

Genome Reassembly From Fragments



Genome

- A **genome** is the encoding of hereditary information for an organism in its DNA
- The mathematical model of a genome is a **string of character**, where each character is one of **'A'**, **'C'**, **'G'**, or **'T'**, which stand for the names of the four nucleotides that occur on a DNA backbone

Quoted from Wikipedia:


- An analogy to the human genome stored on DNA is that of instructions stored in a book:
 - The book (genome) contains 23 chapters (chromosomes);
 - Each chapter contains 48 to 250 million letters (A,C,G,T) without spaces;
 - Hence, the book contains over 3.2 billion letters total;
 - The book fits into a cell nucleus the size of a pinpoint;
 - At least one copy of the book (all 23 chapters) is contained in most cells of our body.

Quoted from Wikipedia:

- An analogy to the human genome is that of instruction book. This is what we care about for the next project...
 - The book (genome) contains 23 chapters (chromosomes);
 - Each chapter contains 48 to 250 million letters (A,C,G,T) without spaces;
 - Hence, the book contains over 3.2 billion letters total;
 - The book fits into a cell nucleus the size of a pinpoint;
 - At least one copy of the book (all 23 chapters) is contained in most cells of our body.

Genome Sequencing

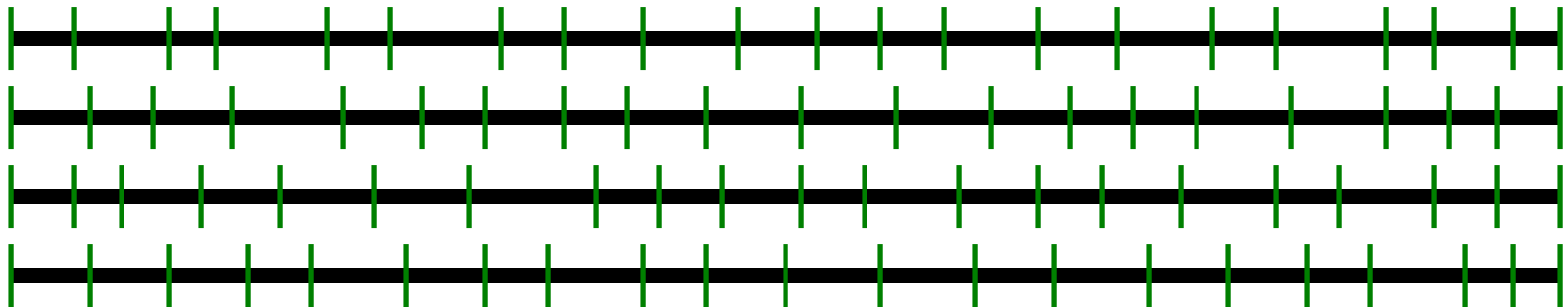
- The **Human Genome Project** was designed to determine the entire sequence of human DNA and to map its mathematical model (genotype) to physical and functional manifestations in a person (phenotype)
- **Sequencing** is done “piece-by-piece” because it is effectively impossible to do anything directly with 3.2 billion nucleotides



length = 3.2 billion

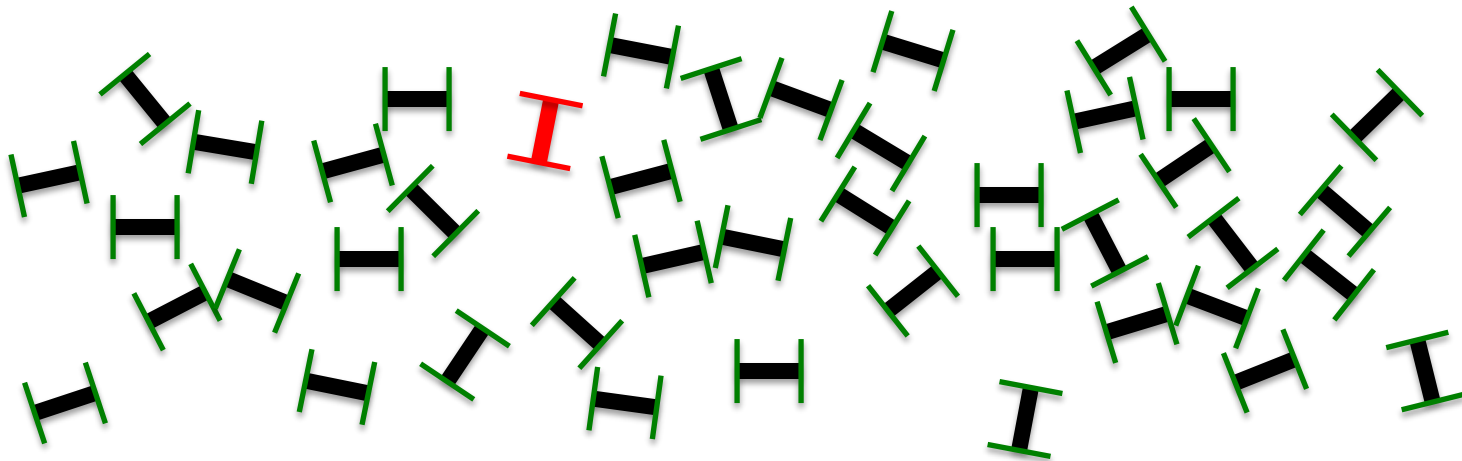
Genome Sequencing: Step 1

- Use enzymes that can cut up many strands of the same DNA (each a string of length about 3.2 billion letters or “bases”) into pieces at different locations, creating a “soup” of *fragments* each of much smaller length (on the order of 1000)



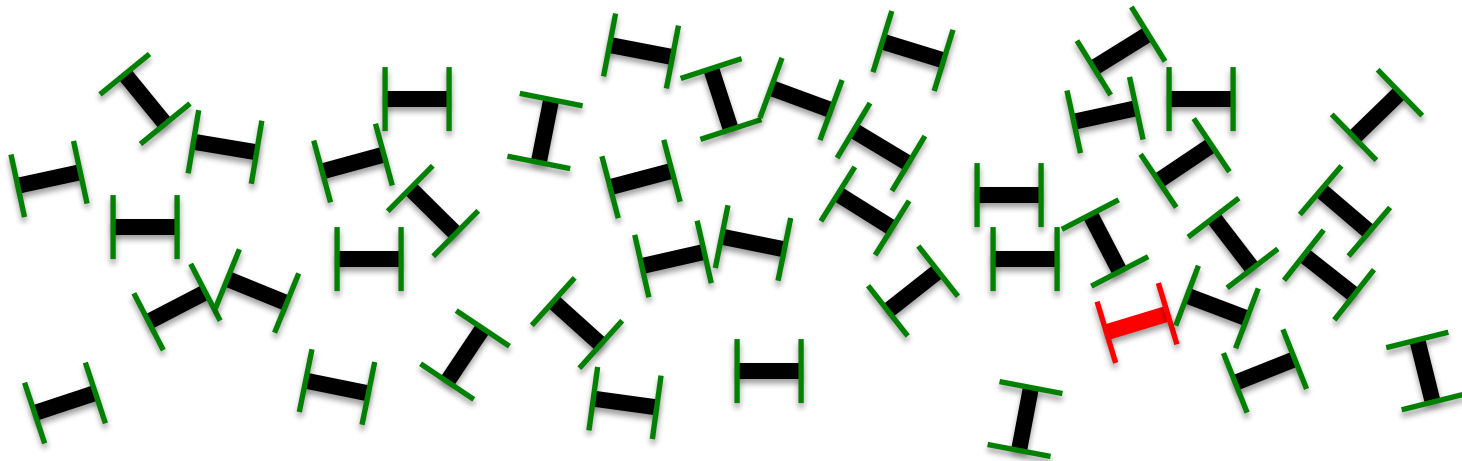
Genome Sequencing: Step 2

- Use machines that can physically sequence each of these fragments to determine their mathematical models
 - Example: *"TCTAAGCCTA..."*



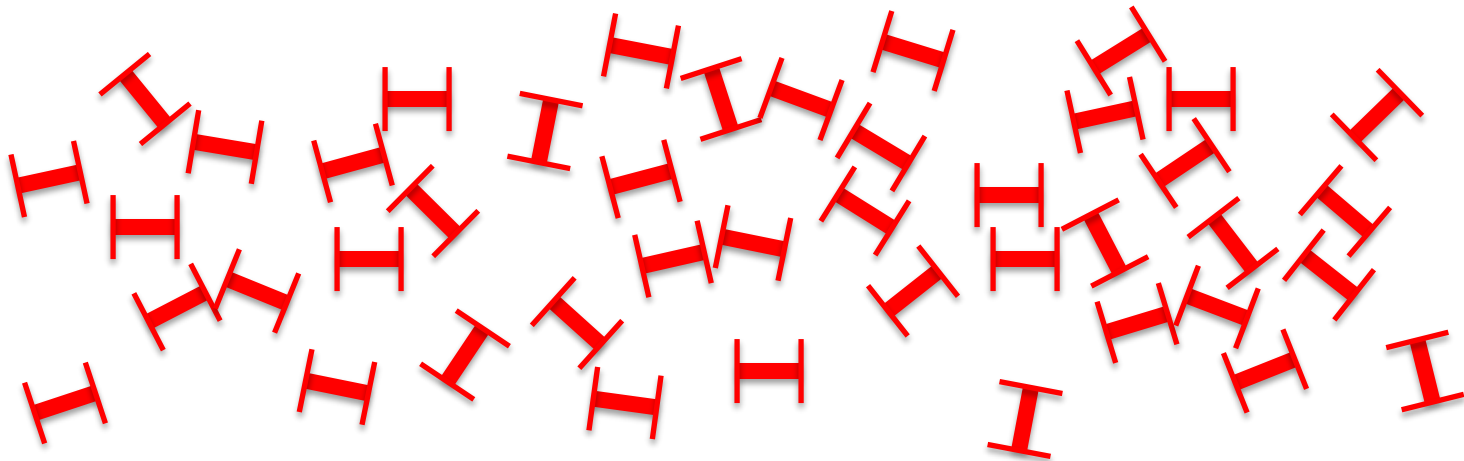
Genome Sequencing: Step 2

- Use machines that can physically sequence each of these fragments to determine their mathematical models
 - Example: *"AGTAGAACG..."*



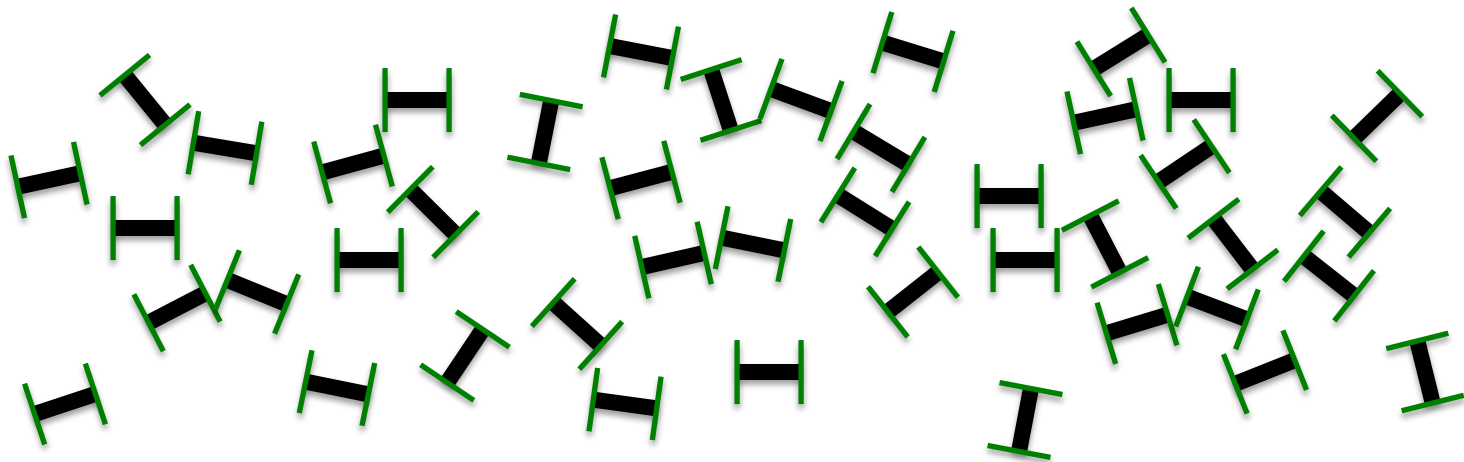
Genome Sequencing: Step 2

- Use machines that can physically sequence each of these fragments to determine their mathematical models



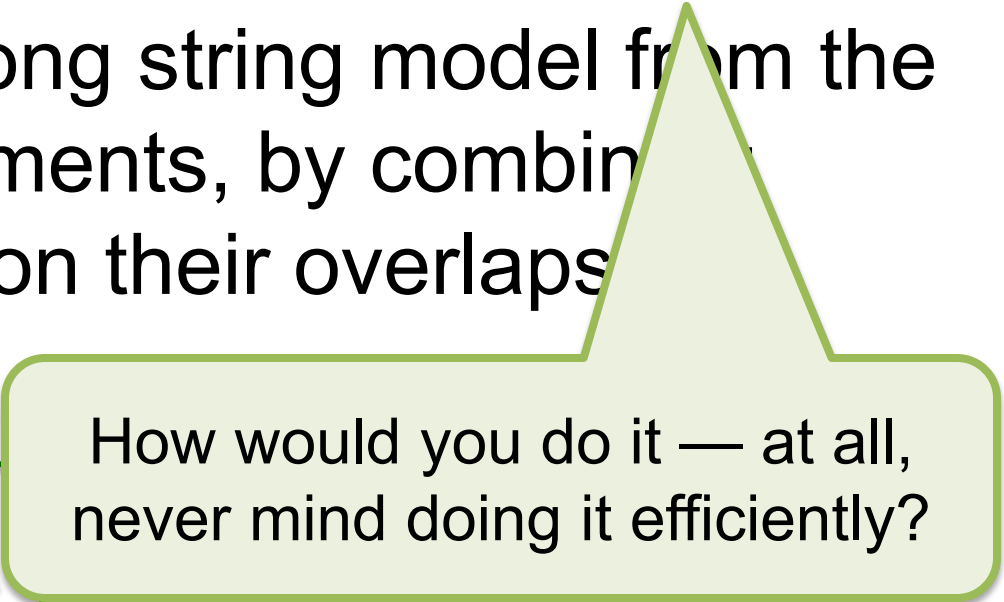
Genome Sequencing: Step 3

- Use computer algorithms to **reassemble** the original very long string model from the models of its fragments, by combining fragments based on their overlaps

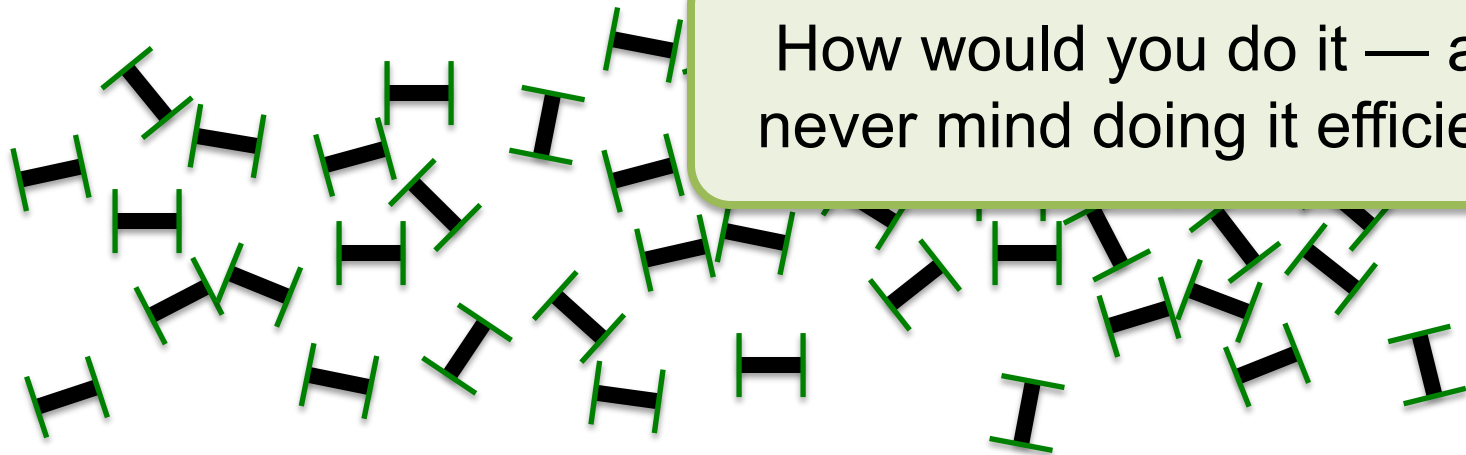


Genome Sequencing: Step 3

- Use computer algorithms to **reassemble** the original very long string model from the models of its fragments, by combining fragments based on their overlaps

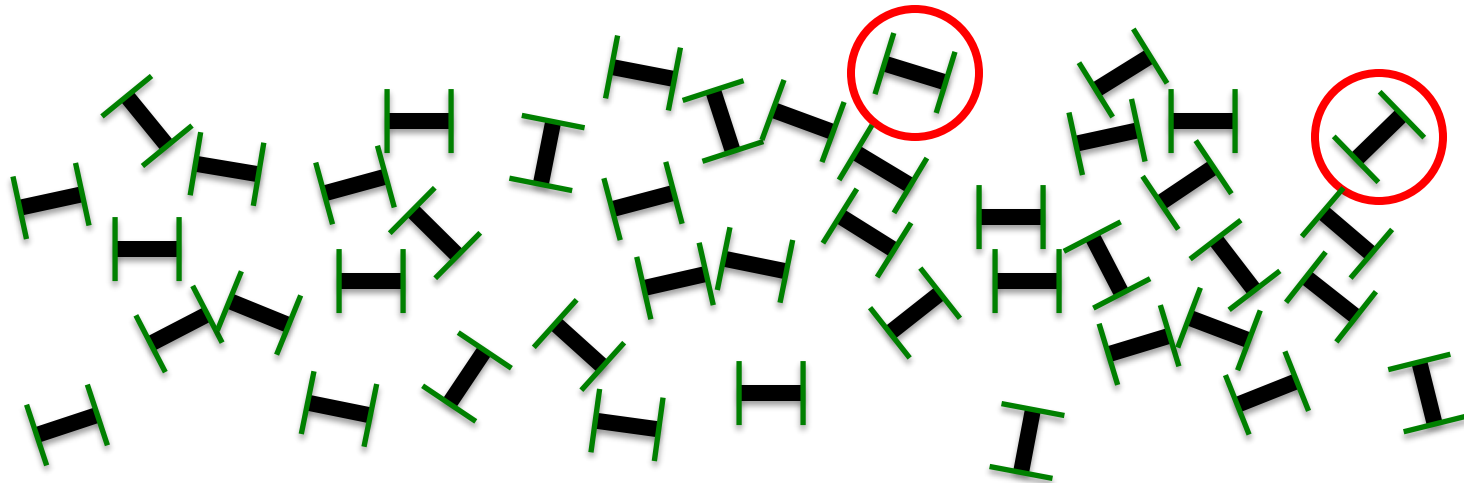


How would you do it — at all, never mind doing it efficiently?



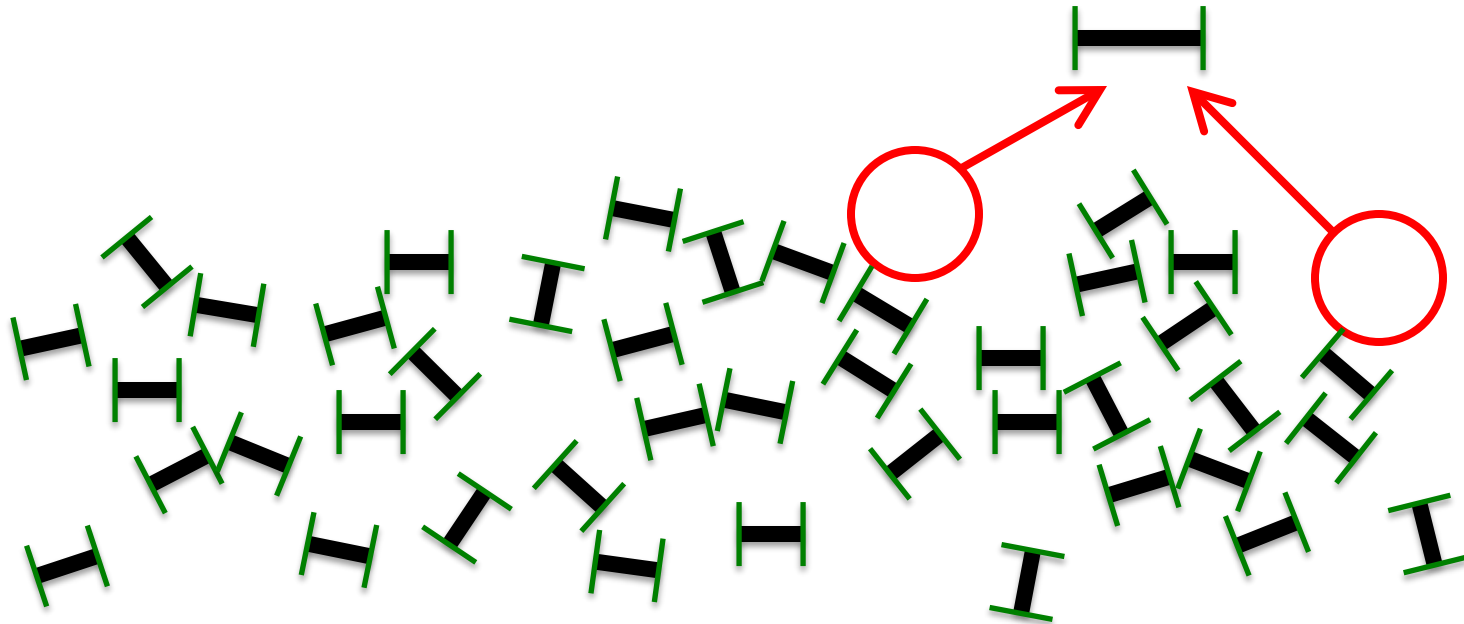
Greedy Reassembly: Step 1

- A naïve (but still interesting) idea is to pick two fragments with *the most overlap* and to *combine* them into a longer fragment



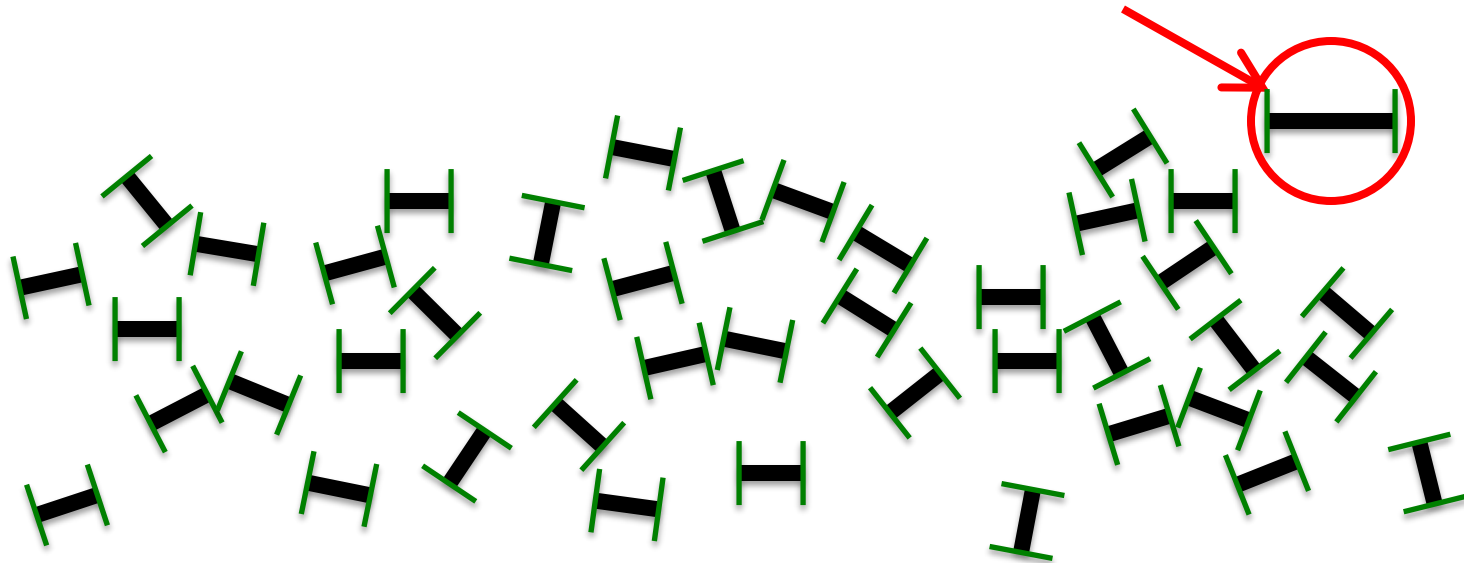
Greedy Reassembly: Step 1

- A naïve (but still interesting) idea is to pick two fragments with *the most overlap* and to *combine* them into a longer fragment



Greedy Reassembly: Step 1

- A naïve (but still interesting) idea is to pick two fragments with *the most overlap* and to *combine* them into a longer fragment



Finding Overlaps

- Given two strings, what is the longest string that is a *prefix* of one and a *suffix* of the other?
- Example of one pair of strings:

s1 = "AGTAGAACG"

s2 = "CGAGGTAGT"

Finding Overlaps

- Given two strings, what is the longest string that is a *prefix* of one and a *suffix* of the other?
- Example of one pair of strings:

$s1 = \text{"AGTAGAA} \mathbf{CG}$ "

$s2 = \mathbf{CG}$ AGGTAGT"

Finding Overlaps

- Given two strings, what is the longest string that is a *prefix* of one and a *suffix* of the other?
- Example of one pair of strings:

$s1 = \text{"AGTAGAACG"}$

$s2 = \text{"CGAGGTAGT"}$

Finding Overlaps

- Given two strings, what is the longest string that is a *prefix* of one and a *suffix* of the other?
- Example of one pair of strings:

$s1 = \text{"AGTAGAACG"}$

$s2 = \text{"CGAGGTAGT"}$

The longest string that is a prefix of one and a suffix of the other is **"AGT"**.

Combine

- If these two strings have *the most overlap of any pair* in the “soup”, then we remove these two strings from the “soup”:

"**AGT**AGAACG"

"CGAGGT**AGT**"

and replace them by this one:

"CGAGGT**AGT**AGAACG"

Combine

- If these two strings belong to *any pair* in the “sorted” set, then delete these two strings from the set.

"**AGT**AGAACG"

"CGAGGT**AGT**"

and replace them by this one:

"CGAGGT**AGT**AGAACG"

The idea is that both the shorter strings could have been fragments of this longer string.

Combine

- If these two strings have *the most overlap of any pair* in the “soup”, then we remove these two strings from the “soup”:

"**AGT**AGAACG"

"CGAGGT

and rep

"CGAGG

Notice that math model of the “soup” is a *finite set of string of character*, so in a Java program it can be of type `Set<String>`.

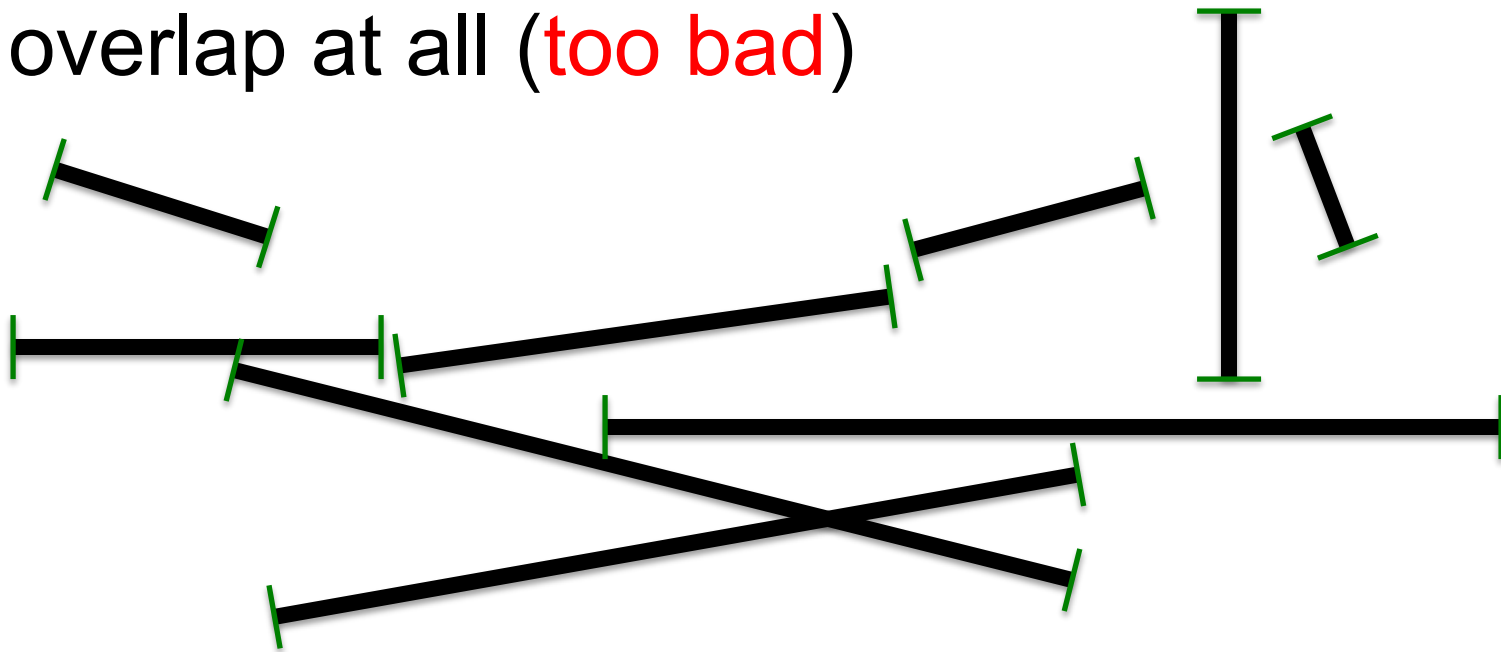
Greedy Reassembly: Step 2

- Continue the process until there is only one fragment in the “soup” (**declare success**)



Greedy Reassembly: Step 2

- Continue the process until there is only one fragment in the “soup” (declare success), or until no two fragments overlap at all (**too bad**)



Success?

- Even if there is only one fragment left, it might not be the original long string that was chopped up — but it's a good guess!
 - And after all, we are just guessing; critical information is lost when the long strand is chopped up into fragments, but we can reassemble it from fragments with high probability if enough copies of the original string are chopped up into fragments

Project

- The project is to do greedy reassembly, not for a genome of length 3.2 billion, but rather for a reasonably short piece of text (e.g., the *Gettysburg Address*), many copies of which have been chopped up into random fragments for you to reassemble

Resources

- Wikipedia: Genome
 - <http://en.wikipedia.org/wiki/Genome>
- Wikipedia: Human Genome Project
 - http://en.wikipedia.org/wiki/Human_Genome_Project
- Wikipedia: Whole Genome Sequencing
 - http://en.wikipedia.org/wiki/Genome_sequencing
- Wikipedia: Sequence Assembly
 - http://en.wikipedia.org/wiki/Sequence_assembly