JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 8, pp. 1-8, August 2020
https://doi.org/10.9708/jksci.2020.25.08.001

K-mer Based RNA-seq Read Distribution Method For Accelerating De Novo Transcriptome Assembly

Hwijun Kwon*, Inuk Jung*

*Ph.D candidate, School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea
*Professor, School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea

[Abstract]

In this paper, we propose a gene family based RNA-seq read distribution method in means to accelerate the overal transcriptome assembly computation time. To measure the performance of our transcriptome sequence data distribution method, we evaluated the performance by testing four types of data sets of the Arabidopsis thaliana genome (Whole Unclassified Reads, Family-Classified Reads, Model-Classified Reads, and Randomly Classified Reads). As a result of de novo transcript assembly in distributed nodes using model classification data, the generated gene contigs matched 95% compared to the contig generated by WUR, and the execution time was reduced by 4.2 times compared to a single node environment using the same resources.

▸ **Key words**: Gene Family, De novo transcriptome assembly, Distribution, Acceleration, Classification model, K-mer

[요 약]

본 논문에서는 드노보 전사체 어셈블리의 수행시간을 단축하기 위해 RNA-seq 서열을 유전자계 정보를 활용하여 여러 노드로 분산이 가능한 방법을 제시한다. 제안하는 전사체 서열 데이터 분산 기법의 성능을 측정하기 위해 애기장대의 리드를 4개의 데이터 셋(전체 비분류 리드, 완전 분류 리드, 모델 분류 리드, 무작위 분류 리드)으로 구성하여 실험을 수행하였다. 전체 비분류 데이터와 비교하여 생성된 유전자 콘티그(Contig)는 95% 일치하였고 동일한 리소스들을 사용하는 단일 노드에 비해 본 연구에서 제시하는 분산환경분산 환경 기반의 어셈블리 수행시간은 4.2배 단축되었다.

▸ **주제어**: 유전자 패밀리, 드노보 전사체 어셈블리, 분산 환경, 가속화, 분류 모델, 케이머

• First Author: Hwijun Kwon, Corresponding Author: Inuk Jung
*Hwijun Kwon (fjrzlgnlwns@naver.com), School of Computer Science and Engineering, Kyungpook National University
*Inuk Jung (inukjung@knu.ac.kr), School of Computer Science and Engineering, Kyungpook National University
• Received: 2020. 07. 28, Revised: 2020. 08. 11, Accepted: 2020. 08. 13.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 차세대 염기서열 분석법이 널리 보급되면서 빠르고 낮은 비용으로 유전자를 분석할 수 있게 되었다[1]. 차세대 염기서열 분석법으로 생성된 정보는 유전자가 50~200nt 길이의 염기서열로 전체 유전자의 파편화된 유전자 조각의 염기서열 정보이다. 유전자 조각 정보들은 게놈 어셈블리 기술로 재조립되어 전체 유전자 서열 정보가 된다.

게놈 어셈블리 기술에는 드노보(De novo) 어셈블리 기술이 있다. 드노보 어셈블리는 “처음부터”라는 의미 그대로 알려지지 않은 생물 종의 염기서열을 밝히기 위한 어셈블리 기술이다. 드노보 어셈블리 과정에서 필요한 NGS 데이터는 10Gb 이상이며 정확한 유전자 재조립을 위해 수십 번에서 수백 번 반복하여 어셈블리한다.

이러한 규모의 유전자 단편 데이터를 한 번 어셈블리 하는데 드는 시간이 Trinity 어셈블러를 기준으로 100~1000 시간이 소요된다[2]. 이러한 상황은 밝혀지지 않은 생물 종의 유전자 분석에 대한 진입장벽을 높이는 문제가 될 수 있다. 이를 해결하기 위해 기존의 어셈블러들은 CPU 사용을 극대화할 수 있도록 병렬프로그래밍 방식을 도입하여 수행시간을 단축하였다. 그러나 이러한 병렬프로그래밍 방식은 자체적인 한계와 드노보 어셈블리에 사용되는 데이터의 크기가 커짐에 따라 수행시간 단축이 어려워지고 있다.

본 연구에서는 유전자들의 전사체(Transcript)를 완성하기 위한 전사체 드노보 어셈블리에 드는 시간을 단축하기 위해 대표적 어셈블러인 Trinity를 분산 환경에서 사용할 수 있는 방법을 Fig 1과 같이 제안한다. 이를 위해 유전자 서열을 기반으로 하는 분류 모델을 생성하고 이를 통해 유전자 단편들을 유전자 패밀리 별로 분류한다. 분류된 단편 집합을 분산된 장치에 나누어 드노보 어셈블리를 수행하여 빠르고 신뢰성 있는 전사체 어셈블리를 수행할 수 있도록 한다.

2장에서는 전사체 어셈블리 기술과 수행시간 향상과 관련된 연구에 관해 서술하고 3장에서는 실험에 사용된 데이터를 설명한다. 4장에서는 분류 모델 생성 방법, 입력 데이터 전처리, 분류과정에 관해 설명하고 5장에서는 생성된 모델을 4장에서 제안하는 방식을 통해 실험한 결과를 보여주고 있다. 마지막으로 6장에서는 연구 결과에 대한 결론과 한계, 향후 연구 계획에 관해 서술하며 끝맺는다.

II. Related Works

현재까지 다양한 전사체 어셈블러들이 개발 됐으며 널리 사용되고 있는 10개의 전사체 어셈블러를 비교한 최근 연구가 수행됐다[3]. 해당 연구에서는 절대적으로 우수한 어셈블러는 없음을 보고하였으며, 그 이유로는 수많은 전사체들의 복잡한 서열 특성을 하나의 어셈블러에서 모두 고려하기가 어려우며 전사체 어셈블리 알고리즘의 향상 여지가 있음을 논하였다.

이에 반해 앞서 언급한 어셈블리 시간 단축에 관한 연구는 비교적 적은 실정이다. 즉, 어셈블리 시간을 단축하기 위한 가속화 기술 또는 분산 환경 연구가 부재한 실정이다. 어셈블리 속도를 단축시키기 위해 대게 GPU 또는 FPGA 하드웨어 활용으로 서열 정렬 과정을 가속하거나[4][5] 어셈블리 과정의 일부를 분산 환경에서 구현하고자 하였다[6]. 그러나 [7] 연구에서 보여주듯이 여러 단계로 구성된 복잡한 어셈블리 과정 전체가 효율적으로 가속화가 될 수 없음을 보여주고 있다. 예를 들어, 케이머 카운팅 문제는 비교적 맵리듀스 기술을 통해 가속화가 가능함을 보였다[8]. 그러나 드 브루인 그래프 기반의 콘티크 확장과 같이 많은 메모리를 필요로 하는 어셈블리 과정 단일 노드에서 처리가 돼야 하는 제약 조건으로 병렬화가 되기 매우 어렵다.

드노보 어셈블리를 가속하는 방법에는 연산장치의 하드웨어 수를 늘려 작업을 처리하는 스케일 업(Scale-up) 방식과 같은 성능의 연산장치를 분산 환경에서 작업하는 방식의 스케일 아웃(Scale-out) 방식이 있다.

본 논문에서는 가장 널리 사용되고 있는 Trinity 전사체 어셈블러를 기반으로 전사체 어셈블리 가속화를 위한 방법을 제시한다. Trinity의 경우 각 모듈이 병렬화 되어있다. Trinity 어셈블러는 Inchworm, Chrysalis, Butterfly 3가지의 모듈로 구성된 프로그램이다. Inchworm에서는 유전자 파편을 조립하여 콘티그라는 긴 염기서열을 만들고 Chrysalis에서는 서로 일치하는 부분이 많은 콘티그들을 클러스터링하여 드 부루인 그래프(De Bruijn Graph)를 생성한다. 마지막으로 Butterfly에서 생성된 드 부루인 그래프에서 만들어지는 경로를 생성하여 조립된 콘티그를 형성하여 최종적으로 원본 유전자들을 재조립한다[9]. 하지만 코어 수의 증가에 따라 속도 향상량이 점점 감소하는 모습이 확인되었고 병렬화가 불가능한 모듈이 존재하였다[10][11].

스케일 아웃 방식으로 드노보 어셈블리의 속도를 향상하는 방법으로는 어셈블리 과정에서 생성되는 유전자 그래프인 드 브루인 그래프를 분산 환경에서 생성하는 방식이 있다. 대표적으로 구글의 Pregel 프레임워크를 통해 생성한 어셈블러를 이용한 연구가 있다[12]. 하지만 전체 어셈블리 과정에서 드 브루인 그래프를 생성하는 과정은 전체의 30~40% 차지하므로 성능향상에 한계가 있다.

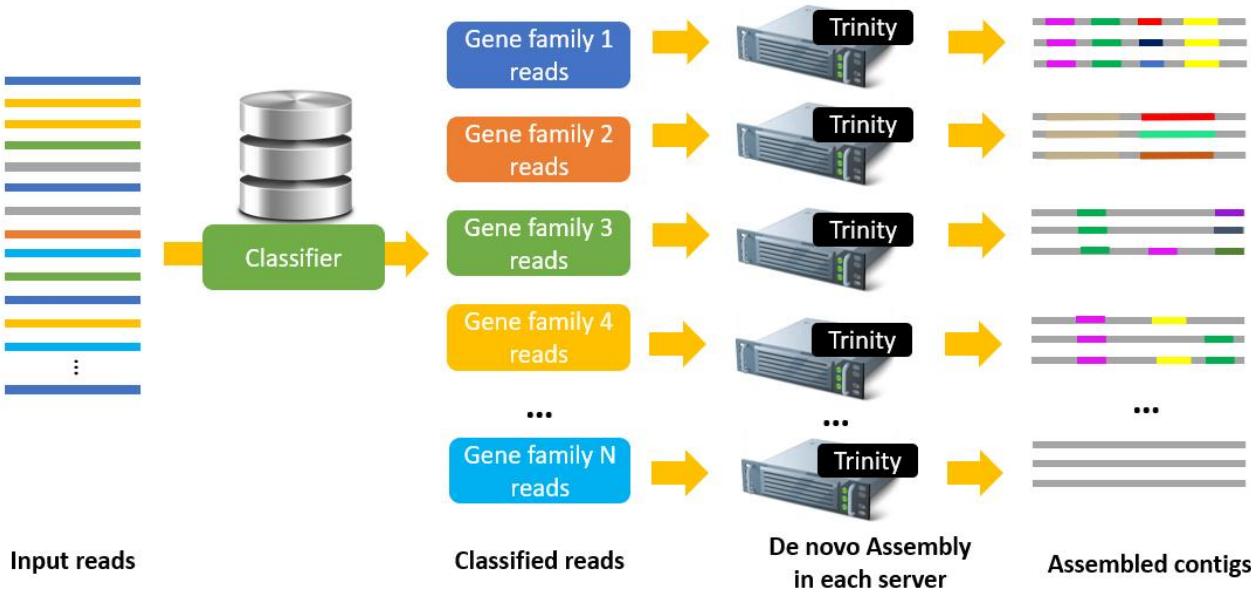

Fig. 1. Architecture Of K-mer Based Classification Model

III. Data

본 연구에서는 실험을 위한 데이터로 애기장대의 전사체 (RNA-seq) 데이터를 사용하였다. 애기장대는 식물 발달 및 생리학 연구에 주로 사용되는 모델 식물로서 염색체 수가 적고 개체의 생활 주기가 짧은 특징으로 유전자에 대해 가장 많이 연구되었다[13]. 이러한 이유로, 애기장대의 유전자 데이터는 다양한 기관과 프로젝트에서 연구되어 그 데이터의 양이 많고 검증된 데이터가 많아 실험에 적합하여 선정하였다.

1. 참조게놈서열

참조게놈서열은 어셈블리에서 Reference라고 불리며 염색체 전체의 DNA 염기서열이다. 본 연구에서는 The Arabidopsis Information Resource (TAIR) 사이트에서 제공하는 TAIR10 버전의 애기장대의 참조게놈서열 데이터를 사용하였다.

어노테이션(Annotation)은 유전자 정보를 가지고 있는 데이터로서 참조게놈서열의 특정 영역에 있는 모든 유전자 정보를 기록하고 있다. 앞서 참조게놈서열에 의해 맵핑된 리드를 유전자 영역별로 추출하기 위해 사용한다. 어노테이션은 참조게놈서열과 같은 버전인 TAIR10을 사용하였다.

2. Gene Sequence Data

어셈블리 대상이 되는 전사체로는 단백질 생성을 하는 단백질 코딩 유전자들의 서열들을 사용하였다. 참조게놈서열와 어노테이션 정보만 있다면 유전자 정보를 추출할 수 있기에 불필요하다고 생각될 수 있다. 그러나 참조게놈서열와 어노테이션 정보는 대표적인 염기서열 데이터와 유전자에 대한 정보이고 실제 개체 간 차이가 있다. 예를 들어 AT1G46912 유전자 영역에는 AT1G46912.1 또는 AT1G46912.2의 유전자가 존재할 수 있다. 이러한 작은 서열 차이에 대한 정보를 모두 포함하여 분류 모델을 생성하여야 더욱 정확하게 리드를 분류할 수 있으므로 유전자 서열 데이터는 이러한 서열 차이를 포함하는 동일 유전자를 포함한다. 또한, 유전자 서열 데이터는 cDNA 염기서열 데이터로서 DNA 내에서 단백질을 생성하는 핵심적인 부분의 정보만을 담고 있으므로 이러한 유전자 서열 데이터를 사용하여 분류 모델을 생성하였다.

본 연구에서는 Araport 11 버전의 cDNA 유전자 서열 데이터를 사용하였다.

3. Input Data

리드란 NGS 결과 데이터로서, 시퀀싱 대상이 되는 DNA 서열을 짧은 길이로 읽은 파편화된 서열들이다[1]. 이러한 리드들은 기본적으로 차세대 시퀀싱의 특성상 왼쪽 리드와 오른쪽 리드 두 개로 구성되어있고 그 길이는 50~200nt 이다.

전사체 어셈블리 수행을 위한 RNA-seq 데이터는 SRA (Sequence Read Archive) 데이터베이스에서 수집하였다.

본 연구에서는 이 SRA 플랫폼에서 공개한 리드 중 Access number가 SRX5525170인 리드를 사용하였다 [14]. SRX5525170 리드는 cDNA를 대상으로 시퀀싱 된 길이가 49이고 약 6830만 쌍의 좌, 우 리드를 가진 데이터

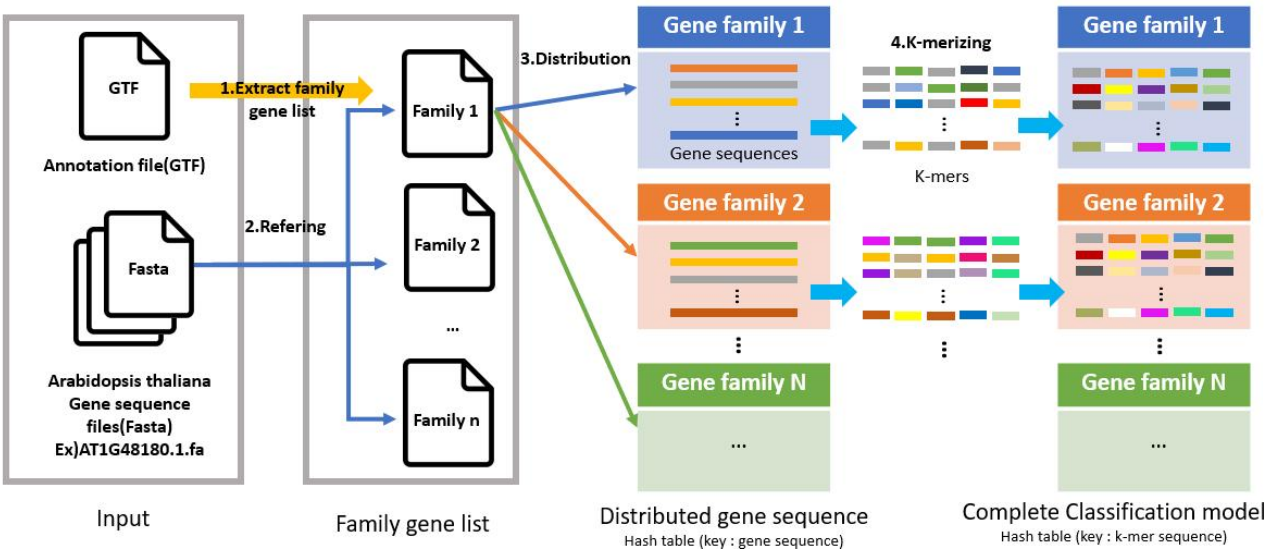

Fig. 3. Constructing a K-mer Based Gene Family Classification Model

이다(Fasta file 기준, 11.6G가량의 용량).

4장에서 구현한 모델의 실험 및 성능 측정을 위해 전체 리드 데이터에서 애기장대가 가진 10개의 유전자 패밀리 서열에 매핑 되는 일부의 리드를 추출하여 약 880만 쌍의 리드가 실험에 사용되었다. 선택된 유전자 패밀리는 Fig 2와 같으며 유전자 수가 많은 순으로 유전자 패밀리 10개를 선정하였다.

추출한 리드 데이터를 Whole Unclassified Read (WUR), Model-classified Read (MCR), Family-Classified Read (FCR), Random Read (RR) 4개의 그룹으로 분류하였다.

3.1. Whole Unclassified Reads (WUR) Data Set

전체 리드 중 앞서 선택된 10개의 유전자 패밀리에 속한 유전자 서열들에 매핑 되는 리드 전체 집합이다. MCR에 대한 대조군 실험 데이터로서 분류되지 않은 리드 데이터셋이다.

3.2. Family-Classified Read (FCR) Data Set

WUR 데이터셋을 패밀리 별로 분류하여 저장한 데이터셋이다. 분류 모델이 WUR을 이상적으로 분류한 결과가 FCR이 될 수 있다.

3.3. Model-Classified Read (MCR) Data Set

WUR 데이터를 분류 모델을 이용하여 분류한 데이터셋. 10개의 패밀리에 대해 분류하고 분류되지 않는 리드들은 따로 저장한다. WUR 데이터셋에 대한 실험군 데이터로서 분류 모델에 의해 분류된 리드 데이터셋이다.

3.4. Random Read (RR) Data Set

WUR에 속하는 임의의 리드를 일정한 개수(WUR Read 개수의 10%)로 구성된 데이터이다. 마지막으로 10개의 무작위로 분산된 리드 (Random Read - RR)로 구성된 그룹이 있다.

IV. Method

본 논문에서 제안하는 전사체 데이터 분류를 수행하기 위해 먼저 유전자 서열 데이터를 분류 모델을 생성하고 리드들을 모델에 입력할 수 있도록 전처리한 뒤 모델에 입력한다.

```
197 EF-hand containing proteins
211 C2H2 Transcription Factor Family
255 Cytoplasmic ribosomal protein gene family
261 Cytochrome P450
279 Organic Solute Cotransporters
307 Receptor kinase-like protein family
362 Glycosyltransferase Gene Families
379 Glycoside Hydrolase Gene Families
444 Miscellaneous Membrane Protein Families
610 Acyl Lipid Metabolism Family
```

Fig. 2. Selected Gene Family

1. Constructing a K-mer Based Gene Family Classification Model

분류 모델을 생성하기 위해서는 어노테이션과 유전자 서열 데이터가 필요하다. 어노테이션에서 각각의 패밀리에 속한 유전자 id를 추출하여 각각의 패밀리 별로 속한 유전자 list를 생성한다. 그리고 유전자 list에 속한 유전자 id를 유전자 서열 데이터에서 찾아 패밀리 별로 유전자 서열

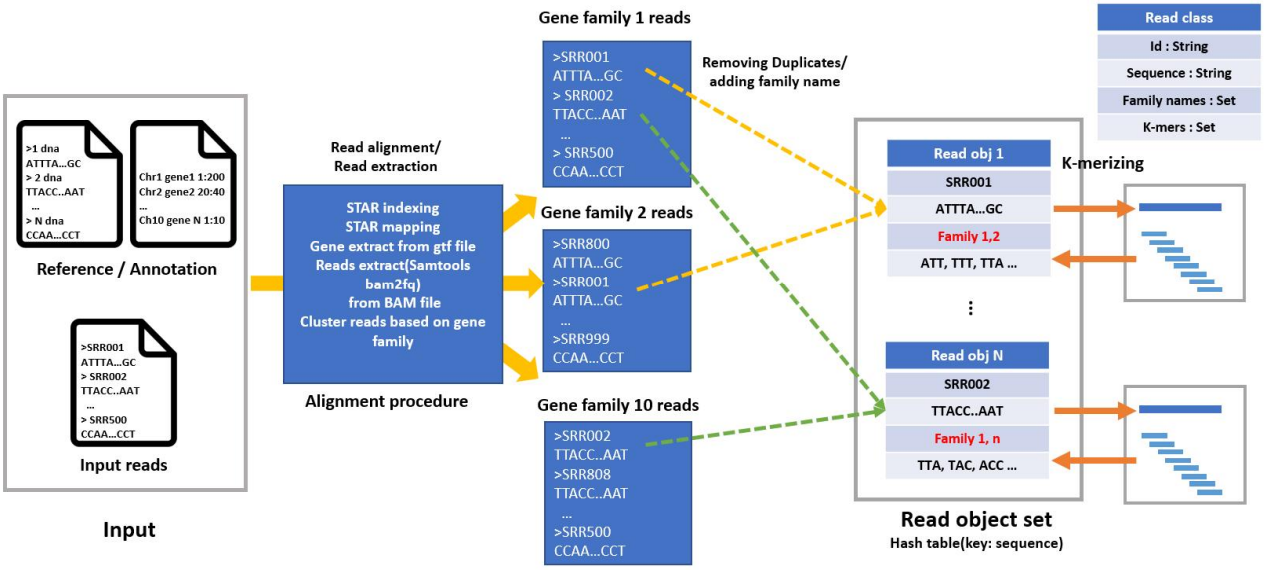

Fig. 4. Processing Input Read Data Procedure

데이터들을 해시테이블에 저장한다. 저장된 염기서열 데이터는 원본 염기서열이므로 상보 서열(Complement sequence), 역 서열(Reverse sequence), 역 상보 서열(Reverse-complement)들을 추가한다. 다음으로 패밀리별로 저장된 염기서열 데이터를 케이머화(K-merizing) 한다. 케이머화란 Fig 4와 같이 염기서열 데이터를 길이 K를 가지는 작은 부분 서열 데이터로 만들어 주는 작업을 의미하고 생성된 부분 서열을 케이머(K-mer)라 지칭한다. 이렇게 생성된 케이머들은 중복을 제거하기 위해 패밀리 별로 생성된 해시테이블에 저장한다.

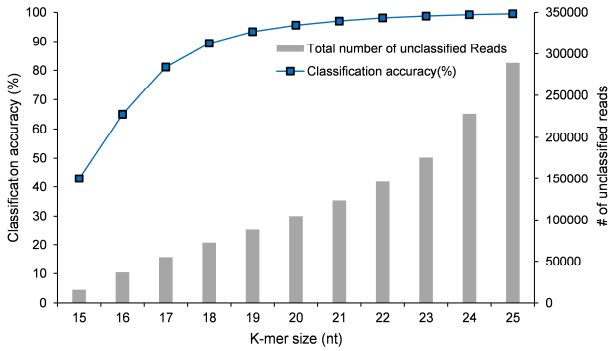

Fig. 5. Classification Performance

2. Processing Input Read Data

이 과정은 앞서 언급한 리드 데이터셋 그룹 중 WUR을 생성하기 위한 단계로 STAR Aligner를 통한 리드 Alignment/Extraction 과정과 리드를 객체화하고 생성된 리드 객체를 초기화하여 해시테이블에 저장하는 과정으로 구성된다.

이 과정을 통해 생성된 리드 객체 집합은 WUR이 되고 WUR로부터 태그된 패밀리 별로 분류하여 FCR을 생성할 수 있다. MCR은 앞서 생성한 분류 모델을 통해 리드 분류에서 생성되고 RR의 경우 단순히 WUR에서 임의로 리드를 나누어 주는 방식으로 생성할 수 있다.

입력으로 참조게놈서열, 어노테이션, 리드 데이터셋이 사용된다. 입력 리드들은 STAR Aligner를 통해 참조게놈서열에 매핑 되고 Binary Sequence Alignment/Map(BAM) 파일이 결과로 생성된다. 생성된 BAM 파일에서 어노테이션에 명시되어 있는 앞서 선택한 유전자 패밀리의 유전자 영역을 기반으로 리드를 추출하면 유전자 패밀리 별로 리드를 추출할 수 있다. 추출된 리드들은 리드 클래스 객체로 초기화되어 저장된다.

3. Reads Classification Using Classification Model

처리된 리드 데이터들은 앞서 생성된 분류 모델에 의해 분류된다. 리드의 케이머와 패밀리 케이머가 일치하면 해당 리드는 그 패밀리에 분류된다.

V. Result

제안하는 분류 모델이 앞서 제시한 문제들에 대해 얼마나 잘 해결할 수 있는지 분석하였다. 핵심 1절에서는 하이퍼파라미터인 케이머 크기를 변화하며 분류 모델의 분류 성능을 측정하였다. 2절에서는 본 문제에서 중점적으로 해결하고자 하는 실행속도에 중점을 맞추어 성능평가를 하였고 3

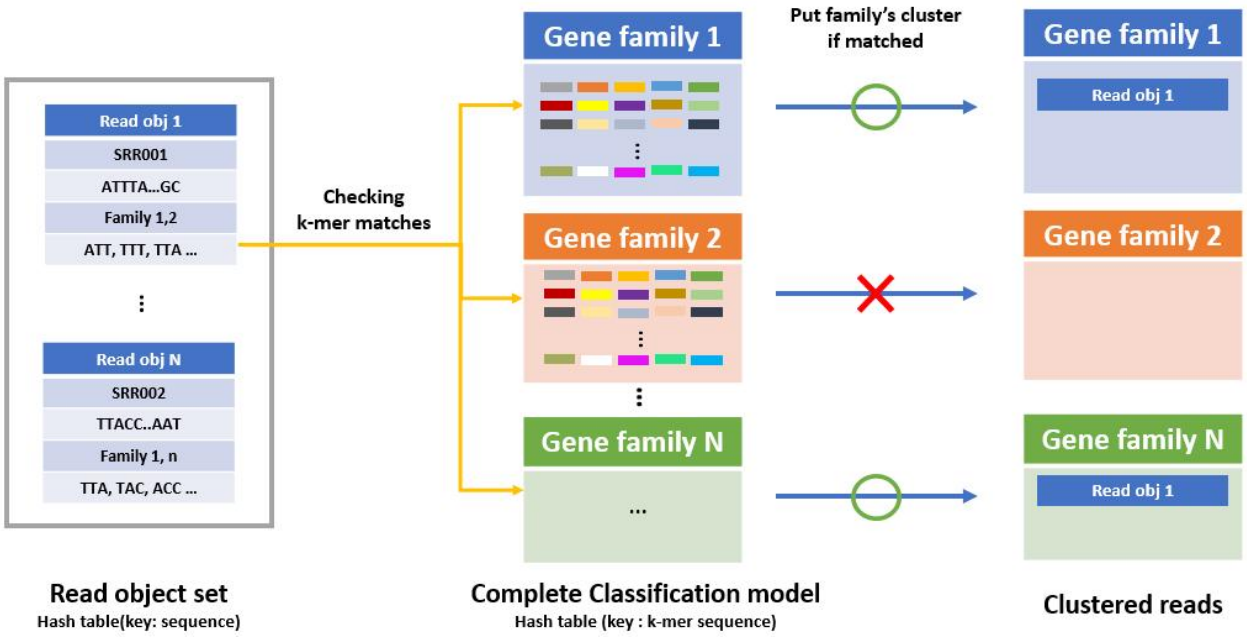

Fig. 6. Reads Classification Using Classification Model

절에서는 분류된 리드가 생성한 결과 염기서열 데이터가 얼마나 원본 염기서열 데이터와 일치하는지 비교한다.

Table 1. Trinity Runtime For Each Family

Gene Family	Runtime (seconds)			
	WUR	FCR	MCR	RR
EF-hand containing proteins	1632	104	109	92
C2H2 Transcription Factor		80	95	
Cytoplasmic ribosomal protein Gene Family		344	307	
Cytochrome P450		78	91	
Organic Solute Cotransporters		149	152	
Receptor kinase-like protein Family		119	127	
Glycosyltransferase Gene Families		215	193	
Glycoside Hydrolase Gene Families		220	199	
Miscellaneous Membrane Protein Families		240	252	
Acyl Lipid Metabolism Family		412	386	

1. Classification Performance

분류 모델의 분류 성능을 측정하기 위해 분류 모델의 케이머 크기를 변화시키며 리드의 분류 정확도를 측정하였다. 케이머 크기 20부터 95% 이상의 분류 능력을 확인할 수 있었다. 케이머 크기가 작아지게 되면 서로 다른 패밀리에서 같은 케이머를 보유하게 되어 원래의 패밀리가 아닌 다른 패밀리에도 리드가 할당되어 분류 성능이 저하되는 결과를 확인할 수 있었다.

2. Runtime Comparison

3장에서 정의한 4개의 데이터셋에 대해 수행시간을 비교하였다. 구현 환경은 리눅스 우분투 16.04 운영체제, 32 CPU Core, 256Gb 메모리이며 분류 모델 생성 및 분산처리에 사용된 언어는 Python 3.7이다. 사용된 어셈블러는 Trinity 2.9.1버전이다.

WUR 데이터셋을 사용하여 890만 쌍의 리드 1.2Gb 전체를 Trinity에 memory 20Gb, CPU 10 core를 할당하여 수행한 결과 Linux Time 모듈의 Real time 기준으로 27분 12초가 소요되었다.

FCR 데이터셋은 패밀리 별로 리드가 나뉘어 있으므로 각각의 패밀리 리드 데이터셋 별로 Trinity를 같은 환경의 노드에서 수행하였다. 그 결과 각각의 패밀리 별 수행시간은 Table 1과 같다. 전체 노드는 동시에 수행되기 때문에 전체 노드 중 최대 수행시간을 가지는 노드의 수행시간이 작업 완료 시간이므로 412초가 소요된 것을 확인할 수 있다. 이는 WUR 데이터셋 처리 속도보다 약 4배 빠르다.

Table 2. The Number Of Matched Gene ID

	WUR	FCR	MCR	RR
WUR	1329	-	-	-
FCR	1299	1358	-	-
MCR	1307	1317	1347	-
RR	352	353	356	410

MCR (K-mer=20) 기반으로 분류한 결과, Trinity 수행 시간은 각 패밀리 별로 91~386초가 소요되었으며 분산 환경에서 전체 작업의 최종 수행시간은 가장 오래 걸린 노드의 수행시간 386초를 기준으로 WUR 데이터의 수행시간에 비해 약 4.2배 빨랐다.

RR 데이터셋의 경우 전체 수행시간이 92~96초의 영역에 분포하였다.

3. Quality Assessment

Trinity로 생성된 Assembled 콘티그 들을 실제 애기장대 유전자와 비교하여 각각의 데이터셋이 유전자 파편들을 잘 조립하였는지 확인하는 과정이다.

결과의 질적 평가를 위해 Blastx 프로그램을 활용하여 애기장대의 실제 유전자와 얼마나 일치하는 유전자를 생성하였는지 확인하였다. 사용된 Blastx는 버전 2.10이며 옵션값 “-evalue 1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6”을 사용하였다. 사용된 DB는 Araport11 버전의 Arabidopsis Thaliana Protein 데이타베이스를 사용하였다.

Blastx 결과 파일은 Trinity 결과를 데이타베이스와 비교하여 일치하는 단백질서열을 열거한다. Table 2은 각각의 데이터셋에 대한 일치된 유전자 개수와 데이터셋간의 공통 유전자 개수이다. WUR과 FCR이 97.7%, MCR이 98.3%, RR 26.5%의 유사도를 보였다. 그리고 Table 3은 생성된 RNA 염기서열의 개수와 염기서열의 길이에 대한 지표이다. WUR과 FCR, MCR은 N10~50지점에서 약 95% 이상의 일치함을 확인할 수 있었다. N10~50 이란 전체 게놈 길이의 10~50%로 가장 짧은 콘티그의 염기서열 길이로 정의한다[15].

Table 3. Status Of Transcripts

	WUR	FCR	MCR	RR
# of transcripts	2892	3078	3055	2938
N10	1121	1135	1163	582
N20	943	907	892	472
N30	811	789	770	405
N40	674	669	637	357
N50	572	552	534	317
Median	348	348	334	276
Average	462.96	458.09	448.32	318.59

VI. Conclusions

본 연구에서는 전사체 어셈블리 속도 향상을 위해 유전자의 케이머를 기초로 분류 모델을 설계 및 구현하였다. 또한, 애기장대 데이터를 4개의 그룹으로 나누어 분류 모델에서 실험하였다. 실험결과 분류된 리드를 사용한 분산 환경에서의 드노보 어셈블리 과정이 약 4배 정도의 속도 향상이 있었다. 그리고 분류 모델에서 분류된 리드를 어셈블하여 생성된 유전자가 기존의 방식으로 생성된 유전자와 비교하여 약 98%로 대부분이 일치하고 생성된 유전자의 길이가 비슷한 것을 확인할 수 있었다. 이를 통해 본 연구에서 제안하는 케이머 기반 리드 분류 모델이 드노보 어셈블리를 기존의 스케일 업 방식의 어셈블리 방식과 같은 신뢰성을 가지며 더 빠르게 어셈블리할 수 있음을 증명하였다.

본 연구에서는 개발한 분류 모델은 드노보 어셈블리 시에 알려진 유전자를 기반으로 생성하였다. 하지만 드노보 어셈블리의 특성상 알려지지 않은 종이 알려진 유전자를 가지지 않을 수 있다. 현재 제안하는 분류 모델은 이러한 상황에 대한 처리가 명확하지 않은 한계가 있다.

향후 연구에서는 앞서 연구 한계로 언급한, 알려지지 않은 유전자를 가지는 생물 종의 드노보 어셈블리를 진행함과 동시에 본 연구에서 제안하는 분류 모델이 낼 수 있는 성능을 확인할 것이다. 또한, 머신러닝을 통한 분류 모델을 생성하여 각각의 패밀리에 대한 노드들을 학습하여 오차율을 줄이는 방법에 관한 연구를 진행할 예정이다.

ACKNOWLEDGEMENT

This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

REFERENCES

[1] Mardis, Elaine R. "The impact of next-generation sequencing technology on genetics." Trends in genetics, Vol. 24, No. 3, pp. 133-141, Mar 2008, DOI: 10.1016/j.tig.2007.12.007

[2] Robert Henschel, Matthias Lieber, Le-Shin Wu, Phillip M. Nista, Brian J. Haas, and Richard D. LeDuc, “Trinity RNA-Seq assembler performance optimization”, In Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond (XSEDE

'12). Association for Computing Machinery, New York, NY, USA, Article 45, pp. 1-8, Jul 2012, DOI: 10.1145/2335755.2335842

[3] Hölzer, Martin, and Manja Marz. "De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers." GigaScience, Vol. 8, May 2019, DOI: 10.1093/gigascience/giz039

[4] Goswami, Sayan, et al. "Gpu-accelerated large-scale genome assembly." 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, May 2018, DOI: 10.1109/IPDPS.2018.00091

[5] Varma, B. Sharat Chandra, et al. "FAssem: FPGA based acceleration of de novo genome assembly." 2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines. pp. 173-176, Apr 2013, ,DOI: 10.1109/FCCM.2013.25.

[6] Ellis, Marquita, et al. "diBELLA: Distributed long read to long read alignment." Proceedings of the 48th International Conference on Parallel Processingm, Num 70, pp. 1-11, Aug 2019, DOI: 10.1145/3337821.3337919

[7] Henschel, Robert, et al. "Trinity RNA-Seq assembler performance optimization." Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond , Jul 2012, DOI: 10.1145/2335755.2335842.

[8] Haas, B., Papanicolaou, A., Yassour, M. et al, "De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis", Nature Protocols 8, pp. 1494-1512, Jul 2013, DOI: 10.1038/nprot.2013.084

[9] Kim, C.S., Winn, M.D., Sachdeva, V. et al. "K-mer clustering algorithm using a MapReduce framework: application to the parallelization of the Inchworm module of Trinity", BMC Bioinformatics 18, Nov 2017, DOI: 10.1186/s12859-017-1881-8

[10] Zhao, Q., Wang, Y., Kong, Y. et al. "Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study", BMC Bioinformatics 12, Dec 2011, DOI: 10.1186/1471-2105-12-S14-S2

[11] Wagner, Michael & Fulton, Ben & Henschel, Robert. "Performance Optimization for the Trinity RNA-Seq Assembler", Tools for High Performance Computing 2015, pp. 29-40, Jan 2016, DOI: 10.1007/978-3-319-39589-0_3.

[12] D. Yan, H. Chen, J. Cheng, Z. Cai and B. Shao, "Scalable De Novo Genome Assembly Using Pregel," 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, pp. 1216-1219, Jan 2018, DOI: 10.1109/ICDE.2018.00114.

[13] Lamesch P, Berardini TZ, Li D, et al, "The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools", Nucleic Acids Res, pp. D1202-D1210, Jan 2012, DOI:10.1093/nar/gkr1090

[14] NCBI SRA database(Arabidopsis Thaliana), https://www.ncbi.nlm.nih.gov/sra/SRX5525170%5baccn%5d

[15] Manchanda, N., Portwood, J.L., Woodhouse, M.R. et al. "GenoMeQC: a quality assessment tool for genome assemblies and gene structure annotations", BMC Genomics 21, No 193, Mar 2020, DOI: 10.1186/s12864-020-6568-2

[16] Saw, A.K., Raj, G., Das, M. et al., "Alignment-free method for DNA sequence clustering using Fuzzy integral similarity", Scientific Reports volume 9, Num 3753, Mar 2019, DOI: 10.1038/s41598-019-40452-6

[17] Bedre, R, Mandadi, K., "GenFam; A web application and database for gene family-based classification and functional enrichment analysis", Plant Direct, Vol. 3, pp. 1- 7, Dec 2019, DOI: 10.1002/pld3.191

[18] Chabikwa, T.G., Barbier, F.F., Tanurdzic, M. et al. "De novo transcriptome assembly and annotation for gene discovery in avocado, macadamia and mango.", Nature, Scientific Data vol. 7, Num. 9, Jan 2020, DOI: 10.1038/s41597-019-0350-9

[19] Seokjun Seo, Minsik Oh, Youngjune Park, Sun Kim, "DeepFam: deep learning based alignment-free method for protein family modeling and prediction", Bioinformatics, Vol. 34, Num 13, pp. 254-262, Jul 2018, DOI: 10.1093/bioinformatics/bty275

[20] Weizhong Li, Limin Fu, Beifang Niu, Sitao Wu, John Wooley, "Ultrafast clustering algorithms for metagenomic sequence analysis", Bioinformatics, Vol. 13, Num. 6, pp. 656-668, Nov 2012, DOI: 10.1093/bib/bbs035

Authors

Hwijun Kwon received the B.S. and M.S. degrees in School of Computer Science and Engineering Kyungpook National University Korea in 2016, 2018, respectively. He is currently taking PhD course in Computer Science and Engineering from Kyungpook National University. He is interested in bioinformatics, de novo assembly Machine learning.

Inuk Jung received the Ph.D. degree in bioinformatics from Seoul National University, Korea, in 2017. He is currently an assistant professor at Kyungpook National University. His current research interests include bioinformatics, big data analysis and machine learning.