

생물정보학자의 알고리즘, 기계학습 그리고 인공지능

이도훈
서울대학교 생물정보연구소

소개

- **Computational biologist / Bioinformatician**
- **Bio**
 - **2013~2017** B.S., School of Biological Sciences, SNU (CS minor)
 - **2017~2021** Ph.D., Interdisciplinary Program in Bioinformatics, SNU (Prof. Sun Kim)
 - **2021~** Postdoctoral fellow, Bioinformatics Institute, SNU

소개

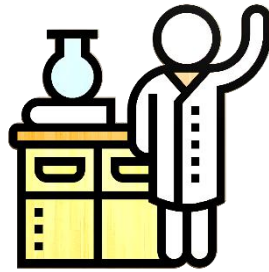
- **Computational biologist / Bioinformatician**
- **Bio**
 - **2013~2017** B.S., School of Biological Sciences, SNU (CS minor)
 - **2017~2021** Ph.D., Interdisciplinary Program in Bioinformatics, SNU (Prof. Sun Kim)
 - **2021~** Postdoctoral fellow, Bioinformatics Institute, SNU
- **Research Interest**
 - **Computational Biology - Computational Epigenomics**
 - Biology of DNA methylation, and its clinical implications
 - Computational approaches for (epigenetic) intratumor heterogeneity
 - **AI in Bioinformatics**
 - AI-based modeling of epigenomes

목차

1. 생물정보학자의 알고리즘
 - 왜 필요한가?
 - 어떻게 배우는가?
 - 나의 실력은?
2. 생물정보학자의 기계학습
3. 생물정보학자의 인공지능

생물정보학자의 알고리즘

생물정보학자에게 알고리즘이…왜 필요한가?



생명과학자

생물정보학자에게 알고리즘이...왜 필요한가?



생명과학자



전자현미경

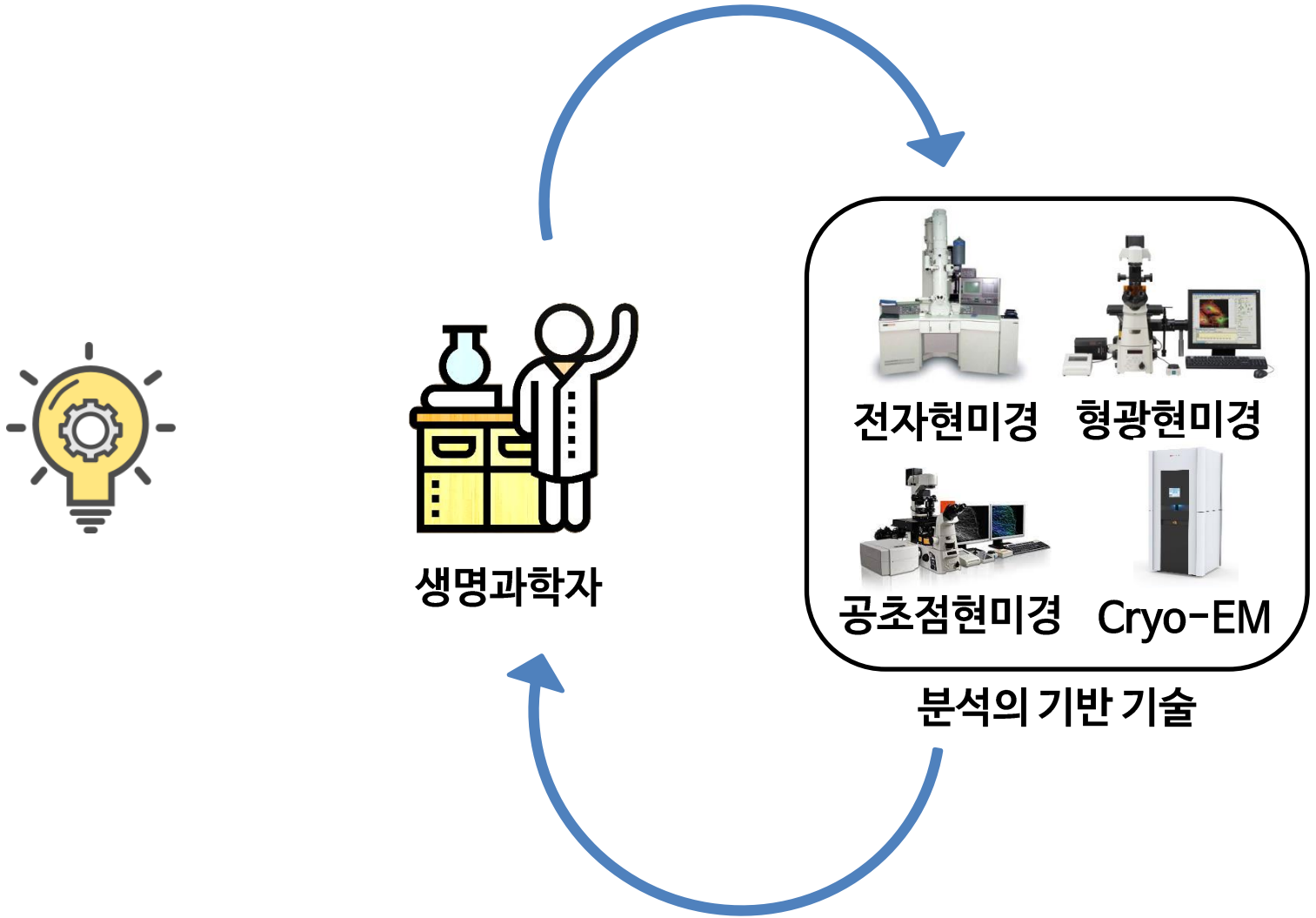
형광현미경

공초점현미경

Cryo-EM

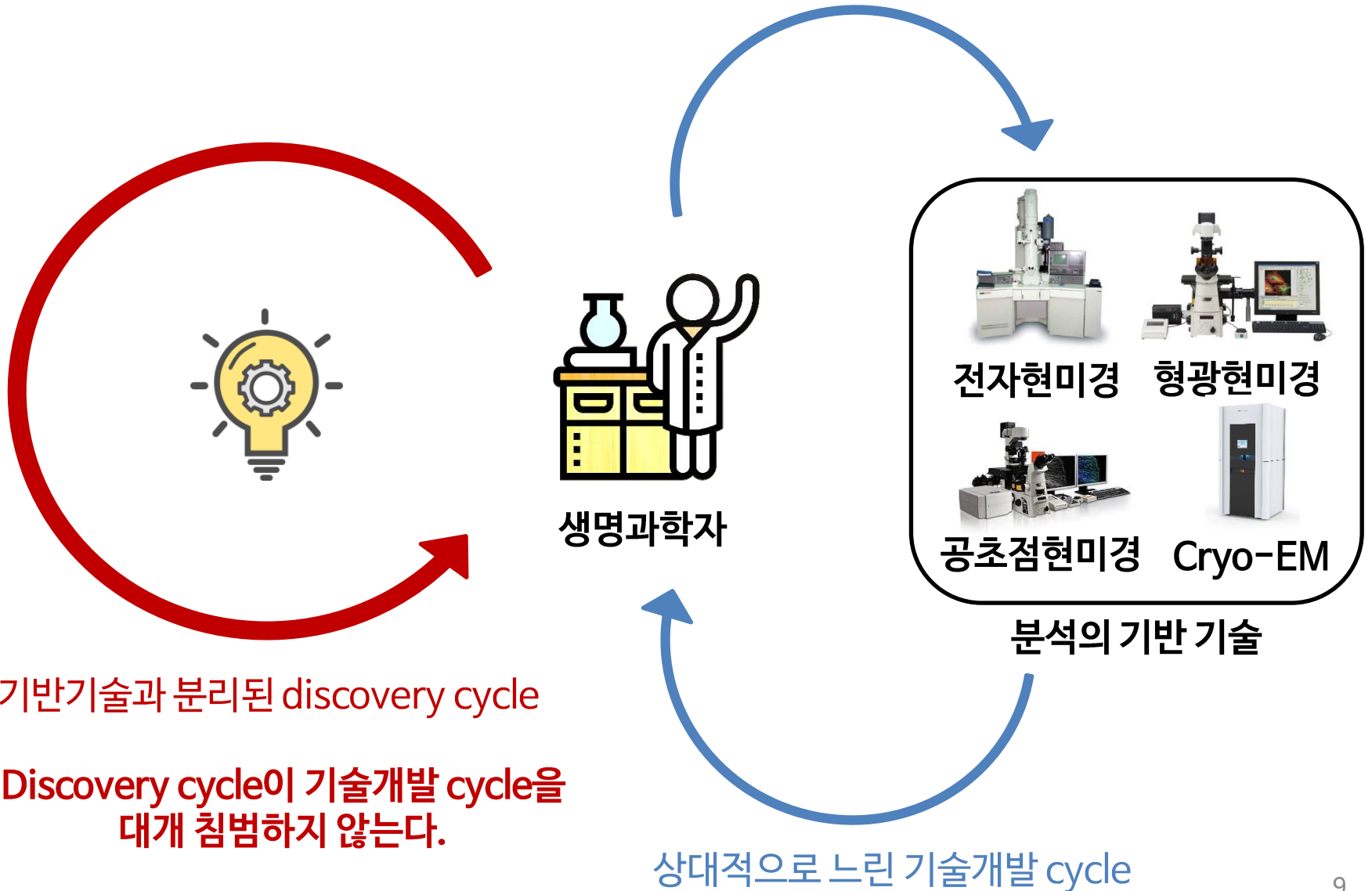
분석의 기반 기술

생물정보학자에게 알고리즘이...왜 필요한가?

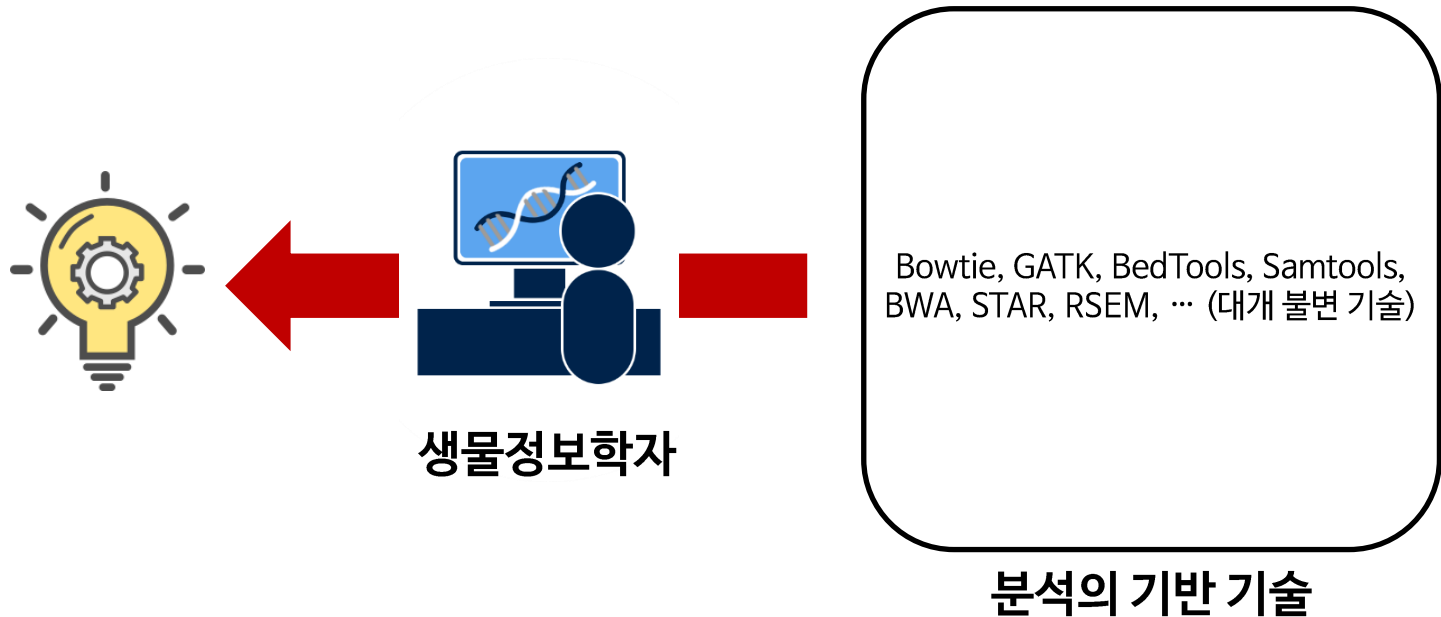


상대적으로 느린 기술개발 cycle

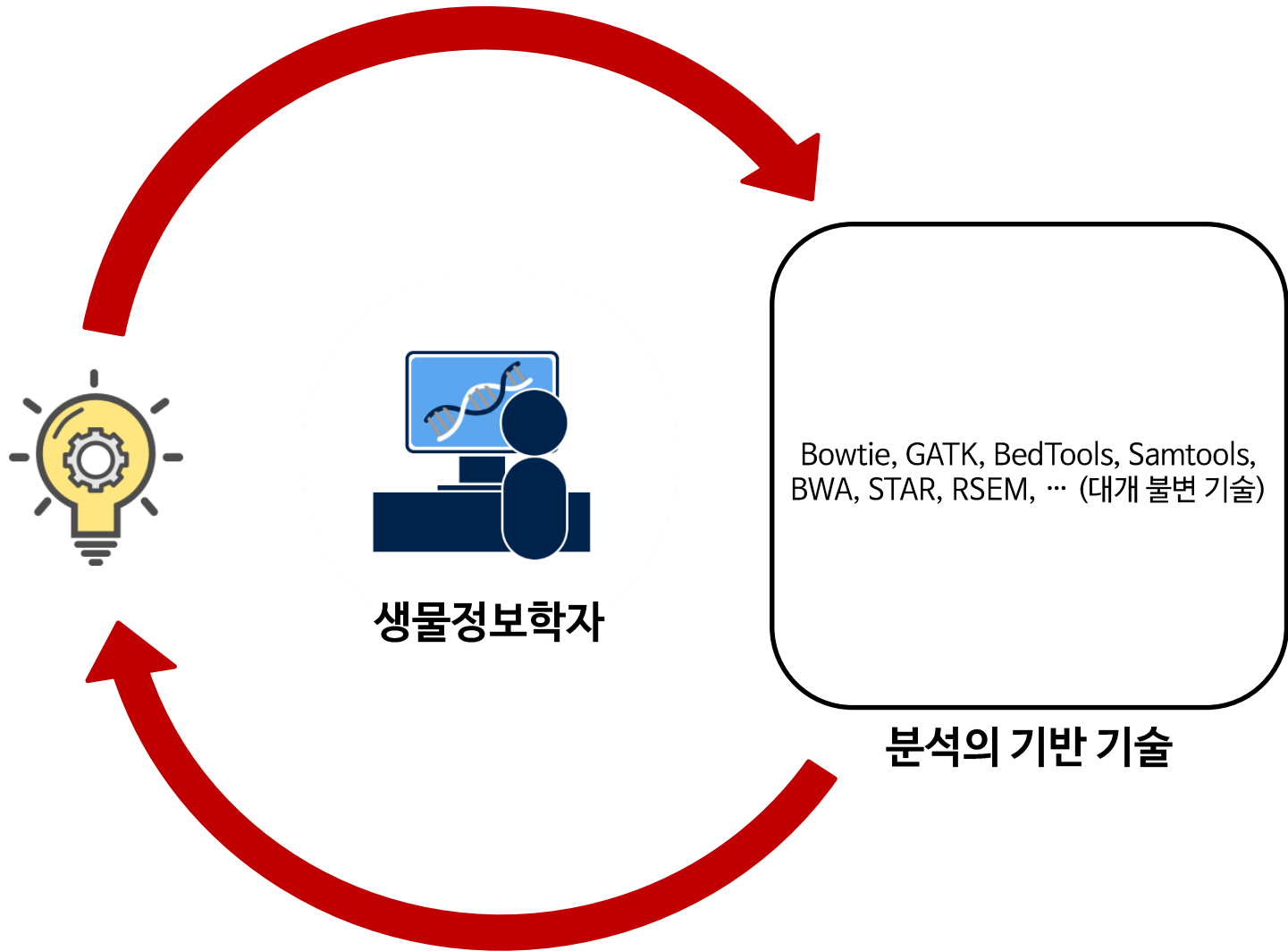
생물정보학자에게 알고리즘이...왜 필요한가?



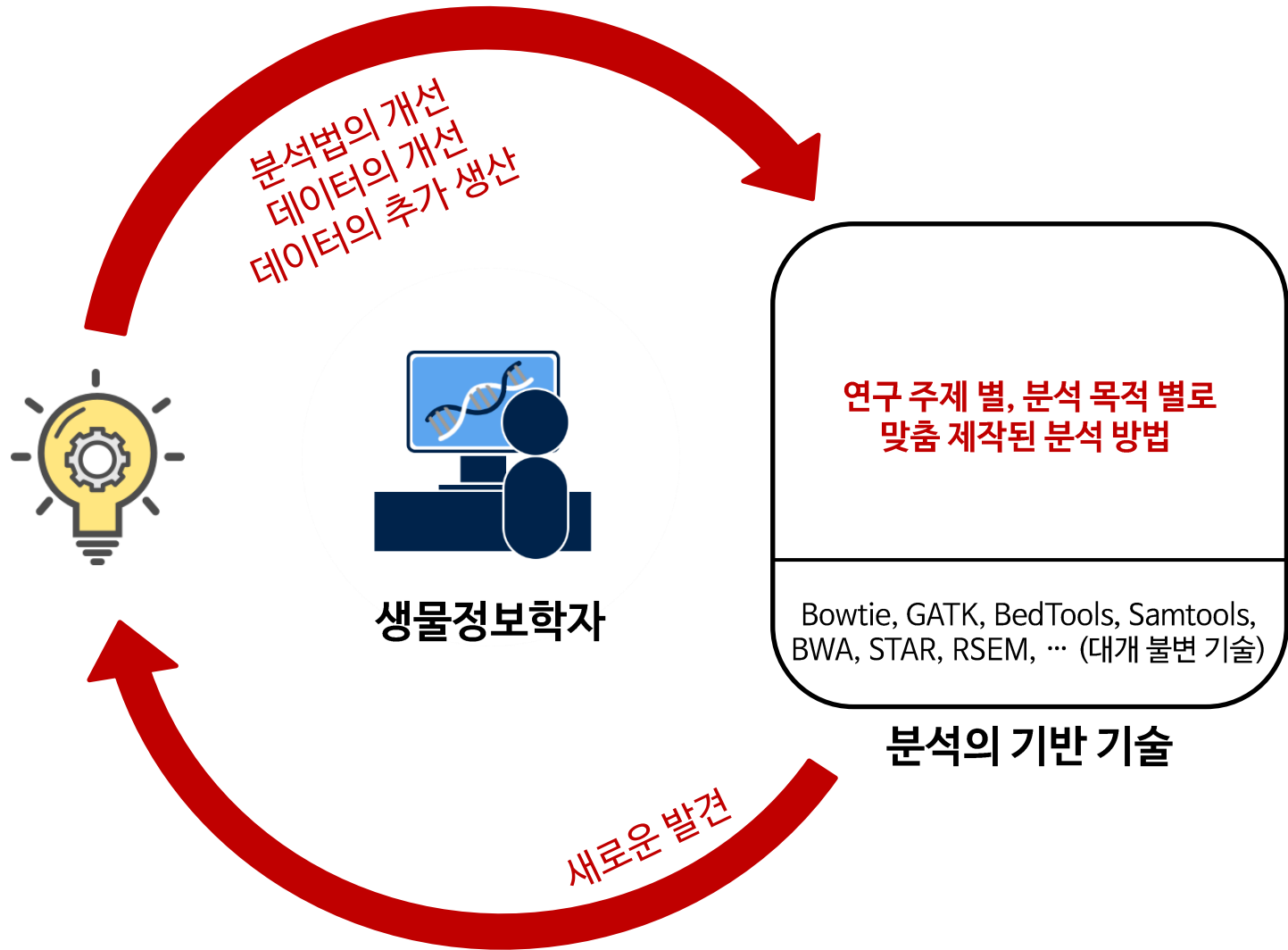
생물정보학자에게 알고리즘이...왜 필요한가?



생물정보학자에게 알고리즘이...왜 필요한가?



생물정보학자에게 알고리즘이...왜 필요한가?



Discovery와 기술개발 cycle이 많은 경우 함께 진행된다.

생물정보학자에게 알고리즘이…왜 필요한가?

〈제 1 목표〉
빠른 discovery cycle

생물정보학자에게 알고리즘이...왜 필요한가?



분석 결과

〈제 1 목표〉
빠른 discovery cycle

생물정보학자에게 알고리즘이...왜 필요한가?



분석 결과



논의

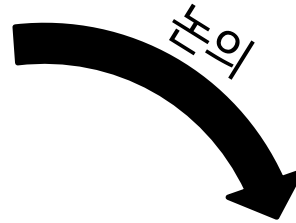
〈제 1 목표〉
빠른 discovery cycle

"이런이런 cell type에서 이런이런 genomic feature를 추가해서 분석해보는 건 어떨까요? 오! 여기 저희 연구와 관련된 매우 큰 public data가 있네요. 한번 추가해서 써 보죠!"

생물정보학자에게 알고리즘이...왜 필요한가?



분석 결과



논의

〈제 1 목표〉
빠른 discovery cycle

"이런이런 cell type에서 이런이런 genomic feature를 추가해서 분석해보는 건 어떨까요? 오! 여기 저희 연구와 관련된 매우 큰 public data가 있네요. 한번 추가해서 써 보죠!"

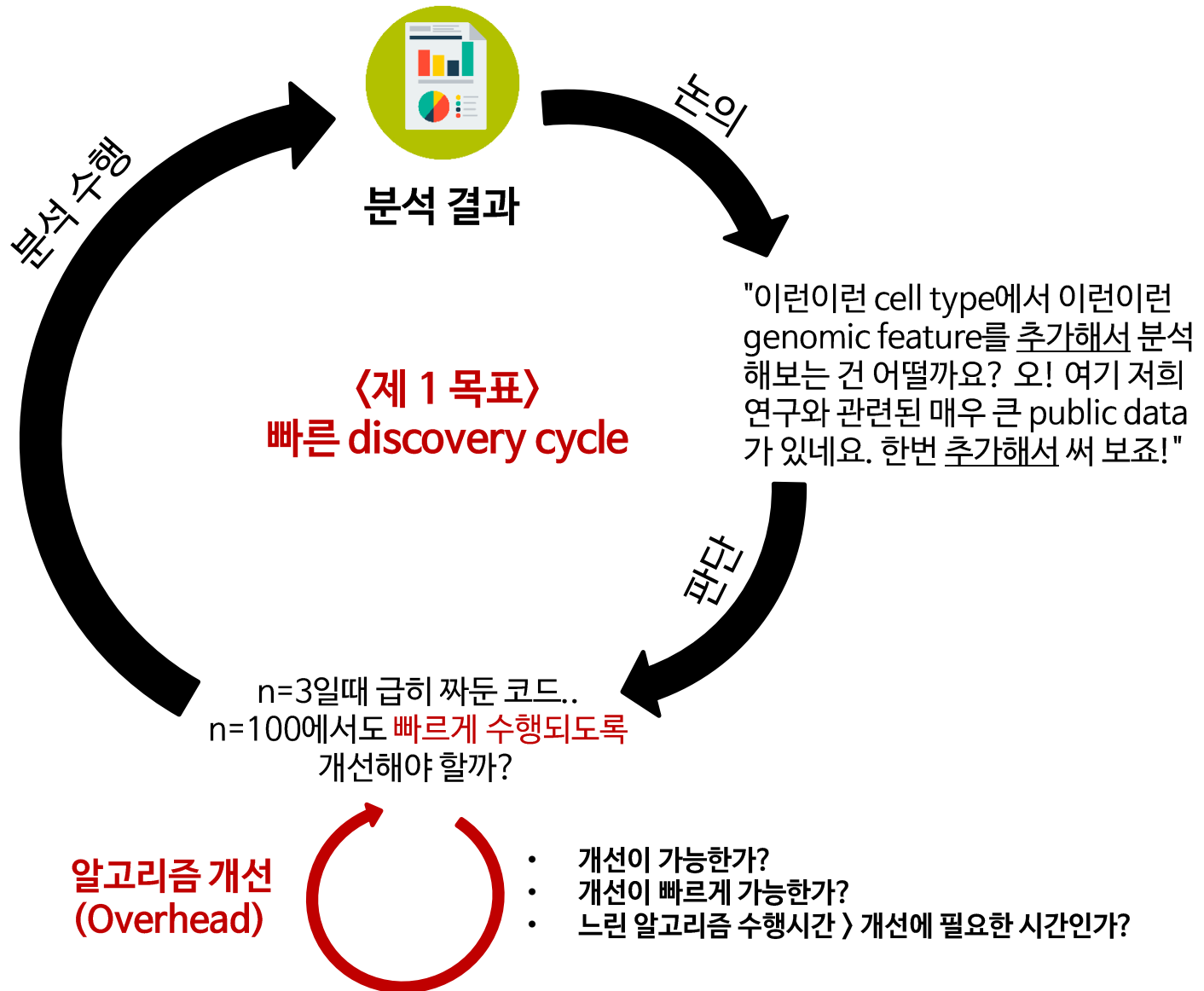


판단

n=3일때 급히 짜둔 코드..
n=100에서도 빠르게 수행되도록
개선해야 할까?

- 개선이 가능한가?
- 개선이 빠르게 가능한가?
- 느린 알고리즘 수행시간 > 개선에 필요한 시간인가?

생물정보학자에게 알고리즘이...왜 필요한가?



생물정보학자에게 알고리즘이...왜 필요한가?

- 알고리즘을 조금만 이해하면..
 - 코드를 급히 짜되, 수행 시간에 대한 감을 잡을 수 있고 개선점 파악이 가능하다.
 - "이건 구현은 간단한데 $O(n^2)$ 이구나. 나중에 시간이 좀 걸리더라도 $O(n \log n)$ 정도로 개선할 수 있을 거야."
 - 굳이 개선하지 않아도 되는 부분을 판단할 수 있다.
 - 급히 짠 코드의 수행 시간 <<< (코드 수정 시간) + (수정된 코드의 수행 시간)이라면... 안 고치는게 낫다!

생물정보학자에게 알고리즘이...왜 필요한가?

- 알고리즘을 조금만 이해하면..
 - 코드를 급히 짜되, 수행 시간에 대한 감을 잡을 수 있고 개선점 파악이 가능하다.
 - "이건 구현은 간단한데 $O(n^2)$ 이구나. 나중에 시간이 좀 걸리더라도 $O(n \log n)$ 정도로 개선할 수 있을 거야."
 - 굳이 개선하지 않아도 되는 부분을 판단할 수 있다.
 - 급히 짠 코드의 수행 시간 <<< (코드 수정 시간) + (수정된 코드의 수행 시간)이라면... 안 고치는게 낫다!
- 구현된 알고리즘을 잘 가져다 쓸 줄만 알아도..
 - 알고리즘 개선에 필요한 overhead가 급격히 줄어든다.
 - 매번 알고리즘을 바닥부터 구현할 필요는 없다. 내 것보다 더 최적화가 잘 된 라이브러리를 쓰자.
 - 다만 코드를 봤을 때 알고리즘을 이해할 수 있는 수준까지는 공부 필요하다.
 - 코드를 급히 짜더라도 처음부터 scalable하고 빠른 코드를 짤 수 있다.
 - 이상적인 결과

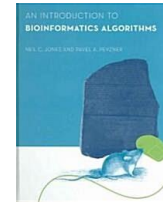
생물정보학 알고리즘...어떻게 배우나?

- 기본 알고리즘
 - Baekjoon Online Judge (<https://www.acmicpc.net/>)

- Rosalind (<https://rosalind.info/>)

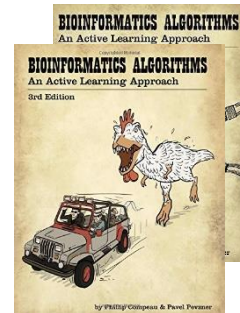
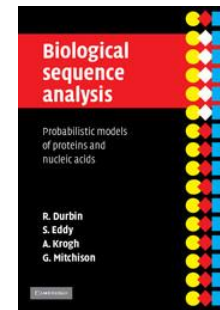
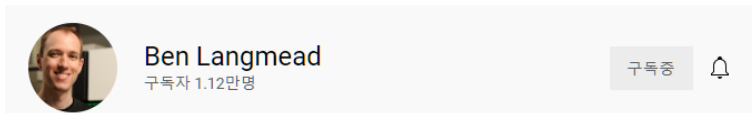
- 교과서들

- An introduction to bioinformatics algorithms, Jones and Pevzner
- **Bioinformatics algorithms - An active learning approach**, Compeau and Pevzner
- Biological sequence analysis, Durbin et al



- 유튜브

- Ben Langmead 채널 (Bowtie!) - 문자열 알고리즘



생물정보학 알고리즘…나의 실력은?

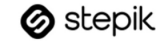
- 실력을 객관적으로 판단하기 위해서 경진대회에 참여하는 건 좋은 방법
- 알고리즘 경진대회는 많다.
- 생물정보학 알고리즘 경진대회는?

Bioinformatics contest

- 2017 / 2018 / 2019 / 2021 총 4회 개최



Bioinformatics contest



- 2017 / 2018 / 2019 / 2021 총 4회 개최

- 4회 모두 참가

-	2017
-	2018 (21 st / ~3000)
-	2019 (21 st / 3329)
-	2021 (34 th / ~4000)

- 2020년 이후는 2년마다 개최
- 온라인으로 진행
- 예선 (1주간 진행) 통과 시 본선 (24시간 동안 진행) 참여

Bioinformatics contest



- 2017 / 2018 / 2019 / 2021 총 4회 개최

- 4회 모두 참가

-	2017
-	2018 (21 st / ~3000)
-	2019 (21 st / 3329)
-	2021 (34 th / ~4000)

- 2020년 이후는 2년마다 개최

- 온라인으로 진행

- 예선 (1주간 진행) 통과 시 본선 (24시간 동안 진행) 참여

Prizes

1st prize:

Whole Genome Sequencing*

2nd and 3rd prizes:

Whole Exome Sequencing*

4th and 5th prizes:

23andMe or Genotek DNA Service*

6th-30th prizes:

Honorable Prize: Bioinformatics T-Shirt

구미가 당기는 상품들

A surprise

Bioinformatics Contest 2019: prizes > 받은편지함 x



Ekaterina Vyahhi <k@bioinf.me>
apap7에게 ▾

2019년 3월 4일 (월) 오후 6:19 ★ < 답장 ⋮

🌐 영어 ▾ > 한국어 ▾ [메일 번역](#)

[영어 번역 안함](#) x

Hello Dohoon Lee!

Congratulations on your outstanding performance in the Bioinformatics Contest 2019!

Your took the place #21 but your result was were close to the top 20. Therefore, we would like to reward you with an honorary prize – our branded bioinformatics T-shirt.

To get your prize please fill in [this form](#). It will take some time to manufacture and send it as we are creating a limited edition exclusively for the winners. We expect that the T-shirts will be ready by the end of March and will reach all winners during April.

Thank you for participating. We wish you success in all your future endeavors!

Best regards,
Ekaterina Vyahhi and the Contest Team

A surprise

Bioinformatics Contest 2019: prizes > 받은편지함 x



Ekaterina Vyahhi <k@bioinf.me>
apap7에게 ▾

2019년 3월 4일 (월) 오후 6:19 ★ < 답장 ⋮

🌐 영어 ▾ > 한국어 ▾ [메일 번역](#)

[영어 번역 안함](#) x

Hello Dohoon Lee!

Congratulations on your outstanding performance in the Bioinformatics Contest 2019!

Your took the place #21 but your result was were close to the top 20. Therefore, we would like to reward you with an honorary prize – our branded

an honorary prize – our branded bioinformatics T-shirt.

expect that the T-shirts will be ready by the end of March and will reach all winners during April.

Thank you for participating. We wish you success in all your future endeavors!

Best regards,
Ekaterina Vyahhi and the Contest Team

Honorary prize



어떤 문제들이 나오나?

- Exact solution이 있는 문제만 나오는 것은 아니다 (좋은 approximate solution을 찾는 문제)
- 심지어 Full score를 아예 받을 수 없는 문제들도 있다.
- 시간 제한이 없다. = 느린 알고리즘이라도 24시간 안에 끝나고, 정확하기만 하면 점수를 받을 수 있다.
- 엄밀히 말해 알고리즘 경진대회라고 할 수는 없고, "생물정보학 경진대회"에 가깝다.
- 분야
 - 알고리즘
 - 생명과학
 - 기계학습 / 최적화 / 인공지능 ...
- 언어
 - 제한 없음
 - 주로 Python을 사용

알고리즘 예제: Metabolite Annotation (2021 Qual, Prob2)

- Mass spectrometry를 이용하여 metabolite identification을 하려 한다.
- MS 실험에서는 ionization에 의한 adduct의 gain/loss가 일어나며, 최종 측정되는 signal은
$$\text{(signal)} = \text{(metabolite의 m/z)} + \text{(adduct m/z)} + \text{(noise)}$$
가 된다.
$$\underbrace{\hspace{1cm}}_s \quad \underbrace{\hspace{1cm}}_m \quad \underbrace{\hspace{1cm}}_a \quad \underbrace{\hspace{1cm}}_{\Delta}$$
- Metabolite와 adduct의 m/z 값 후보 list가 주어졌을 때, 각 signal이 어떤 metabolite와 어떤 adduct의 조합으로 만들어졌는지 알아내어라. (Δ 를 최소화하는 조합. 단, 항상 $m + a > 0$ 이다)

알고리즘 예제: Metabolite Annotation (2021 Qual, Prob2)

Metabolites = [1.000002, 0.000001]

Adducts = [0.500000, -0.500000]

Signals = [0.500001, 0.500002, 0.500003, 1.000000, 0.000001]

} 여기서 1개씩 골라서 더했을 때

↑
이 값과 가장 가까워지는 조합을 찾아라!

|Metabolites|, |Adducts|, |Signals| ~ 10⁴

알고리즘 예제: Metabolite Annotation (2021 Qual, Prob2)

- Binary search 기반으로 해결 + 약간의 병렬화
- 기본적인 수행 시간 고려와 알고리즘 선택은 중요하다

알고리즘 예제: Transposable Elements (2019 Qual, Prob3)

- Linear genome에서, d 개보다 작은 error를 가지고 적어도 n 번 나타나는 문자열 e 를 찾기
- 반복서열을 찾는 문제, but error에 대한 고려를 어떻게 하는지가 중요

알고리즘 예제: Transposable Elements (2019 Qual, Prob3)

- Linear genome에서, d 개보다 작은 error를 가지고 적어도 n 번 나타나는 문자열 e 를 찾기
- 반복서열을 찾는 문제, but error에 대한 고려를 어떻게 하는지가 중요

2 7 3
GAGTCATCGGACGATCC



찾는 문자열 |

ACGTAGC

찾는 문자열의 occurrence |

2 1M1I2M1D1M1X1M
11 3M1I1M1X1M

알고리즘 예제: Transposable Elements (2019 Qual, Prob3)

1. k-mer frequency로 후보군 선정 (충분히 큰 k에 대해서 시도)
 - 2-1. 후보 k-mer가 n번 이상 나타날 경우
 - k-mer 주변 문자열들을 탐색
 - 2-2. 후보 k-mer가 n번 미만 나타날 경우
 - 해당 후보 k-mer로 우선 문자열을 유추하고
 - 문자열과 linear genome 사이의 local alignment를 다시 수행하여 나머지 문자열을 찾아냄
- 모든 해결을 자동화할 필요는 없다.
- 충분히 후보군을 좁힌 상태라면, manual한 검증을 두려워하지 말 것!

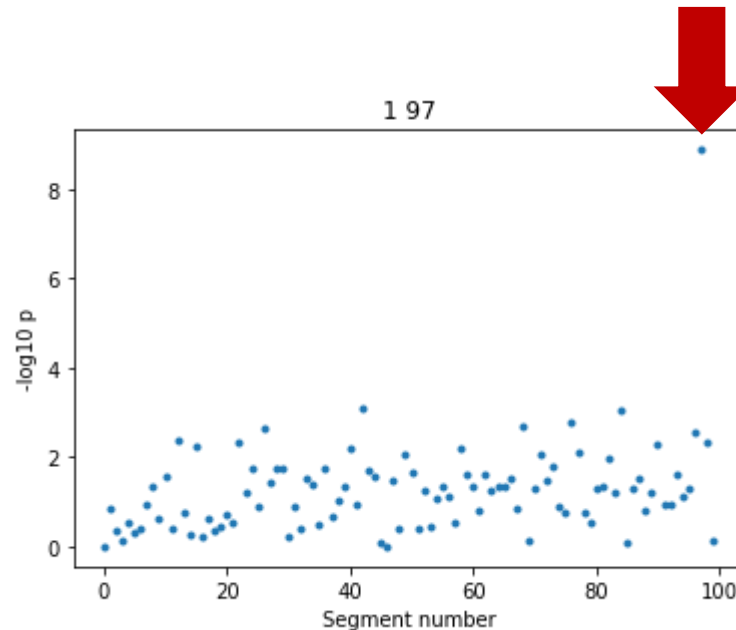
수동 검증 예제: Causative Mutation (2021 Final, Prob2)

- GWAS 를 수행하고, causative mutation을 발굴하라.
- Association이 있는 genomic segment 를 정답으로 제시하되,
- 범위가 narrow 할수록 좋은 점수를 받는다.

수동 검증 예제: Causative Mutation (2021 Final, Prob2)

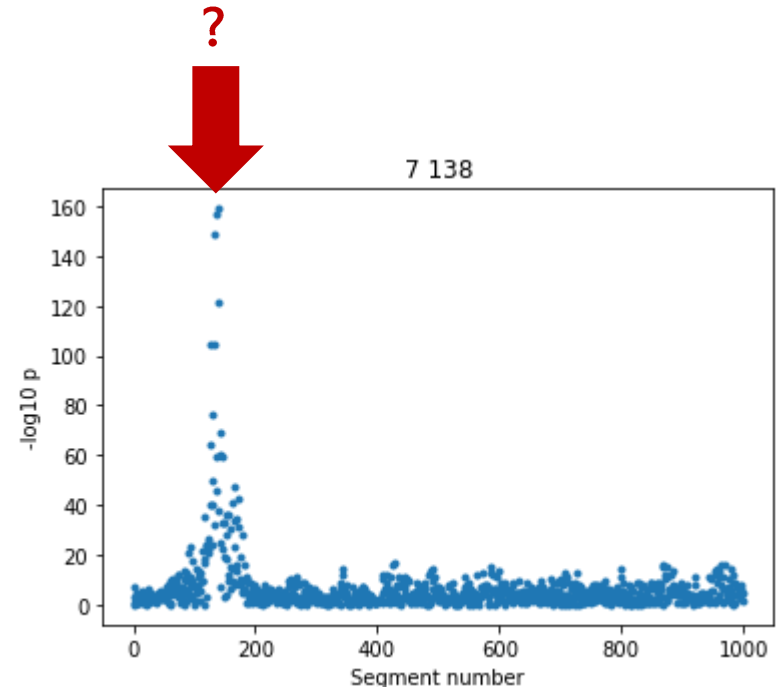
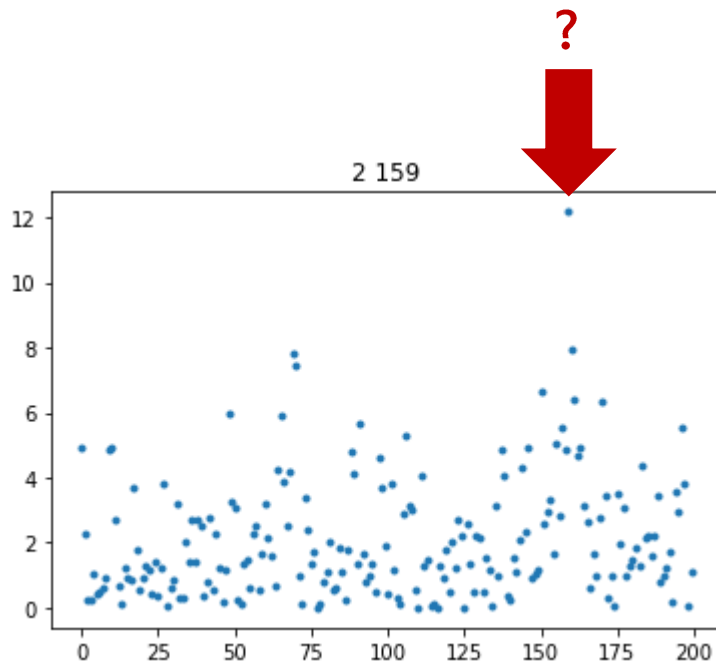
- GWAS 를 수행하고, causative mutation을 발굴하라.
- Association이 있는 genomic segment 를 정답으로 제시하되,
- 범위가 narrow 할수록 좋은 점수를 받는다.

1번 test. 정답이 명확하다



수동 검증 예제: Causative Mutation (2021 Final, Prob2)

- Recombination까지 시뮬레이션했기 때문에,
- 뒤로 갈수록 어느 한 segment 로 특정하기 애매한 경우가 많다.



수동 검증 예제: Causative Mutation (2021 Final, Prob2)

- 힌트: 제출 횟수에 제한이 있고, 각 제출 시 얻는 점수로 맞았는지/틀렸는지 알 수 있다

Causative Mutation (haploid version)

There are limits on each test case! In each test with one test case, you can make ten submissions, and in each test with ten test cases, you can make 100 submissions.

수동 검증 예제: Causative Mutation (2021 Final, Prob2)

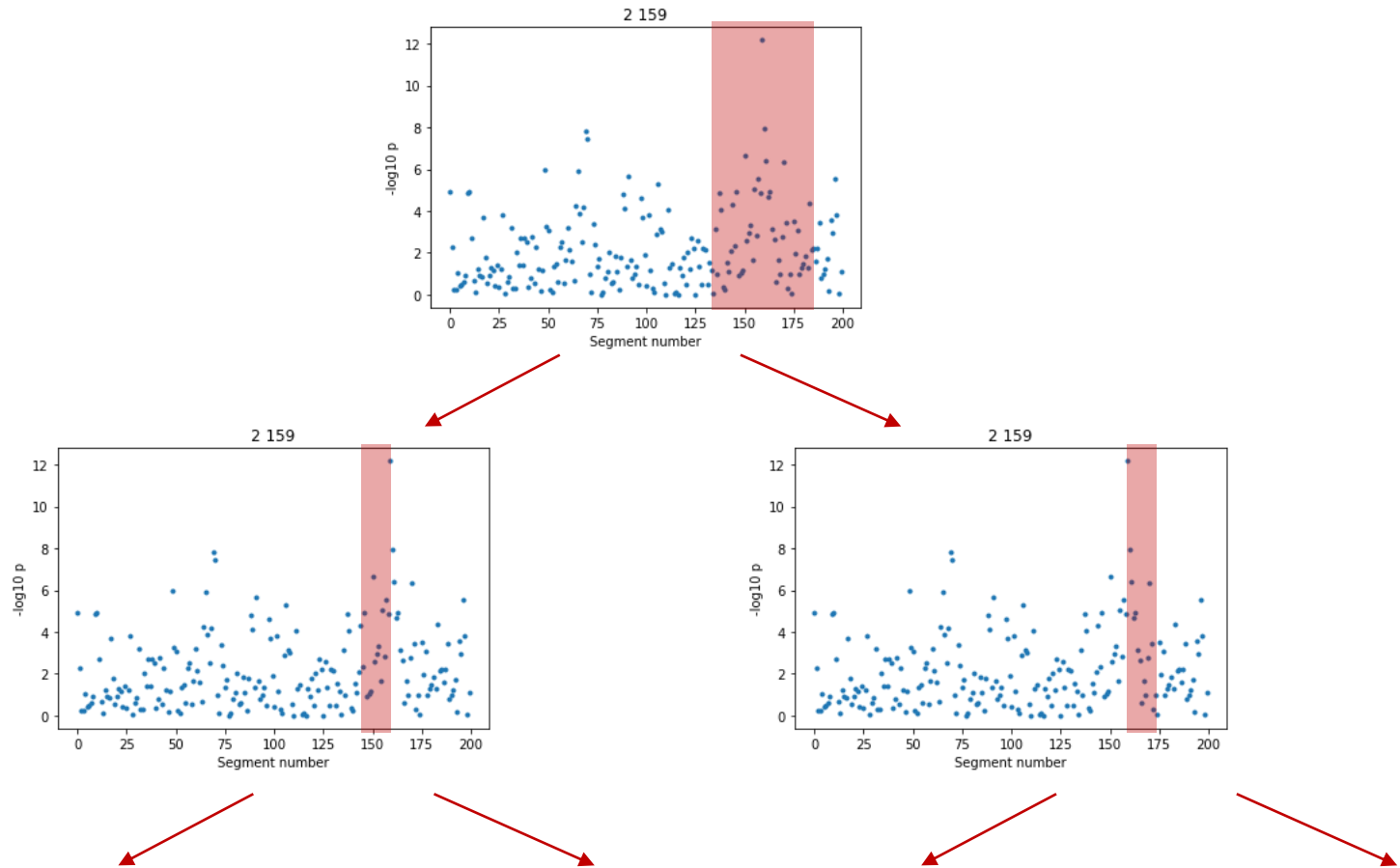
- 힌트: 제출 횟수에 제한이 있고, 각 제출 시 얻는 점수로 맞았는지/틀렸는지 알 수 있다
= 결국 제한 횟수 내에 최적의 방법으로 후보군을 narrow down 하라는 뜻

Causative Mutation (haploid version)

There are limits on each test case! In each test with one test case, you can make ten submissions, and in each test with ten test cases, you can make 100 submissions.

수동 검증 예제: Causative Mutation (2021 Final, Prob2)

- 아이디어: Binary search로 정답 segment를 찾자



생명과학 예제: Recombination of Plasmids (2018 Final, Prob1)

- pDEST plasmid와 pENTR plasmid 서열이 주어졌을 때, site-specific recombination 결과로 나타나는 최종 plasmid 서열을 구하라.

생명과학 예제: Recombination of Plasmids (2018 Final, Prob1)

- pDEST plasmid와 pENTR plasmid 서열이 주어졌을 때, site-specific recombination 결과로 나타나는 최종 plasmid 서열을 구하라.

>pDEST

ACGCACAGTCAAAGAACATCTGTCCTGAGGCCCTAAAGCT...



>pENTR1

TGTGATGCCTTCCCTTCTTGGTTTCTTAAGCATCACACTTT...



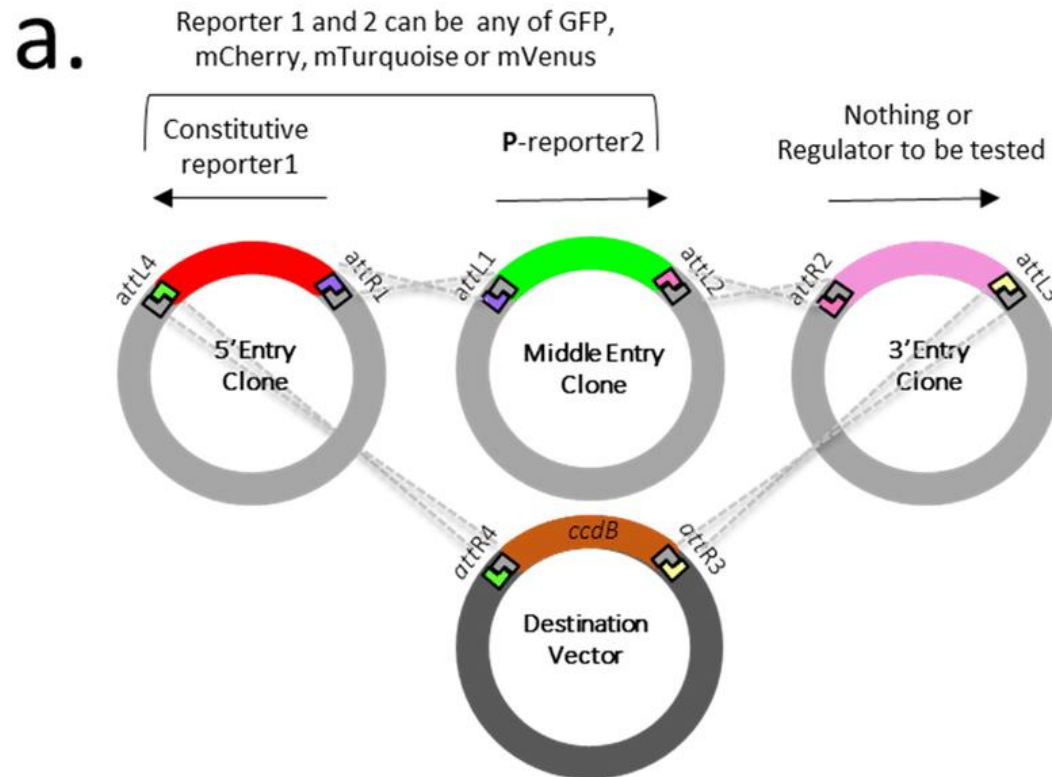
ACGCACAGTCAAAGAACATCTGTCCTGAGGCCCTAAAGCT...

생명과학 예제: Recombination of Plasmids (2018 Final, Prob1)

- Gateway cloning 을 미리 알고 있었다면 쉽게 풀리는 문제

생명과학 예제: Recombination of Plasmids (2018 Final, Prob1)

- Gateway cloning 을 미리 알고 있었다면 쉽게 풀리는 문제
- 각 plasmid에 어떤 att 서열이 있는지 찾고, 짝이 맞는 att 서열끼리 recombination

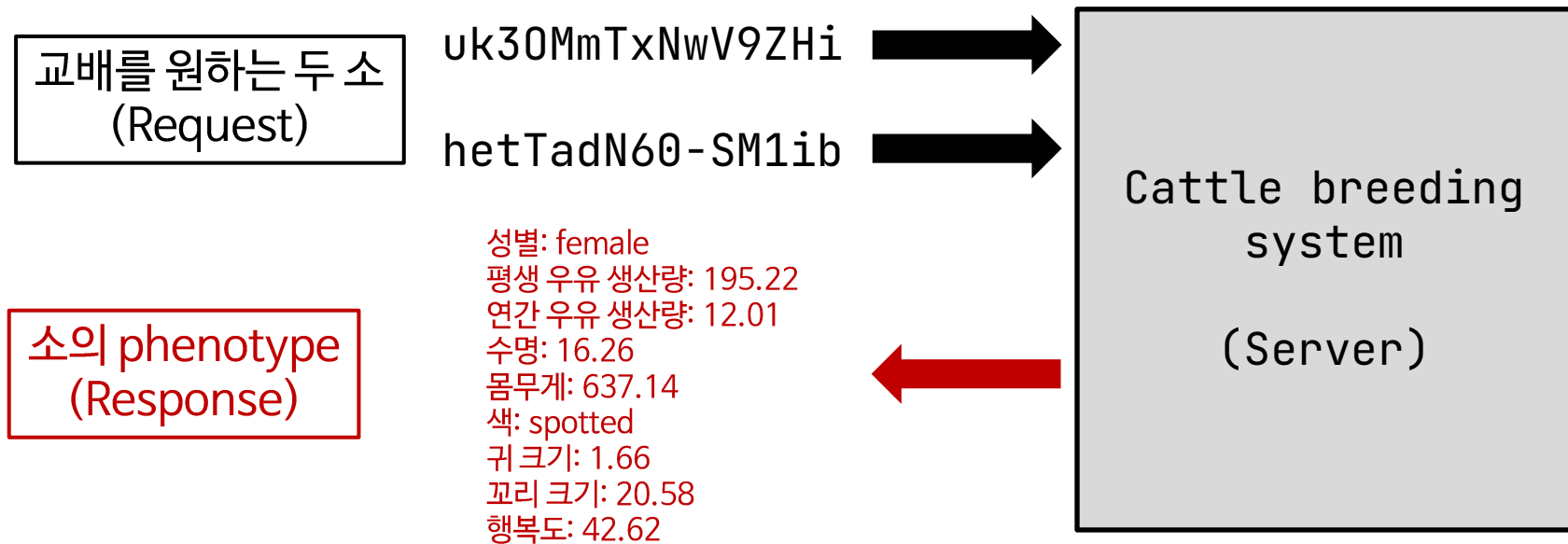


생명과학 예제: Recombination of Plasmids (2018 Final, Prob1)

- 문헌 조사나, 배경 지식이 중요할 수도 있다.

참신한 예제: Cattle Breeding (2018 Final, Prob5)

- 참신한 interactive 문제 → 육종 시뮬레이션
- 초기 50마리 소들의 population 이 주어짐 (각 소는 id로 구분)
- 교배를 원하는 두 소의 id를 전송하면 자손의 phenotype을 응답해주는 server와 통신을 반복하여, 평생 우유 생산량이 큰 소를 만들어내는 것이 목적



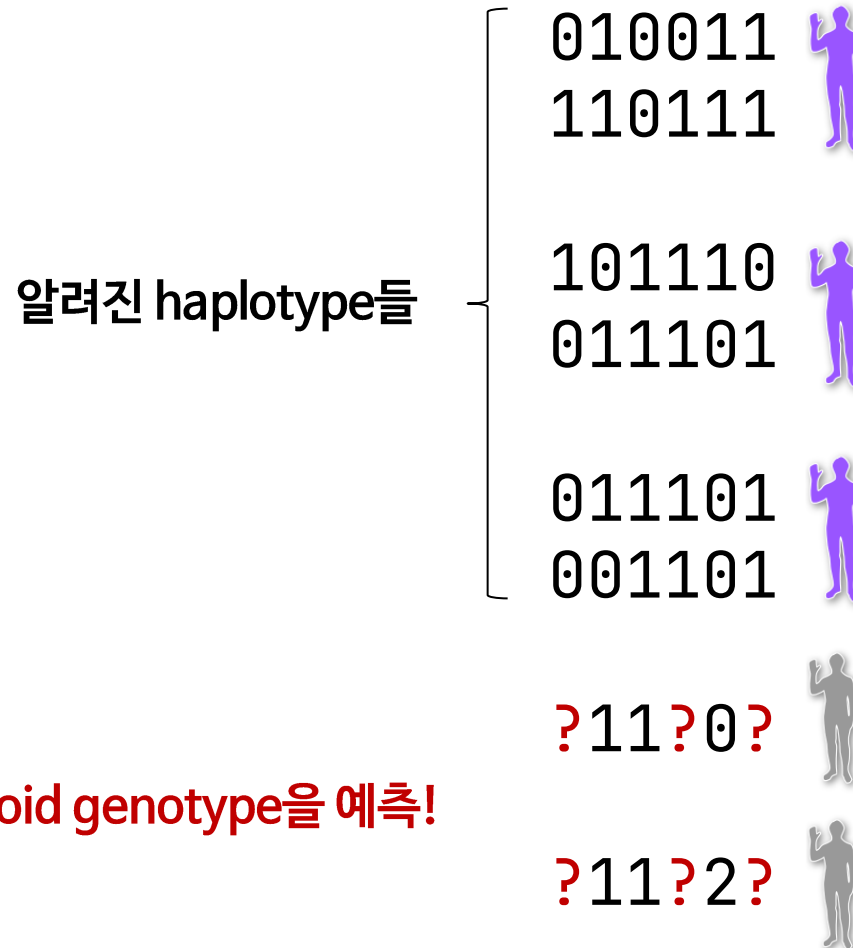
생물정보학자의 기계학습

기계학습...언제 쓰는가?

1. $X \rightarrow Y$ 관계가 있다는 것을 안다. (Feature - Target 관계)
2. X와 Y의 관계를 통계적으로 모델링하고 싶은데, 복잡한 모델을 구현하기에는 노력 대비 아웃풋이 적다.
3. 간단한 기계학습 모델을 간단히 시도해본다.
 - 3-1. 잘 되면, 사용한다.
 - 해석 가능한 모델이라면 모델 해석으로부터 insight를 얻는다.
 - 3-2. 성능이 조금 부족한 것 같으면, 그 모델을 baseline으로 두고 문제에 조금 더 맞춤형의 통계 모델을 구축하여, 성능이 오르는지 본다.

예제: Genotype imputation (2021 Final, Prob1)

- 여러 사람의 haplotype-resolved genotype 정보가 주어졌을 때, **missing genotype**을 imputation하는 문제



Missing diploid genotype을 예측!

Solution?

010011
110111



101110
011101



011101
001101




?11?0?





?11?2?





Solution?

010011
110111 

101110
011101 

011101
001101 


?11?0? 


?11?2? 


IMPUTE2, MACH – HMM-based


BEAGLE – Graphical model-based


Solution?

010011
110111 

101110
011101 

011101
001101 

?11?0? 

?11?2? 


IMPUTE2, MACH – HMM-based


BEAGLE – Graphical model-based





24시간 이내 구현은 무리라고 판단!


접근 방법 - 단순화하자!

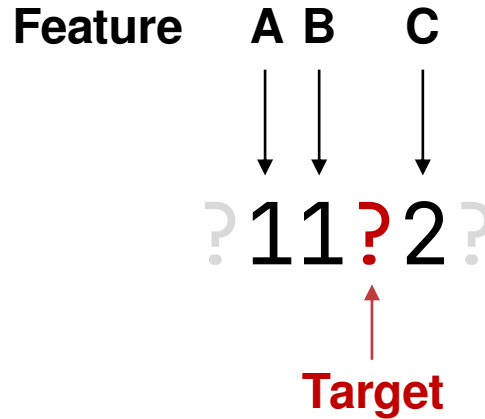
010011 
110111

101110 
011101


011101 
001101


?11?0? 


?11?2? 





접근 방법 - 단순화하자!

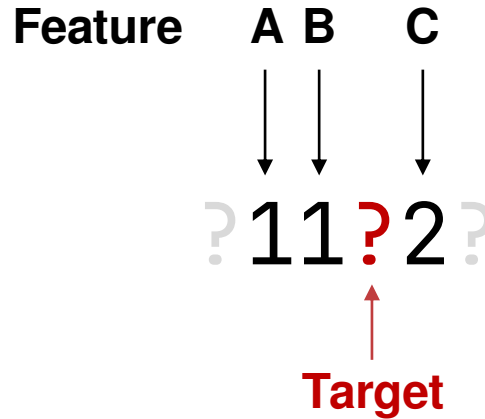
010011 
110111

101110 
011101

011101 
001101

?11?0? 

?11?2? 

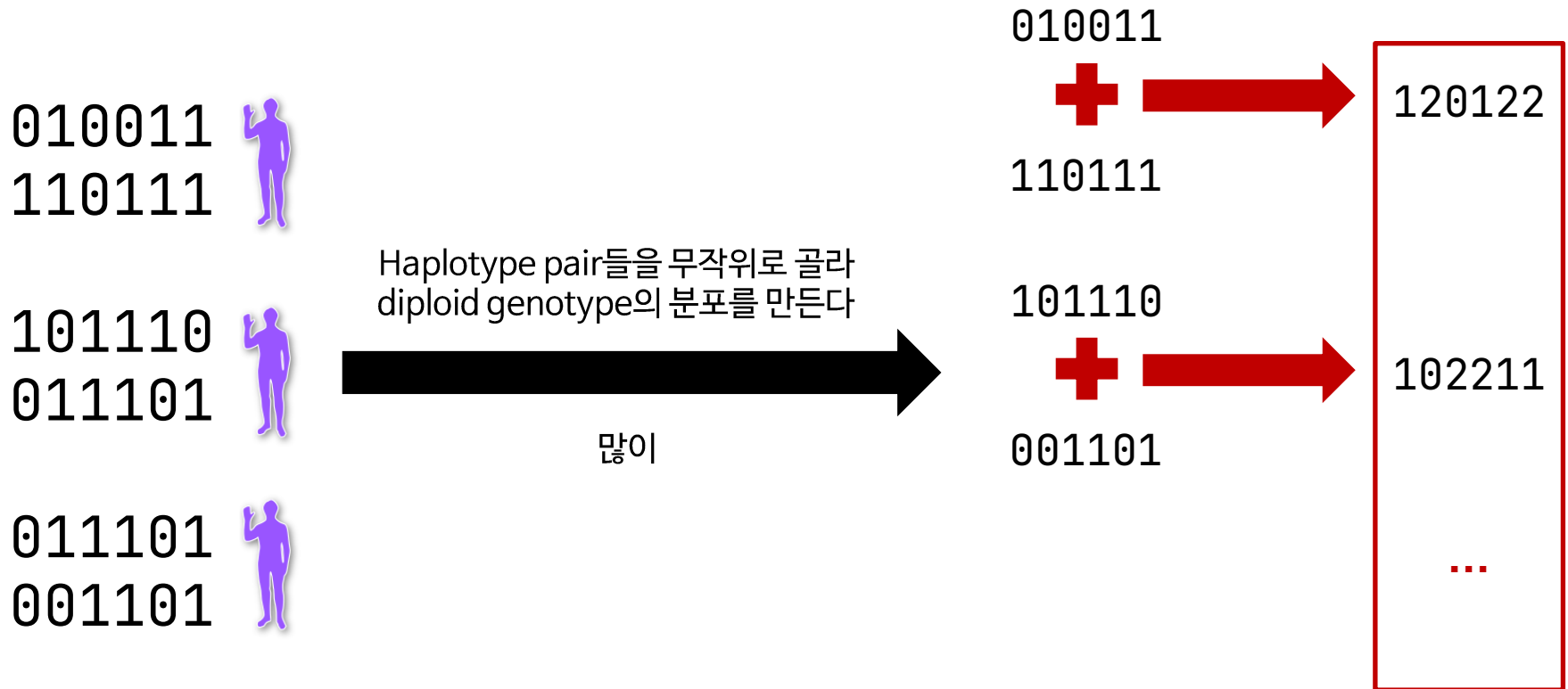


A	B	C
1	1	2

예측
→
하는 ML 모델을 학습시키면..?

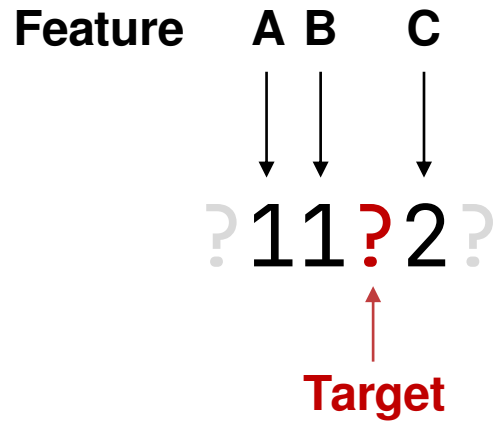
?

접근 방법 - Training set의 구성



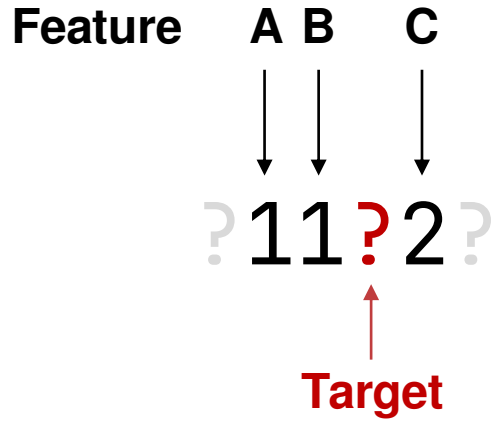
접근 방법 - Training set의 구성

목표

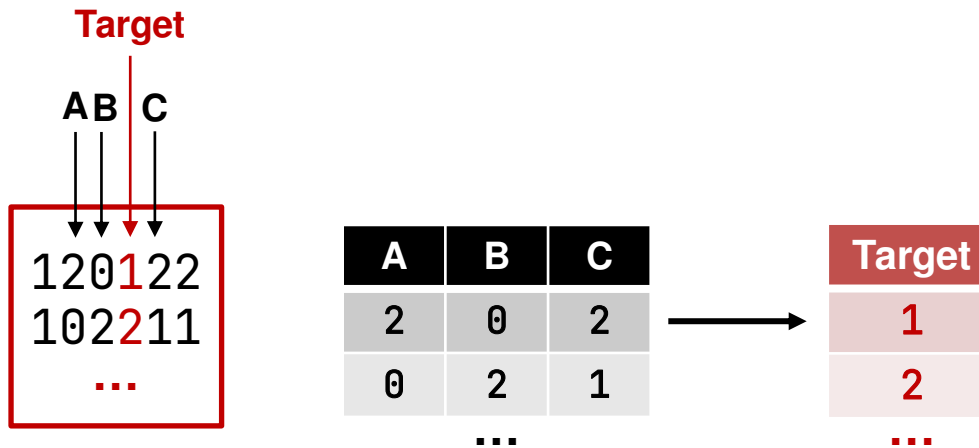


접근 방법 - Training set의 구성

목표



학습 데이터



접근 방법 - 학습 및 예측

신경
망

A	B	C
2	0	2
0	2	1
...		

Random Forest

모델 학습

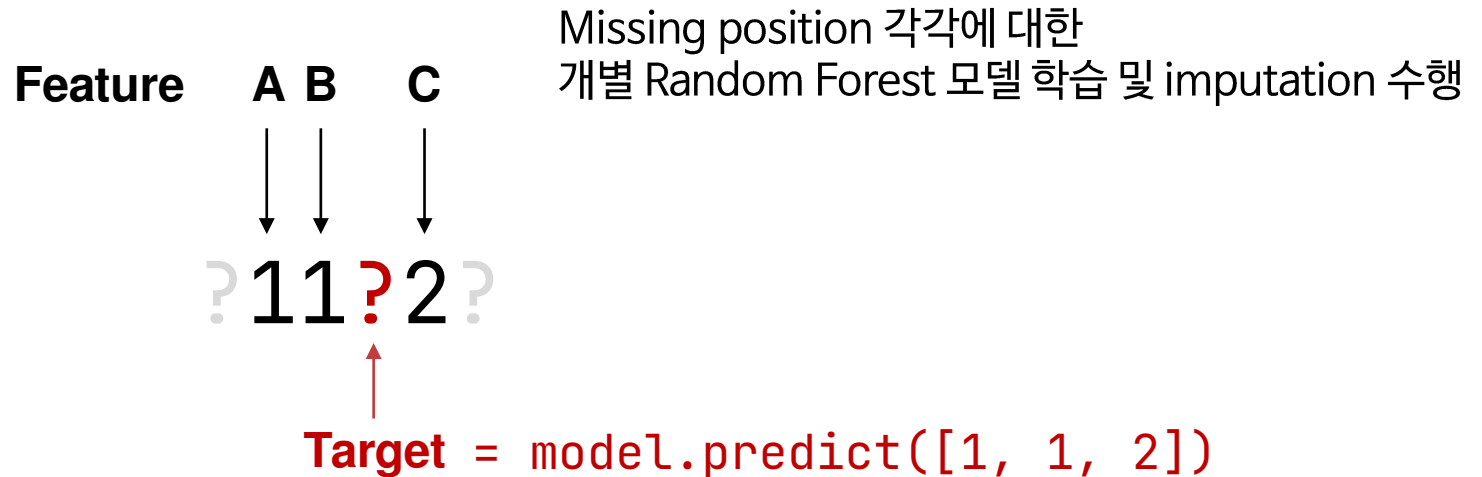
Target
1
2
...

접근 방법 - 학습 및 예측

학습

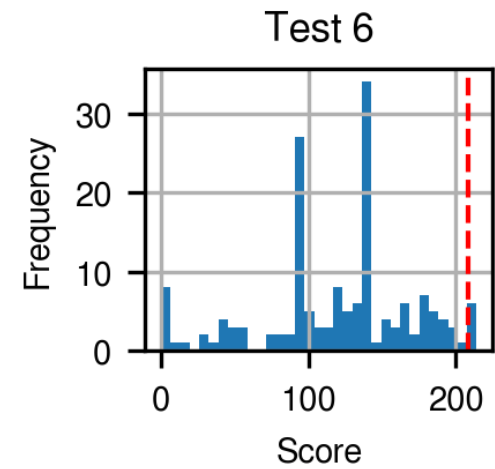
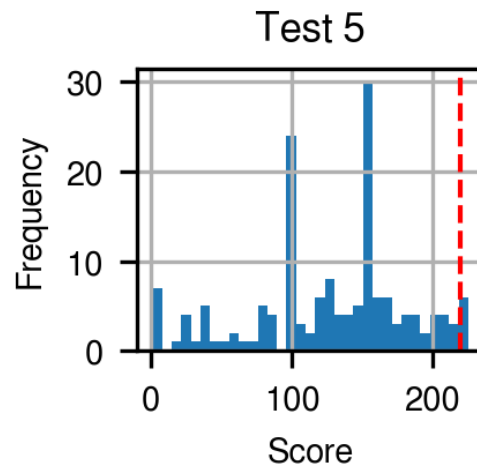
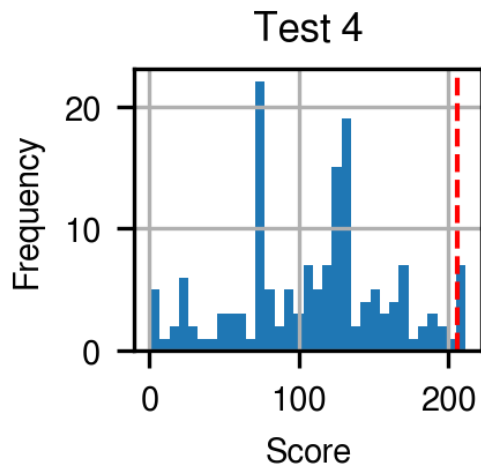
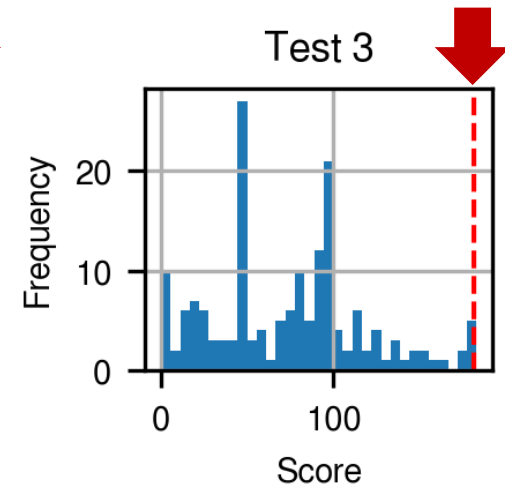
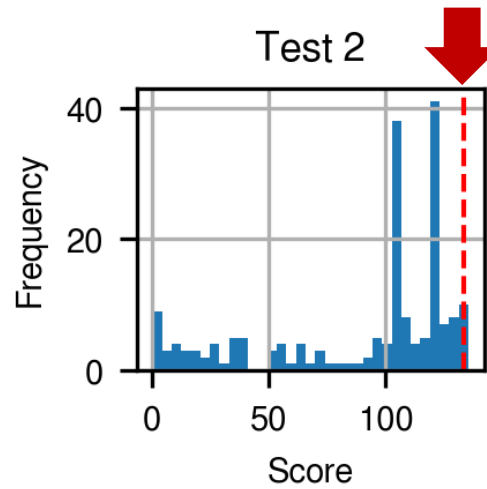
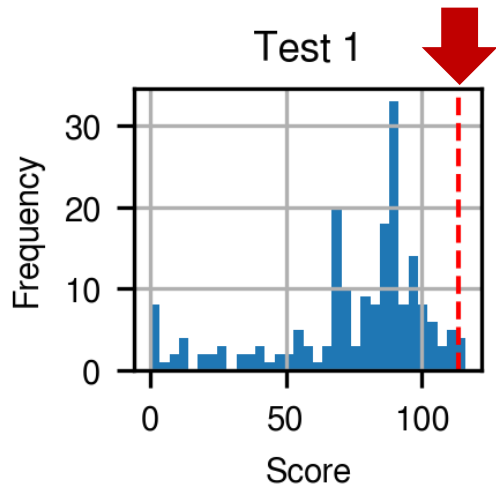


예측



결과

My score



예제: Cluster the reads (2018 Final, Prob4)

- 여러 bacteria로부터 나온 16S rRNA sequence 서열들을 잘 clustering 할 수 있는가?

```
> read_1  
ACGCTGCTT  
> read_2  
ACGGGACTACCTT  
> read_3  
ACGTCGGTT  
> read_4  
ACGAATATCTT
```

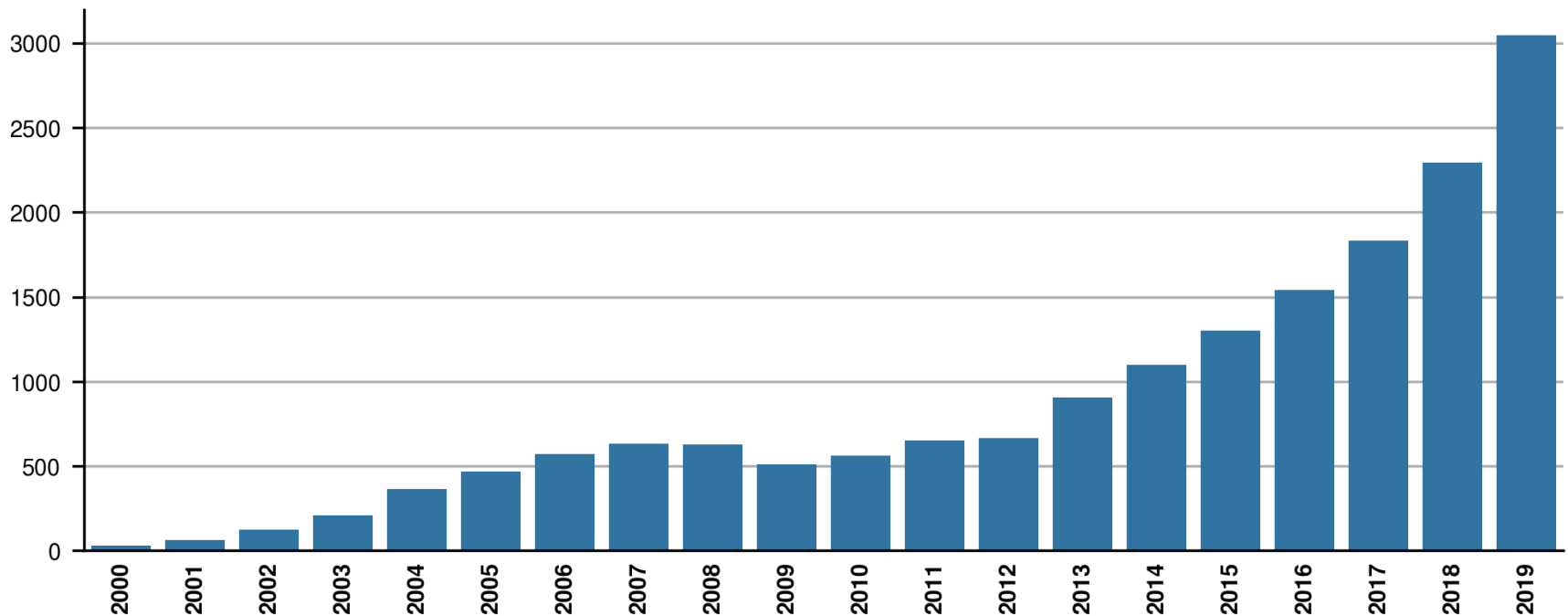


```
read_1 1  
read_2 2  
read_3 1  
read_4 third
```

생물정보학자의 인공지능

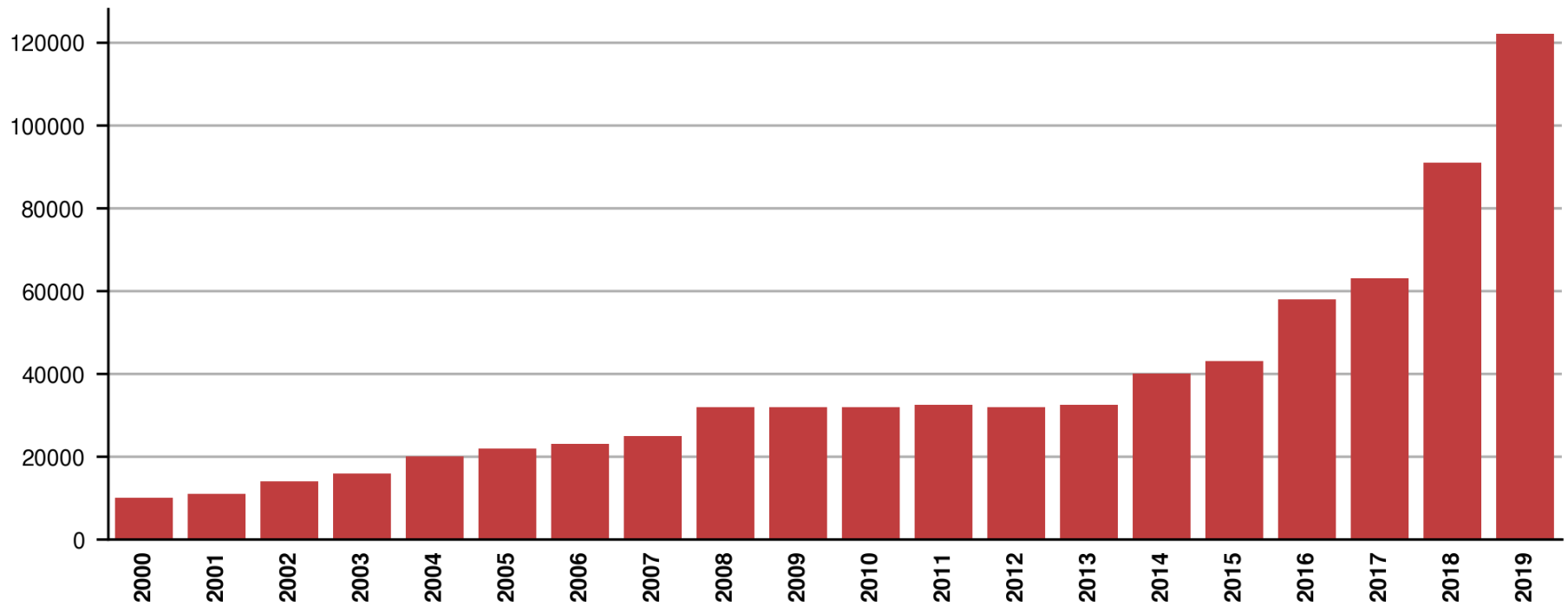
We bioinformaticians are in the AI era

Number of AI+Bioinformatics articles (2000-2019, PubMed)



We bioinformaticians are in the AI era

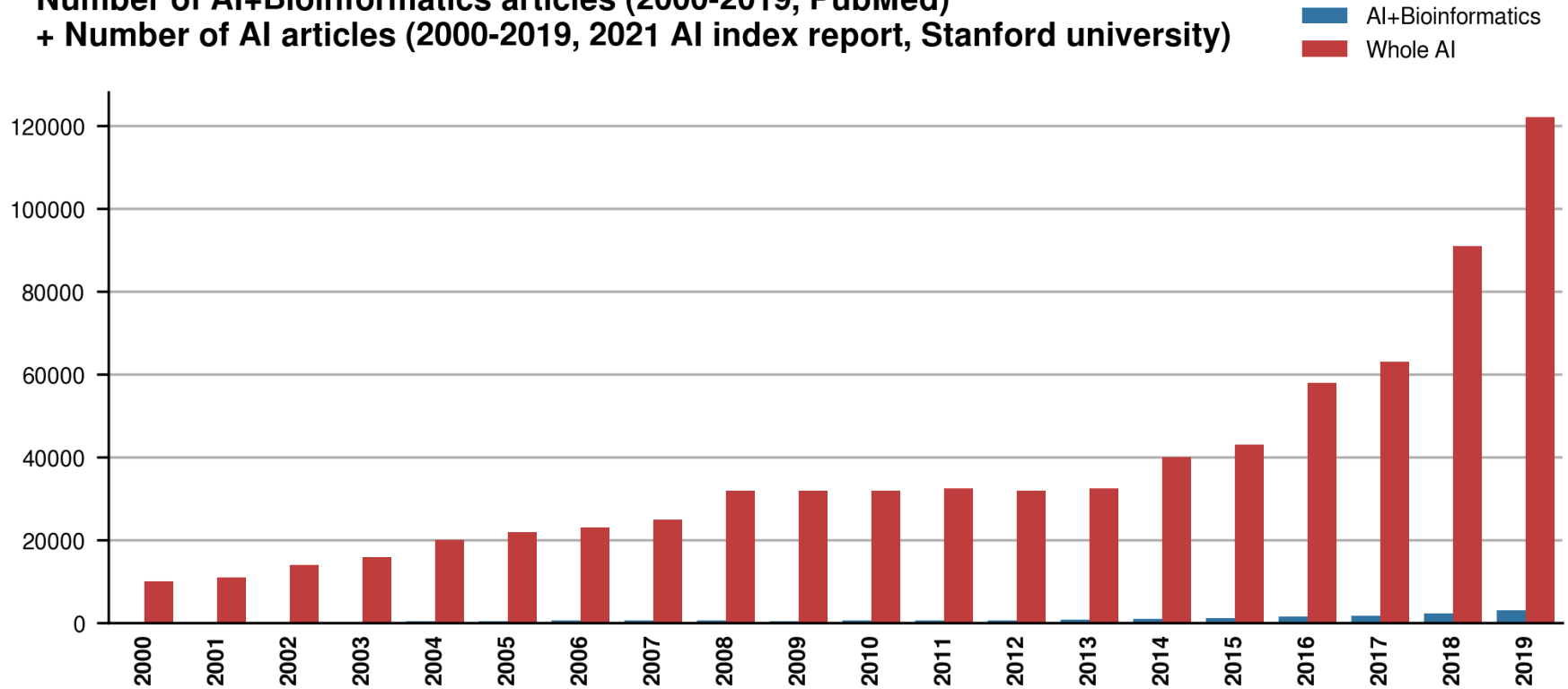
Number of AI articles (2000-2019, 2021 AI index report, Stanford university)



2021 AI index report, Stanford University, 재가공

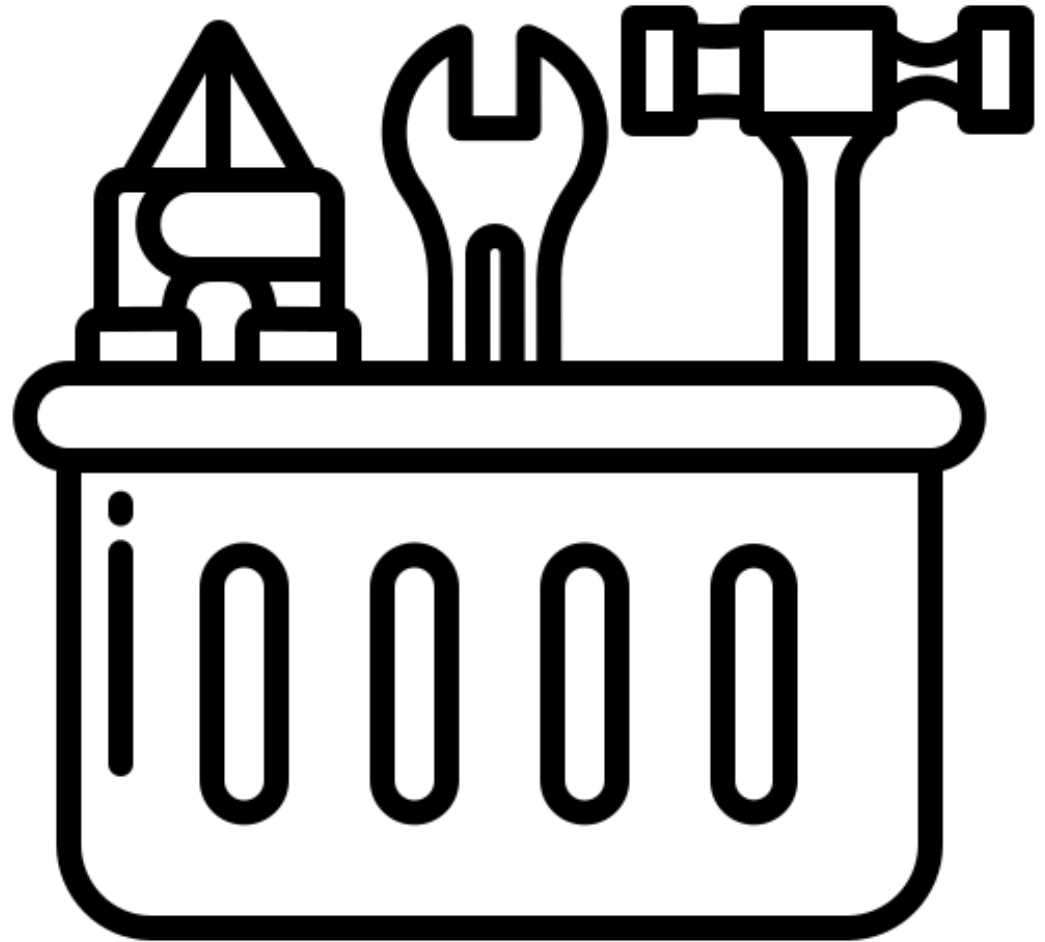
We bioinformaticians are in the AI era

Number of AI+Bioinformatics articles (2000-2019, PubMed)
+ Number of AI articles (2000-2019, 2021 AI index report, Stanford university)



2021 AI index report, Stanford University, 재가공

Great opportunities



Trends

- 2021년, AI Bioinformatics 분야에는 어떤 연구들이 있었나?

AlphaFold2

Article

Highly accurate protein structure prediction with AlphaFold

<https://doi.org/10.1038/s41586-021-03819-2>

Received: 11 May 2021

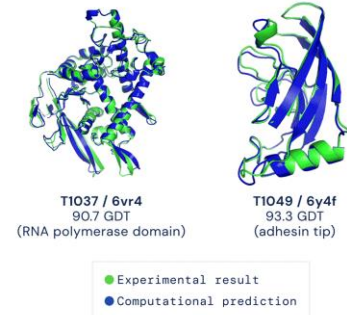
Accepted: 12 July 2021

Published online: 15 July 2021

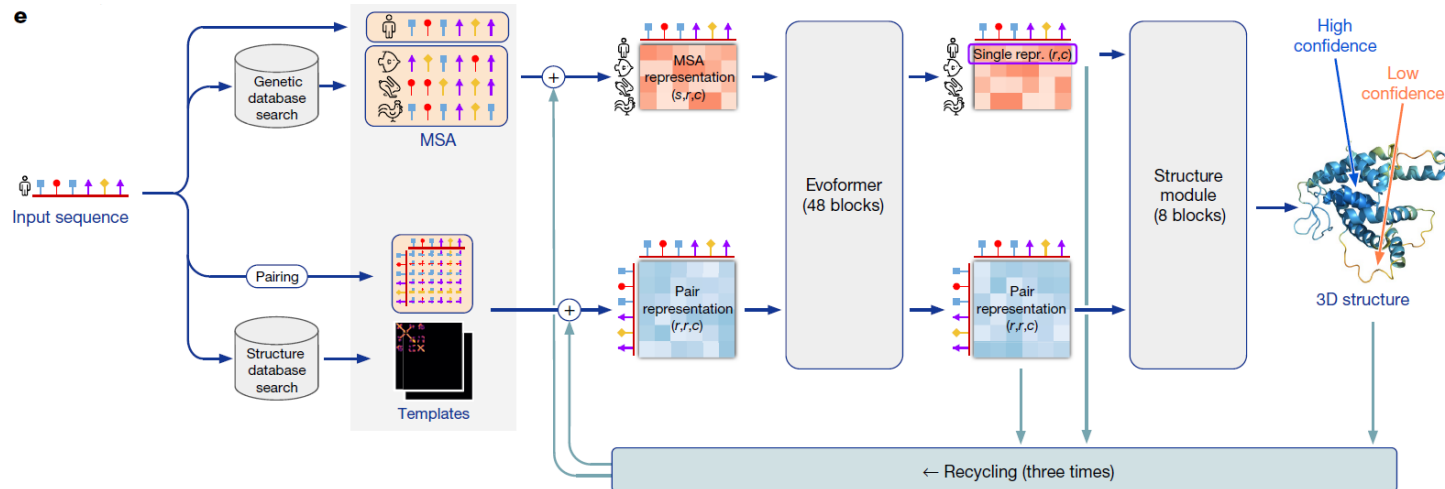
Open access

Check for updates

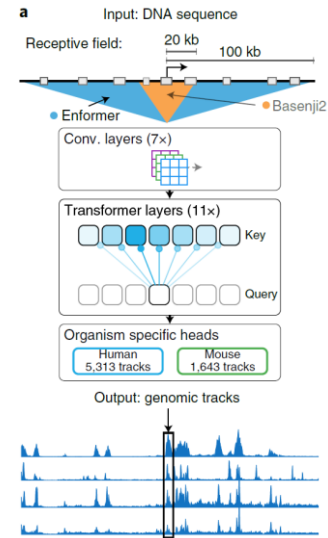
John Jumper^{1,4} , Richard Evans^{1,4}, Alexander Pritzel^{1,4}, Tim Green^{1,4}, Michael Figurnov^{1,4}, Olaf Ronneberger^{1,4}, Kathryn Tunyasuvunakool^{1,4}, Russ Bates^{1,4}, Augustin Židek^{1,4}, Anna Potapenko^{1,4}, Alex Bridgland^{1,4}, Clemens Meyer^{1,4}, Simon A. A. Kohl^{1,4}, Andrew J. Ballard^{1,4}, Andrew Cowie^{1,4}, Bernardino Romera-Paredes^{1,4}, Stanislav Nikolov^{1,4}, Rishub Jain^{1,4}, Jonas Adler¹, Trevor Back¹, Stig Petersen¹, David Reiman¹, Ellen Clancy¹, Michal Zielinski¹, Martin Steinegger^{2,3}, Michalina Pacholska¹, Tamas Berghammer¹, Sebastian Bodenstein¹, David Silver¹, Oriol Vinyals¹, Andrew W. Senior¹, Koray Kavukcuoglu¹, Pushmeet Kohli¹ & Demis Hassabis^{1,4}



아미노산 서열로부터 단백질 3차구조를 예측하는 AI 모델 (DeepMind)



Enformer



ARTICLES **nature methods**

<https://doi.org/10.1038/s41592-021-01252-x>

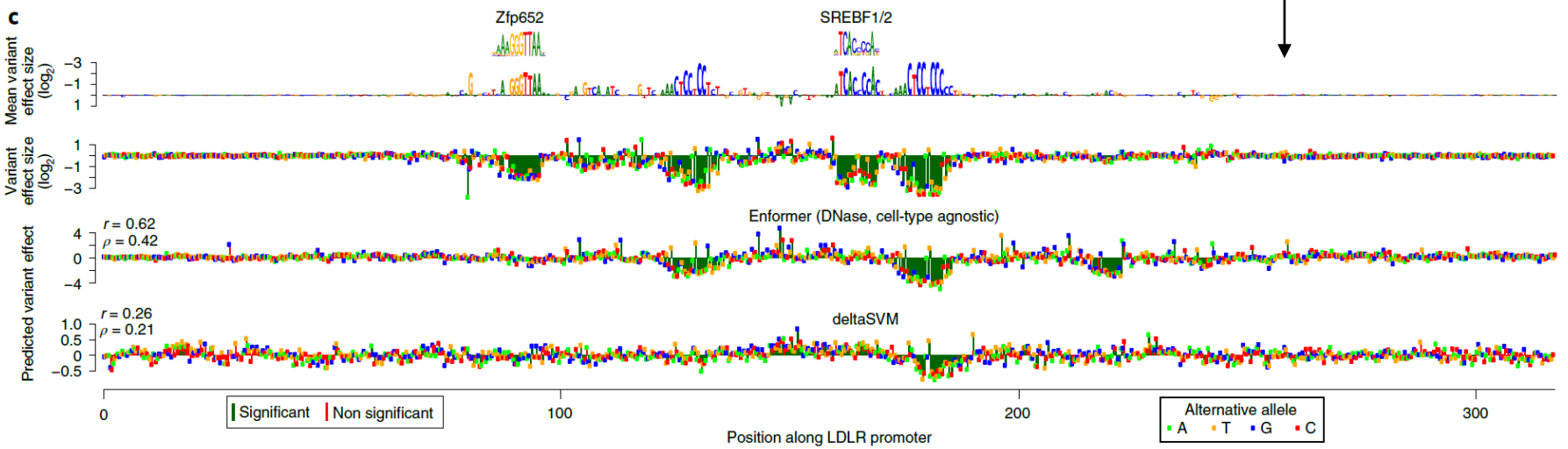
Check for updates

Effective gene expression prediction from sequence by integrating long-range interactions

Žiga Avsec¹, Vikram Agarwal^{2,4}, Daniel Visentin^{1,4}, Joseph R. Ledsam^{1,3}, Agnieszka Grabska-Barwinska¹, Kyle R. Taylor¹, Yannis Assael¹, John Jumper¹, Pushmeet Kohli¹ and David R. Kelley²

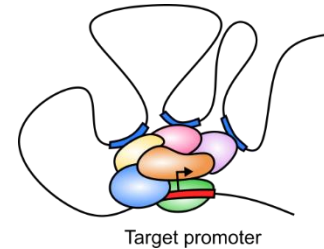
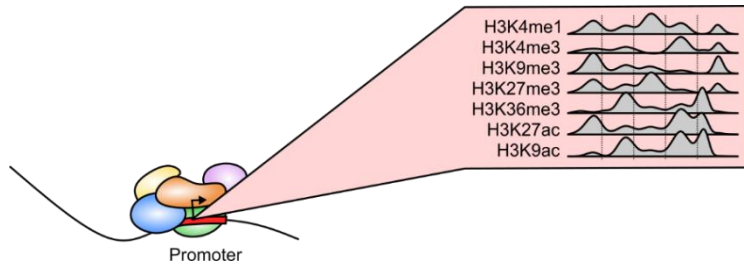
Sequence로부터 유전자 발현량을 예측하는 AI 모델 (DeepMind)

학습 모델 해석을 통해 개별 variant의 effect 예측 가능



Chromoformer

Histone modification과 염색질 3차구조를 이용하여 유전자 발현을 예측하는 AI 모델

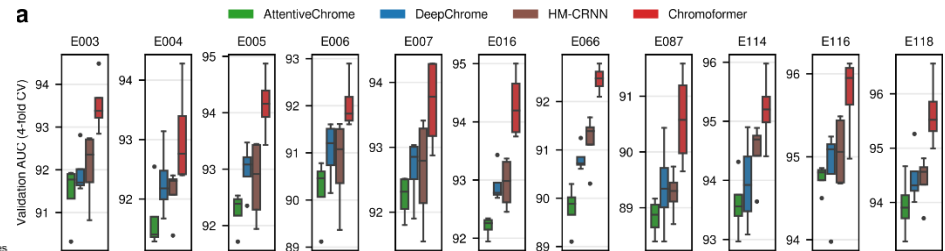
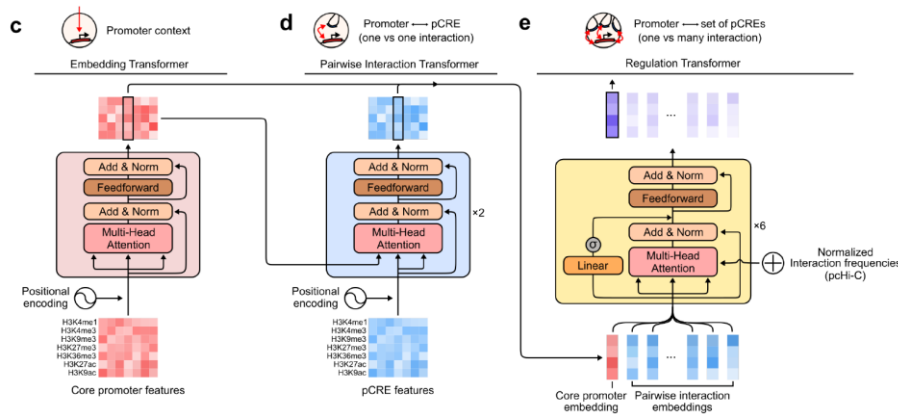


1차원 Histone modification 정보로 유전자 발현을 예측하는 모델은 많다.

이 모델링에 Chromatin 3차원 구조를 반영한다면?

Transformer 써서 모델링 해보자!

성능 향상이 관찰!



EVE (evolutionary model of variant effect)

Article

Disease variant prediction with deep generative models of evolutionary data

<https://doi.org/10.1038/s41586-021-04043-8>

Received: 18 December 2020

Accepted: 20 September 2021

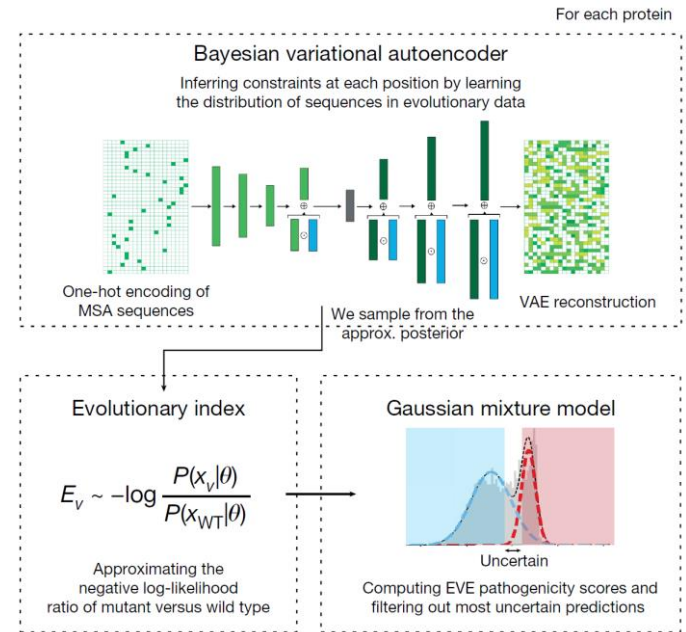
Published online: 27 October 2021

Check for updates

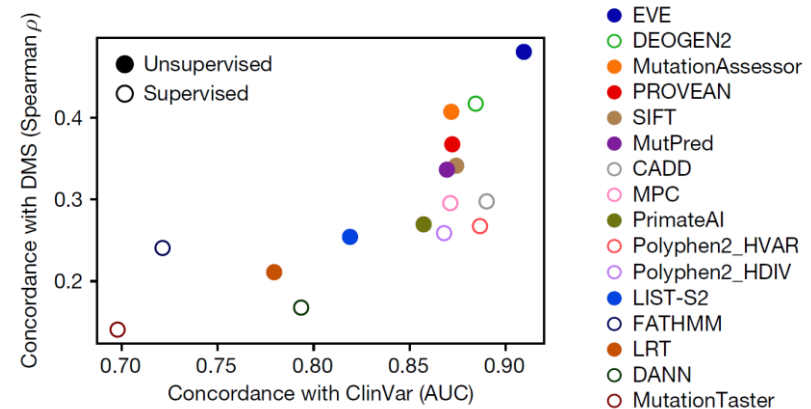
Jonathan Frazer^{1,4}, Pascal Notin^{2,4}, Mafalda Dias^{1,4}, Aidan Gomez², Joseph K. Min¹, Kelly Brock¹, Yarin Gal^{2,5} & Debora S. Marks^{1,3,6}

Quantifying the pathogenicity of protein variants in human disease-related genes would have a marked effect on clinical decisions, yet the overwhelming majority (over 98%) of these variants still have unknown consequences¹⁻³. In principle, computational methods could support the large-scale interpretation of genetic variants. However, state-of-the-art methods⁴⁻¹⁰ have relied on training machine learning models on known disease labels. As these labels are sparse, biased and of variable quality, the resulting models have been considered insufficiently reliable¹¹. Here we propose an approach that leverages deep generative models to predict variant pathogenicity without relying on labels. By modelling the distribution of sequence variation across organisms, we implicitly capture constraints on the protein sequences that maintain fitness. Our model EVE (evolutionary model of variant effect) not only outperforms computational approaches that rely on labelled data but also performs on par with, if not better than, predictions from high-throughput experiments, which are increasingly used as evidence for variant classification¹²⁻¹⁶. We predict the pathogenicity of more than 36 million variants across 3,219 disease genes and provide evidence for the classification of more than 256,000 variants of unknown significance. Our work suggests that models of evolutionary information can provide valuable independent evidence for variant interpretation that will be widely useful in research and clinical settings.

AI 기반 variant pathogenicity 예측 모델 (Broad Institute)



c



EVE (evolutionary model of variant effect)

AI 기반 variant pathogenicity 예측 모델 (Broad Institute)

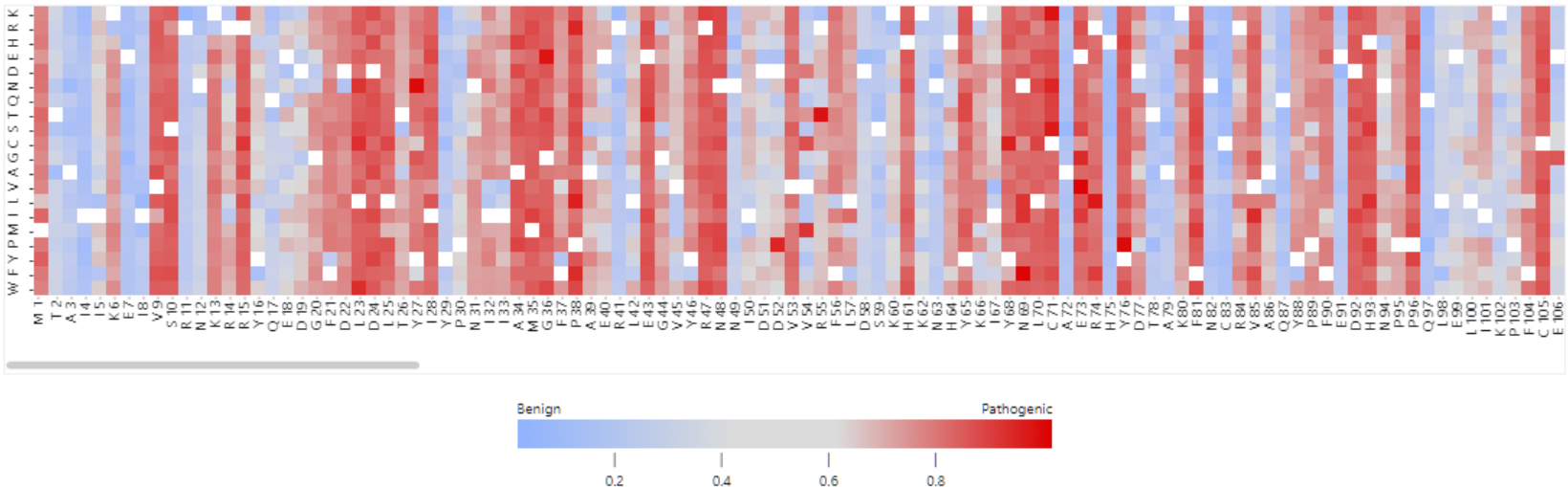
<https://evemodel.org/>

EVE Home Search Download ▾

PTEN_HUMAN

Download Data

[Heatmap](#) | [Variants Table](#) | [Statistics Summary](#)



Heatmap of EVE scores. Positions with no EVE score are shown in white.

Perspectives

- AI에 기반한 non-coding variant 해석으로 전장유전체 해독에 한 걸음 더 가까워질 것으로 기대함
 - Non-coding variant의 effect size는 coding에 비해 작지만, 주로 유전자 발현 조절의 이상을 유발한다는 점에서, multi-omics 를 엮는 factor로서의 무궁무진한 해석 가능성이 있음.
 - 염색질 3차원 구조 정보의 활용이 핵심이 될 것으로 생각

Article

Analyses of non-coding somatic drivers in 2,658 cancer whole genomes

<https://doi.org/10.1038/s41586-020-1965-x>

Received: 19 January 2018

Accepted: 2 December 2019

Published online: 5 February 2020

Open access

Esther Rheinbay^{1,2,3,73}, Morten Muhlig Nielsen^{4,73}, Federico Abascal^{5,73}, Jeremiah A. Wala^{1,6,73}, Ofer Shapira^{1,7,73}, Grace Tiao¹, Henrik Hornshøj¹, Julian M. Hess¹, Randi Istrup Juul⁴, Ziao Lin^{1,8}, Lars Feuerbach⁹, Radhakrishnan Sabarinathan^{10,11}, Tobias Madsen⁴, Jaegil Kim¹, Loris Mularoni^{10,11}, Shimin Shuai^{12,13}, Andrés Lanzós^{14,15,16}, Carl Herrmann^{17,18}, Yosef E. Maruvka^{1,2}, Ciyue Shen^{19,20}, Samirkumar B. Amin^{21,22}, Pratiti Bandopadhyay¹⁷, Johanna Bertl⁴, Keith A. Boroevich²³, John Busanovich¹⁷, Joana Carlevaro-Fita^{14,15,16}, Dimple Chakravarty^{24,25}, Calvin Wing Yiu Chan^{17,26}, David Craft²⁷, Priyanka Dhingra^{28,29}, Klev Diamanti³⁰, Nuno A. Fonseca³¹, Abel Gonzalez-Perez^{32,33}, Qianyun Guo³², Mark P. Hamilton³³, Nicholas J. Haradhvala^{1,2}, Chen Hong^{3,26}, Keren Isaev^{12,34}, Todd A. Johnson²², Malene Juul⁴, Andre Kahles³⁵, Abdullah Kahraman³⁶, Youngwook Kim³⁷, Jan Komorowski^{30,38}, Kiran Kumar¹⁷, Sushant Kumar³⁹, Donghoon Lee³⁹, Kjong-Van Lehmann³⁵, Yilong Li^{40,41}, Eric Minwei Liu^{28,29}, Lucas Lochovsky⁴², Keunchil Park³⁷, Oriol Pich^{10,11}, Nicola D. Roberts⁴¹, Gordon Saksena¹, Steven E. Schumacher¹⁷, Nikos Sidiropoulos⁴³, Lina Sieverling^{3,26}, Nasa Sinnott-Armstrong⁴⁴, Chip Stewart¹, David Tamborero^{10,11}, Jose M. C. Tubio^{45,46,47}, Husen M. Umer²⁰, Liis Uusküla-Reimand^{48,49}, Claes Wadelius⁵⁰, Lina Wadti¹², Xiaotong Yao⁵¹, Cheng-Zhong Zhang^{52,53}, Jing Zhang³⁹, James E. Haber⁵⁴, Asger Hobolth³², Marcin Imielinski^{51,55}, Manolis Kellis^{1,56}, Michael S. Lawrence^{1,2}, Christian von Mering³⁶, Hidewaki Nakagawa⁵⁷, Benjamin J. Raphael⁵⁸, Mark A. Rubin^{59,60,61}, Chris Sander^{39,20}, Lincoln D. Stein^{12,13}, Joshua M. Stuart⁶², Tatsuhiro Tsunoda^{23,63,64}, David A. Wheeler⁶⁵, Rory Johnson^{14,16}, Jüri Reimand^{12,34}, Mark Gerstein^{39,42,66}, Ekta Khurana^{39,28,60,67}, Peter J. Campbell^{13,41}, Núria López-Bigas^{10,11,67}, PCAWG Drivers and Functional Interpretation Working Group⁶⁸, PCAWG Structural Variation Working Group⁶⁹, Joachim Weischenfeldt^{13,68,74}, Rameen Beroukhi^{1,6,70,74}, Iñigo Martincorena^{3,74}, Jakob Skou Pedersen^{4,32,74}, Gad Getz^{1,2,3,71,74} & PCAWG Consortium⁷²



A bit disappointing,
but great starting point

Acknowledgement

2020 Lab Members



Prof. Sun Kim
Bio & Health Lab, SNU

<https://bhi-kimlab.github.io/>

감사합니다