Biotecnología de Plantas
Posgrado

Cinvestav

Transcriptómica

Clase 3 - Ensamble de novo

Bionformática y Bioestadística 2023

Selene L. Fernández-Valverde
regRNAlab.github.io
@SelFdz

Objetivos de aprendizaje

En esta clase aprenderemos:

- Como funciona el ensamble de RNA-Seq.
- A usar Trinity para ensamblar datos de novo.

¿Qué es el ensamble de transcriptomas *de novo*?

Es tomar lecturas pequeñas de RNA-Seq y convertirlas a transcritos completos **sin la ayuda un genoma como referencia.**

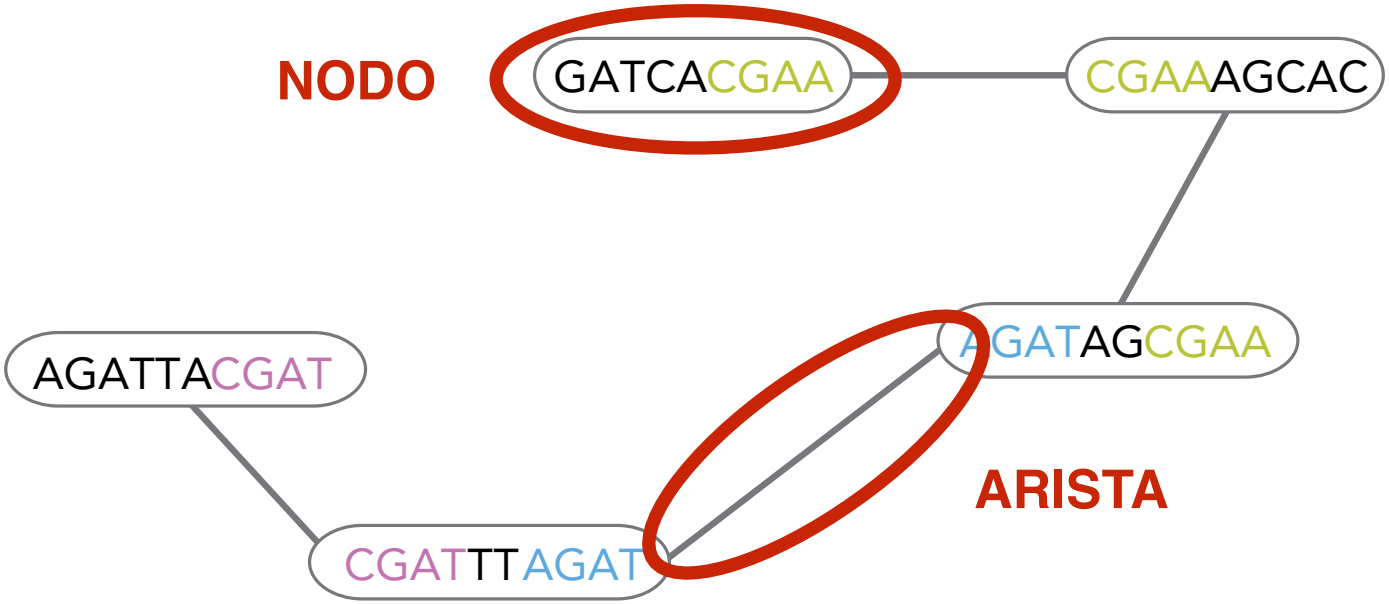
Revisión histórica

El primer problema de ensamble de secuencias que se enfrentó fue el ensamble de genomas.

Los primeros ensambladores de secuencias usados con secuencias tipo Sanger utilizaban un paradigma conocido como ‘overlap-layout-consensus’. Este tipo de aproximaciones calculaban todas las posibles superposiciones congruentes, creando un grafo de sobrelape de secuencias. Cada **nodo** en el grafo corresponde a una **lectura** y cada **arista** denota el ***sobrelape entre dos secuencias***. Este grafo se usa para calcular contigs consenso.

Grafo de sobrelape de secuencias

Figure 2: Overlap Graph of Five Reads

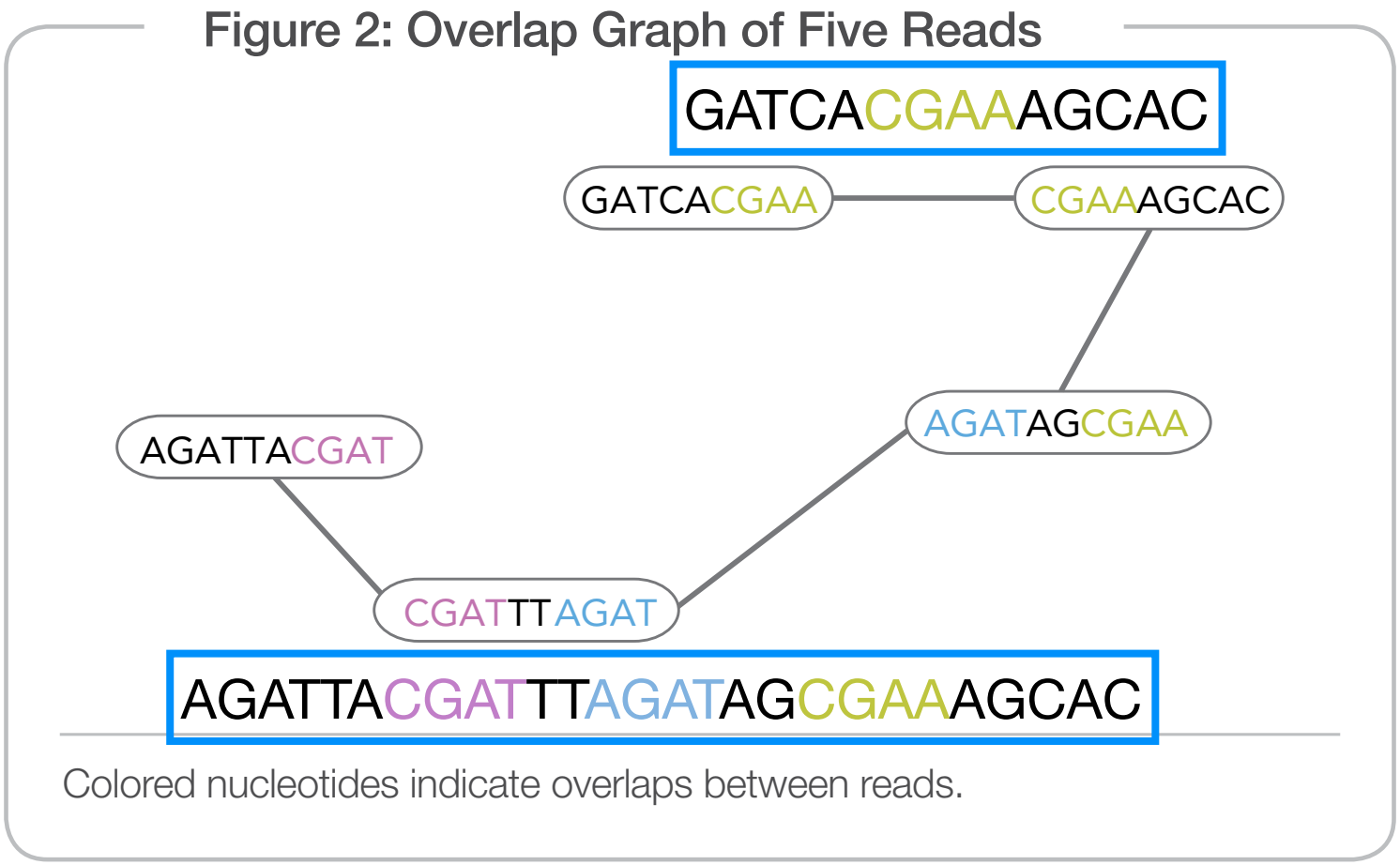


Colored nucleotides indicate overlaps between reads.

Anatomía de un sobrelape

Los sobrelapes nos permiten pegar una secuencia con otra e inferir una secuencia continua más larga.

Grafo de sobrelape de secuencias



Pero con millones de secuencias …

- Los grafos de sobrelape de secuencias son muy costosos en términos computacionales dado que hay que calcular todos los sobrelapes de todas las secuencias contra todas.

- Dado su costo computacional no son prácticos para usarlos con datos de secuenciación masiva.

El regreso del k-mer

¿Qué es un k-mer (o k-mero)?

Se refiere a todas las posibles subsecuencias de tamaño k que existen en una secuencia. El número de k-meros en una secuencia es:

L - k +1

L = longitud de la secuencia

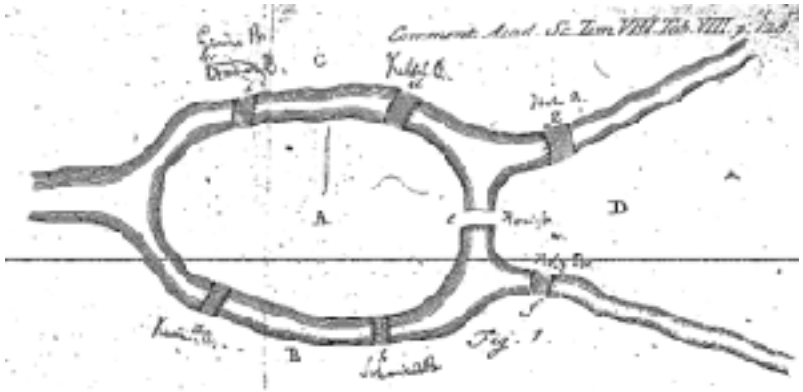
k = tamaño de subsecuencias

El despertar del grafo

- Hace 300 años, en la ciudad de Konisberg (ahora Caliningrado, Rusia) se planteó un problema conocido como el “problema de los puentes de Konisberg”

- Este problema plantea que, dado que hay 7 puentes que cruzan el río en esta ciudad, ¿es posible visitar cada parte de la ciudad cruzando cada puente una sola vez y regresando al lugar de inicio?

- En 1735, el gran matemático Leonhard Euler resolvió este problema usando el primer grafo, iniciando así la rama de las matemáticas conocida como “teoría de grafos”.



El despertar del grafo de Bruijn

- En 1946 el matemático holandés Nicolaas de Bruijn se interesó en el problema de las “supercadenas”. Brevemente el problema consistía en encontrar la supercadena circular más corta que contenga todas las “subcadenas” posibles de tamaño k (k-mers) de un alfabeto determinado.

- de Bruijn resolvió este problema tomando prestada la solución de Euler al problema de los Puentes de Königsberg.

¿Y que tiene que ver el Sr. de Bruijn con transcriptómica?

Este descubrimiento es un buen ejemplo de conocimiento que - al principio básico - se vuelve muy aplicable en un futuro.

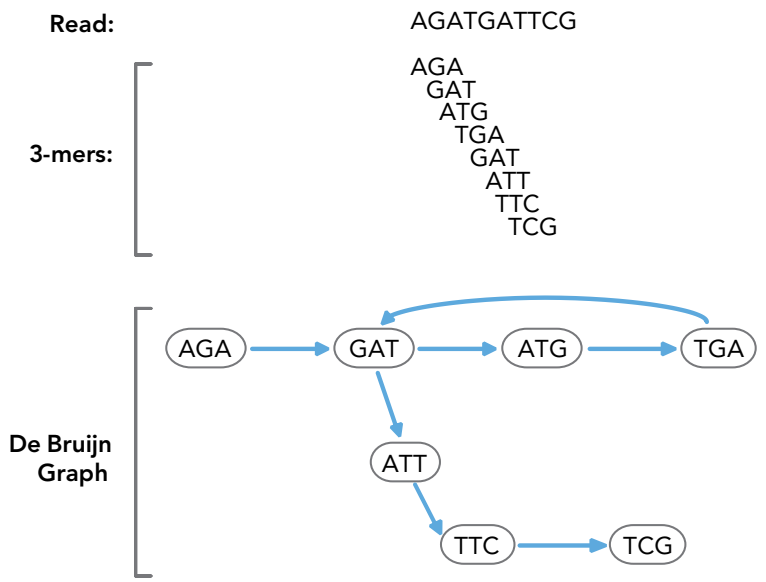
A finales de los 90's, varios matemáticos y bioinformáticos vieron las similitudes entre el problema resuelto por de Bruijn y el problema del ensamble de secuencias. En 2001 Pevzner, Tang y Waterman propusieron el primer programa que utilizaba grafos de Bruijn para ensamblar genomas. Tenía la ventaja de tener menos errores en zonas repetitivas.

Si embargo, esta estrategia fue poco utilizada hasta la llegada de la secuenciación masiva. Dada la cantidad de datos y el tamaño extremadamente corto de las secuencias era mucho más eficiente ensamblar datos usando grafos de Bruijn.

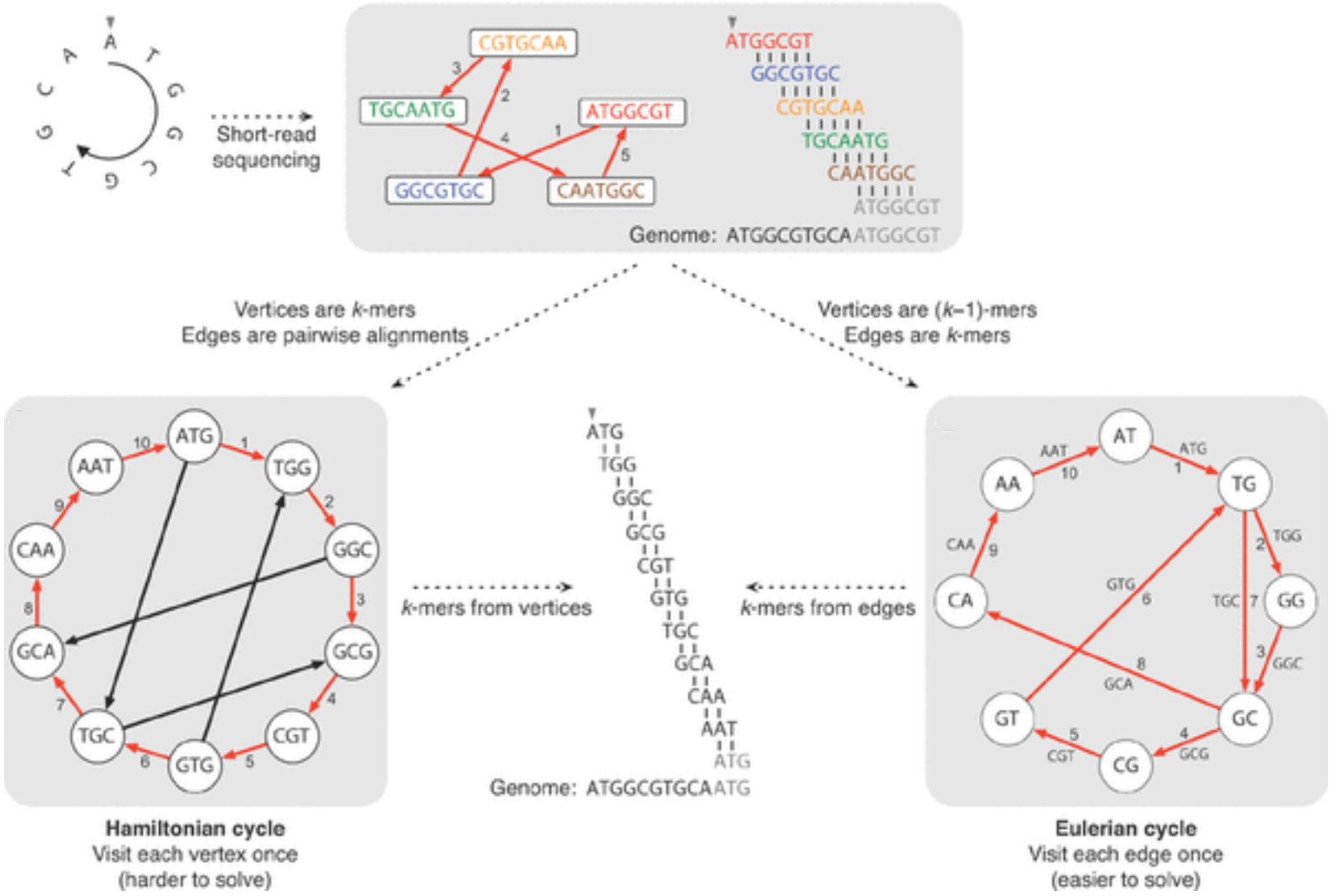
Generando un grafo de Bruijn

- **Paso 1** - Toma las secuencias y rómpelas en pequeños pedazos de tamaño k (k-mers).
- **Paso 2** - Construye un grafo de Bruijn usando esas piezas
- **Paso 3** - Reconstruye la secuencia original buscando los 'caminos' dentro del grafo

Figure 3: De Bruijn Graph for Read with K=3



The length of overlaps is k-1=2. Gray arrows indicate where all the k-mers derived from the one read are placed in the graph. Blue arrows indicate the order of the k-mers and their overlaps.

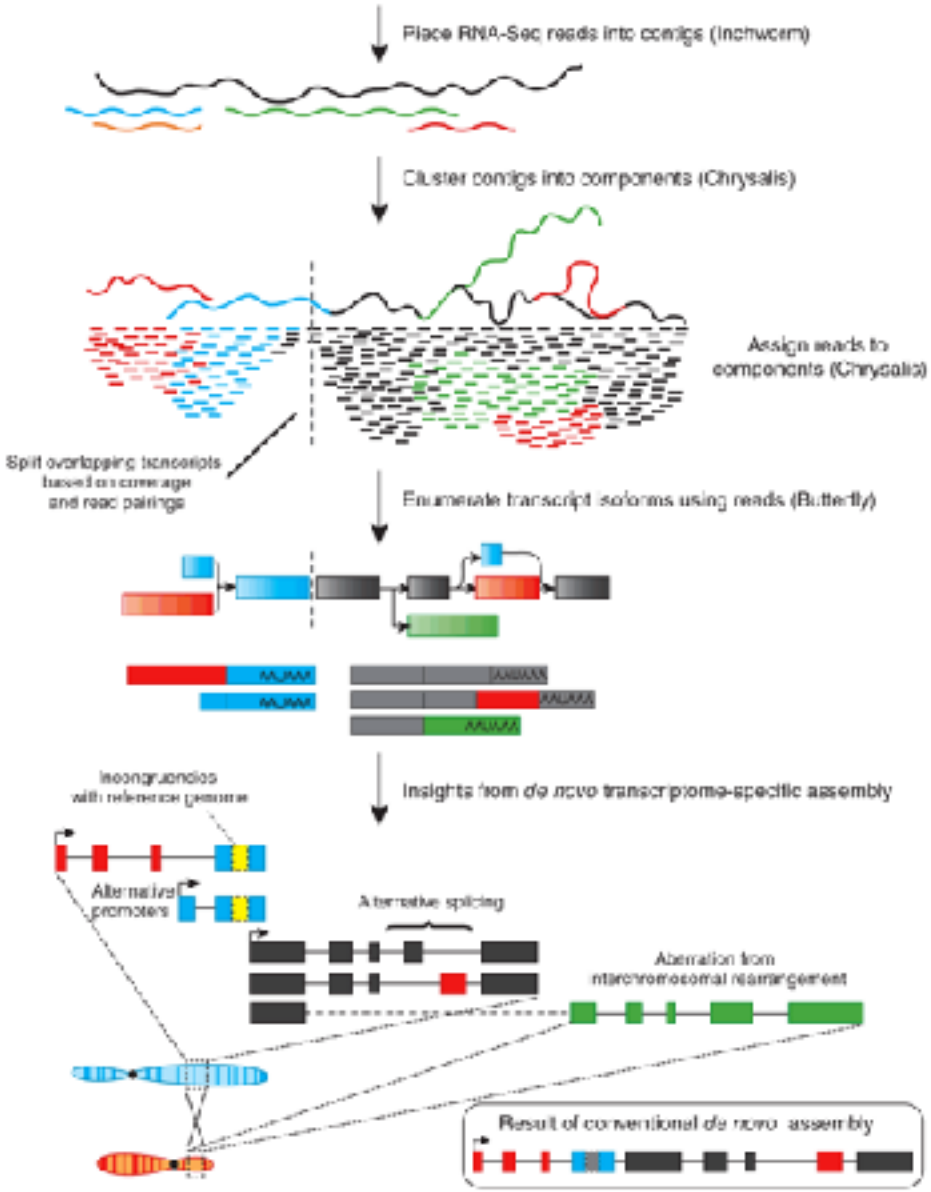


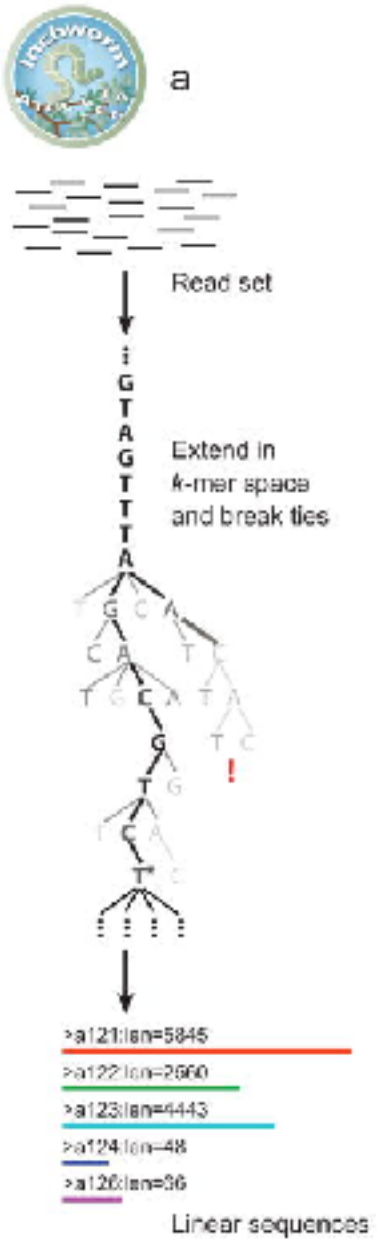
http://www.nature.com/nbt/journal/v29/n11/full/nbt.2023.html#bx1

Programas para ensamblar transcriptomas *de novo*

- **Trinity** (https://github.com/trinityrnaseq/trinityrnaseq)
- Trans-ABySS (https://github.com/bcgsc/transabyss)
- SOAPdenovo-Trans (http://soap.genomics.org.cn/SOAPdenovo-Trans.html)
- Velvet/Oases (https://www.ebi.ac.uk/~zerbino/oases/)

Es un método para la reconstrucción de transcriptomas basados en datos de RNA-Seq. Trinity combina 3 módulos principales



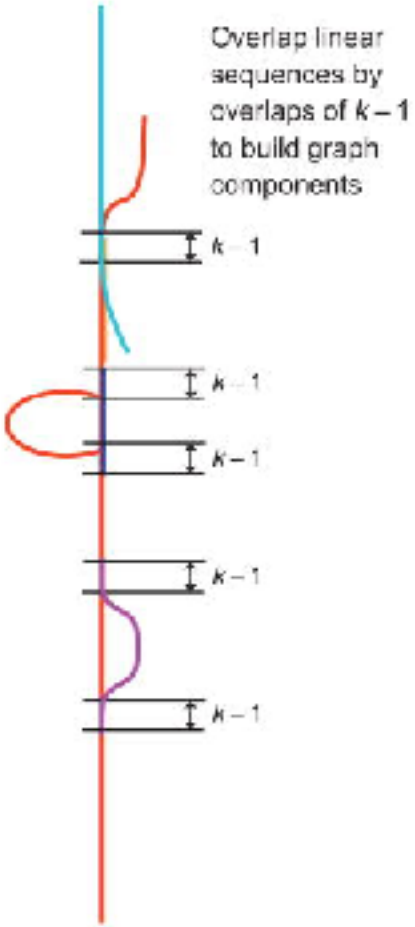


Inchworm

- Crea un catálogo de kmeros sobrelapantes
- Guarda estos kmeros y sus secuencia (no crea grafo aún)
- Toma el kmero más abundante y lo usa como semilla
- Extiende el extremo 3’ guiado por cobertura
- Si hay un empate, busca de forma recursiva las opciones para identificar kmeros que provean la mayor cobertura cumulativa
- Se realizan extensiones hasta que ya no hay más kmeros compatibles
- Se extiende después el extremo 5’
- Reporta el contig más largo y remueve los kmeros usados del catálogo
- Reinicia este proceso con una nueva semilla
- Para cuando ya no hay más kmeros en el catálogo

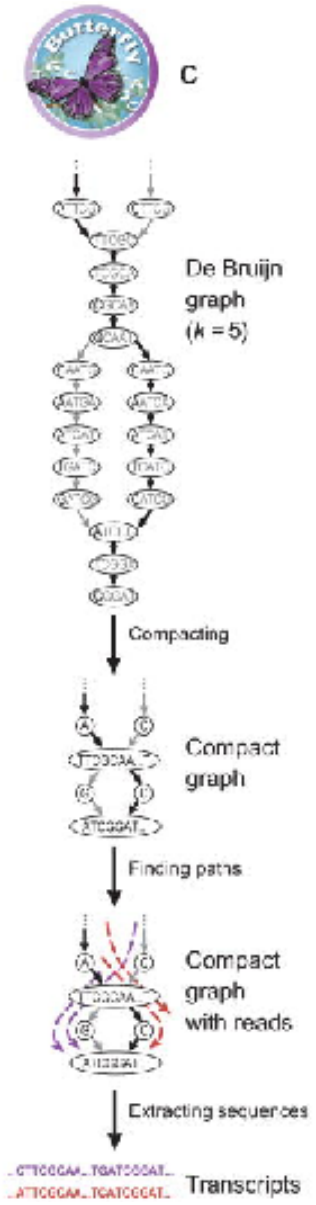
Chrysalis

b



- Inchworm no puede reconstruir isoformas
- Toma los contigs generados por inchworm que no tienen kmeros completos compatibles en sus extremos
- Explora kmeros parciales (k-1) para reagrupar contigs relacionados
- Si existe soporte Chrysalis une contigs generados por Inchworm
- Finalmente construye un grafo de Bruijn para cada grupo
- El grafo se ramifica en sitios de variación
- Esto resulta en un grafo por gen

Butterfly



- Funciona en cada grafo independiente de manera paralela
- Colapsa estructuras no ramificadas del grafo
- Embebe las lecturas originales dentro del grafo rastreando el camino de cada lectura y verificando la congruencia de lecturas en pares
- Emite transcritos ensamblados completos incluyendo isoformas y paralogos

Práctica - ensamblando un transcriptoma usando Trinity

https://liz-fernandez.github.io/PBP_transcriptomics_2023/