Uso de algoritmos de aprendizaje automático aplicados a bases de Datos genéticos (HapMap)

Jorge Pulido Lozano
Master Universitario en Bioinformática y Bioestadística - Programación para la bioinformática

María Jesús Marco Galindo, Pau Andrio Balado, Brian Jiménez García

26/12/16

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento 3.0 España de Creative Commons

B) GNU Free Documentation License (GNU FDL)

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos (HapMap)
Nombre del autor:	Jorge Pulido Lozano
Nombre del consultor:	María Jesús Marco Galindo, Pau Andrio Balado, Brian Jiménez García
Fecha de entrega (mm/aaaa):	12/2016
Área del Trabajo Final:	Programación para la Bioinformática (TFM_bio_A3)
Titulación:	Master Universitario en Bioinformática y Bioestadística - (interuniversitario: UOC, UB)

Resumen del Trabajo (máximo 250 palabras):

En los últimos años el volumen de información relativa a genoma humano se ha visto incrementado de manera exponencial, obligando al desarrollo de bases de datos biológicas y herramientas de computación para su análisis. La magnitud y complejidad de los datos conlleva la aparición de técnicas de aprendizaje automático que permite obtener nueva información relevante y abrir nuevas vías de investigación.

En el presente trabajo se ha aplicado un algoritmo de aprendizaje automático basado en la información contenida en una serie de bases de datos relacionales de consultas estructuradas (MySQL), con información de los SNP´s de todo el genoma de 11 grupos étnicos. Dicho algoritmo escrito en lenguaje R permite la clasificación de los datos en clúster según la semejanza de estos, y el análisis estadístico para determinar la eficacia de la clasificación. El sistema está diseñado para operar sobre un servidor web en lenguaje PHP donde el usuario puede ver los resultados obtenidos por la ejecución del algoritmo.

El algoritmo permite una correcta clasificación de los individuos por su grupo étnico, de modo que el sistema puede predecir con una precisión del 90% del valor Kappa de Cohen, el grupo étnico al que pertenece un individuo solamente con la información de sus SNP´s.

El algoritmo puede utilizarse para identificar el grupo étnico y el núcleo familiar de un individuo según sus SNP´s suponiendo una mejora al requerir menor información frente a la identificación por micro satélites (STR).

Abstract (in English, 250 words or less):

In recent years, the volume of information about the human genome has been increasing exponentially, driving the biological databases' development and computer tools for their analysis. Due to the amount and complexity that the data carries, it was

necessary to create techniques for automatic learning which allow obtaining new patterns and investigation´s lines.

In the present work, an automatic learning algorithm has been applied to a database, built in MySQL, with all SNPs detected in the human genome from 11 ethnic groups. The code, written in R, allows a cluster analysis and a graphical representation base on the similarity of the SNPs as well as a statistical test to determine the classification training accuracy. The whole system is designed to work in a user-friendly PHP web server where the user can see and interact with the data through the code.

The algorithm allows for an almost perfect classification built on their ethnic group, so the training can perform a good prediction with more than 90% accuracy of Cohen's Kappa value, concerning the use of SNPs to differentiate the population

The algorithm used in this work can be used to identify the ethnic group plus the family which one person belongs due to theirs SNPs. This is quite a small improvement since it didn´t need as much genetic information as the microsatellite identification does.

Palabras clave (entre 4 y 8):

HapMap, PHP, SNP´s, Aprendizaje automático, minería de datos, MySQL, R, bioinformática, bioestadística

Índice

Lista de figuras

Introducción

Contexto y justificación del Trabajo

El trabajo correspondiente al trabajo final de master (en adelante TFM) se ha desarrollado para crear un portal web para visualizar modelos de clasificación y de predicción sobre el conjunto de datos del consorcio HapMap.

En la actualidad, el avance que se está produciendo en áreas de la biología y/o medicina son remarcablemente altos lo que supone que el número de datos producidos en dichos avances van a la par. Este volumen masivo de información debe procesarse para que los investigadores puedan sacar conclusiones de los mismo.

Por este motivo el desarrollo de la bioinformática ha tenido que desarrollarse en paralelo al de las investigaciones para poder procesar los conjuntos de datos en tiempos menores, como puede ser el caso del desarrollo de superordenadores (p.ej. MareNostrum, Barcelona). Sin embargo, el acceso a dichos ordenadores no está al alcance de toda la comunidad científica por lo que es necesario el desarrollo de aplicaciones web almacenadas en servidores para compensar esta carencia (p.ej. Swissmodel MoDEL).

El desarrollo de esta aplicación nace de la necesidad de procesar y tratar los datos obtenidos en el proyecto HapMap de una forma sencilla e intuitiva para extrapolar nuevas conclusiones acerca del proyecto. Para ello se desarrollará una aplicación web que correrá en un modo local del ordenador para inferir nuevas conclusiones sobre el conjunto de datos de HapMap.

El objetivo del proyecto HapMap el desarrollo de un mapa de los haplotipos humanos, que describen los patrones más comunes que originan la variación génica humana. Para ello un consorcio de universidades y empresas privadas de todo el mundo desarrollaron técnicas nuevas para la secuenciación las zonas cromosómicas donde se encuentran los haplotipos.

Las enfermedades según estudios presentan una causa multi-génica, la diferencia en un haplotipo entre dos personas afectadas puede arrojar luz a cerca del riesgo a padecer dicha enfermedad. El uso de la información contenida en HapMap se utilizará para determinar cuáles son los genes que afectan a la salud. Por este motivo es importante HapMap, por que permitirá comprender las complejas causas detrás de las enfermedades y así detectar quien podría ser más susceptible de padecer diabetes (p. ej.) si se comparan los haplotipos de dos personas, uno que sufra de diabetes y otro que no. Lamentablemente, el proyecto HapMap ha sido cancelado debido a fallos en la seguridad y al descenso de la participación en el proyecto en los últimos años [1].

Para llevar a cabo el uso de técnicas de aprendizaje automático (en adelante ML) sobre las que se fundamente el TFM será necesario el desarrollo de una base de datos (en adelante BBBDDD) que alberge el repositorio de HapMap y sobre el que se ejecutara ML para verse en la aplicación web. La aplicación web permitirá la lectura y búsqueda de los distintos haplotipos según unos parámetros establecidos/seleccionados por parte del usuario, además, de realizar la clasificación de los datos según una variable establecida previamente por el programador web basado en un análisis estadístico del funcionamiento del ML.

De este modo el marco teórico sobre el que gira este TFM es el aprendizaje de técnicas de programación web/BBBDDD, así como de ML para poder realizar búsquedas/predicciones sobre un conjunto de datos. Así, el programador web aplicará el uso de un algoritmo que permitirá la clasificación por la computadora en términos de ML.

Objetivos del Trabajo

1. Objetivos generales

- Introducir y probar los datos de HapMap en una base de datos SQL (MySQL).
- Familiarizarse con los conceptos claves de Machine Learning y minería de datos.
- Desarrollar un algoritmo (R) para realizar Machine Learning sobre los datos y que sean mostrados en la web.
- Diseñar la página web (PHP) sobre la que se realizara el TFM y dotarla de contenido y funcionalidad.

2. Objetivos específicos

- Descargar y Preprocesar los datos para crear las BBBDDD.
- Identificar las mejores maneras para clasificar los datos.
- Cargar los datos en BBBDDD para realizar el algoritmo.
- Realizar el algoritmo para ML.
- Conceptuar la página web en PHP (Diagrama de estructura) y definir las secciones o partes que la compondrán.
- Relacionar la BBBDDD y la página web con el aprendizaje automático.
- Operacionalizar la página para su funcionamiento.

Enfoque y método seguido

Para llegar a cumplir los objetivos descritos anteriormente, se necesitarán conocimientos de programación en lenguaje PHP, así como en dataframes asociados al citado lenguaje. Dichos conocimientos se obtendrán durante el proceso con la ayuda de manuales dedicados al tema [2].

También se necesitarán conocimientos de BBDD estructuradas tipo SQL, programas de entorno de bioestadística R y se obtuvieron conceptos caves de programación en Django, así como los paquetes necesarios y uso de ML en lenguaje Python.

El trabajo consistirá en una separación de áreas donde se abordarán los diferentes asuntos de manera individualizada para posteriormente agruparse dentro de la estructura final (portal web).

Se trabajará en la creación de la página web con PHP para dotarla de contenido y enlaces de interés relacionados al proyecto HapMap, al mismo tiempo se introducirá los datos en la BBDD y se trabajará con ella identificando los campos más significativos para su posterior uso en ML. Además, se realizarán pruebas sobre la BBDD para que posteriormente puedan usarse en la página web cuando la BBDD esté vinculada a la web.

Por último, la dotación de capacidad de ML permitirá una mejoría de la búsqueda de los patrones realizados con la BBDD. La integración de ML se hará a través de un entorno y lenguaje de programación enfocado al análisis estadístico R. La elección de dicho software en pos de otros como Python ha sido la posibilidad de integración con PHP y de un mayor conocimiento en este lenguaje [3].

Ente enfoque por separado es necesario para asegurar una viable realización del proyecto ya que se ahorraría tiempo y recursos para luego dedicárselo a la conexión de las diferentes partes entre la BBDD, el programa lanzadera R y con el portal web.

Así pues, en una primera aproximación se tendera a trabajar con los manuales y recursos para familiarizarse con los programas realizando los tutoriales que ahí contengan, cogiendo una base y moldes para su posterior aplicación, los manuales de usuario/instalación oportunos irán incluidos en la sección de anexos. Las pruebas de familiarización de los distintos componentes del TFM se realizaron utilizando como base el conjunto de datos Iris [4] recogidos por Ronald Fisher y utilizados como referencia en ML.

En una segunda fase se procederá a realizar los moldes de la página web con la información no vinculante a la BBDD (servidor index.php [22] y demás) y al mismo tiempo a realizar búsquedas con la BBDD para poder luego determinar los campos de consulta que el usuario introducirá en la página web para la obtención de los datos.

Durante las primeras dos fases se procedió a la selección de los lenguajes de programación más aptos por el programador para la realización del TFM. En el siguiente capítulo cuando sea oportuno según el área de trabajo, se determinarán las causas que llevaron la selección de un lenguaje en pos de otro.

En la tercera fase, se vincularán los trabajos individuales y se comprobara que el código funciona. Esta fase contempla la depuración del código para su versión final y mejoras de la organización de las diferentes partes que componen el TFM.

La última fase consiste en la redacción de la memoria de trabajo, que aquí compete.

Planificación del Trabajo

1. **Tareas**

    a. Para la BBDD:
        i. Seleccionar la BBDD con la que se va a trabajar (MySQL) y realizar pruebas conceptuales siguiendo los tutoriales/manuales.
        ii. Introducir los datos en la BBDD ya pre procesados tanto para ML como PHP.
        iii. Realizar pruebas para determinar cuál es la mejor manera para clasificar, agrupar los datos, …
    b. Para ML:
        i. Realizar Clustering por medio de Kmeans.
        ii. Análisis del entrenamiento por ML sobre los datos.
    c. Para la página web:
        i. Seleccionar el lenguaje con la que se va a trabajar en la página web (PHP) y realizar pruebas conceptuales siguiendo los tutoriales/manuales.

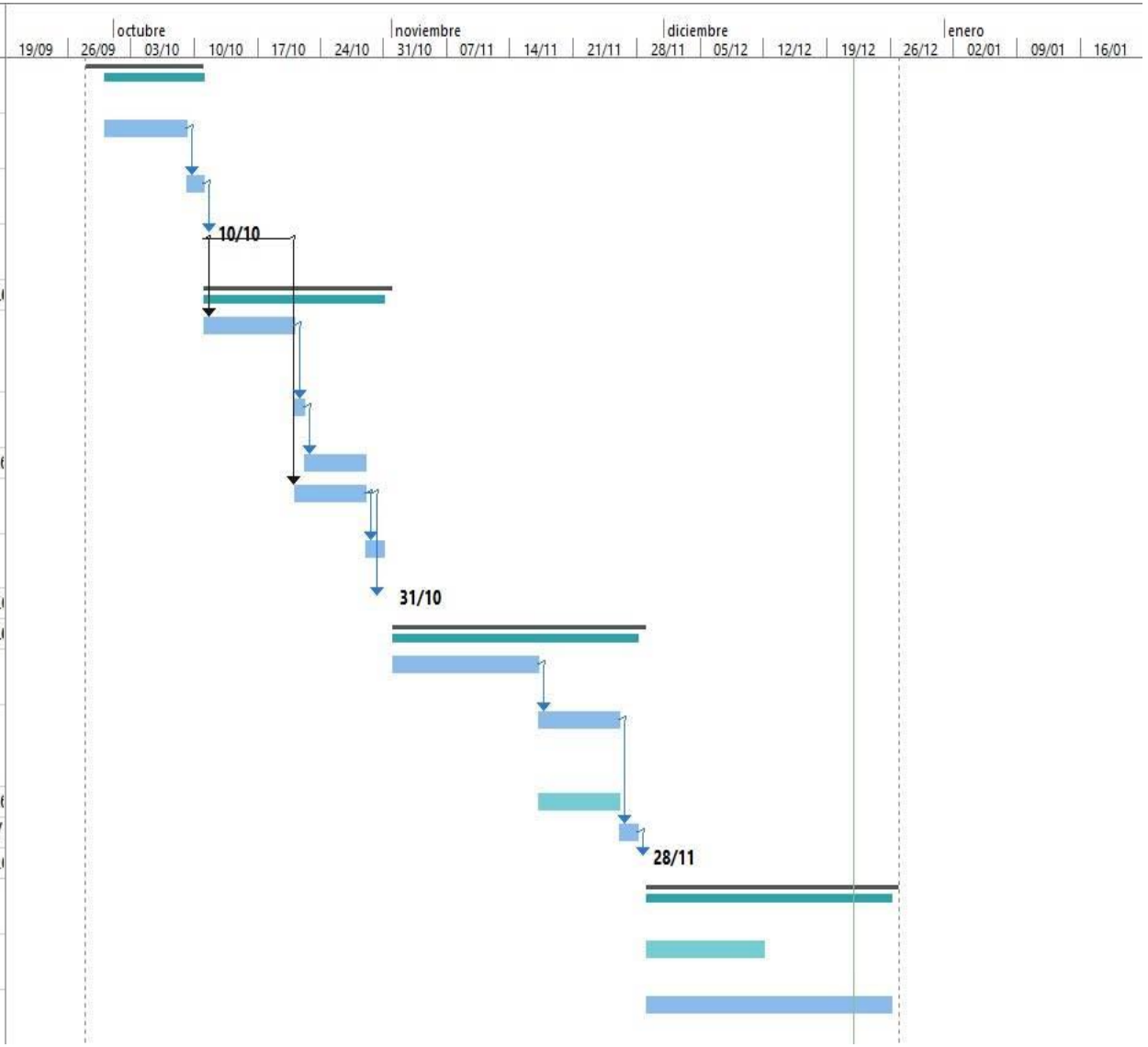
ii. Establecer la disposición general de las distintas partes que formaran la web (Diagrama de estructura).
iii. Llenar de contenido la página web.
iv. Vincular la BBDD con el programa lanzadera y con la página web.
v. Realizar pruebas para observar la estabilidad de la página web y su correcta funcionalidad.

d. Redacción de la memoria de trabajo.

2. **Calendario**

A continuación, se presentan la lista de tareas del presente proyecto, así como su diagrama de Gantt asociado.

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		**Informa Plan de trabajo**	**11 días**	**mié 28/09/16**	**lun 10/10/16**
2		Redactar el informe	7 días	vie 30/09/16	sáb 08/10/16
3		Entregar el informe	2 días	dom 09/10/16	lun 10/10/16
4		Plan de trabajo terminado	0 días	lun 10/10/16	lun 10/10/16
5		**PEC 2**	**18 días**	**mar 11/10/1**	**lun 31/10/1**
6		Introduccion a la BD y Paginas web dinamicas	8 días	mar 11/10/16	jue 20/10/16
7		Introducir datos en la BD	1 día	vie 21/10/16	vie 21/10/16
8		Probar la BD	6 días	sáb 22/10/1	vie 28/10/1
9		Dar de contenido la	7 días?	vie 21/10/16	vie 28/10/16
10		Redactar informe de	2 días?	sáb 29/10/16	dom 30/10/16
11		Entrega PEC 2	0 días	lun 31/10/1	lun 31/10/1
12		**PEC 3**	**22 días**	**mar 01/11/1**	**lun 28/11/1**
13		Vincular BD a la pagina web	12 días	mar 01/11/16	mié 16/11/16
14		Pruebas de depuracion de la pagina	7 días	jue 17/11/16	vie 25/11/16
15		ML con R	7 días	jue 17/11/1	vie 25/11/1
16		Redactar informe	2 días?	sáb 26/11/1	dom 27/11/
17		Entrega PEC 3	0 días	lun 28/11/1	lun 28/11/1
18		**Memoria de trabajo Final**	**21 días**	**mar 29/11/16**	**lun 26/12/16**
19		Vincular R con Php	10 días	mar 29/11/16	dom 11/12/16
20		Escribir la memoria	20 días	mar 29/11/16	dom 25/12/16



3. **Hitos**

Durante la elaboración del TFM se marcaron los siguientes hitos como se recogen en el diagrama de Gantt.

a. Entrega del Plan de trabajo (PEC1).
b. Entrega del PEC2.
    i. Crear una BBDD para que albergue los datos pre-procesados.
    ii. creación del portal web.
c. Entrega del PEC3.
    i. ML con los datos del proyecto HapMap.
    ii. Crear un portal web que permita interactuar con el software desarrollado y visualizar los resultados.
d. Entrega de la memoria final de trabajo (PEC4 – Entrega final).
e. Aprobación del proyecto (Tribunal y calificación)

4. **Análisis de Riesgos**

a. Falta de conocimientos basados tanto en la elección del motor del portal web como del de la BBDD y del programa lanzadera.
b. Abarcar más de lo que el propio proyecto precisa sobre el portal web
c. Entorno general del firmante (p. ej.: circunstancias profesionales).
d. Incidencias externas (perdidas de información, fallos del ordenador)

Breve sumario de productos obtenidos

- PEC1 -> Plan de trabajo.
- PEC2 -> Desarrollo del trabajo - fase 1.
- PEC3 -> Desarrollo del trabajo - fase 2.
- PEC 4 – Entrega final -> Memoria y presentación del proyecto TFM.

Breve descripción de los otros capítulos de la memoria

Portal web para el análisis de HapMap

Descripción de los aspectos más relevante del diseño y desarrollo del proyecto, así como de los productos obtenidos. Este capítulo estará compuesto por los siguientes apartados.

- Datos consorcio HapMap: Esto será la base fundamental del TFM. Su utilización estará referenciada en los demás apartados.
- Bases de datos (BBDD): introducción y manipulación de los datos en una BBDD tipo MySQL. Tanto ML como PHP estarán vinculadas a estas BBDD, para solicitar los datos sobre los que correrá el algoritmo.
- Aprendizaje automático (ML): Uso de un algoritmo en lenguaje R para la agrupación y predicción sobre los datos contenidos en la DB.
- PHP: Visualización de los datos en un entorno web, de modo que el usuario pueda interactuar con los datos de la BS y con el algoritmo propuesto.

Los algoritmos y códigos creados para la realización del TFM serán guardados en el GIT del proyecto [22]. Los archivos almacenados en el GIT representan el trabajo final del TFM, además cuando sea necesario cual se pondrán partes de dichos archivos en su capítulo correspondiente para explicar/justificar el uso del código en referencia al TFM.

Conclusiones

En este apartado se localizarán las conclusiones y reflexiones obtenidas al finalizar el TFM. Las conclusiones elaboradas por el alumno durante el desarrollo de este proyecto, reflejaran no solo las obtenidas por el estudio del mismo sino también las personales.

Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

Bibliografía

Lista numerada de las referencias bibliográficas utilizadas dentro de la memoria y del TFM. En cada lugar donde se utilice una referencia dentro del texto, esta indicara con un numero de referencia, por ejemplo: [1].

Anexos

Listado de apartados que son demasiado largos o extensos para ser incluidos dentro de la memoria. En este capítulo se incluirán los códigos en texto plano tanto de los algoritmos como de la estructura de los servidores PHP con comentarios sobre su interpretación y aplicación.

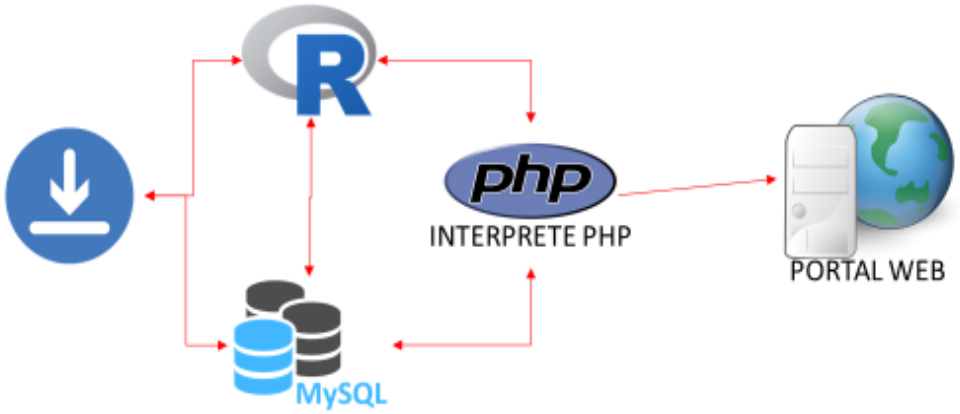
En cada lugar donde se utilice una referencia dentro del texto y asociado a una parte de los anexos, se indicará con una clave de referencia, por ejemplo: [A1].

Portal web para el análisis de HapMap

El TFM fue descompuesto en tareas más sencillas pero interconectadas entre ellas (Tabla/Figura 1), de modo que los resultados finales que se obtuvieron en la fase en cuestión será usada de base en la siguiente.

Las fases que componen el proyecto son las siguientes:

- Datos del consorcio HapMap
- Creación y manejo de BBDD MySQL
- Aprendizaje automático (ML)
- PHP programación web



Tabla/Figura 1 Fases del TFM y la relación entre las mismas

Al final de cada apartado se comentarán las ventajas/desventajas del proceso realizado, así como los resultados parciales obtenidos en los mismos.

De este modo, al final del apartado de "PHP programación web" todos los resultados de cada proceso estarán incluidos dentro del portal web diseñado para este proyecto.

Para la realización de este proyecto se trabajó en un entorno Linux (Ubuntu), la elección de este sistema operativo (SO) se justifica en las características descritas en el anexo 1 [A1]. En dicho anexo también se incluye los pasos a seguir para la instalación de los diferentes componentes que serán necesarios para la elaboración de este proyecto.

Datos del consorcio HAPMAP

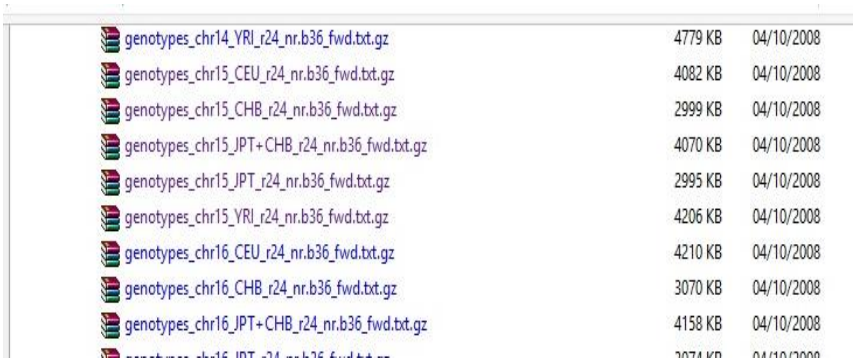
A) Descarga de los datos del proyecto

Los datos usados para la elaboración de este proyecto se descargaron desde los repositorios oficiales, del conjunto total de los SNP´s del genoma completo correspondientes a los 11 grupos étnicos estudiados [5] (Tabla/Figura 2).

Grupo de estudio	Codigo	Nº Participantes
African Ancestry in SW USA	ASW	87
Chinese in Metropolitan Denver, CO, USA	CHD	109
Gujarati Indians in Houston, Texas, USA	GIH	101
Han Chinese in Beijing, China	CHB	139
Japanese in Tokyo, Japan	JPT	116
Luhya in Webuye, Kenya	LWK	110
Maasai in Kinyawa, Kenya	MKK	184
Mexican Ancestry in Los Angeles, CA, USA	MEX	90
Toscani in Italia	TSI	102
Yoruba in Ibadan, Nigeria	YRI	209
UTAH RESIDENTS WITH ANCESTRY FROM NORTHERN AND WESTERN EUROPE	CEU	174
	Total	1421

Tabla/Figura 2 Relación del número participantes del proyecto HapMap

También, se descargaron los datos referentes a la selección de un cromosoma concreto y de ciertas poblaciones para la elaboración del algoritmo de ML [6] (Tabla/Figura 3).



Tabla/Figura 3 Repositorio HapMap

La elección de este conjunto de datos viene determinada por la elección del propio trabajo, aunque la aplicación de todo el proyecto puede ser extrapolable a otros conjuntos de datos como se comentara más adelante con el data set de Iris.

El conjunto de datos de HapMap está estructurado por cromosomas y por población estudiada de modo que incluyendo a los cromosomas sexuales y mitocondrial dispondríamos de una biblioteca génica de SNP´s de 275 archivos en formato RAR. Una vez descomprimidos el peso de los archivos asciende a más de 16 GB.

La cantidad de información contenida en esos 275 archivos tiene tal magnitud que para la elaboración del algoritmo de ML es necesario aligerar la carga de información que el programa lanzadera debe procesar antes de dar la respuesta al usuario. Por este motivo, se centró el

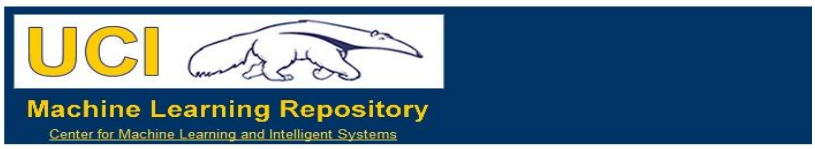
proceso de ML en el cromosoma 15 con una representación de 360 individuos participantes de las siguientes poblaciones:

- CEU: Residentes de Utah (EEUU) con ancestros europeos del norte y oeste de la colección CEPH
- CHB: Etnia Han en Beijing (China)
- JPT: Japoneses de Tokio (Japón)
- YRI: Yoruba en Ibadán, Nigeria
- ASIA: Etnia Han en Beijing (China) + Japoneses de Tokio (Japón)

Esta elección del cromosoma ha sido totalmente arbitraria. Sin embargo, la del repositorio donde estaban almacenados los datos [6] no lo fue tanto. Como se comentará en el apartado de ML, la inclusión del grupo ASIA afecta a la eficacia del análisis y tanto en el apartado de ML como en el capítulo de conclusiones se hará referencia a este efecto.

Por último, cabe recalcar que los 360 individuos estudiados son suficientes para aplicar la normalización de los datos según dicta el teorema central del límite (TCL) [3. Capítulo 3 pg. 80-81] durante el proceso de ML. El TCL podrá aplicarse, ya que los diferentes individuos de las diferentes poblaciones tenderán a asemejarse al resto de miembros de su población, de modo que podrá ser utilizado para inferir datos generales de dicha población.

Para Finalizar la recopilación de los datos necesarios, se descargaron los datos referentes al estudio de Ronald Fisher [4] (Tabla/Figura 4). La utilización de este conjunto de datos ha sido seleccionada por principalmente por tener una amplia aceptación de su uso como ejemplo para la elaboración de técnicas de clasificación por ML [7] aplicable a varios lenguajes de programación como pueden ser Python [8] o R [9].



Iris Data Set

Download: Data Folder, Data Set Description

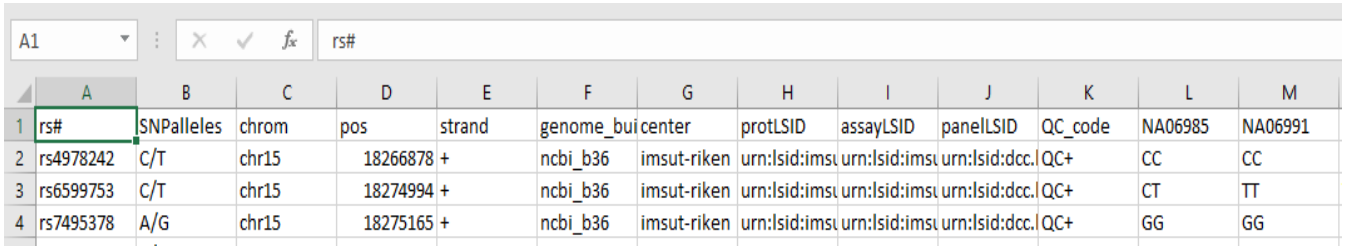
Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1152482

Tabla/Figura 4 Repositorio oficial del Iris Data set

B) Preprocesado de los datos

Los archivos del consorcio HapMap se encuentran estructurados como se muestran en la siguiente figura (Tabla/Figura 5).



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	rs#	SNPalleles	chrom	pos	strand	genome_bui	center	protLSID	assayLSID	panelLSID	QC_code	NA06985	NA06991
2	rs4978242	C/T	chr15	18266878	+	ncbi_b36	imsut-riken	urn:lsid:imsu	urn:lsid:imsu	urn:lsid:dcc.l	QC+	CC	CC
3	rs6599753	C/T	chr15	18274994	+	ncbi_b36	imsut-riken	urn:lsid:imsu	urn:lsid:imsu	urn:lsid:dcc.l	QC+	CT	TT
4	rs7495378	A/G	chr15	18275165	+	ncbi_b36	imsut-riken	urn:lsid:imsu	urn:lsid:imsu	urn:lsid:dcc.l	QC+	GG	GG

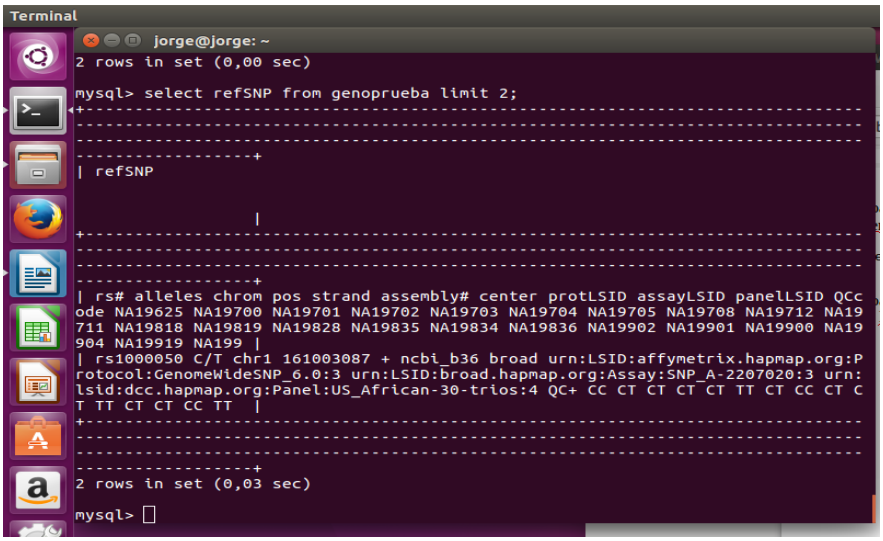
Tabla/Figura 5 Formato de los archivos HapMap

La relación de campos es la siguiente:

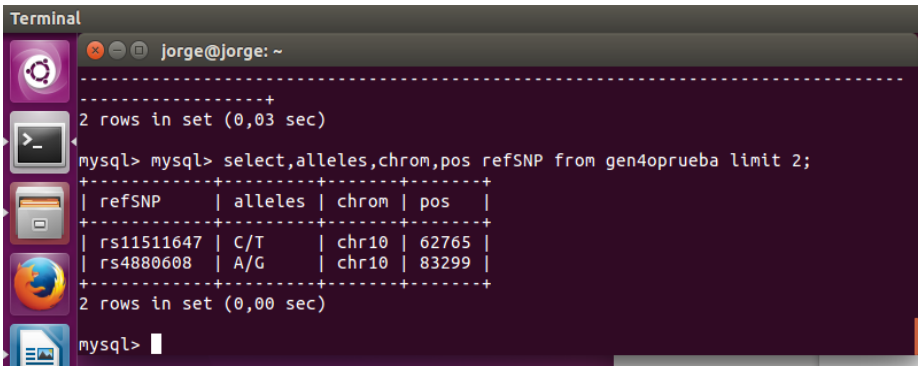
- Col1 (rs#): identificador numérico del SNP según su fecha de descubrimiento
- Col2 (SNPalleles): SNP
- Col3 (chrom): Cromosoma donde se encuentra el SNP. Irán numerados de 1 al 22, incluyendo el X, Y e M (mitocondrial).
- Col4 (pos): Posición del SNP en el genoma.
- Col5 (strand): Cadena donde se mapeo el SNP.
- Col6 (genome_build): Versión del ensamblador de las secuencias (versión 36), actualmente versión 107 [10].
- Col7 (center): Centro de investigación que realizo el genotipado.
    - Sanger [11]
    - Broad [12]
    - Perlegen [13]
    - Affymetrix [14]
    - Bcm: Baylor College of Medicine [15]
    - Illumina [16]
    - Imsut: Instituto de ciencias médicas de Tokio [17]
    - Ucsf: universidad de San Francisco, California (EEUU) [18]
    - Mcgill: Universidad McGill (Canadá) [19]
    - Chmc: Centro medico infantil de Cincinnati (EEUU) [20]
- Col8 (protLSID): Protocolo usado para el genotipado.
- Col9 (assayLSID): Ensayo del genotipado.
- Col10 (panelLSID): Grupo étnico al que pertenece la muestra referenciada (Tabla/Figura 1).
- Col11 (QC_code): Código QC
- Col12 y siguientes (NAXXXX): Genotipos observados en los diferentes individuos. El código hace referencia al número de catálogo del instituto Coriell. [21]

Los datos obtenidos de la base de HapMap tal como estaba en su formato nativo no se encontraban en las condiciones óptimas para ser tratados en la BBDD elegida para el proyecto, en el apartado “creación y manejo de BBDD MySQL” se describirá la manera en que se crearon las diferentes BBDD.

Sin embargo, al introducir los datos de la tabla creada específicamente para contener toda la información de los SNP´s según son descomprimidos los datos podemos observar que existe un error de estructuración que provoca que todos los datos sean introducidos en el primer campo de la tabla MySQL (Tabla/Figura 6), cuando lo deseable para la realización de consultas a la BBDD sería un formato tabulado (Tabla/Figura 7).



Tabla/Figura 6 lectura de datos errónea en MySQL debido al formato espacio delimitado



Tabla/Figura 7 Formato tabulado estándar aceptado por MySQL

El error producido en la inserción de los datos de la Tabla/Figura 6 se debe a que los archivos txt del consorcio HapMap se encuentran delimitados por espacio y la manera de introducirlos requiere que estén delimitados por tabulación. Por este motivo los datos deben ser pre procesados hasta que estén en un formato óptimo para la BBDD.

Para que el procesado de los archivos de HapMap coincidan tanto los campos del archivo con los de la tabla se siguieron diferentes enfoques como algunos de los siguientes ejemplos.

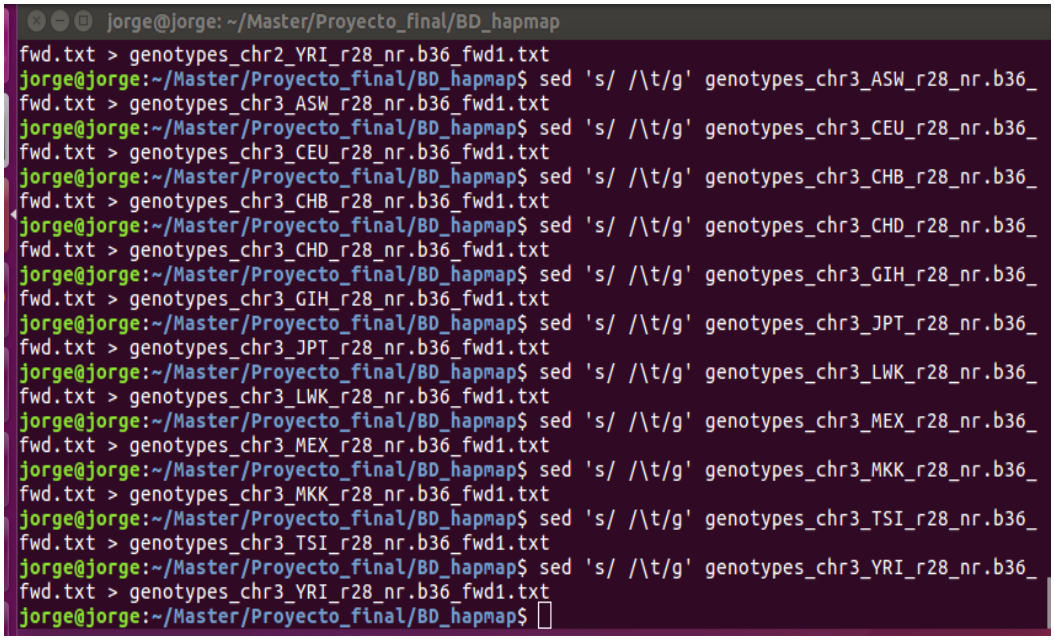
- Convertir los txt en csv o en un txt tabulado utilizando Excel.
- Usar el comando GAWK en el terminal para crear un archivo nuevo con formato.
- Usando cat “file” | tr ‘t’ ‘\t’
- sed 's/ /\t/g' “file”.

La opción de convertir los archivos por medio de Excel en un principio podría ser la opción más factible debido a que el conjunto de aplicaciones de Microsoft Office esta globalmente arraigado en todos los ámbitos de nuestra vida binaria.

La idea en principio parece buena, pero cargar los 275 archivos en formato Excel, para luego guardarlos en el formato ideal (csv o txt (delimitado por tabulaciones)) supone un consumo de energía y recursos. Pero los datos referentes al cromosoma 15 si serán tratados por Excel antes de obtener el formato de archivo ideal para importa a la BBDD. Esto será esencial para el ML.

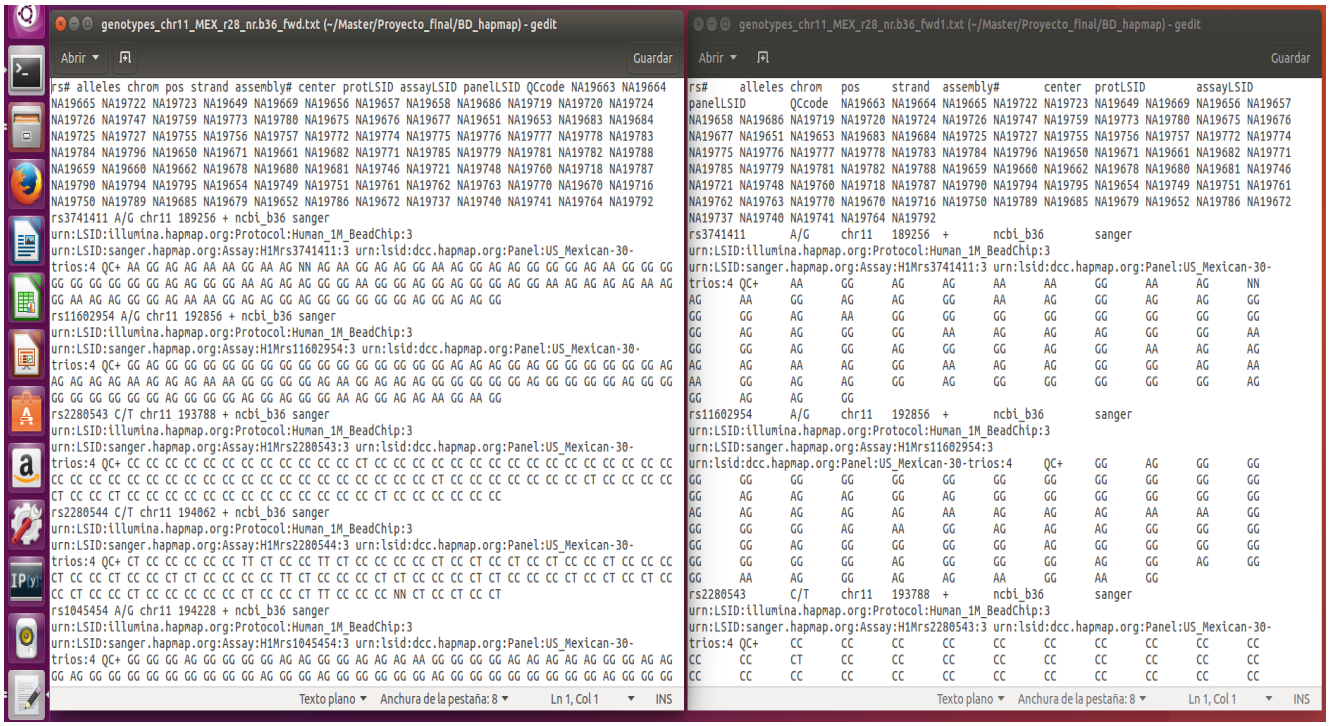
Al final se convirtieron los archivos del genoma completo de HapMap usando el siguiente comando en el terminal (Tabla/Figura 8).

```
sed 's/ /\t/g' genotypes_chr10_ASW_r28_nr.b36_fwd.txt >
genotypes_chr10_ASW_r28_nr.b36_fwd1.txt
```



Tabla/Figura 8 Cambio de formato de los archivos HapMap de espacio delimitado a tabulado

Con este comando procedemos a crear un archivo nuevo con la tabulación ya especificada, ya que sustituye los espacios que encuentra en cada entrada por una tabulación permitiendo así que sea más fácilmente interpretado por la BBDD a la hora de importarlos. En la siguiente imagen (Tabla/Figura 9) se puede apreciar el cambio de formato de los archivos de HapMap. Con estos archivos se procederá a crear las BBDD en MySQL con el formato estándar de la BBDD (Tabla/Figura 7).



Tabla/Figura 9 Cambio de tabulación archivos HapMap para su entrada en MySQL

Sin embargo, en el siguiente apartado (Creación y manejo de BBDD MySQL) se describirán los cambios posteriores que se realizaron sobre las BBDD para una mejor interpretación de las tablas.

Los archivos referentes al cromosoma 15 fueron pre procesados íntegramente con Excel para obtener el formato ideal sobre el que se trabajara en la parte de ML. Teniendo como partida un archivo cualquiera, se eliminaron aquellos campos irrelevantes para el análisis por ML como pueden ser los siguientes comentados para obtener la siguiente imagen (Tabla/Figura 10).

- Posición -> La posición en el cromosoma viene asociada al número de referencia.
- Chrom -> Todos los archivos corresponden al cromosoma 15, en el caso de querer clasificar por este campo, el algoritmo no encontraría diferencias entre los grupos.
- Campos relacionados con los centros/protocolos -> No son relevantes debido a que cada centro realizo parte del proyecto centrándose en un grupo de estudio. Su inclusión habría generado más ruido en la clasificación entorpeciendo la interpretación de estos.

E9 | NN

	A	B	C	D	E	F	G
1	rs#	SNPalleles	pos	panelLSID	NA06985	NA06991	NA06993
2	rs1000040	A/G	23302034	CEU	AA	AA	AA
3	rs100018	A/G	58867580	CEU	AA	AA	AA
4	rs1000221	A/G	87428400	CEU	GG	GG	GG
5	rs1000281	C/T	70340476	CEU	TT	CT	CC
6	rs1000290	C/T	98767132	CEU	TT	TT	CT
7	rs1000301	A/G	86830288	CEU	GG	GG	GG
8	rs1000306	A/G	58840013	CEU	AG	GG	AG
9	rs1000347	C/T	24480828	CEU	NN	CT	CT
10	rs1000471	C/T	87787587	CEU	CT	CT	CC

Tabla/Figura 10 Punto de partida archivos a utilizar en ML

Una vez realizado esta selección de variables, se procedió a realizar los pasos correspondientes para la obtención de formato final de los archivos (Tabla/Figura 12). Los pasos realizados fueron los siguientes.

Se seleccionaron solamente los 3000 primeros SNP´s comunes a todos los grupos étnicos estudiados, para ello se ordenaros los datos por el campo "rs#" y se seleccionaron aquellas filas en las que para los 5 archivos implicados tuvieran el mismo valor de campo "SNPalleles". Esta selección de datos es necesaria para reducir el volumen de datos que el sistema debe procesar antes de dar una respuesta. También para evitar que posteriormente se formen clústeres no deseados.

En ciertas ocasiones suelen haber una indisponibilidad de datos para un individuo, por lo que se representaría con el valor "NN" como se aprecia en la Tabla/Figura 10 (celda E9). Para evitar esta incongruencia se recurrirá a la moda de la población para rellenar ese dato. Esta acción fue realizada para homogeneizar el conjunto de datos, de modo que un miembro de una población dada tendrá más probabilidades de presentar el mismo SNP que otro miembro de la misma población para la posición referida del cromosoma. La moda se obtuvo con la siguiente formula

"=INDICE(E2:CP2;MODA(COINCIDIR(E2:CP2;E2:CP2;0)))" siendo, E2:CP2 (p.ej.) el intervalo de la primera referencia de SNP a obtener la moda. Este valor será sustituido por aquellos valores "NN" de la matriz de datos. A continuación, se transpondrán los datos (Tabla/Figura 11) para que los individuos con el código del instituto Coriell [21] queden como filas del archivo pues son estos, el conjunto de datos sobre los que se realizara el aprendizaje automático; pues el clúster se realizara por pertenencia a una población dada.

O14

	A	B	C	D	E	F
1	Individuo	rs1000040	rs100018	rs1000221	….	POBLACION
2	NA06985	AA	AA	GG	…..	CEU
3	NA06991	AA	AA	GG	…..	CEU
4	NA06993	AA	AA	GG	…..	CEU
5	NA06994	AG	AA	GG	…..	CEU
6	NA07000	AG	AA	GG	…..	CEU
7	NA07019	AA	AA	GG	…..	CEU
8	NA07022	AA	AA	GG	…..	CEU
9	NA07029	GG	AA	GG	…..	CEU

Tabla/Figura 11 Transposición de los datos para el proceso de ML

Por último, en el procesado llevaremos a cabo una transformación de los SNP´s en un código numérico, ya que, los intérpretes de ML necesitan que para el aprendizaje los datos columnas de una matriz sean de tipo numérico y no factorial como se aprecia en la Tabla/Figura 11. Para ello utilizaremos el siguiente código que será comparado con el del SNPalleles correspondiente a dicha entrada de rs#.

- 0 -> 0 alelos mutados
- 1 -> 1 alelo mutado
- 2 -> ambos mutados

Tomemos por ejemplo el proceso para la entrada rs1000040, suyo SNPalleles era "A/G" (Tabla/Figura 10). De modo que, si el SNP de NA06985 para ese rs# fuera "AG" o "GA" le

correspondería el valor 1, si fuera “AA” 0 y 2 para “GG”. Este proceso fue realizado con la siguiente formula “=SI(transpo!B2=transpo!B$92;1;SI(transpo!B2=transpo!B$93;0;2))” de modo que el valor de la celda (transpo!B2) se compara con el SNP de 1 alelo mutado (B$92) y el de 0 alelos mutados (B$93). Así obtendremos la tabla construida para aplicar ML, la cual será guardada como un archivo txt delimitado por tabulaciones.

I11

	A	B	C	D	E	F
1	Individuo	rs1000040	rs100018	rs1000221	....	POBLACION
2	NA06985	0	0	2	.....	CEU
3	NA06991	0	0	2	.....	CEU
4	NA06993	0	0	2	.....	CEU
5	NA06994	1	0	2	.....	CEU
6	NA07000	1	0	2	.....	CEU
7	NA07019	0	0	2	.....	CEU
8	NA07022	0	0	2	.....	CEU
9	NA07029	2	0	2	.....	CEU

Tabla/Figura 12 Formato Chrom15 final

Durante la elaboración del código de algoritmo de ML y su depuración, se comprobó que este archivo necesitaba ser dividido en dos por razones que serán descritos en el apartado de ML cuando se trabaje sobre estos archivos.

Pero obteniendo una perspectiva de la información contenida en la Tabla/Figura 12, el código usado por el instituto Coriell para identificar a los participantes [21] es insuficiente. Por este motivo se desarrollará un nuevo repositorio donde se incluya la información más relevante de los participantes como puede ser el sexo, el grupo familiar o población al que pertenece, como puede ser la presencia o no de algún gen mutado (Tabla/Figura 13).

familia: Bloc de notas

Archivo Edición Formato Ver Ayuda

Num ID	Catalog ID	Perfil genetico	Sex	Family	Relationship	Gene	Mutation
1	NA17962 CHD	Female	proband				
2	NA17965 CHD	Male	proband				
3	NA17966 CHD	Female	proband				
4	NA17967 CHD	Male	proband				
5	NA17968 CHD	Female	proband				
6	NA17969 CHD	Male	proband				
7	NA17970 CHD	Female	proband				
8	NA17972 CHD	Male	proband				
9	NA17974 CHD	Male	proband				
10	NA17975 CHD	Male	proband				
11	NA17976 CHD	Male	proband				
12	NA17977 CHD	Female	proband				
13	NA17978 CHD	Female	proband				
14	NA17979 CHD	Male	proband				

Tabla/Figura 13 Repositorio creado para recopilar la información de las familias que participaron en HapMap

La obtención de estos datos se realizó para su comprobación en caso de aparición de outliers durante el proceso de ML, además de poder obtener más información acerca de los propios participantes ya que en el archivo del Chrom15 solo se indica código Coriell.

Para finalizar los datos del repositorio de Iris flower [4] no requirieron de preprocesado de ningún tipo, ya que están pensados para su inmediata aplicación sobre algoritmos de ML. La

única salvedad es que requerirá de un parámetro extra durante su carga en la BBDD, la cual será descrita en el siguiente apartado para su correcta.

C) Resultados

Los resultados incluidos en este apartado consistirán en:

- 275 archivos del repositorio HapMap en formato txt delimitados por tabulaciones: 1 archivo por cromosoma-población.
- 2 archivo en formato csv con la información del Chrom15 final teniendo como principal diferencia la incorporación del grupo ASIA (JPT+CHB).
- 1 archivo del repositorio de Iris flower Data set en formato txt delimitados por tabulaciones.
- 1 archivo del repositorio del instituto Coriell en formato txt delimitados por tabulaciones.

Por motivos de espacio, los archivos del repositorio HapMap que se transformaron para estar delimitados por tabulaciones no serán incluidos en el GIT del proyecto [22] por medio del tutorial del anexo 3 [A3], por lo que solo se subirán al mismo el archivo correspondiente al "newcrom15.csv"," fullcrom15.csv", "crom15.csv", "Iris.txt" y "familia.txt".

D) Ventajas/Desventajas

- Ventajas
    - Evitará procesados a posteriori (salvo excepción).
    - Al trabajar en otro SO se guarda así una copia de seguridad de los archivos generados en el procesado.
- Inconvenientes
    - Encontrar la mejor opción para convertir los archivos al formato adecuado supone un consumo de tiempo y recursos si no se tiene conocimientos del lenguaje de Ubuntu.
    - Convertir todos los archivos al formato correspondiente.
    - Necesario de un segundo SO (Windows) para procesar los archivos relacionados con el cromosoma 15 implicados en el ML.

Creación y manejo de BBDD MySQL

A) Elección de la BBDD

Al Inicio de la elaboración del proyecto fue necesario la selección del mejor motor de BBDD que se ajustase tanto a las necesidades del proyecto como a las del programador. La elección del tándem de BBDD-web supone el elemento crítico del proyecto considerando que es la BBDD la base sobre la que el proyecto en si trabaja.

En un principio se planteó la posibilidad de trabajar sobre un lenguaje de programación Python por su versatilidad para la implementación de ML por medio del paquete/librería scikit-learn [25]. Esta elección inicial supondría que los datos deberían ser alojados en una BBDD compatible con Python, pudiendo elegir entre BBDD SQL (MySQL) o NoSQL (MongoDB o CassandraBD) [26].

La elección de una BBDD SQL frente a las NoSQL para la elaboración del proyecto se basó principalmente en el conocimiento de la organización y tecnología de BBDD SQL obtenidas durante las asignaturas que componen el programa de estudios de este master. De este modo, los tiempos de desarrollo han sido menores permitiendo una mejor adaptación al calendario del proyecto. Por la capacidad de obtener resultados en días frente a semanas de haberse elegido un BBDD NoSQL.

SQL es un lenguaje de búsquedas (query) ampliamente reconocido y usado en toda la comunidad de desarrolladores informáticos y sencillo de aprender. Otro factor es que estas BBDD presentan una completa librería de herramientas para tratar las tablas, esto podrá verse más adelante cuando se realice el procesado de los datos en MySQL sobre el conjunto completo de datos de HapMap.

Sin embargo, se sopeso usar otra BBDD tipo NoSQL, pero los costes de aprendizaje de estos supondrían un retraso en el calendario ya que por ejemplo la principal diferencia entre MongoDB frente a las otras dos citadas es que la primera se basa en un modelo de documentos en comparación al basado en tablas de los otros dos.

La necesidad de procesar los datos, seleccionando aquellos campos que serían útiles para la elaboración del algoritmo, necesitaba que los datos se encontraran en formato de tabla porque como se vio en la Tabla/Figura 5, hay ciertos campos (p.ej. assayLSID) que no son útiles para la clasificación de los datos y que podría suponer un aumento del ruido en los mismo.

Por otra parte, aunque CassandraBD sigue un patrón de organización bastante similar a MySQL [27] en estructura y funcionalidad habría supuesto readaptarse para realizar lo mismo que con MySQL solo que en el lenguaje con el que CassandraBD trabaja. Del mismo modo CassandraBD no ofrecía la flexibilidad que las BBDD SQL presentan por lo que sería necesario adaptar las tablas a la opción de búsqueda que se desean realizar.

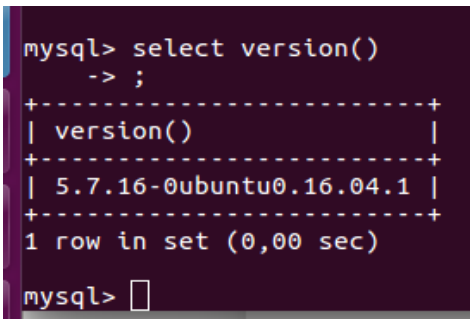
Otro motivo para descartar CassandraBD es que la principal plataforma de destino de CassandraBD es java, lenguaje de programación totalmente desconocido y que requeriría de tiempo para entender el manejo de una máquina virtual de Java (JVM). Para concluir las razones de la elección sobre CassandraBD es que esta no requiere de un patrón “leer-antes-escribir” (read-before-write pattern en inglés) el cual es necesario para evitar errores como el mostrado en la Tabla/Figura 6.

En el apartado anterior (Datos del consorcio HapMap) el preprocesado de los datos descrito sigue el modelo utilizado para la BBDD seleccionada, pero como ahí se indicó será necesario volver a procesarlos de una manera más global al tener en cuenta el número de archivos incorporados a una única tabla en vista de que será más óptimo cambiar procesarlo todo de una vez en lugar de hacerse en cada archivo individual del consorcio HapMap.

MySQL soporta un conjunto bastante amplio de lenguajes de programación con los que operar:

- C#
- C++
- Java
- Perl
- PHP
- Python

La elección final del lenguaje sobre el que correrá la BBDD (en nuestro caso PHP) se explicará detenidamente en los apartados correspondientes de esta memoria. A través del terminal de Linux podemos determinar la versión de la BBDD con la que trabajamos, ya que es necesario tener las versiones correctas y actualizadas del software con el que se va a trabajar para evitar incompatibilidades como por ejemplo la aparición de la llamada “white screen of death”.

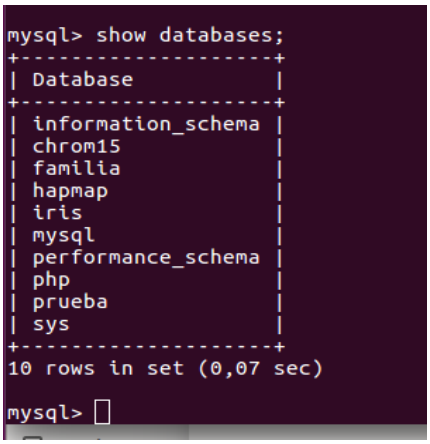


Tabla/Figura 14 Versión de MySQL sobre la que se trabaja en el entorno Linux Ubuntu

B) Creación de las bases de datos y las correspondientes tablas

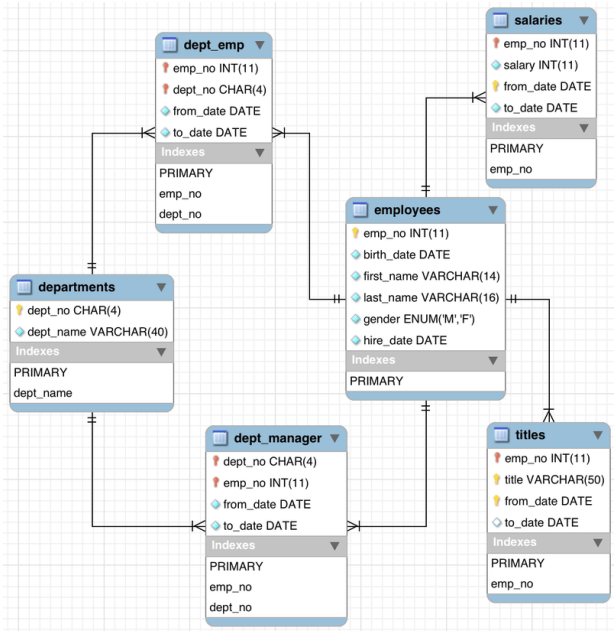
En el anexo 4 [A4] se describe la manera de crear las distintas BBDD y las tablas que van a ser usadas en TFM. Estas tablas/BBDD serán las siguientes (Tabla/Figura 15).

- HapMap
- Familia
- Chrom15
- Iris



Tabla/Figura 15 BBDD de MySQL implicadas en la realización del TFM

Aunque MySQL permite relacionar diferentes BBDD por medio del código "foreign () references (BBDD(tabla))" para referenciar por ejemplo que un campo de la tabla 1 BBDD 1 es el mismo que aparece en la tabla 1 BBDD 2 (Tabla/Figura 16). Esta opción no fue considerada debido a que las 4 tablas realizaran diferentes funciones dependiendo de la aplicación que vaya a realizar.



Tabla/Figura 16 Modelo entidad relación de una BBDD, imagen de ejemplo para ilustrar la correlación entre las diferentes BBDD (imagen no relacionada con el TFM)

En un principio la BBDD HapMap, familia y Chrom15 comparten por ejemplo el campo de perfil genético o que Chrom15 es una versión invertida y reducida de la información contenida en HapMap, pudiéndose relacionar ambas BBDD entre ellas. Respecto a la BBDD familia ocurriría una situación similar a la descrita en este párrafo.

Sin embargo, las BBDD se independizaron para promover una mayor estabilidad del sistema, y así evitar errores por malas relaciones programadas entre ellas. Al ser independientes el propio sistema realizara la conexión pertinente con la BBDD establecida en el momento en el que el usuario realice una pregunta requerida.

Otro motivo por el que las BBDD son independientes es porque la información contenida en ellas es diferente en el sentido estricto de la palabra. La BBDD HapMap se compone principalmente por campos categóricos, al igual que sucede con la BBDD familia. En cambio, las BBDD Chrom15 y Iris principalmente son numéricas teniendo pocos campos categóricos que serán utilizados para la predicción de categorías.

Todas las BBDD se crearon en relación a los datos que contenían para que el número de modificaciones fuera el mínimo, de modo que la opción de búsqueda por el comando SELECT no fuera el determinante como se ha sugerido con el uso de CassandraBD.

En el caso de la BBDD HapMap, se tuvo en cuenta que los participantes de cada población estudiada eran diferentes tanto en número (Tabla/Figura 2) como en el código Coriell identificativo [21], por tanto, se optó por renombrar los campos de la tabla para hacer referencia al número máximo de participantes de un grupo.

Este cambio de nomenclatura ocasiona que en ciertos valores de campos de referencia de individuos por el código Coriell presente valores omitidos ("NA") ocasionando que esta BBDD debido a su volumen de información sea considerada no apta para ML y la modificación realizada con los datos del cromosoma 15

La construcción de la BBDD de Chrom15 se realizó teniendo como partida los archivos referentes al cromosoma 15 [22], de modo que el primer campo hará referencia al código Coriell y los siguientes a los 3000 SNP´s comunes para las poblaciones implicadas. Pero para este caso se requirieron de 2 tablas construidas siguiendo el mismo código [A4] pero que serán llenadas con los dos archivos csv del Chrom15 final generados especialmente para el ML.

La BBDD de familia de acuerdo con el código de [A4] presentara únicamente la función de consulta para obtener información extra acerca de los participantes, pero la información ofrecida por el instituto Coriell [21] no está completa por lo que en muchos casos la relación familiar no está indicada y por si fuera poco solo se tiene constancia de unos pocos individuos que presentan genes afectados.

La inclusión de esta BBDD, permitiría clarificar lagunas de información pero que en ningún caso será usada para la determinación de nuevas hipótesis. Por este motivo su adicción es considerada más como un extra que como parte del proyecto inicialmente propuesto.

La BBDD iris se creó atendiendo a la máxima similitud con la estructura de los datos originales, de modo que la estructura no fuera diferente para poder comparar los procesos durante la fase de ML ya sea cargando los datos desde la BBDD o como desde el propio repositorio del software de ML.

C) Introducir de los datos y procesado de los datos

Llenar de contenidos las tablas previamente creadas se realiza con el comando INSERT siguiendo la siguiente pauta [2].

```
INSERT INTO TABLA (campo1, campo2, …, campoN)
VALUES ('valor1', 'valor2', …, 'valorN')
```

Este comando de entrada de los datos es el estándar para el lenguaje SQL, pero esta modalidad solo sería apta cuando el investigador no tuviera un volumen de datos como el que estamos

manejando en el TFM. El tiempo que se necesitaría para entrar línea a línea cada dato supondría un consumo de tiempo no justificable para un trabajo de este tipo.

En su defecto y para evitar esta pesada tarea, se recurrirá al comando LOAD DATA que permite a las tablas auto rellenarse recurriendo a la información contenida en un archivo. Los formatos de archivos soportados por este comando son los siguientes:

- Csv
- Txt

Es importante recalcar que dependiendo del tipo de archivo del que partimos tendremos que modificar el comando de entrada de los archivos, ya que cada formato presenta ciertas características no relacionables entre ellas, que afectan a la manera en la que son importadas.

Durante el apartado anterior ("Datos del consorcio HapMap") se hizo hincapié en la necesidad de convertir los archivos txt de HapMap por un formato que fuera manejable por MySQL. En el caso de trabajar con archivos txt deben estar tabulados de modo que cada línea represente un registro (row) y cada campo el valor correspondiente al atributo/columna creado en la tabla.

Pero en el caso de los archivos csv los valores de los atributos no se encuentra separados por tabulaciones, sino que están separados por el carácter ";" obligando a indicárselo al comando de entrada para que pueda automáticamente relacionar los valores con su atributo.

Por tanto, podemos establecer que el comando LOAD DATA presenta una estructura bifásica, la primera de ellas indicara que archivo debe cargarse en que tabla y la segunda base se corresponden con las opciones extra de entrada como puede ser ignorar ciertas líneas, o determinar qué carácter es el delimitador de valores [30 cap. 4.3.3 pág. 457-458].

Para los archivos txt:

```
LOAD DATA LOCAL INFILE 'genotypes_chr1_ASW_r28_nr.b36_fwd.txt' into table HapMap IGNORE 1 LINES;
```

Para los archivos csv:

```
LOAD DATA LOCAL INFILE 'Chrom15' into table Chrom15 FIELDS TERMINATED BY ';' LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

La orden de comando IGNORE 1 LINES ocasiona que el inicio de carga de los datos comience en el registro correspondiente a la segunda línea. Este comando es útil cuando el archivo de origen tiene encabezados, ya que de no incluir el comando la primera entrada de la tabla se correspondería con los encabezados del archivo y por tanto parecería que están duplicados los nombres de los atributos de la tabla. Es especialmente importante si vamos a realizar tareas de ML ya que evitaría que los atributos tuvieran diferentes características ocasionando errores de lectura.

La orden FIELDS TERMINATED BY 'x' permite al intérprete de MySQL determinar cómo están estructurados los valores de los atributos, de modo que cada vez que el carácter aparezca en el documento que se va a importar este será tratado como una tabulación permitiendo la correcta ordenación de los valores en sus respectivos atributos. Este comando es de especial importancia cuando se importan archivos con formato csv dado que el propio formato te lo estructura así.

Para terminar con la nomenclatura del comando LOAD DATA, la orden LINES TERMINATED BY “” se utiliza para indicar cuando termina un registro y comienza el otro. En la mayoría de las ocasiones se refiere a un salto de línea, aunque si el fichero se encuentra bien construido con acabar la línea con intro genera ese salto de línea necesario. Por lo tanto, la mayoría de las veces este comando puede ser omitido en el terminal al importar un archivo.

Una vez entonces determinado el proceso de importación de archivos procedemos a realizar la entrada de los datos en sus correspondientes tablas. Para las BBDD familia, Iris y Chrom15 la importación se realizó en un solo paso al tratarse de únicamente 1 archivo, pero para el caso de repositorio completo de HapMap introducir los 275 en una carpeta supuso tener en cuenta los cambios de cromosoma y población que contenían los nombres de los archivos debido a que todos los datos se introducían en la misma tabla (Tabla/Figura 17).

```
mysql> load data local infile 'genotypes_chr13_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 56991 rows affected, 56991 warnings (2,60 sec)
Records: 56991  Deleted: 0  Skipped: 0  Warnings: 56991

mysql> load data local infile 'genotypes_chr14_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 49078 rows affected, 49078 warnings (2,25 sec)
Records: 49078  Deleted: 0  Skipped: 0  Warnings: 49078

mysql> load data local infile 'genotypes_chr15_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 45565 rows affected, 45565 warnings (2,03 sec)
Records: 45565  Deleted: 0  Skipped: 0  Warnings: 45565

mysql> load data local infile 'genotypes_chr16_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 48100 rows affected, 48100 warnings (2,17 sec)
Records: 48100  Deleted: 0  Skipped: 0  Warnings: 48100

mysql> load data local infile 'genotypes_chr17_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 40768 rows affected, 40768 warnings (1,82 sec)
Records: 40768  Deleted: 0  Skipped: 0  Warnings: 40768

mysql> load data local infile 'genotypes_chr18_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 44751 rows affected, 44751 warnings (2,02 sec)
Records: 44751  Deleted: 0  Skipped: 0  Warnings: 44751

mysql> load data local infile 'genotypes_chr19_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 27866 rows affected, 27866 warnings (1,26 sec)
Records: 27866  Deleted: 0  Skipped: 0  Warnings: 27866

mysql> load data local infile 'genotypes_chr20_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 38787 rows affected, 38787 warnings (1,68 sec)
Records: 38787  Deleted: 0  Skipped: 0  Warnings: 38787

mysql> load data local infile 'genotypes_chr21_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 21123 rows affected, 21123 warnings (0,94 sec)
Records: 21123  Deleted: 0  Skipped: 0  Warnings: 21123

mysql> load data local infile 'genotypes_chr22_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 21890 rows affected, 21890 warnings (0,97 sec)
Records: 21890  Deleted: 0  Skipped: 0  Warnings: 21890

mysql> load data local infile 'genotypes_chrM_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 5 rows affected, 5 warnings (0,01 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 5

mysql> load data local infile 'genotypes_chrX_ASW_r28_nr.b36_fwd1.txt' into table hapmap ignore 1 lines;
Query OK, 39065 rows affected, 39065 warnings (1,72 sec)
Records: 39065  Deleted: 0  Skipped: 0  Warnings: 39065
```

Tabla/Figura 17 Carga de los datos previamente tabulados en la BBDD HapMap

Terminado el proceso se realizó una prueba sencilla para comprobar que los datos habían sido correctamente introducidos, para ello y a través del terminal se comprobó el numero diferentes de cromosomas y centros que había detectado con el comando SELECT COUNT (DISTINCT “atributo”) FROM “Tabla” (Tabla/Figura 18).

```
mysql> select count(distinct chrom) from hapmap;
+----------------------+
| count(distinct chrom) |
+----------------------+
|                   25 |
+----------------------+
1 row in set (0,93 sec)

mysql> select distinct center from hapmap;
+-------------+
| center      |
+-------------+
| sanger      |
| broad       |
| perlegen    |
| affymetrix  |
| bcm         |
| illumina    |
| imsut-riken |
| ucsf-wu     |
| mcgill-gqic |
| chmc        |
+-------------+
10 rows in set (14,17 sec)
```

Tabla/Figura 18 Comprobación de la carga de datos en la BBDD

En ocasiones podrían darse fallos de escritura en los datos originales de partida como por ejemplo un el uso de mayúsculas por minúsculas al determinar el valor de un campo o nomenclatura diferente. Como se determina en la Tabla/Figura 2, el número de poblaciones estudiadas son 11, pero el número de poblaciones distintas que nos aparece al interrogar a la BBDD por este atributo son 17 (Tabla/Figura 19).

```
mysql> select distinct panelLSID from hapmap;
+-----------------------------------------------------+
| panelLSID                                           |
+-----------------------------------------------------+
| urn:lsid:dcc.hapmap.org:Panel:US_African-30-trios:4 |
| urn:lsid:dcc.hapmap.org:Panel:CEPH-30-trios:1       |
| urn:lsid:dcc.hapmap.org:Panel:CEPH-60-trios:4       |
| urn:LSID:dcc.hapmap.org:Panel:Han_Chinese:2         |
| urn:LSID:dcc.hapmap.org:Panel:Han_Chinese:1         |
| urn:lsid:dcc.hapmap.org:Panel:Han_Chinese:4         |
| urn:lsid:dcc.hapmap.org:Panel:US_Chinese:4          |
| urn:lsid:dcc.hapmap.org:Panel:US_Gujarati:4         |
| urn:LSID:dcc.hapmap.org:Panel:Japanese:2            |
| urn:LSID:dcc.hapmap.org:Panel:Japanese:1            |
| urn:lsid:dcc.hapmap.org:Panel:Japanese:4            |
| urn:lsid:dcc.hapmap.org:Panel:Luhya_Kenyan:4        |
| urn:lsid:dcc.hapmap.org:Panel:US_Mexican-30-trios:4 |
| urn:lsid:dcc.hapmap.org:Panel:Maasai_Kenyan-60-trios:4 |
| urn:lsid:dcc.hapmap.org:Panel:Italian:4             |
| urn:LSID:dcc.hapmap.org:Panel:Yoruba-30-trios:1     |
| urn:lsid:dcc.hapmap.org:Panel:Yoruba-60-trios:4     |
+-----------------------------------------------------+
17 rows in set (18,31 sec)
```

Tabla/Figura 19 Error de entrada de datos BBDD

Estos “errores” de escritura deberán ser corregidos para una correcta interpretación de la información, así como para un mejor desarrollo del portal web. En Excel este proceso es considerado sencillo gracias al comando REEMPLAZAR. Sin embargo, el número de registros que presenta la BBDD HapMap, bien refiriéndonos al número de archivos implicados o a la propia tabla de MySQL, no permite realizar el reemplazo por esta técnica.

Para ello se recurrió al comando homologo que presenta el motor de la BBDD [30].

```
update HapMap set panelLSID = replace (panelLSID,'urn:lsid:dcc.HapMap.org:Panel:Han_Chinese:2','CHB');
```

Con este comando realizamos búsquedas en el atributo especificado para buscar el valor del campo a cambiar y que los sustituya por el que nosotros deseamos. Este cambio se repitió para los atributos “panelLSID”, “center”, “protLSID” y “SNPalleles” de las bases de datos HapMap, para una mejor lectura de los campos cuando se realicen los procesos requeridos por el usuario (Tabla/Figura 20).

```
mysql> select distinct panelLSID from hapmap;

+-----------+
| panelLSID |
+-----------+
| ASW       |
| CEU       |
| CHB       |
| CHD       |
| GIH       |
| JPT       |
| LWK       |
| MEX       |
| MKK       |
| TSI       |
| YRI       |
+-----------+
11 rows in set (2 min 55,01 sec)
```

Tabla/Figura 20 PanelLSID tras procesado MySQL

Los cambios realizados en las BBDD permitirán una mayor rapidez en las búsquedas dentro de MySQL y del portal web además de permitir una mejor programación de los formularios del portal al ser utilizado un código más sencillo que el incluido en los archivos originales de HapMap.

Al atributo “assayLSID” no se le pudo realizar este proceso debido a que cada valor por registro del atributo “assayLSID” de toda la BBDD es único y fue imposible encontrar unos valores que permitieran actualizar ese campo a gusto del programador (Tabla/Figura 21).

```
mysql> select assayLSID from hapmap limit 5;
+-------------------------------------------------+
| assayLSID                                       |
+-------------------------------------------------+
| urn:LSID:sanger.hapmap.org:Assay:H1Mrs4124251:3  |
| urn:LSID:broad.hapmap.org:Assay:SNP_A-8575115:3  |
| urn:LSID:sanger.hapmap.org:Assay:H1Mrs28446478:3 |
| urn:LSID:broad.hapmap.org:Assay:SNP_A-8709646:3  |
| urn:LSID:sanger.hapmap.org:Assay:H1Mrs11240767:3 |
+-------------------------------------------------+
5 rows in set (0,00 sec)
```

Tabla/Figura 21 Ejemplo de imposibilidad de consenso assayLSID

Por otra parte, las BBDD de familia (Tabla/Figura 12) y Chrom15 (Tabla/Figura 13) tuvieron un pre procesado anterior a su carga en sus BBDD por lo que no fue necesario modificarlos a través de la consola de MySQL y dicho procesado fue realizado con la opción de Excel al tratarse de un

volumen menor de datos a cargar. La mejora de los campos repercutirá a la hora de realizar las consultas a través del portal web.

En el portal web se incluirá una leyenda de las abreviaciones para entender mejor las opciones por las que el usuario podrá realizar las consultas.

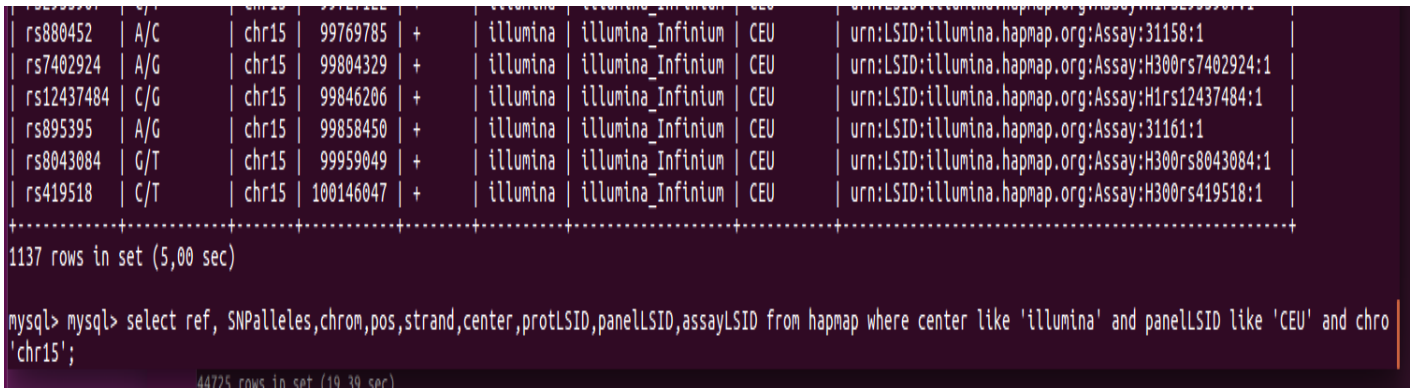
D) Pruebas de estabilidad de la BBDD. Consultas BBDD.

Tras cargar los datos en las diferentes BBDD y procesarlas se procedió a comprobar que los datos se habían introducido correctamente. Para ello se realizaron consultas interrogando a las BBDD. Al igual que ocurrió con el comando LOAD DATA el comando de SELECT se compone de dos partes, la primera indica los atributos a visualizar y la segunda indica las condiciones a imponer a la búsqueda.

```
SELECT campo1,campo2, …, campoN FROM “nombre_tabla” ORDER BY campo1 WHERE campo1 like “condición_campo1” and campo2 like “condición_campo2” LIMIT X;
```

Este comando es solo un ejemplo de las múltiples opciones con las que se puede trabajar con MySQL. La parte del código entre SELECT … FROM indica al motor de búsquedas que debe crear una tabla mostrando únicamente esos campos seleccionados (Tabla/Figura 22), si en lugar de los campos se indicase “*” mostraría todos los campos que tuviera dicha tabla.

El comando ORDER BY ordena la tabla alfabéticamente según el campo indicado. LIMIT muestra únicamente el número de entradas indicadas como argumento y la cláusula WHERE es el análogo al condicional de Excel, mostrando únicamente los registros que cumplan las condiciones indicadas. En la siguiente imagen se puede observar el resultado del procesado de las tablas por medio del comando UPDATE anteriormente descrito.



Tabla/Figura 22 Query condicionada de MySQL aplicada sobre la BBDD HapMap

Otra de las funciones que nos ofrece MySQL es la posibilidad de agrupar nuestros datos para formar clústeres con el comando GROUP BY, el cual permite obtener al usuario información acerca del número de entradas que coinciden con el parámetro en cuestión (Tabla/Figura 23).

Esta opción en relación al TFM no es aplicable utilizando a término dicho ya que en el apartado de ML la agrupación se hará en relación al parámetro población (panelLSID) de modo que este resultado podría fácilmente obtenible con el comando SUMMARY () del motor de ML. En este punto del proyecto nos permite comprobar que los datos fueron correctamente introducidos.

En las Tablas/Figura 24 y 25 se verificaron que la BBDD de familia se encontraba operativa. La búsqueda realizada en la Tabla/Figura 24 permite comprobar que efectivamente tenemos 11 valores asociados al cada número de identificación (Num_ID). En esta BBDD el número de identificación solo hace referencia a su posición dentro de cada población, de modo que como ocurre en la Tabla/Figura 25 solo presentamos 1 registro que coincida con las condiciones impuestas en el query.

Para finalizar la comprobación de la entrada de datos que se utilizaran para poner a punto el algoritmo de ML se procedió a mostrar los archivos que componen el cabecero del repositorio de la BBDD Iris (Tabla/Figura 26). La utilización del paquete de Iris flower data set para su aplicación en ML bien podría realizarse cargando estos mismos archivos desde el propio repositorio del software lanzadera que correrá el algoritmo o desde la BBDD creada explícitamente con sus datos.

```
mysql> select panelLSID,count(*) as Perfiles from hapmap group by panelLSID;
+------------+-----------+
| panelLSID  | Perfiles  |
+------------+-----------+
| ASW        |   1543731 |
| CEU        |   4031093 |
| CHB        |   4056784 |
| CHD        |   1312343 |
| GIH        |   1409510 |
| JPT        |   4055077 |
| LWK        |   1527403 |
| MEX        |   1453659 |
| MKK        |   1532587 |
| TSI        |   1420526 |
| YRI        |   3985822 |
+------------+-----------+
11 rows in set (2 min 51,61 sec)

mysql> select chrom,count(*) as entradas from hapmap group by chrom;
+--------+-----------+
| chrom  | entradas  |
+--------+-----------+
| chr1   |   2085268 |
| chr10  |   1379462 |
| chr11  |   1331711 |
| chr12  |   1272378 |
| chr13  |   1010192 |
| chr14  |    823577 |
| chr15  |    733637 |
| chr16  |    760319 |
| chr17  |    636041 |
| chr18  |    775814 |
| chr19  |    422715 |
| chr2   |   2146537 |
| chr20  |    738582 |
| chr21  |    347748 |
| chr22  |    371982 |
| chr3   |   1724266 |
| chr4   |   1612217 |
| chr5   |   1633103 |
| chr6   |   1754504 |
| chr7   |   1411241 |
| chr8   |   1412019 |
| chr9   |   1197365 |
| chrM   |       860 |
| chrX   |    739112 |
| chrY   |      7885 |
+--------+-----------+
25 rows in set (9,03 sec)
```

Tabla/Figura 23 Relación de número de registros según el filtro por un campo específico

```
mysql> select Num_ID,Catalog_ID,sexo,Perfil_genetico from familia where Num_ID like 'ID001';
+--------+------------+--------+----------------+
| Num_ID | Catalog_ID | sexo   | Perfil_genetico |
+--------+------------+--------+----------------+
| ID001  | NA17962    | Female | CHD            |
| ID001  | NA19625    | Female | ASW            |
| ID001  | NA20845    | Male   | GIH            |
| ID001  | NA18524    | Male   | CHB            |
| ID001  | NA06984    | Male   | CEU            |
| ID001  | NA19794    | Female | MEX            |
| ID001  | NA21295    | Male   | MKK            |
| ID001  | NA18484    | Female | YRI            |
| ID001  | NA20502    | Female | TSI            |
| ID001  | NA19017    | Female | LWK            |
| ID001  | NA18939    | Female | JPT            |
+--------+------------+--------+----------------+
11 rows in set (0,00 sec)
```

Tabla/Figura 24 BBDD familia ID001

```
mysql> select num_id, catalog_id, perfil_genetico, sexo, familia, relacion from familia where num_id like 'ID005' and sexo like 'Male' and perfil_genetico like 'ASW
';
+--------+-----------+---------------+-----+--------+---------+
| num_id | catalog_id | perfil_genetico | sexo | familia | relacion |
+--------+-----------+---------------+-----+--------+---------+
| ID005  | NA19703    | ASW           | Male | 2368    | father   |
+--------+-----------+---------------+-----+--------+---------+
1 row in set (0,11 sec)
```

Tabla/Figura 25 Query 1 resultado BBDD familia

```
mysql> select * from iris limit 5;
+-------------+------------+-------------+------------+--------+
| SepalLength | SepalWidth | PetalLength | PetalWidth | Class  |
+-------------+------------+-------------+------------+--------+
|         5.1 |        3.5 |         1.4 |        0.2 | setosa |
|         4.9 |        3.0 |         1.4 |        0.2 | setosa |
|         4.7 |        3.2 |         1.3 |        0.2 | setosa |
|         4.6 |        3.1 |         1.5 |        0.2 | setosa |
|         5.0 |        3.6 |         1.4 |        0.2 | setosa |
+-------------+------------+-------------+------------+--------+
5 rows in set (0,00 sec)
```

Tabla/Figura 26 Head BBDD Iris

En este punto las BBDD que serán usadas en el TFM se encuentran totalmente operativas y listas para su uso tanto dentro del software de ML como en el del portal web.

E) Resultados

La relación de resultados en este punto fue la siguiente:

- BBDD HapMap: Recopilación del repositorio HapMap correctamente estructurada para su uso en el portal web.
- BBDD familia: Información referente a los 1421 participantes del proyecto HapMap.
- BBDD Iris: Iris flower Data set en formato tabla SQL.
- BBDD Chrom15_final: Datos del cromosoma 15 para ML, 1 tabla con la población ASIA incluida y otra tabla sin ella.

F) Ventajas/inconvenientes

- Ventajas:
    - Poder procesar los 275 archivos del consorcio HapMap.
    - Fácil carga de los archivos en el software de ML, evitando un tercer procesado en otro lenguaje.
    - Reducir los tiempos de carga de las búsquedas por SELECT o en la interface con PHP al acotar los nombres de los valores contenidos en ellas.
- Inconvenientes:
    - Doble procesado de los datos que requieren de tiempo de ejecución.

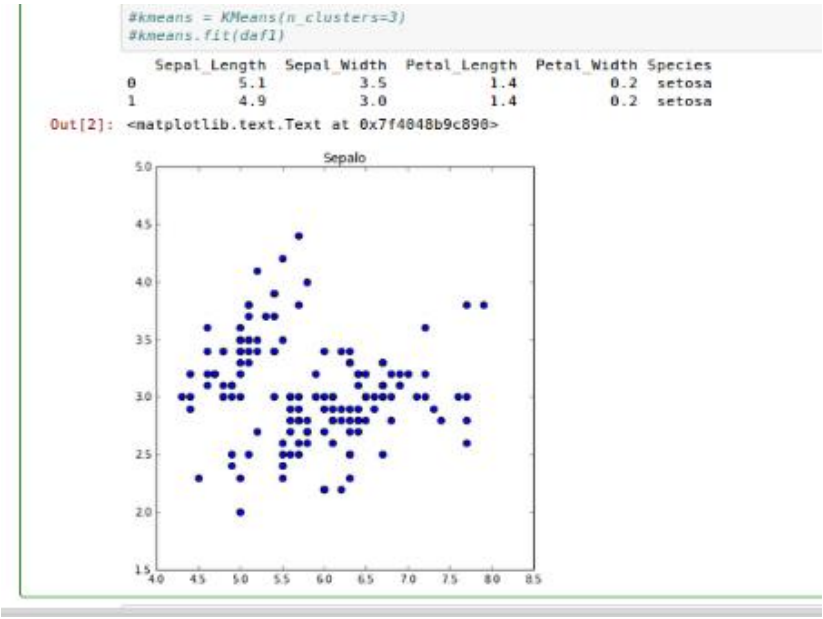
- Constatar que el proceso ha sido llevado a cabo correctamente por medio de la ejecución de búsquedas.

Aprendizaje automático (ML)

A) Elección del motor para el ML

En la actualidad existen varios softwares de acceso libre escritos en algunos de los lenguajes de programación más utilizados (C++ o Python p.ej.). Uno de los criterios principales para elegir el motor de ML es que pueda realizar una correcta conexión entre la BBDD y el portal web.

La elección de R [31] como motor de ML se debe principalmente a un mayor conocimiento de este lenguaje, mientras que con la librería scikit-learn [32] de Python se tienen unos conocimientos básicos. A pesar de esta diferencia de conocimiento de estas dos tecnologías se realizaron modelos de ML en ambos lenguajes (versión Python Tabla/Figura 27) para determinar cuál de ellos era el más óptimo y se ajustaba por tanto mejor a los objetivos del TFM. En el anexo 6 [A6] se mostrará el código en lenguaje Python para la realización del análisis del clúster.



Tabla/Figura 27 Modelo de clustering en Python usando Iris data set

Se desechó la idea de usar Python principalmente por que requería el desarrollo del portal web en un entorno Django, aplicación web de Python de la que no se tenía conocimiento de este sistema y cuyo motivo de rechazo será descrito y argumentado en el apartado de "programación web PHP".

Aun a pesar de seleccionar el motor R, se comprobó que en ambos motores se trabajaba bien con la conexión de MySQL a través del uso de las librerías pertinentes incluidas dentro de su paquete de datos por lo que, en este caso, el uso de MySQL no es un factor limitante (Tabla/Figura 28).

```
In [121]: # variante con mysqldb
          import MySQLdb as mdb
          import pandas as pd
          con=mdb.connect('localhost', 'root', 'pulido84', 'hapmap') # conectar con la DB
          sql="SELECT refSNP,alleles,chrom,strand from hapmap order by refSNP asc limit 5" # busqueda
          df_mysql = pd.read_sql(sql,con)
          print df_mysql
          print df_mysql.describe()

               refSNP alleles  chrom strand
          0       rs10     A/C   chr7      +
          1       rs10     A/C   chr7      +
          2       rs10     A/C   chr7      +
          3       rs10     A/C   chr7      +
          4  rs1000000     A/G  chr12      +
                 refSNP alleles chrom strand
          count       5       5     5      5
          unique      2       2     2      1
          top       rs10     A/C  chr7      +
          freq        4       4     4      5
```

```
> library(DBI)
> library(RMySQL)
> mydb1 = dbConnect(dbDriver("MySQL"), user="root", password="pulido84", dbname="chrom15", host="127.0.0.1")
> ch15=dbReadTable(mydb1,"chrom15")
Warning message:
In .local(conn, statement, ...) :
  Unsigned INTEGER in col 3 imported as numeric
> ch155=ch15[ch15$ref,ch15$SNPalleles,ch15$chrom,ch15$strand]
Error in `[.data.frame`(ch15, ch15$ref, ch15$SNPalleles, ch15$chrom, ch15$strand) :
  unused argument (ch15$strand)
> ch155=ch15[,1:5]
> head(ch155)
        ref SNPalleles chrom    pos strand
1  rs4124251        A/G  chr1  97215      +
2  rs9629043        C/T  chr1 554636      +
3 rs28446478        A/C  chr1 576058      +
4 rs12565286        C/G  chr1 711153      +
5 rs11240767        C/T  chr1 718814      +
6  rs3094315        A/G  chr1 742429      +
```

Tabla/Figura 28 Demostración del proceso de carga de las BBDD en Python (MySQLdb+pandas) [arriba] y en R (DBI+RMySQL) [debajo]

Otro argumente a favor de la elección de R es el entorno User-friendly que presenta Rstudio. El propio programa presenta una estructura orientada para un manejo ágil y dinámico, sin que se requiera grandes conocimientos de programación. Al igual que ocurre con la tecnología predictiva de palabras de que disponen actualmente los teléfonos móviles, Rstudio hace uso de un modelo análogo facilitando tras unas pocas letras autocompletar el comando requerido o a una serie de opciones que concuerden con lo escrito; además ofrece información acerca de la función y de cómo completarla para evitar errores de escritura.

Otra característica de Rstudio y que destaca sobre Python es la gestión del código de errores, ya que al cometer un error en Python te indica en que línea se encuentra el error, pero de manera algo confusa para los principiantes. En cambio, Rstudio te determina que fallos se ha encontrado como puede ser que el objeto en cuestión no exista, faltas de ortografía al escribir el código o indicar los motivos por los que no puede realizar un gráfico porque las dimensiones de los datos no concuerdan, por poner algunos ejemplos.

Esta elección de realizar ML a través de un programa lanzadera puede determinarse como un paso innecesario debido a que de haber sido Python el lenguaje principal del TFM, la carga de las librerías y su visualización web habría resultado más sencilla en un principio. Por poner un ejemplo, la programación del estadístico kappa de Cohen para determinar el nivel de acuerdo entre el modelo original y las predicciones basadas en un aprendizaje no supervisado requiere de conocimientos de Python avanzados y que al programador le requeriría de tiempos de desarrollo mayores para la obtención de un resultado óptimo.

R y su interface guiada de Rstudio, permiten evitar la realización de largas líneas de código para realizar procesos sencillos por medio de la carga de estas librerías. Las librerías de R vienen

estructuradas de modo que un simple comando ejecute varios pasos o comandos de forma que reduce la carga de programación para llegar a un público más general.

Pongamos por ejemplo el uso de la librería Caret [40] para la predicción de clases utilizándose la citada librería [45] (Tabla/Figura 29).

```
Console ~/
> require(class)
> require(lattice)
> require(e1071)
Loading required package: e1071
> require(ggplot2)
> library(caret)
> normalize=function(x) {
+     num=x - min(x)
+     denom=max(x) - min(x)
+     return (num/denom)
+ }
> iris_normalizado=as.data.frame(lapply(iris[1:4], normalize))
> set.seed(1234)
> irisx=sample(2, nrow(iris), replace=TRUE, prob=c(0.67, 0.33))
> iris.training1x=iris[irisx==1, 1:4]
> iris.test1x=iris[irisx==2, 1:4]
> iris.trainLabels1x=iris[irisx==1, 5]
> iris.testLabels1x=iris[irisx==2, 5]
> iris_pred1x=knn(train=iris.training1x, test=iris.test1x, cl=iris.trainLabels1x, k=3)
> confusionMatrix(iris.testLabels1x,iris_pred1x)
Confusion Matrix and Statistics
```

Tabla/Figura 29 Código R predicción Clases usando la librería Caret

El código anterior mostrara los resultados de realizar un test estadístico comparando los dos argumentos programados (iris.testLabels1x e iris_pred1x), enmascarando y automatizando los procesos descritos en [40 pgs.23-24] dando como resultado final una tabla comparativa y los resultados del test (Tabla/Figura 30).

```
Console ~/
> confusionMatrix(iris1x.testLabels,iris1x_pred)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa          9          0         0
  versicolor      0         15         0
  virginica       0          1        15

Overall Statistics

               Accuracy : 0.975
                 95% CI : (0.8684, 0.9994)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : 7.374e-15

                  Kappa : 0.9615
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.000            0.9375           1.0000
Specificity                 1.000            1.0000           0.9600
Pos Pred Value              1.000            1.0000           0.9375
Neg Pred Value              1.000            0.9600           1.0000
Prevalence                  0.225            0.4000           0.3750
Detection Rate              0.225            0.3750           0.3750
Detection Prevalence        0.225            0.3750           0.4000
Balanced Accuracy           1.000            0.9688           0.9800
>
```

Tabla/Figura 30 Estadístico Kappa de Cohen para la BBDD iris

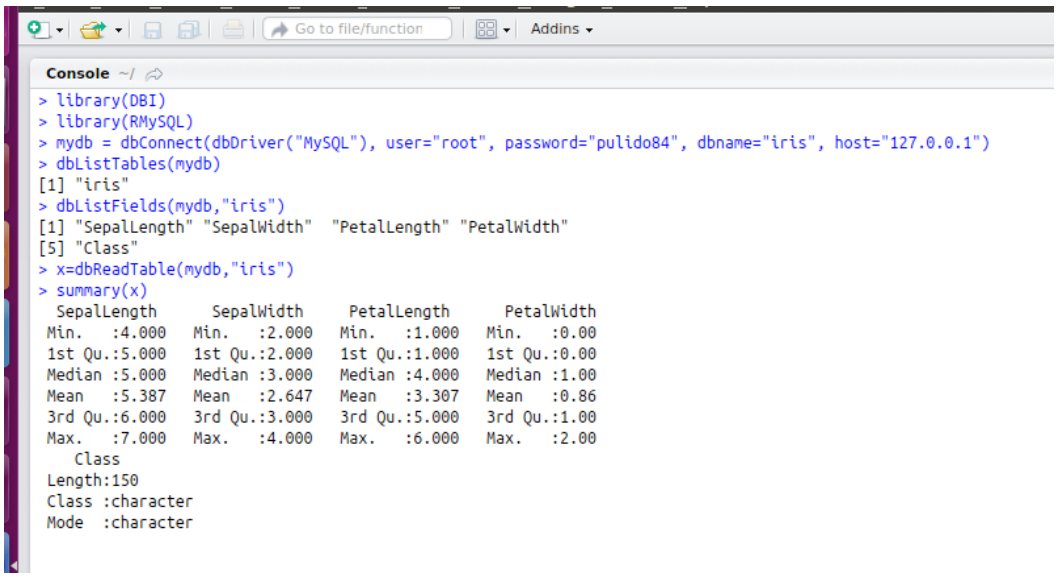
El estadístico Kappa [46] además de indicar la exactitud con la que se ha realizado la predicción, indica que el argumento sobre el que se computa el código (iris$Class) puede usarse para controlar los factores a los que está asociado, permitiendo así una mejora sustancial en la predicción de los datos.

Con este ejemplo se pretende justifica la elección de R al tener un formato de computación menor que el requerido de haberse utilizado Python y que la conexión con el servidor web se puede obtener con el comando EXEC () de PHP.

B) Modelo Iris Flower data set

Para realizar una aproximación inicial y comprender así las mecánicas de ML, se recurrió a un modelo más sencillo. El conjunto de datos utilizados para este propósito se seleccionó por su amplia aceptación como modelo de estadístico y ejemplo típico para el desarrollo de aplicaciones de ML.

El paquete de datos Iris, recogido por Fisher [47], se compone de la medición de 150 especímenes de 3 especies (setosa, virginica y versicolor) correspondientes al género Iris. En cada espécimen de la colección se le practicaron 4 mediciones, la anchura y la longitud de los pétalos y sépalos. Con las 4 mediciones Fisher determino un modelo en el que podría diferenciar entre las 3 especies de Iris [7].



Tabla/Figura 31 Resumen (summary) de la BBDD Iris con Rstudio

El modelo Iris permite una clara diferenciación entre el aprendizaje supervisado y el no supervisado.

El aprendizaje supervisado al tratarse de una disciplina de la programación reciente no presenta una definición al uso; sin embargo, se ha llegado al consenso de que el modelo supervisado supone la capacidad que puede presentar una máquina para aprender sin necesidad de que este programado para ello [49].

En este modelo supervisado es necesario que se tengan las “soluciones” sobre las que se va a trabajar antes de realizar el análisis discriminante; suponiendo como ejemplo el conjunto de

datos de Iris la clasificación por ML (kmeans) [22] realizada se contrastaría frente a las categorías de especie de Iris (Iris$class), obteniendo así una relación de aciertos/fallos del proceso de clasificación (Tabla/Figura 32).

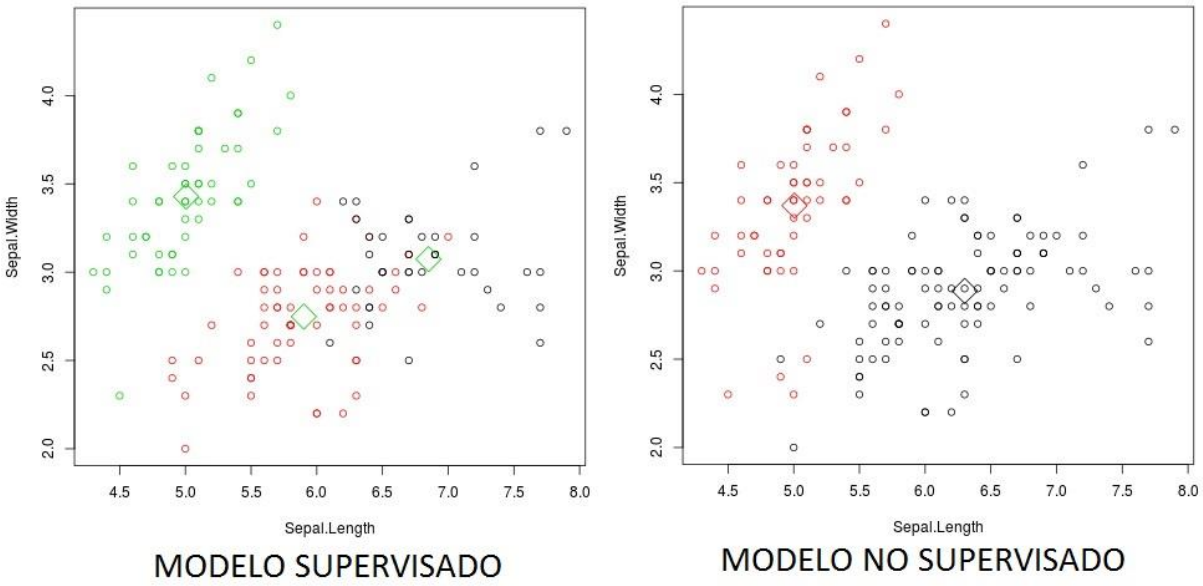
```
> table(y,kc$cluster)

y               1  2  3
  setosa\r      0  0 50
  versicolor\r 24 26  0
  virginica\r  48  2  0
>
```

Tabla/Figura 32 Relación de las especies de Iris y el clúster asignado por el comando kmeans ()

En la anterior Tabla/Figura se puede observar que la clasificación por grupos no siempre corresponde al 100% respecto a los datos originales, sabiendo que en los datos originales de Fisher [47] hay 50 observaciones de plantas por especie. Con respecto a la especie setosa se comprueba que todos sus registros se han agrupado en un mismo clúster, en contraste con la especie versicolor cuyo mayor núcleo está compuesto por 48 observaciones perdiendo dos que son catalogadas dentro del tercer clúster en el que se encuentra el grueso de los datos de la especie virginica.

Por otro lado, el modelo no supervisado se ajusta a las observaciones desconociendo las categorías a las que pertenece cada elemento, siendo tratadas como variables aleatorias. Este es la principal diferencia entre los modelos supervisados y los no supervisados [48].

Si no se tuviera la relación de especies para el data set de Iris se podría llegar al error de que solamente habría 2 especies en lugar de las 3 descritas, ocasionado porque parte del conjunto de versicolor se fusiona con los de virginica (Tabla/Figura 33).



Tabla/Figura 33 Previsión clustering entre un modelo supervisado y no supervisado en ML tomando la BBDD iris como modelo

Se entiende que el modelo supervisado presenta una mayor exactitud que el modelo no supervisado, puesto que los avances en inteligencia artificial se encuentran en fases iniciales.

Este es el principal argumento por el cual se procedió a realizar un aprendizaje supervisado con los modelos propuestos en la BBDD iris y la BBDD chrom15.

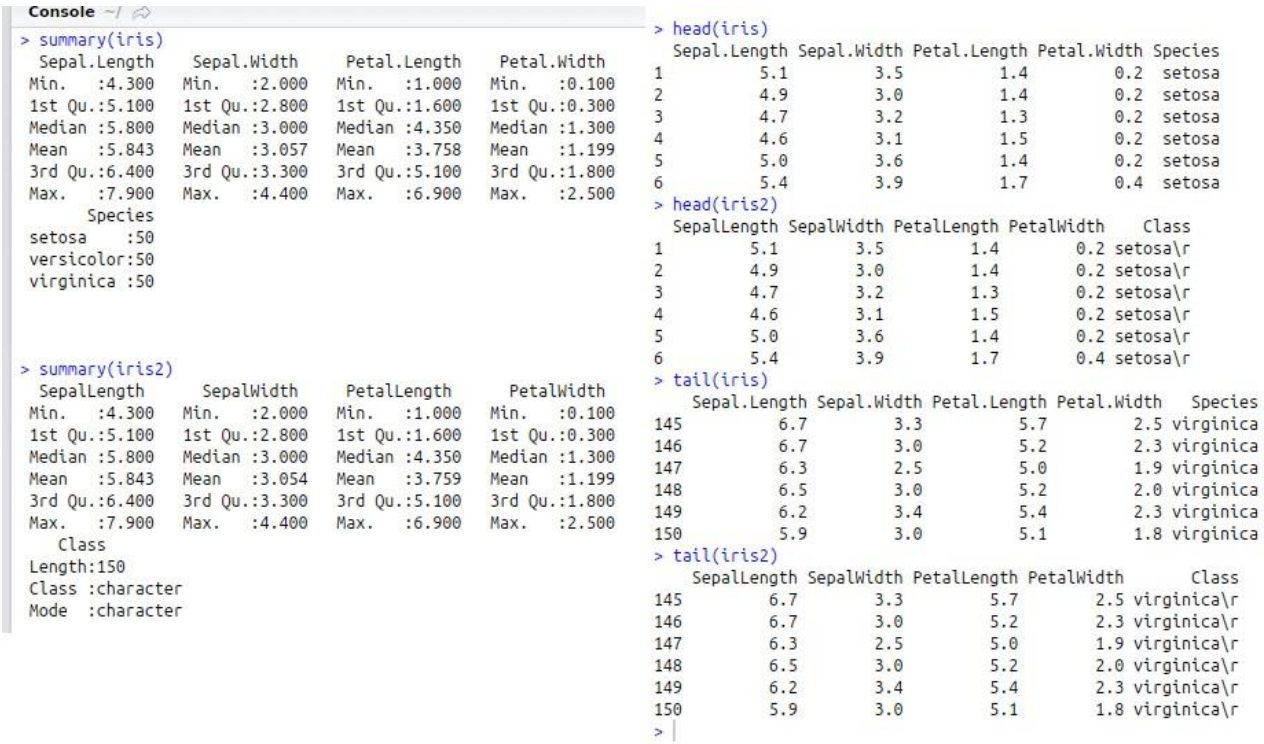
La elección de Iris es por tanto la opción más lógica como modelo simplificado antes de trabajar con los datos de HapMap, debido además a que el código generado y descrito en esta sección podrá ser fácilmente extrapolado a los datos del TFM simplemente cargando los datos correspondientes.

Otro motivo por el cual usar Iris es para determinar si la importación de los datos desde una BBDD podría necesitar de ajustes previos en la consola de Rstudio para obtener la misma calidad de resultados que los datos cargados desde el propio programa, permitiendo obtener una previsión de la forma de utilización de las librerías de R DBI [37] y RMySQL [38].

El código realizado para el análisis de Iris por R se encuentra en el repositorio del GIT bajo el nombre "RcodeTFM" [22]. Durante esta sección se realizarán las diferentes partes del código y serán contractadas con lo obtenido por medio del comando DATA () y el uso de LIBRARY(RMySQL) [38].

Para diferenciar los datos que procedan desde la BBDD de aquellos que vienen ya instalados con el software, se denominarán como iris a los que proceden del propio software y iris2 a los de la BBDD.

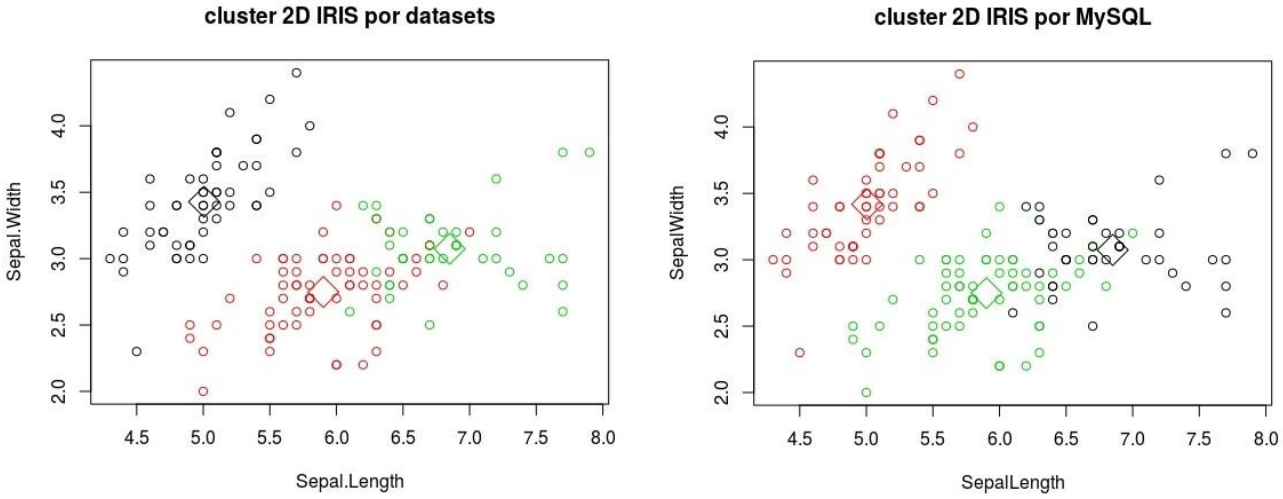
La entrada de datos por DATA () o READ.TABLE() será considerada la parte no común del código de "RcodeTFM" [22], sin embargo, con el comando SUMMARY () se obtendrá un resumen del conjunto de datos y con los comandos HEAD () Y TAIL () se obtendrán los primeros y últimos registros de los datos cargados.



Tabla/Figura 34 Comparativa entre la carga de Iris por medio de DATA (Iris) y RMySQL (Iris2)

Se comprueba por tanto que la carga de datos y el contenido de ambos es exactamente el mismo por lo que se puede deducir que la aplicación de un algoritmo obtendrá el mismo resultado en

ambos paquetes de datos (Tabla/Figura 34 y 35). Al confirmarse que ambos data sets se corresponden entre, los siguientes pasos se realizaran utilizando la BBDD iris desde MySQL.



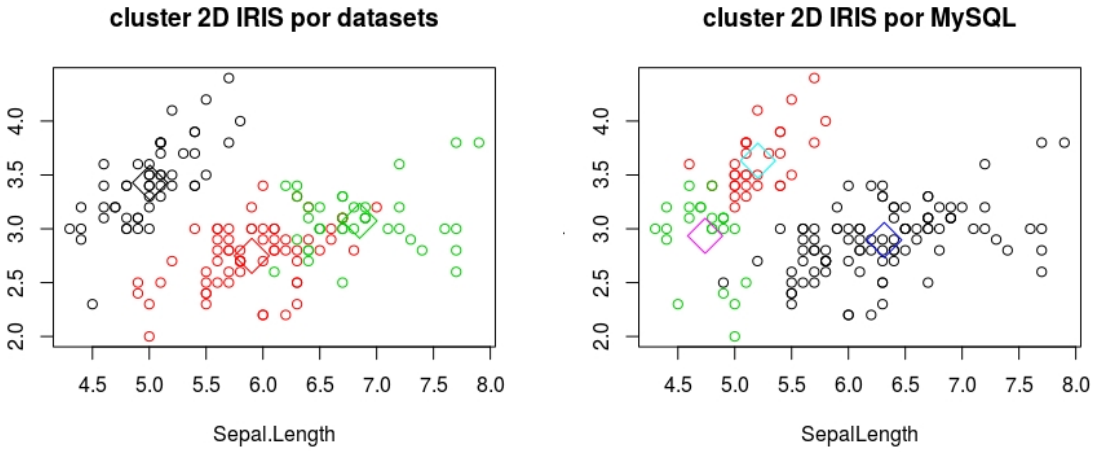
Tabla/Figura 35 Clúster de Iris data por Data (izqda.) y por BBDD (dcha.)

Antes de continuar examinando los datos de iris para el ML, es importante recalcar que la importación de los datos debe ser realizada correctamente, puesto que un simple error al introducir o a cargar los datos en la BBDD puede ocasionar errores de lectura de esta BBDD. Un error común y que puede afectar al resultado es la inclusión de una línea de registro extra con información falsa o sin ella.

Este fallo puede ocasionar la aparición de outliers que distorsionen el análisis originando para el caso de iris una cuarta clase o alterar las agrupaciones como se observa en la Tabla/Figura 32 por lo que una clasificación para una clase obtiene registros de una clase diferente.

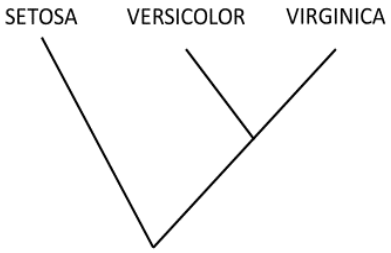
El aprendizaje automático no es del todo perfecto y de momento depende de la interpretación del investigador para obtener conclusiones.

Por lo que se refiere a esta cuestión una posible interpretación es que en ocasiones se pueden dar que la similitud de los registros, aunque sean de categorías (en iris especies) diferentes sean catalogadas dentro del mismo, y al indicarle que existen cierto número de grupos se fuerce al sistema a realizar correcciones que conducirán a interpretaciones erróneas (Tabla/Figura 36).



Tabla/Figura 36 Ejemplo de un clustering mal realizado (dcha.) comparado con el correcto (izqda.)

De igual manera que con ML pueden realizarse agrupaciones en clústeres de los datos como en este caso determinar el número de especies presentes en un conjunto de datos dados y obtener conclusiones como que para el caso de Iris versicolor y virginica probablemente estén más relacionadas entre sí que si se comparan con setosa.



Tabla/Figura 37 Posible árbol filogenético del genero Iris según los datos de Fisher

Prosiguiendo con el análisis, se debe agregar que ML permite otras aplicaciones como pueden ser:

- Aprendizaje, entrenamiento (learning, training): Reproducir patrones conocidos para realizar predicciones basadas en estos patrones.
    - Ingeniería de factores: Consiste en ARREGLAR los datos para una correcta aplicación del algoritmo basado en predicciones de datos.
    - Arboles de decisiones
- Regresión linear: Calculo del número de componentes principales (PCA) de un modelo en el que sus atributos se hallan relacionados entre sí. [60]
- Minería de datos: Uso de ML para descubrir patrones desconocidos en los datos, tiene función exploratoria.

En vista de las múltiples aplicaciones que ofrece ML se procederá a entrenar al ordenador para que determine patrones en la BBDD iris. El entrenamiento consistirá en la selección de un número de entradas que servirán como ejemplo de los patrones a buscar por el ordenador. Mientras tanto otras líneas quedaran sin catalogar para que el ordenador aplique la lógica adquirida durante el entrenamiento para así determinar a qué categoría corresponde cada

entrada. Como resultado se tendrá una lista comparativa entre la predicción realizada y la categoría real.

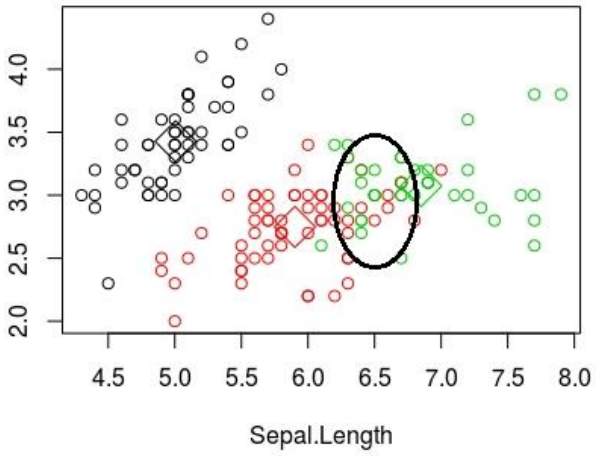
En la Tabla/Figura 30 se muestra el resultado del entrenamiento. Puede observarse que el valor de kappa es de 0,96 lo que indica un casi perfecto acuerdo entre la predicción y los datos originales. Igualmente, el p-valor es lo bastante bajo como para ser rechazado en un posible contraste de hipótesis.

Dicho lo anterior el valor kappa no es perfecto porque el sistema ha cometido un falle en la clasificación puesto que ha referenciado una entrada catalogada como versicolor en los datos originales como virginica en las predicciones (Tabla/Figura 38).

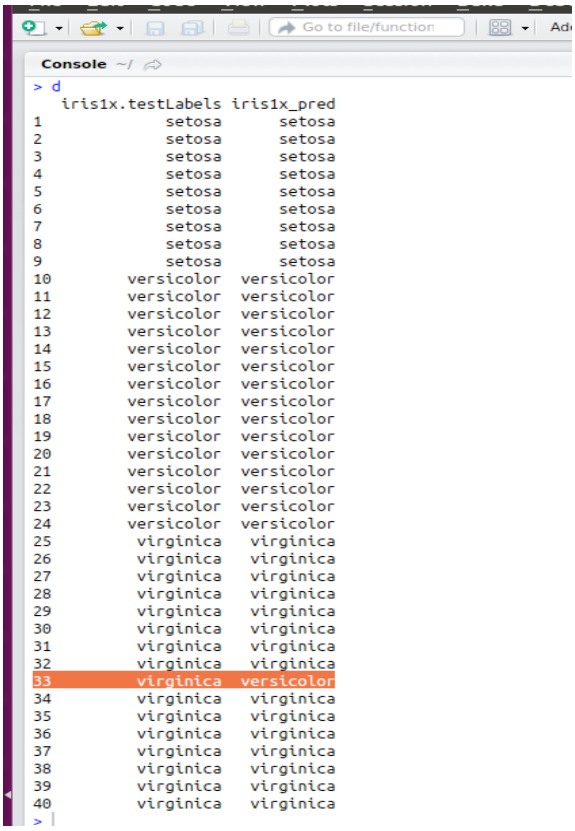
	Reference		
Prediction	setosa	versicolor	virginica
setosa	9	0	0
versicolor	0	15	0
virginica	0	1	15

Tabla/Figura 38 Comparación entre la catalogación original y la obtenida por ML en BBDD iris

Este hecho parece confirmar que ambas especies (virginica y versicolor) son tan próximas que harían falta más parámetros para incluirlos en el análisis y así independizar más la nube de puntos que genera y evitar la zona de fusión entre ambas (Tabla/Figura 39). Con un simple comando de R podemos interrogar al aprendizaje sobre el registro que ha cambiado de categoría (Tabla/Figura 40).



Tabla/Figura 39 Zona de fusión entre las especies virginica y versicolor



Tabla/Figura 40 Resultado de la comparación entre los datos originales y la predicción, remarcando el fallo de predicción de categoría en la BBDD iris

En resumen, se puede inferir que para este conjunto de datos el aprendizaje realizado sobre los datos de la BBDD iris es bastante bueno por lo que, de introducirse nuevos datos no catalogados según la especie en la BBDD, habrá un 97,5% de que dicho registro se catalogue en su correspondiente categoría de especie.

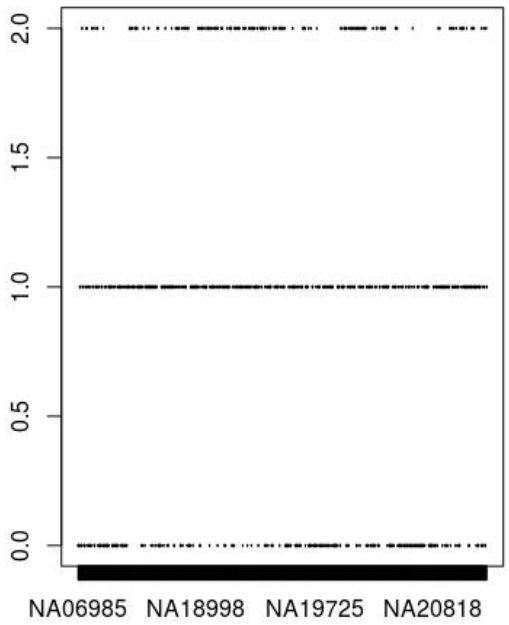
El código para la realización de este análisis se encuentra en el archivo “RcodeTFM” [22].

C) HapMap Cromosoma 15

En cuanto al uso de los datos del repositorio HapMap, cabe recalcar que, debido al gran volumen de archivos implicados, su completa utilización queda descartada debido a que los tiempos de cargas en un ordenador medio supondría que cualquier acción por mínima que fuera podría ocasionar fallos en el mismo por falta de memoria RAM o procesadores. Esa bajada de rendimiento puede fácilmente ser comprobada simplemente abriendo un archivo cualquiera de HapMap en formato de txt o Excel. En vista del inconveniente de recursos informáticos disponibles para la realización de la aplicación del código de aprendizaje automático se seleccionó el cromosoma 15 y una serie de poblaciones como se describió en el apartado de “Datos del consorcio HapMap”.

Por lo que se refiere a la inferencia del código descrito en el apartado anterior tal como están los datos de partida, como se muestran en la Tabla/Figura 5, no podría realizarse ya que la mayoría de los campos son categóricos de modo que los datos se superpondrían unos encima

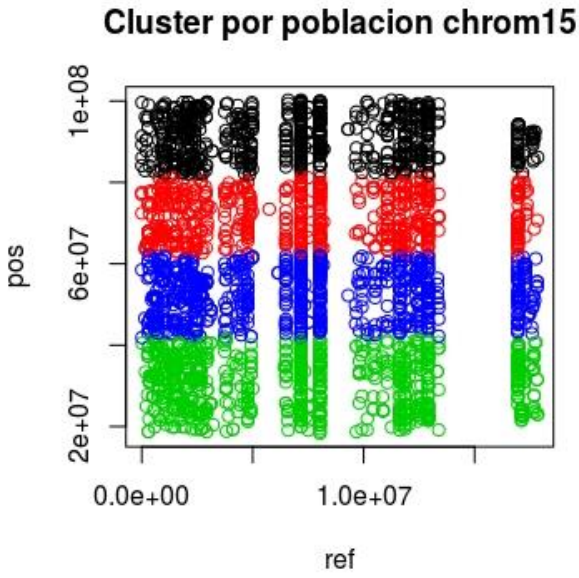
de otros enmascarando la información contenida en ella (Tabla/Figura 41) imposibilitando la interpretación de los resultados.



Tabla/Figura 41 Solapamiento de datos usando los datos originales de HapMap

Otro factor que se tuvo en cuenta para la realización grafica de los clústeres en la utilización de aquellos campos que serían fácilmente convertibles de un factor categórico a uno numérico (variable rs#) pasarían a ser útiles para la realización de una gráfica si se ayudasen de la variable pos.

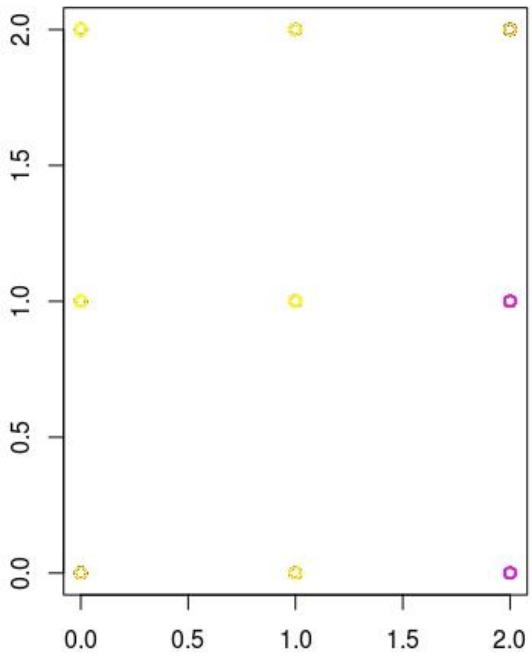
Por más que se intentase modificar los campos y la información contenida en ella provocaba un error de clasificación debido a que utilizaba la variable "pos" como delimitador de la diferenciación de clases por lo que dos referencias con el mismo valor de posición en el cromosoma 15 pero perteneciente a dos poblaciones distintas (p. ej. CEU y YRI) serían considerados parte del mismo clúster y no diferentes como debieran serlo (Tabla/Figura 42).



Tabla/Figura 42 Error de clustering usando los datos originales del cromosoma 15

Este inconveniente propicio replantearse la forma de tratamiento de los datos. Para ello se recurrió a un análisis de componentes principales (PCA) en el que supondremos que los datos contenidos en ellos tienden a una distribución normal y siempre y cuando el número de observaciones sea mayor de 50 para que así pueda aplicársele el teorema central del límite [3]. Razón por la cual se realizó en Excel el tratamiento de los datos referidos al cromosoma 15 descrito en el apartado "Datos del consorcio HapMap" puesto que el en proceso se utilizó la moda de los datos referentes a los individuos participantes para rellenar aquellos campos de los que no se tuviera constancia, normalizando en el proceso a la población en relación a los SNP´s obteniendo así lo presentado en la Tabla/Figura 12.

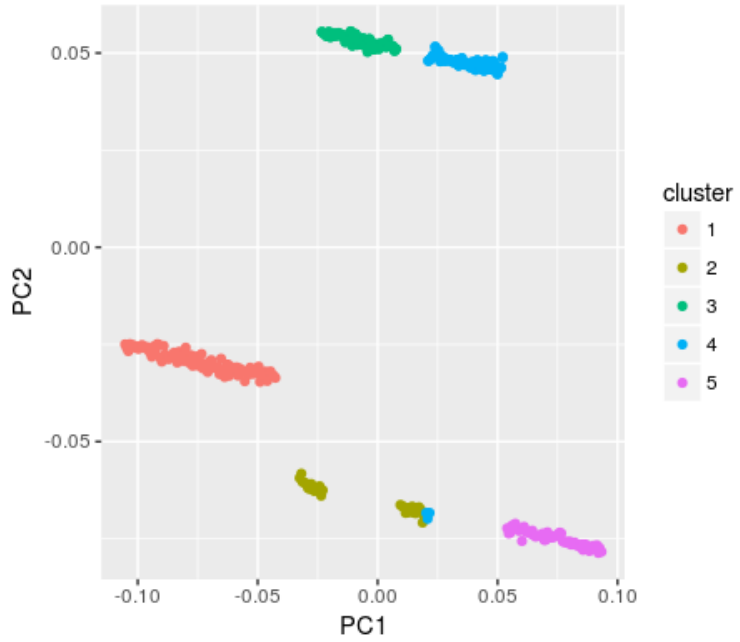
De no realizarse un PCA a los datos y se intentase una representación análoga a la realizada con la BBDD iris, utilizando en este caso dos campos de referencia de SNP´s para representar a los 360 individuos estudiados, obtendríamos la siguiente Tabla/Figura.



Tabla/Figura 43 Representación errónea sobre la BBDD Chrom15 después de su procesado

Es por esto por lo que se decidió por una representación por PCA puesto que permitía una clara diferenciación de las nubes de puntos de cada población para mejorar la representación gráfica. En el proceso de elaboración de la mejor gráfica se emplearon una serie de librerías R [33-44] que ofrecían una serie de herramientas para su implementación. Sin embargo, al finalizar el proceso, las librerías utilizadas para la representación gráfica fueron las siguientes:

- Ggfortify [44]
- Cluster [35]



Tabla/Figura 44 Clustering por PCA de BBDD Chrom15 (considerando al grupo ASIA) usando la función autoplot ()

Como se aprecia en la Tabla/Figura anterior y a diferencia de la mostrada en la Tabla/Figura 43, las nubes de puntos correspondientes a las distintas poblaciones son perfectamente reconocibles e independientes permitiendo así un análisis. La simplicidad e interpretación de la gráfica de la Tabla/Figura 44 se corresponderá con formato elegido para la representación de los datos en el portal web.

De igual manera se realizaron modelos de entrenamiento sobre la BBDDs del cromosoma 15 para determinar si la inclusión del grupo denominado ASIA (CHB+JPT) supondría una mejora del modelo o no (Tabla/Figura 45).

```
> confusionMatrix(asia.testLabels,asia_pred)
Confusion Matrix and Statistics

          Reference
Prediction ASIA CEU CHB JPT YRI
      ASIA   18   0   4   8   0
      CEU     0  22   0   0   0
      CHB    13   0   0   0   0
      JPT    12   0   0   3   0
      YRI     0   0   0   0  30

Overall Statistics

               Accuracy : 0.6636
                 95% CI : (0.5673, 0.7509)
    No Information Rate : 0.3909
    P-Value [Acc > NIR] : 6.777e-09

                  Kappa : 0.558
 Mcnemar's Test P-Value : NA
```

```
> confusionMatrix(panel.testLabels,panel_pred)
Confusion Matrix and Statistics

          Reference
Prediction ASW CEU CHB JPT LWK MEX MKK TSI YRI
       ASW  21   0   0   0   0   0   0   0   0
       CEU   0  26   0   0   0   0   0   0   0
       CHB   0   0  13   4   0   0   0   0   0
       JPT   0   0   3   9   0   0   0   0   0
       LWK   0   0   0   0  32   0   0   0   0
       MEX   0   0   0   0   0  34   0   0   0
       MKK   0   0   0   0   0   0  29   0   0
       TSI   0   0   0   0   0   0   0  33   0
       YRI   0   0   0   0   0   0   0   0  31

Overall Statistics

               Accuracy : 0.9702
                 95% CI : (0.9396, 0.9879)
    No Information Rate : 0.1447
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9662
 Mcnemar's Test P-Value : NA
```

```
> confusionMatrix(noasia.testLabels,noasia_pred)
Confusion Matrix and Statistics

          Reference
Prediction CEU CHB JPT YRI
       CEU  22   0   0   0
       CHB   0  10   3   0
       JPT   0   2  10   0
       YRI   0   0   0  30

Overall Statistics

               Accuracy : 0.9351
                 95% CI : (0.8549, 0.9786)
    No Information Rate : 0.3896
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.909
 Mcnemar's Test P-Value : NA
```

Tabla/Figura 45 Entrenamiento de las diferentes BBDD para la obtención del estadístico Kappa

La inclusión del grupo ASIA (Tabla/Figura 45 izqda.) supone un deterioro del modelo debido al bajo valor de Kappa y de exactitud mostrado con respecto a la BBDD en la que se eliminó ASIA como categoría (Tabla/Figura 45 dcha.).

Atendiendo a la escala de kappa para el acuerdo entre dos interpretaciones [46] se puede apreciar que la incorporación de la población ASIA al modelo genera un acuerdo "moderado" frente a un acuerdo "casi perfecto" en "noASIA", donde se excluyeron las repeticiones de entradas que se correspondían a "JPT" o "CHB" y que fueron catalogadas como ASIA. En la parte central de la Tabla/Figura 45 se añadieron más poblaciones al análisis para comprobar que el modelo puede ampliarse a un conjunto de datos mayor y se aprecia que a mayor número de entradas para predecir (y por ende a entrenar) el sistema mejora su capacidad predictiva. Acorde con lo obtenido por este análisis la BBDD utilizada para la realización de clasificación de datos en el portal web será la BBDD "noASIA".

D) Resultados

La relación de resultados en este punto es:

- Código para el clustering de los datos y su representación gráfica.
- Código para la predicción de clases y su análisis por el test estadístico Kappa de Cohen.

Los conjuntos de datos se agruparán dentro de un mismo archivo llamado "RcodeTFM" [22] indicando por medio de apartados la función del código y que se realiza en cada línea del mismo en relación a los resultados mostrados en este apartado.

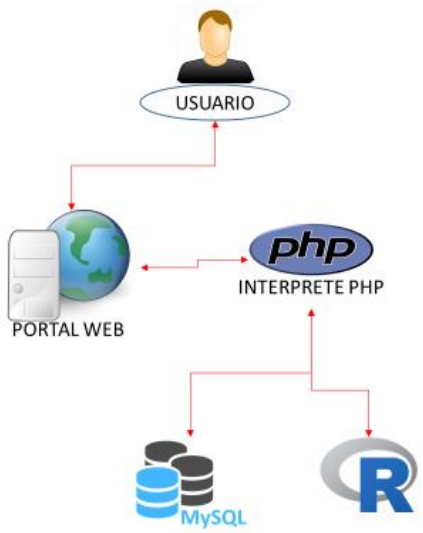
E) Ventajas e Inconvenientes

- Ventajas
    - Uso de un modelo más sencillo para la elaboración del código.
    - Fácil extrapolación del código al conjunto de datos.
- Inconvenientes
    - Selección de las herramientas adecuadas para la fase del TFM (librerías, paquetes de instalación, …)
    - Posibilidad de errores de conexión con el portal y con la BBDD.
    - Los datos deben estar correctamente procesados

PHP programación web

A) Elección del motor de programación web

La interacción entre el usuario y la página web desarrollada se hará a través del servidor web por medio de la dirección IP (en modo local o localhost) en la que se encuentra la información que se requiere (Tabla/Figura 46) de modo que a través del código incluido en cada elemento PHP de la página realice las conexiones y acciones que en ellas se encuentre.



Tabla/Figura 46 Diagrama de flujo de los componentes del TFM

Por tanto, este elemento es el más importante del TFM. Sobre él debe volcarse todo lo realizado anteriormente. Debe estar cohesionado entre los diferentes componentes que lo constituyen para que el portal web y por ende el proyecto salga adelante y constituya un éxito [50]. Este factor podría considerarse limitante si se tratase de una página web corporativa o de divulgación de algún tipo, sin embargo, el portal web desarrollado funciona únicamente en modo local (localhost) y tiene un carácter educativo.

La creación de esta aplicación web se le exigirá el mismo baremo de creación que si se tratase de una página abierta al público. Por tanto, su interface deberá cumplir alguno de los siguientes puntos [51]

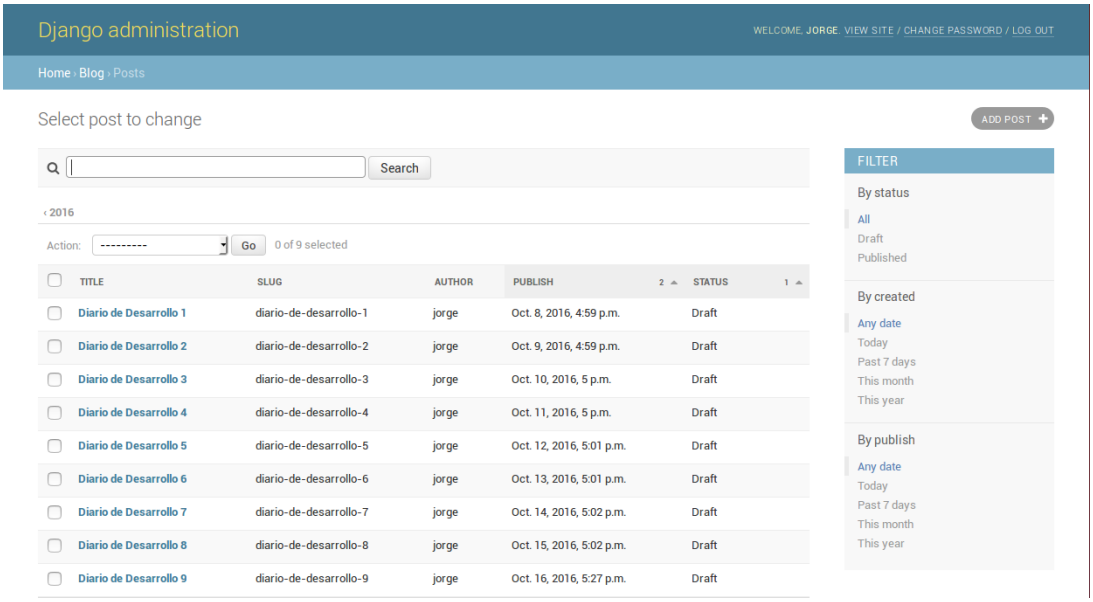
- Conocer al público objetivo al que va dirigida la pagina
- Usar párrafos cortos
- Incluir hiperenlaces a paginas útiles relacionadas con el tema y el publico
- Evitar el síndrome del click and scroll: Dada la magnitud del proyecto esta opción no es aplicable debido a que sobresaturaría la página web con información entrando en conflicto con el segundo punto citado.
- Colocar lo más importante en la parte de arriba para que el usuario no tenga que buscar.

Estos consejos serán tomados en cuenta para que el entorno web sea lo más user-friendly y por ende más atractivo al internauta.

Aun atendiendo a los consejos descritos para el desarrollo web, la principal cuestión reside en el motor web se va a trabajar atendiendo a las siguientes cuestiones.

1) Conocimientos básicos previos de la tecnología a emplear

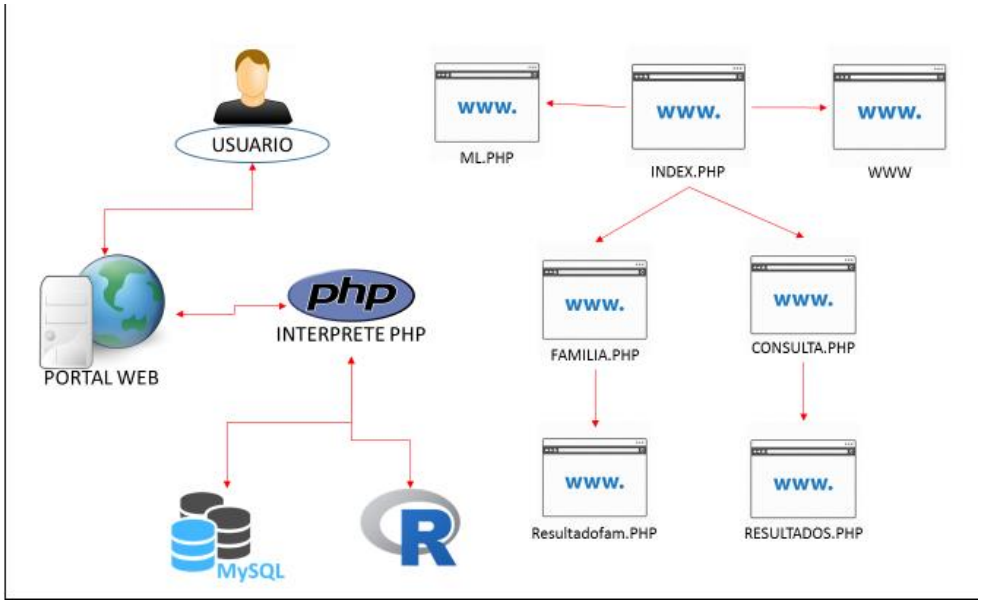
Durante el desarrollo del proyecto se ha estado sugiriendo que una de las opciones posibles para este proyecto sería haberlo realizarlo en lenguaje Python utilizando Django para ello [52]. Se estuvo considerando la idea, incluso se realizaron pruebas conceptuales [53] (Tabla/Figura 47) usando este motor, pero se desechó la idea por las siguientes circunstancias.



Tabla/Figura 47 Idea descartada del desarrollo de un blog del TFM en Django

El estudiante no tenía conocimientos previos de Django por lo que deberían ser adquiridos durante el proceso del TFM, sin embargo, los costes de energía y tiempo para la adquisición de las habilidades para ello habría supuesto una prolongación del calendario del proyecto.

Se eligió en su lugar el motor de PHP para la realización del proyecto considerando que todos los elementos seleccionados para la realización de cada parte del TFM se relacionan entre ellas (Tabla/Figura 48) por medio de una llamada en su código a pesar de que con el uso de Python se hubieran necesitado menos elementos implicados para alcanzar los objetivos previstos.



Tabla/Figura 48 Diagrama de flujo del portal web en PHP

Los conocimientos en el leguaje de PHP, adquiridos previamente, para este proyecto se encontraban en consonancia con los estándares establecidos para cumplir los requerimientos propuestos de manera, que su utilización permitió un reajuste del calendario para priorizarse en áreas en las que se requirieran más recursos como podría ser el procesado a través de MySQL o el código de R.

2) Simplicidad del sistema de gestión de archivos

La estructura en carpetas sobre la que Django trabaja suponía un problema a la hora de encarar el proyecto, en Django la unidad principal es el proyecto, el cual puede presentar varias aplicaciones, pero en contra una aplicación no puede ser usada por dos proyectos a menos que se encuentren duplicadas. En la siguiente Tabla/Figura 49 el proyecto se denominaría "TFM" y la aplicación "HapMap".

```
TFM
├── mysite
|       __init__.py
|       settings.py
|       urls.py
|       wsgi.py
├── manage.py
└── hapmap
    ├── migrations
    |       __init__.py
    ├── __init__.py
    ├── admin.py
    ├── models.py
    ├── tests.py
    └── views.py
```

Tabla/Figura 49 Estructura de archivos del proyecto TFM y la aplicación HapMap creados por Django

Obsérvese que en el proceso se crean multitud de carpetas y archivos que son necesarios para su correcta ejecución siendo "manage.py" el programa Python que deberá ser ejecutado en el terminal para la carga de la/s páginas web por medio del comando PYTHON MANAGE.PY RUNSERVER y para actualizar los contenidos generados en estas carpetas el comando MIGRATE.

Un ejemplo de la aplicación de estos comandos se encontraría en actualizar el archivo "settings.py" de la carpeta "mysite" para utilizar MySQL como BBDD ya que por defecto se genera una conexión con otra BBDD. A diferencia que ocurre con el servidor apache2 [A1] que en su instalación se incluyen tanto la BBDD MySQL, PHP y el servidor web; en Django es necesario que se instalen los paquetes de datos necesarios para la conexión con otras BBDD [53]. Tras la instalación y antes de ejecutar el comando de MIGRATE, en el archivo "settings.py" (Tabla/Figura 50) [54] indicándole los siguientes parámetros necesarios para la conexión como:

- Engine: Motor de la BBDD.
- Name: Nombre de la BBDD a utilizar.
- User: Root por defecto.
- Password: Contraseña para iniciar el programa de BBDD.

```
# Database
# https://docs.djangoproject.com/en/1.10/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'hapmap',
        'USER': 'root',
        'PASSWORD': 'pulido84',
        'HOST': 'localhost',
        'PORT': '',
    }
}
```

Tabla/Figura 50 Configuración archivo settings.py para permitir la conexión con MySQL

La escritura en HTML [2] no requiere más que un editor de texto elementar, como emacs o un archivo txt y verse en modo local en cualquier navegador. La estructura se divide en dos partes principales, la cabecera (<head>) y el cuerpo (<body>) teniéndose en cuenta que mientras no haya errores de sintaxis el texto englobado en el cuerpo será mostrado en el navegador (Tabla/Figura 51).

En contraste con lo establecido para Django, apache2 no requiere de un sistema de carpetas tan elaborado, sino que los archivos PHP deben estar localizados en la carpeta "/var/html/www/" del SO Ubuntu, para luego ser cargada a través del buscador web. Los cambios que se generen en los archivos PHP (por medio del editor de textos) pueden ser fácilmente actualizados para su visualización únicamente refrescando la página (F5) permitiendo así un mayor dinamismo en la depuración de la página sin requerir de reconexiones con el servidor a través del terminal para observar los cambios como sucedería con Python.

3) Simplicidad de sintaxis de programación

Para consolidar la idea del complejo intricado de archivos necesarios para la visualización web en Python es necesario una correcta sintaxis entre los archivos "urls.py" de la carpeta de la aplicación y del proyecto Django. Si no se encontraran bien referenciadas el resultado sería una página que no se puede cargar para visualizar (white screen of death). Es probable que pudiera meterse comandos de textos para determinar el punto en el que ha fallado la conexión en el código de la página web, pero por los motivos que se han determinado en párrafos anteriores no se siguió utilizando esta vía de trabajo por lo que no se pudo contrastar.

La sintaxis utilizada por Python difiere de la utilizada en PHP por el uso de "templates" o plantillas en las que se cargan las bibliotecas necesarias al principio con el comando IMPORT FROM…AS… y la necesidad de definir como objetos cada elemento que compondrá la página web [53 y 61].

Por el contrario, en PHP basta únicamente con la ejecución de un sencillo código para comprobar la eficacia y sencillez con la que PHP opera. Con el siguiente ejemplo basta para ilustrar lo dicho en párrafos anteriores (Tabla/Figura 51). Cualquier cambio que se realice entre las entradas <body> </body> será cargado dentro del servidor de manera inmediata. Además de que la configuración de opciones de fuente/párrafo/estilo/… disponibles en cualquier procesador de textos pueden ser utilizados por un sencillo comando (p.ej. para realizar encabezados con <h1>" texto" </h1>).

Tabla/Figura 51 Primeros pasos dentro de la programación web

Como se ha dicho, la elección de PHP como motor de programación web se debió a las siguientes características:

- Mejor conocimiento del lenguaje de programación.
- Rapidez de depuración de errores o formatos.
- Necesario únicamente el explorador de internet y el archivo PHP para ejecutarse.
- Conectividad con los otros componentes de la estructura web (BBDD y R).

B) Conexión entre MySQL y PHP

En el anexo 5 [A5] se detalla el código necesario para comprobar la conexión entre PHP y MySQL a través del servidor apache. Esta comprobación puede determinarse de dos maneras:

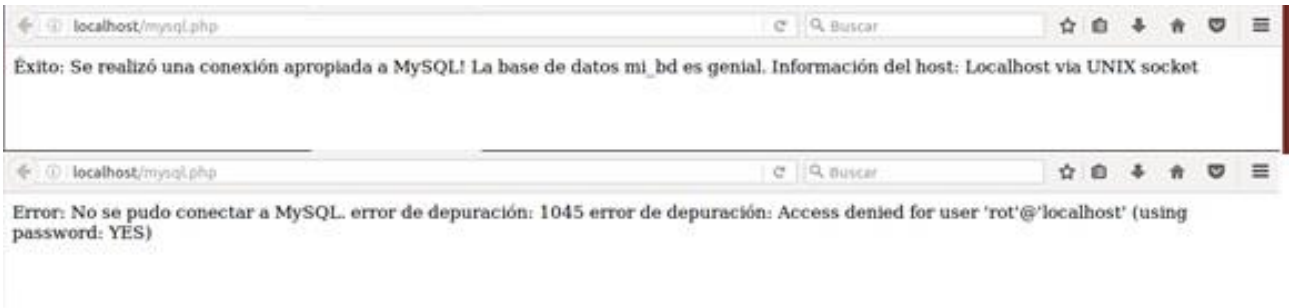
- A través del archivo info.php creado durante la instalación de LAMP [A1] (Tabla/Figura 52).
- A través del archivo mysql.php creado a partir del código descrito en el anexo 5 [A5] (Tabla/Figura 53).

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.12-dev - 20150407 - $Id: 241ae00989d1995ffcbbf63d579943635faf9972 $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	*no value*	*no value*
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off
mysqli.rollback_on_cached_plink	Off	Off

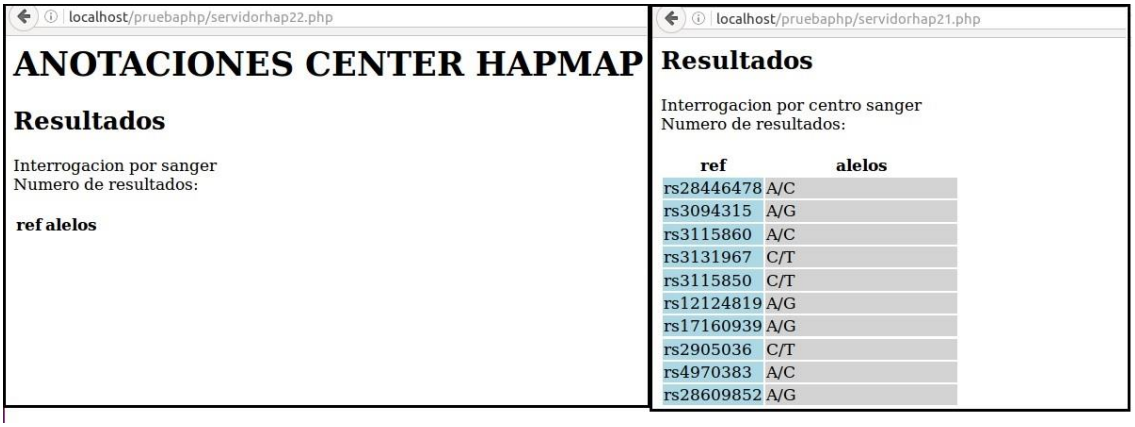
Tabla/Figura 52 Archivo info.php (no incluido en el GIT [22])



Tabla/Figura 53 archivo mysql.php (no incluido en el GIT [22]) para comprobar la correcta conexión entre MySQL y PHP

Con el objetivo de la creación de los diferentes servidores web que compondrán el TFM, se realizaron pruebas conceptuales de carga de la BBDD HapMap para depurar la forma en la que debían mostrarse los denominados servidores HapMapresult.php [22] y familiaresult.php [22].

Antes de proseguir se hará una mención a que la escritura para la conexión entre PHP y MySQL deben estar en consonancia con las versiones instaladas. Con esto se quiere recalcar que cualquier anomalía o referencia durante la escritura en PHP a la hora de vincularlo con la BBDD puede ocasionar la llamada “PHP's white screen of death” (Tabla/Figura 54 izqda.).



Tabla/Figura 54 Comparación entre obtener la white screen of death (izqda.) y lo esperado (dcha.)

Ahora bien, se ha comentado que este error se debe a un fallo entre las versiones de escritura sobre las que se trabaja (PHP 5 o PHP 7) ya que como se remarca en los cuadrados rojos en la Tabla/Figura 55 el cambio de "mysql" por "mysqli" durante la ejecución del script ocasiona la aparición del fallo de interpretación. En algún momento entre las versiones de PHP 5 y PHP 7 hubo algún problema con las licencias de uso entre MySQL y Apache por lo que el código que funcionaba en PHP 5 ocasionaba la PHP's white screen of death si se aplicaba en PHP 7.

```
15
16  $mysqli = new mysqli("localhost", "root", "pulido84", "hapmap");
17
18  /* comprobar la conexión */
19  if (mysqli_connect_errno()) {
20      printf("Falló la conexión: %s\n", mysqli_connect_error());
21      exit();
22  }
```

Tabla/Figura 55 Fragmento del código de HapMapresult [22] para la conexión con MySQL remarcando la diferencia necesaria para su correcto funcionamiento en PHP 7

Al código de servidores que mostraran los resultados (HapMapresult.php [22] y familiaresult.php [22]) de consultas a las BBDD se le incorporo un contador de resultados para mostrar los registros que se corresponden con las opciones programadas por durante la consulta. Se implementó esta opción para la determinación del número de entradas de HapMap se verían afectadas ya que el trabajo de ML se realizó con una BBDD con menor número de entradas y con diferente formato. En la Tabla/Figura 56 se muestra la sección del código de HapMapresult.php [22] donde se encuentra la función de contador remarcada en rojo y en la Tabla/Figura 57 la comparación entre realizar la misma búsqueda en MySQL y PHP. Este código mostrado también permite que se muestren todos los resultados de la consulta en pantalla.

```
27  if ($resultado = $mysqli->query($consulta)) {
28          /* determinar el número de filas del resultado */
29          $numero = mysqli_num_rows($resultado);
30          /* obtener el array de objetos */
31          echo "<h2>Resultados</h2>";
32          echo "Interrogacion por los parametros siguientes: $ref,$alleles,$c
33          echo "Numero de resultados: $numero<br><br>";
34          echo "<table>";
35          echo "<tr><th>ref</th><th>alelos</th><th>Chrom</th><th>posicion</th
36          for ($i=0;$i<$numero;$i++)
37          {
38          $fila=mysqli_fetch_array($resultado);
```

Tabla/Figura 56 Código para la incorporación de la opción contador en el servidor HapMapresult

Tabla/Figura 57 Comprobación de una misma query en MySQL y PHP

C) Conexión R-PHP

La conexión entre R y PHP se realiza de una manera diferente a la descrita entre MySQL-PHP. Este cambio se debe a la propia naturaleza de los motores implicados puesto que R es un sistema enfocado a objetos de modo que estos son guardados en la memoria de ejecución y ser utilizados según se valla leyendo el código.

Existen múltiples opciones por las que se pueden cargar graficas usando el lenguaje de PHP [56 y 57] pero estas presentaban una estructuración y complejidad superior a los propios conocimientos que se tenían del propio lenguaje, además implicaba la utilización de archivos Json [58] los cuales habrían implicado la introducción en el lenguaje Java [59] y su aplicación en el proyecto ocasionando retraso en el programa establecido. Al poseer conocimientos lenguaje R la adaptación para realizar ensayos de ML fue más rápida que la que se hubiera necesitado tanto para usar Java como Python.

El otro motivo de la elección de R para integrarse en PHP es la posibilidad de ejecutar programas/archivos externos a través de PHP por medio del comando EXEC () [55] (Tabla/Figura 58). En esta Tabla/Figura se encuentra enmarcadas en cuadros de colores las partes más

importantes del código y a continuación se encuentra comentado en el mismo color que el recuadro la explicación de la ejecución de dicho código.

De este modo seremos capaces de cargar un archivo de R en el que se encuentre el código a ejecutar. Teniendo en cuenta que con PHP se podrá seleccionar una variable (número de clúster a identificar) que entrará en la ruta del archivo de R devolviéndonos la solución del código (en el caso del TFM una imagen de clustering) y que el archivo PHP mostrará.

ml.php

```php
<?php
// ml.php

echo "<form action='ml.php' method='get'>";
echo "Numero de agrupaciones (cluster) a generar: <input type='text' name='N' />";
echo "<input type='submit' />";
echo "</form>";

if(isset($_GET['N']))
{
  $N = $_GET['N'];        Parametro de entrada de Cluster, introducido por el usuario

  // execute R script from shell
  // this will save a plot at temp.png to the filesystem
  exec("/usr/lib/R/bin/Rscript ch15.R $N");        Funcion de ejecucion de un archivo externo a PHP

  // return image tag
  $nocache = rand();
  echo("<img src='temp.png?$nocache' />");        Resultado del script de R y que sera mostrado en ml.php
}
?>
```

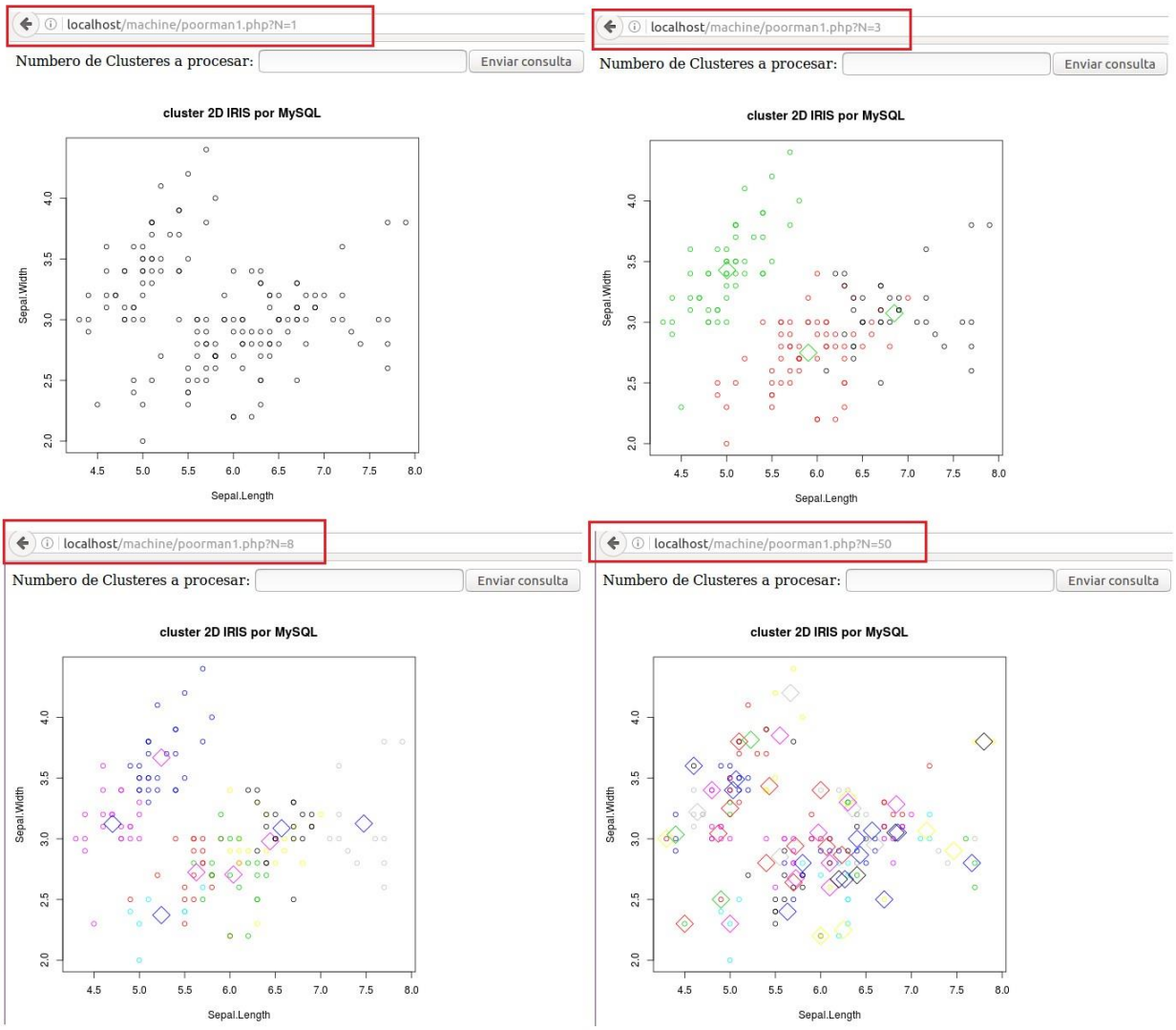
ch15.R

```r
library(DBI)
library(RMySQL)        Librerias requeridas para cargar la grafica y la BBDD
library(ggfortify)
library(cluster)
mydb = dbConnect(dbDriver("MySQL"), user="root", password="pulido84", dbname="chrom15", host="127.0.0.1")        Conexion BBDD
ch15=dbReadTable(mydb,"chrom15")
args <- commandArgs(TRUE)
N <- args[1]        Solicita el argumento introducido por el usuario en ml.php
png(filename="temp.png", width=500, height=500)
autoplot(pam(ch15[-3003], N), frame = TRUE, frame.type = 'norm')        Grafica a procesar
dev.off()
```

Tabla/Figura 58 Código de los archivos necesarios para la ejecución del ML en PHP

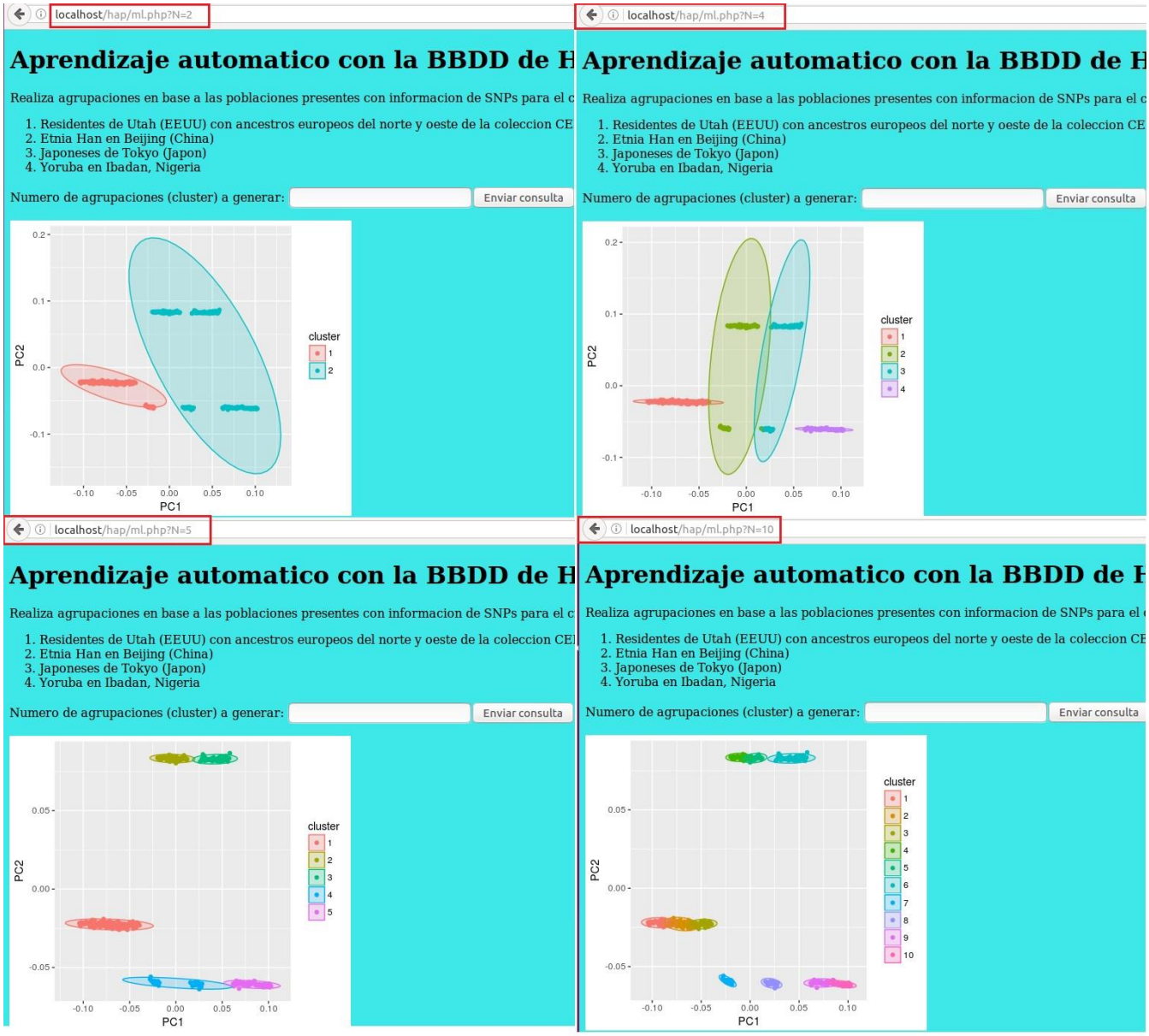
Se debe agregar que para que los programas puedan compartir información de modo bidireccional se le deben dar permisos de lectura a las carpetas implicadas en a través del comando SUDO CHMOD 777 “ruta/archivo” para que así no haya errores al ejecutarse el comando EXEC ().

Por lo que se refiere a su aplicación, como en el apartado de ML se trabajó con el modelo de la BBDD iris. Siguiendo el formato establecido en las figuras anteriores se consiguió procesar la BBDD iris para realizar el clustering (Tabla/imagen 59).



Tabla/Figura 59 Clustering BBDD iris corriendo en el motor de PHP con diferente número de clústeres (1,3,8,50)

Después se procedió a la aplicación del código R para la ejecución del clustering por PCA de los datos de la BBDD Chrom15 (Tabla/Figura 60), donde se procedió a realizar un clustering basado en el código descrito en la Tabla/Figura 58 ya corriendo en la versión final del archivo "ml.php", en esta imagen se puede observar que en remarcada en rojo se encuentra la dirección web del servidor con la selección de clúster a generar introducida por el usuario y que la selección se corresponde con la imagen generada con el código como se puede comprobar en R con el código correspondiente de "RcodeTFM.txt" [22].



Tabla/Figura 60 Clustering BBDD Chrom15 corriendo en el motor de PHP con diferente número de clústeres (2,4,5,10)

D) PORTAL WEB

Una vez realizada las pertinentes comprobaciones de que los diferentes elementos del sistema se hayan correctamente conectados entre ellos se procedió a editar los diferentes archivos PHP para su maquetado final. Se analizarán los diferentes componentes remarcando las partes más relevantes de su código y su aplicación en el TFM.

I. Index.php

La elaboración del archivo index.php se construyó atendiendo a los conceptos para desarrollar una página web user-friendly [51]. En la Tabla/Figura 61 se encuentran las partes del código más relevantes usadas en index.php [22]:

- Estructuración en lista
- Saltos de pagina
- Botón de acción

- Carga de imágenes
- Tabla en formato PHP
- Hiperenlaces

En la Tabla/Figura 61 se encuentra remarcado en recuadros de colores las partes más significativas y exclusivas de cada archivo PHP, junto al que se encuentra la explicación del código en el mismo color que su recuadro. A continuación, en la Tabla/Figura 62 se mostrará la imagen de la ejecución del código PHP en su versión final.

Este formato de se repetirá para cada archivo PHP del TFM, salvo el de ml.PHP que ya ha sido descrito en la Tabla/Figura 58, remarcando las partes más importantes (Tabla/Figura 63 - 66).

INDEX.PHP

```
11  <p>
12  <ol>
13  <li> <a href="#primero">Iniciar busquedas</a>
14  <li> <a href="#segundo">Que es un SNP? Como nos afectan?</a>
15  <li> <a href="#tercero">Proyecto internacional hapmap</a>
16  <li> <a href="#cuarto">Participantes proyecto hapmap</a>
17  <li> <a href="#quinto">Enlaces de interes</a>
18  </ol>
19  </p>
22  <p>
23  INICIA UNA BUSQUEDA en hapmap
24  <form action=consulta.php method=post>
25  <input type=submit name=submit0 value=Iniciar_Busqueda>
26  </form>
27  </p>
84  <a name="cuarto"><h2>Participantes proyecto hapmap</h2></a>
85  <p>
86  <img src=map.jpeg height=300 width=600>
87  </p>
88  Los participantes del proyecto fueron 1421 personas (incluidos grupos familiares) de todo el mundo repartidas en 11 grupos etnicos de origen diverso y relacionadas.
89  <table BORDER=1 WIDTH=600>
90            <tr>
91               <td width=100 height=100 ALIGN=CENTER>Perfil Genetico</td>
92               <td width=100 height=100 ALIGN=CENTER>Codigo</td>
93               <td width=100 height=100 ALIGN=CENTER>Num. de participantes</td>
94               <td width=100 height=100 ALIGN=CENTER>Perfil Genetico</td>
95               <td width=100 height=100 ALIGN=CENTER>Codigo</td>
96               <td width=100 height=100 ALIGN=CENTER>Num. de participantes</td>
97               <td width=100 height=100 ALIGN=CENTER>Perfil Genetico</td>
98               <td width=100 height=100 ALIGN=CENTER>Codigo</td>
99               <td width=100 height=100 ALIGN=CENTER>Num. de participantes</td>
100              <td width=100 height=100 ALIGN=CENTER>Perfil Genetico</td>
101              <td width=100 height=100 ALIGN=CENTER>Codigo</td>
102              <td width=100 height=100 ALIGN=CENTER>Num. de participantes</td>
103           </tr>
147 <a name="quinto"><h2>Enlaces de interes</h2></a>
148 <p>
149 <a href=ftp://ftp.ncbi.nlm.nih.gov/hapmap//genotypes/2010-08_phaseII+III/forward/>
150 Descarga los archivos
151 </a>
```

Estructuracion enformato lista ->
Saltos de pagina ->

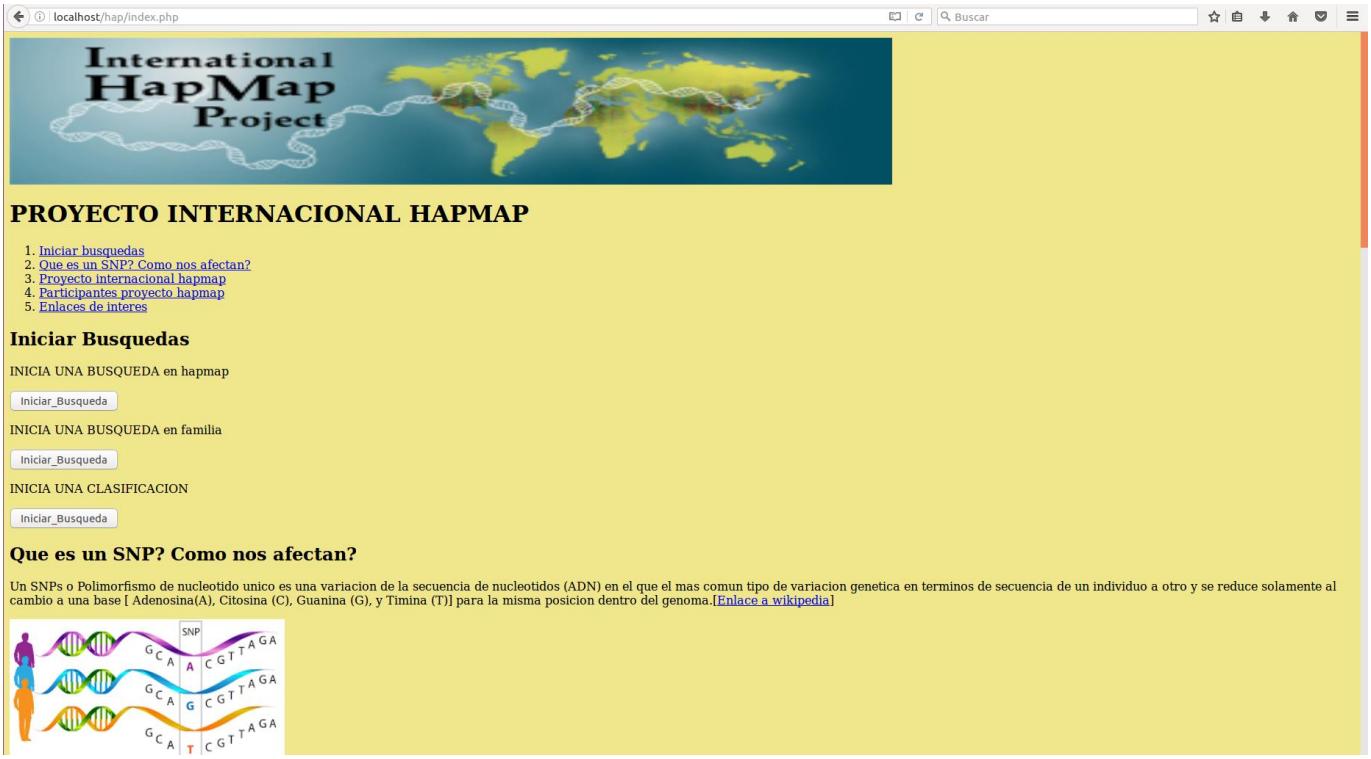
Boton accion para acceder al archivo php indicado

Imagen cargada desde el archivo contenedor de los archivos php

Tabla en formato PHP -> <tr> </tr>

Hiperenlace a una pagina web externa

Tabla/Figura 61 Partes más relevantes de la página archivo index.php



Tabla/Figura 62 Página principal del TFM (index.php) versión final

II. Consulta.php y familia.php

En la Tabla/Figura 63 se incluyen las opciones más relevantes usadas para la construcción de estos archivos (Consulta.php y familia.php) [22]:

- Botón de acción para iniciar una búsqueda en MySQL tres seleccionar unos campos.
- Menús desplegables de selección.
- Hiperenlaces a paginas externas.

Ambos archivos (Consulta.php y familia.php) se construyeron de forma análoga ya que la función a realizar, interrogar a una BBDD, es la misma en ambos casos. En el recuadro rojo de la Tabla/Figura 63 se muestra un ejemplo de formulario desplegable en este recuadro se debe programar correctamente los parámetros que los servidores HapMapresult.php y familiaresult.php deben incorporar a la query de MySQL. Es por tanto necesario el procesado realizado y descrito en el apartado de "Creación y manejo de BBDD MySQL" y visto en la Tabla/Figura 20.

Consulta.php y Familia.php

```
11  BUSCAR UN POLIMORFISMO SNP EN HUMANOS BASADO EN LOS PARAMETROS A SELECCIONAR:
12  <br><br>
13  <form action=hapmapresult.php method=post>
14  <label>Por favor, seleccione los parametros sobre los que se quiere buscar:</label>
15  <br>
16  <p>
17  Clave de identificacion de referencia
18  <input type=text name=ref value=ref10>
19  <br><br>
20  <br>
21  Alelos
22  <select name=alleles size=1>
23  <option value=>-</option>
24  <option value=-/A>-/A</option>
25  <option value=-/C>-/C</option>
26  <option value=-/G>-/G</option>
27  <option value=-/T>-/T</option>
28  <option value=A/C>A/C</option>
29  <option value=A/G>A/G</option>
30  <option value=A/T>A/T</option>
31  <option value=C/G>C/G</option>
32  <option value=C/T>C/T</option>
33  <option value=G/T>G/T</option>
34  </select>
118 <ol>
119 <li> Sanger
120 <a href=http://www.sanger.ac.uk/>Sanger main web page</a>
121 <li> Broad: Instituto Broad servicio de genomica
122 <a href=http://genomics.broadinstitute.org/>Broad main web page</a>
123 <li> Perlegen
124 <a href=https://www.ncbi.nlm.nih.gov/probe/docs/distrperlegen/>Perlegen main web page</a>
125 <li> affymetrix
```

Boton de accion para acceder a hapampresult.php

Menu desplegable para el campo SNPalleles de MySQL. El usuario indicara de esta manera por cual alelo quiere interrogar a la BBDD. El resto de los campos que forman el formulario de consulta.php o familia.php se construyeron del mismo modo. Para evitar elecciones no deseada se incorporo la opcion <option value=>-</option> Asi no se interrogara por ese campo a la BBDD.

Leyenda adjunta para un mejor uso por parte del usuario, se crean hipervinculos para acceder a la paginas web oficiales de por ejemplo los centros que han participado en el proyecto hapmap.

Tabla/Figura 63 Partes más relevantes de la página archivo consulta.php

Tabla/Figura 64 Servidores de consulta BBDD (familia.php y consulta.php) versión final

III. HapMapresult.php y Familiaresult.php

En la Tabla/Figura 65 se remarcan las opciones más relevantes usadas para la construcción de estos archivos (HapMapresult.php y familiaresult.php) [22]:

- Recopilación de los parámetros seleccionados en el formulario anterior.
- Código de búsqueda en MySQL con formato para seleccionar los registros en los que coincidan los parámetros seleccionados.
- Comando para la ejecutar la búsqueda en MySQL.
- Contador de registros de resultados de la consulta.
- Impresión en pantalla por los campos interrogados.
- Comando para la visualización de todos los campos.
- Creación de la tabla donde se van a cargar los datos de MySQL.

Debido a las similitudes en la estructuración y sintaxis que los archivos HapMapresult.php y familiaresult.php presentan se tomó el archivo HapMapresult.php como modelo para destacar las partes más significativas.

Hapmapresult.php y Familiaresult.php

```
<?php
/* 1. Entrada */
$ref = $_POST["ref"];
$alleles = $_POST["alleles"];
$center = $_POST["center"];
$perfil = $_POST["perfil"];
$chrom = $_POST["chrom"];
$protocol = $_POST["protocol"];
```

Parametros recogidos del formulario anterior y que serviran como "objetos" de seleccion en la query de MySQL.

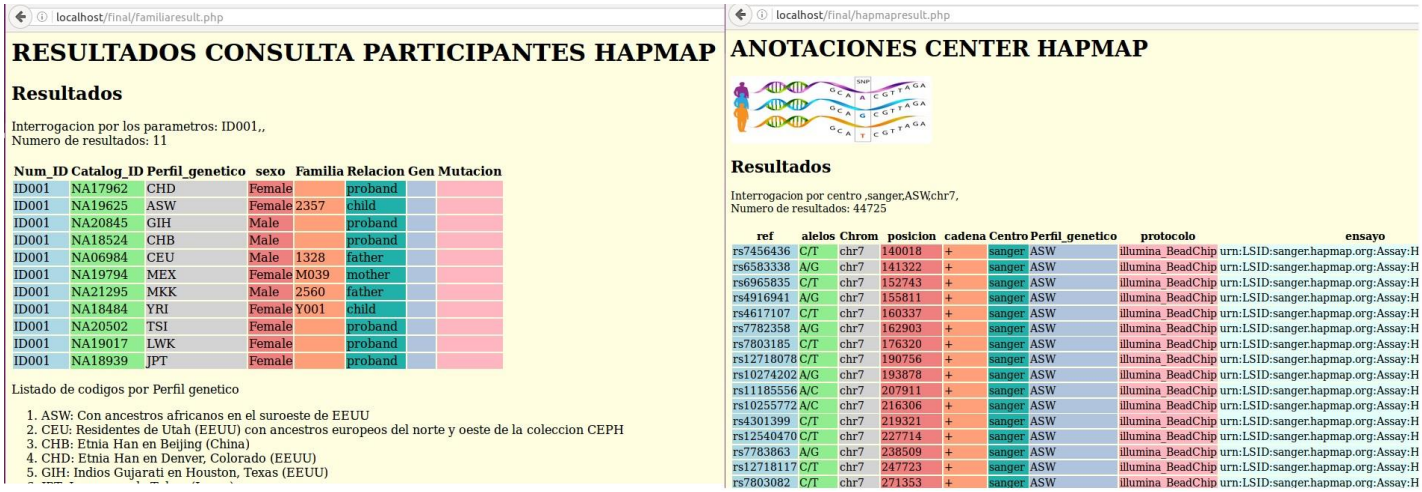
Comando SELECT de MySQL donde usara los parametros anteriores para acotar las busquedas

```
$consulta = "SELECT ref,SNPalleles,chrom,pos,strand,center,panelLSID,protLSID,assayLSID from hapmap where ref like '%$ref%

if ($resultado = $mysqli->query($consulta)) {
        /* determinar el número de filas del resultado */
        $numero = mysqli_num_rows($resultado);
        /* obtener el array de objetos */
        echo "<h2>Resultados</h2>";
        echo "Interrogacion por los parametros siguientes: $ref,$alleles,$center,$perfil,$chrom,$protocol<br>";
        echo "Numero de resultados: $numero<br><br>";
        echo "<table>";
        echo "<tr><th>ref</th><th>alelos</th><th>Chrom</th><th>posicion</th><th>cadena</th><th>Centro</th><th>Perfil_genetic
        for ($i=0;$i<$numero;$i++)
        {
        $fila=mysqli_fetch_array($resultado);
        echo "<tr>";
        echo "<td bgcolor=lightblue>",$fila["ref"],"</td>";
        echo "<td bgcolor=lightgreen>",$fila["SNPalleles"],"</td>";
        echo "<td bgcolor=lightgray>",$fila["chrom"],"</td>";
        echo "<td bgcolor=lightcoral>",$fila["pos"],"</td>";
        echo "<td bgcolor=lightsalmon>",$fila["strand"],"</td>";
        echo "<td bgcolor=lightseagreen>",$fila["center"],"</td>";
        echo "<td bgcolor=lightsteelblue>",$fila["panelLSID"],"</td>";
        echo "<td bgcolor=lightpink>",$fila["protLSID"],"</td>";
        echo "<td bgcolor=lightcyan>",$fila["assayLSID"],"</td>";
        }
        echo "</table>";

        /* liberar el conjunto de resultados */
```

Ejecucion de la consulta de MySQL, contador de registros obtenidos, confirmacion de parametros a interrogar y codigo para la visualizacion de todos los resultados posibles.

Creacion de la tabla para recoger los resultados de la busqueda de MySQL.

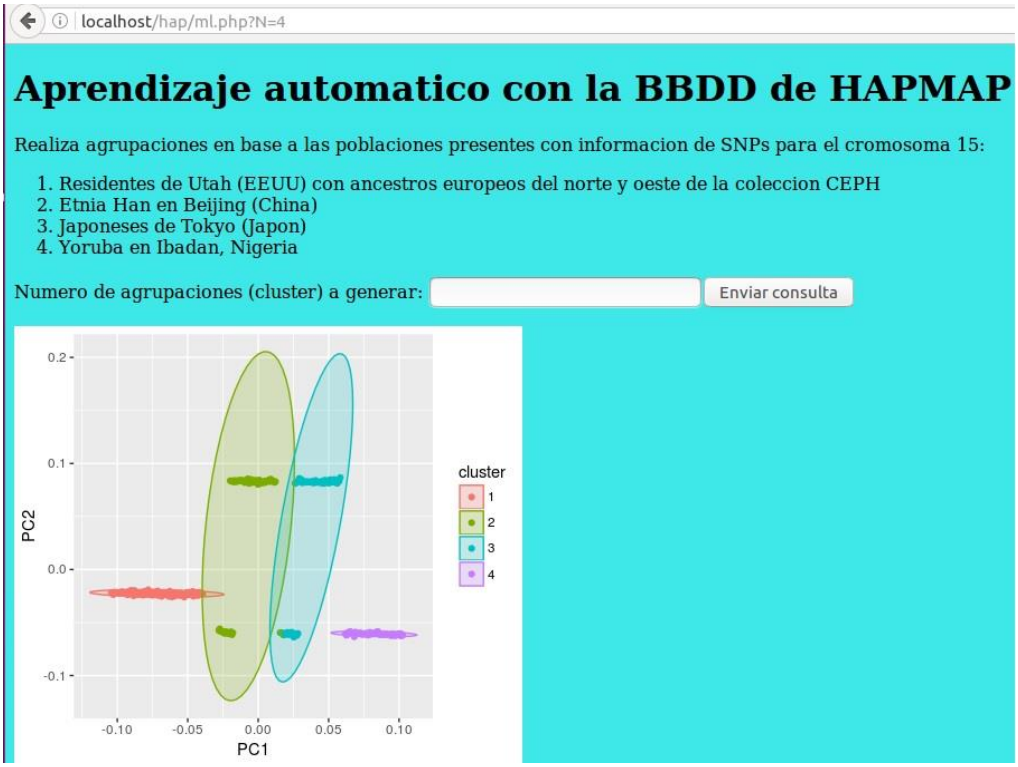
Tabla/Figura 65 Partes más relevantes de la página archivo HapMapresult.php



Tabla/Figura 66 Servidores dedicados a la recogida y visualización de BBDD (familiaresult.php y HapMapresult.php)

IV. ML.php

El código del archivo “ml.php” se encuentra en el repositorio de GIT [22] y cuyo contenido fue comentado en la Tabla/Figura 58.



Tabla/Figura 67 Servidor para la representación de clústeres (ml.php)

E) Resultados

La relación de resultados en la parte final del TFM será la siguiente:

- Página web de cabecera (index.php) del proyecto con información general y enlaces a los diferentes componentes de la web.
- Páginas web enlazadas para la consulta de la BBDD HapMap (consulta.php y HapMapresult.php).
- Páginas web enlazadas para la consulta de la BBDD familia (familia.php y familiaresult.php).
- Página web para la realización de un clúster basado en el aprendizaje automático sobre la BBDD Chrom15 (ml.php) junto con el rscript de ejecución (ch15.R).
- Archivos en formato csv con la información de familia, chrom15 (ASIA [crom15.csv], noASIA [newcrom15.csv] y Panel [fullcrom15.csv]).
- Archivo txt con los códigos usados para el clustering/análisis de las diferentes BBDD en R.
- Carpeta shinyPCA_tfmjorge: Contiene los archivos ui.R y server.R para la realización del clustering por PCA sobre la aplicación web de R la librería shiny [63]

Los archivos arribas descritos se encuentran alojados en el servidor del GIT del proyecto [22] para su consulta.

F) Ventajas/Inconvenientes encontrados durante todo el proyecto

- Ventajas
    - Aprendizaje de diferentes lenguajes de programación.
    - Tratamiento de Big data para la realización de un proyecto de gran envergadura.
- Inconvenientes
    - Que el proyecto original fuera abandonado impidiendo de este modo conseguir los resultados previstos.
    - Manipulación de unos datos incompletos por el argumento anterior y la necesidad de modificar su estructura para ajustarse a los objetivos que se han descrito en esta memoria.

Conclusiones

El proyecto HapMap nació con esta intención, descifrar a través de los SNP´s las diferencias que podrían acompañar o influenciar en enfermedades multigénicas para encontrar patrones en el genoma que pudieran atajar la investigación en años venideros además de mapear las zonas del código genético que difieren entre personas.

La información que se obtiene de nuestro genoma aumenta día a día y por tanto debe ser procesada mediante el uso de herramientas informáticas, por lo que este último campo de trabajo debe actualizarse para poder ejecutar y analizar la información obtenida. Así pues, se necesitan del desarrollo de técnicas de aprendizaje automático para poder inferir sobre un conjunto de datos.

Las aplicaciones del aprendizaje automático se encuentran aplicadas en múltiples aplicaciones como puede ser en YouTube o Amazon donde al usuario puede navegar en estos servidores por medio de las "recomendaciones", las cuales son realizadas en base al historial de consultas realizadas permitiendo al ordenador "aprender" y adelantarse al propio usuario.

Al llegar al término de esta memoria se puede decir que el aprendizaje automático aplicado, ha permitido el aprendizaje de las técnicas necesarias para la búsqueda automática de patrones sobre un conjunto de datos y su clasificación posterior.

Durante la realización del TFM se han obtenido los conocimientos necesarios para familiarizarse con los conceptos de aprendizaje automático y su aplicación sobre los datos del consorcio HapMap para realizar una clasificación basada en nuestro caso por la etnia a la que pertenecían los participantes. Para la obtención de este algoritmo fue necesaria la obtención de conocimientos en diversos lenguajes de programación (PHP p.ej.) o en programas de software libre para la realización del análisis de ML.

Siguiendo con el uso de herramientas para el ML se destaca que, aunque el área de trabajo se encuentra en etapas iniciales presenta una vertiente de la informática que podrá ser aplicada en multitud de campos profesionales tales como la economía, robótica por citar algunos ejemplos.

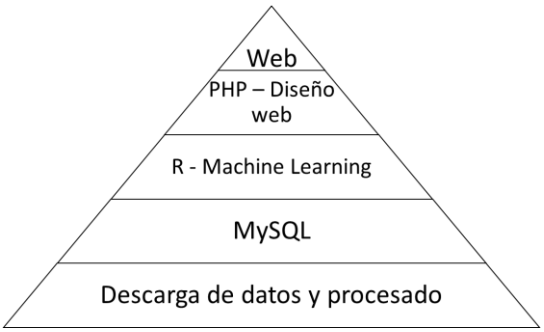
En el transcurso de las tareas descritas en el TFM aparte de una mejor compresión del entorno informático sobre el que se trabaja se ha obtenido una comprensión de las diferentes herramientas informáticas consultadas de modo que no solo permite un crecimiento por parte del estudiante en las áreas referenciadas, sino que también permite la autocrítica una vez realizada de poder haber realizado este TFM de alguna otra forma.

A título personal se tiene que expresar un pesar por la retirada de recursos e investigaciones del proyecto HapMap, el cual presentaba un potencial no explotado que podría haber ampliado el conocimiento del genoma, no solo en lo referente a las llamadas "enfermedades raras" sino que podría también permitir la explicación de los grados de afección de algunas enfermedades.

Sin embargo, otras ramas de las ciencias, como sus posibles aplicaciones a la ciencia forense o de identificación de personas [61-62], se habrían beneficiado del uso de la información contenida en el repositorio HapMap.

En la sección de introducción como objetivo final se planteó la creación de una página web donde ese ejecutaría el algoritmo de ML y permitiría la visualización grafica de la clasificación de los datos en términos de la variable étnica de los mismos en dicha página. Este objetivo final fue descompuesto en objetivos parciales de modo que la suma de sus partes compusiera la página web presentada en esta memoria. A lo largo de la memoria se ha ido describiendo las diferentes tareas realizadas para el cumplimiento de estos objetivos incluyéndose a la finalización de cada fase los resultados obtenidos en ese punto.

La descomposición del TFM en diferentes fases fue necesaria para la elaboración del TFM permitiendo así la obtención de hitos secundarios sobre los que el trabajo se iría construyendo, por este motivo se consideraría el conjunto del TFM como una pirámide de trabajo (Tabla/Figura 68) donde la cúspide seria el producto final.



Tabla/Figura 68 Pirámide de trabajo del TFM

Al llegar al término de este trabajo se puede decir que los objetivos se han cumplido satisfactoriamente obteniéndose una página web escrita en lenguaje PHP sobre la que se ejecuta un código de R para la clasificación de una BBDD SQL. También se incluyó la creación de otras páginas web relacionadas para ofrecer un valor añadido al TFM y presentar así una navegación intuitiva por sus diferentes componentes.

Teniendo como referencia el diagrama de Gantt incluido en esta memoria, se comprueba que la planificación seguida se ha ajustado a los parámetros establecidos, aunque es cierto que respecto a la planificación original se produjo una desviación de la temporización lo que requirió reajustar la misma para que se consiguieran alcanzar las metas propuestas.

La metodología propuesta para este TFM ha ido cambiando según se realizaba, desde cambios de elección de BBDD al servidor web que soportaría la aplicación. Estos cambios se realizaban para ajustar el avance del proyecto al calendario y a la pericia del programador. A lo largo de la memoria se ha ido describiendo los métodos descritos en cada fase y argumentado las decisiones que llevaron a su implementación en el producto final.

También, se determinaron aquellas acciones que fueron tomadas en cuenta pero que no tomaron forma. La elección de un método frente a otro fue determinante para que se pudiera completar el proyecto según calendario. Un ejemplo lo tenemos en la incorporación de R al proyecto y la necesidad de conectarlo con PHP, ya que la opción más lógica hubiera sido la ejecución de todo el código (ML y web) en lenguaje Python. Los cambios, al no limitarse a una plataforma, permitieron obtener una mayor complejidad del entramado con el que los diferentes softwares interactúan entre ellos.

Se puede determinar que a la vista de los resultados obtenidos y aun teniendo en cuenta los cambios de metodología descritos en cada fase del proyecto podría considerarse adecuada.

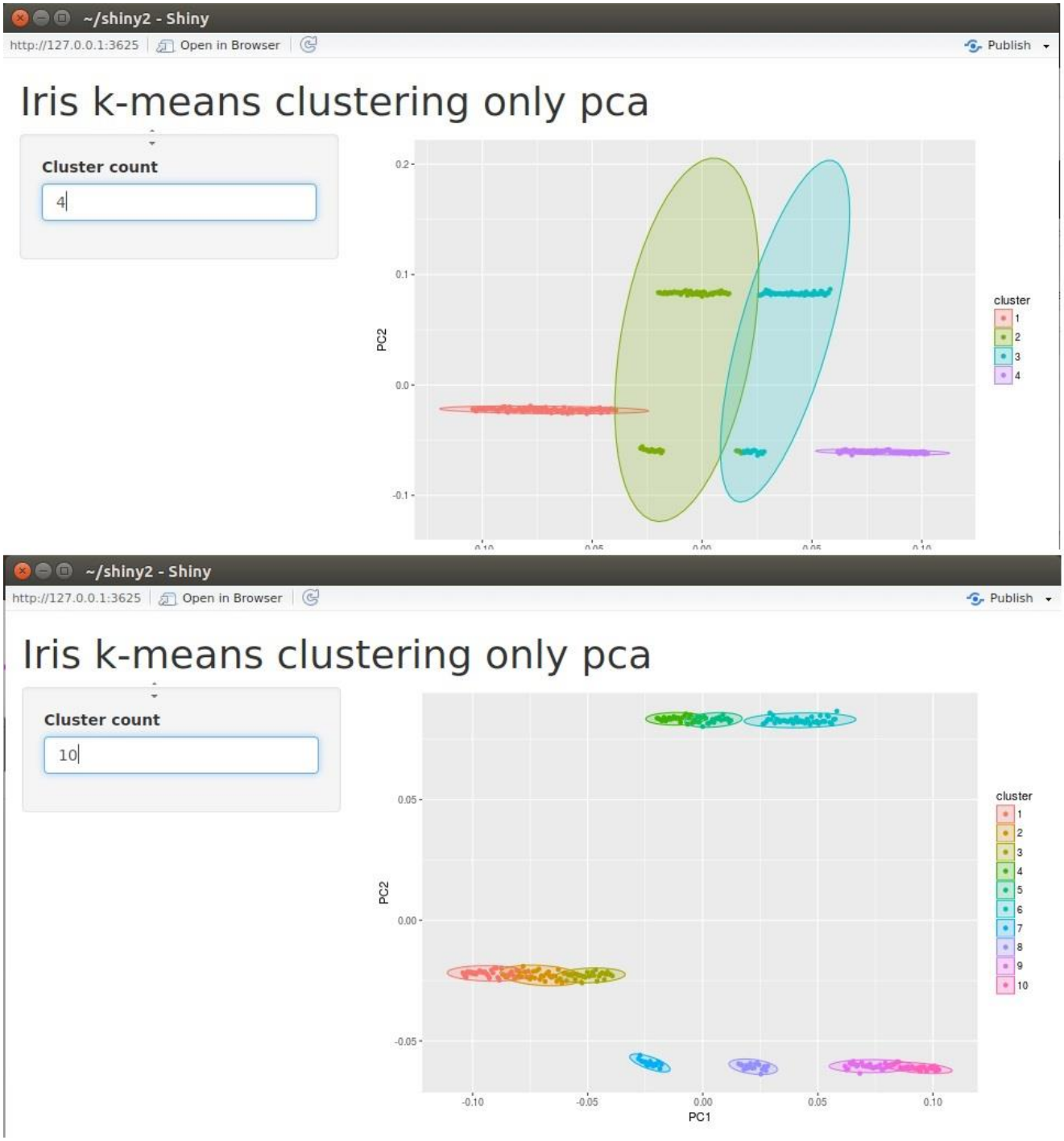
En vista a las herramientas de trabajo mencionadas en esta memoria, pero no implementadas puede determinarse la voluntad del que suscribe para la obtención de los medios necesarios para adquirir los conocimientos necesarios para poder realizar este mismo proyecto de vías diferentes e incluso (a título personal) mejorar el trabajo presentado para dotarlo de más características.

Cabe citar que el uso de Python permite la elaboración de todo este proyecto usando este lenguaje como motor del mismo. Lamentablemente, el diseño web en este lenguaje (Django) estaba fuera del alcance del programador y por ello se tuvo que recurrir a otro motor. Django pretende ser una forma mejorada de hacer páginas web respecto a Wordpress y que actualmente presenta un número considerable de libros de programación con los que se puede hacer aplicaciones como la creación de un portal web con opción de compra por citar un ejemplo.

Otra línea de trabajo que no se ha aplicado y que se espera realizar es aprender las mecánicas necesarias para trabajar con BBDD no relacionales o NoSQL como pueden ser CassandraBD o MongoDB. Aprender a programar en lenguaje java debido a las múltiples aplicaciones que puede ofrecer para el desarrollo web es otro de los objetivos propuestos tras la consecución del TFM.

Para finalizar, durante la elaboración del TFM se encontró una herramienta escrita íntegramente en R para el desarrollo web denominada ShinyR [64]. Esta herramienta de programación suponía un paradigma ya que podría haber permitido realizar el clustering de los datos del cromosoma 15 bajo un solo motor de trabajo sin necesidad de recurrir a HTML/PHP. Dado el conocimiento de R presentado en esta memoria se procedió a reproducir el servidor ml.php bajo el motor de shiny. El resultado se encuentra en la carpeta “shinyPCA_tfmjorge” del GIT [22], en la siguiente imagen (Tabla/Figura 69) se muestra los resultados obtenidos y son comparables a los mostrados en la Tabla/Figura 60.

La aplicación descrita en el párrafo anterior es un ejemplo las líneas de trabajo por explotar y que se ha incluido en esta memoria.



Tabla/Figura 69 Reproducción del servidor ml.php en ShinyR

Glosario

Análisis de componentes principales (PCA): Técnica estadística que consiste en la reducción de la dimensión de los datos a un número menor de variables denominada componentes principales basándose en el uso de mínimos cuadrados.

BBDD: Bibliotecas que contiene los datos de diversas temáticas de manera estructurada y categorizados de diferente manera. Las informaciones contenidas en estas bibliotecas presentan entre si alguna relación entre ellas que es utilizada para su clasificación.

CSV: Formato de archivo para la visualización de datos en forma de tablas.

Django: Esqueleto o armazón para el desarrollo de aplicaciones web gratuito y de código abierto, escrito en Python.

Haplotipos: Conjunto de variaciones que se encuentran a nivel de secuencia del ADN ya sea en un cromosoma, un loci o locus.

HapMap: Proyecto internacional para la catalogación de los haplotipos en el genoma humano para determinar las regiones de semejanza y diferencias entre las personas.

Hardware: Parte física de un ordenador compuesto por los dispositivos ópticos de lectura, placa base, memorias (RAM, ROM), …

***JavaScript Object Notation* (Json)**: Formato de texto utilizado para el intercambio de datos.

LAMP: Acrónimo usado para describir el conjunto de software compuesto por Linux, Apache, MySQL y PHP. La combinación de estas tecnologías es usada para definir y gestionar una estructura web.

MySQL: Sistema de gestión de bases de datos (BBDD) de código abierto basado en lenguaje de consulta estructurado (SQL).

Outliers: Elemento/s de un conjunto de datos que son significativamente diferente al resto provocando una distorsión del patrón de los datos. Su presencia ocasiona la generación de ruido en los datos.

PHP: Lenguaje de programación de código abierto diseñado para el desarrollo de páginas web dinámicas.

Polimorfismos de un solo nucleótido (SNP): Tipo de polimorfismo que ocasiona la variación de un único o unos pocos nucleótidos. Se consideran un tipo de mutación genética.

Precisión: Grado por el cual en estadística el resultado obtenido se corresponde con la realidad.

P-valor: Dato estadístico para utilizado para el contraste de hipótesis, indica la probabilidad de tener un resultado lo bastante extremo como los que se han obtenido.

Python: Lenguaje de programación de orientado a objetos que puede aplicarse desde la creación de aplicaciones al desarrollo web.

R: Entorno y lenguaje de programación de código abierto para el análisis estadístico. Es uno de los lenguajes más utilizados por la comunidad científica.

RAR: Formato de archivos para la compresión de datos para que estos ocupen menos datos.

SO: Conjunto de programas de un sistema informático que gestiona los recursos de hardware y permite al software acceder a los recursos que dispone un ordenador.

Software: Parte intangible de un ordenador, se corresponde con los programas por los que el usuario interacciona con el hardware.

Teorema central del límite (TLC): Teorema estadístico que indica que bajo ciertas condiciones la distribución de los datos se ajusta un una distribución normal o gaussiana.

TXT: Formato de archivo para documentos que carecen de cualquier formato tipográfico constituido únicamente por caracteres.

XLSX: Formato de datos empleados por el software de office Excel. Está constituido por hojas de cálculo para realizar operaciones lógicas y/o aritméticas.

Bibliografía

[1] NCBI retiring HapMap Resource. (Octubre 2016). Retrieved from https://www.ncbi.nlm.nih.gov/variation/news/NCBI_retiring_HapMap/

[2] Enrique Blanco García, Fundamentos de informática en entornos bioinformáticos, Editorial OUC, Barcelona, 2012.

[3] Sunil K. Mathur, Statistical Bioinformatics with R, Editorial Academic Press, Londres, 2010.

[4] Iris Data set. (Noviembre 2016). Retrieved from https://archive.ics.uci.edu/ml/datasets/Iris

[5] Fase II y III de HapMap.(Octubre 2016). Retrieved from ftp://ftp.ncbi.nlm.nih.gov/HapMap//genotypes/2010-08_phaseII+III/forward/

[6] Cromosoma 15 de las poblaciones seleccionadas. (Noviembre 2016). Retrieved from ftp://ftp.ncbi.nlm.nih.gov/HapMap/genotypes/2008-10_phaseII/fwd_strand/non-redundant/

[7] Wikipedia Iris Flower Data test. (Noviembre 2016). Retrieved from https://en.wikipedia.org/wiki/Iris_flower_data_set

[8] The Iris Dataset in Python. (Noviembre 2016). Retrieved from http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

[9] K Means Clustering in R (example). (Noviembre 2016). Retrieved from https://www.r-bloggers.com/k-means-clustering-in-r/

[10] Última versión de ensamblador de secuencia NCBI. (Diciembre 2016). Retrieved from https://www.ncbi.nlm.nih.gov/mapview/stats/BuildStats.cgi?taxid=9606&build=107&ver=0

[11] Sanger main web page. (Octubre 2016). Retrieved from http://www.sanger.ac.uk/

[12] Broad main web page. (Octubre 2016). Retrieved from http://genomics.broadinstitute.org/

[13] Perlegen main web page. (Octubre 2016). Retrieved from https://www.ncbi.nlm.nih.gov/probe/docs/distrperlegen/

[14] Affymetrix main web page. (Octubre 2016). Retrieved from http://www.affymetrix.com/estore/index.jsp

[15] Baylor College of Medicine web page. (Octubre 2016). Retrieved from https://www.bcm.edu/

[16] Illumina main web page. (Octubre 2016). Retrieved from http://www.illumina.com/>Illumina

[17] Instituto de ciencias medicas de Tokyo. (Octubre 2016). Retrieved from http://www.ims.u-tokyo.ac.jp/imsut/en/

[18] Universidad de San Francisco, California. (Octubre 2016). Retrieved from https://www.ucsf.edu/

[19] Universidad McGill (Canada). (Octubre 2016). Retrieved from https://www.mcgill.ca/

[20] Centro medico infantil de Cincinnati. (Octubre 2016). Retrieved from https://www.cincinnatichildrens.org/

[21] Repositorio HapMap del Instituto Coriell. (Octubre 2016). Retrieved from https://catalog.coriell.org/1/NHGRI/Collections/HapMap-Collections/HapMap-Project

[22] GIT tfmjorge. (Diciembre 2016). Retrieved from https://bitbucket.org/PauAndrio/tfmjorge/src/18893d2f1c1b1c92dbf2e0904e5c515cc0866fce?at=master

[23] Ubuntu (Diciembre 2016). Retrieved from https://www.ubuntu.com/download/desktop

[24] Instalacion de software R en Ubuntu. (Noviembre 2016). Retrieved from https://www.datasciencecriot.com/how-to-install-r-in-linux-ubuntu-16-04-xenial-xerus/kris/

[25] Scikit-learn Python web page. (Octubre 2016). Retrieved from http://scikit-learn.org/stable/

[26] Comparacion BBDD. (Octubre 2016). Retrieved from http://db-engines.com/en/system/Cassandra%3BMongoDB%3BMySQL

[27] A Practical Introduction To Cassandra Query Language . (Octubre 2016). Retrieved from http://abiasforaction.net/a-practical-introduction-to-cassandra-query-language/

[28] Learn Git with Bitbucket Cloud. (Octubre 2016). Retrieved from https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

[29] MySQL 5.7 Reference Manual. (Octubre 2016). Retrieved from http://downloads.mysql.com/docs/refman-5.7-en.pdf

[30] Modificar valores de una columna de MySQL. (Noviembre 2016). Retrieved from http://www.aprenderaprogramar.com/foros/index.php?topic=531.0

[31] R web page. (Noviembre 2016). Retrieved from https://www.r-project.org/

[32] Scikit learn. (Noviembre 2016). Retrieved from http://scikit-learn.org/stable/

[33] Manual paquete R "tools". (Noviembre 2016). Retrieve from https://stat.ethz.ch/R-manual/R-devel/library/tools/html/00Index.html

[34] Manual paquete R "fpc". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/fpc/fpc.pdf

[35] Manual paquete R "cluster". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/cluster/cluster.pdf

[36] Manual paquete R "HSAUR". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/HSAUR/HSAUR.pdf

[37] Manual paquete R "DBI". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/DBI/DBI.pdf

[38] Manual paquete R "RMySQL". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf

[39] Manual paquete R "ggplot2". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf

[40] Manual paquete R "caret". (Noviembre 2016). Retrieve from ftp://cran.r-project.org/pub/R/web/packages/caret/caret.pdf

[41] Manual paquete R "e1071". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/e1071/e1071.pdf

[42] Manual paquete R "lattice". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/lattice/lattice.pdf

[43] Manual paquete R "lfda". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/lfda/lfda.pdf

[44] Manual paquete R "ggfortify". (Noviembre 2016). Retrieve from https://cran.r-project.org/web/packages/ggfortify/ggfortify.pdf

[45] Brett Lantz, Machine Learning with R second edition, Packt Publishing Ltd, Birmingham, 2015

[46] Anthony J. Viera y Cols., Understanding interobserver agreement: the kappa statistic, Family Medicine, Fam Med 2005;37(5):360-3.

[47] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals Eugen. 7 (1936) 179-188.

[48] Aprendizaje Supervisado y no Supervisado. (Diciembre 2016). Retrieved from http://redesneuronares.blogspot.com.es/

[49] Aprendizaje Supervisado. (Diciembre 2016). Retrieved from https://inteligenciaartificial101.wordpress.com/2015/10/20/aprendizaje-supervisado/

[50] Ejemplos de páginas web bien y mal diseñadas. (Noviembre 2016). Retrieved from http://instintobinario.com/ejemplo-de-paginas-web-bien-y-mal-disenadas/

[51] Consejos para crear una web. (Noviembre 2016). Retrieved from http://www.samueldiosdado.com/05/15-consejos-para-crear-una-web-user-friendly/

[52] Django project. (Octubre 2016). Retrieved from https://www.djangoproject.com/

[53] Antonio Melé, Django by example, Packt Publishing Ltd, Birmingham, 2015

[54] Conectar Mysql con Python & Django.(Octubre 2016).Retrieved from http://blog.johnserrano.co/mysql-con-python-django/

[55] Funcion Exec(). (Noviembre 2016). Retrieved from http://php.net/manual/es/function.exec.php

[56] 4 Best Chart Generation options with PHP componentes. (Noviembre 2016). Retrieved from https://www.sitepoint.com/4-best-chart-generation-options-php-components/

[57] JpGraph. (Noviembre 2016). Retrieved from http://jpgraph.net/

[58] Wilipedia Json. (Noviembre 2016). Retrieved from https://es.wikipedia.org/wiki/JSON

[59] Introduccion al lenguaje java. (Noviembre 2016). Retrieved from http://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava1/

[60] Julian J. Faraway, Linear Models with R Second edition, CRC press Taylor & Francis Group, Florida (EEUU), 2015

[61] Escribiendo su primera aplicación en Django, parte 1. (Noviembre 2016). Retrieved from https://docs.djangoproject.com/es/1.10/intro/tutorial01/

[61] Beatriz Sobrino y col., SNPs in forensic genetics: a review on SNP typing methodologies, Forensic Science International 154 (2005) 181–194.

[62] CODIS FBI. (Diciembre 2016). Retrieved from https://www.fbi.gov/services/laboratory/biometric-analysis/codis/codis-and-ndis-fact-sheet

[63] Manual paquete R “shiny”. (Diciembre 2016). Retrieved from https://cran.r-project.org/web/packages/shiny/shiny.pdf

[64] ShinyR. (Diciembre 2016). Retrieved from https://shiny.rstudio.com/

Anexos

Anexo 1. UBUNTU 16.04 y LAMP [A1]

La realización del TFM se llevó a cabo en un entorno Debian GNU/Linux (Ubuntu 16.04 LTS). La elección de este SO se debe a los siguientes motivos (entre muchos otros).

- Aplicaciones fáciles de instalar a través del terminal con la opción $ SUDO APT-GET "PROGRAMA" o $ SUDO APT INSTALL "PROGRAMA
- Seguridad. Existen muy pocos virus para este SO, además las ejecuciones de las aplicaciones se hacen como usuario y es necesario la contraseña de administrador (root) para realizar instalaciones o dar permisos de escritura lectura.
- Rapidez de procesado. A diferencia de otros SO como Windows, Ubuntu consume menos recursos para ejecutarse.

Esta elección prima sobre trabajar en un SO Windows, los cuales requerirían de la instalación de una máquina virtual para cargar tanto Ubuntu como el resto de aplicaciones usadas. Otro problema sería que tanto los recursos como el espacio en el disco duro estarían siendo compartidos con Windows, provocando una ralentización del trabajo.

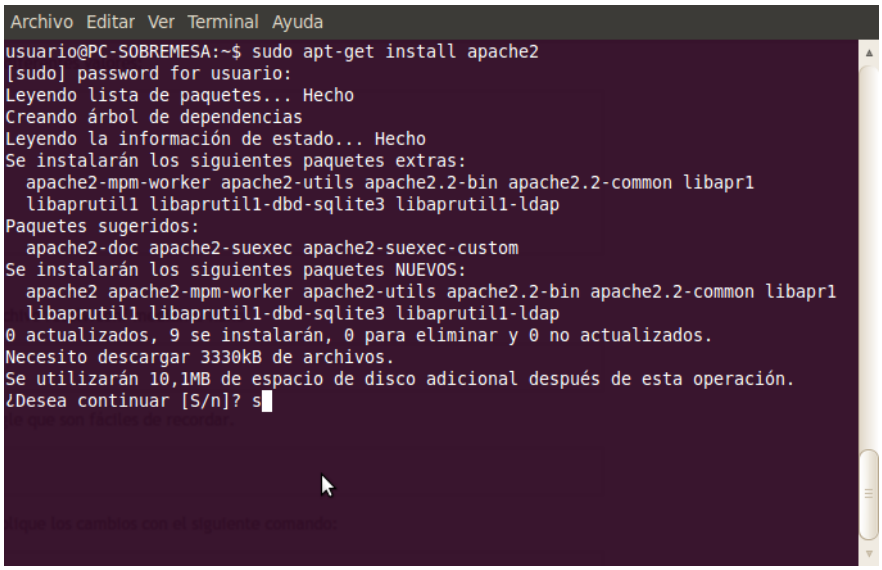
La descarga se realizó desde [23] y se instaló en una partición del disco duro de forma fácil e intuitiva (Tabla/Figura 70).

Tabla/Figura 70 Instalación Ubuntu

Tras la instalación y actualización del SO con los últimos drivers se procedió a la instalación del servidor servidor web sobre el que correrá el TFM. LAMP hace referencia a los diferentes programas necesarios para poner en marcha un servidor web en modo local.

Los pasos necesarios para la instalación de LAMP y su puesta a punto en marcha se realizaron con los siguientes comandos a través de la terminal y como superusuario.

apt-get update
apt-get upgrade -> Actualización del SO
apt-get install apache2 –> Instalacion de Apache2 (Tabla/Figura 71)

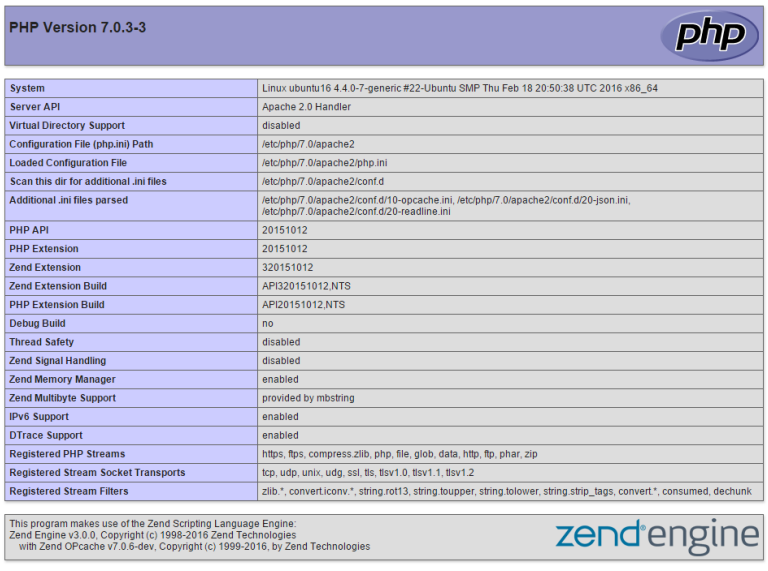


Tabla/Figura 71 Instalación Apache2 Ubuntu

sudo apt-get install mysql-server php-mysql -> Instalación de MySQL
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql -> Instalación de PHP. Se comprobará la correcta instalación creando un archivo "info.php" en la carpeta /var/www/html/info.php.

El archivo "info.php" tendrá el siguiente código <?php phpinfo(); ?>

De modo que al acceder a la siguiente dirección (http://localhost/info.php)a través del web browser obtendremos un resumen del servidor (Tabla/Figura 72)



PHP Version 7.0.3-3

System	Linux ubuntu16 4.4.0-7-generic #22-Ubuntu SMP Thu Feb 18 20:50:38 UTC 2016 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies

Tabla/Figura 72 Comprobación de la instalación del servidor web

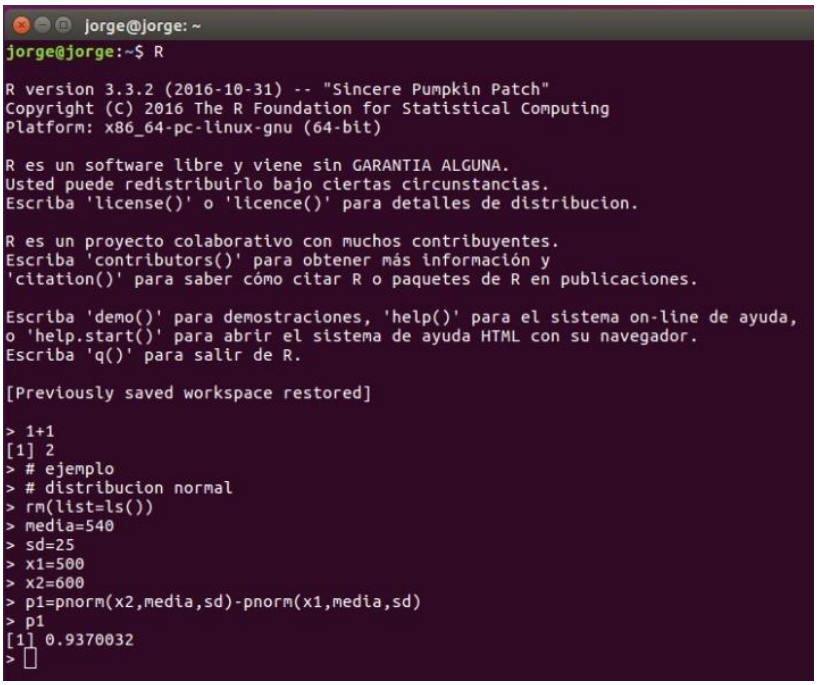
service apache2 restart -> Reiniciar el servidor Apache tras la instalación.
sudo apt-get update -> actualizar Ubuntu.

Anexo 2. R Statistical computing and graphics [A2]

La Instalación del software R y su interface Rstudio se realizó de la siguiente manera siguiendo el tutorial [24].

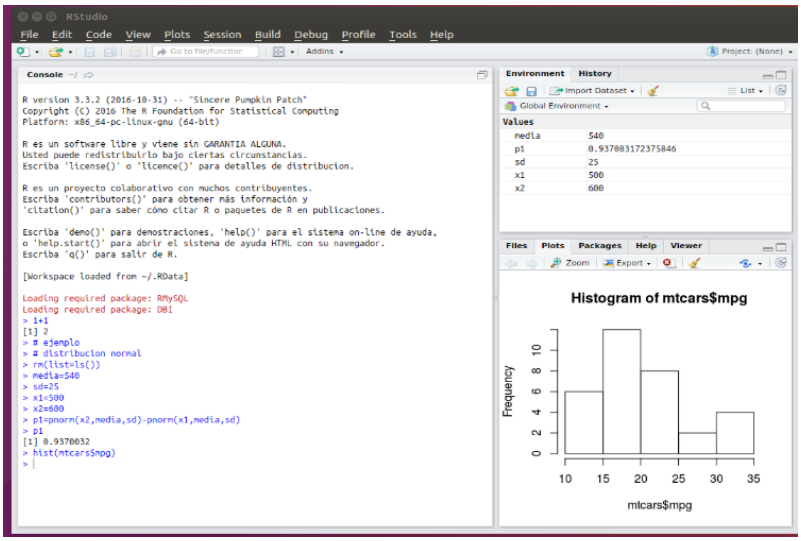
sudo echo "deb http://cran.rstudio.com/bin/linux/ubuntu xenial/" | sudo tee -a /etc/apt/sources.list
gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9
gpg -a --export E084DAB9 | sudo apt-key add -
sudo apt-get update
sudo apt-get install r-base r-base-dev
sudo apt-get install gdebi-core
wget https://download1.rstudio.org/rstudio-1.0.44-amd64.deb
sudo gdebi -n rstudio-1.0.44-amd64.deb
rm rstudio-1.0.44-amd64.deb
sudo apt-get update

Tras la instalación el programa Rstudio se encontrará en el lanzador de nuestro SO. La utilización de R puede realizarse a través del terminal (Tabla/Figura 73) de Linux escribiendo en el mismo como se aprecia en la parte superior de la siguiente imagen.



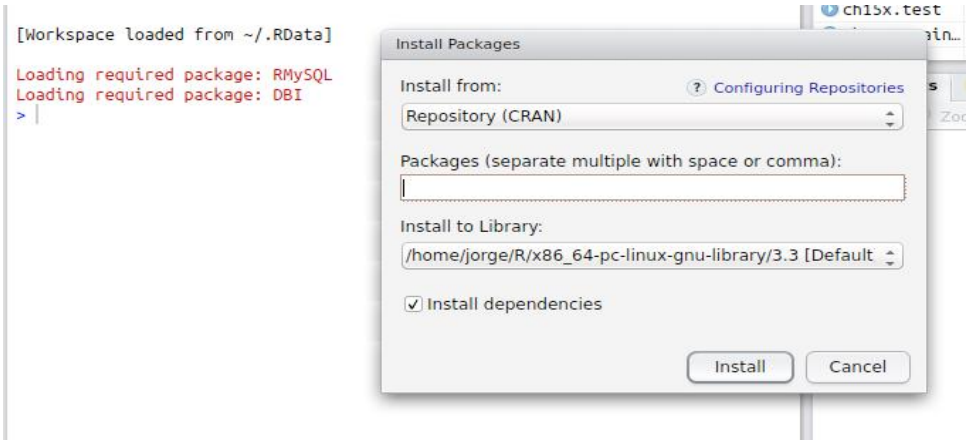
Tabla/Figura 73 Calculo de probabilidad normal en R terminal

Sin embargo, la realización del código se basó en el programa Rstudio que corre bajo el mismo motor que R, la diferencia frente al terminal de R es que la interface es más atractiva para el usuario (Tabla/Figura 74), además presenta ayuda de escritura y depuración de los errores cometidos durante la escritura.



Tabla/Figura 74 Calculo de probabilidad normal en Rstudio

Otro motivo para el uso de Rstudio es la facilidad con la que se pueden instalar los paquetes de datos necesarios para realizar los diferentes procesos estadísticos o de ML (Tabla/Figura 75).



Tabla/Figura 75 Instalación de paquetes de datos en Rstudio

Relación de librerías instaladas para el TFM:

- Tools: Contiene herramientas para trabajar con los diferentes paquetes de R [33].
- Fpc: Paquete para realizar diferentes métodos de clustering [34].
- Cluster: Paquete para el análisis de clústeres [35].
- HSAUR: Paquete con diferentes data sets [36].
- DBI: Paquete para relacionar R con bases de datos [37].
- RMySQL: Paquete para que mejora la conexión de DBI con una base de datos SQL [38].
- Ggplot2: Paquete para la realización de gráficos [39].
- Caret: Paquete para la clasificación y entrenamiento de los datos [40].
- E1071: Herramientas para el análisis de clases [41].
- Lattice: Visualización de datos con múltiples variables [42].
- Lfda: Herramientas para el análisis discriminante de Fisher [43].
- Ggfortify: Herramienta para la visualización de clústeres y PCA [44].

- Shiny: programación web en lenguaje R [63].

Anexo 3. Bitbucket GIT [A3]

El upload de los archivos generados durante la ejecución del TFM se realizaron siguiendo el tutorial de bitbucket [28]. El proceso será el siguiente:

1. Introducir los archivos a subir en la carpeta correspondiente.
2. Con el terminal de Linux acceder a esa carpeta.
3. Escribir el comando git add "nombre_archivo"
4. Git config –global user.email "correo_logging_bitbucket"
5. Commit -m "Initial commit"
6. Comprobar en [22] que los archivos han sido satisfactoriamente guardados en la carpeta "source"

Anexo 4. Tablas MySQL [A4]

En esta sección irán los códigos para la creación de las diferentes BBDD usadas y las tablas del proyecto HapMap. En esta sección se mostrará únicamente una fracción del código por motivos de lectura.

Desde el terminal de Linux escribir la dirección (path) hacia donde se encuentren los datos que hay que cargar en la BBDD, tras esto accederemos a la BBDD con el siguiente comando y validándolo con la contraseña estipulada para MySQL.

>mysql -u root -p –local-infile

>create database "nombre BBDD" {familia, HapMap, chrom15}

>use "nombre BBDD"

#Para la BBDD familia:

```
>CREATE TABLE `familia` (
`Num_ID` varchar(255) NOT NULL,
`Catalog_ID` varchar(255) NOT NULL,
`Perfil_genetico` varchar(255) NOT NULL,
`Sexo` varchar(255) NOT NULL,
`Familia` varchar(255) NOT NULL,
`Relacion_Fam` varchar(255) NOT NULL,
`Gen` varchar(255) NOT NULL,
`Mutacion` varchar(255) NOT NULL,
KEY ` Catalog_ID ` (`Catalog_ID `),
KEY ` Num_ID ` (`Num_ID `)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Para la BBDD HapMap

```
>CREATE TABLE `HapMap` (
`ref` varchar(255) NOT NULL,
`SNPalleles` varchar(255) NOT NULL,
`chrom` varchar(255) NOT NULL,
```

```
`pos` int(10) unsigned NOT NULL,
`strand` char(1) NOT NULL,
`assembly` varchar(255) NOT NULL,
`center` varchar(255) NOT NULL,
`protLSID` varchar(255) NOT NULL,
`assayLSID` varchar(255) NOT NULL,
`panelLSID` varchar(255) NOT NULL,
`QCcode` varchar(255) NOT NULL,
` ID001` varchar(255) NOT NULL,
` ID002` varchar(255) NOT NULL,
` ID003` varchar(255) NOT NULL,
` ID004` varchar(255) NOT NULL,
` ID005` varchar(255) NOT NULL,
……

` ID209` varchar(255) NOT NULL,
KEY `chrom` (`chrom`),
KEY `SNPalleles` (`SNPalleles`),
KEY `ref` (`ref`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Para la BBDD Chrom15

>CREATE TABLE `Chrom15` (
` Catalog_ID ` varchar(255) NOT NULL,
` rs1000040` varchar(255) NOT NULL,
` rs100018` varchar(255) NOT NULL,
` rs1000221` varchar(255) NOT NULL,
` rs1000281` varchar(255) NOT NULL,
……

` Poblacion` varchar(255) NOT NULL,
KEY ` Catalog_ID ` (`Catalog_ID `)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

# Para la BBDD Iris

>CREATE TABLE `iris` (
`SepalLength` float (3,1) NOT NULL,
`SepalWidth` float (3,1) NOT NULL,
`PetalLength` float (3,1) NOT NULL,
`PetalWidth` float (3,1) NOT NULL,
`Class` varchar(255) NOT NULL);
```

Anexo 5. Código para la comprobación de conexión entre PHP y MySQL [A5]

El código aquí mostrado para su ejecución debe incluirse dentro de un archivo con extensión php y lanzado a través del explorador en modo local. También se es necesario cumplimentar los datos de usuario, password, para que se realice la conexión, en caso contrario no se realizara la conexión y únicamente mostrara el mensaje de error de conexión.

```
1  <body>
2  <html>
3  <?php
4  /* incluimos los datos de conexión al servidor MySQL */
5  include("conexion.php");
6  /* Progrmación por procesos */
7  if($c=mysqli_connect ($cfg_servidor,$cfg_usuario,$cfg_password)){
8  print "<br>La conexión con  el servidor de bases de datos mediante procesos se ha realizado con exito<br>";
9  }else{
10 print "<br>No ha podido realizarse la conexión mediante procesos<br>";
11 }
12 if(mysqli_close($c)){
13 print "<br>Se ha cerrado la conexión, mediante procesos, con el servidor de bases de datos<BR>";
14 }
15 ?>
16 </html>
17 </body>
```

Anexo 6. Código Python para la representación gráfica y ML de la BBDD iris [A6]

```
1  %matplotbit inline
2  import MySQLdb
3  from mysql.connector import (connection)
4  import pandas as pd
5  import numpy as np
6  import matplotbit-pyplot as plt
7  from mpl_toolkits.mplot3d import Axes3D
8  from sklearn import cluster,datasets
9  conn=MySQLdb.connect (host="localhost", user="root", passwd="pulido84", db="iris")
10 cursor=conn.cursor()
11 cursor.execute(select * from iris)
12 rows=cursor.fetchall()
13 daf=pd.DataFrame ([[ij for ij in ij] for i in rows])
14                                                    daf.rename(columns={0:"Sepal_Length", 1:"Sepal_Width",2:"Pepal_Length",3:"Pepal_Width",4:"Species"},inplace=TRUE);
15 print daf.head(2)
16 kmeans = KMeans(n_clusters=3)
17 kmeans.fit(daf)
18 figu=plt.figure(1,figsize=(8,6))
19 ax=Axes3D(figu,elev=-150,azim=110)
20 ax.scatter(daf[:,[0]],daf[:,[1]],daf[:,[2]],c=kmeans.labels_)
21 ax.set_title("taxonomian de las 150 muestras utilizando k means")
22 ax.set_xlabel("longitud del sepalo")
23 ax.w_xaxis.set_ticklabels([])
24 ax.set_ylabel("ancho del sepalo")
25 ax.w_yaxis.set_ticklabels([])
26 ax.set_zlabel("longitud del petalo")
27 ax.w_zaxis.set_ticklabels([])
```