Algorithmic Approaches for Biological Data, Lecture #17

Katherine St. John

City University of New York American Museum of Natural History

6 April 2016

<ロト <四ト <注入 <注下 <注下 <



• Recap: Recursion

<ロ> (日) (日) (日) (日) (日)



- Recap: Recursion
- Assembling Sequence Reads

э.

・ロト ・回ト ・ヨト ・



- Recap: Recursion
- Assembling Sequence Reads
- Reframing Biology Questions

・ロト ・ 日 ト ・ 日 ト ・



- Recap: Recursion
- Assembling Sequence Reads
- Reframing Biology Questions
- Overlap & Hamiltonian Graphs

Image: A match a ma



- Recap: Recursion
- Assembling Sequence Reads
- Reframing Biology Questions
- Overlap & Hamiltonian Graphs
- Hamiltonian & Eulerian Paths

Image: A match a ma

		Mesozoic			
	Paleozoic		Jurassic	Cretaceous	Cenozoic
oraniates	hagfishes				
cramates	conodonts				
vertebrate	lampreys cartilaginous fishes				
	ray-finned fishes				
ja	wed fishes coelacanths				
	bony fishes				
	lobe-finned fishesamphibians				
	mammals and	other synapsids			
	tetrapods turtles				
	amniotes		marine reptiles		
	reptiles		lepidosaurs		
diapsids			crocodilia	ns	
	archosaur	-	birds and	other dinosaurs	
	arcriosau	-			

Smithsonian Institute

• Works well for problems that break into pieces that can be solved independently

(日) (同) (三) (三)

		Mesozoic			
	Paleozoic	Triassic	Jurassic	Cretaceous	Cenozoic
craniates	hagfishes				
Cramatos	conodonts lampreys				
vertebrate					
	ray-finned fishes				
8	wed fishes coelacanths				
	bony fishes lungfishes				
	lobe-finned fishes amphibians				
	tetrapods turtlas				
			marine reptiles		
	amniotes				
reptiles		lepidosaurs crocodilians			
	diapsids				
	archosaur		birds and	other dinosaurs	

Smithsonian Institute

- Works well for problems that break into pieces that can be solved independently
- Two parts:

(日) (同) (三) (三)

		Mesozoic			
	Paleozoic		Jurassic	Cretaceous	Cenozoic
oraniates	hagfishes				
cramates	conodonts				
vertebrate	lampreys cartilaginous fishes				
	ray-finned fishes				
ja	wed fishes coelacanths				
	bony fishes				
	lobe-finned fishesamphibians				
	mammals and	other synapsids			
	tetrapods turtles				
	amniotes		marine reptiles		
	reptiles		lepidosaurs		
diapsids			crocodilia	ns	
	archosaur	-	birds and	other dinosaurs	
	arcriosau	-			



- Works well for problems that break into pieces that can be solved independently
- Two parts:
 - Base Case: smallest possibility

< ロ > < 同 > < 回 > < 回 > < 回 > < 回

		Mesozoic			
	Paleozoic	Triassic	Jurassic	Cretaceous	Cenozoic
craniates	hagfishes				
Cramatos	conodonts lampreys				
vertebrate					
	ray-finned fishes				
8	wed fishes coelacanths				
	bony fishes lungfishes				
	lobe-finned fishes amphibians				
	tetrapods turtlas				
			marine reptiles		
	amniotes				
reptiles		lepidosaurs crocodilians			
	diapsids				
	archosaur		birds and	other dinosaurs	



- Works well for problems that break into pieces that can be solved independently
- Two parts:
 - Base Case: smallest possibility (for trees: almost always the leaves)

(日) (同) (三) (三)

		Mesozoic			
	Paleozoic		Jurassic	Cretaceous	Cenozoic
oraniates	hagfishes				
cramates	conodonts				
vertebrate	lampreys cartilaginous fishes				
	ray-finned fishes				
ja	wed fishes coelacanths				
	bony fishes				
	lobe-finned fishesamphibians				
	mammals and	other synapsids			
	tetrapods turtles				
	amniotes		marine reptiles		
	reptiles		lepidosaurs		
diapsids			crocodilia	ns	
	archosaur	-	birds and	other dinosaurs	
	arcriosau	-			



- Works well for problems that break into pieces that can be solved independently
- Two parts:
 - Base Case: smallest possibility

(for trees: almost always the leaves)

▶ Recursive Call: calls the function on a "smaller" problem

		Mesozoic			
	Paleozoic		Jurassic	Cretaceous	Cenozoic
oraniates	hagfishes				
cramates	conodonts				
vertebrate	lampreys cartilaginous fishes				
	ray-finned fishes				
ja	wed fishes coelacanths				
	bony fishes				
	lobe-finned fishesamphibians				
	mammals and	other synapsids			
	tetrapods turtles				
	amniotes		marine reptiles		
	reptiles		lepidosaurs		
diapsids			crocodilia	ns	
	archosaur	-	birds and	other dinosaurs	
	arcriosau	-			



- Works well for problems that break into pieces that can be solved independently
- Two parts:
 - Base Case: smallest possibility (for trees: almost always the leaves)
 - Recursive Call: calls the function on a "smaller" problem (for trees: internal nodes)

< ロ > < 同 > < 三 > < 三

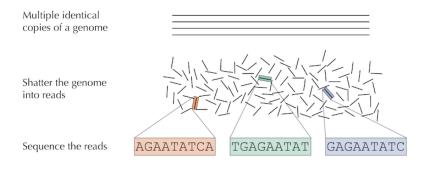
		Mesozoic			
	Paleozoic		Jurassic	Cretaceous	Cenozoic
oraniates	hagfishes				
cramates	conodonts				
vertebrate	lampreys cartilaginous fishes				
	ray-finned fishes				
ja	wed fishes coelacanths				
	bony fishes				
	lobe-finned fishesamphibians				
	mammals and	other synapsids			
	tetrapods turtles				
	amniotes		marine reptiles		
	reptiles		lepidosaurs		
	diapsids		crocodilia	ns	
	archosaur	-	birds and	other dinosaurs	
	arcriosau	-			



- Works well for problems that break into pieces that can be solved independently
- Two parts:
 - Base Case: smallest possibility
 - (for trees: almost always the leaves)
 - Recursive Call: calls the function on a "smaller" problem (for trees: internal nodes)
- PythonTutor Demo

K. St. John (CUNY & AMNH)

- **(())) (())) ())**



Assemble the genome using overlapping reads AGAATATCA GAGAATATC TGAGAATAT ...TGAGAATATCA...

· · · IGAGAAIAICA · ·

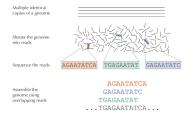
Compeau & Pevzner, Vol 1, Chapter 3

K. St. John (CUNY & AMNH)

Algorithms #17

6 April 2016 4 / 14

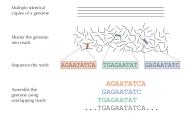
(日) (同) (日) (日)



Compeau & Pevzner, Vol 1, Chapter 3

• Given a bunch of sequence reads (of length k), how do you assembly them?

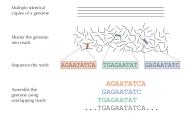
< ロ > < 同 > < 三 > < 三



Compeau & Pevzner, Vol 1, Chapter 3

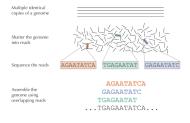
- Given a bunch of sequence reads (of length k), how do you assembly them?
- In pairs/triples:

(日) (同) (三) (三)



Compeau & Pevzner, Vol 1, Chapter 3

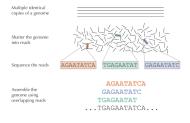
- Given a bunch of sequence reads (of length k), how do you assembly them?
- In pairs/triples:
 - 10 copies of a single-stranded sequence



Compeau & Pevzner, Vol 1, Chapter 3

- Given a bunch of sequence reads (of length k), how do you assembly them?
- In pairs/triples:
 - 10 copies of a single-stranded sequence
 - ▶ k=4

(日) (同) (三) (三)



Compeau & Pevzner, Vol 1, Chapter 3

- Given a bunch of sequence reads (of length k), how do you assembly them?
- In pairs/triples:
 - 10 copies of a single-stranded sequence
 - ▶ k=4
 - Some errors in sequencing.

/□ ▶ 《 ⋽ ▶ 《 ⋽

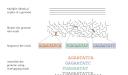
Reframing Biology Questions



TGAGAATATTGAGAATATCA.... • How can we store the information in the computer?

(日) (同) (三) (三)

Reframing Biology Questions

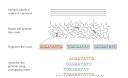


• How can we store the information in the computer?

• What additional information and what structures do we need?

(日) (同) (三) (三)

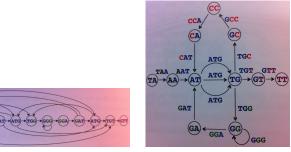
Reframing Biology Questions



- How can we store the information in the computer?
- What additional information and what structures do we need?
- How do you represent overlaps in a graph?

-

Useful Graphs: Overlap & deBruijn



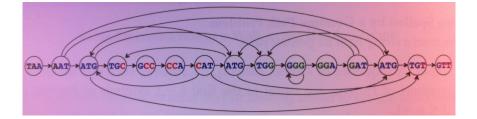
overlap graph

deBruijn graph

< ∃ >

Compeau & Pevzner, Vol 1, Chapter 3

AAT



イロト イヨト イヨト イヨト

GAT TC

• The vertices are the reads.

-

▲ □ ► ▲ □ ► ▲

GA

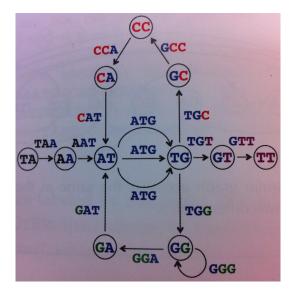
- The vertices are the reads.
- There's an edge from u to v if the Suffix(u) = Prefix(v).

-∢ ∃ ▶

- The vertices are the reads.
- There's an edge from u to v if the Suffix(u) = Prefix(v).
- Each read came from the original sequence, so, should be in the final sequence.

- The vertices are the reads.
- There's an edge from u to v if the Suffix(u) = Prefix(v).
- Each read came from the original sequence, so, should be in the final sequence.
- Final sequence corresponds to a path that visits all the vertices (called Hamiltonian path).

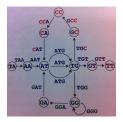
- **4 ∃ ≻** 4



K. St. John (CUNY & AMNH)

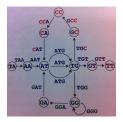
3 6 April 2016 10 / 14

(本語)と (本語)と (本語)と

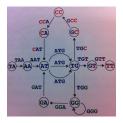


• Label edges by *k*-mers.

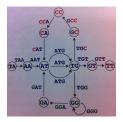
Image: A math a math



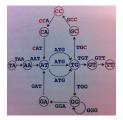
- Label edges by k-mers.
- Label vertices by the prefixes/suffixes.



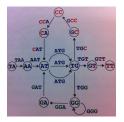
- Label edges by *k*-mers.
- Label vertices by the prefixes/suffixes.
- An Eulerian path through a graph visits every edge exactly once.



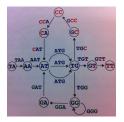
- Label edges by *k*-mers.
- Label vertices by the prefixes/suffixes.
- An Eulerian path through a graph visits every edge exactly once.
- The completed sequence is one of the Eulerian paths of the graph.



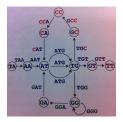
• Finding Hamiltonian paths is NP-hard.



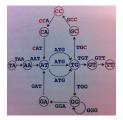
- Finding Hamiltonian paths is NP-hard.
- Eulerian paths exist if the out-degree = in-degree for every vertex.



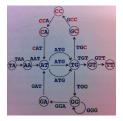
- Finding Hamiltonian paths is NP-hard.
- Eulerian paths exist if the out-degree = in-degree for every vertex.
- Since tractable, will compute Eulerian paths.



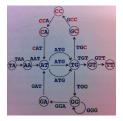
- Finding Hamiltonian paths is NP-hard.
- Eulerian paths exist if the out-degree = in-degree for every vertex.
- Since tractable, will compute Eulerian paths.
- And then, if more than one, choose (using additional information).



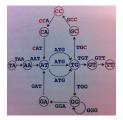
• Input: Reads of multiple sequences



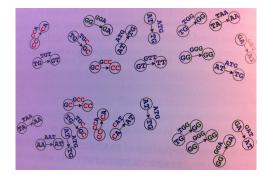
- Input: Reads of multiple sequences
- Make *k*-mers of the reads.



- Input: Reads of multiple sequences
- Make *k*-mers of the reads.
- Build deBruijn graphs from the reads.



- Input: Reads of multiple sequences
- Make *k*-mers of the reads.
- Build deBruijn graphs from the reads.
- (Find Eulerian paths...)



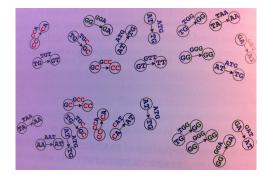
• Coding the deBruijn graph functions in lab today.

K. St. John (CUNY & AMNH)

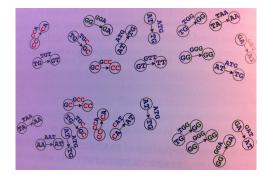
Algorithms #17

6 April 2016 14 / 14

Image: A math a math



- Coding the deBruijn graph functions in lab today.
- Email lab reports to kstjohn@amnh.org



- Coding the deBruijn graph functions in lab today.
- Email lab reports to kstjohn@amnh.org
- Challenges available at rosalind.info