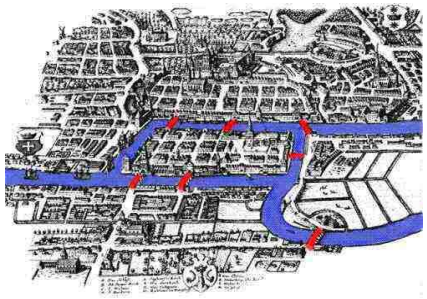
De Bruijn Graphs for DNA Sequencing (Part 1)

These slides based on:
An Introduction to Bioinformatics Algorithms (Jones and Pevzner, 2004)

The Bridges of Königsberg Problem

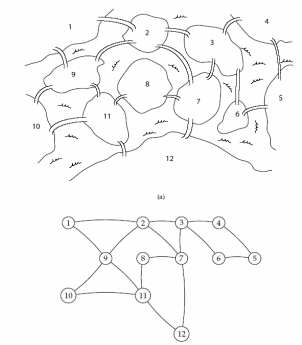
Find a tour crossing every bridge just once
Leonhard Euler, 1735



Bridges of Königsberg

Eulerian Cycle Problem

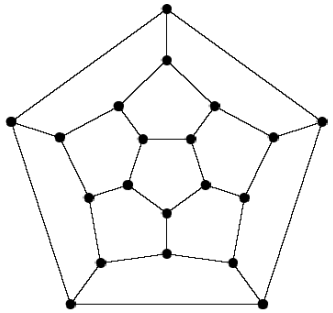
- Find a cycle that visits every ***edge*** exactly once
- Linear time



More complicated Königsberg

Hamiltonian Cycle Problem

- Find a cycle that visits every ***vertex*** exactly once
- NP-complete



Game invented by Sir William Hamilton in 1857

Shortest Superstring Problem

- Problem: Given a set of strings, find a shortest string that contains all of them
- Input: Strings $f_1, f_2, \ldots, f_n$
- Output: A string s that contains all strings $f_1, f_2, \ldots, f_n$ as substrings, such that the length of s is minimized

- **Complexity:** NP-complete
- **Note:** this formulation does not take into account sequencing errors

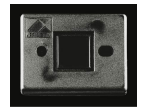
Sequencing by Hybridization (SBH): History

- **1988:** SBH suggested as an an alternative sequencing method. Nobody believed it will ever work

*First microarray prototype **(1989)***

- **1991:** Light directed polymer synthesis developed by Steve Fodor and colleagues.

*First commercial DNA microarray prototype w/16,000 features **(1994)***



- **1994:** Affymetrix develops first 64-kb DNA microarray

*500,000 features per chip **(2002)***

How SBH Works

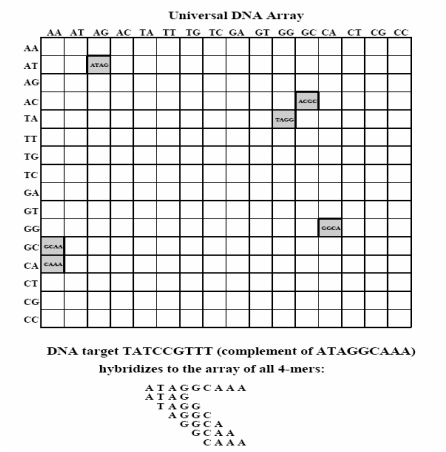
- Attach all possible DNA probes of length k to a flat surface, each probe at a distinct and known location. This is called a **DNA array**.
- Apply a solution containing fluorescently labeled DNA fragment (many many copies!) to the array.
- The DNA fragment hybridizes with those probes that are complementary to substrings of length k of the fragment.

How SBH Works (cont' d)

- Using a spectroscopic detector, determine which probes hybridize to the DNA fragment to obtain the k–mer composition of the target DNA fragment.

- Apply the combinatorial algorithm (below) to reconstruct the sequence of the target DNA fragment from the k-mer composition.

Hybridization on DNA Array



k-mer composition

- ***Spectrum(s,k)***: *unordered* multiset of all possible *(n – k + 1)* *k*-mers in a string *s* of length *n*
- The order of individual elements in *Spectrum(s,k)* does not matter (it's a set!)
- For *s* = TATGGTGC the following is *Spectrum(s,3):*

  {ATG, GGT, GTG, TAT, TGC, TGG}
- NB: for now, we are assuming that every *k*-mer occurs exactly once.

Different sequences – the same spectrum

- Different sequences may have the same spectrum:

  Spectrum(GTATCT,2)=

  Spectrum(GTCTAT,2)=

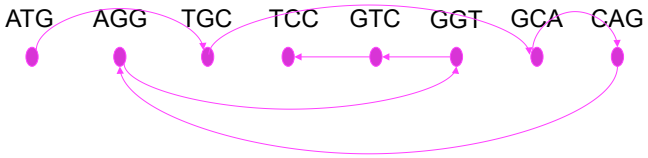
  {AT, CT, GT, TA, TC}

The SBH Problem

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

SBH: Hamiltonian Path Approach

S = { ATG AGG TGC TCC GTC GGT GCA CAG }



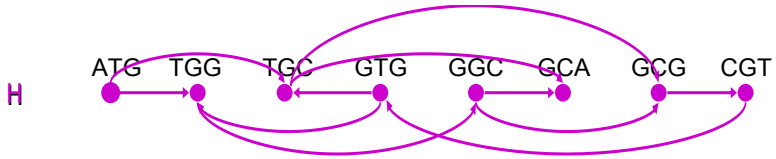
ATGCAGGTCC

Path visited every VERTEX once

SBH: Hamiltonian Path Approach

A more complicated graph:

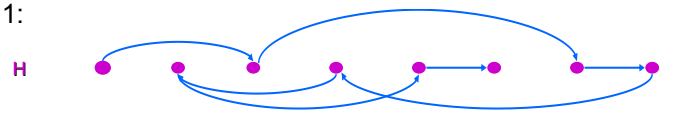
S = { ATG TGG TGC GTG GGC GCA GCG CGT }



SBH: Hamiltonian Path Approach

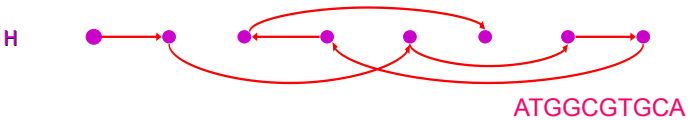
S = { ATG TGG TGC GTG GGC GCA GCG CGT }

Path 1:



ATGCGTGGCA

Path 2:



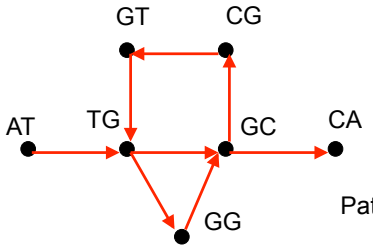
ATGGCGTGCA

SBH: Eulerian Path Approach

S = { ATG, TGC, GTG, GGC, GCA, GCG, CGT }

Vertices correspond to (k-1)-mers: { AT, TG, GC, GG, GT, CA, CG }

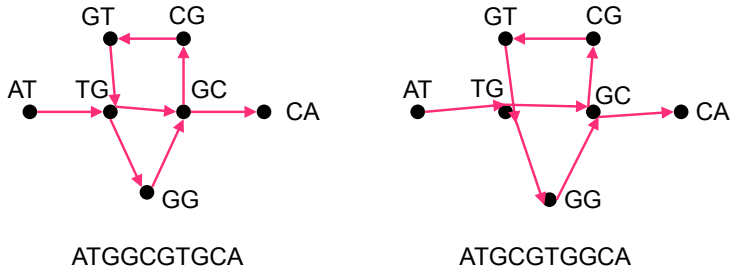
Edges correspond to k-mers from S



Path visited every EDGE once

SBH: Eulerian Path Approach

S = { AT, TG, GC, GG, GT, CA, CG } corresponds to two different paths:



Euler Theorem

- A digraph is balanced if for every vertex the number of incoming edges equals to the number of outgoing edges:

  $in(v)=out(v)$

- **Theorem**: *A connected digraph is Eulerian if and only if each of its vertices is balanced.*

Euler Theorem: Proof

- Eulerian → balanced

  for every edge entering *v* (incoming edge) there exists an edge leaving *v* (outgoing edge). Therefore

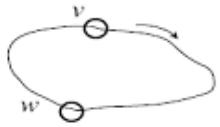
  $in(v)=out(v)$

- Balanced → Eulerian

  ???

Algorithm for Constructing an Eulerian Cycle

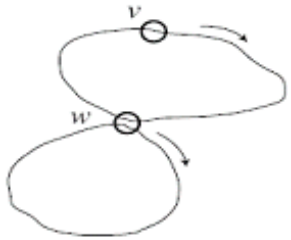
a. Start with an arbitrary vertex *v* and form an arbitrary cycle with unused edges until a dead end is reached. Since the graph is Eulerian this dead end is necessarily the starting point, i.e., vertex *v*.



(a)

Algorithm for Constructing an Eulerian Cycle (cont'd)

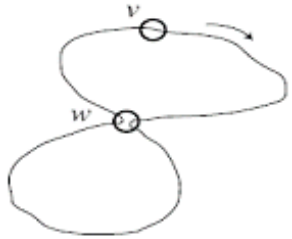
b. If cycle from (a) above is not an Eulerian cycle, it must contain a vertex *w*, which has untraversed edges (G connected). Perform step (a) again, using vertex *w* as the starting point. Once again, we will end up in the starting vertex *w*.



(b)

Algorithm for Constructing an Eulerian Cycle (cont'd)

c. Combine the cycles from (a) and (b) into a single cycle and iterate step (b).



(c)

N.B.: The proof gives an algorithm for constructing an Eulerian cycle: Hierholzer's algorithm. Running time: O(m), where m=no. of edges.

Euler Theorem: Extension

- **Theorem**: *A connected digraph has an Eulerian path if*
- *a) it is balanced (in this case, it contains an Eulerian cycle), or*
- *b) by adding one edge, it becomes balanced (in this case, it contains an Eulerian path which is not a cycle).*
- ***N.B.:** b) is equivalent to: all but two vertices, say s and t, are balanced, while in(s)=out(s)-1 and in(t)=out(t)-1.*

Some Difficulties with SBH

- **Fidelity of Hybridization:** difficult to detect differences between probes hybridized with perfect matches and 1 or 2 mismatches
- **Array Size:** Effect of low fidelity can be decreased with longer *k*-mers, but array size increases exponentially in *k*. Array size is limited with current technology.
- **Practicality:** SBH is still impractical. As DNA microarray technology improves, SBH may become practical in the future
- **Practicality again**: Although SBH is still impractical, it spearheaded expression analysis and SNP analysis techniques