

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Máster en Bioinformática y Biología Computacional

TRABAJO FIN DE MÁSTER

DESARROLLO DE UN WORKFLOW PARA EL ANÁLISIS GENÉTICO DE *CAMPYLOBACTER JEJUNI*

Autor: José Antonio Barbero Aparicio

Directores: José Francisco Díez Pastor y Beatriz Melero Gil

Ponente: Alberto Suárez González

Febrero 2019

DESARROLLO DE UN WORKFLOW PARA EL ANÁLISIS GENÉTICO DE *CAMPYLOBACTER JEJUNI*

Autor: José Antonio Barbero Aparicio
Directores: José Francisco Díez Pastor y Beatriz Melero Gil
Ponente: Alberto Suárez González

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero 2019

Resumen

Resumen

Palabras Clave

Abstract

Key words

Agradecimientos

Índice general

Índice de Figuras	IX
Índice de Tablas	X
1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos y enfoque	2
1.3. Metodología y plan de trabajo	2
1.3.1. Metodología	2
1.3.2. Plan de Trabajo	2
2. Análisis genético de <i>Campylobacter jejuni</i>. Estado del arte	7
2.1. Introducción	7
2.2. <i>Campylobacter jejuni</i>	7
2.3. Técnicas en bioinformática	7
2.3.1. Control de calidad	7
2.3.2. Ensamblado	7
2.3.3. Anotación	7
2.3.4. Análisis pangenómico	7
2.3.5. Workflows en bacterias / Workflows en general	7
2.4. Herramientas en informática	7
2.4.1. Virtualización	7
2.4.2. Plataforma de soporte del workflow	7
2.4.3. Herramientas GUI	7
3. Sistema, diseño y desarrollo	9
3.1. Estructura general del sistema	9
3.2. Desarrollo y configuración de Docker	9
3.3. Desarrollo y configuración del workflow en Galaxy	9
3.4. Desarrollo de la capa externa con Python	9
3.5. Otros aspectos del desarrollo	9

4. Experimentos Realizados y Resultados	11
4.1. Bases de datos y protocolo	11
4.2. Sistemas de referencia	11
4.3. Escenarios de pruebas	11
4.4. Experimentos del sistema completo	11
5. Conclusiones y trabajo futuro	13
Glosario de acrónimos	15
Bibliografía	16
A. Manual de utilización	19
B. Manual del programador	21

Índice de Figuras

Índice de Tablas

1.1. Plan de Trabajo	2
--------------------------------	---

1

Introducción

1.1. Motivación del proyecto

Campylobacter jejuni es una bacteria Gram negativa que, a pesar de tener unas condiciones complicadas de crecimiento [1], es la zoonosis bacteriana que produce un mayor número de intoxicaciones alimentarias en los países tanto desarrollados como en vías de desarrollo. Por ejemplo, en la EU en el año 2016 se declararon del orden de 250.000 casos comprobados [2]. El coste debido a la campilobacteriosis se estima en la EU en torno a 2,4 billones de euros anuales. La fuente de contaminación más habitual es el consumo de carne de pollo poco cocinada [3]. El grupo de investigación Tecnofood lleva varios años investigando sobre las fuentes de contaminación de este microorganismo a lo largo de la cadena alimentaria [1, 3, 4]. En la actualidad se dispone de una colección de *Campylobacter* spp. de alrededor de 2000 cepas. Con el fin de obtener una información más precisa sobre la persistencia de este microorganismo a lo largo de la cadena alimentaria, se han aislado varios genotipos persistentes en el matadero. De estos se han secuenciado con un equipo MiSeq (Illumina) 45 de ellas.

El proyecto consiste en diseñar un workflow que permita, a partir de los datos obtenidos en formato fastq proporcionados por el equipo, conseguir realizar las fases de trimming y evaluación de la calidad de las secuencias obtenidas, obtención de contigs, assembling y anotación, para poder tener la información de la secuencia de genes del genoma completo [5, 6, 7] de las cepas de *Campylobacter jejuni* secuenciadas. En la actualidad, existen varios programas desarrollados por varios grupos de investigación internacional que realizan las funciones demandadas. Se trata de buscar la solución más eficaz y fácil de implementar y que dé los mejores resultados, por lo que habrá que comparar diferentes programas y estrategias. Adicionalmente, se requiere incorporar en este workflow, o en análisis paralelos [8], la posibilidad de detectar insertos de origen viral y/o plásmidos en el genoma y herramientas que permitan la comparación rápida de los genomas de las distintas cepas aisladas, algunas de ellas pertenecientes a cepas altamente clonales. Esta herramienta se ha demandado por parte de un grupo sin conocimientos informáticos, por lo que se requiere desarrollar un entorno de fácil uso por su parte.

El proyecto plantea una colaboración entre los grupos ADMIRABLE y TECNOFOOD de la Universidad de Burgos. Especializados en informática y ciencia y tecnología de los alimentos respectivamente. Dada esta combinación de disciplinas, el proyecto se encuentra en el marco de los trabajos considerados dentro del campo de la bioinformática.

Tarea 1 - Desarrollo de la imagen Docker	
Tarea 1.1 - Adaptar la imagen previa orientada a Galaxy	50 horas
Tarea 1.2 - Permitir desplegar la imagen en un servidor	30 horas
Tarea 2 - Desarrollo del workflow en Galaxy	
Tarea 2.1 - Selección de las herramientas	50 horas
Tarea 2.2 - Configuración paramétrica	30 horas
Tarea 2.3 - Habilitar ejecución del workflow de manera programática	30 horas
Tarea 3 - Desarrollo de la interfaz de uso	
	60 horas
Tarea 4 - Desarrollo del sistema de tratamiento de datos de salida	
	50 horas

Cuadro 1.1: Plan de Trabajo

1.2. Objetivos y enfoque

El objetivo principal del proyecto es el desarrollo del workflow que permita el análisis de las cepas de *Campylobacter jejuni*, para lo que se utilizarán Galaxy y las herramientas disponibles en la «tool shed» (conjunto de herramientas que ofrece Galaxy para su instalación) para cada paso. Galaxy es una herramienta que permite análisis computacionales de datos biológicos. El segundo objetivo se centra en crear una herramienta gráfica a través de la cual se pueda facilitar el uso de este workflow, que se llevará a cabo utilizando Python, Qt y la API de Galaxy. Además, para facilitar el despliegue y ejecución de la herramienta desarrollada, se va a crear un contenedor Docker. Por lo tanto, el siguiente objetivo parcial será desarrollar la imagen Docker que sirva de base. Finalmente, se desea tener alguna forma sencilla de tratar los datos de salida del workflow, por lo que dentro de la interfaz gráfica se incluirán herramientas con las que gestionar toda la información resultante en forma de gráficos, tablas tipo hoja de cálculo, formatos pdf, etc.

Objetivos del proyecto:

- Desarrollar el workflow necesario para el análisis de las cepas en Galaxy
- Crear un sistema Docker sobre el que desplegar el proyecto
- Desarrollar una interfaz gráfica para simplificar la utilización de la aplicación
- Añadir un sistema de gestión sencilla de los datos de salida.

1.3. Metodología y plan de trabajo

1.3.1. Metodología

La metodología utilizada en el desarrollo del proyecto, dada su cercanía en su estructura a un proyecto de software tradicional, será de tipo ágil, basada en reuniones en cada sprint. La carga de trabajo, como aproximación a falta de conocer ciertos requisitos que puedan surgir durante el desarrollo, se dividirá en la estructura siguiente.

1.3.2. Plan de Trabajo

Sprint 1 (18/9/2018 - 3/10/2018)

El primer sprint ha estado centrado tanto en definir con más exactitud la dirección del proyecto como en un primer acercamiento a las principales herramientas con las que va a desarrollarse.

Tras unos primeros pasos con Galaxy [9] y Docker [10], se ha tomado como referencia el trabajo Bioinfworkflow de Sergio Chico [11] como base para la imagen Docker del proyecto. Dado que el proyecto de Github daba algunos problemas en la instalación, se ha desarrollado un script propio que produce los mismos resultados.

Una vez se ha tenido disponible la imagen de Docker, el sprint se ha centrado en algunos aspectos importantes para partes futuras del desarrollo. Entre ellos destaca la investigación acerca del formato de los workflows de Galaxy (.ga) ya que en un futuro será necesario generar este tipo de ficheros para introducirlos en Galaxy. También resulta relevante la investigación acerca de las posibilidades que ofrece la API de Galaxy [11] y su utilidad en Bioblend [12], que nos facilitan la opción de utilizar Galaxy sin necesidad de hacerlo a través de su interfaz.

Sprint 2 (4/10/2018 - 17/10/2018)

La primera semana de este sprint ha estado dirigida a lograr una imagen Docker de Galaxy que contenga un set de herramientas básicas para formar un primer workflow. Se han valorado varias opciones de instalación en las que se han utilizado tanto la imagen básica de Galaxy [13] como la imagen de Bioinfworkflow [11]. Finalmente, se ha optado por utilizar Bioinfworkflow ya que parte de las herramientas necesarias ya estaban incluidas. Para realizar esta tarea se ha creado un nuevo fichero Dockerfile así como el listado de herramientas necesarias para su instalación.

Sprint 3 (18/10/2018 - 31/10/2018)

El sprint ha estado centrado en la correcta ejecución del workflow con las herramientas iniciales desde Galaxy. Durante el proceso de configuración, han surgido varias complicaciones que han impedido terminar el workflow completo en este sprint. En un principio, han surgido problemas con el filtrado de calidad utilizando Prinseq. Este problema no ha llegado a ser resuelto en este sprint a falta de tratar el tema con el grupo de Tecnofood. A continuación, se encontraron ciertos problemas con el formato de salida de la herramienta Prokka. A pesar de que la salida está marcada como formato gff3, un parámetro interno lo etiquetaba como gff. Esto impedía que la salida de Prokka pudiese ser utilizada como entrada en las herramientas siguientes.

Dados estos errores, se decidió trabajar en paralelo con la API de Galaxy desde Python, para intentar ejecutar tanto las herramientas como el workflow de una manera menos restringida. Finalmente, se ha llevado a cabo el desarrollo necesario para subir los ficheros a un historial y ejecutar cada una de las herramientas del workflow desde Python.

Sprint 4 (1/11/2018 - 14/11/2018)

La prioridad en este punto se ha centrado en completar el workflow desde Galaxy. Han surgido varios problemas en esta tarea. La primera es un bug en Mac por el cual los archivos eliminados dentro de Docker no se eliminan del todo y quedan fijados en un fichero residual. Quizá esto pueda deberse a que OSX no soporta de manera nativa la virtualización a nivel de sistema operativo, sino que se basa en Hyperkit para crear una capa de virtualización. Esto implica que cada cierto tiempo hay que eliminar la imagen completa de Docker para poder liberar espacio, dado el gran tamaño de los archivos con los que se trabaja. Debido a ello, la tarea de completar el workflow se ha visto retrasada. Además, la ejecución de la herramienta Roary a través de Galaxy ejecuta sin errores pero no devuelve ningún resultado, lo que ha impedido continuar con la parte final del workflow.

Además de la tarea ya comentada, en este sprint se ha trabajado en el acceso al contenedor Docker desde otro ordenador en una red local. Con el objetivo de desplegar el servicio en un servidor.

También se ha tomado contacto con la interfaz gráfica, realizando unas pruebas en las que simplemente se muestra alguna información extraída de la API de Galaxy en etiquetas creadas con PyQt.

Sprint 5 (15/11/2018 - 28/11/2018)

Al igual que en los sprints anteriores, gran parte de la carga de trabajo se ha centrado en resolver problemas en la ejecución del workflow a través de Galaxy. Se han realizado numerosas ejecuciones para comprobar si las salidas de cada herramienta eran las correctas. Esto ha servido para concluir que, al parecer, un fallo en la herramienta Roary incluida en Galaxy, impide que los ficheros retornados tengan contenido. Esto se ha comprobado a través de la ejecución de Roary standalone con los mismos datos de entrada y los mismos parámetros, obteniendo de esta manera los ficheros correctos.

Para ahorrar en tiempos de ejecución se han utilizado los ficheros fasta ya generados previamente, no los generados con la herramienta Spades de nuestro propio workflow. En el próximo sprint se tratará de integrar la parte previa a los pasos que ya son correctos.

También se ha añadido un fichero .gitignore para evitar la existencia de ficheros irrelevantes en el repositorio.

Parte del trabajo de este sprint se ha destinado a la redacción de la propuesta de proyecto y de la introducción a la documentación del mismo.

Sprint 6 (29/11/2018 - 12/12/2018)

El trabajo de este sprint se ha centrado en fragmentar el proceso del workflow lo máximo posible para detectar dónde se están produciendo los errores. Inicialmente se ha acortado el workflow hasta el paso de ensamblaje con Spades, ya que los problemas surgían en este punto. Posteriormente se ha ejecutado el workflow individualmente en lugar de por colecciones. De esta manera, se ha detectado que el problema se está dando en la ejecución de Spades con dos secuencias concretas: 590 y 443. Tras investigar probando varias ejecuciones con diferentes parámetros, se ha llegado a la conclusión de que no era un fallo de configuración de la herramienta, sino de hardware. Al ceder a docker una cantidad mayor de memoria RAM (8 Gb), el problema se ha solucionado. A continuación, se ha pasado a ejecutar de nuevo por colecciones hasta el paso de Spades, para comprobar si con este cambio ha sido suficiente para que la ejecución sea correcta con este formato.

También se ha realizado una modificación del fichero .gitignore para mantener los archivos .pdf generados por L^AT_EX.

Sprint 7 (13/12/2018 - 26/12/2018)

La tarea principal de este sprint ha sido la creación de una herramienta Roary que poder integrar en Galaxy. Se valoró la opción de crear una nueva imagen Galaxy instalando la herramienta desde la creación inicial de Docker, pero finalmente se ha optado por subir esta versión de Roary al *tool shed* de Galaxy.

A continuación, se han realizado varias comprobaciones del funcionamiento de la integración de esta herramienta, con resultados positivos. Sin embargo, al ejecutar el workflow completo,

parecen surgir de nuevo errores en la parte de Prokka, provocados por la ejecución previa de Spades.

También se ha añadido el apartado de la Introducción de la documentación.

Sprint 8 (27/12/2018 - 9/1/2019)

El trabajo en este sprint se ha enfocado a completar definitivamente la ejecución del workflow solucionando los errores existentes. La máquina virtual con la que se realizaban estas pruebas disponía de 4 GB de RAM, insuficiente para la ejecución con este conjunto de datos. Esto causaba algunos errores poco descriptivos al ejecutarlo. El equipo de pruebas se ha aumentado a 32 GB de RAM, cediendo 16 GB a la máquina virtual, solucionando así estos errores.

A continuación, se ha desarrollado un script Python a partir del cual realizar todas las tareas necesarias para ejecutar el workflow, utilizando la API de Galaxy.

Se ha invertido el poco tiempo restante del sprint en el desarrollo de la documentación, definiendo la estructura de los apartados de «estado del arte» y «sistema, diseño y desarrollo».

2

Análisis genético de *Campylobacter jejuni*. Estado del arte

2.1. Introducción

2.2. *Campylobacter jejuni*

2.3. Técnicas en bioinformática

2.3.1. Control de calidad

2.3.2. Ensamblado

2.3.3. Anotación

2.3.4. Análisis pangenómico

2.3.5. Workflows en bacterias / Workflows en general

2.4. Herramientas en informática

2.4.1. Virtualización

2.4.2. Plataforma de soporte del workflow

2.4.3. Herramientas GUI

3

Sistema, diseño y desarrollo

3.1. Estructura general del sistema

3.2. Desarrollo y configuración de Docker

3.3. Desarrollo y configuración del workflow en Galaxy

3.4. Desarrollo de la capa externa con Python

3.5. Otros aspectos del desarrollo

4

Experimentos Realizados y Resultados

4.1. Bases de datos y protocolo

4.2. Sistemas de referencia

4.3. Escenarios de pruebas

4.4. Experimentos del sistema completo

5

Conclusiones y trabajo futuro

Glosario de acrónimos

- **IS**: Iris Subject
- **DCT**: Discrete Cosine Transform
- **WED**: Weighted Euclidean Distance

Bibliografía

- [1] Lourdes García-Sánchez, Beatriz Melero, Isabel Jaime, Marja-Liisa Hänninen, Mirko Rossi, and Jordi Rovira. *Campylobacter jejuni* survival in a poultry processing plant environment. *Food Microbiology*, 65:185–192, aug 2017.
- [2] *Campylobacteriosis - Annual Epidemiological Report 2016 [2014 data]*.
- [3] Lourdes García-Sánchez, Beatriz Melero, Ana Ma Diez, Isabel Jaime, and Jordi Rovira. Characterization of *Campylobacter* species in Spanish retail from different fresh chicken products and their antimicrobial resistance. *Food Microbiology*, 76:457–465, dec 2018.
- [4] Beatriz Melero, Pekka Juntunen, Marja-Liisa Hänninen, Isabel Jaime, and Jordi Rovira. Tracing *Campylobacter jejuni* strains along the poultry meat production chain from farm to retail by pulsed-field gel electrophoresis, and the antimicrobial resistance of isolates. *Food Microbiology*, 32(1):124–128, oct 2012.
- [5] Clifford G. Clark, Chrystal Berry, Matthew Walker, Aaron Petkau, Dillon O. R. Barker, Cai Guan, Aleisha Reimer, and Eduardo N. Taboada. Genomic insights from whole genome sequencing of four clonal outbreak *Campylobacter jejuni* assessed within the global *C. jejuni* population. *BMC Genomics*, 17(1):990, dec 2016.
- [6] Ann-Katrin Llarena, Eduardo Taboada, and Mirko Rossi. Whole-Genome Sequencing in Epidemiology of *Campylobacter jejuni* Infections. *Journal of clinical microbiology*, 55(5):1269–1275, may 2017.
- [7] S. Zhao, G. H. Tyson, Y. Chen, C. Li, S. Mukherjee, S. Young, C. Lam, J. P. Folster, J. M. Whichard, and P. F. McDermott. Whole-Genome Sequencing Analysis Accurately Predicts Antimicrobial Resistance Phenotypes in *Campylobacter* spp. *Appl. Environ. Microbiol.*, 82(2):459–466, jan 2016.
- [8] C. P. A. Skarp, O. Akinrinade, A. J. E. Nilsson, P. Ellström, S. Myllykangas, and H. Rautealin. Comparative genomics and genome biology of invasive *Campylobacter jejuni*. *Scientific Reports*, 5(1):17300, dec 2015.
- [9] Galaxy. <https://usegalaxy.org/>.
- [10] Docker - Build, Ship, and Run Any App, Anywhere. <https://www.docker.com/>.
- [11] Sergio Chico. :whale: Docker with Galaxy for Bioinformatic Bacterial Sequencing Workflows: Serux/docker-galaxy-BioInfWorkflow. <https://github.com/Serux/docker-galaxy-BioInfWorkflow>, June 2018. original-date: 2018-05-17T01:10:05Z.
- [12] Galaxy API. <https://galaxyproject.org/develop/api/>.
- [13] Björn Grüning. :whale::bar_chart::books: Docker Images tracking the stable Galaxy releases.: bgruening/docker-galaxy-stable. <https://github.com/bgruening/docker-galaxy-stable>, October 2018. original-date: 2014-08-12T13:26:14Z.



Manual de utilización



Manual del programador