컴퓨터 알고리즘 프로젝트 2016112185 컴퓨터공학과 박지수

문제 정의 Problem Definition

컴퓨터 알고리즘 프로젝트는 생물이 가지는 유전체, 즉 유전 정보를 담고있는 DNA sequence를 활용하여 DNA를 reconstruct하는 것이다. 프로젝트를 진행하기 위해서는 알파벳 ATGC로 이루어진 My genome으로부터 생성된 short reads가 필요하다. 이러한 short reads로부터 원래의 My genome을 복구하는 것이 이번 프로젝트의 주요 문제이다. My genome은 rand() 함수을 사용하여 데이터를 랜덤하게 생성하여 사용을 하였다. 인간의 genome은 약 30억 개로 이루어져 있지만 30억 개를 모두 활용할 경우 시간이 오래 걸리고 데이터를 정확하게 찾았는지 알 수 없기 때문에 크기를 점점 늘려가고 short read의 길이는 줄여가면서 프로젝트를 진행하였다.

데이터 설명 Data Explanation

My genome의 길이: 197000

Short Read의 길이: 800

Short Read의 개수: 492

Short Read로부터 분할한 Short Node의 길이: 400

알고리즘 Algorithm

선택한 알고리즘은 Denovo 방식의 그래프의 eulerian path 알고리즘이다. 수업 시간에 배운 eulerian path 방식은 short read 를 k-mer 로 작게 나눈 후 그래프의 노드가 아닌 간선을 방문하는 경로를 만들기 위해 k-1-mer 로 다시 나누어 debruijn 그래프를 생성을 하고 경로를 탐색한다. 하지만 k-1-mer 로 분할을 할 경우 간선의 길이가 너무 길고 sequence 의 길이가 1 씩 증가되면서 복구되기 때문에 효율성이 떨어진다고 판단을 하였다. 따라서 이를 응용하여 하나의 short read 를 k-mer 이라 설정을 하고 (k-1)-mer 가 아닌 중복되는 부분을 더 줄여 path 를 탐색할 때마다 복구되는 DNA sequence 가 더 길도록 코드를 작성하였다.
알고리즘의 순서는 다음과 같다.

1. My genome으로부터 동일한 길이의 short reads를 중복되는 부분이 존재하도록 분할을 하고 short reads의 길이를 섞는다.
2. Short reads를 L-mer과 R-mer로 분할을 하여 각각을 그래프의 vertex로 만든다.
3. Short reads를 다시 탐색하면서 방문한 edge를 이차원배열에 저장하여 Directed Debruijn Graph를 생성한다.
4. OutDegree 〉 InDegree조건을 사용하여 Eulerian Path의 시작 Vertex를 찾고 Debruijn Graph를 Eulerian Path 방식으로 탐색을 한다.
5. 여러 경로 중 가장 긴 경로를 Reconstructed DNA sequence로 설정을 한다.

   (sequence의 길이가 길어질수록 길이가 동일한 경로는 줄어들고 최종적으로 1가지밖에 나오지

않아 길이를 비교하지 않아도 된다.)

시간복잡도 & 공간복잡도

시간복잡도

: 한붓그리기와 동일한 방식으로 경로를 탐색하는 것이기 때문에 시간복잡도는 O(V)이다.

(V: vetex 의 수)

공간복잡도

: Directed Debruijn Graph 를 이차원배열로 저장하기 때문에 필요한 공간은 O(V^2)
(V: vetex 의 수)

벤치마크와 비교

벤치마크 : brute-force graph path search

알고리즘과 비교하기 위해 선택한 벤치마크는 brute-force graph path search 알고리즘이다. 이 알고리즘은 깊이우선탐색방식(DFS) 방식으로 가능한 모든 경로를 생성해내고 많은 경로 중 하나의 경로만을 선택하는 방식이다. 가장 적합한 경로를 찾기 위해서 모든 vertex를 시작 vertex로 하여 경로를 탐색해야 하므로 시간, 공간적으로 아주 비효율적인 알고리즘이다.

시간복잡도: O(V!)

동일한 조건으로 실행한 결과: 15분~20분

나의 알고리즘 : eulerian path search

수많은 vertex중에서 단 하나의 vertex를 시작 vertex로 선택하기 때문에 좋은 퍼포먼스를 보인다.

시간복잡도: O(V)

시간: 12초

정확도: 100%

알고리즘의 장점 & 단점

장점

- 선형시간에 경로를 탐색할 수 있다.

  시간복잡도 : O(V)

- 정확도가 높다.

복구된 DNA sequence는 My genome과 100% 일치

단점

- 길이가 길어질수록 그래프가 복잡해진다.
- 많은 저장공간이 필요하다.
    - 메모리 공간 부족으로 인하여 my genome의 길이가 짧고 short read의 길이를 길게 할 수 밖에 없었다.

향후 알고리즘 개선 방법

프로젝트를 진행하면서 맞닥뜨렸던 가장 큰 문제는 저장공간 부족이었다. 그래프를 인접 행렬(이차원배열) 형태로 구현을 하여 저장공간을 많이 차지하여 my genome의 길이를 더 길게 하지 못하였고 short read 또한 짧게 할 경우 더 많은 short read가 생성되기 때문에 경로를 탐색하던 중 프로그램이 자동 중단되는 현상이 발생하였다. 따라서 저장공간 문제를 해결하기 위해서 알아본 결과 인접 리스트로 구현을 하면 메모리 공간을 훨씬 절약할 수 있다는 것을 알게 되었다. 인접 리스트는 연결 리스트의 형태로 저장이 되는 것이기 때문에 공간복잡도가 O(m+n)으로 훨씬 효율적이라고 한다. 따라서 인접 행렬을 인접 리스트 형태로 수정한다면 공간 문제를 해결할 수 있을 것이다.