|  |
| --- |
| **[소프트웨어 학과]** |
| [Intron-retention 탐색을 위한  효율적인 알고리즘 개발 및 연구] |
| **[중간보고서]** |
| **Version 1.0** |
| **2022 10. 26.** |

학번: 201820736

이름: 정민규

지도교수: 조다정

목 차

[요약 3](#_Toc117804299)

[1. 개요 4](#_Toc117804300)

[1.1 연구 배경 4](#_Toc117804301)

[2. 관련 선행 연구 조사 4](#_Toc117804302)

[2.1 Comprehensive characterisations of intronic mis-splicing mutations in human cancers [1] 4](#_Toc117804303)

[2.1.1 요약 4](#_Toc117804304)

[2.1.2 살펴볼 내용 4](#_Toc117804305)

[2.2 FRASER (Find RAre Splicing Events in RNA-seq) [2] 5](#_Toc117804306)

[2.2.1 요약 5](#_Toc117804307)

[2.2.2 살펴볼 내용 5](#_Toc117804308)

[2.3 iREAD: a tool for intron retention detection from RNA-seq data [4] 6](#_Toc117804309)

[2.3.1 요약 6](#_Toc117804310)

[2.3.2 살펴볼 내용 6](#_Toc117804311)

[2.4 Schema for Parallel Insertion and Deletion: Revisited [5] 7](#_Toc117804312)

[2.4.1 요약 7](#_Toc117804313)

[2.4.2 살펴볼 내용 7](#_Toc117804314)

[3. 문제 정의 7](#_Toc117804315)

[3.1 문제 정의 과정 7](#_Toc117804316)

[3.1.1 설계에 관한 고민 7](#_Toc117804317)

[3.1.2 문제 정의와 해결 방법 접근에 관한 고민 8](#_Toc117804318)

[3.1.2.1 접근법 1 8](#_Toc117804319)

[3.1.2.2 접근법 2 8](#_Toc117804320)

[3.2 문제 정의 9](#_Toc117804321)

[4. 추후 진행 계획 10](#_Toc117804322)

[참고자료 10](#_Toc117804323)

# 요약

This project attempts to present an efficient intron retention search algorithm by defining and modeling the problem from a computational theoretical perspective for intron retention, one of the mis-splicing mutations. First, mis-splicing, which occurs during splicing in mRNA, refers to the failure to splice introns except for exons, which have important genetic information on proteins in which exons and introns exist during the transcription process. In other words, the exon that should remain is cut or the intron that should be removed remains. Each of these is called exon skipping and intron retention. In this study, we understand and use existing studies. We look at the way from a biological perspective and look at how they got the data and how they approached it. We also look at the study of string insertion and deletion from a computational theoretical perspective, and devise how to use them and more efficient algorithms.

As a previous studies, research on mis-splicing mutation, tools such as FRASER and iREAD for intron retention detection, and manipulation of strings from a computational theoretical point of view were investigated. Through this process, we were able to define the problem and get hints on how to approach it. The problem is defined as follows - "A computational theoretical model design that receives genes after splicing as input (i.e., genes after splicing operation is completed by a transcription) and determines whether intron retention has occurred in the corresponding genes.". This is the same as 'a computational theoretical model design in which can be said to have occurred intron retention if it is determined that a pattern of more than a certain length () is in by pattern matching the pattern of and '.

So far, we've redefined the problem as above and are at the stage of thinking about how to approach it and looking at how it can be used in real data. After the interim report, the model is designed, the model is simply implemented in python, and then model will be examined with the existing intron retention detection tools (FRASER, iREAD, etc.) how accurate it has and what it misses. And if the model is completed to some extent, we will investigate how we can visually show this algorithm. Through this, it aims to help the general public understand this algorithm.

# 1. 개요

## 1.1 연구 배경

본 연구는 mis-splicing mutation들 중 하나인 intron retention에 관하여 계산 이론 관점으로 문제를 정의하고 모델링하여 효율적인 intron retention 탐색 알고리즘을 제시하려 한다. 먼저 mRNA에서 splicing 중에 일어나는 mis-splicing은 전사과정에서 exon과 intron이 존재하는 단백질에 중요 유전 정보를 가지고 있는 exon을 제외한 intron을 splice하는 것에 실패한 것을 말한다. 즉 남아야 하는 exon이 잘리거나 혹은 제거 되야 하는 intron이 남아있는 경우이다. 각각 exon skipping, intron retention이라고 말한다. 본 연구에서는 기존에 수행되었던 연구들과 과제들을 이용한다. 생물학 관점에서 접근했던 방식을 보며 데이터를 구하고 어떻게 접근을 했는지 살펴본다. 또한 계산 이론적 관점에서 string insertion, deletion에 대한 연구를 살펴보고 어떻게 이용하며 더욱 효율적인 알고리즘을 구상한다.

현재까지 진행된 내용은 문제를 재정의했고 어떻게 접근할 것인지에 대한 구상들을 하며 실제 데이터에서 어떻게 사용가능한지 살펴보고 있는 단계에 있다.

# 2. 관련 선행 연구 조사

## 2.1 Comprehensive characterisations of intronic mis-splicing mutations in human cancers [1]

### 2.1.1 요약

본 논문에서는 intronic mis-splicing mutation을 종합적으로 구분한다. 이를 위하여 3022개의 정상 대조군과 함께 1134개의 pan-cancer 전체 genome들과 전사체를 분석을 진행했다. Splicing 분석은 rate based로 진행되었으며 그 결과로 678개의 somatic intronic mutation이 발생했으며 309개의 deep intronic single nucleotide mutation이 85개의 deep intronic mutation 등을 확인했다. 또한 이 논문의 결과는 deep intron들의 splicing code들에 작용하는 noncoding mutation이 종양 발생에 기여한다는 이전에 인식하지 못했던 메커니즘을 밝히는 것이다.

### 2.1.2 살펴볼 내용

본 논문을 통하여 정의들과 새로운 지식을 얻을 수 있었다. 먼저 Deep intronic은 실제 junction에서 20 bp(base pair)이상인 것이며 proximal intronic은 실제 junction에서 3~20 bp 내에 속한 것들을 말한다. 위 결과 상에서Abnormal splicing (mis-splicing mutation)들은 exon-skipping, partial intron retention, partial exon skipping 순서로 빈번히 일어났다. 여기서 대립유전자 특이성을 splicing event들의 검증으로 사용했다. 그리고 proximal intronic mis-splicing mutation의 특성으로 SS(splice site)의 donor와 acceptor을 위한 canonical dinucleotides인 GT와 AG에서 proximal intronic이 미치는 locational effect가 가장 큰 것으로 나타났다. 특히 partial exon skipping과 관련된 돌연변이는 acceptor SS에서 donor SS보다 더 자주 발생한다. 이는 cryptic acceptor들이 cryptic donor보다 partial exon skipping이 더 자주 일어난다는 것에 대한 동의라고도 볼 수 있다. 또한 acceptor의 strength이 돌연변이 형성에 영향을 끼친다는 것을 알 수 있었다. 이후 deep intronic splicing mutation의 특성에 대한 설명과 실험 데이터를 어떻게 정제했는지, ratio based splicing 분석에 대한 설명과 대립유전자 특성에 따라 검증하는 내용이 본 논문에 있다.

## 2.2 FRASER (Find RAre Splicing Events in RNA-seq) [2]

### 2.2.1 요약

본문의 연구 동기는 다음과 같다. Splicing에 영향을 미치는 유전적 변이들은 희귀질환의 주요 요인이며 이를 식별하는 것은 도전과제로 남아있다는 것이다. 하지만 RNA-seq 데이터에서 aberrant splicing 현상을 탐지하는 전문화된 방법은 존재하지 않았다고 한다. 따라서 FRASER (Find Rare Splicing Events in RNA-seq)라는 분석 방법을 제시한다. FRASER은 splice site를 감지하여 alternative splicing과 intron retention을 평가하며 denoising autoencoder을 이용하여 잠재적 교란 요인 (latent confounder)을 조절할 수 있다. 이에 대한 추정결과를 over-dispersed count fraction distribution을 이용하여 제공한다. 이러한 FRASER은 GTEx dataset의 48개 조직에서 희귀한 near-splice site mutation에 대해서도 잘 작동한다. 지금까지 분석된 희귀 질병 dataset에 대해서 응용을 하면 TAZ의 aberrant exon truncation을 재정렬하는 것으로 새로운 진단을 할 수 있다. 위를 종합하면 FRASER이 희귀 질병에 대한 RNA-seq 기반 진단을 위해 중요한 도구로서 사용될 수 있다고 생각한다.

### 2.2.2 살펴볼 내용

본문에서 살펴볼 내용으로 FRASER은 splice metrics(splice site 5’, 3’을 본문에 있는 식을 사용하여 행렬로 표현한 것)한 것과 가중치(intron retention에 대한 alternative donor이 사용된다.)를 이용하여 exon skipping에서 어떠한 종류의 aberrant splicing event를 찾아낼 수 있게 하는 과정을 볼 필요가 있다. 또한 앞서 말한 aberrant splicing event를 찾기 위해 기존에 있던 OUTRIDER[3]와 비슷한 접근법을 가지는 것을 알 수 있다. 이 접근법은 aberrantly expressed gene들을 찾는 것을 목표로 하며 data 내의 confounder에 대해서 자동적으로 제어를 하는 autoencoder을 사용하는 것이다. FRASER은 splice metrics에 대해 read count ratios에 대한 모델을 생성한다. 이는 RNA-seq read count에서 얻는 splice metric값에 beta binomial 모델을 맞추는 것을 말하며 sample 전반에 있는 co-variation들을 수정한다. 이 행렬을 이용하여 alternative splicing과 가중치(앞서 말한 것과 동치이며 이를 이용해 splicing efficiency metric을 생성하여 사용한다.)들을 사용해서 partial과 full intron retention을 탐지한다.

FRASER은 본문에서 설명된 annotation table과 대응하는 bam file을 예제로 사용해볼 수 있다. 실행 과정은 1) 먼저 sample annotation과 bam 파일의 관련있는 염기들의 수를 이용하여 FraserDataSet을 생성한다. Splice metrics / splicing efficiency metric를 구하고 필요없는 intron을 걸러낸다. 2) FRASER 명령어를 이용하여 전체 pipeline을 실행한다. 3) FRASER results 함수를 이용하여 FraserDataSet에 있는 결과표를 추출한다. 4) 추가로 사용자가 여러 분석 구성을 직접 FraserDataSet object에 맞추는 것으로 원하는 결과를 만들 수 있다.

FRASER의 접근방법을 사용해서 정확한 data의 가공과 결론 도출법을 배울 수 있다.

## 2.3 iREAD: a tool for intron retention detection from RNA-seq data [4]

### 2.3.1 요약

본 논문에서의 동기는 다음과 같다. Intron retention(이하 IR)이 전사체의 정보를 얻기 위해 관련 분야에서 연구 대상으로 떠오르고 있다. 이러한 IR이 주요 생물학적 기능 (유전체 expression 정규화 등)을 수행하며, 복합병과 관련이 있다는 것을 알게 되었다. 하지만 IR을 감지하는 것에 관한 방법들이 제한적이므로 새로운 IR detection 방법이 필요하다. 따라서 본 논문은 iREAD를 제시한다. iREAD (intron REtention Analysis and Detector)는 IR event들을 감지하기 위하여 제시하는 새로운 도구로, 유전체 수준에서의 대용량 고효율로 RNA-seq data를 탐색한다. iREAD는 input으로 해당 gene의 정보를 갖고 있는 BAM file, 전사체와 genome의 intron coordinate(intron에 관한 정보)를 포함하고 있는 text file을 받는다. 그리고 output으로 IR이 감지된 것들을 출력한다.

### 2.3.2 살펴볼 내용

본 논문의 iREAD는 Python으로 구현이 되어있어 살펴보고 응용하기에 편할 것 같다는 장점이 있다. iREAD의 알고리즘은 다음과 같다. 1) 먼저 요약에서 말한 두 개의 input file을 입력 받는다. 이에 대해 더욱 자세히 살펴보면 BAM file의 경우 reference genome에 염기쌍을 대응하여 생성된 것이다.(STAR과 같은 도구를 사용하여 얻는다.) text file은 어떠한 exon이나 isoform, gene들과 overlap되지 않는 coordinate of independent intron의 정보를 모은 file이다. 2) 먼저 independent intronic region을 Samtools를 이용하여 BAM file로부터 추출한다. 이후 Bedops를 이용하여 염기쌍들을 읽어들인다. 3) read-spanning length와 coordinate of the independent introns를 고려하여 reads를 센다. 이는 결과물이 확장될 수 있는 spliced reads를 포함할 수 있기 때문이다. 4) iREAD는 exon-intron junction reads의 수를 세고 기록하는데, 이는 intron들의 retention에 대한 직접적인 증거로 사용된다. 5) retained intron의 염기쌍들은 일반적으로 전체 intron region에서 flatly distributed되기 때문에 ‘flatness’를 특정화 하기 위한 score을 개발했다. 이 score은 다음을 고려한다. 각 intron의 entropy (각 염기쌍의 수를 벡터로 표현한 행들들의 정규화를 통하여 구한 확률질량분포의 entropy로 이후에 모든 entropy를 다시 normalize한다. 이를 NE-score이라고 한다), NE-score을 구하며 얻은 총 reads의 수와 junction reads, 그리고 intron의 길이FPKM (Fragments Per Kilobase of transcript per Million)를 고려하는 것이다. 여기서 각 4개의 값을 이용하여 threshold 값을 정하고 이를 이용하여 IR을 감지한다.

위 논문을 통하여 입력된 gene에 대한 정보들의 어떤 수치를 넘기면 IR이 일어났다고 감지할 수 있는 방법이 있음을 알게 되었다. 이를 이용하여 계산 이론적 모델을 사용한 것과 비교할 수 있을 것이다. 또한 이 논문을 살펴보며 mis-splicing (intron retention)을 모델링 할 수 있을 가능성이 있는지에 대해 다음과 같은 질문에 대해 생각해보았다. 1) right linear인 DFA를 만들어 주어진 string이 accept 되는지 여부를 이용할 수 있는지 2) independent intron coordinate를 분석하여 overlapping (주어진 string의 substring과 길이, 각 alphabet들이 모두 일치하는지) 하는 것으로 확인할 수 있는지. 이를 고려하며 본 프로젝트의 문제 정의와 해결 방안에 대해 생각하기 시작했다.

## 2.4 Schema for Parallel Insertion and Deletion: Revisited [5]

### 2.4.1 요약

본 논문은 parallel insertion과 deletion에 대한 schema를 제시하고 연구한 논문이다. 목적은 p-schema를 기반으로 한 insertion, deletion에 대해, syntactic congruence (구문 합동)에 따라 language equation을 해결할 수 있는 알고리즘을 제안하는 것이다. 또한 이 알고리즘은 주어진 식이 해를 가지는 것 뿐만 아니라 그것의 최대, 혹은 최소 해의 모든 집합을 만드는 것을 결정함을 보인다.

### 2.4.2 살펴볼 내용

DNA computation operation은 turing-universal임이 증명되어 있다. 따라서 DNA computation operation에 p-schema를 적용할 수 있는지를 살펴본다. 먼저 p-schema는 parallel operation schema로 word들의 tuple들의 집합을 말한다. p-schema의 parallel insertion은word에 language가 insert될 두 요소의 사이를 나누는 법(어떤 word를 두 조각으로 나누어 첫 번째 조각이 끝나는 것과 두 번째 조각이 시작하는 것으로 나누고 이 사이에 들어가는 word가 있음을 정의하고 n-tuple에 대해서도 확장)을 정의한다. 이의 inverse operation로 p-schema에서 parallel deletion 또한 정의되며 이 schema에 따른 대수 성질 또한 증명이 되어있다. 이 p-schema는 contextual insertion을 parallelize하는 것에만 의존하지 않는다. 그리고 이러한 정의를 시작으로 의 factor(prefix와 suffix 사이에 들어갈 수 있는 word) 는 를 만족하는 word를 정의하며 관련 theorem들을 설명한다. (이 때 prefix와 suffix 중 하나는 길이가 0이어도 된다. )

아직 language equation과 대수를 잘 모르는 부분이 많아 본 논문의 전체를 이해할 수는 없었지만 전체적인 흐름을 이해를 하려고 노력을 해보았다. 그리고 위 논문을 통하여 내가 어떠한 부분에 대한 이해도가 모자라고 더 공부가 필요한지 확실하게 알 수 있게 되었다.

# 3. 문제 정의

## 3.1 문제 정의 과정

### 3.1.1 설계에 관한 고민

계산이론적 관점에서 문제에 대해 세부적인 정의를 하기 이전에 어떠한 방법들을 사용할 수 있을지 고민을 해보았다. 먼저 intron을 기준으로 설계할 것인지 exon을 기준으로 설계할 것인지를 정해야 할 것이다. intron을 기준으로 할 경우, 2.3에서 사용된 independent intron (모든 exon과 overlap 되지 않는 intron)을 알고 있다고 전제한다. 이 때 input string (gene)에 대해서 어떻게 처리를 할 것인지를 먼저 생각해보자. Input string에서 모든 independent intron을 pattern matching 하여 deletion을 한 경우, Input string에 derivation을 한 결과와 본래 input string을 비교하여 어떻게 intron retention이 일어났는지 판단할 수 있을지 고민해야 한다. exon을 기준으로 할 경우는 어떤가? 인간의 경우에 intron과 exon의 비율이 9대 1정도로 exon의 양이 적기 때문에[1] 고려해야할 요소가 적어질 수 있다는 장점이 있다. 하지만 exon에 대한 data set을 구할 수 있는지, intron을 기준으로 했을 때와 마찬가지로 derivation을 진행한 input string에서 어떻게 intron retention이 일어났는지 판단할 수 있는 명확한 근거를 생각해야 한다.

### 3.1.2 문제 정의와 해결 방법 접근에 관한 고민

### 3.1.2.1 접근법 1

input string 를 splice 하는 production이 있다고 했을 때, 다음을 정의한다. : input alphabets (A, G, T, C, 등이 될 수 있음. reads의 alphabet들), : splicing production rule, = independent intron의 집합 (, , : splice production을 끝낸 의 집합으로 , 이다. 이 때 문제는 “과 주어진 gene들을 비교하여 intron retention을 detect할 수 있는가?”로 정의 한다. 여기에 대해서 간단한 고찰을 해보자. 은 exon들의 집합이므로 주어진 gene과 비교하여 각 , gene들을 하나씩 읽으며 alphabet을 비교하는 DFA를 설계한다. 만약 final state로 가지 못한다면 intron retention이 일어난 것이며 혹은 단순히 와 gene의 길이가 같지 않으면 intron retention이 일어난 것이다. 이에 따른 문제점으로 splice production rule은 어떻게 정의할지, 이미 온전한 splice가 되었다고 생각할 수 있는 를 구할 수 있다면 문제의 의미가 있는지, 집합에 주어진 gene이 include 되어 있지 않다면 (즉 가능한 의 원소들을 모두 구했다고 가정한 것) 그저 비교만 하여 intron retention을 판단하는 것이 계산이론적 관점에서 develop인지 혹은 develop할 수 있는지, 그리고 시간이나 메모리와 같은 cost, resource를 어떻게 줄일 수 있을지 고민해봐야 한다.

### 3.1.2.2 접근법 2

주어진 gene이 splice가 끝난 이후의 gene을 입력으로 받는 경우를 생각해보자. 이 입력은 iREAD의 input data와 같다. 이 때 다음을 정의한다. : input string (gene)의 집합, : input alphabets (A, G, T, C, 등이 될 수 있음. reads의 alphabet들), : splicing production rule, = independent intron의 집합 (, : 원소들의 reverser의 집합 ()이다. 여기에 대해서 간단한 고찰을 해보자. 먼저 의 원소들을 derivation한다고 했을 때 1) (의 원소 하나)의 substring이 , 에 속한다면 intron retention이 일어난 것, 2) 의 정의에 따라 의 원소들은 exon과 overlap 되지 않으므로 이들의 substring (길이가 충분한 k보다 큰)이 의 substring과 같다면 intron retention이 일어난 것이다. 1)의 경우는 intron 전체가 retention 된 경우이며 2)의 경우는 intron의 일부가 retention 된 경우이다. 또한 2)에서 길이가 충분히 k보다 크다는 것의 k가 의미하는 것은intron의 시작 reads, 혹은 끝 reads들의 길이이다. (e.g., k = 2; intron의 시작 혹은 끝 reads string이 exon에 포함되어 있을 수 있음) 여기서 k를 어떻게 설정할 것인지, heuristic 하게 설정할 것인지를 고민해야한다. heuristic하게 설정하는 것은 명확한 근거가 될 수 없을 수도 있으므로 적절한 이유를 찾는 것이 좋을 것 같다. 그리고 의 비교를 하는 DFA를 어떻게 설계할 것인지, 계산을 할 때 (set들의 원소를 각각 비교해야 하므로 의 크기를 a, 의 크기를 각각 b, c, 각 집합의 원소 string의 최대 길이를 al, bl, cl이라고 하자. 이 때 단순 비교의 경우 시간 복잡도는 KMP를 사용할 경우 O(a\*b\*max(al, bl) + a\*c\*max(al, cl))이 될 것인데 이를 어떻게 개선할 수 있을지 또한 고민해야한다. 단, 이 접근법의 경우 모델의 설계를 수정한다면 intron retention 뿐만 아니라 exon skipping에도 적용할 수 있을 것이다.

## 3.2 문제 정의

3.1을 바탕으로 본 프로젝트에서 해결하고자 하는 문제를 ‘splicing이 끝난 이후의 gene을 입력으로 받고 (즉 전사체 등에 의하여 splicing operation이 완료된 이후의 gene) 해당하는 gene들에 intron retention이 일어났는지를 판단하는 계산 이론적 모델 설계’로 정의한다.

이에 따라 다음을 정의한다. : input string (splicing이 끝난 gene)의 집합, : intron들의 prefix로 5’ splice site에 속하는 string이다. (), : intron들의 suffix로 3’ splice site에 속하는 string이다. : 에 속하는 각 원소들의 문자로 이루어진 집합으로 를 원소로 갖는 집합이다. : 에 속하는 각 원소들의 문자로 이루어진 집합으로 로 정의한다. 이에 으로 정의한다. 이를 통해 는 intron이 가지는 pattern (prefix-suffix)의 집합을 정의한 것으로 알 수 있다. 이렇게 했을 때, 문제는 다음과 같이 재정의할 수 있다. ‘와 의 pattern을 pattern matching하여 일정 길이() 이상의 pattern이 에 있다고 판단이 되면 가 intron retention이 일어났다고 말할 수 있는 계산 이론적 모델 설계’를 구하는 것이다. 이 때 왜 intron의 prefix와 suffix만 고려해도 되는가? 이는 intron이 splicing될 때를 생각해보면 된다. 어떤 intron이 prefix, infix, suffix로 이루어져 있다고 하자. 만약 정상적으로 잘려진 intron이라면 w에는 prefix, infix, suffix가 전부 없어야 한다. 이 때 만약 prefix가 w에 없거나 suffix가 w에 없는 상황을 보면 먼저 prefix가 w에 없는 경우, 남아있는 intron은 suffix를 포함하고 있다. 반대로 suffix가 w가 없는 경우, 남아있는 intron은 prefix를 포함하고 있다. 따라서 infix를 고려하는 것보다 prefix와 suffix를 고려하는 것으로 문제를 정의한다면 intron 전체에 대해서 문제를 정의하는 것보다 문제의 범위가 줄어드는 것이다. 이 때 pattern matching algorithm으로 Aho-Corasick 알고리즘을 응용할 예정이다.

위 문제를 해결하는 연구가 어떠한 의미를 가지는지에 대해서 생각해보자. 기존의 선행 연구들은 실험적 데이터를 분석하고 이에 대한 과학적 측정치, 혹은 정량적으로 분석을 했다. 하지만 본 연구는 이러한 기존의 연구들에 대해 계산이론 모델을 적용하여 성능과 정확도를 높여보는 것을 목표로 한다. 만약string에 대한 operation으로 intron retention을 감지할 수 있게 된다면 계산이론으로 확장하여 우리가 알고 있는 계산이론의 지식을 더 응용할 수 있게 된다. 따라서 접근 방식의 관점을 넓혀 다양한 후속 연구를 할 수 있도록 한다.

# 4. 추후 진행 계획

3.2에서 정의한 문제에 대해 해결 방법을 연구한다. 또한 이를 효율적인 알고리즘을 통하여 성능을 개선시킬 수 있는지 연구를 목적으로 프로젝트를 진행한다. 기존에 있는 여러 알고리즘과 모델을 비교하면서 적용을 할 수 있는지, 할 수 있다면 어떻게 성능이 개선되는 지를 비교한다. 이를 위해서 모델을 python으로 간단하게 구현을 해본다음 기존에 존재 하는 intron retention detecting tool (FRASER, iREAD 등)과 같은 입력을 대상으로 어느 정도의 정확도를 가지는지, 놓치는 부분이 무엇인지에 대해 연구를 진행한다. 어느 정도 모델이 완성되었다면 어떻게 시각적으로 본 알고리즘을 보여줄 수 있는지 조사를 한다. 이를 통하여 일반인들에게 본 알고리즘에 대한 이해에 도움을 주는 것을 목표로 한다.

# 참고자료

[1] Hyunchul Jung, Kang Seon Lee, Jung Kyoon Choi, 2020, Comprehensive characterisation of intronic mis-splicing mutations in human cancers, <https://doi.org/10.1038/s41388-020-01614-3>

[2] Christian Mertes, Ines Scheller, Julien Gagneur, 2022, FRASER (Find RAre Splicing Events in RNA-seq)

[3] Brechtmann F, Mertes C, Matusevičiūtė A, Yépez VA, Avsec Ž, Herzog M, Bader DM, Prokisch H, Gagneur J, 2018, “OUTRIDER: A Statistical Method for Detecting Aberrantly Expressed Genes in RNA Sequencing Data.” *The American Journal of Human Genetics*, **103**(6), 907 - 917. ISSN 0002-9297, doi: [10.1016/j.ajhg.2018.10.025](https://doi.org/10.1016/j.ajhg.2018.10.025).

[4] Li HD, Funk CC, Price ND. iREAD: a tool for intron retention detection from RNA-seq data. BMC Genomics. 2020 Feb 6;21(1):128. doi: 10.1186/s12864-020-6541-0. PMID: 32028886; PMCID: PMC7006120.

[5] LILA KARI, SHINNOSUKE SEKI, 2011, SCHEMA FOR PARALLEL INSERTION AND DELETION: REVISITED, International Journal of Foundations of Computer Science Vol. 22, No. 7, DOI: 10.1142/S0129054111008945