SPAdes

A Genome Assembly Algorithm Designed for Single-Cell Sequencing

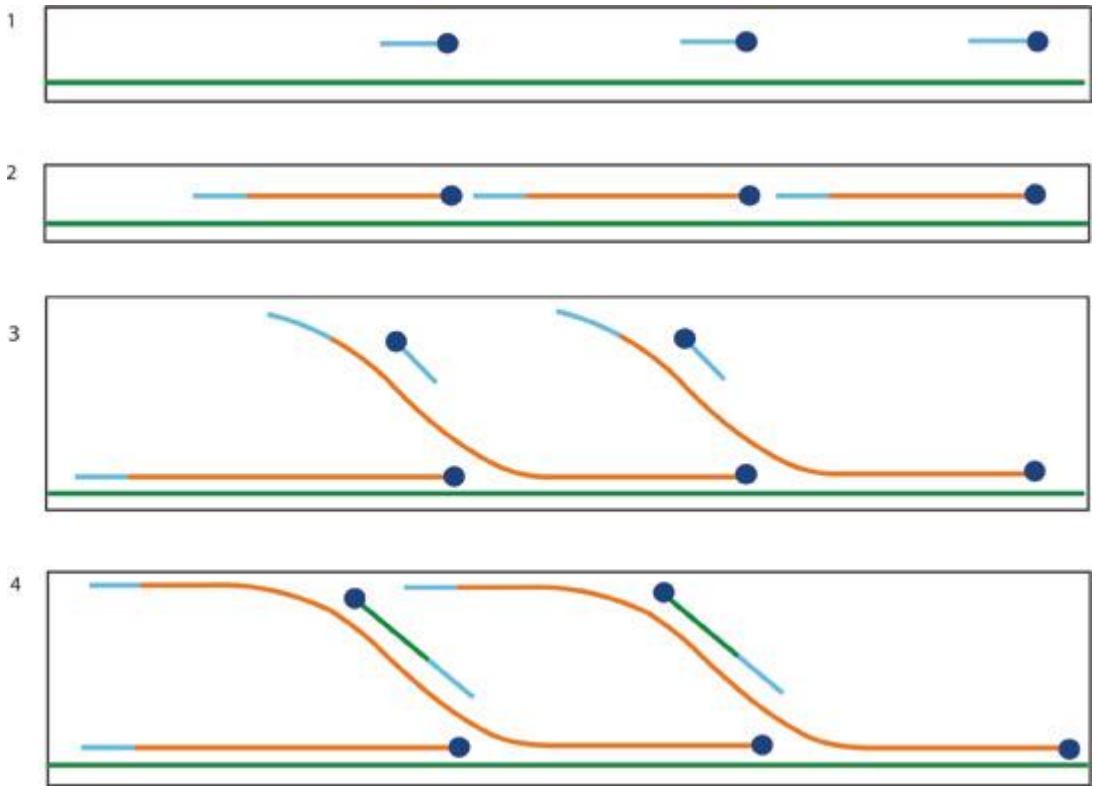
Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

What is Single Cell Sequencing?

- Single-Cell Sequencing: the process of assembling a genome from DNA extracted from a **single cell** (or otherwise a very small amount of DNA)
- Traditional sequencing approaches – DNA is provided by a colony of cells.
- However, not always possible or ideal:
  - Many bacterial species cannot be cultured in laboratory environments
  - Population averaging makes it difficult to track how cells mutate (e.g. in cancer)
- Because initial DNA quantity is small, different techniques need to be used

  (Multiple Displacement Amplification)
- These techniques introduce complications into data

Multiple Displacement Amplification



1. Random hexamer (6 bp) primers attach to DNA
2. A high fidelity DNA polymerase extends the copy
3. When the DNA polymerase hits the site of another primer, the strand is displaced
4. The displaced strand then acts as a template for the process to repeat
5. Multiple repetitions result in a highly branched structure of DNA that can be cleaved with nucleases to give many overlapping reads from a single strand of DNA.

Figure from Spits C, LeCaignec C, DeRycke M, et al. Whole-genome multiple displacement amplification from single cells. Nat Protoc. 2006;1(4):1965-70.

Consequences of Multiple Displacement Amplification

- Can introduce amplification bias because primers bind randomly (some areas may be represented by more reads by several orders of magnitude than others)
- Introduces **chimeric reads**: distant sequence fragments in the genome are concatenated
- Introduces **chimeric read-pairs**: read-pairs with abnormal insert sizes or incorrect orientations
- These issues need to be corrected by assembly algorithm

Overview of SPAdes

- Stage 1 – Assembly Graph Construction
- Stage 2 – *k*-bimer Adjustment
- Stage 3 – Paired Assembly Graph Construction
- Stage 4 – Output of Contigs of Paired Assembly Graph

Step 1: Assembly Graph Construction – de Bruijn Graphs

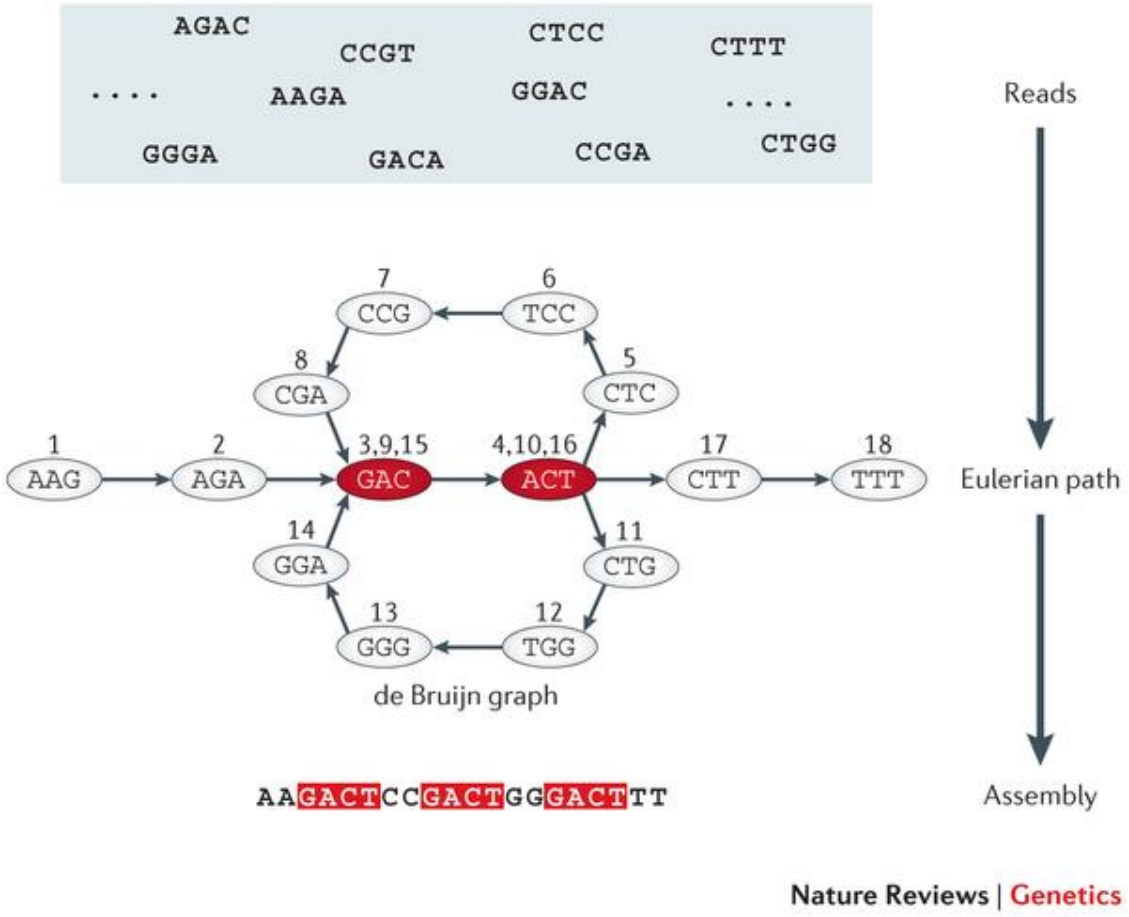
De Bruijn Graphs are used by most fragment assembly algorithms:

- Define a ***k*-mer** as a string of fixed length "*k*"; let READS be set of strings from readsFor each *k*-mer *a* that occurs as a substring of a string in READS, create vertices *u, v* in G such that *u* is labeled with prefix(*a*) and v with suffix(*a*); create a directed edge $u \rightarrow v$
  - (In this construction, edges are labeled as k-mers and vertices as (k-1)-mers)
- Glue together vertices in G if they have the same label.

Terminology:

- **Coverage of an edge** is the number of reads that contain the edge's *k*-mer.
- A **hub** in a de Bruijn graph has outdegree $\neq 1$ or indegree $\neq 1$; **h-edges** start from hubs.
- An **h-path** is a path where start and end vertices are hubs, and immediate edges are not hubs.
- If a is the i-th edge in an h-path, then **OFFSET**(a) = i.

Example of a de Bruijn Graph



In this graph, k = 4.

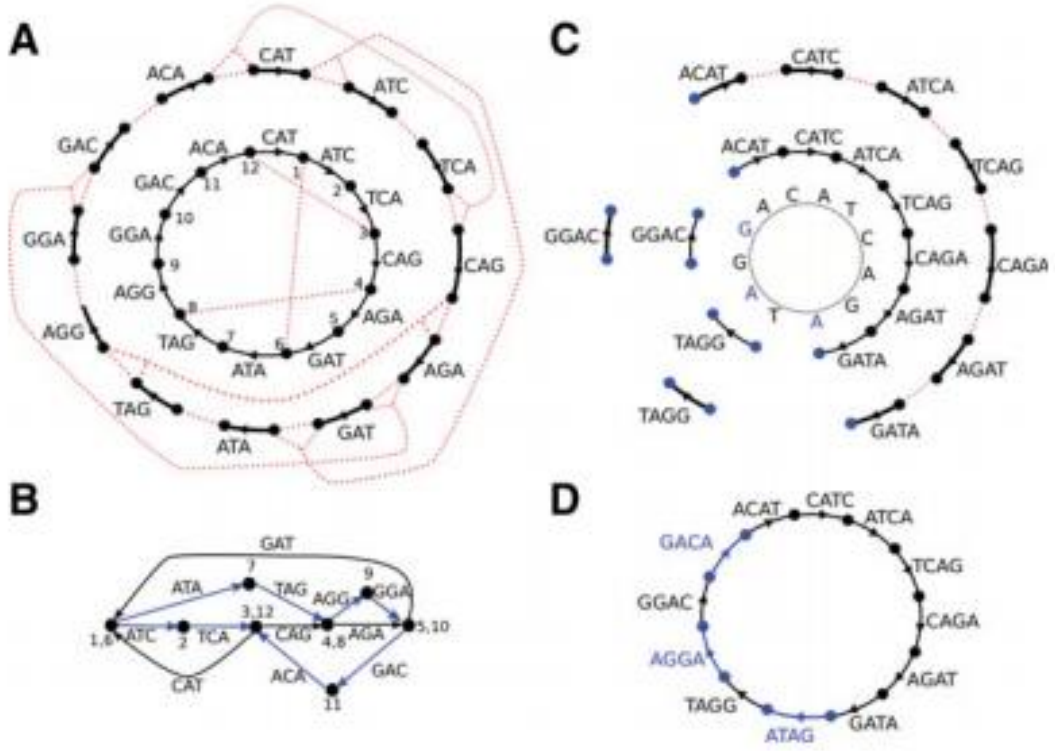
The assembly results from finding a Euler path.

The hubs in this graph are AAG, GAC, ACT and CTT, and the h-edges in this graph are the edges that start at the hubs.

Figure from Berger B, Peng J, Singh M. Computational solutions for omics data. Nat Rev Genet. 2013;14(5):333-46.

Multisized de Bruijn Graphs

- The choice of *k* is important to the construction of a de Bruijn graph; smaller *k* results in more tangled graphs as more repeats will be glued, whereas larger *k* may not adequately detect overlaps, leading to fragmented graphs.
  - Smaller *k* works better with low-coverage regions
  - Larger *k* works better with high-coverage regions
- MDA reads can vary drastically in coverage from region-to-region
- Therefore, SPAdes implements multi-sized de Bruijn graphs that allow for variable *k*.

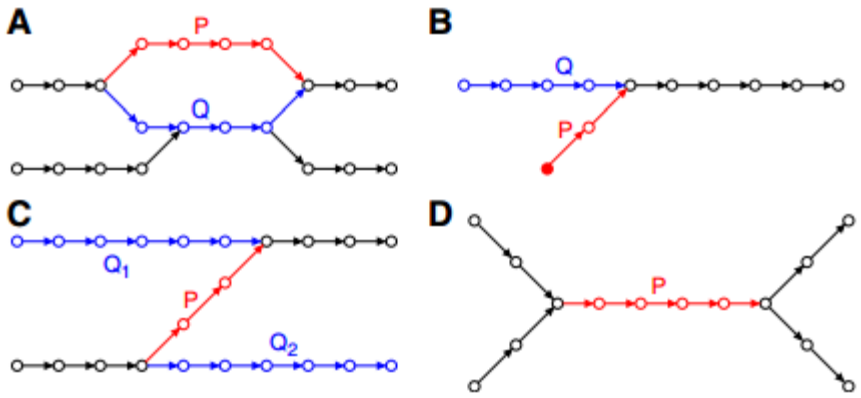


The 3-mer de Bruijin graph of these set of reads is overly tangled (A and B), but the 4-mer is fragmented. A 3,4-mer graph is better at finding an alignment.

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

Detection of Errors in de Bruijn Graphs

SPAdes looks for several types of graph topography structures within the de Bruijn graph that result from errors in reads:



- Miscalled bases or indels may cause **bulges** (A)
- Errors near end of reads may cause **tips** (B)
- Chimeric reads may cause erroneous connections (C)

Furthermore, low quality reads may not map to the genome and result in **isolated h-paths.** The goal is to preserve h-paths arising from repeats (D) while pruning low-coverage h-paths.

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

Correcting Errors in de Bruijn Graphs

- To handle isolated h-paths, tips and erroneous connections, SPAdes removes h-paths with low coverage of reads or low length
  - To do this, SPAdes implements a gradual h-path removal strategy
  - For removal of bulges and erroneous connections, SPAdes iterates through all h-paths in increasing order of coverage
  - For removal of tips, SPAdes iterates through all h-paths in increasing order of length
  - To preserve h-paths arising from repeats, SPAdes only deletes h-paths if its start vertix has at least 2 outgoing edges, and its end vertex has at least two in-coming vertices
  - Lists are updated as h-paths are deleted
- Bulges are removed; but because bulges may still contain information about assemblies, SPAdes records information about removed edges from bulges before discarding them.
  - Any removed bulges have their edges maintained in a data structure that can be back-tracked when mapping reads to the assembly graphs.

Step 2:Using Read-Pairs

- Read-pairs are frequently used in sequencing; they are two reads with a known distance between them.
  - Distance information helps in resolving structural rearrangements such as deletions or inversions.
- A ***k*-bimer** is a triple $(a|b, d)$, where a and b are k-mers, and d is the estimated distance between a and b in the genome
- SPAdes first extracts k-bimers from read-pairs – originally have an inexact estimate of distance
- SPAdes then conducts k-bimer adjustment – transforms original k-bimers into a set of k-bimers with exact or near-exact distance estimates

k-bimer Adjustment

- B-transformation: read-pairs (Bireads) are transformed into *k*-bimers.
  - For pair of reads r_1 and r_2 that are separated by distance d, a k-bimer $(a_1|a_2, d - i_1 + i_2)$ is formed where a_1, a_2 are k-mers from r_1, r_2 and i_1, i_2 are the starting positions of a_1, a_2 in r_1, r_2
  - Each k-bimer defines a pair of edges in the de Bruijn graph previous constructed (recall that an edge is a k-mer in the Bruijn graph), called a "**biedge**"

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

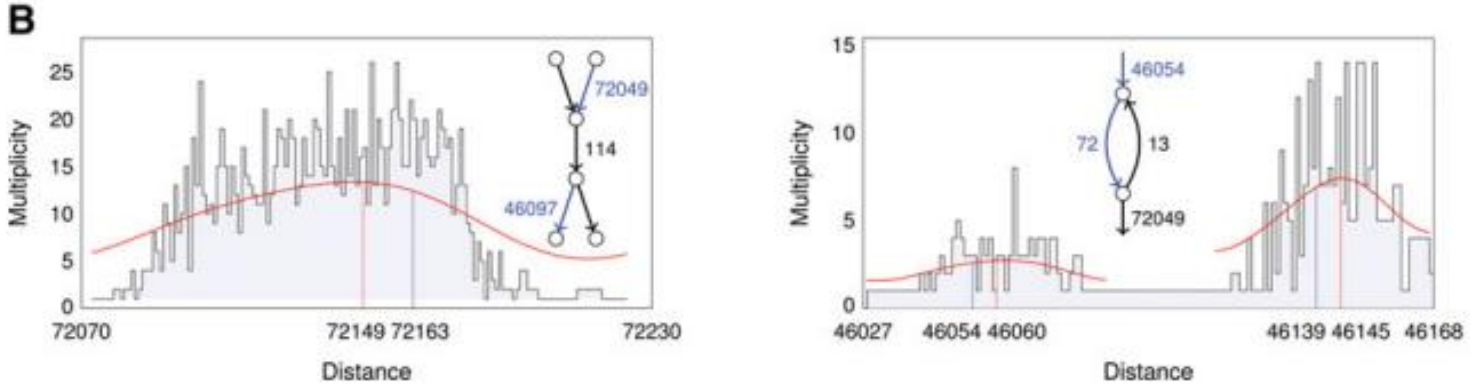
k-bimer Adjustment

- H-transformation: biedges are transformed into h-biedges
  - Each bi-edge that resides on an h-path path(A) and path(B) provides a distance estimate for h-edges A and B. We can construct an h-biedge from this information:
  - Given a bi-edge (a|b, d), an h-biedge H(a|b, d) = (H-EDGE(a)|H-EDGE(b), D), where H-EDGE(x) refers to the edge of the k-mer in the de Bruijn graph, and D = d + offset(a) – offset(b)

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

k-mer Adjustment (cont)

- A-transformation: transforms h-biedge histograms into single/small numbers of h-biedges
  - A h-biedge histogram is a multi-set of h-biedges (A|B, *) that have the same fixed h-edges A and B, and variable distance estimates * between A and B
  - A fast Fourier transform is performed on the h-biedge histogram, with the peaks of the resultant histogram serving as estimate(s) of distances between A and B.



- E-transformation: the adjusted h-biedges are used to recalculate distances of biedges

Figures from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

Step 3: Paired Assembly Graph Construction

- Now that it has both the de Bruijn graph G and the set of adjusted biedges, SPAdes seeks to find Eulerian cycle in the de Bruijn graph that is consistent with all biedges.
  - A cycle C is consistent with a biedge (a|b, d) if there are instances of a and b at a distance d in C.
  - For a set of biedges BE, a cycle C is BE-consistent if it is consistent with all biedges in BE.

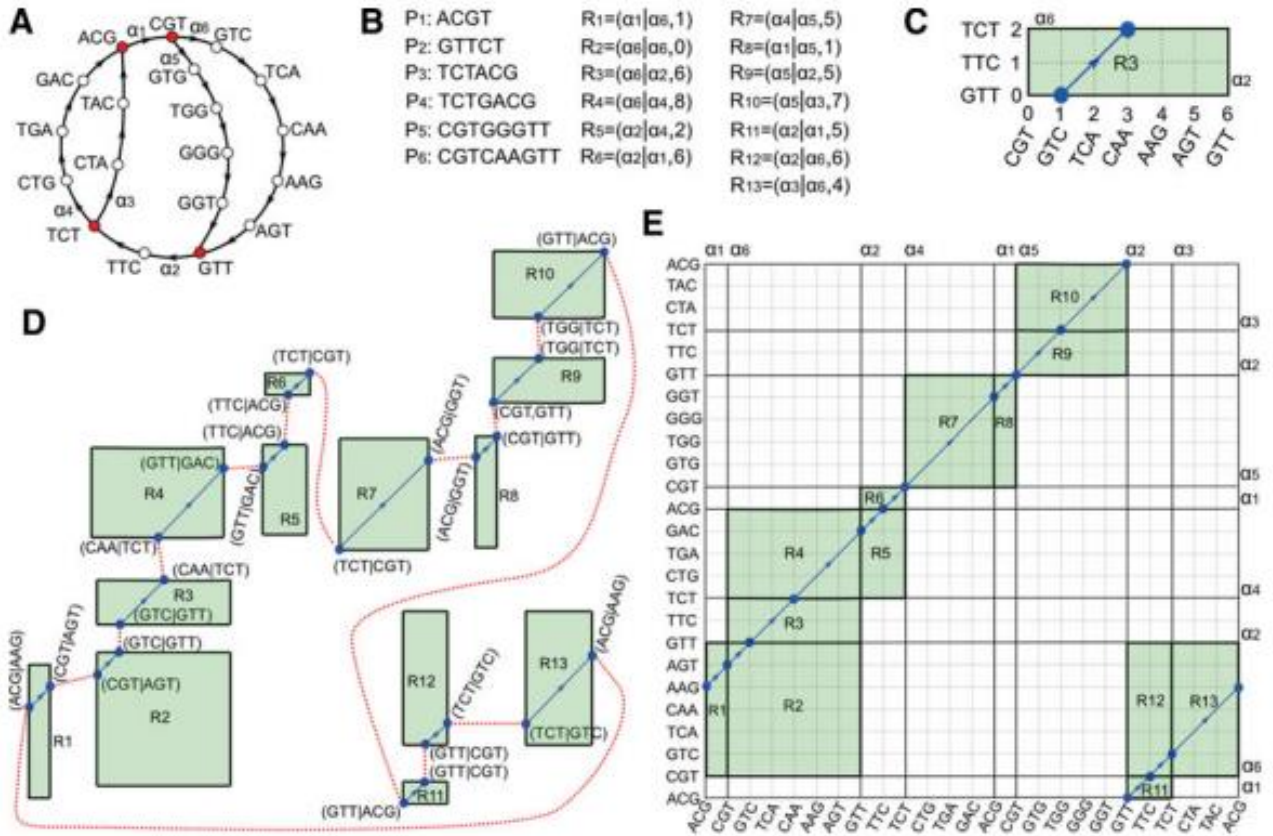
A de Bruijn Approach to the Set of Biedges

- Consider a construction of the de Bruijn graph from the set of biedges (the biedge graph):
  - For each biedge (*a*|*b*, *d*) in BE, create vertices *u*, *v* in G such that *u* is labeled with START(*a*|*b*, *d*) and v with END(*a*|*b*, *d*); create a directed edge $u \rightarrow v$ and label with (a|b, d)
  - Glue together vertices in G if they have the same label.
- H-paths in this biedge graph "spell out" paths in G that are shared by BE-consistent cycles
- You can pair the biedge graph with the assembly graph to find BE-consistent cycles in the assembly graph

H-Biedge Construction

- However, because the number of biedges is much greater than the number of h-biegdes, SPAdes uses h-biedges to mimic construction of the bi-edge graph to save time and space.
- Let (A|B, D) be a h-biedge; FIRST(A|B, D) is the biedge with the minimal offset amongst all biedges in E(A|B, D), and LAST(A|B, D) is the biedge with maximal offset.
- Let HBE be a set of h-biedges; we construct the H-biedge graph:
  - For each h-biedge *(A|B, D)* in HBE, create vertices *u, v* in G such that *u* is labeled with START(FIRST(*A|B, D)*) and v with END(LAST(*A|B, D)*); create a directed edge $u \rightarrow v$ and label with (A|B, D)
  - Glue together vertices in G if they have the same label.
- This works because in the biedge graph, the h-biedges do not share edges and thus partition the biedge graph into edge-disjoint subpaths; the h-biedge graph effectively substitutes each subpath with a single edge.

H-biedge Graph Construction Example



- A – the assembly graph has six h-paths (P_1 to P_6)
- B – a list of all h-biedges (a_1 to a_6) constructed from all bi-reads, which were paired with a separation of d = 5.
- C – the h-path between the h-biedge ($a_2|a_6$, 6) can be represented as a rectangle diagram with edges P_2 and P_6 and a 45 degree line segment y = x + (d – D) which defines all bivertices with genomic distance d in the rectangle
- D – Vertices from different rectangle diagrams are glued together
- E – rectangles are then assembled onto a grid, which yields the path through the genome

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

Step 4 – Output of Contigs

- Tracing outputing through the genome in paired assembly graph gives us contig assemblies for the genome!

Overall Benchmarks

TABLE 1. COMPARISON OF ASSEMBLIES FOR SINGLE-CELL (ECOLI-SC) AND STANDARD (ECOLI-MC) DATASETS

Assembler[a]	*# contigs*	*N50 (bp)*	*Largest (bp)*[b]	*Total (bp)*[c]	*Covered (%)*[d]	*MA*[e]	*MM*[f]	*CG*[g]
Single-cell *E. coli* (ECOLI-SC)								
EULER-SR	1344	26662	126616	4369634	87.8	21	11.0	3457
SOAPdenovo	1240	18468	87533	4237595	82.5	13	99.5	3059
Velvet	**428**	22648	132865	3533351	75.8	2	**1.9**	3117
Velvet-SC	872	19791	121367	4589603	93.8	2	**1.9**	3654
E+ V−SC	501	32051	132865	4570583	93.8	2	6.7	3809
SPAdes-single	1164	42492	166117	4781576	**96.1**	**1**	6.2	3888
SPAdes	1024	**49623**	**177944**	4790509	**96.1**	**1**	5.2	**3911**
Normal multicell sample of *E. coli* (ECOLI-MC)								
EULER-SR	295	**110153**	221409	4598020	99.5	10	5.2	4232
IDBA	**191**	50818	164392	4566786	99.5	4	1.0	4201
SOAPdenovo	192	62512	172567	4529677	97.7	1	26.1	4141
Velvet	198	78602	196677	4570131	**99.9**	4	1.2	4223
Velvet-SC	350	52522	166115	4571760	**99.9**	**0**	1.3	4165
E+V-SC	339	54856	166115	4571406	**99.9**	**0**	2.9	4172
SPAdes-single	445	59666	166117	4578486	**99.9**	**0**	**0.7**	4246
SPAdes	195	86590	**222950**	4608505	**99.9**	2	3.7	**4268**

Figure from Bankevich A, Nurk S, Antipov D, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455-77.

Explaining the Improvements

- Multi-sized de Bruijn graph employs different values of k to match coverage of reads.
- Clustering allows for accurate estimation of distances
- Implementation of Rectangle Graphs approach
- Implementation of corrections to assembly graph are novel:
  - Some other assemblers use bulge removal, but sometimes bulges contain valuable information – SPAdes retains information from bulges in a data structure
  - Gradual removal of h-paths with smallest coverage/length as opposed to the deletion of h-paths below a certain threshold that other assemblers typically do

Questions?