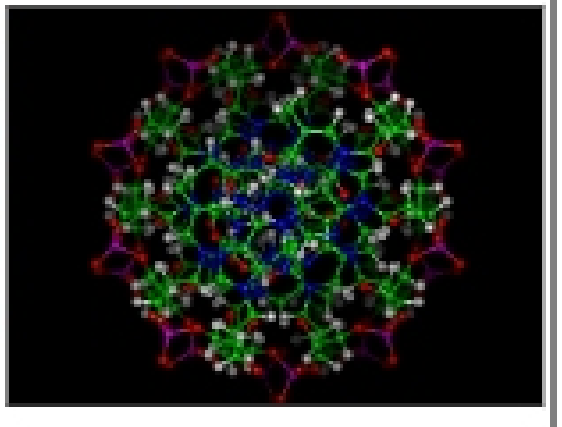
Class 26: Computing Genomes, Genomes Computing



David Evans
http://www.cs.virginia.edu/evans

cs200: Theory of Computation
University of Virginia Computer Science

Final Exam Sneak Preview

- Handout available now
- Honor policy: you may discuss these problems with others and use any resources you want until the Final
- No notes or other resources may be used during the final
- Intent is to give you an idea what to expect on the final and a chance to start thinking about some problems
  - Don't attempt to memorize answers: need to understand things since the actual questions may be different

Menu

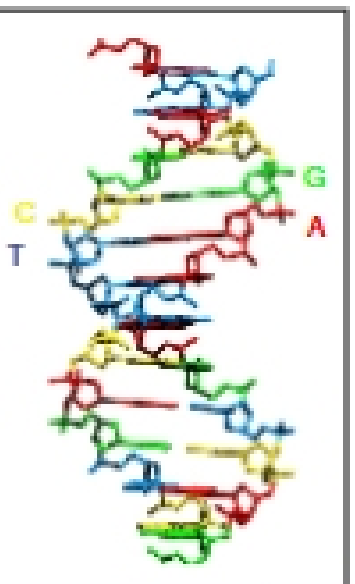
- Computing Genomes (PS6, Problem 6)
  - Crash course in biology
- **Busy Beaver result!**
- Computing with Genomes
- Conclusion

Genome Assembly Problem

In order to assemble a genome, it is necessary to combine snippets from many reads into a single sequence. The input is a set of n genome snippets, each of which is a string of up to k symbols. The output is the smallest single string that contains all of the input snippets as substrings.

DNA

- Sequence of nucleotides: adenine (A), guanine (G), cytosine (C), and thymine (T)
- Two strands, A must attach to T and G must attach to C



Central Dogma of Biology

- RNA makes copies of DNA segments
- RNA describes sequences of amino acids
- Chains of amino acids make proteins
- Proteins make us

Human Genome

- 3 Billion Base Pairs
  - Each nucleotide is 2 bits (4 possibilities)
  - 3 B pairs * 1 byte/4 pairs = 750 MB

1 CD ~ 650 MB

- Every sequence of 3 base pairs one of 20 amino acids (or stop codon)
  - 21 possible codons, but $4^3 = 64$ possible
  - So, really only 750MB * (21/64) ~ 250 MB
- Much of it (> 95%) is may be junk (doesn't encode proteins, but some might be important)

Human Genome Race

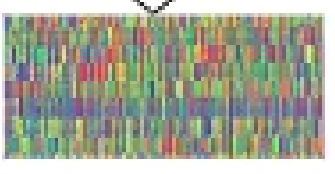
Francis Collins (Director of public National Center for Human Genome Research) (Picture from UVa Graduation 2000)

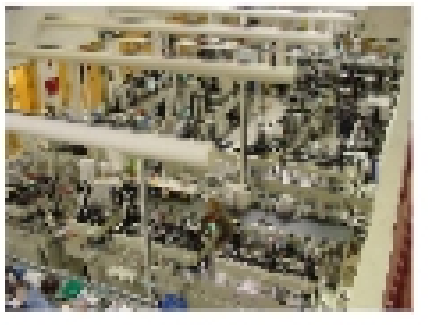
VS.

Craig Venter (President of Celera Genomics)

- UVa CLAS 1970
- Yale PhD
- Tenured Professor at U. Michigan
- San Mateo College
- Court-martialed
- Denied tenure at SUNY Buffalo

Reading the Genome





Whitehead Institute, MIT

Gene Reading Machines

- One read: about 700 base pairs
- But…don't know where they are on the chromosome

Read 3 TACCCGTGATCCA

Read 2 TCCAGAATAA

Read 1 ACCAGAATACC

Actual Genome AGGCATACCAGAATACCCGTGATCCAGAATAAGC

Genome Assembly Problem

Read 1 ACCAGAATACC

Read 2 TCCAGAATAA

Read 3 TACCCGTGATCCA

Input: Genome fragments (but without knowing where they are from)

Ouput: The full genome

Decision Problem

$GA = \{ < \{ x_1, x_2, \ldots, x_n \}, m > \text{ I where}$
each x_i is a string and
there is a string X of length m
that includes all of the x_i strings
as substrings $\}$

If we had a decider for GA, can we find the length of the shortest common superstring?

Yes. Try all possible m values from 1, 2, …, $\Sigma|x_i|$

Is GA In Class NP?

$GA = \{ < \{ x_1, x_2, \ldots, x_n \}, m > \mid$ where
each x_i is a string and
there is a string X of length m
that includes all of the x_i strings
as substrings $\}$

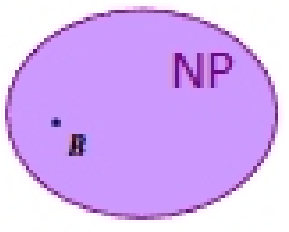
Yes. The string X is a polynomial-verifiable certificate.
- Check it includes each substring
- Check its length is $\leq m$

Is GA NP-Complete?

$GA = \{ < \{ x_1, x_2, \ldots, x_n \}, m > \mid$ where
each x_i is a string and
there is a string X of length m
that includes all of the x_i strings
as substrings $\}$

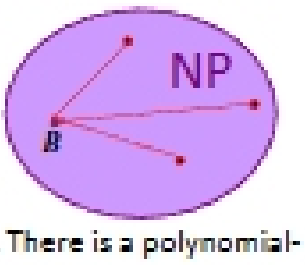
NP-Complete

A language B is in NP-complete if:



1. $B \in$ **NP**

✓



2. There is a polynomial-time reduction from every problem $A \in$ **NP** to B.

?

To Prove NP-Hardness

- Pick some known NP-Complete problem X.
- Show that a polynomial-time solver for Y could be used to build a polynomial-time solver for X.
- This proves that there is no polynomial-time solver for Y (unless **P** = **NP**).

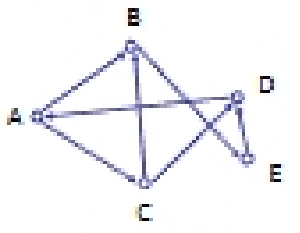
Possible Choices…

$(a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \ldots$

3SAT

$(<\{3, 5, 12, 13, 17\}, 30)$

SUBSET-SUM



HAMPATH

By definition, all must work. Every NP-Complete problem can be reduced to every NP-Complete problem.

In practice, some will work much more easily than others. Try to pick a problem "close" to the target problem.

Busy Beaver Challenge

Ruixin Yang