



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE TIANGUISTENCO

DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TESIS PARA OBTENER EL GRADO DE:
INGENIERO EN SISTEMAS COMPUTACIONALES**

**“Desarrollo de una herramienta
de visualización Integral de datos omicos”**

PRESENTA:

JUAN JOSÉ MARTÍNEZ ULLOA

ASESOR:

**DR. JESÚS ESPINAL ENRÍQUEZ
MTRA. KAROL BACA LOPEZ**

TIANGUISTENCO ESTADO DE MÉXICO, MARZO 2018

Resumen.

Abstract.

Agradecimientos.

Índice

1. Introducción	2
1.1. Planteamiento del Problema	2
1.2. Justificación	3
1.3. Delimitación	3
1.4. Hipótesis	3
1.5. Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos	4
1.6. Aportaciones de la tesis	4
2. Estado del Arte	4
3. Tecnologías genómica.	4
4. Microarreglos	4
4.1. Formato	6
4.2. Plataformas	7
4.3. Metilación	7
5. Propuesta de Solución	8
6. Metodología	8
6.1. Diseño de estudio	8
6.2. Circos plot	8
6.3. Galaxy	10

1. Introducción

1.1. Planteamiento del Problema

El cáncer de mama es una enfermedad compleja y heterogénea con más de 1,300,000 casos y 450,000 muertes cada año en todo el mundo. Esta enfermedad se caracteriza por diferentes áptectos biológicos como desregulización de la expresión génica, alteraciones genómicas del ADN, etc. Todo esto da lugar al inicio y desarrollo del carcinoma de mama. En éstos últimos años el uso de datos ómicos, como los basados en microarreglos (microarrays) y secuenciación, esta en su pleno () en el campo de la biomedicina. Todos estos datos permiten estudiar enfermedades desde un punto de vista biomolecular. Con esto, se ofrecen grandes oportunidades para mejorar tanto la comprensión de la enfermedad, como el desarrollo de nuevos métodos para el diagnóstico y tratamiento del paciente, sin embargo, el análisis de estos datos

producidos por estas tecnologías, es bastante complejo por lo que es necesario la aplicación de avanzadas técnicas de análisis y cálculos computacionales que permiten obtener la información biológica disponibles. Hasta el día de hoy todos estos datos óptenidos de diversos experimentos, datos de muy alta calidad, datos clínicamente bien anotados, y una gran cantidad de datos de canceres analizados con el fin de encontrar anomalías recurrentes que sean importantes de la enfermedad y estos datos se guardan en diversas plataformas que permiten tener una gran cantidad de información, pero estos datos no son tan fáciles de analizar, por ello, la bioinformática ayuda a manejar estructurar y organizarla para que sea más fácil de comprender. Circos plot es una de las herramientas que existen para la visualización de datos, ideal para explorar las relaciones entre objetos y posiciones. Esta herramienta es flexible, aunque originalmente fue diseñada para visualizar datos genómicos, se puede crear figuras a partir de datos en cualquier campo, desde la genómica hasta la visualización de la migración al arte matemático. Esta herramienta puede ser automatizada. Está controlada por archivos de configuración de texto plano, lo que lo hace difícil su incorporación en *pipeline* de adquisición de datos, análisis e informes. Todo esto hace difícil su utilización en instalación y desarrollo de la visualización.

1.2. Justificación

La cantidad de información generada a través de las tecnologías de secuenciación masiva son enormes. La integración de toda esa información puede ayudar a tomar decisiones de índole biomédica e incluso clínica. Aunque existen plataformas ya conocidas capaces de presentar esa información de un modo amigable, la programación de dichas herramientas continúa siendo complicada para un usuario final. Es por esto, que generar una herramienta con gran capacidad de visualización, integración de información y facilidad de implementación para un usuario final es de la mayor importancia.

Con el fin de mejorar el rendimiento y la cantidad de tiempo de los investigadores del Instituto Nacional de Medicina Genómica, es fundamental sistematizar este software para reducir el tiempo de programación y la investigación. La sistematización de circos plot, brindará la posibilidad de que el investigador ahorre en tiempo de programación o en dado caso que no se conozca nada del uso nativo de circos plot, leer todo el manual de uso de dicho software, para que ocupe su mayor tiempo en la investigación y solo tome varios minutos para diseñar su gráfica circular llamada circos plot.

1.3. Delimitación

El desarrollo de tecnologías de secuenciación masiva es abrumador. Para poder integrar toda la información proveniente de dichas tecnologías es necesario contar con herramientas adecuadas para la obtención, análisis y visualización de las tecnologías anteriormente mencionadas.

Ante esta problemática, resulta altamente relevante generar una herramienta capaz de condensar en un solo golpe de vista varias capas de información, para que, de este modo, se pueda analizar con un mayor detalle los datos obtenidos.

Circos plot es una herramienta de apoyo visual que permite lograr la visualización a nivel de genoma completo varios tipos diferentes de datos: Expresión, mutaciones, metilación, citobandas, etc., de una manera amigable para el usuario final.

Aunque circos plot tiene grandes ventajas en cuanto a la visualización, su programación es altamente complicada, por lo que generar un back-end que integre de modo sencillo varios tipos de datos, será de la mayor utilidad.

1.4. Hipótesis

El desarrollo de una herramienta de Visualización integral permitirá el análisis de la información de los datos derivados de múltiples plataformas.

1.5. Objetivos

1.5.1. Objetivo General

Implemetar una herramienta de visualización que integre datos genómicos derivados de múltiples plataformas.

1.5.2. Objetivos Específicos

- Realizar el pretatamiento de los datos de transcrito (Microarreglos).
- Realizar el pretatamiento de los datos de Metilación.
- Desarrollar algoritmo computacional para hacer la búsqueda en la referencia del genoma humano con los datos de transcrito y metilación.
- Desarrollar software para preprocesar datos de distintos tipos de tecnologías genómicas.
- Desarrollar software para integrar al backbone de circos plot varias capas de información proveniente del paso anterior.
- Generar opciones de visualización de fácil manejo para el usuario final.

1.6. Aportaciones de la tesis

La presente Tesis aportara

2. Estado del Arte

3. Tecnologías genómica.

Las tecnologías genómicas es el conjunto de herramientas orientadas al estudio integral del funcionamiento, contenido, evolución del genoma. Es una de las áreas más vanguardistas de la biología. La genómica usa conocimientos derivados de distintas ciencias como la biología molecular, la bioquímica, la informática, la estadística, las matemáticas y la física. Para entender un poco más de estas tecnologías y de los datos que se obtiene de las antes mencionadas, hablaremos de las tecnologías genómicas que son: Microarreglos y Metilación.

4. Microarreglos

Un microarreglo de ADN (ácido desoxirribonucleico) también llamado DNA chip, también llamado **oligonucleotido de DNA chip o gene chip** que consiste en pequeños fragmentos de ADN de los cuales representa un gen diferente [1], es una superficie sólida a la cual se une una colección de fragmentos de

ADN. Las superficies empleadas para fijar el ADN son muy variables y pueden ser de vidrio, plástico e incluso de silicona. Los chips de ADN se usan para analizar la expresión diferencial de genes. Su funcionamiento consiste, básicamente, en medir el nivel de hibridación entre la sonda específica (probe, en inglés), y la molécula diana (target), y se indican generalmente mediante fluorescencia y a través de un análisis de imagen, lo cual indica el nivel de expresión del gen.

Por lo tanto, los microarreglos son una potente fuente de obtención de perfiles de expresión de genes sometidos a diferentes condiciones. Identificar los patrones de los niveles de expresión será muy útil para compararlos y poder estudiar las respuestas de los genes.

Aplicando una serie de procesos experimentales y computacionales sobre los microarreglos se obtiene una matriz numérica bidimensional que consta de los genes de poblaciones distintas como individuos y de las condiciones experimentales a las que expusieron las células como variables en el caso que se quiera estudiar a los genes, o a la inversa, si es que se quiere realizar un estudio comparativo de las condiciones a que se somete. Cada uno de los valores de la matriz representa el nivel de expresión de un determinado gen bajo una cierta condición experimental.

Estas matrices son de grandes dimensiones puesto que existen una gran cantidad de condiciones experimentales y genes. En la figura 1 se puede observar el modelo de la presentación de los microarreglos. Cada fila representa un gen, el cual debe de ser indicado, cada columna representa una condición experimental, cuyo nombre también debe de ser identificado. Los valores de la matriz son los niveles de expresión de los genes para la condición experimental.

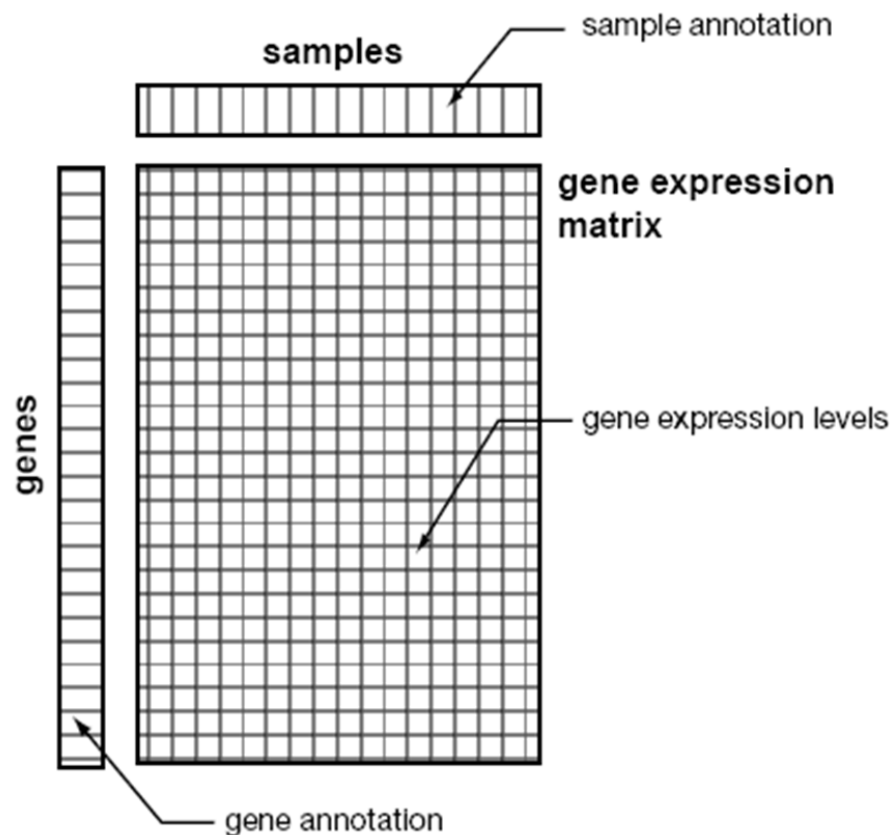


Figura 1: Chip de un Microarreglo

Dado que realizar un análisis de estas matrices de grandes dimensiones es una tarea prácticamente imposible, se hace necesarias técnicas computacionales que permitan analizar todos estos datos y entonces realizar el análisis biológico.

Actualmente existen diferentes bases de datos a nuestro alcance a través de Internet que unifican y facilitan toda esta información genética además de ofrecer diversas herramientas para el análisis de esta gran cantidad de información. Algunas de estas bases de datos por ejemplo son las que hay en el EMBL (European Molecular Biology Laboratory), el SIB (Swiss Institute of Bioinformatics), el EBI (European Bioinformatics Institute) o el NCBI (National Center for Biotechnology Information). El EBI y el NCBI son los que más información contienen y por lo tanto los más utilizados. El tamaño de los microarreglos es de 1.28 cm x 1.28 cm, hay 500,000 ubicaciones en cada matriz y por lo general tiene millones de cadenas de ADN construidas en cada ubicación, cada cadena contiene 25 pares de bases (Figura 1).

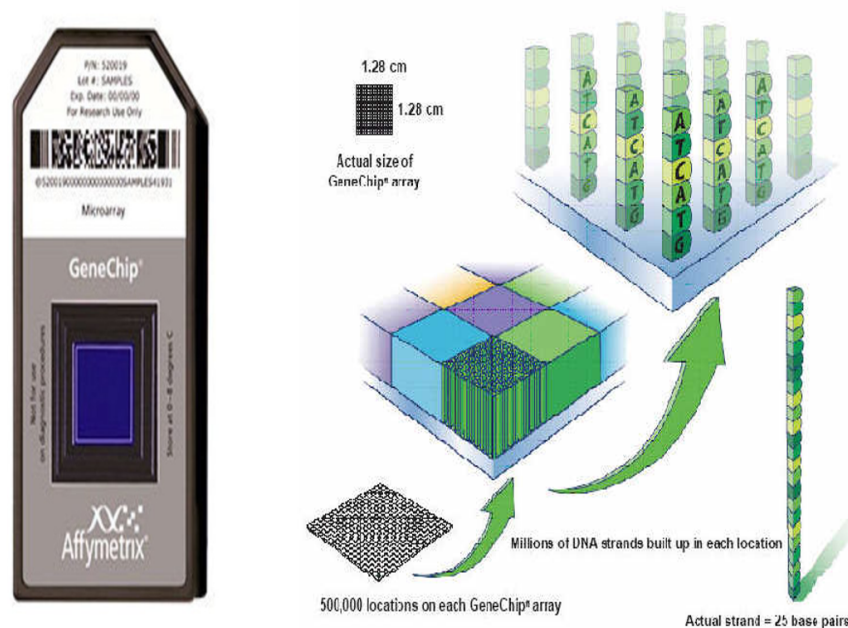


Figura 2: Chip de un Microarreglo.

En estos chips se imprimen las secuencias biológicas en un chip, de manera que se puede cuantificar la transcripción en una matriz numérica.

4.1. Formato

Los archivos están disponibles en un formato de valores separados por comas (CSV). Estos son archivos de texto sin formato con cada fila terminada por un carácter de nueva línea. Los datos en campos separados están entre comillas y separados por comas. Ninguno de los campos de datos contiene ninguno de estos caracteres: comillas, nueva línea, retorno de carro o tabulación.

Estos archivos se usan principalmente en aplicaciones de hojas de cálculo y programas de bases de datos (como bases de datos SQL). Los datos están formateados de tal manera que estos dos usos sean relativamente fáciles. Se tiene en cuenta que algunos de los archivos y los campos de datos en ellos son grandes.

La primera fila de cada archivo contiene los títulos de los campos que figuran en las filas siguientes.

Cada fila después de la primera fila contiene anotaciones para un solo conjunto de sondas. Todas las anotaciones para ese conjunto de sonda están contenidas en esa única fila. En algunos campos, como las anotaciones de dominio de proteínas, puede haber más de una anotación para un único conjunto de sondas. En este caso, los valores múltiples están separados por la cadena '///'.

En muchos tipos de anotaciones, los subcampos están separados por '//'. Por ejemplo, una anotación para un "GO Biological Process" puede aparecer como "7155 // cell adhesion // predicted / computed". En este caso, las secciones corresponden a 'ID // Descripción // Evidencia', pero el significado de los subcampos varía entre los diferentes tipos de anotación, como se describe a continuación.

Los campos vacíos se indican con '- - -'. El hecho de utilizar una cadena de este tipo en lugar de dejar el campo vacío es que hace que la naturaleza columnar de los datos sea más visible en ciertos programas de hoja de cálculo. Algunas columnas en algunos archivos no contienen datos. Para ayudar a los usuarios a combinar datos de varios archivos, dichas columnas vacías no se eliminan. Por lo tanto, cada archivo tiene las mismas columnas en el mismo orden.

Algunos campos, como 'Chip', contienen el mismo valor para cada conjunto de sonda en un archivo. Aunque estos datos son redundantes en cualquier archivo individual, son útiles para los usuarios que combinan datos de varios archivos.

4.2. Plataformas

Actualmente existen diferentes bases de datos a nuestro alcance a través de Internet que unifican y facilitan toda esta información genética además de ofrecer diversas herramientas para el análisis de esta gran cantidad de información. Algunas de estas bases de datos por ejemplo son las que hay en el EMBL (European Molecular Biology Laboratory), el SIB (Swiss Institute of Bioinformatics), el EBI (European Bioinformatics Institute) o el NCBI (National Center for Biotechnology Information). El EBI y el NCBI son los que más información contienen y por lo tanto los más utilizados.

Algunas de las plataformas donde se aloja los datos de microarreglos son varias y una de las más conocidas en la comunidad científica es TCGA (The Cancer Genome Atlas). TCGA es una colaboración entre el Instituto Nacional del Cáncer (NCI) y el Instituto Nacional de Investigación del Genoma Humano (NHGRI) que ha generado mapas completos y multidimensionales de los principales cambios genómicos en 33 tipos de cáncer. El conjunto de datos TCGA, que comprende más de dos petabytes de datos genómicos, se ha hecho públicamente disponible, y esta información genómica ayuda a la comunidad de investigación del cáncer a mejorar la prevención, el diagnóstico y el tratamiento del cáncer.

Algunas de las plataformas son:

- Illumina
- Affymetrix SNP 6.0

4.3. Metilación

La metilación es un proceso epigenético que participa en la regulación de la expresión genética de dos maneras, directamente al impedir la unión de factores de transcripción, e indirectamente proporcionando la estructura cerrada de la cromatina[?, ?, ?, ?, ?, ?].

5. Propuesta de Solución

La solución ante la problemática planteada, consiste fundamentalmente en incorporar a circos plot a un interfaz de usuario amigable para el usuario final agregando la mayoría de las funciones del software para su buen funcionamiento, implementando de manera subyacente un programa escrito en python para el mapeo de los datos con el diccionario de referencia del genoma humano para obtener el formato que se requiere para correr el software. Así mismo realizar incorporar circos plot a Galaxy que una plataforma que contiene herramientas para análisis de datos para investigación científica y de esta manera implementar circos plot en el laboratorio de genómica computacional en el Instituto Nacional de Medicina Genómica

6. Metodología

6.1. Diseño de estudio

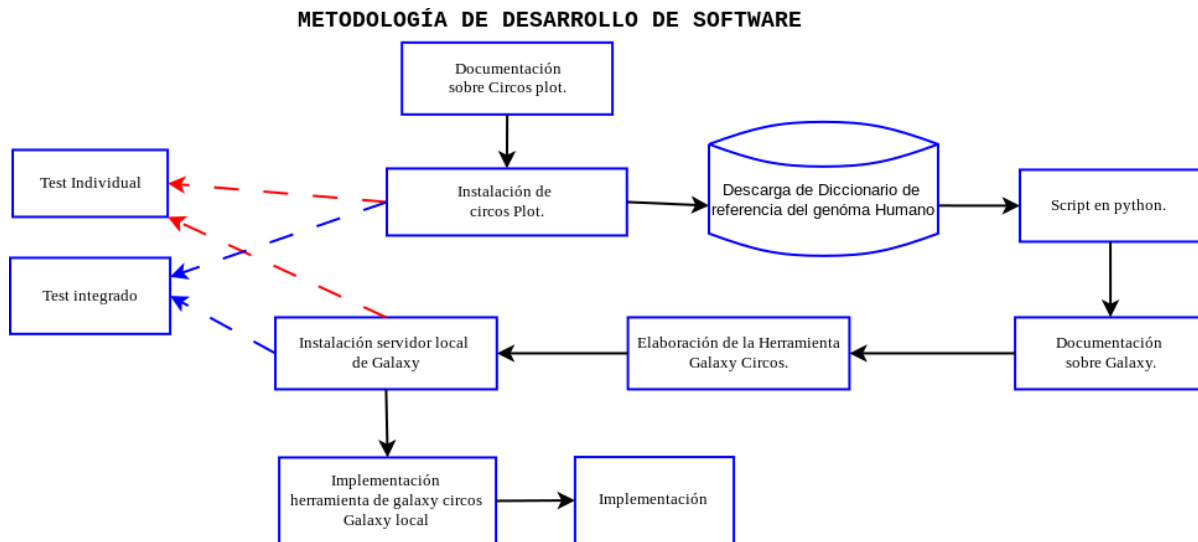


Figura 3: Metodología

6.2. Circos plot

6.2.1. ¿Qué es Circos plot?

Circos fue diseñado inicialmente para mostrar datos genómicos (particularmente genómica del cáncer y genómica comparativa) y biología molecular. Tiene características específicas que abordan los retos típicos en el dibujo de este tipo de datos, que tienden a ser muy escasa y abarcan un gran número de escalas de longitud.

Circos fue concebido originalmente para visualizar datos genómicos tales como alineaciones y variación estructural. Con el tiempo, se agregó soporte para pistas de datos 2D, tales como trazados de línea, dispersión, mapa de calor e histograma.

A medida que crecía la popularidad de Circos, provocada por una infografía de página completa del New York Times , comenzó a utilizarse para visualizar otros datos, no sólo genómicos.

Es una herramienta de visualización para facilitar la identificación y análisis de similitudes y diferencias que surgen de las comparaciones de genomas. La herramienta es eficaz en la visualización de la variación en la estructura del genoma y, en general, cualquier otro tipo de relaciones de posición entre los intervalos genómicos. Tales datos se producen rutinariamente mediante alineaciones de secuencias, matrices de hibridación, mapeo del genoma y estudios de genotipado. Circos utiliza un diseño de ideograma circular para facilitar la visualización de relaciones entre pares de posiciones mediante el uso de cintas, que codifican la posición, el tamaño y la orientación de los elementos genómicos relacionados. Circos es capaz de mostrar datos como diagramas de dispersión, líneas y histogramas, mapas de calor, mosaicos, conectores y texto. Las imágenes de mapa de bits o vectoriales pueden crearse a partir de entradas de datos estilo GFF y archivos de configuración jerárquica, que pueden generarse fácilmente mediante herramientas automatizadas, lo que hace que Circos sea adecuado para un despliegue rápido en el análisis de datos y en las tuberías de generación de informes.

Circos ha aparecido en muchas publicaciones, tanto científicas como generales. Ha cambiado la forma en que la comunidad científica visualiza alteraciones genómicas (cambios en un genoma en el tiempo, o diferencias entre dos o más genomas). Una aplicación oportuna de este enfoque es la creación de cifras eficaces que muestran cómo los genomas del cáncer difieren de los sanos (por ejemplo, COSMIC: Censo de las mutaciones somáticas en el cáncer).

La comunidad científica biológica ha adoptado a Circos de todo corazón. Por ahora, Circos ha aparecido en las portadas de las publicaciones de Nature y Science , que son las principales revistas científicas del mundo como por ejemplo Bioinformatics, GenomeBiology, nature, Science, Nucleic Acids Research, American Scientist, Genome Research, PortFolio, Wired, PNAS, PLOS, entre otras.

Circos plot es una idea original de **Martin Krzywinski** y de un grupo de trabajo tras de él. El artículo donde habla sobre esta nueva herramienta de visualización (*Circos: an information aesthetic for comparative genomics*) fue publicado el 9 de septiembre del 2009 y actualmente tiene 2973 citas, esto quiere decir que es una herramienta verdaderamente utilizada no solo en la biología y todas sus ramas si no en muchos otros campos de la ciencia.

Instalación de circos Primero, descargamos Circos <http://circos.ca/software/download>. El contenido de la distribución se describe a continuación. No necesitamos mover o editar ningún archivo en la distribución principal.

Listing 1: Suponiendo que desea instalar en ROOT= /software/circos

```
1 $ cd ~
2 $ mkdir software
3 $ mkdir software/circos
4 $ cd software/circos
5 # Descargar Circos y colocarlo en el directorio software/circos
6 $ wget http://circos.ca/distribution/circos-0.69-6.tgz
7 # Descargar la versión mas actual (Recomendación) .
8 #Descomprime
9 $ tar xvfz circos-0.69-6.tgz
10 ...
11 circos-0.69-6/data/karyotype/karyotype.arabidopsis.txt
12 circos-0.69-6/data/karyotype/karyotype.zeamays.txt
13 circos-0.69-6/data/karyotype/karyotype.oryzasativa.txt
```

```

14 # Crea un enlace simbolico a current
15 $ ln -s circos-0.69-6 current
16 # Comprobamos si se creo el enlace simbolico.
17 $ ls -lh
18 drwxrwxr-x 9 juanjo juanjo 4.0K nov 29 10:36 circos-0.69-6/
19 -rw-rw-r-- 1 juanjo juanjo 22M nov 29 10:36 circos-0.69-6.tgz
20 lrwxrwxrwx 1 juanjo juanjo 13 nov 29 10:30 current -> circos-0.69-6/
21 # Borramos el archivo tgz, si usted quiere

```

Para instalar los módulos GD y Perl en Ubuntu, usamos apt-get.

```

1 $ sudo apt-get -y install libgd2-xpm-dev

```

CORRIENDO CIRCOS

Circos utiliza banderas de línea de comandos, que son obligatorias. Por lo menos, debe especificar el archivo de configuración de imagen usando -conf.

Es una buena idea agregar el **bin/** directorio en la distribución para PATH que pueda ejecutar **bin/circos** desde cualquier lugar.

Añadimos al **root= /software/circos/current** como se describió anteriormente, añadimos esto a nuestro **./bashrc** **./bash-profile**.

```

1 $ export PATH=~/.software/circos/current/bin:$PATH

```

Ejecutamos explícitamente cualquiera **./bashrc** **./bash-profile** para que esto surta efecto

```

1 $ ./ .bashrc
2 # o
3 $ ./ .bash_profile

```

Finalmente, probamos que nuestro PATH ha sido modificado,

```

1 $ cd ~
2 $ echo $PATH
3 ~/.software/circos/current/bin: ...
4 $ which circos
5 ~/.software/circos/current/bin/circos

```

Revisando si faltan módulos Perl

Verificamos si tenemos algún módulo faltante

```

1 $ circos -modules
2 ok      1.36 Carp
3 ok      0.38 Clone
4 ok      2.63 Config::General
5 ok      3.56 Cwd
6 ok      2.158 Data::Dumper
7 ok      2.54 Digest::MD5
8 ok      2.85 File::Basename
9 ok      3.56 File::Spec::Functions
10 ok     0.2304 File::Temp
11 ok      1.51 FindBin
12 ok      0.39 Font::TTF::Font
13 ok      2.53 GD

```

```

14 ok      0.2 GD::Polyline
15 ok      2.45 Getopt::Long
16 ok      1.16 IO::File
17 ok      0.413 List::MoreUtils
18 ok      1.41 List::Util
19 ok      0.01 Math::Bezier
20 ok      1.9997 Math::BigFloat
21 ok      0.07 Math::Round
22 ok      0.08 Math::VecStat
23 ok      1.03 Memoize
24 ok      1.53_01 POSIX
25 ok      1.26 Params::Validate
26 ok      1.64 Pod::Usage
27 ok      2.05 Readonly
28 ok 2016060801 Regexp::Common
29 ok      2.64 SVG
30 ok      1.19 Set::IntSpan
31 ok      1.6611 Statistics::Basic
32 ok      2.53_01 Storable
33 ok      1.20 Sys::Hostname
34 ok      2.03 Text::Balanced
35 ok      0.60 Text::Format
36 ok      1.9726 Time::HiRes

```

Cuando tenemos estas cosas ya tenemos `circos plot` instalado en nuestro SO

6.3. Galaxy

Galaxy es una plataforma abierta basada en la web para la investigación biomédica computacional accesible, reproducible y transparente.

Accesible: los usuarios sin experiencia en programación pueden especificar fácilmente parámetros y ejecutar herramientas y flujos de trabajo.

Reproducible: Galaxy captura información para que cualquier usuario pueda repetir y comprender un análisis computacional completo.

Transparente: los usuarios comparten y publican análisis a través de la web y crean páginas, documentos interactivos y basados en la web que describen un análisis completo.

Para obtener instalado galaxy necesitamos seguir los pasos siguientes:

Requisitos

- UNIX / Linux o Mac OSX
- Python 2.7

Empezar

Para producción o usuario único

Clonar Galaxy desde GitHub

```

1 $ git clone -b release_17.09 https://github.com/galaxyproject/galaxy.git

```

Comenzarlo

Galaxy requiere algunas cosas para ejecutar: un `virtualenv`, archivos de configuración y módulos dependientes de Python. Sin embargo, iniciar el servidor por primera vez creará / adquirirá estas cosas según sea necesario. Para iniciar Galaxy, simplemente ejecute el siguiente comando en una ventana de terminal:

```

1 $ cd ~
2 $ cd /galaxy

```

```

3 # En contadas ocaciones se requiere dar permisos de lectura escritura y ejecución para ello ponmos en
  terminal:
4 $ chmod -R 777 run.sh
5 $ sh run.sh

```

```

./run.sh /home/juanjo/galaxy
+ ... /home/juanjo/galaxy
juanjo@JuanJo ~/galaxy> ./run.sh
Activating virtualenv at ./venv
Requirement already satisfied: pip==8.1 in ./venv/lib/python2.7/site-packages
Requirement already satisfied: bx-python==0.7.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 2))
Requirement already satisfied: MarkupSafe==0.23 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 3))
Requirement already satisfied: PyYAML==3.11 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 4))
Requirement already satisfied: SQLAlchemy==1.0.15 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 5))
Requirement already satisfied: mercurial==3.7.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 6))
Requirement already satisfied: numpy==1.9.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 7))
Requirement already satisfied: pycryptos==2.6.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 8))
Requirement already satisfied: uwsgi==2.0.15 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 9))
Requirement already satisfied: bz2file==0.98 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 16))
Requirement already satisfied: ipaddress==1.0.18 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 17))
Requirement already satisfied: Paste==2.0.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 19))
Requirement already satisfied: PasteDeploy==1.5.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 19))
Requirement already satisfied: docutils==0.12 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 20))
Requirement already satisfied: wchartype==0.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 21))
Requirement already satisfied: repoze.lru==0.6 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 22))
Requirement already satisfied: Routes==2.4.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 23))
Requirement already satisfied: WebOb==1.4.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 24))
Requirement already satisfied: WebHelpers==1.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 25))
Requirement already satisfied: Makos==1.0.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 26))
Requirement already satisfied: pytz==2015.4 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 27))
Requirement already satisfied: Babel==2.4.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 28))
Requirement already satisfied: Babel==2.4.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 28))
Requirement already satisfied: dictobj==0.3.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 30))
Requirement already satisfied: nose==1.3.7 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 31))
Requirement already satisfied: Parsley==1.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 32))
Requirement already satisfied: six==1.10.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 33))
Requirement already satisfied: Whoosh==2.7.4 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 34))
Requirement already satisfied: galaxy.sequence.util==1.0.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 35))
Requirement already satisfied: h5py==2.7.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 36))
Requirement already satisfied: pywallf==1.5.1 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 39))
Requirement already satisfied: python-dateutil==2.5.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 40))
Requirement already satisfied: docopt==0.6.2 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 41))
Requirement already satisfied: Cheetah==2.4.4 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 44))
Requirement already satisfied: Markdown==2.6.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 45))
Requirement already satisfied: bioblend==8.7.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 48))
Requirement already satisfied: boto==2.38.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 49))
Requirement already satisfied: requests==2.10.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 50))
Requirement already satisfied: requests-toolbelt==0.4.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 51))
Requirement already satisfied: kombu==3.0.39 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 54))
Requirement already satisfied: amqp==1.4.8 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 55))
Requirement already satisfied: anyjson==0.3.3 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 56))
Requirement already satisfied: psutil==4.1.0 in ./venv/lib/python2.7/site-packages (from -r requirements.txt (line 59))

```

Figura 4: Consola corriendo galaxy.

Esto iniciará el servidor Galaxy en el host local y el puerto 8080. Luego se puede acceder a Galaxy desde nuestro navegador web en **http://localhost:8080**. Después de comenzar, el servidor de Galaxy imprimirá la salida a la ventana del terminal. Para detener el servidor Galaxy, se puede presionar las Ctrl+C en la ventana de la terminal desde la que se está ejecutando Galaxy. **Próximos pasos** Convertirse en administrador

Para controlar Galaxy a través de la interfaz de usuario (instalación de herramientas, administración de usuarios, creación de grupos, etc.), los usuarios deben convertirse en administradores . Solo los usuarios registrados pueden convertirse en administradores. Para otorgar privilegios de administrador a un usuario, se completaron los siguientes pasos:

- En el directorio config/ viene el archivo de configuración pero lo encontramos como **galaxy.ini.sample** pero galaxy no lo reconoce así pero podemos tenerlo simplemente copiarlo como se muestra aquí:

```

1 $ cd /galaxy/config
2 $ cp galaxy.ini.sample galaxy.ini
3 $ vi galaxy.ini

```

- Agregamos el correo electrónico de inicio de sesión Galaxy del usuario al archivo de configuración config/galaxy.ini. Como se muestra aquí:

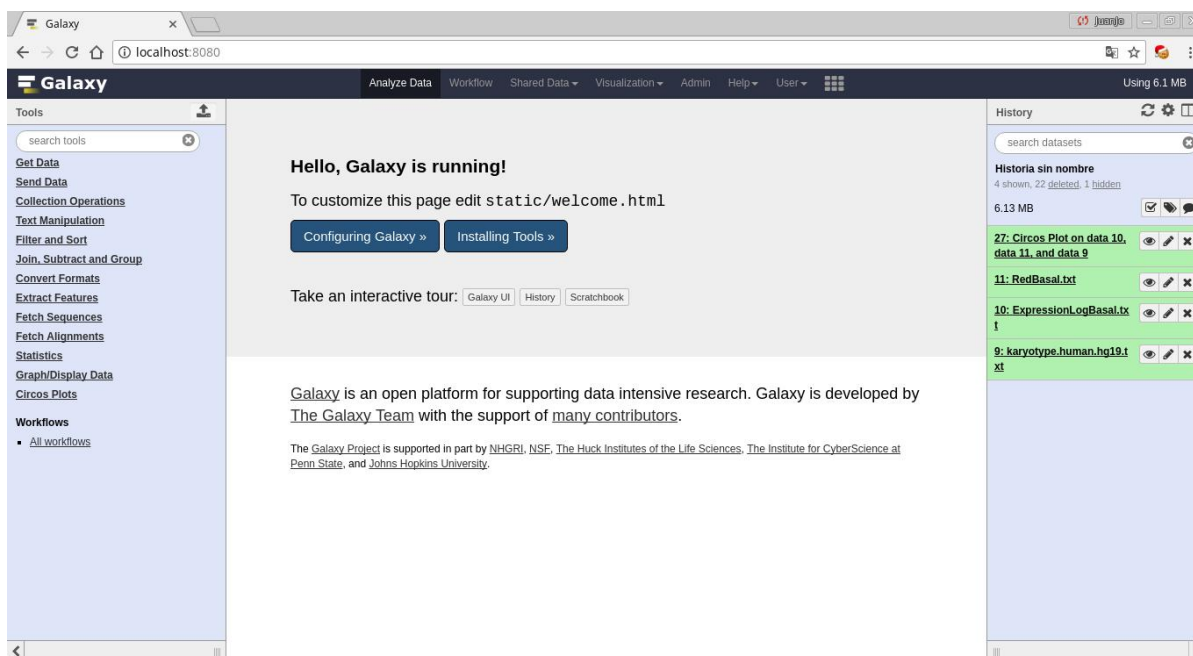


Figura 5: Pantalla principal de Galaxy.

```
1 # Esta linea viene comentada por lo que hay que descomentarlo y ponemos:
2 admin_users = jjmartinez@inmegen.edu.mx
```

- Reinicié Galaxy después de modificar el archivo de configuración para que los cambios surtan efecto.

Referencias

- [1] Microarreglos de ADN y sus aplicaciones en investigaciones biomédicas. Revista CENIC. Ciencias Biológicas, 38 (2), 132-135.
- [2] Metilación del ADN: marcador diagnóstico y pronóstico de cáncer. Gaceta médica de México, 142(1), 81-82.