Sequencing and assembly
Eivind Coward
Pål Sætrom

Topics
- Graphs
- DNA sequencing
- “2nd generation sequencing”
- Sequencing by hybridization

Graphs consist of vertices (nodes) and edges

G = G(V, E)

Connected and acyclic

Degree of a vertex \( v \) \( d(v) \):
- Edges connecting vertex \( v \)
- \( \sum d(v) = 2|E| \)

Connected vs. disconnected

Complete: edge between all pairs of vertices
- Cycle: path from a vertex back to itself
- Acyclic graph: no cycles

Variations / generalizations
- Allowing edges from a vertex to itself
- Allowing multiple edges between the same pair of vertices

Are the formulas for vertex degrees still valid?

Directed graphs

DAG: Directed acyclic graph

- Edges are now ordered pairs
- \( \text{in}(v) \sim \text{incoming edges} \)
- \( \text{out}(v) \sim \text{edges going out} \)
- \( \sum \text{in}(v) = \sum \text{out}(v) \)

Weighted graphs

- Each edge is labeled with a weight
- Typically represents a cost function that is to be minimized or maximized along a path
- Many applications for both directed and undirected graphs
Bridges in Königsberg – a famous graph

- Can you walk through the city and across the seven bridges once? (Euler cycle)
- Why?

Variant problem: Hamilton cycles

- Can you visit all vertices once and end up in starting vertex?
- Euler cycle?
- Complexity?

Hamilton cycle is NP-complete

Showing TSP is NP-complete:
1. Set cost = k on all edges
2. Add new edges (cost = ∞) to connect all nodes
3. Is there a TSP route with cost = k * |V|?

DNA sequencing – Sanger method

Chromatograms
- Output from the sequencing machine
- Converted to a DNA sequence in an automated process called basecalling (using software called basecaller)

Near beginning of sequence Near end of sequence

Sequencing the human genome?
- Max length ~1000 nts
  - Limited by electrophoresis resolution
  - "Sequencing reads"
- Sequencing strategies
  - Sequential "primer walk"?
    - ~3M (sequential) reactions
    - Foiled by repeat regions
      - No unique primers
      - Self-complementary DNA
    - ~50% of human genome is repeated
      - 1M Alu (~300 bp)
      - 200K LINE (~1500 bp)
      - Duplicated genes (25%)
  - Alternatives?
**Shotgun sequencing** the human genome

1. Randomly break genome into small (>500 nt) fragments
2. Amplify fragments
   - Place into vector (plasmid) with selection gene (antibiotic resistance)
   - Transform vector into bacteria
   - Grow bacteria under selection (antibiotic)
   - Isolate DNA
3. Sequence individual amplified fragments
4. Assemble fragments into complete genome

**Assembling DNA fragments**

- Fragments of long string
  - \( F = \{ \text{TATA, TAA, CATA, TATA, TACCA, GCAT} \} \)
- Find string that explains fragments
  - Trivial: concatenate all fragments
    - \( \text{ATAGTAACATATATATAGCAGCAT} \)
  - Better: shortest superstring
    - Shortest superstring problem (SSP)
    - Given a set of strings, find a shortest string that contains all of them

**Shortest superstring problem (SSP)**

Given a set of strings, find a shortest string that contains all of them

**Input**: Strings \( F = \{ s_1, s_2, \ldots, s_n \} \)

**Output**: A string \( s \) that contains all \( s_i \in F \) as substrings, such that \( |s| \) is minimized

**Complexity?**

**Algorithm?**

**SSP as TSP in prefix graph**

**Alternative SSP graph**

- Assume \( s_1, s_2, \ldots, s_n \) is the optimal ordering
- Length of \( s \) in terms of overlap length
  - \( H = \sum_{i=1}^{n} |\operatorname{ov}(s_i, s_{i+1})| \)
- Overlap graph
  - Fully connected, directed graph
  - \( w(s, s') = |\operatorname{ov}(s, s')| \)
- SSP \( \rightarrow \) Longest TSP in overlap graph
  - Heuristics?
Greedy approximation algorithm for SSP

GreedySSP\(F\):
1. while (|F| > 1):
   1. overlaps = |ov\((s_i, s_j)\)| for all \(s_i, s_j\) in \(F\)
   2. \((i, j) = \max_{i,j}\) (overlaps)
   3. replace \((s_i, s_j)\) with merge\((s_i, s_j)\)
2. return \(F\)
GreedySSP in practice

\[ F = \{\text{TATAA, CATAGCAT}\} \]

<table>
<thead>
<tr>
<th>CATAGCAT</th>
<th>TATAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Optimal:

\[ |\text{TATAATAGCATAA}| = 10 \]
\[ |\text{CATAGCATATAA}| = 12 \]

Approximation ratio for GreedySSP

- GreedySSP has approximation ratio of at least 2
  - Ex: \( F = \{\text{accccc, ccccc, cccct}\} \)
  - Poor choice
    \[ |\text{acccccctcccccc}| = 13 \]
  - Optimal
    \[ |\text{acccccct}| = 8 \]
  - What if \( \{\text{ac}, \text{c}^*, \text{c}^*\} \)?
- Approximation ratio of 2.5 has been shown

SSP is an idealized problem

- Errors in sequenced DNA fragments
  - Typical 1-3% error rate for reads
- Fragments from both strands
  - Must consider both orientations during assembly
- Repeated sequences
  - SSP will return too short string
    - Ex: \( GATATATAGC \)
      - \( GATA \)
      - \( ATAT \)
      - \( TATA \)
      - \( ATAT \)
      - \( TATA \)
      - \( ATAG \)
      - \( TAGC \)
      - \( GATATAGC \)

Fragment assembly in practice

1. Overlap
   - For all pairs: Find longest suffix/prefix overlap (with errors)
   - Sanger: mismatch errors most frequent
   - Algorithm?
2. Layout
   - Merge reads
   - Handle repeats
3. Consensus
   - Handle errors
   - Multiple alignment (overlapping reads)

Layout merges reads into contigs

- Greedy algorithm merge reads based on overlap
  - Build sequence (contig) bottom up
  - Repeats give too short contigs
- Hierarchical sequencing
- Mate-pair reads can handle repeats

Hierarchical shotgun sequencing

1. Break chromosome into “manageable” parts
   - Breakpoints “known”
   - Size suited for bacterial amplification
2. Shotgun sequence
3. Assemble parts into genome
Mate-pair reads can handle repeats

1. Break DNA into fragments of ~ fixed lengths
   - Ex: 200 bp, 2K bp, 20K bp
2. Sequence both ends of fragments
3. Use ends to anchor contigs

Mate-pair reads example

DNA: GATATAGC

Fragments: GATA, ATAT, TATA, ATAT, TATA, ATAG, TAGC

Mate-pairs: GAnxxxxxAG, ATxxxxxGC

Fragment-based assembly:

GATA
ATAT
TATA
ATAG
TAGC
GATATAGC

Mate-pair guided assembly:

GAxxxxxAG
GATA
ATAT
TATA
ATAG
TAGC
ATxxxxxGC
GATATATAGC

Alternatives to Sanger sequencing

- Sequencing by hybridization
- 2nd generation sequencing
  - Pyrosequencing
  - Cyclic reversible termination
  - Sequencing by ligation
- Single molecule sequencing (3rd generation sequencing)

“2nd generation” sequencing

- Also called:
  - Next Generation Sequencing (NGS)
  - Massively Parallel Sequencing
  - High Throughput Sequencing (HTS)
- Parallelized sequencing
- PCR-based DNA amplification
- Different technologies:
  - Pyrosequencing
  - Cyclic reversible termination
  - Sequencing by ligation

PCR-based DNA amplification on “chips”

Instead of bacterial growth:
1. Ligate adapters to DNA
2. Run PCR with adapter-specific primers
   - Oil/water emulsion PCR
   - Bridge PCR

Pyrosequencing – Roche/454

Characteristics
- Multiple consecutive nts incorporated
- Signal height gives length
- Errors for long polyNs
Cyclic reversible termination – Illumina/Solexa

- Reads single nucleotides

Sequencing by ligation – SOLiD/ABI

- DNA read as di-nucleotides
- Inherent error detection

“2nd generation” comparison

<table>
<thead>
<tr>
<th>Platform</th>
<th>Library preparation</th>
<th>HiD sensitivity</th>
<th>Read length (bp)</th>
<th>Overlap</th>
<th>Machine</th>
<th>Fees</th>
<th>Core</th>
<th>Biomedical applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiSeq2000/2500, Illumina TruSeq</td>
<td>TruSeq_Sequencing_Kit</td>
<td>60</td>
<td>100-150</td>
<td>High</td>
<td>HiSeq 2000/3000</td>
<td>&gt; USD 20000</td>
<td>HiD</td>
<td>Somatic mutation detection in cancer, SNP genotyping, exome sequencing, targeted gene sequencing, whole genome sequencing, whole exome sequencing, targeted gene discovery, whole genome amplicon sequencing, RNA sequencing, ChIP-sequencing, ATAC-seq, bisulfite sequencing, RNA-Seq, methylite, CHIP-sequencing, microarray, genotyping, SNP detection, metagenomics, microbial identification, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition</td>
</tr>
<tr>
<td>HiSeq2000/2500, Illumina TruSeq</td>
<td>TruSeq_Sequencing_Kit</td>
<td>60</td>
<td>100-150</td>
<td>High</td>
<td>HiSeq 2000/3000</td>
<td>&gt; USD 20000</td>
<td>HiD</td>
<td>Somatic mutation detection in cancer, SNP genotyping, exome sequencing, targeted gene sequencing, whole genome sequencing, whole exome sequencing, targeted gene discovery, whole genome amplicon sequencing, RNA sequencing, ChIP-sequencing, ATAC-seq, bisulfite sequencing, RNA-Seq, methylite, CHIP-sequencing, microarray, genotyping, SNP detection, metagenomics, microbial identification, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition, microbial species identification, microbial functional profiling, microbial whole genome sequencing, microbial metatranscriptomics, microbial metagenomics, microbial functional genomics, microbial phylogenomics, microbial evolutionary genomics, microbial population structure, microbial community composition</td>
</tr>
</tbody>
</table>

“Traditional” and “Next generation” sequencing

- Sanger sequencing: 800 bp
- Roche 454: 400 bp (indel errors)
- Solexa/Illumina: 50 bp
- ABI SOLiD: 25 bp ("color space")

- Shorter sequences and different error profiles present new challenges for alignment and assembly

The assembly problem: the overlap graph

- Read = vertex, overlap = edge
- Assembly = path visiting each vertex exactly once
- Hamiltonian path in graph theory
- Proven to be difficult (NP-hard, i.e. time grows exponentially with problem size)
- Can be solved by heuristics, used in many programs suitable for Sanger sequencing (Phrap, TIGR Assembler, CAP, ARACHNE, …)
- Next generation sequencing: big graph, harder to solve

Sequencing applications

- De novo sequencing – Assembly
- Re-sequencing
  - Mapping to reference genome
  - Variation detection
- RNA sequencing
  - Mapping
  - Expression profiling
- ChIP-seq
  - Identify DNA regions bound by regulatory proteins
  - Mapping
- Metagenomics
  - Identify species in sample
  - Assembly
  - Mapping
**de Bruijn graph**
- Vertices represent \(k\)-mers
- Edges represent \((k+1)\)-mers
- Example for the sequence ACTTACTACA with \(k = 3\):

```
ACT  CTT
CTA  TTA
TAC  ACA
```

**Sequence assembly by traversing the de Bruijn graph**
- Construct one de Bruijn graph for all reads (typically \(k\) around 20)
- An assembly corresponds to a traversal of the graph, using each edge once.
- This is called an Eulerian path.
- Surprisingly, finding an Eulerian path is easy (Fleury’s algorithm, 1883), in contrast to the Hamiltonian path problem.

**So then, assembly is easy??**
- Alas, no.
- First: While a valid assembly corresponds to a Eulerian path, the converse is not true (integrity of reads is lost)
- In practice, this seems not to be the biggest problem (for \(k\) around 20), but:
- Sequencing errors mess up the graph and cause a lot of complications.

**Next generation assembly using de Bruijn graph formulations**
- Newbler
  (454 Life Sequences’ glorious black box)
- Velvet (Flicek & Birney, 2009)
- EULER-SR (Chaisson and Pevzner, 2008)
- Several others

**Sequencing by hybridization**
1. Attach all possible \(l\)-mer probes to slide
   - 4 probes
   - Location known
   - Multiple locations for robustness
2. Add fluorescently labeled DNA sample
   - DNA hybridize complementary probes
3. Acquire image
   - Signal: \(l\)-mer present in DNA sample
   - Analyze spots (probes) to identify \(l\)-mer composition
4. Assembly
Sequencing by hybridization – Assembly
• l-mer array gives l-mer spectrum
  – l = 3, s = \text{TATACATAG}
  – Spectrum(s, l) = \{AGC, ATA, CAT, GCA, TAA, TAG, TAT\}
• Assembly problem
  – Given Spectrum(s, l), find s

Reconstruct a string from its l-mer composition
Input: A set, S, representing all l-mers from string s
Output: String s such that Spectrum(s, l) = S

Sequencing by hybridization (SBH) problem

SBH as a graph
• Overlap graph for SBH
  – Valid edges in G must have |ov(s, s)| = l – 1
  – “Simpler” than SSP graph
    S = \{AGC, ATA, ATA, CAT, GCA, TAA, TAG, TAT\}
• s is a Hamiltonian path (NP-complete)

Reformulate SBH as Euler path problem
• Let edges represent l-mers
• Let vertices represent (l-1)-mers
• s given by Euler path in graph
  S = \{AGC, ATA, ATA, CAT, GCA, TAA, TAG, TAT\}

Properties of Eulerian graphs
• Directed graph Eulerian if it contains Euler cycle
• Vertex v is balanced
  – indegree(v) = outdegree(v)
• Euler cycle
  – All vertices must be balanced
  – Number of times entered = number of times left

Constructing Euler cycle in balanced graph
1. Start in any vertex v.
2. Follow edges until no possible edges. Then we are in v again.
3. Two possibilities
   a) Path P is an Euler cycle
   b) Exist at least one vertex w in P with unvisited edges and path through w
      – P forms balanced subgraph
      – Path starting in w must form balanced subgraph
3 b) 4
**Eulerian paths**

- For the SBH problem, we need an Eulerian path, rather than a cycle.
- A vertex is called **semibalanced** if the difference between indegree and outdegree is 1.

**Theorem**: A connected graph has an Eulerian path if and only if it contains at most two semibalanced vertices and all other vertices are balanced.

The semibalanced vertices are the start and end points of the path. Construction is similar to Euler cycle.

---

**Mapping problem**

Given set of reads $R$ and reference genome $G$, identify all positions of each $r_i \in R$ in $G$.

**Input**: A set of strings $R$ and string $G$.

**Output**: A set of position sets $P$ such that $P_r = \{p_{i_1}, \ldots, p_{i_n}\}$ is the positions where $r_i$ "is found" in $G$.

**Complexity?**

**Algorithm?**

Mapping is an important task, since a reference genome is often available.