWeb aprovecha un script ya desarrollado para incluirlo como función externa a la aplicación y eso forma parte del desarrollo en equipo de potentes aplicaciones.

Es relevante comentar que, a diferencia de las tablas y resto de gráficos de la aplicación que tienen una reacción inmediata, **Biocircos** requiere de entre uno y 5 segundos para mostrar los cambios en los filtros. Considerando la

complejidad del gráfico, no se ha considerado un tiempo excesivo para la aplicación.

En la Figura 18 podemos ver un ejemplo de gráfico Biocircos que responde a la selección de genes que tengan 7 copias. En este caso existen dos genes con 7 copias para un total de 14 y se puede ver su distribución a lo largo del genoma. Es habitual encontrar dos o más copias muy cercanas pero en este gráfico podemos ver con las líneas amarillas que en dos casos las copias están muy alejadas.

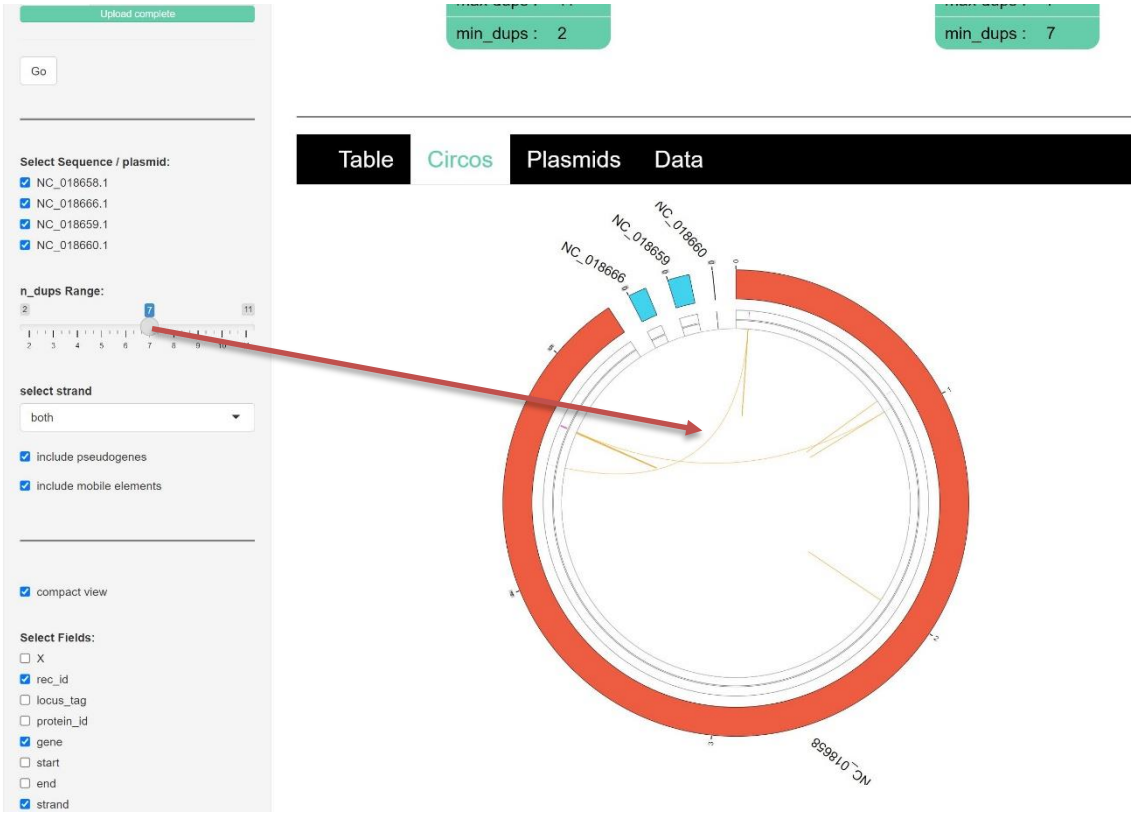


Figura 18: Biocircos de genes con 7 copias

3.3.7 Bloque Resultados: Tabla taxonómica y de cromosomas

En la tercera pestaña del menú se han querido mostrar los datos generales de la cepa. Estos datos no son reactivos, es decir, no varían con los filtros. Son simplemente informativos. Comentar que estos datos se obtienen directamente de Internet, en este caso del repositorio de **GitHub** de **BacDup**.

Para realizar la captura de los datos se lanza una petición a GitHub con la dirección y el identificador de la cepa en GenBank y nos devuelve un archivo de texto plano con los datos:

https://raw.githubusercontent.com/JFsanchezherrero/BacDup/main/developer/data_output/data/**GCF_000299455.1**/input/info.csv

El resultado es el siguiente:

```
,GCF_000299455.1

folder,/home/jfsanchez/git_repos/BacDup/developer/test/db_folder/refseq/bacteria/GCF_000
299455.1

genus,Escherichia

species,Escherichia coli O104:H4 str. 2011C-3493

taxonomy,Bacteria;Proteobacteria;Gammaproteobacteria;Enterobacterales;Enterobacteriaceae
;Escherichia

genome,

annot_file,/home/jfsanchez/git_repos/BacDup/developer/test/db_folder/refseq/bacteria/GCF
_000299455.1/GCF_000299455.1_ASM29945v1_genomic.gbff

format_annot_file,gbk

proteins,

plasmids_number,0

plasmids_ID,
```

Manipulando este archivo con un script de R, gracias a su estructura uniforme sea cual sea la cepa, y seleccionando las líneas y caracteres adecuados, nos permite mostrar los datos de taxonomía en función de la cepa de estudio.

Por otra parte podemos obtener directamente del archivo **length.csv** los datos del cromosoma, plásmidos y longitud de las secuencias. En la figura siguiente vemos las dos tablas resultantes, que como se ha comentado, en este caso no son reactivas a los filtros y quedan determinadas desde el momento que cargamos los archivos iniciales.

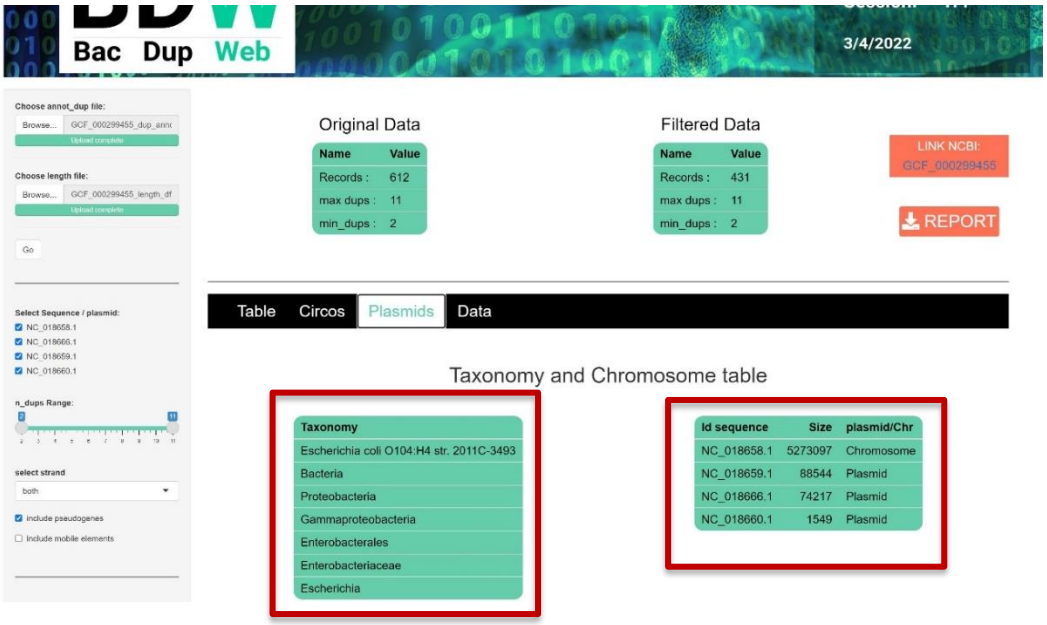


Figura 19: BacDupWeb apartado datos cepa

3.3.8 Bloque Resultados: Frecuencia de genes duplicados

La cuarta y última pestaña de la aplicación es la que hace referencia al detalle de los genes duplicados en cuanto a su frecuencia. En concreto mostramos la distribución de duplicados en función del número de copias que presentan, siempre empezando con los genes con 2 copias y terminando en genes con hasta 75 copias según la cepa.

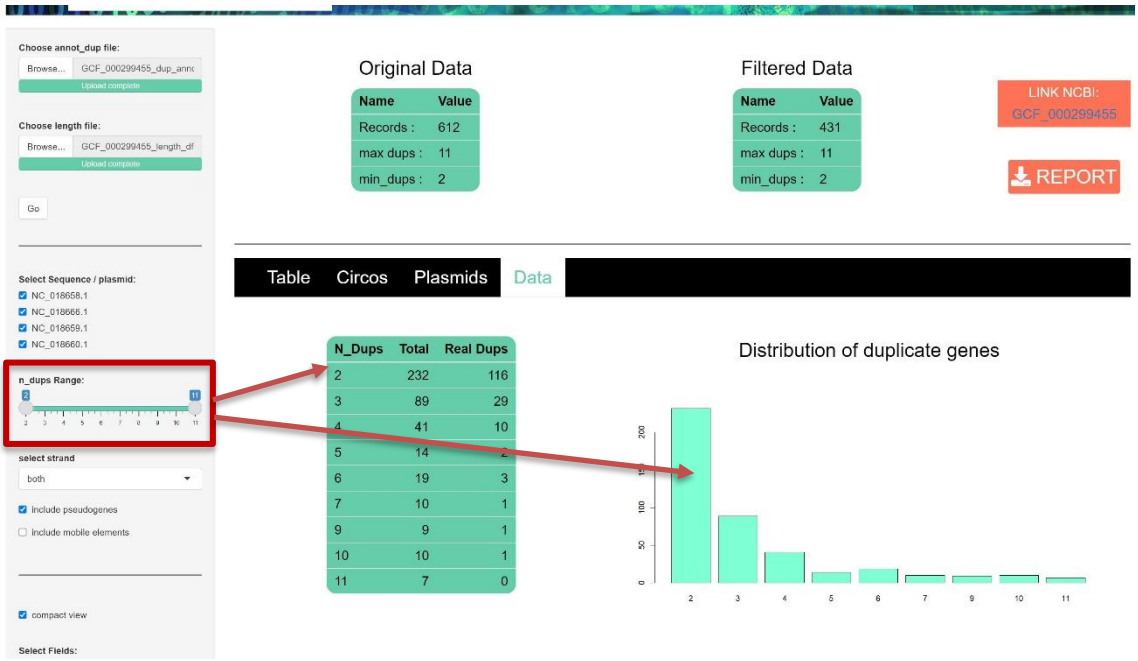


Figura 20: BacDupWeb frecuencia del número de duplicados

En el ejemplo de la Figura 20 vemos que hay 232 registros que hacen referencia a genes con 2 copias. En realidad son justo la mitad, 116 genes con 2 copias que se corresponden con 232 registros. En el archivo original siempre encontraremos que los genes con varias copias tienen frecuencia múltiple del número de copias, pero es cierto que eso se distorsiona a medida que aplicamos filtros como por ejemplo si desactivamos los pseudogenes.

Además de la tabla frecuencial se ha incluido un barplot en función del número de copias. La tabla de duplicados y este gráfico frecuencial sí reaccionan a los filtros aplicados y lo hacen de forma inmediata. Es fácilmente comprobable manipulando el slider de mínimo y máximo número de copias.

3.3.9 Bloque in/out: Funciones de intercambio de datos

Como bloque final tenemos dos funcionalidades complementarias que consisten en la posibilidad de bajarse un informe según la selección y resultados obtenidos mediante un botón de *download* y la posibilidad de consultar los datos de la cepa en **GenBank**, más allá de los mostrados en la pestaña anterior del menú. Mediante el botón link NCBI, la aplicación crea una **dirección url** con el código de la cepa y nos permite viajar a una página web de NCBI con toda la información completa, algunos ya mostrados en BacDupWeb y otros que no.

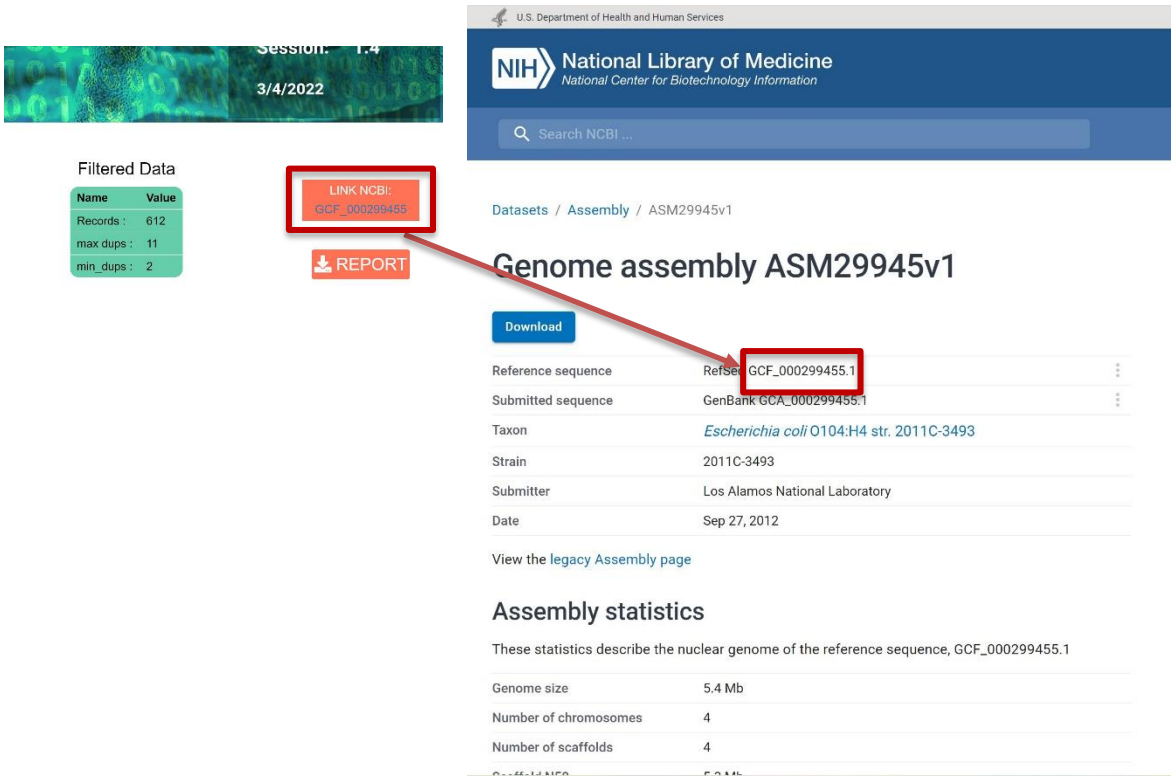


Figura 21: link a ncbi RefSeq / GenBank

3.4 Puesta en producción de BacDupWeb

Aunque R Studio y Shiny tienen la posibilidad de testear la aplicación mediante un servidor local (localhost) en el propio ordenador, se ha considerado interesante también desplegar la aplicación, poniéndola en producción en un servidor en Internet. Para ello existe el entorno Shinyapps.io que permite publicar aplicaciones Shiny de forma muy sencilla desde R Studio [21]. De este modo la aplicación puede ser compartida con el propio tutor, con el equipo de BacDup y con los propios evaluadores si lo consideran. El enlace es el siguiente:

https://bacdupweb.shinyapps.io/BacDupWeb8/

Este apartado podría ir al final de los resultados pero se ha considerado citarlo aquí como fase final de la aplicación desarrollada. En realidad, poner en producción cualquier aplicación web, consiste en el proceso de montarla en un servidor público accesible por cualquier usuario que lo dese, desde todo el mundo y desde cualquier ordenador.

4 Discusión

4.1 Reflexión sobre BacDup y BacDupWeb

Analizando los resultados, en este apartado hacemos una serie de reflexiones sobre la aplicación. Estas observaciones han surgido en el proceso de desarrollo y una vez visto el resultado final.

En la fase inicial de la planificación se consideró muy importante que **BacDupWeb** partiera de los archivos originales *Fasta* de **GenBank** o incluso de las secuencias obtenidas de novo, de forma que el procedimiento creado por **BacDup**, se integrara también dentro de la aplicación web. No obstante, posteriormente se ha visto que esta parte, la de obtener el archivo de duplicaciones, es un proceso previo a la aplicación y que puede llegar a realizarse de forma asíncrona. En efecto, una vez validado el pipeline de **BacDup**, se podría crear un repositorio con todos los archivos dup_annot de todas las cepas conocidas y trabajar sobre estas. En el momento que aparezca una nueva cepa, se pasa por **BacDup** y se guarda su correspondiente archivo dup_annot. Según como no tendría sentido generar cada vez un archivo dup_annot que ya está determinado de salida.

Es a partir de estos archivos **dup_annot** que pueden empezar miles de estudios a partir de consultas sobre dichos archivos, buscando información sobre un gen u otro, sobre genes que tienen más de diez copias, o estudiando genes que solo están en la hebra negativa y que no incluyan pseudogenes. Por lo tanto es normal que el investigador empiece en **BacDupWeb** tal como se ha definido y no generando un archivo de duplicados, proceso que se puede hacer mediante **BacDup** previamente. Eso si, **BacDup** podría tener una aplicación para generar archivos dup_annot a partir de secuencias Fasta y eligiendo, por ejemplo, un comparador entre varios y un nivel de similitud regulable. En cualquier caso he llegado a la conclusión que se trata de una aplicación distinta.

Hay otro motivo técnico para no abordar la parte de **BacDup** y es que trabajar con **Blast** en local es relativamente sencillo, pero no lo es desde un servidor. Para poder hacerlo hay que crear un entorno de trabajo en la nube que contenga las funcionalidades de **Blast** y eso es complejo y largo de ejecutar con lo que no se ha considerado incluir en este TFM.

4.2 Velocidad reactiva y estética de Shiny

La elección del framework **Shiny** era una elección arriesgada en un inicio por el hecho que los ejemplos encontrados parecían muy sencillos y, una vez se trabajaba con archivos más pesados, el resultado pudiera no ser el esperado. El caso es que esto no ha ocurrido y **Shiny** ha cumplido con las características de un sistema de programación que reacciona rápidamente a cambios en los parámetros y filtros.

Respecto a la estética de **Shiny**, que en un momento inicial nos pareció una estética académica y poco profesional, debe reconocerse que, las indicaciones sobre el uso de CSS y JS, han permitido modificar la apariencia y se ha podido diseñar un entorno muy personalizado para la aplicación.

4.3 Líneas de futuro

Se puede considerar que **BacDupWeb** sería más potente aún si la carga de archivos pudiera ser directamente de archivos procedentes de **GenBank**, en base a las secuencias originales y que el proceso de comparación y búsqueda de duplicados fuera integrado en la misma aplicación. Esta funcionalidad sería especialmente interesante en el caso de secuenciación de novo de patógenos pues hay que pasar dichas secuencias por **BacDup** una vez obtenido el ADN.

Otra línea de futuro sería la posibilidad de trabajar con varias cepas e incluso especies distintas de forma que, en lugar de buscar genes o proteínas duplicadas dentro del mismo genoma o de sus plásmidos, se estudiaran duplicaciones entre genomas distintos de especies distintas de bacterias o, porque no, de arqueas y procariotas en general.

5 Valoración económica

Hay que comentar que cualquier aplicación web de uso compartido y generalista requiere de un servidor de acceso público. Para el desarrollo académico es cierto que se suelen encontrar cuentas gratuitas con limitaciones funcionales, con limitaciones en el peso de los archivos tratados, limitaciones de tiempo de uso, etc.

Pero si se desea poner en producción aplicaciones como **BacDupWeb** de forma general, consultable por varios usuarios y sin limites temporales, hay que disponer de servidores mínimamente potentes, con un potencial flujo simultáneo de usuarios y con una cierta capacidad de procesador y memoria, aparte de la del propio ordenador del usuario. En este sentido **BacDupWeb** requerirá del pago de una cuota mensual del servidor donde se ponga en producción. Como ejemplo se adjuntan las tarifas del servidor de **Shinyapps.io** en la fecha de elaboración de este documento. En concreto dispone de 5 modalidades de uso.

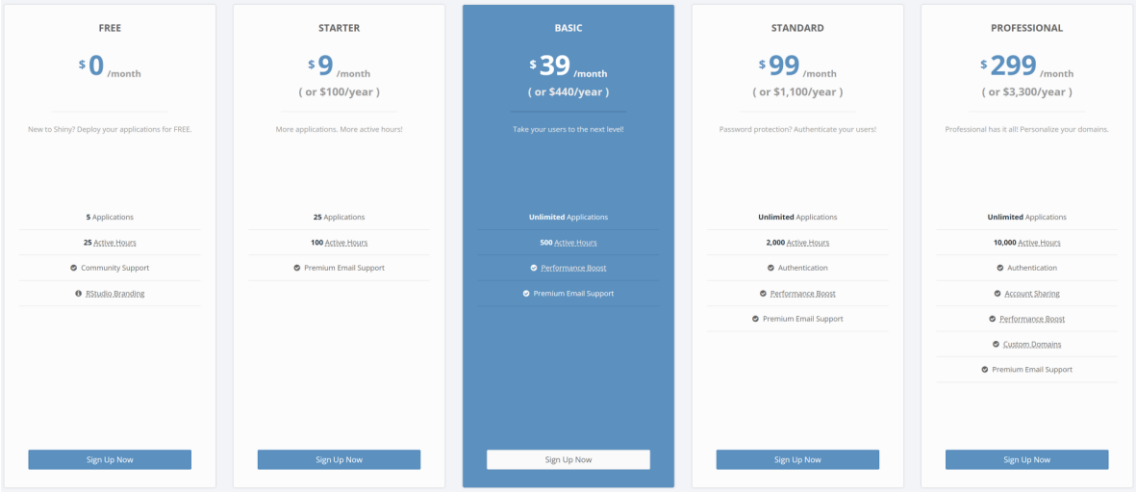


Figura 22: Tabla de precios del servidor Shiny Apps en junio de 2022

Para este desarrollo se ha utilizado la cuenta gratuita que permite disponer de hasta 5 aplicaciones simultáneas pero con la limitación de 25 horas de uso al mes. Gracias a que **RStudio** con **Shiny** ya dispone de un servidor localhost, las pruebas del día a día se pueden realizar en local sin la presión de las 25 horas. Para realizar la puesta en producción para compartir con el tutor, para confirmar que en servidor público de **Shiny** dispone de toda la infraestructura informática para el correcto funcionamiento, para acceder desde otros ordenadores vía navegador, en principio puede considerarse que las 25 horas mensuales son suficientes para un desarrollo académico, al menos en nuestro caso ha sido así.

6 Conclusiones

En líneas generales, el objetivo inicial de crear una aplicación web para el análisis de duplicidades génicas, se ha conseguido. El resultado final obtenido abarca las cuatro grandes características que una aplicación web debe cumplir que son la estética, la funcional, la de usabilidad y el de una alta velocidad de consulta de los datos. Esta última característica, la de la velocidad de consulta de los datos según el cambio en los filtros, que era la prioridad en el diseño, consideramos que se ha logrado con éxito, al menos con todos los ejemplos testeados.

En cuanto a la parte más personal, es importante destacar que este trabajo ha permitido abordar muchos conceptos en un mismo proyecto y eso consolida los conocimientos adquiridos durante el máster. Es interesante repasarlos:

En cuanto a programación:

- Análisis de herramientas como Django de Python, Mysql, Flask, SQlite
- Programación en Shiny
- Programación de scripts de R
- Mejorar estética de páginas web con CSS, HTML y JS
- Carga y manipulación de archivos CSV

En cuanto a intercambio de información:

- Realizar links a bancos de datos como NCBI en función de nuestros archivos
- Importar datos de repositorios GitHub
- Realizar informe descargable mediante estructura R Markdown

En cuanto a entorno de trabajo:

- Trabajar con control de versiones GitHub
- Trabajar con servidor localhost de R Shiny
- Trabajar con editor de programación R Studio (y también Visual Studio Code)
- Desplegar en producción con shinyApps.io en servidor público
- Conocer un gestor de bibliografía como Mendeley
- Aprender sobre el formato y buenas prácticas de los artículos científicos.

En cuanto a conocimientos biológicos:

- Aprender la estructura genómica de las bacterias.
- Conocer la problemática de las infecciones nosocomiales.
- Conocer las duplicaciones génicas, su número y frecuencia y su potencial implicación en la virulencia de las infecciones.

Respecto la planificación, como ya se ha comentado, se centraron los esfuerzos en ampliar las funcionalidades de la aplicación más allá de la tabla principal en lugar de integrar en la aplicación el pipeline de **BacDup** partiendo de las secuencias originales. Esta ha sido una de las diferencias pero consideramos que ha sido una buena elección dicha priorización e invertir en una visualización dinámica que no existía, en lugar de una funcionalidad que ya existe aunque no sea de tipo web.

Otro de los objetivos que no se ha abordado, en este caso por tiempo, es la posibilidad de combinar la información de 2 cepas e incluso más para buscar duplicaciones cruzadas, es decir, no dentro del mismo genoma si no comparando genomas entre distintas cepas e incluso distintas especies.

Un último apunte hace referencia en que esta aplicación sería interesante que cada usuario pudiera registrarse e incluso tener sus *settings* habituales de uso. Estas secciones de administración formarían parte del desarrollo evolucionado hacia una aplicación profesional multiusuario.

7 Glosario

BacDup: Conjunto de trabajos, artículos, investigaciones y desarrollos para el análisis de genes duplicados mediante un pipeline y que han formado parte de un estudio publicado.

BacDupWeb: Aplicación web de visualización dinámica de datos desarrollada a partir del trabajo BacDup.

Infección Nosocomial: Infección bacteriana adquirida a partir de la fecha del ingreso en el hospital (y no antes). Aunque pueda existir un cierto solape, la condición de nosocomial se establece bajo la condición de que el paciente ingresó por motivos distintos a la infección detectada en el hospital.

CNV: Acrónimo de *Copy Number Variation* y que hace referencia al número de copias que presenta un gen o un tramo de adn a lo largo de un genoma, ya sean idénticas, ya sean similares en un porcentaje elevado.

Pseudogenes: Se trata de secuencias de adn que responden a la estructura de un gen pero que no codifican a proteína, o en general, no presentan funcionalidad alguna, al menos conocida o detectada.

Fagos: Se trata de secuencias de virus formados por material genético de adn o arn, que infectan a las bacterias.

Sintenia: Sintenia o bloques de sintenia, hacen referencia a tramos de adn de un genoma, que los encontramos idénticos o con pocas variaciones en otra parte del mismo genoma o en genomas de individuos o especies distintas. Estos tramos pueden ser desde genes enteros a secuencias de pocas bases y nos estarán indicando el camino evolutivo que han podido tener dichos genes entre especies.

Framework: Entorno de programación creado con un objetivo concreto, que facilita el desarrollo de una parte del código, con librerías ya predefinidas con funciones de alto nivel muy concretas. Además de las librerías suelen ir acompañados de utilidades complementarias.

Frontend, Backend: Conceptos de programación muy generales pero particularmente usados en los desarrollos web, que hacen referencia a la parte visual, a la parte que interacciona con el usuario y a la parte de procesado de la información y preparación de tablas y gráficos.

Python: lenguaje de programación de alto nivel, con una sintaxis considerada de las más sencillas del mercado, con una amplia comunidad de usuarios, subiendo en los ránquines de lenguajes más usados en el mundo, base de la mayoría de aplicaciones de inteligencia artificial, y de código libre.

Mysql, SQLite: Gestores de bases de datos

Shiny: Se trata de un framework diseñado para realizar páginas web a partir de lenguaje de programación R. Incluye por defecto un bloque UI de frontend, un bloque server de backend y utilidades como un servidor local para realizar pruebas en tiempo real.

Django: Framework para realizar desarrollos web con lenguaje de programación Python. Es el equivalente de Shiny en R. También dispone de un servidor en tiempo de ejecución.

Dashboard: Concepto que engloba las aplicaciones que muestran datos en tiempo real basados en tablas y gráficos que se adaptan al cambio de los datos. La idea concreta es la de monitorizar datos de forma clara y fácil de entender, con colores vistosos y distribución de conceptos bien diseñada.

CSS, JS: Lenguajes de programación complementarios a HTML para mejorar la estética del código original HTML de presentación de páginas web.

Pipeline: Protocolo de pasos a seguir para realizar un estudio desde los datos iniciales hasta el resultado final, de modo que una vez diseñado el proceso, éste queda definido y para cada conjunto de datos y parámetros de entrada, el pipeline nos da un resultado potencialmente distinto al final.

8 Bibliografía

[1] R. Zaragoza, P. Ramírez, and M. J. López-Pueyo, “[Nosocomial infections in intensive care units],” *Enfermedades infecciosas y microbiologia clinica*, vol. 32, no. 5, pp. 320–327, 2014, doi: 10.1016/J.EIMC.2014.02.006.

[2] M. B. Zamora, D. Edecio, S. Zamora, and V. Morales Pérez, “Infección nosocomial. Un importante problema de salud a nivel mundial,” *Rev Latinoam Patol Clin Med Lab*, vol. 62, no. 1, pp. 33–39, 2015, Accessed: Feb. 21, 2022. [Online]. Available: www.medigraphic.com/patologiaclinicawww.medigraphic.org.mx

[3] P. M. Olaechea, J. Insausti, A. Blanco, and P. Luque, “Epidemiología e impacto de las infecciones nosocomiales,” *Medicina Intensiva*, vol. 34, no. 4, pp. 256–267, May 2010, doi: 10.1016/J.MEDIN.2009.11.013.

[4] C. J. Murray *et al.*, “Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis,” *The Lancet*, vol. 399, no. 10325, pp. 629–655, Feb. 2022, doi: 10.1016/S0140-6736(21)02724-0/ATTACHMENT/B227DEB3-FF04-497F-82AC-637D8AB7F679/MMC1.PDF.

[5] O. Brynildsrud, S. Gulla, E. J. Feil, S. F. Nørstebø, and L. D. Rhodes, “Identifying copy number variation of the dominant virulence factors msa and p22 within genomes of the fish pathogen Renibacterium salmoninarum,” *Microb Genom*, vol. 2, no. 4, p. e000055, Apr. 2016, doi: 10.1099/MGEN.0.000055/CITE/REFWORKS.

[6] R. Jugas *et al.*, “CNproScan: Hybrid CNV detection for bacterial genomes,” *Genomics*, vol. 113, no. 5, pp. 3103–3111, Sep. 2021, doi: 10.1016/J.YGENO.2021.06.040.

[7] D. Gevers, K. Vandepoele, C. Simillion, and Y. van de Peer, “Gene duplication and biased functional retention of paralogs in bacterial genomes,” *Trends in Microbiology*, vol. 12, no. 4, pp. 148–154, 2004, doi: 10.1016/J.TIM.2004.02.007.

[8] M. S. Bratlie, J. Johansen, B. T. Sherman, D. W. Huang, R. A. Lempicki, and F. Drabløs, “Gene duplications in prokaryotes can be associated with environmental adaptation,” *BMC Genomics*, vol. 11, no. 1, p. 588, Oct. 2010, doi: 10.1186/1471-2164-11-588.

[9] J. F. Sanchez-Herrero, M. Bernabeu, A. Prieto, M. Hüttener, and A. Juárez, “Gene Duplications in the Genomes of Staphylococci and Enterococci,” *Frontiers in Molecular Biosciences*, vol. 7, p. 160, Jul. 2020, doi: 10.3389/FMOLB.2020.00160/BIBTEX.

[10] M. Bernabeu *et al.*, “Gene duplications in the E. coli genome: Common themes among pathotypes,” *BMC Genomics*, vol. 20, no. 1, pp. 1–11, Apr. 2019, doi: 10.1186/S12864-019-5683-4/FIGURES/6.

[11] "JFsanchezherrero/BacDup: Bacterial gene duplication analysis pipeline." https://github.com/JFsanchezherrero/BacDup (accessed May 26, 2022).

[12] C. Camacho *et al.*, "BLAST+: Architecture and applications," *BMC Bioinformatics*, vol. 10, Dec. 2009, doi: 10.1186/1471-2105-10-421/1471_2105_10_421_PDF.PDF.

[13] "albamgarces/TFM_UOC_AMoya: Bioinformatics Master's dissertation in progress." https://github.com/albamgarces/TFM_UOC_AMoya/ (accessed May 28, 2022).

[14] "What Is GitHub? A Beginner's Introduction to GitHub." https://kinsta.com/knowledgebase/what-is-github/ (accessed Jun. 02, 2022).

[15] K. D. Pruitt, T. Tatusova, G. R. Brown, and D. R. Maglott, "NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy," *Nucleic Acids Research*, vol. 40, no. D1, Jan. 2012, doi: 10.1093/NAR/GKR1079.

[16] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "GenBank," *Nucleic Acids Research*, vol. 36, no. SUPPL. 1, Jan. 2008, doi: 10.1093/NAR/GKM929.

[17] "Framework: qué es, para qué sirve y algunos ejemplos." https://www.edix.com/es/instituto/framework/ (accessed Jun. 02, 2022).

[18] "Curso Django - YouTube." https://www.youtube.com/playlist?list=PLU8oAlHdN5BmfvwxFO7HdPciOCmmYneAB (accessed May 31, 2022).

[19] "GRÁFICOS en R 📈 [TUTORIALES de todos los tipos de GRÁFICAS]." https://r-coder.com/graficos-r/ (accessed Jun. 01, 2022).

[20] "Shiny - Shiny HTML Tags Glossary." https://shiny.rstudio.com/articles/tag-glossary.html (accessed Jun. 01, 2022).

[21] "Cómo poner una App Shiny en Producción - Ander Fernández." https://anderfernandez.com/blog/poner-app-shiny-en-produccion/ (accessed Jun. 01, 2022).

9 Anexo: Repositorio GitHub

En el repositorio de BacDupWeb en GitHub podemos encontrar toda la información, archivos e instrucciones para trabajar con BacDupWeb. En el documento *Readme* del repositorio se describe todo el contenido. Al repositorio se puede acceder mediante la siguiente dirección:

https://github.com/dibanezmal/BACDUPWEB

Aunque en el archivo ReadMe se describen odos los elementos del repositorio, hemos considerado incluirlos también en la memoria. En GitHub encontraremos lo siguiente:

- El código Shiny/R de BacDupWeb con sus funciones e imágenes externas.
- La dirección de shinyapps.io donde podemos usar la aplicación.
- 5 juegos de archivos (dup_annot y length) correspondientes a 5 cepas.
- Un manual de uso de BacDupWeb
- La presente memoria en pdf
- Capturas, imágenes auxiliares y figuras de la memoria.