Bioinformatics-themed projects in Discrete Mathematics

Art Duval

University of Texas at El Paso

Joint Mathematics Meeting
MAA Contributed Paper Session on
Discrete Mathematics in the Undergraduate Curriculum
San Diego
January 10, 2018

Bioinformatics and Discrete Mathematics

- Bioinformatics: Use of mathematics and computer science in biology. Increasingly important as biology creates huge databases. UTEP has an M.S. program in bioinformatics.

Bioinformatics and Discrete Mathematics

- Bioinformatics: Use of mathematics and computer science in biology. Increasingly important as biology creates huge databases. UTEP has an M.S. program in bioinformatics.
- Discrete Mathematics: Sophomore-level course for computer science and math majors (including future high school and middle school teachers). Topics include: sets/functions/relations, combinatorics, graph theory.

Bioinformatics and Discrete Mathematics

- Bioinformatics: Use of mathematics and computer science in biology. Increasingly important as biology creates huge databases. UTEP has an M.S. program in bioinformatics.
- Discrete Mathematics: Sophomore-level course for computer science and math majors (including future high school and middle school teachers). Topics include: sets/functions/relations, combinatorics, graph theory.
- My task: Put bioinformatics in Discrete Mathematics course.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

Summary of projects

1. Indel; relations: Build insertion-deletion (symmetric) relation from insertion and deletion (anti-symmetric) relations.
2. Sequence alignment (Smith-Waterman); induction (recursive algorithm): Work through examples of a recursive algorithm, and (maybe) prove it works, by induction.
3. Mass spectrometry; counting: If you build a complete database of precise molecular weights of all fragments of strings of molecules, how big would it have to be?
4. Reaction networks (Petri nets); directed graphs: Simulate and analyze dynamical systems from directed graph.
5. Sequence reassembly (SBH); Euler path: Fragments of length b are the edges, and fragments of length $b-1$ are the vertices; then the entire sequence is an Euler trail.
6. Reconstructing phylogenetic trees (UPGMA); rooted trees: Work through examples of an algorithm that re-creates a weighted rooted tree from pairwise distances.

SBH (Euler trail) project in detail: the idea

- DNA is long string of letters
- shotgun method: make many copies of fragments of string
- reconstruction is "shortest superstring problem"
- note, but ignore, noisy data, repeated long fragments

SBH (Euler trail) project in detail: the idea

- DNA is long string of letters
- shotgun method: make many copies of fragments of string
- reconstruction is "shortest superstring problem"
- note, but ignore, noisy data, repeated long fragments
- Sequencing by hybridization (SBH): all fragments of small fixed length b (find with "DNA chip")

Example

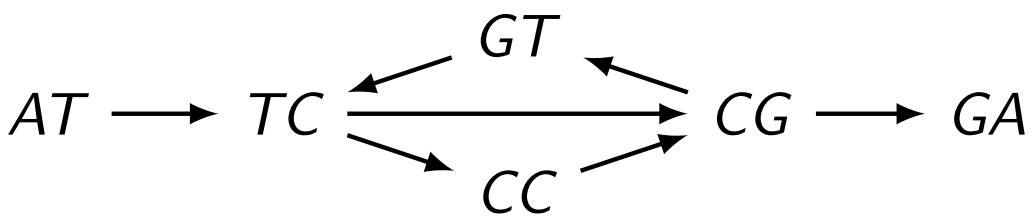
$b = 3$: $ATC, CCG, CGA, CGT, GTC, TCC, TCG$

SBH (Euler trail) project in detail: the idea

- DNA is long string of letters
- shotgun method: make many copies of fragments of string
- reconstruction is "shortest superstring problem"
- note, but ignore, noisy data, repeated long fragments
- Sequencing by hybridization (SBH): all fragments of small fixed length b (find with "DNA chip")
- Idea: fragments are edges, subfragments of length $b-1$ are vertices

Example

$b = 3$: $ATC, CCG, CGA, CGT, GTC, TCC, TCG$

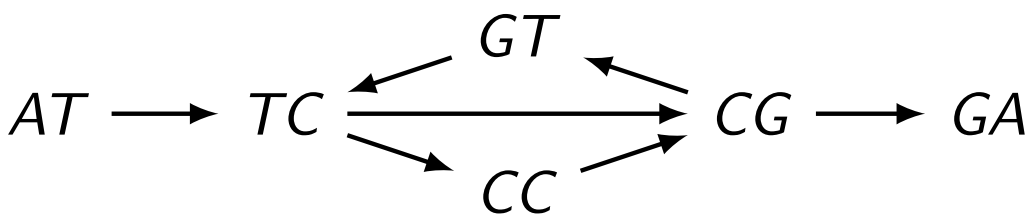


SBH (Euler trail) project in detail: the idea

- DNA is long string of letters
- shotgun method: make many copies of fragments of string
- reconstruction is "shortest superstring problem"
- note, but ignore, noisy data, repeated long fragments
- Sequencing by hybridization (SBH): all fragments of small fixed length b (find with "DNA chip")
- Idea: fragments are edges, subfragments of length $b-1$ are vertices, and we look for an Eulerian path (all edges).

Example

$b = 3$: $ATC, CCG, CGA, CGT, GTC, TCC, TCG$



$ATCCGTCGA$ or $ATCGTCCGA$.

SBH (Euler trail) project in detail: the assignment

1. Given a string of length 12 (the answer, in some sense). Find the vertices (fragments of length $b - 1 = 2$) and the edges (fragments of length $b = 3$); use these to draw the graph (8 vertices, 10 edges); show how one Euler path reconstructs the string.

SBH (Euler trail) project in detail: the assignment

1. Given a string of length 12 (the answer, in some sense). Find the vertices (fragments of length $b - 1 = 2$) and the edges (fragments of length $b = 3$); use these to draw the graph (8 vertices, 10 edges); show how one Euler path reconstructs the string.
2. Given 10 fragments of length $b = 3$, reconstruct the original string. Parts guide students in parallel to question 1: Find the vertices and the edges; draw the graph (9 vertices, 10 edges); find an Euler path; reconstruct the original string.

SBH (Euler trail) project in detail: the assignment

1. Given a string of length 12 (the answer, in some sense). Find the vertices (fragments of length $b-1=2$) and the edges (fragments of length $b=3$); use these to draw the graph (8 vertices, 10 edges); show how one Euler path reconstructs the string.
2. Given 10 fragments of length $b=3$, reconstruct the original string. Parts guide students in parallel to question 1: Find the vertices and the edges; draw the graph (9 vertices, 10 edges); find an Euler path; reconstruct the original string.
3. I generate random strings of length 40 (one string for each student); find the fragments of length 4 for each string; each student must reconstruct the string (pick just one, if several possibilities). No guiding parts.

SBH (Euler trail) project in detail: the assignment

1. Given a string of length 12 (the answer, in some sense). Find the vertices (fragments of length $b-1=2$) and the edges (fragments of length $b=3$); use these to draw the graph (8 vertices, 10 edges); show how one Euler path reconstructs the string.
2. Given 10 fragments of length $b=3$, reconstruct the original string. Parts guide students in parallel to question 1: Find the vertices and the edges; draw the graph (9 vertices, 10 edges); find an Euler path; reconstruct the original string.
3. I generate random strings of length 40 (one string for each student); find the fragments of length 4 for each string; each student must reconstruct the string (pick just one, if several possibilities). No guiding parts.

My automation: Program to generate random strings; program to generate all fragments of a string, and a related program to check answers (and help find errors).

How to find topics

One original suggestion was to work in bioinformatics-themed examples when introducing new discrete mathematics topics. This was hard to do, as the students were not familiar with the bioinformatics examples or contexts, so it was not helpful in conveying new discrete mathematics topics. This why I went to projects.

How to find topics

One original suggestion was to work in bioinformatics-themed examples when introducing new discrete mathematics topics. This was hard to do, as the students were not familiar with the bioinformatics examples or contexts, so it was not helpful in conveying new discrete mathematics topics. This why I went to projects.

To find topics, I spoke to as many biologists and bioinformatics people as I could. "Tell me about things you are working on or that you have seen, and what mathematics you use." Genomics especially well-suited.

How to find topics

One original suggestion was to work in bioinformatics-themed examples when introducing new discrete mathematics topics. This was hard to do, as the students were not familiar with the bioinformatics examples or contexts, so it was not helpful in conveying new discrete mathematics topics. This why I went to projects.

To find topics, I spoke to as many biologists and bioinformatics people as I could. "Tell me about things you are working on or that you have seen, and what mathematics you use." Genomics especially well-suited.

One difficulty is that, although bioinformatics (especially genomics) is full of algorithms, techniques, and definitions that use discrete mathematics ideas, some of these are more advanced, or don't lend themselves to advancing learning objectives of course.

Refining projects

- Not everything worked so well, especially the first time.

Refining projects

- Not everything worked so well, especially the first time.
- One I have completely abandoned (Petri nets).

Refining projects

- Not everything worked so well, especially the first time.
- One I have completely abandoned (Petri nets).
- Often hard to get students to see interesting mathematical point.

Refining projects

- Not everything worked so well, especially the first time.
- One I have completely abandoned (Petri nets).
- Often hard to get students to see interesting mathematical point.
- Some projects are reduced to little more than stepping through an algorithm.

Refining projects

- Not everything worked so well, especially the first time.
- One I have completely abandoned (Petri nets).
- Often hard to get students to see interesting mathematical point.
- Some projects are reduced to little more than stepping through an algorithm.
- Mass spectrometry: The actual example of reconstructing a string of molecules from the molecular weights of some of its fragments is now extra credit (main body of project is just counting size of database, and related counting problems).

Refining projects

- Not everything worked so well, especially the first time.
- One I have completely abandoned (Petri nets).
- Often hard to get students to see interesting mathematical point.
- Some projects are reduced to little more than stepping through an algorithm.
- Mass spectrometry: The actual example of reconstructing a string of molecules from the molecular weights of some of its fragments is now extra credit (main body of project is just counting size of database, and related counting problems).
- Change some examples every time; others are too hard to change.

How they are used in class

- Projects with 1-2 weeks to work on each one, (ideally) assigned shortly after discussing relevant discrete mathematics topics in class; 3-4 projects per semester.

How they are used in class

- Projects with 1-2 weeks to work on each one, (ideally) assigned shortly after discussing relevant discrete mathematics topics in class; 3-4 projects per semester.
- Take about 50 minutes to discuss and explain each project.

How they are used in class

- Projects with 1-2 weeks to work on each one, (ideally) assigned shortly after discussing relevant discrete mathematics topics in class; 3-4 projects per semester.
- Take about 50 minutes to discuss and explain each project.
- I find and post websites and articles for students to look at outside of class, for help and for further details and context.

How they are used in class

- Projects with 1-2 weeks to work on each one, (ideally) assigned shortly after discussing relevant discrete mathematics topics in class; 3-4 projects per semester.
- Take about 50 minutes to discuss and explain each project.
- I find and post websites and articles for students to look at outside of class, for help and for further details and context.
- Students turn in written report, which I grade myself.

How they are used in class

- Projects with 1-2 weeks to work on each one, (ideally) assigned shortly after discussing relevant discrete mathematics topics in class; 3-4 projects per semester.
- Take about 50 minutes to discuss and explain each project.
- I find and post websites and articles for students to look at outside of class, for help and for further details and context.
- Students turn in written report, which I grade myself.
- Students *may* write computer code to solve problem (but not use code written by others), with brief explanation of how they wrote it; not many students have done this, and usually it hasn't been helpful.

Website and acknowledgements

http://www.math.utep.edu/Faculty/duval/class/2300/171/homework.html

(and similar pages for other semesters, since Spring 2010, excluding Fall 2017), to see a semester's worth of projects, including supporting websites and resources.

Website and acknowledgements

http://www.math.utep.edu/Faculty/duval/class/2300/171/homework.html

(and similar pages for other semesters, since Spring 2010, excluding Fall 2017), to see a semester's worth of projects, including supporting websites and resources.

- Thanks to NIH grant 1T36GM078000-01 "Enhancement of Quantitative Science in Biology Curricula" for time to develop projects.
- Thanks to Steve Aley, Elizabeth Walsh, Ming-Ying Leung, Max Shpak, Leo Saldivar, and everyone else at UTEP who talked to me about biology.
- Thanks to Simons Foundation Collaboration Grant 516801 for my travel today.

Website and acknowledgements

http://www.math.utep.edu/Faculty/duval/class/2300/171/homework.html

(and similar pages for other semesters, since Spring 2010, excluding Fall 2017), to see a semester's worth of projects, including supporting websites and resources.

- Thanks to NIH grant 1T36GM078000-01 "Enhancement of Quantitative Science in Biology Curricula" for time to develop projects.
- Thanks to Steve Aley, Elizabeth Walsh, Ming-Ying Leung, Max Shpak, Leo Saldivar, and everyone else at UTEP who talked to me about biology.
- Thanks to Simons Foundation Collaboration Grant 516801 for my travel today.
- Thanks to you for your attention!