On Algorithmic Complexity of Biomolecular Sequence Assembly Problem

Giuseppe Narzisi Ph.D.

Bioinformatics Scientist
New York Genome Center
email: gnarzisi@nygenome.org

(in collaboration with *Bud Mishra* and *Michael C. Schatz*)

1st International Conference on Algorithms for Computational Biology
Tarragona, Spain, July 1-3, 2014

Highlights

- No new theoretical results. Sorry!
- Overview of the most popular formulations over the last 20 years.
- Similarity and differences among paradigms.
- Examples of logically consistent solutions which are intractable/unfeasible in the context of biology.

Material valuable to theoreticians as they develop new formulations as well as to developers of new pipelines and algorithms.

Highlights

- No new theoretical results. Sorry!
- Overview of the most popular formulations over the last 20 years.
- Similarity and differences among paradigms.
- Examples of logically consistent solutions which are intractable/unfeasible in the context of biology.

Material valuable to theoreticians as they develop new formulations as well as to developers of new pipelines and algorithms.

Highlights

- No new theoretical results. Sorry!
- Overview of the most popular formulations over the last 20 years.
- Similarity and differences among paradigms.
- Examples of logically consistent solutions which are intractable/unfeasible in the context of biology.

Material valuable to theoreticians as they develop new formulations as well as to developers of new pipelines and algorithms.

Highlights

- No new theoretical results. Sorry!
- Overview of the most popular formulations over the last 20 years.
- Similarity and differences among paradigms.
- Examples of logically consistent solutions which are intractable/unfeasible in the context of biology.

Material valuable to theoreticians as they develop new formulations as well as to developers of new pipelines and algorithms.

Highlights

- No new theoretical results. Sorry!
- Overview of the most popular formulations over the last 20 years.
- Similarity and differences among paradigms.
- Examples of logically consistent solutions which are intractable/unfeasible in the context of biology.

Material valuable to theoreticians as they develop new formulations as well as to developers of new pipelines and algorithms.

Outline

Outline

Outline

DNA sequencing

$\Downarrow$

600 GBp in 9 Days (HiSeq 2000) | **120GB** in 27 hrs (HiSeq 2500)

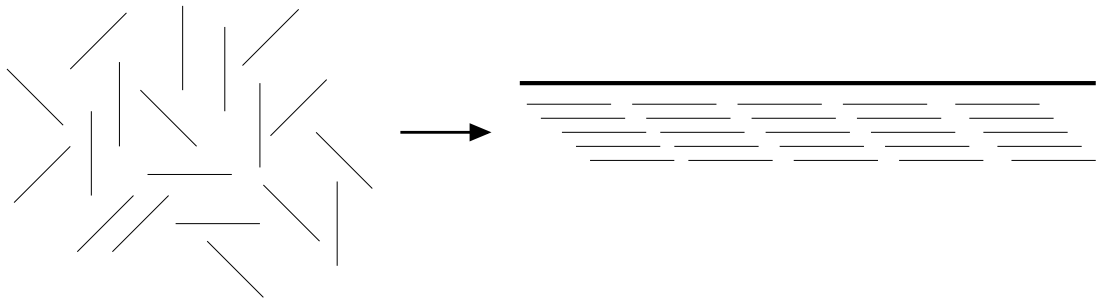
$\approx \geq$**100x coverage** of a human genome of **100Bp sequence reads**

$\Downarrow$

No human halpotypic genome assembly yet

Shotgun sequence assembly

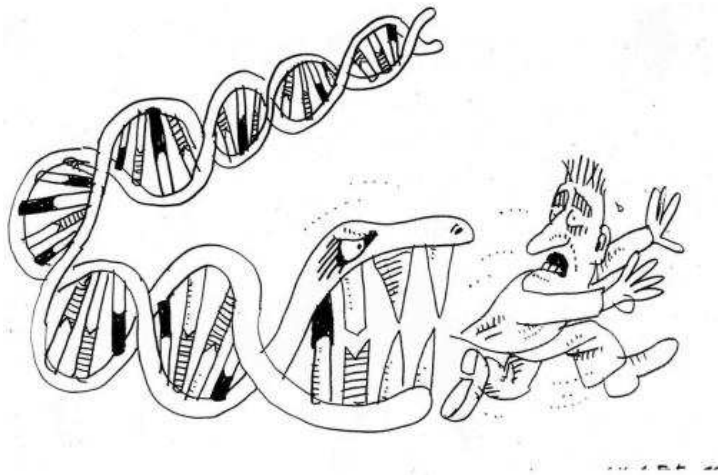
- DNA sequence is sheared into a large number of small fragments.



- **Assume**: If two sequence reads share the same string of letters (overlap), then they might have originated from the same genomic location.
- **Goal**: Join the sequences together using a computer program called assembler (similar to solving a jigsaw puzzle).
- Use long-range data to resolve complex genomic structures.

Why is de-novo sequence assembly so difficult?

1. $\mathcal{NP}$-complete: natural reduction to the *Shortest Superstring Problem* (easy for totally random DNA sequences).
2. Genomic structures: repeated regions, rearrangements, segmental duplications etc.
3. Sequencing-Technology Dependent: algorithms must change to accommodate changes to read-length or nature and availability of long-range information.



The Sense of the Approximation

A wicked problem in search for a correct solution

Definition (Wicked Problem)

A **wicked** problem is a problem that is difficult or impossible to solve because of *incomplete*, *contradictory*, and *changing* requirements that are often difficult to recognize.

[Rittel and Webber: Dilemmas in a general theory of planning. Policy Sciences (1973)]

Incomplete, contradictory, changing requirements = genome structure

⇓

Not complete and biologically correct mathematical formulation!

⇓

Difficult to have a *sense of the approximation* of the sequence relative to the true sequence as they are being assembled

The Sense of the Approximation

A wicked problem in search for a correct solution

Definition (Wicked Problem)

A **wicked** problem is a problem that is difficult or impossible to solve because of *incomplete*, *contradictory*, and *changing* requirements that are often difficult to recognize.

[Rittel and Webber: Dilemmas in a general theory of planning. Policy Sciences (1973)]

Incomplete, contradictory, changing requirements = genome structure

$\Downarrow$

Not complete and biologically correct mathematical formulation!

$\Downarrow$

Difficult to have a *sense of the approximation* of the sequence relative to the true sequence as they are being assembled

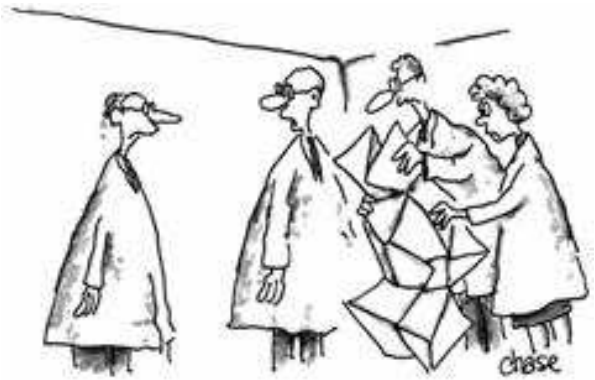
Genome Sequencing – History & Accuracy

Did we solve the problem?

- **1995**: *Haemophilus Influenzae* - 1.8 Mbp, $\sim$ 30h.
- **2001**: 1st Human Genome draft - 3 billion bp (genotypic), cost: $3 billion!.
- **2014**: Human Genome sequencing cost down to $1000.

How well did we do?

- High rates of misassembly.
  [Semple, *Bioinformatics for Geneticists*, 2003]
- "Revolution Postponed: Why the Human Genome Project Has Been Disappointing"
  [Stephen S. Hall, *Scientific American*, 2010]
- Need for Quality Assessment! $\Rightarrow$ Assemblathons (but only very recently, starting in 2011).



Why did we not try to do better?

- "Since the problem is $\mathcal{NP}$-hard (shortest superstring), any efficient reconstruction procedure must resort to heuristics." [Kececioglu and Myers, *Algorithmica*, 1995].

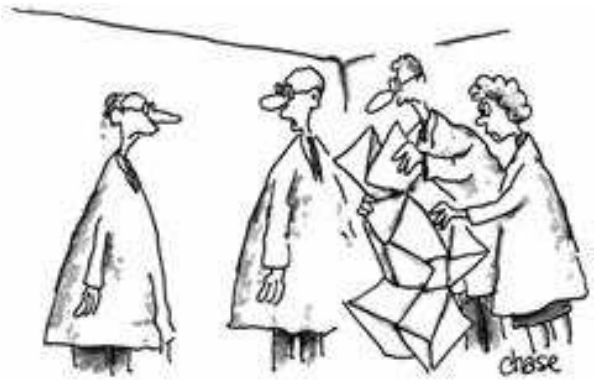
Genome Sequencing – History & Accuracy

Did we solve the problem?

- **1995**: *Haemophilus Influenzae* - 1.8 Mbp, $\sim$ 30h.
- **2001**: 1st Human Genome draft - 3 billion bp (genotypic), cost: $3 billion!.
- **2014**: Human Genome sequencing cost down to $1000.

How well did we do?

- High rates of misassembly.
  [Semple, *Bioinformatics for Geneticists*, 2003]
- "Revolution Postponed: Why the Human Genome Project Has Been Disappointing"
  [Stephen S. Hall, *Scientific American*, 2010]
- Need for Quality Assessment! $\Rightarrow$ Assemblathons (but only very recently, starting in 2011).



Why did we not try to do better?

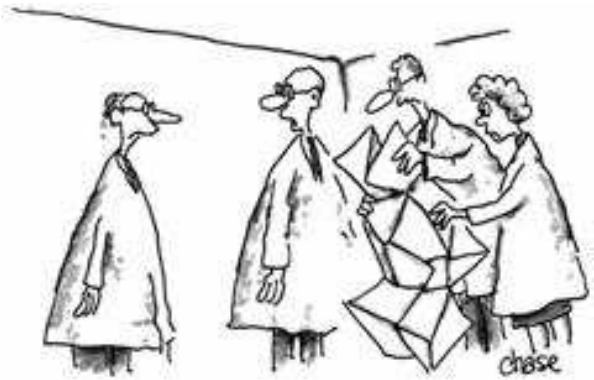
- "Since the problem is $\mathcal{NP}$-hard (shortest superstring), any efficient reconstruction procedure must resort to heuristics." [Kececioglu and Myers, *Algorithmica*, 1995].

Genome Sequencing – History & Accuracy

Did we solve the problem?

- **1995**: *Haemophilus Influenzae* - 1.8 Mbp, $\sim$ 30h.
- **2001**: 1st Human Genome draft - 3 billion bp (genotypic), cost: $3 billion!.
- **2014**: Human Genome sequencing cost down to $1000.

How well did we do?



- High rates of misassembly.
  [Semple, *Bioinformatics for Geneticists*, 2003]
- "Revolution Postponed: Why the Human Genome Project Has Been Disappointing"
  [Stephen S. Hall, *Scientific American*, 2010]
- Need for Quality Assessment! $\Rightarrow$ Assemblathons (but only very recently, starting in 2011).

Why did we not try to do better?

- "Since the problem is $\mathcal{NP}$-hard (shortest superstring), any efficient reconstruction procedure must resort to heuristics." [Kececioglu and Myers, *Algorithmica*, 1995].

Outline

Outline

Shortest Superstring Problem (SSP)

Definition (Shortest Superstring Problem)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ find the shortest string R (reconstruction) such that $\forall i$, r_i is a substring of R.

Simple theoretical abstraction, but it yields biologically implausible solutions because:

1. It does not account for sequencing errors.
2. it does not model fragment orientation (the sequence source can be one of the two DNA strands), and
3. most importantly, it fails in the presence of *repeats*, as it encourages repeat-induced compressions.

Richard Karp's statement in 2003: *The shortest superstring problem [is] an elegant but flawed abstraction: [since it defines assembly problem as finding] a shortest string containing a set of given strings as substrings.*

Shortest Superstring Problem (SSP)

Definition (Shortest Superstring Problem)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ find the shortest string R (reconstruction) such that $\forall i$, r_i is a substring of R.

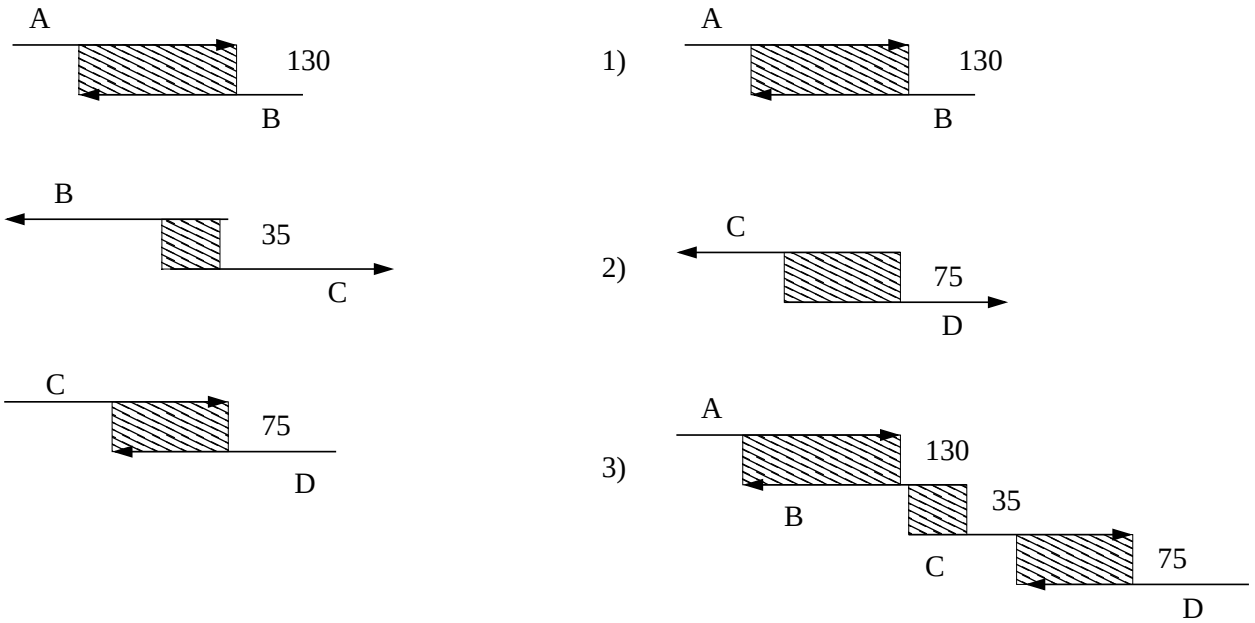
Simple theoretical abstraction, but it yields biologically implausible solutions because:

1. It does not account for sequencing errors.
2. it does not model fragment orientation (the sequence source can be one of the two DNA strands), and
3. most importantly, it fails in the presence of *repeats*, as it encourages repeat-induced compressions.

Richard Karp's statement in 2003: *The shortest superstring problem [is] an elegant but flawed abstraction: [since it defines assembly problem as finding] a shortest string containing a set of given strings as substrings*.

Greedy Strategy
(TIGR 1995, Phrap 1996, CAP3 1999)

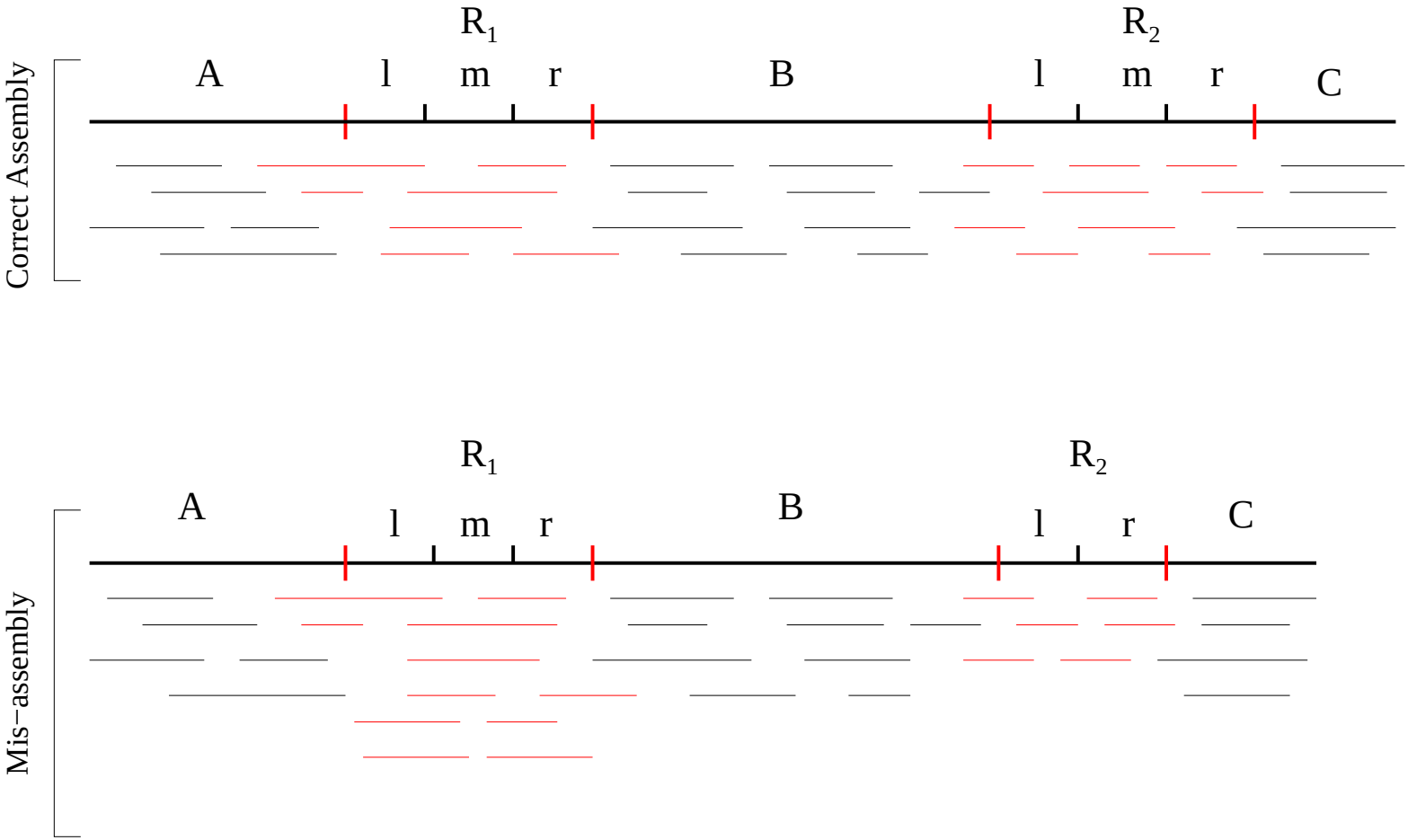
1. Pick the highest scoring overlap.
2. Merge the two fragments (add this new sequence to the pool of sequences).
3. Heuristically correct regions of the overlay in some plausible manner (whenever possible).
4. Regions that do not yield to these error-correction heuristics are abandoned as irrecoverable and shown as gaps.
5. Repeat until no more merges can be done.



The best known greedy algorithm for the SSP has an approximation factor of $2\frac{2}{3}$ [Armen, C., Stein, C. (1996)].

Repeats

- If we look for a reconstruction of minimum length, the reconstructed string can have many errors due to repeats.



Outline

String Graph

Definition (Overlap Graph)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold value k, the **overlap-graph** for S is a weighted bidirected graph $OG^k = (V, E)$ where:

- $V = S = \{r_1, r_2, \ldots, r_n\}$;
- $E = \{(r_i, r_j) : (r_i \rightleftharpoons r_j) \wedge o(r_i, r_j) \geq k, r_i, r_j \in V\}$;
- the weight of each edge (r_i, r_j) is $w(r_i, r_j) = |s_j| - o(r_i, r_j)$.

Definition (String Graph)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold value k, the **string graph** SG^k for S is obtained from the overlap graph OG^k by removing *contained* strings (strings that are substrings of other strings) and *transitively inferable* edges.

String Graph

Definition (Overlap Graph)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold value k, the **overlap-graph** for S is a weighted bidirected graph $OG^k = (V, E)$ where:

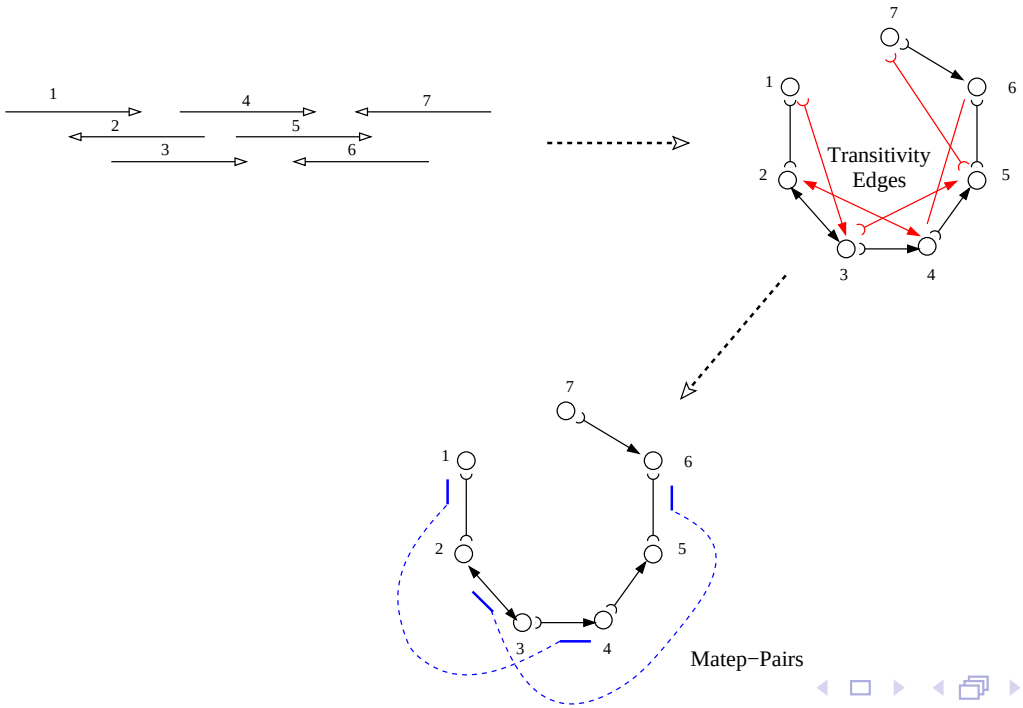
- $V = S = \{r_1, r_2, \ldots, r_n\}$;
- $E = \{(r_i, r_j) : (r_i \rightleftharpoons r_j) \wedge o(r_i, r_j) \geq k, r_i, r_j \in V\}$;
- the weight of each edge (r_i, r_j) is $w(r_i, r_j) = |s_j| - o(r_i, r_j)$.

Definition (String Graph)

Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold value k, the **string graph** SG^k for S is obtained from the overlap graph OG^k by removing *contained* strings (strings that are substrings of other strings) and *transitively inferable* edges.

Overlap-Layout-Consensus
(CELERA/CABOG 2000, Minimus 2007, SGA 2011)

- **Idea:** Construct a graph in which nodes represent reads and edges indicate overlaps.
- **Goal:** Need to solve an Hamiltonian path !
- **Strategy**:
  1. Remove contained and transitivity edges.
  2. Collapse "unique connector" overlaps (chordal subgraph with no conflicting edges).
  3. Use mate-pairs to connect and order the contigs.
- Contigs correspond to nonintersecting simple paths in the reduced graph.



Sequence Assembly Problem (SAP)

Definition (Sequence Assembly Problem)

Given a set of fragment or reads $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold k, the Sequence Assembly Problem (SAP) is the problem of finding an Hamiltonian Path in the string graph SG^k for S such that its weight is minimum.

- Special case of the **Traveling Salesman Problem** (TSP).
- Generalized Hamiltonian path (every node is visited at least once): appeal to parsimony (min weight) could compromise correctness. [Nagarajan and Pop, J. of Comp. Bio. 2009].
- Might have multiple Hamiltonian paths of minimum length.

Sequence Assembly Problem (SAP)

Definition (Sequence Assembly Problem)

Given a set of fragment or reads $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold k, the Sequence Assembly Problem (*SAP*) is the problem of finding an Hamiltonian Path in the string graph SG^k for S such that its weight is minimum.

- Special case of the **Traveling Salesman Problem** (TSP).
- Generalized Hamiltonian path (every node is visited at least once): appeal to parsimony (min weight) could compromise correctness. [Nagarajan and Pop, J. of Comp. Bio. 2009].
- Might have multiple Hamiltonian paths of minimum length.

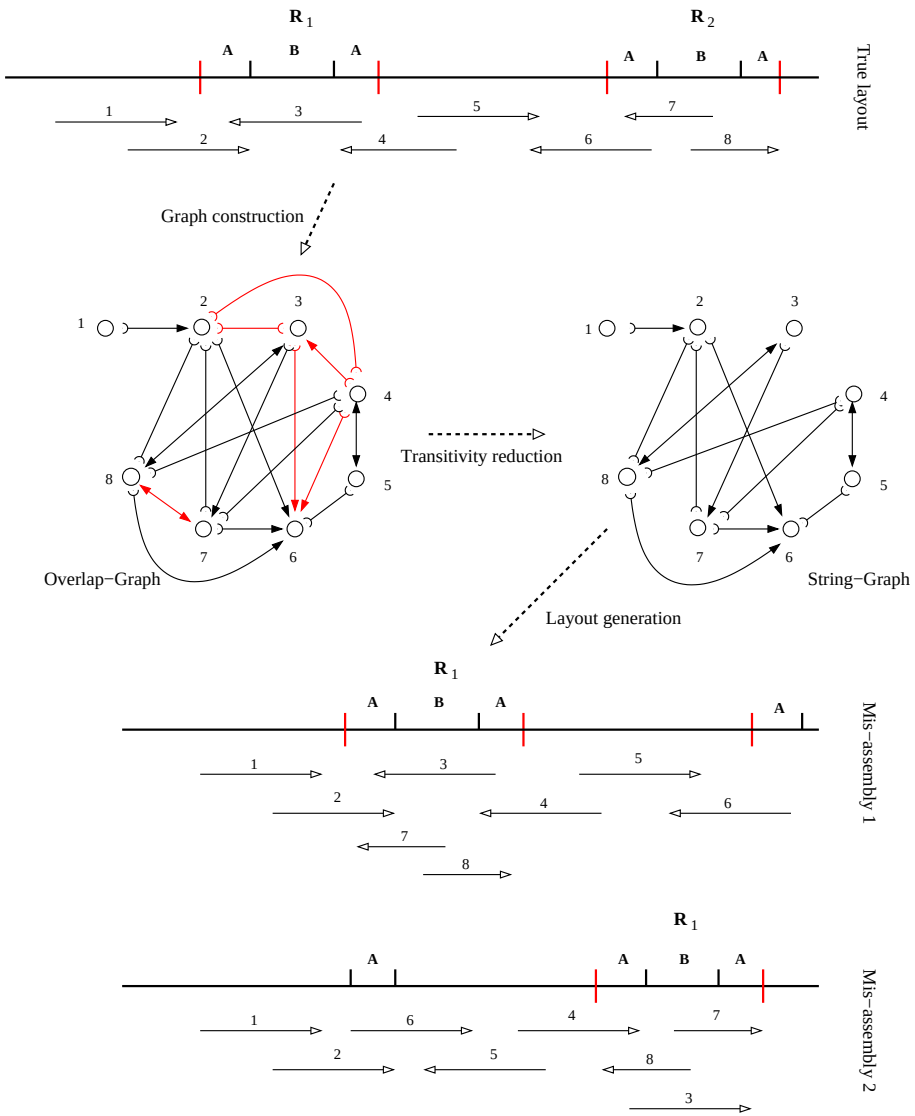
SAP complexity

- The minimum weight Hamiltonian path problem in a directed or undirected graph is $\mathcal{NP}$-complete.
- Since directed graphs are special types of bidirected graphs, we have:

Theorem

The Sequence Assembly Problem is $\mathcal{NP}$-complete.

Mis-assembly using a string graph



- The removal of the transitively inferable edges (in red) produces a string graph where every (Hamiltonian) paths through all nodes creates mis-assemblies.
- The layouts for two of these paths are shown at the bottom: the first one with compression and the second one with both compression and inversion

Outline

De Bruijn Graph

Definition (De Bruijn Graph)

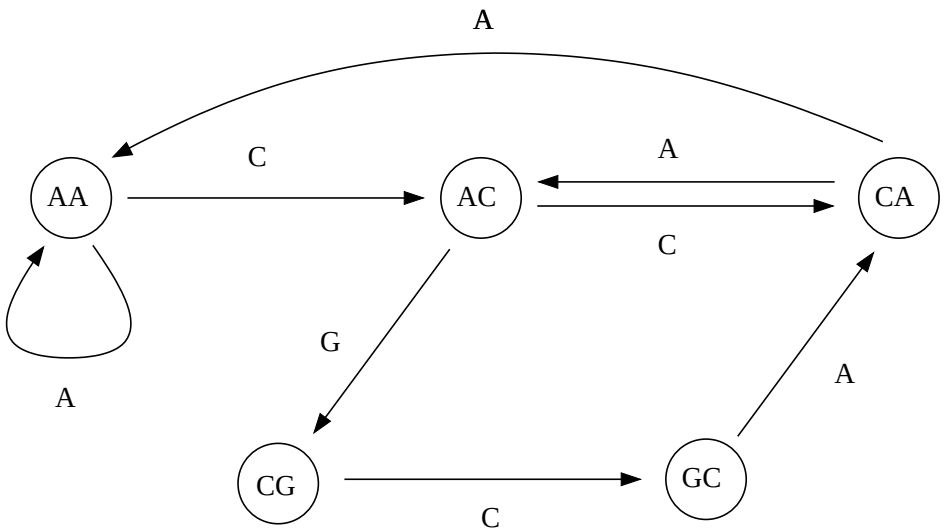
Given a set of strings $S = \{r_1, r_2, \ldots, r_n\}$ and a minimum overlap threshold value k, the de Bruijn graph for S is a directed graph $BG^k = (V, E)$ where:

- $V = \{d \in \Sigma^k \mid \exists i \text{ s.t. } d \text{ is a substring of } r_i \in S\}$;
- $E = \{(d_i, d_j) : \text{if the prefix of length } k-1 \text{ of } d_i \text{ is a suffix of } d_j\}$;

- Every read $r_i \in S$ is translated into a path composed of $(|r_i| - k)$ nodes.
- No weight associated to the edges.

Sequencing by Hybridization
(EULER 2001, Velvet 2008, SOAPdenovo 2009, ALLPATHS-LG 2011)

- **Idea**: Break the reads into overlapping k-mers (a k-mer is a substring of length k). Build a DeBruijn graph in which each edge is a k-mer and the source and destination nodes are respectively the $k-1$ prefix and $k-1$ suffix of the corresponding k-mer.
- **Goal**: find a path that uses all the edges (an Eulerian path) $\rightarrow$ linear time algorithm
- **Note**: At least one Eulerian path if no k-mer appears more than once in the genome.
- **Problem**: Errors in the data can introduce many erroneous edges !

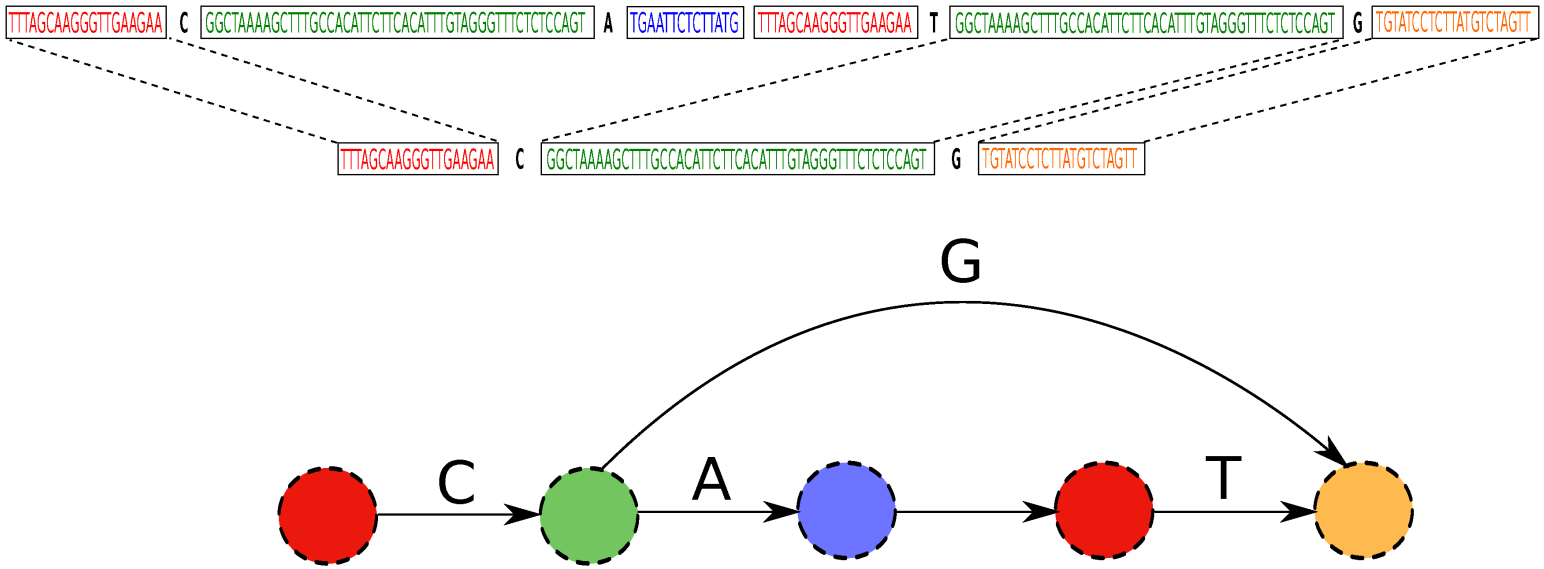


DeBruijn graph for the list $L = \{AAA, AAC, ACA, CAC, CAA, CGC, GCG\}$. The Euler path is:

$AC \rightarrow CA \rightarrow AC \rightarrow CG \rightarrow GC \rightarrow CA \rightarrow AA \rightarrow AA \rightarrow AC$

Example of false bubble in a De Bruijn graph

Near-perfect repeats can introduce artifacts in the De Bruijn graph that mislead assembly methods to make false-positive calls:



- The sequence is segmented as 19-C-49-A-14-19-T-49-G-21
- The longest exact repeat is 49bp long
- Sequence 19-C-49-G can be wrongly interpreted as a long 84bp deletion (instead of 1-base mismatch).

De Bruijn framework complexity

- Polynomial time algorithm for Eulerian paths: Hierholzer's algorithm [Mathematische Annalen 6(1), 30-32 (1873)].
- It might not represent a correct assembly of the input reads (path may not be **read-coherent**).

Definition (Superwalk Problem)

Given a set of reads $S = \{r_1, \ldots, r_n\}$ find a minimum length **superwalk** in the De Bruijn graph BG^k of S. Where a walk is called superwalk of BG^k if $\forall i$, $w(s_i)$ is a subwalk of it.

$\mathcal{NP}$-complete by reduction from the Shortest Superstring Problem. [Medvedev *et. al.* Algorithms in Bioinformatics, Springer LNCS, 2007]

Theorem

The Superwalk Problem is $\mathcal{NP}$-complete.

De Bruijn framework complexity

- Polynomial time algorithm for Eulerian paths: Hierholzer's algorithm [Mathematische Annalen 6(1), 30-32 (1873)].
- It might not represent a correct assembly of the input reads (path may not be **read-coherent**).

Definition (Superwalk Problem)

Given a set of reads $S = \{r_1, \ldots, r_n\}$ find a minimum length **superwalk** in the De Bruijn graph BG^k of S. Where a walk is called superwalk of BG^k if $\forall i$, $w(s_i)$ is a subwalk of it.

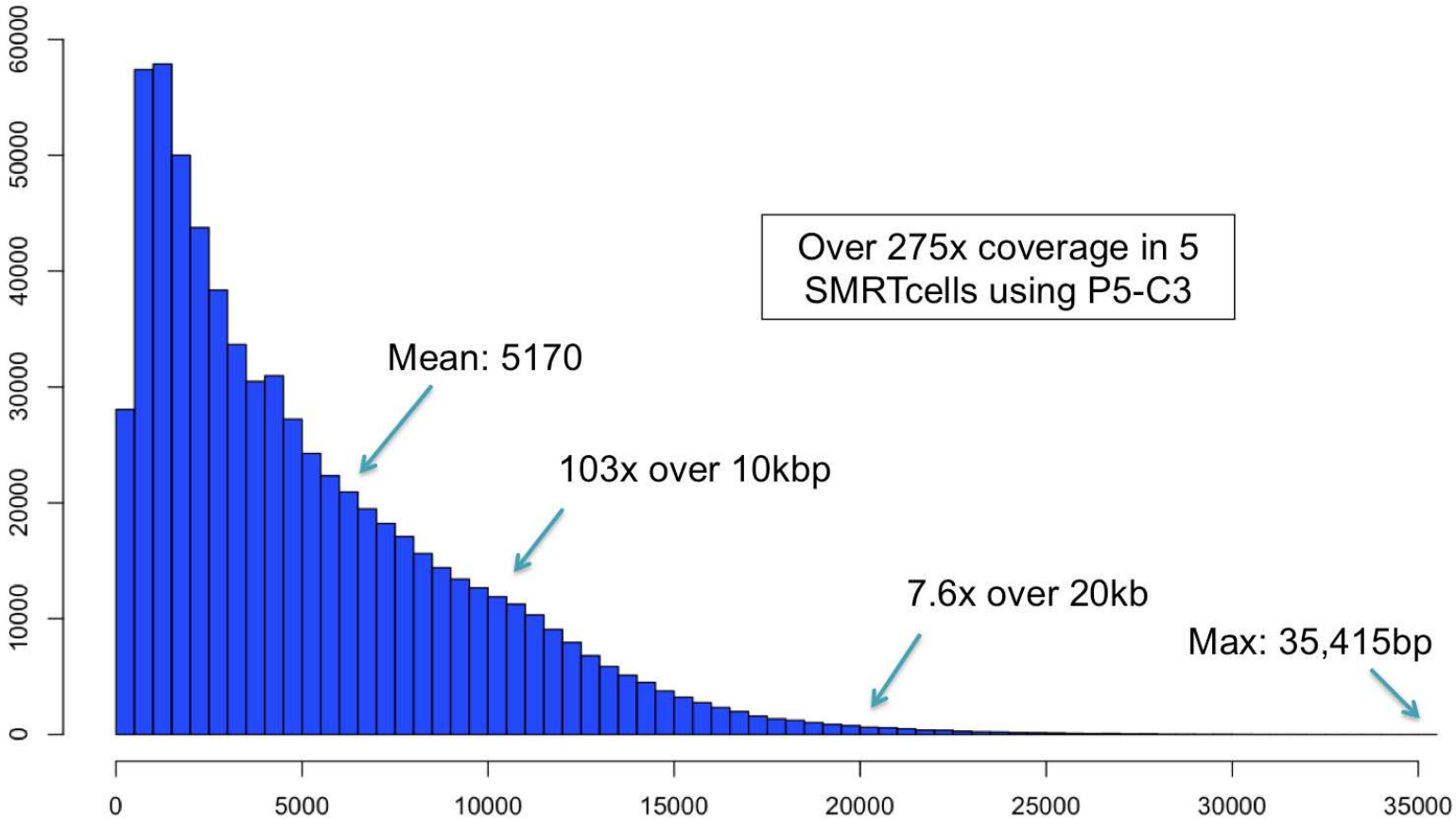
$\mathcal{NP}$-complete by reduction from the Shortest Superstring Problem. [Medvedev *et. al.* Algorithms in Bioinformatics, Springer LNCS, 2007]

Theorem

The Superwalk Problem is $\mathcal{NP}$-complete.

Not everything is lost – long read technology!

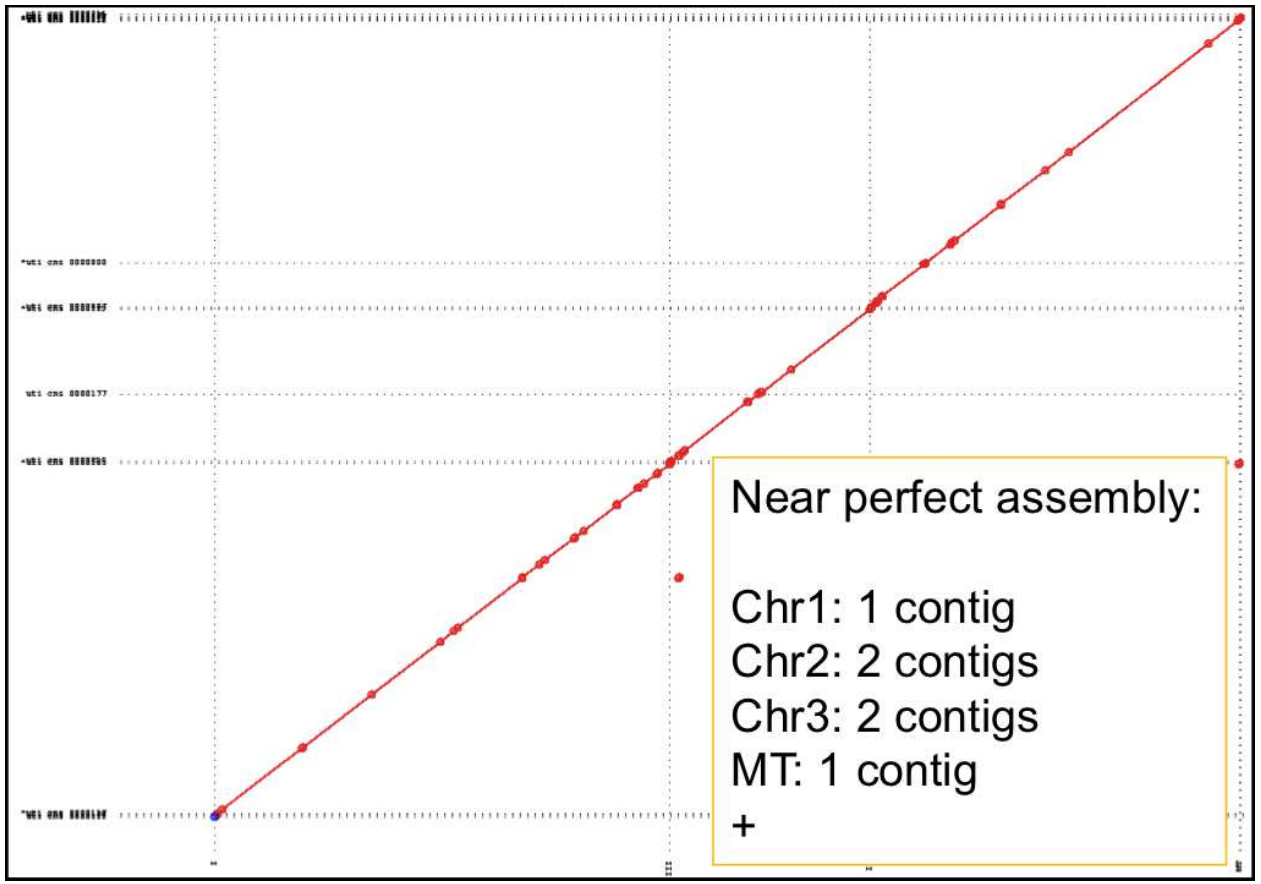
- *S. pombe dg21*: 12.6Mbp; 3 chromo + mitochondria.
- PacBio RS II sequencing at CSHL.



Not everything is lost – Long reads technology!

PacBio assembly using HGAP + Celera Assembler

- 12.7Mbp; 13 non-redundant contigs; N50: 3.83Mbp; >99.98% id



Outline

Discussion (1)

1. The process of abstraction (from biology) is powerful but can lead to unfeasible solutions (e.g., shortest superstring formulation).
2. Develop (biologically) correct formulations!
3. Better understanding and modeling of repeats is a key factors to improve accuracy, but much work needed to achieve the goal of an error-free reconstruction.
4. Urgency to model the haplotypic structure. More complexity for algorithms seeking to discover genetic mutations.

Discussion (2)

Limitations of this work:

1. **Advanced statistical and algorithmic topics**: not covered here. But the interested reader can always investigate further.
2. **Mate-pairs**. Problems with repeats still hold at that size resolution of available mate-pairs. However, experimentally designed mate-pair libraries can help resolve several classes of known repeats.
3. **De novo assembly**: for new species this information is not available and incomplete/contradictory/changing requirements can lead to biologically incorrect formulations (wicked problem).

THE END !

Michael C. Schatz (CSHL)

Bud Mishra (NYU)