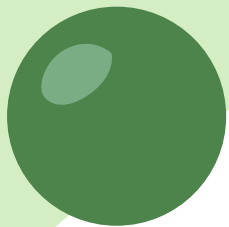


Uso de gráficas de Bruijn para el ensamble de un genoma completo

- Francisco Alejandro Arganis Ramírez
- Erika Yusset Madera Baldovinos

Introducción



El ensamble de un genoma completo a partir de las lecturas de secuenciación es una de las tareas más importantes en biología molecular. Si ya se ha ensamblado el genoma del mismo organismo o de alguno muy similar, se puede realizar un ensamble por genoma de referencia en el que primero se alinean las lecturas contra la referencia para reducir la complejidad del ensamble. Si no se cuenta con un genoma de referencia, es necesario realizar un ***ensamble de novo***.

Existen dos estrategias principales para realizar un ensamble *de novo*: el consenso por disposición de traslapes (overlap layout consensus u OLC) y las gráficas de De Bruijn (De Bruijn graph o DBG). Ambas estrategias solo corresponden a la etapa de construcción de contigs.

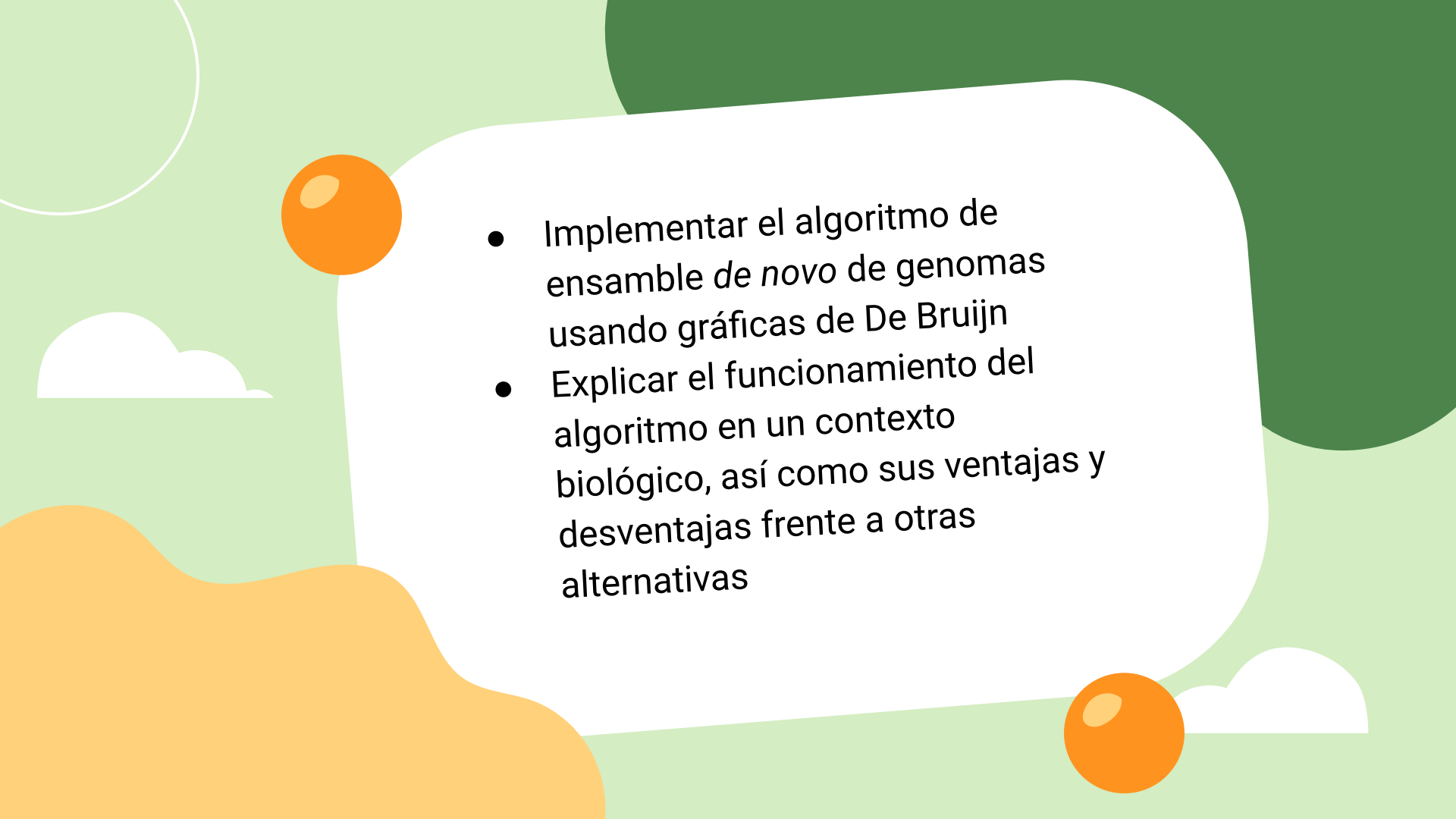
En algoritmos DBG se fragmentan las lecturas en k -meros y se construye una gráfica de De Bruijn en la que cada k -mero corresponde a una arista dirigida del vértice a al vértice b , donde a tiene las primeras $k-1$ bases y b las últimas $k-1$ bases del k -mero. Para los casos en los que se pueden repetir k -meros, la multiplicidad se representa como el peso de las aristas. Los contigs se construyen siguiendo un camino euleriano en la gráfica de De Bruijn.





Objetivos

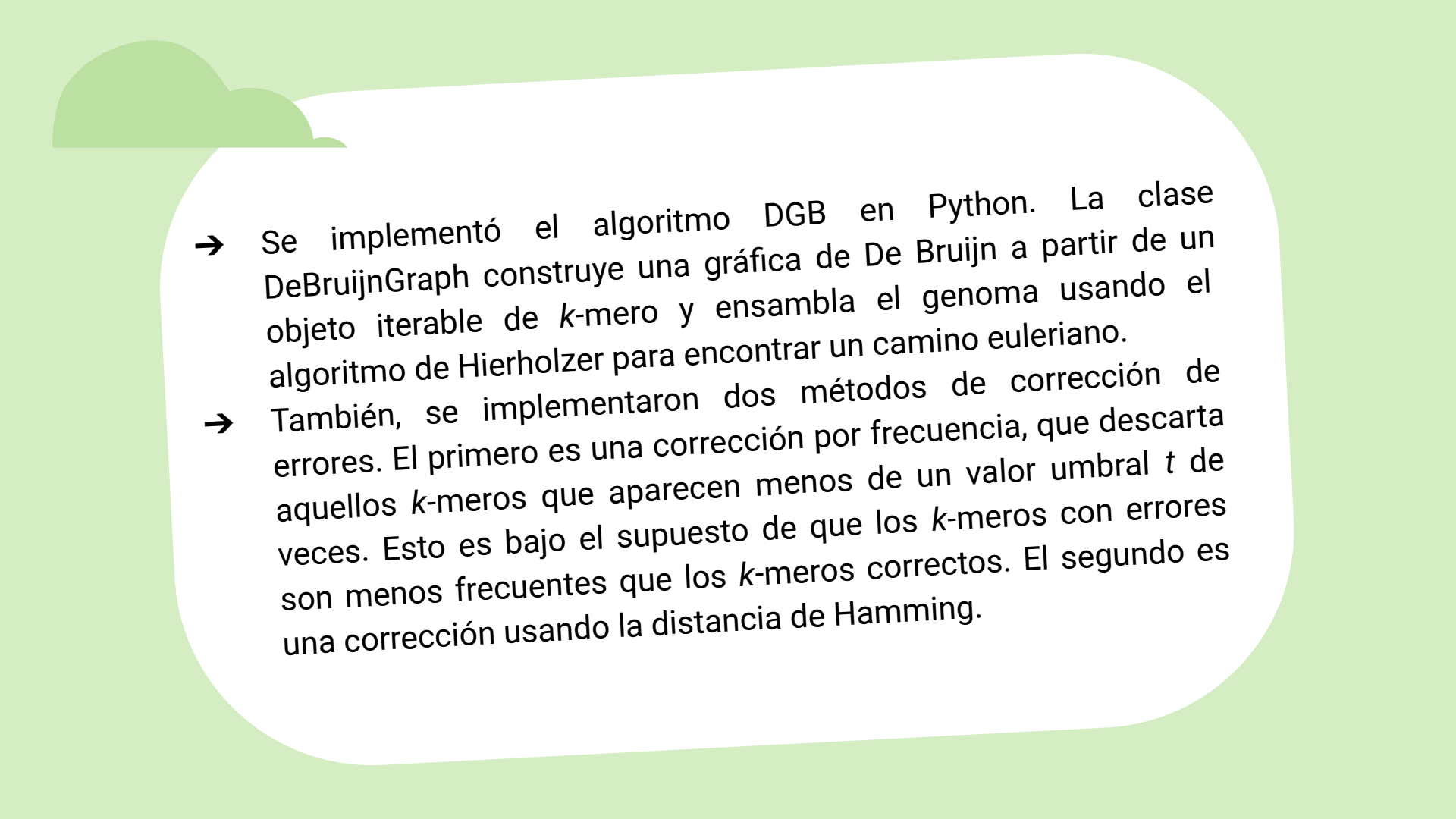
¿Qué queremos lograr con este proyecto?


- 
- Implementar el algoritmo de ensamble *de novo* de genomas usando gráficas de De Bruijn
 - Explicar el funcionamiento del algoritmo en un contexto biológico, así como sus ventajas y desventajas frente a otras alternativas

Metodología

¿Cómo lo hicimos?



- 
- Se implementó el algoritmo DGB en Python. La clase DeBruijnGraph construye una gráfica de De Bruijn a partir de un objeto iterable de k -mero y ensambla el genoma usando el algoritmo de Hierholzer para encontrar un camino euleriano.
 - También, se implementaron dos métodos de corrección de errores. El primero es una corrección por frecuencia, que descarta aquellos k -meros que aparecen menos de un valor umbral t de veces. Esto es bajo el supuesto de que los k -meros con errores son menos frecuentes que los k -meros correctos. El segundo es una corrección usando la distancia de Hamming.



Se descargaron las lecturas de secuenciación de nueva generación del fago Bacata de *Xylella*, secuenciado con la plataforma Illumina MiniSeq, del ENA Browser. Los archivos fastq con las 54495 lecturas, uno por cada dirección, están disponibles en <https://www.ebi.ac.uk/ena/browser/view/ERX5328366>. También, se descargó del NCBI el archivo fasta con el genoma ya ensamblado del mismo fago, de tamaño 56232 bp, disponible en https://www.ncbi.nlm.nih.gov/nuccore/NC_052973.1?report=fasta. La calidad de las lecturas se evaluó con el software FastQC.



Resultados y discusión



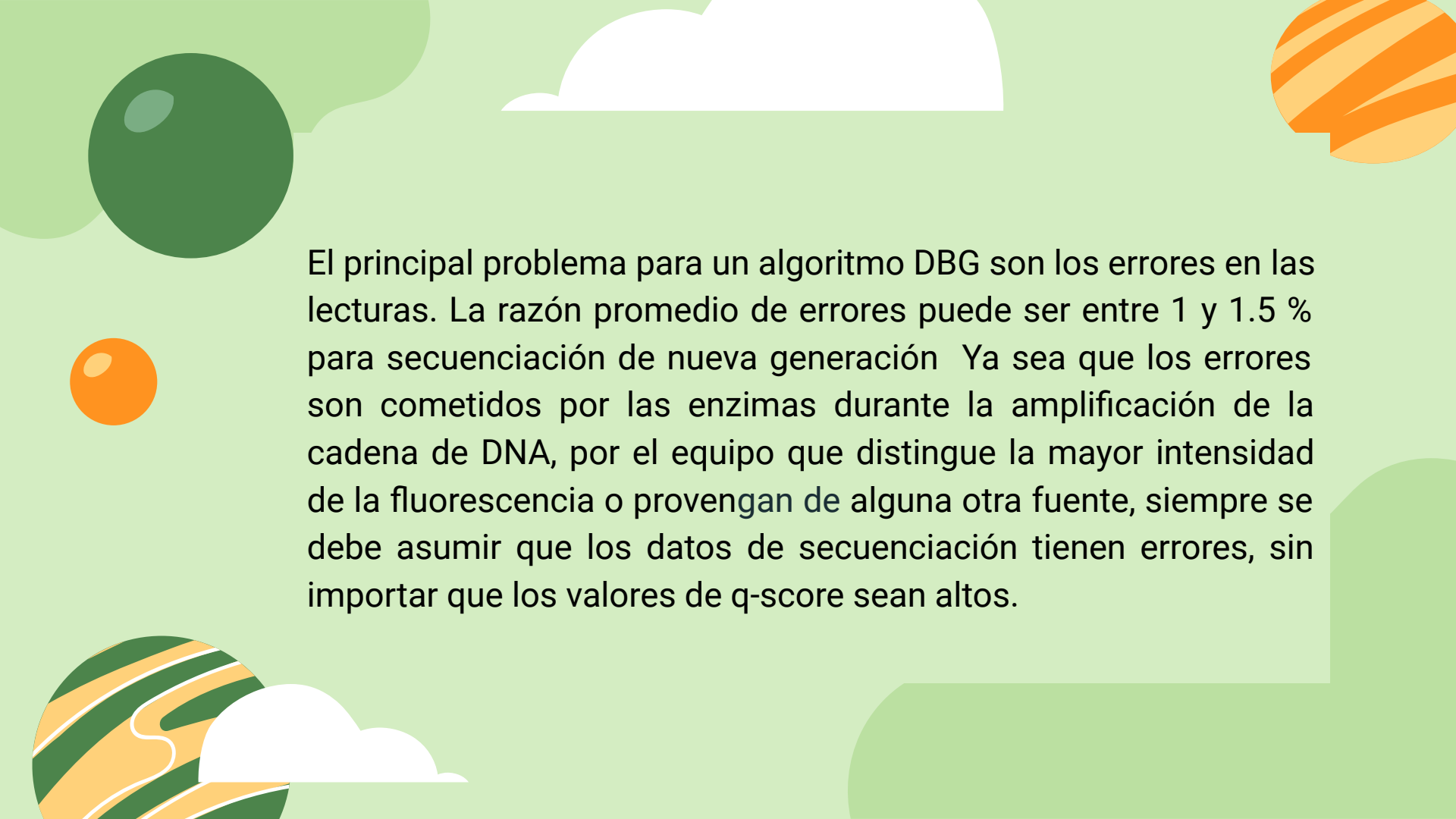
Los resultados del alineamiento del ensamble del genoma del fago Bacata con el algoritmo DBG implementado

Datos de entrada	Tamaño del ensamble (bp)	1er región alineada	2da región alineada	3er región alineada
Datos crudos	3612673	Región: 40466-42426 Coincidencias: 1960/1961	Región: 40466-42418 Coincidencias: 1952/1953	Región: 40466-42413 Coincidencias: 1944/1948
Datos crudos (<i>k</i> -meros únicos)	215852	Región: 29039-29367 Coincidencias: 328/330	Región: 42764-43090 Coincidencias: 326/329	Región: 34526-34849 Coincidencias: 322/326
Datos de calidad	108159	Región: 37114-43918 Coincidencias: 6804/6805	Región: 36427-43221 Coincidencias: 6794/6795	Región: 36427-41028 Coincidencias: 4600/4603
Datos de calidad (<i>k</i> -meros únicos)	3544	Región: 23283-25620 Coincidencias: 2337/2341	Región: 22543-23114 Coincidencias: 572/572	Región: 25588-25978 Coincidencias: 391/391

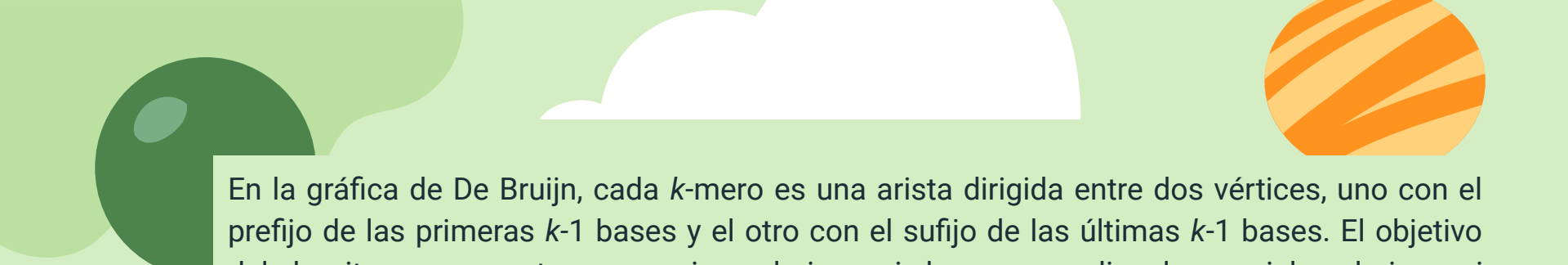
Datos corregidos por frecuencia	4375	Región: 47316-51690 Coincidencias: 4375/4375	NA	NA
Datos corregidos por frecuencia (<i>k</i> -meros únicos)	3989	Región: 36822-40810 Coincidencias: 3989/3989	NA	NA
Datos corregidos por distancia de Hamming	14949	Región: 29023-43971 Coincidencias: 14949/14949	NA	NA
Datos corregidos por distancia de Hamming (<i>k</i> -meros únicos)	14446	Región: 46096-56232 Coincidencias: 10137/10137	Región: 1-4309 Coincidencias: 4309/4309	NA
Datos simulados	3006626	Región: 1-40036 Coincidencias: 40036/40036	Región: 44432-56232 Coincidencias: 11801/11801	Región: 39955-44426 Coincidencias: 4472/4472
Datos simulados (<i>k</i> -meros únicos)	56190	Región: 1-26761 Coincidencias: 26761/26761	Región: 26759-44426 Coincidencias: 17668/17668	Región: 44432-56232 Coincidencias: 11801/11801

Los resultados del alineamiento del ensamble del genoma sintético, para los diferentes valores de k :


k	Tamaño del ensamble (bp)	Coincidencias	Espacios
20	18616	16754/18631 (89.9254 %)	30/18631 (0.1610 %)
25	34703	23283/26146 (89.0500 %)	72/26146 (0.2754 %)
30	50628	23982/26126 (91.7936 %)	2/26126 (0.0077 %)
35	16271	14680/16277 (90.1886 %)	12/16277 (0.0737 %)
40	45077	23670/26133 (90.5751 %)	10/26133 (0.0383 %)
45	16094	14651/16095 (91.0282 %)	2/16095 (0.0124 %)
50	25632	22703/25649 (88.5142 %)	34/25649 (0.1326 %)



El principal problema para un algoritmo DBG son los errores en las lecturas. La razón promedio de errores puede ser entre 1 y 1.5 % para secuenciación de nueva generación. Ya sea que los errores son cometidos por las enzimas durante la amplificación de la cadena de DNA, por el equipo que distingue la mayor intensidad de la fluorescencia o provengan de alguna otra fuente, siempre se debe asumir que los datos de secuenciación tienen errores, sin importar que los valores de q-score sean altos.

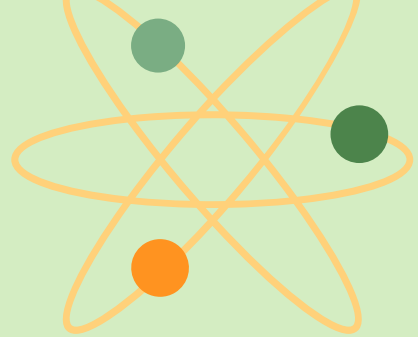


En la gráfica de De Bruijn, cada k -mero es una arista dirigida entre dos vértices, uno con el prefijo de las primeras $k-1$ bases y el otro con el sufijo de las últimas $k-1$ bases. El objetivo del algoritmo es encontrar un camino euleriano, si el genoma es lineal, o un ciclo euleriano, si el genoma es circular. Al pasar por todas las aristas, por construcción de la gráfica, se asegura que existe un traslape entre dos k -meros consecutivos, que es precisamente las $k-1$ bases del vértice en común. Por tanto, al recorrer todos los k -meros se recupera el genoma original.



Cuando una lectura tiene un error, todos los k -meros que pasan por el error se ven afectados. Esto puede causar que la gráfica deje de ser euleriana, que se generen nuevos vértices, que la gráfica se vuelva desconexa u otros cambios en su topología. La estrategia de gráficas de De Bruijn es muy atractiva para el desarrollo de software de ensamblaje a partir de lecturas cortas. En situaciones con datos idealizados, es un algoritmo que puede ensamblar eficientemente las lecturas. Sin embargo, en la práctica presenta muchas dificultades que se tienen que resolver para poder ensamblar correctamente un genoma.

Conclusiones



Ventajas DBG

- Es escalable. La complejidad en tiempo es $O(NL)$ y en espacio es $O(\min(NL, G))$.
- Transforma el problema NP-completo de camino hamiltoniano en un problema de camino euleriano con solución en tiempo lineal.
- La implementación del algoritmo es muy simple.
- Cuando hay pocos k -meros repetidos en el genoma, se puede ensamblar casi todo el genoma sin necesidad de considerar un problema de supercamino.
- Las secuencias repetidas no significan un aumento en el costo computacional, como en OLC.

Desventajas DBG

- No es tolerante a errores de secuenciación. Requiere un algoritmo especializado de corrección de errores como parte del preprocesamiento.
- Los errores y la cobertura de las lecturas pueden hacer que la gráfica sea desconexa y no euleriana.
- Aún si la gráfica es euleriana, hay múltiples caminos eulerianos que corresponden a diferentes genomas. Distinguir el camino correcto es un problema de supercamino y es NP-completo.
- Si hay regiones grandes con secuencias repetidas se producen errores en la secuencia ensamblada a menos que se usen métodos auxiliares para resolver las repeticiones.

Gracias

