Project

Short Read Sequencing Problem
(A Computer Science Problem)

Full DNA Sequence

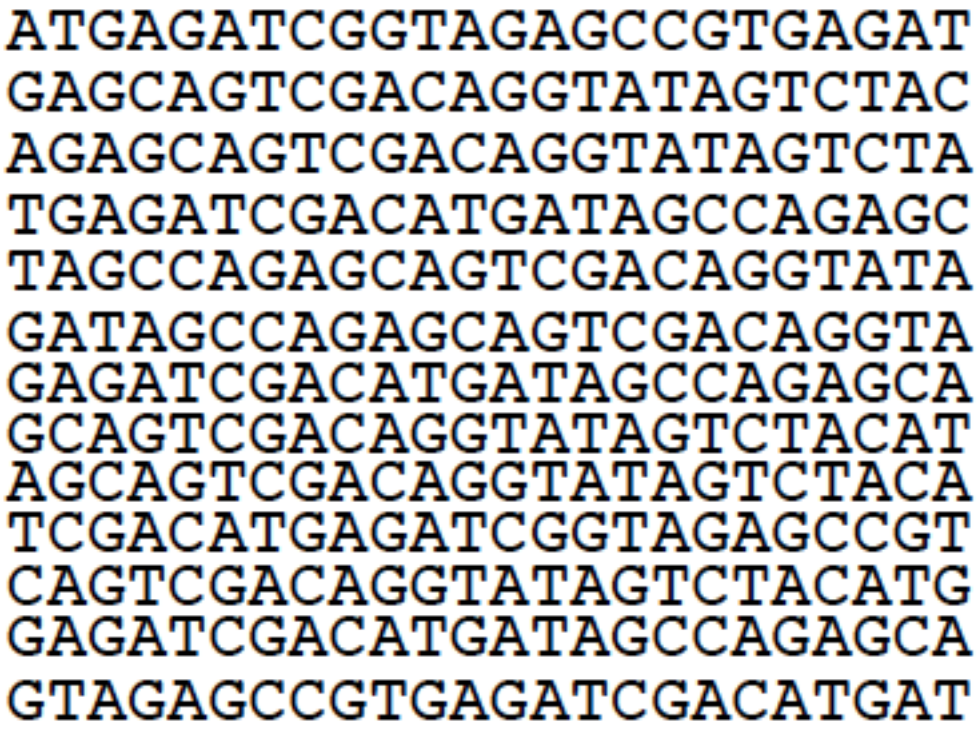
AGAGCAGTCGAC
AGGTATAGTCTA
CATGAGATCGAC
ATGAGATCGGTA
GAGCCGTGAGAT
CGACATGATAGC
CAGAGCAGTCGA
CAGGTATAGTCT
ACATGAGATCGA
CATGAGATCGGT
AGAGCCGTGAGA
TCGACATGATAG
CCAGAGCAGTCG
ACAGGTATAGTC
TACATGAGATCG
ACATGAGATCGG
TAGAGCCGTGAG
ATCGACATGATA
GCCAGAGCAGTC
GACAGGTATAGT
CTACATGAGATC

RECONSTRUCT original sequence !!!

Short read sequencers generate random short substrings from the DNA sequence of a certain length.

ATGAGATCGGTAGAGCCGTGAGAT
GAGCAGTCGACAGGTATAGTCTAC
AGAGCAGTCGACAGGTATAGTCTA
TGAGATCGACATGATAGCCAGAGC
TAGCCAGAGCAGTCGACAGGTATA
GATAGCCAGAGCAGTCGACAGGTA
GAGATCGACATGATAGCCAGAGCA
GCAGTCGACAGGTATAGTCTACAT
AGCAGTCGACAGGTATAGTCTACA
TCGACATGAGATCGGTAGAGCCGT
CAGTCGACAGGTATAGTCTACATG
GAGATCGACATGATAGCCAGAGCA
GTAGAGCCGTGAGATCGACATGAT

Short Reads Difficulties



- We don’t know where each read comes from!
- Can’t identify where the mutations are!
- What do we do?

Project

- Problem: Given M number of short reads of length L, reconstruct the original sequence of length N that those shorts reads come from.
- M, L, N 수는 원하는대로 지정가능. 단 난이도 점수가 달라질 수 있음.
- 문제의 난이도를 높이기위해 여러가지 원하는 요소 추가는 자유임. (시퀀씽 에러, repeat, inversion, insertion, deletion등등)
- 원하는 알고리즘 사용가능
- 비교할 base method(gold standard method) 반드시 포함

Project

- 중간 보고서 제출 – 팀 구성원, 어떤 데이터를 쓸지 (어떻게 데이터를 만들지), 어떤 문제를(define your problem) 어떻게 풀껀지(간단한 아이디어)를 1장짜리 보고서로 제출. 5/20 까지 이클래스 업로드 및 하드카피 제출.
- Presentation 4/28(29), 5/4(5), 5/11(12) 중 택일
- Construct an algorithm and implement it. 프로그램 확인은 발표한고 돌아오는 실습시간에
- 최종 보고서 제출- 발표시간 전까지 eclass 업로드 및 발표일 “발표전에 출력물 교수에게 직접 제출!!!!!!!!” 출력물 준비 안됐으면 보고서 점수 0.

Things to consider

- Complications that you could consider
  - N could be long (up to 3,000,000,000)
  - M could be large (typically 20million~ 200 million)
  - Length of L (typically 32~100)
  - number of mismatches in a read, D
  - denovo or reference?
    - reference genome
    - denovo sequencing -structural variation (insertion, repeats, inversion
    - else

Presentation

- 4~5 mins presentation, 1 min question, 2 questions
- Must include the following slides
  - Introduction
    - Clearly define your problem
    - Explain your data (ex. where you got the data, how you generate the data, or etc.)
    - Input and Output
  - Benchmark - Other algorithm that you compared with your algorithm. ex. trivial mapping algorithm
  - Your Algorithm
  - Result
    - about your machine. (unix/mac/pc, CPU, memory size, etc)
    - time and space complexity in either big O notation or actual time/space
    - Compare with the benchmark
  - Future work
    - cons and pros of your algorithm
    - Something you can do to improve your algorithm

Grading and some tips

- 평가기준
  - 난이도 – 문제의 난이도
  - 정확도 – 알고리즘과 그 구현이 하고자 하는 바를 정확하게 하였는지
  - 명확도 – 누구나 알아들을 수 있게 발표가 clear 한가, bench mark 알고리즘과 비교가 잘이루어졌는지
  - 질문 두번했는지 여부
  - 보고서 점수
- Tips
  - 가능하면 남들과 다른 알고리즘을 사용하면 좋습니다. 같은 구현을 하였을 경우 상대 평가 될 수 있습니다.
  - 빨리 발표하면 좋습니다. 비슷한 발표를 하였을 경우 첫 날 발표한 경우 점수를 더 받을 수 있습니다.
  - 발표 일주일 전까지는 대강 완성을 한다는 생각으로 미리미리 해 놓으세요. 발표 전 몇일은 발표 연습과 보고서를 쓰는데 시간이 걸릴 것 입니다.

설계 프로젝트 평가기준

- 평가기준
  - 난이도(10) – 문제의 난이도, 얼마나 어려운 문제를 했는가
  - 정확도(10) – 알고리즘과 그 구현이 하고자 하는 바를 정확하게 구현 하였는지, 실제 구현 코드 및 실행 확인, 완성도
  - 명확도(10) – 누구나 알아들을 수 있게 발표가 clear 한가, bench mark 알고리즘과 비교가 잘이루어졌는지, 보고서 점수
  - Overall(+/- 10) – 다른 학생들과 비교하였을때 어땠는지에 대한 상대적 평가
  - 질문 두번했는지 – 필수
  - 보고서 (10)

Something you should “Not”

- 발표시간 엄수하세요. 5분지나면 더이상 발표할 수 없으므로 미리 5분 맞춰서 반드시 연습하고 오세요.
- 꼭 필요하다면 시간안걸리는 간단한 프로그램 돌려 결과를 보여주는 것은 가능하나 구현한 코드를 설명하지는 마세요. 어차피 조교랑 프로그램은 다시 검토할 것이며 코드가 아니라 보기쉽게 “알고리즘”을 설명하세요. “발표슬라이드에 코드 넣고 설명하지 마세요!!!"
- 다른 논문 그대로 설명. 논문 review 발표가 아닙니다. 어떤 경우 그냥 논문을 copy & paste해서 알고리즘 설명하는 경우가 있었습니다.
- 구현 미완성 – 어려운 문제를 선택 했을 경우 구현이 잘 안될 수도 있습니다. 그래도 적어도 결과는 나와야합니다. 이유를 설명하고 어떤 때문에 성능이 원하는 만큼 안나왔다는 가능하지만 아예 결과가 도출이 안되면 안됩니다.

Questions?

- 프로젝트에 관련해서 자세한 사항은 수업시간, 쉬는시간, 수업끝나고, 따로 미팅을 잡아서 등등 “교수에게 직접" 질문하세요.