KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML)
Workshop for Life and Medical Scientists

생명정보학 & 머신러닝 워크샵 (온라인)

Deep learning based prediction on noncoding variants from whole genome sequencing data

안준용 _ 고려대학교

KSBI KOREAN SOCIETY FOR BIOINFORMATICS | 한국생명정보학회

본 강의 자료는 한국생명정보학회가 주관하는 BIML 2024 워크샵 온라인 수업을 목적으로 제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우 발생하는 **모든 법적 책임은 전적으로 불법 행위자 본인에게 있음을 경고**합니다.

KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML)
Workshop for Life and Medical Scientists

안녕하십니까?

한국생명정보학회가 개최하는 동계 교육 워크샵인 BIML-2024에 여러분을 초대합니다. 생명정보학 분야의 연구자들에게 최신 동향의 데이터 분석기술을 이론과 실습을 겸비해 전달하고자 도입한 전문 교육 프로그램인 BIML 워크샵은 2015년에 시작하여 올해로 벌써 10년 차를 맞이하게 되었습니다. BIML 워크샵은 국내 생명정보학 분야의 최초이자 최고 수준의 교육프로그램으로 크게 인공지능과 생명정보분석 두 개의 분야로 구성되어 있습니다. 올해 인공지능 분야에서는 최근 생명정보 분석에서도 응용이 확대되고 있는 다양한 인공지능 기반 자료모델링 기법들에 대한 현장 강의가 진행될 예정이며, 관련하여 심층학습을 이용한 단백질구조예측, 유전체분석, 신약개발에 대한 이론과 실습 강의가 함께 제공될 예정입니다. 또한 단일세포오믹스, 공간오믹스, 메타오믹스, 그리고 롱리드염기서열 자료 분석에 대한 현장 강의는 많은 연구자의 연구 수월성 확보에 큰 도움을 줄 것으로 기대하고 있습니다.

올해 BIML의 가장 큰 변화는 최근 연구 수요가 급증하고 있는 의료정보자료 분석에 대한 현장 강의를 추가하였다는 것입니다. 특히 의료정보자료 분석을 많이 수행하시는 의과학자 및 의료정보 연구자들께서 본 강좌를 통해 많은 도움을 받으실 수 있기를 기대하고 있습니다. 또한 다양한 생명정보학 분야에 대한 온라인 강좌 프로그램도 점차 증가하고 있는 생명정보 분석기술의 다양화에 발맞추기 위해 작년과 비교해 5강좌 이상을 신규로 추가했습니다. 올해는 무료 강좌 5개를 포함하여 35개 이상의 온라인 강좌가 개설되어 제공되며, 연구 주제에 따른 연관된 강좌 추천 및 강연료 할인 프로그램도 제공되며, 온라인을 통한 Q&A 세션도 마련될 예정입니다. BIML-2024는 국내 주요 연구중심 대학의 전임 교원이자 각 분야 최고 전문가들의 강의로 구성되었기에 해당 분야의 기초부터 최신 연구 동향까지 포함하는 수준 높은 내용의 강의가 될 것이라 확신합니다.

BIML-2024을 준비하기까지 너무나 많은 수고를 해주신 운영위원회의 정성원, 우현구, 백대현, 김태민, 김준일, 김상우, 장혜식, 박종은 교수님과 KOBIC 이병욱 박사님께 커다란 감사를 드립니다. 마지막으로 부족한 시간에도 불구하고 강의 부탁을 흔쾌히 허락하시고 훌륭한 현장 강의와 온라인 강의를 준비하시는데 노고를 아끼지 않으신 모든 강사분들께 깊은 감사를 드립니다.

2024년 2월

한국생명정보학회장 이 인 석

강의개요

Deep learning based prediction on noncoding variants from whole genome sequencing data

전장유전체는 인간 유전체 모든 지역에 발생하는 다양한 유전변이를 발굴하고, 이를 통해 질병의 유전연관성을 평가한다. 상당 수의 변이들은 논코딩 유전체에 발생하여, 전사조절 및 유전자 발현에 관여하며, 병발생의 시공간적 특이성을 나타낸다. 따라서, 전장유전체 해석을 위해, 최근 발달한 다양한 기능유전체 데이터를 통합하고 평가할 방법에 대한 이해가 필수적이다.

본 강의에서는 전장유전체 데이터로부터 유전변이 예측을 하기 위한 분석 법을 소개한다. 대규모 전장유전체를 효과적이고 빠르게 처리하기 위해 널리 쓰이고 있는 플랫폼인 Hail을 소개하고, 분석에 활용할 유전변이 선별 작업을 시행한다. 딥러닝 기반 알고리즘을 활용하여 각 유전변이의 전사조절 변화 및 위험도를 평가하는 방법을 학습한다.

강의는 다음의 내용을 포함한다:

- 전장유전체 데이터 개요
- 전장유전체 데이터 처리를 위한 Hail 플랫폼 학습
- 딥러닝 기반 유전변이 평가 방법 학습

* 참고강의교재:

- Hail 웹사이트 (https://hail.is/)
- Avsec et al. (2021) Nature Methods

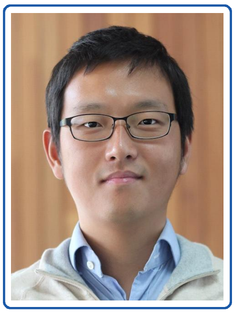
* 교육생준비물 및 필요조건: 구글 코랩, 인터넷

* 강의 난이도: 중급

* 강의: 안준용 교수 (고려대학교 바이오시스템의과학부)

Curriculum Vitae

Speaker Name: Joon-Yong An, Ph.D.



▶ **Personal Info**

Name	Joon-Yong An
Title	Assistant Professor
Affiliation	Korea University

▶ **Contact Information**

Address	145 Anam-ro, Seongbuk-gu, Seoul, South Korea
Email	joonan30@korea.ac.kr
Phone Number	02-3290-5646

Research Interest

Whole genome sequencing, Single cell RNA sequencing, and neurodevelopmental disorders

Educational Experience

2010	B.S. in Molecular Biotechnology, Konkuk University
2011	M.S. in Molecular Biology, University of Queensland (Australia)
2016	Ph.D. in Neuroscience, University of Queensland (Australia)

Professional Experience

2015-2019	Postdoctoral Fellow , University of California, San Francisco
2019-	Assistant Professor, Korea University

Selected Publications (5 maximum)

1. Choi & An, Genetic architecture of autism spectrum disorder: Lessons from large-scale genomic studies, Neuroscience & Biobehavioral Reviews, 2021
2. Werling DW*, Pochareddy S*, JM Choi*, **An JY***, Peng M, ..., Roeder K, Devlin B, Sanders SJ**, Sestan N**, Whole-genome and RNA sequencing reveal variation and transcriptomic coordination in the developing human prefrontal cortex, Cell Reports, 2020
3. Satterstrom FK*, Kosmicki JA*, Wang J*, Breen MS, Rubeis SD, **An JY**, ..., Talkowski ME**, Cutler DJ**, Devlin B**, Sanders SJ**, Roeder K**, Daly MJ**, Buxbaum JD**, Large-scale exome sequencing study implicates both developmental and functional changes in the neurobiology of autism, ***Cell***, 2020
4. **An JY***, Lin K*, Zhu L*, Werling DM*, ..., Talkowski ME**, Devlin B**, Roeder K**, Sanders SJ**, Genome-wide de novo risk score implicates promoter variation in autism spectrum disorder, Science, 2018
5. Werling DM*, Brand H*, **An JY***, Stone MR*, Zhu L*, ..., Devlin B**, Talkowski M**, Sanders SJ**, An analytical framework for whole genome sequence data and its implications for autism spectrum disorder, ***Nature Genetics***, 50:727736, 2018

KSBi-BIML 2023
Deep learning based prediction on noncoding variants from whole genome sequencing data

Joon-Yong An
Korea University

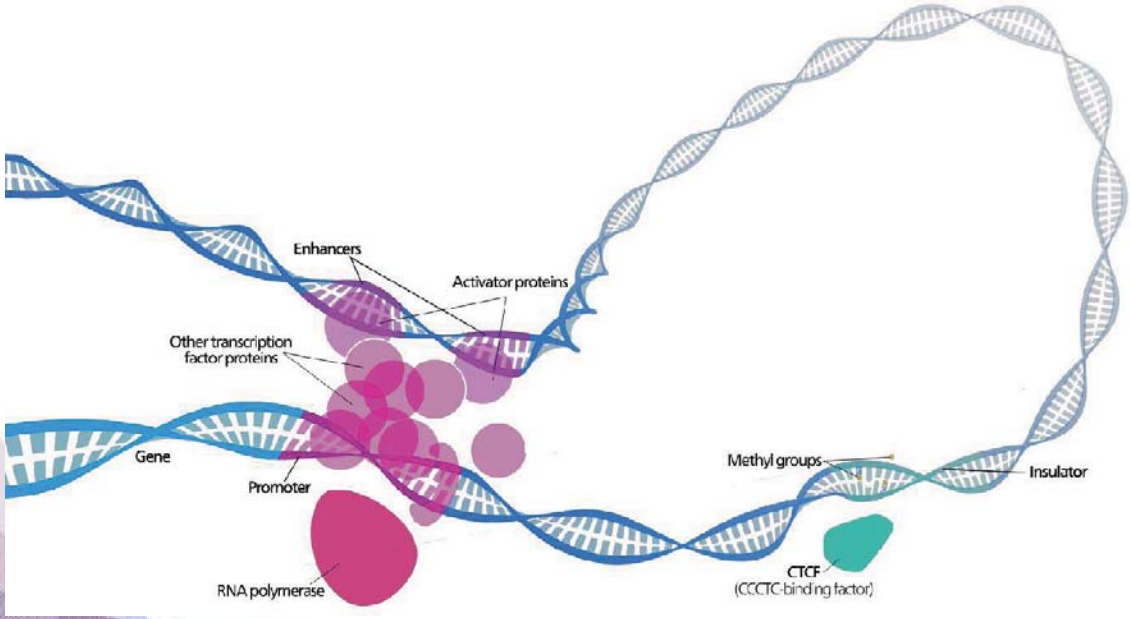
Overview

- Deep learning based approaches for whole genome sequencing studies
- Implication in biological and clinical research
- Tutorial for deep-learning prediction on noncoding variants from WGS data
  - Subset noncoding and qualifying variants from WGS data
  - Predict functional impact of noncoding variants using a deep learning method

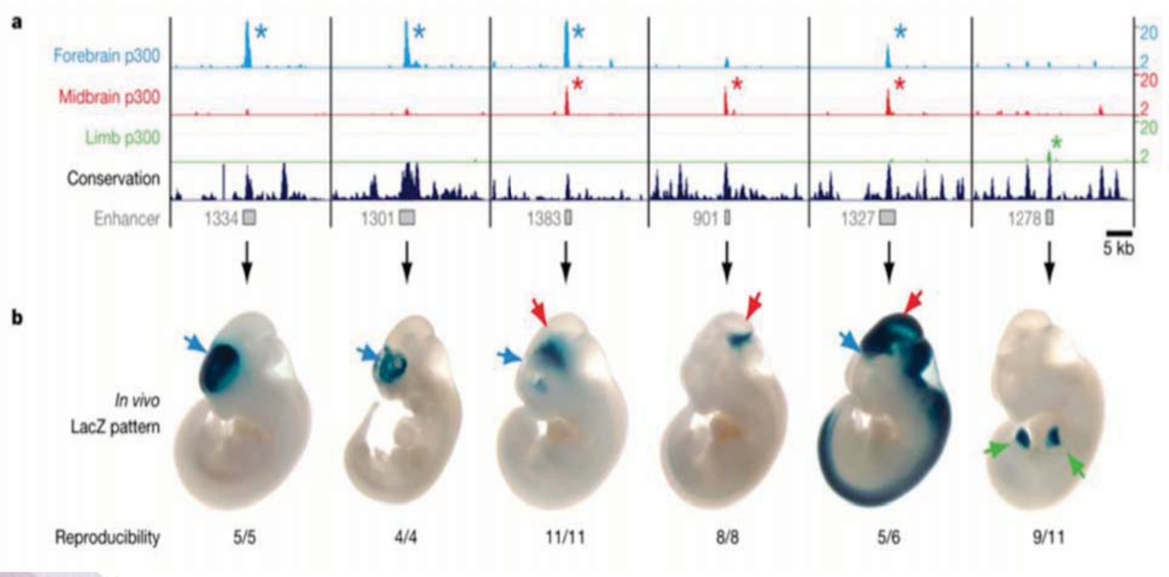
Why whole genome sequencing (WGS)?

- Find all genetic variants in our genome
  - From single nucleotide variants to structural variants
  - Genome-wide association with traits or disorders
  - Assess rare variants in noncoding genome
- Find pathogenic noncoding mutations for disorder
  - Pathogenic: contributes mechanistically to disease, but is not necessarily fully penetrant (i.e., may not be sufficient in isolation to cause disease).

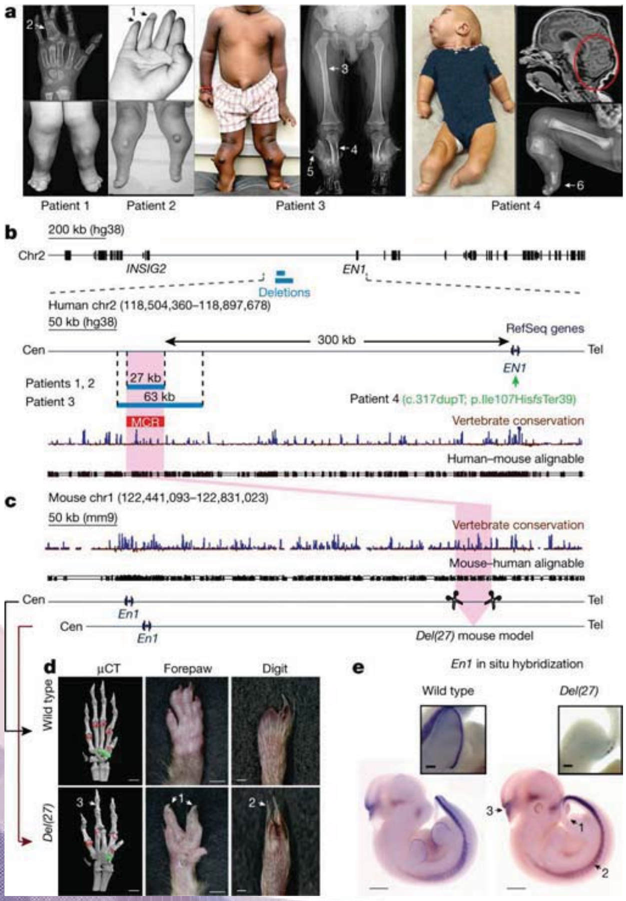
Regulatory elements in noncoding genome



Regulatory elements fine-tune development and cell types



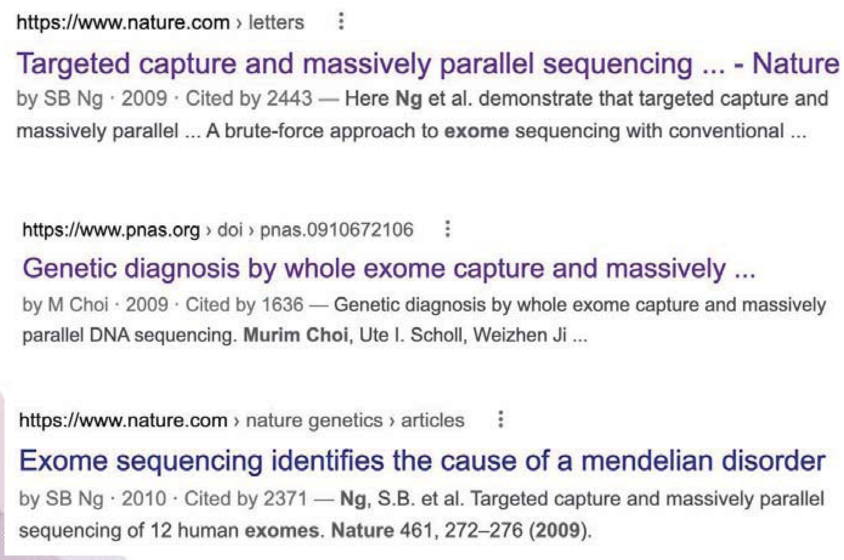
Visel et al. 2009, Nature



- Homozygous 27–63-kilobase deletions located 300 kilobases upstream of the engrailed-1 gene (EN1) in patients with a complex limb malformation featuring mesomelic shortening, syndactyly and ventral nails (dorsal dimelia)

Allou et al. 2021, Nature

Before we get into noncoding, we need to think about how to evaluate coding mutations in disease...



https://www.nature.com › letters

Targeted capture and massively parallel sequencing ... - Nature

by SB Ng · 2009 · Cited by 2443 — Here **Ng** et al. demonstrate that targeted capture and massively parallel ... A brute-force approach to **exome** sequencing with conventional ...

https://www.pnas.org › doi › pnas.0910672106

Genetic diagnosis by whole exome capture and massively ...

by M Choi · 2009 · Cited by 1636 — Genetic diagnosis by whole exome capture and massively parallel DNA sequencing. **Murim Choi**, Ute I. Scholl, Weizhen Ji ...

https://www.nature.com › nature genetics › articles

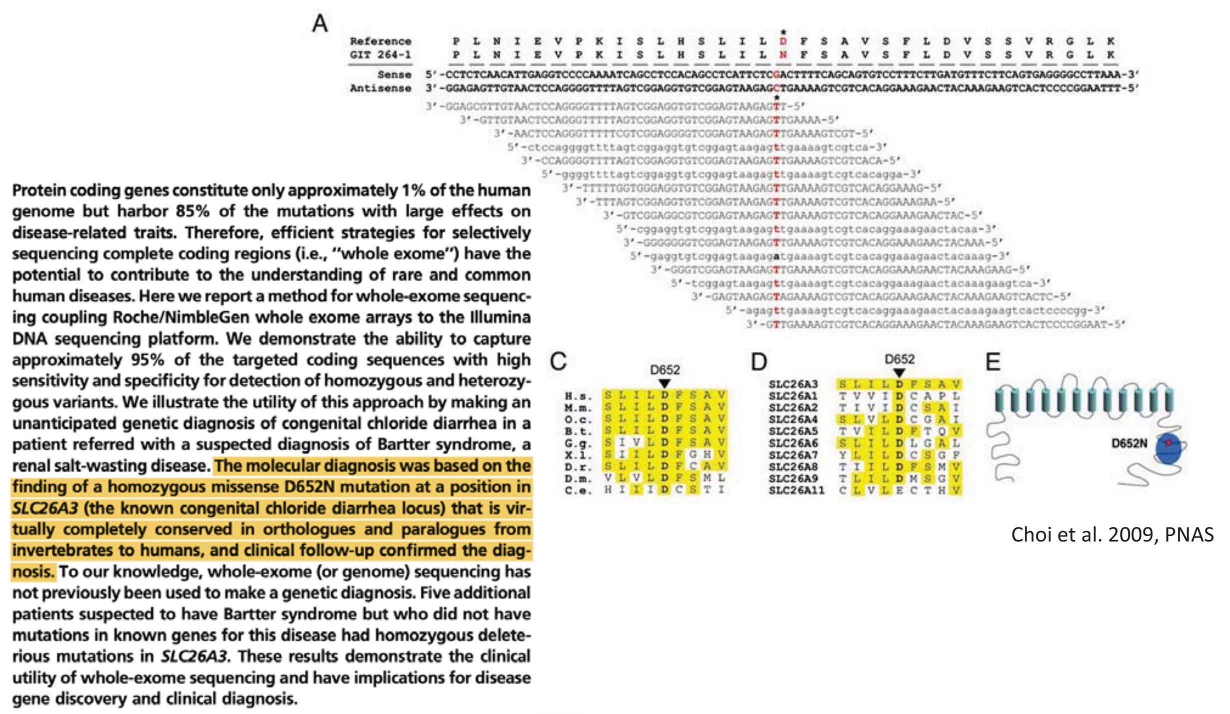
Exome sequencing identifies the cause of a mendelian disorder

by SB Ng · 2010 · Cited by 2371 — **Ng**, S.B. et al. Targeted capture and massively parallel sequencing of 12 human **exomes**. **Nature** 461, 272–276 (**2009**).

Proof-of-concept for exome sequencing (NGS)

First exome analysis for human disease

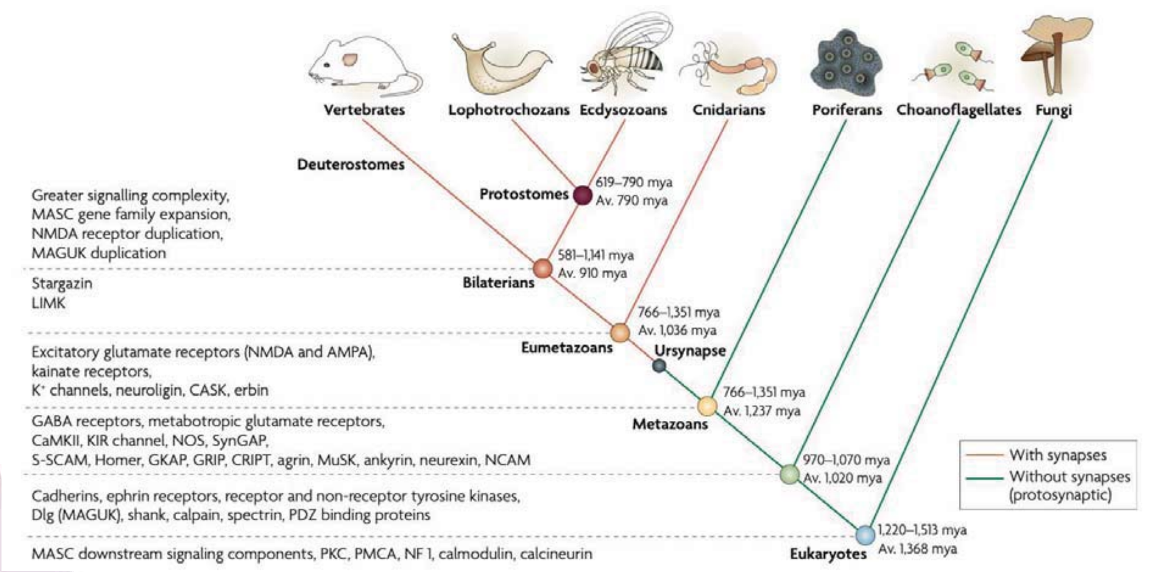
How to prioritize “pathogenic” variant from numerous variants in our genome


A
Reference
GIT 264-1
Sense
Antisense
C
D652
H.s.
M.m.
O.c.
B.t.
G.g.
X.l.
D.r.
D.m.
C.e.
D
D652
SLC26A3
SLC26A1
SLC26A2
SLC26A4
SLC26A5
SLC26A6
SLC26A7
SLC26A8
SLC26A9
SLC26A11
E
D652N

Protein coding genes constitute only approximately 1% of the human genome but harbor 85% of the mutations with large effects on disease-related traits. Therefore, efficient strategies for selectively sequencing complete coding regions (i.e., “whole exome”) have the potential to contribute to the understanding of rare and common human diseases. Here we report a method for whole-exome sequencing coupling Roche/NimbleGen whole exome arrays to the Illumina DNA sequencing platform. We demonstrate the ability to capture approximately 95% of the targeted coding sequences with high sensitivity and specificity for detection of homozygous and heterozygous variants. We illustrate the utility of this approach by making an unanticipated genetic diagnosis of congenital chloride diarrhea in a patient referred with a suspected diagnosis of Bartter syndrome, a renal salt-wasting disease. The molecular diagnosis was based on the finding of a homozygous missense D652N mutation at a position in *SLC26A3* (the known congenital chloride diarrhea locus) that is virtually completely conserved in orthologues and paralogues from invertebrates to humans, and clinical follow-up confirmed the diagnosis. To our knowledge, whole-exome (or genome) sequencing has not previously been used to make a genetic diagnosis. Five additional patients suspected to have Bartter syndrome but who did not have mutations in known genes for this disease had homozygous deleterious mutations in *SLC26A3*. These results demonstrate the clinical utility of whole-exome sequencing and have implications for disease gene discovery and clinical diagnosis.

Choi et al. 2009, PNAS

Conservation in DNA or protein sequences indicate functional units in biological system


Vertebrates
Lophotrochozans
Ecdysozoans
Cnidarians
Poriferans
Choanoflagellates
Fungi
Deuterostomes
Protostomes
619–790 mya
Av. 790 mya
Greater signalling complexity,
MASC gene family expansion,
NMDA receptor duplication,
MAGUK duplication
581–1,141 mya
Av. 910 mya
Bilaterians
Stargazin
LIMK
766–1,351 mya
Av. 1,036 mya
Eumetazoans
Ursynapse
Excitatory glutamate receptors (NMDA and AMPA),
kainate receptors,
K⁺ channels, neuroligin, CASK, erbin
766–1,351 mya
Av. 1,237 mya
Metazoans
GABA receptors, metabotropic glutamate receptors,
CaMKII, KIR channel, NOS, SynGAP,
S-SCAM, Homer, GKAP, GRIP, CRIPT, agrin, MuSK, ankyrin, neurexin, NCAM
970–1,070 mya
Av. 1,020 mya
With synapses
Without synapses (protosynaptic)
Cadherins, ephrin receptors, receptor and non-receptor tyrosine kinases,
Dlg (MAGUK), shank, calpain, spectrin, PDZ binding proteins
1,220–1,513 mya
Av. 1,368 mya
MASC downstream signaling components, PKC, PMCA, NF 1, calmodulin, calcineurin
Eukaryotes

Ryan & Grant 2009, Nat Rev Neuro

Priotizing variants by sequence conservation

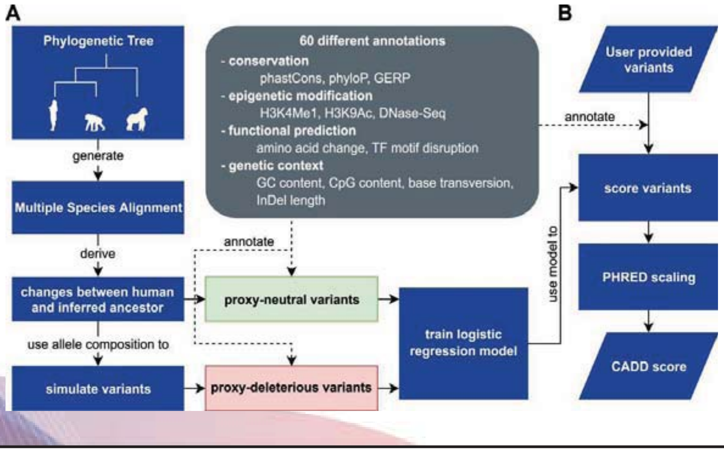
- Nucleotide conservation
  - GERP, PhyloP, PhastCons
- Protein sequence conservation
  - SIFT, PolyPhen2

- These methods has scoring system at each locus or allele (variant)
  - chr1:122222:A:T is SIFT score 0.01, indicating “deleterious” or chr11:122222:C:G is SIFT score 0.13, indicating “benign”
  - chr3:13322 is phylop score 0.5, indicating “highly conserved” or chr3:13300 is phylop score 0.02, indicating “less conserved”

- If you find ~30,000 variants in coding genome, how can we find which scoring system is good?

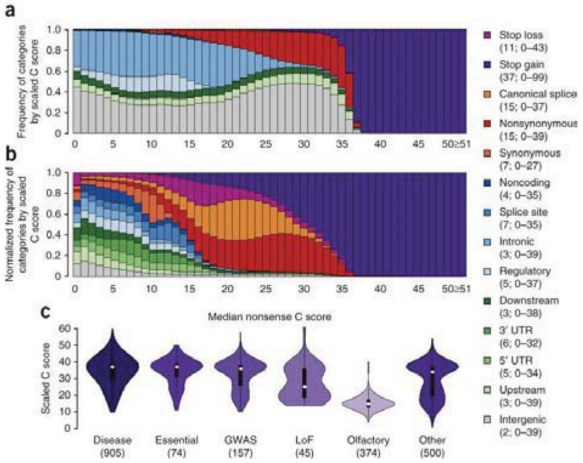
“Ensemble methods”

- Integrate the results of multiple individual predictors can improve performance
- CADD (Combined Annotation–Dependent Depletion) by Kircher et al. (2014) Nat Gen
  - Training a support vector machine (SVM) with a linear kernel on features derived from the 63 annotations
- Condel, DANN, Eigen, MetaSVM, MetaLR, KGGSeq, REVEL, LINSIGHT, GWAVA
  - Supervised or unsupervised learning for predictive features for pathogenic variants

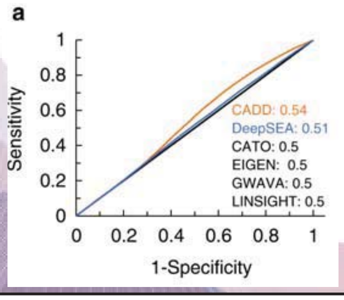

A
B
Phylogenetic Tree
generate
Multiple Species Alignment
derive
changes between human and inferred ancestor
use allele composition to
simulate variants
60 different annotations
- conservation
phastCons, phyloP, GERP
- epigenetic modification
H3K4Me1, H3K9Ac, DNase-Seq
- functional prediction
amino acid change, TF motif disruption
- genetic context
GC content, CpG content, base transversion, InDel length
annotate
proxy-neutral variants
proxy-deleterious variants
train logistic regression model
use model to
User provided variants
annotate
score variants
PHRED scaling
CADD score

Kircher et al. (2014) Nat Gen

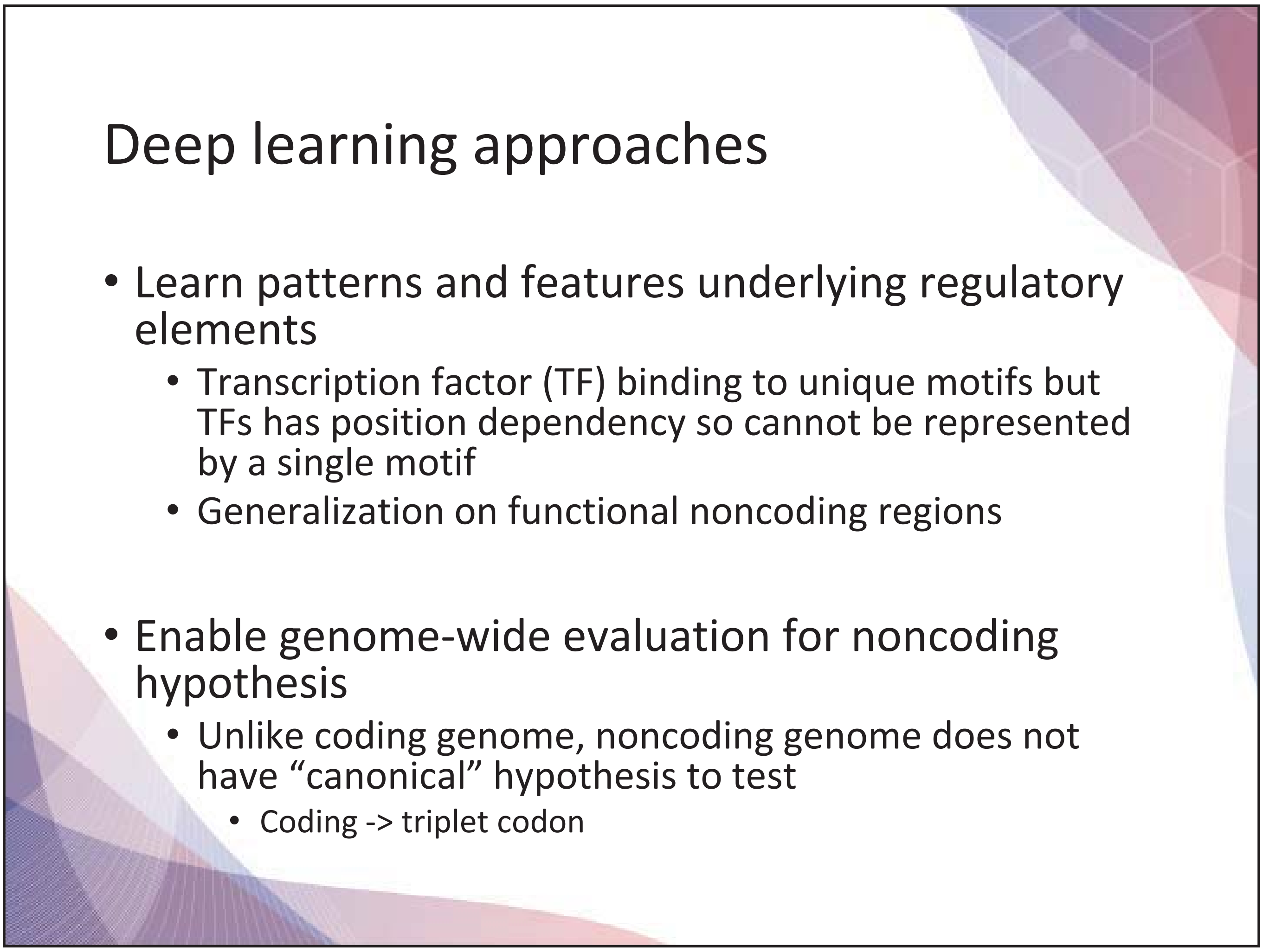
Low predictive power on noncoding mutations



Kircher et al. (2014) Nat Gen



“Biological relevance of computationally predicted pathogenicity of noncoding variants”
Liu et al. (2019) Nat Comm



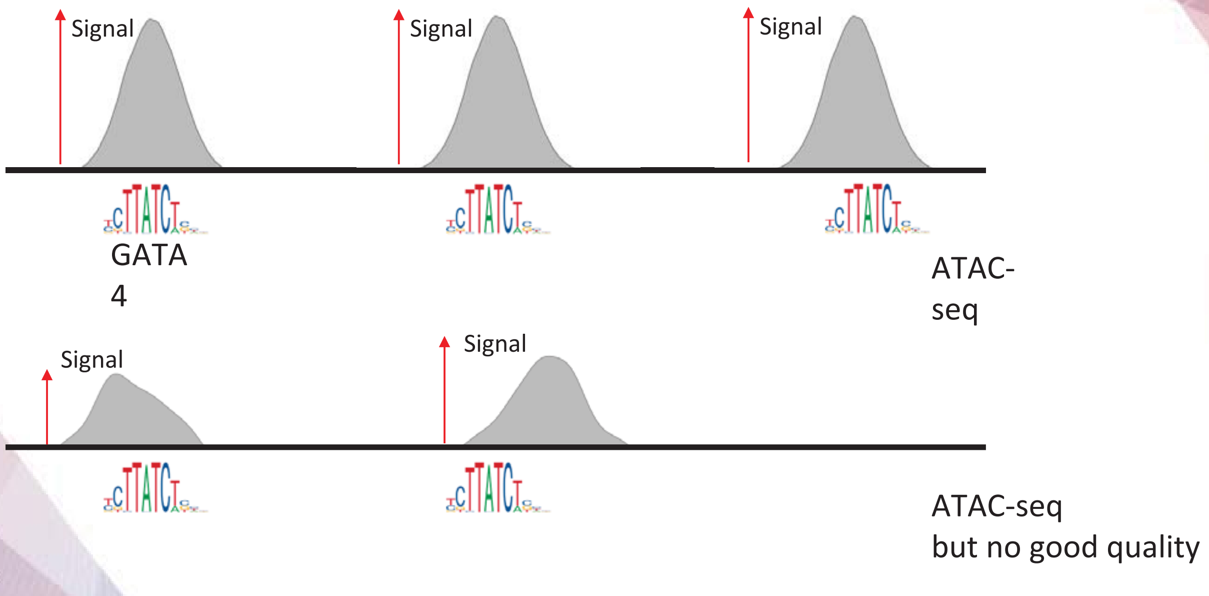
Deep learning approaches

- Learn patterns and features underlying regulatory elements
  - Transcription factor (TF) binding to unique motifs but TFs has position dependency so cannot be represented by a single motif
  - Generalization on functional noncoding regions
- Enable genome-wide evaluation for noncoding hypothesis
  - Unlike coding genome, noncoding genome does not have “canonical” hypothesis to test
    - Coding -> triplet codon

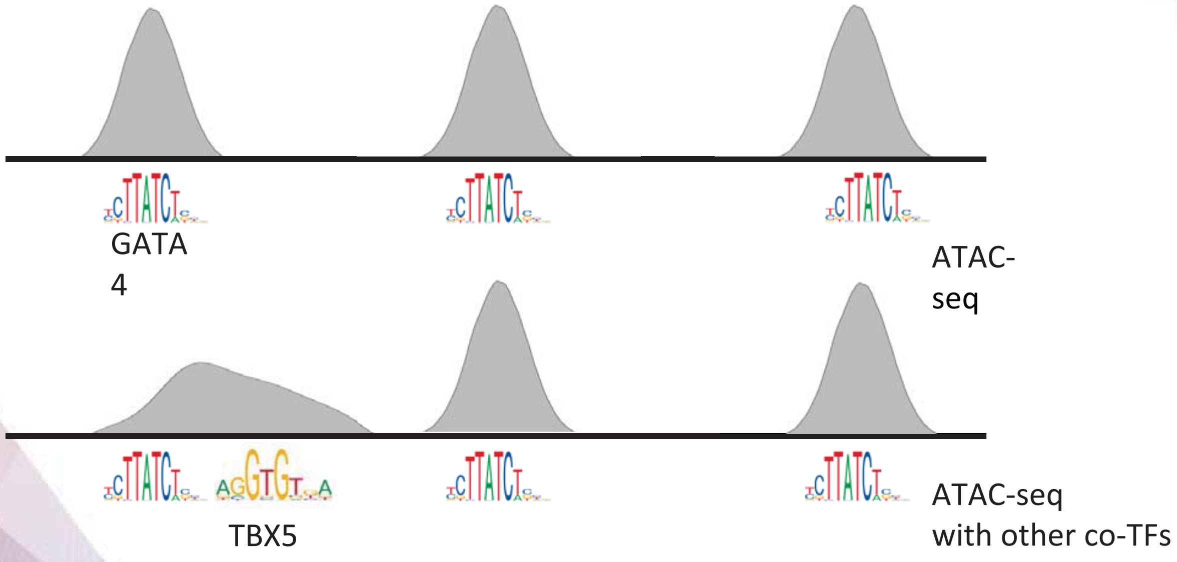
Deep learning approaches

- DeepSEA (2015)
- Basenji2 (2020)
- Enformer (2021)
- Sei (2022)
- Orca (2022)

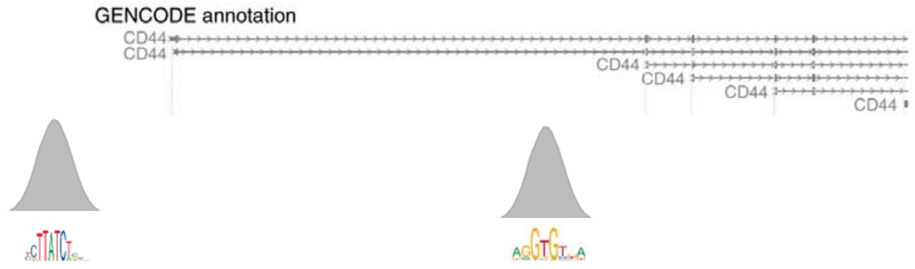
There are techniques to find regulatory elements in noncoding genome but...



There are techniques to find regulatory elements in noncoding genome but...



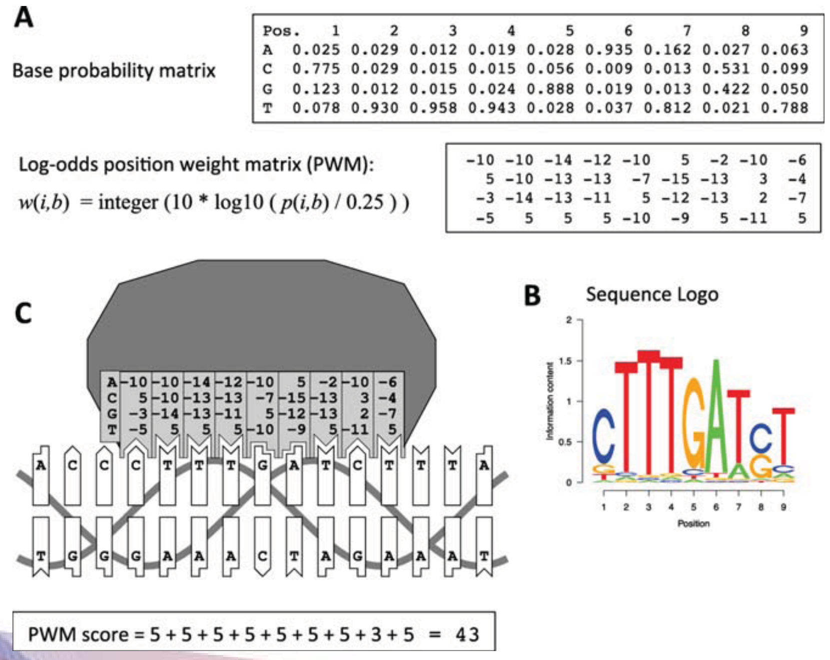
Predict functional regions in noncoding genome



Prediction

Prediction

TFs has position dependency



A

Base probability matrix

Pos.	1	2	3	4	5	6	7	8	9
A	0.025	0.029	0.012	0.019	0.028	0.935	0.162	0.027	0.063
C	0.775	0.029	0.015	0.015	0.056	0.009	0.013	0.531	0.099
G	0.123	0.012	0.015	0.024	0.888	0.019	0.013	0.422	0.050
T	0.078	0.930	0.958	0.943	0.028	0.037	0.812	0.021	0.788

Log-odds position weight matrix (PWM):

$w(i,b) = \text{integer}\,(10 * \log10\,(\,p(i,b)\,/\,0.25\,)\,)$

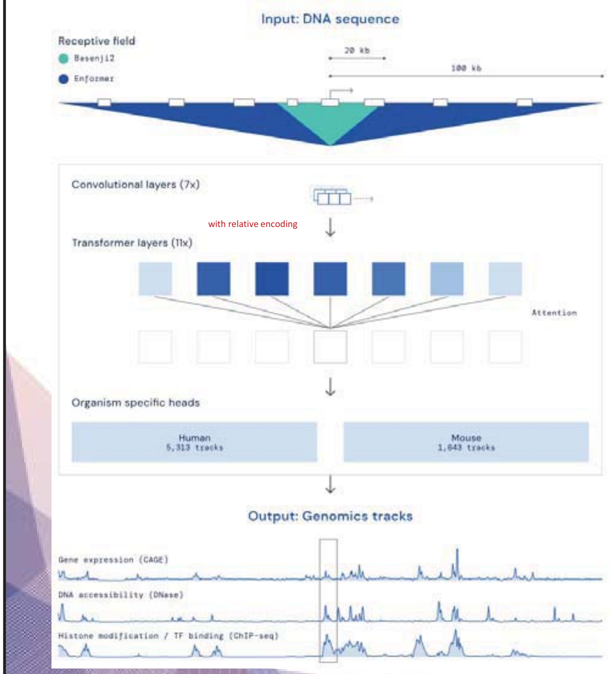
-10	-10	-14	-12	-10	5	-2	-10	-6
5	-10	-13	-13	-7	-15	-13	3	-4
-3	-14	-13	-11	5	-12	-13	2	-7
-5	5	5	5	-10	-9	5	-11	5

PWM score = 5 + 5 + 5 + 5 + 5 + 5 + 5 + 3 + 5 = 43

Enformer (2021)

- Deep learning model architecture composed of convolution layers and attention layers enabling the expansion of input sequences up to 200 kb
- Training dataset including 5,313 human and 1,643 mouse genomic signal tracks
  - Cell type specific information
- Attention layers outperformed dilated convolutions across all model sizes, numbers of layers, and numbers of training data points

Enformer (2021)



Avsec et al. 2021, Nature Methods

- 200 kb genomic sequences with variants
- The receptive field (window size) is increased compared to Basenji2, due to using attention layers

Convolutional layers with pooling

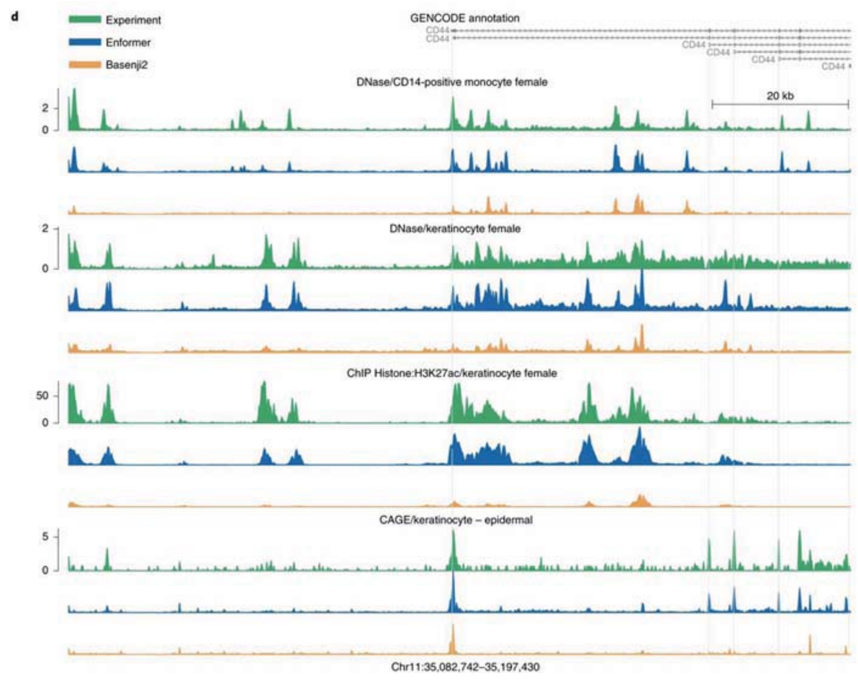
- To reduce the spatial dimension from 196,608 bp to 1,536 each 128 bp bins, i.e. feature extraction
- Use attention pooling in place of max pooling in Basenji2

Transformer layers

- To capture long-range interactions across the sequence, and encode as relative position
- Relative positional encodings provide a parameterized baseline for how actively two positions in the sequence should influence each other during the layer's transformation as a function of their pairwise distance

- Enformer trained for multi genomic epigenetic signal including human and mouse dataset
- Predicted genomic tracks for each chromatin feature used to the training model

Enformer (2021)



Function prediction of noncoding variants using enformer

1. Input
   - 196 kb sequences with variants transformed to one-hot encoded matrix

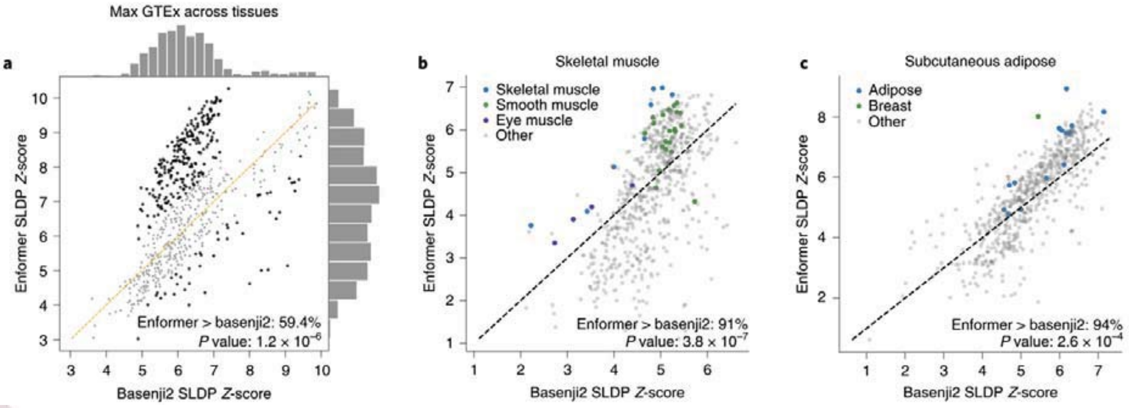
2. Output
   - Genomic sequence track for each chromatin feature
   - Contribution score of prioritizing enhancer-gene pairs with a sequence-based model, computing the gradient of the CAGE target at the TSS for input
   - The enhancer–gene score was obtained by summing the absolute gradient × input scores in the 2-kb window centered at the enhancer
   - Variant scores computed by the difference between alternative predictions and reference predictions

Function prediction of noncoding variants using enformer

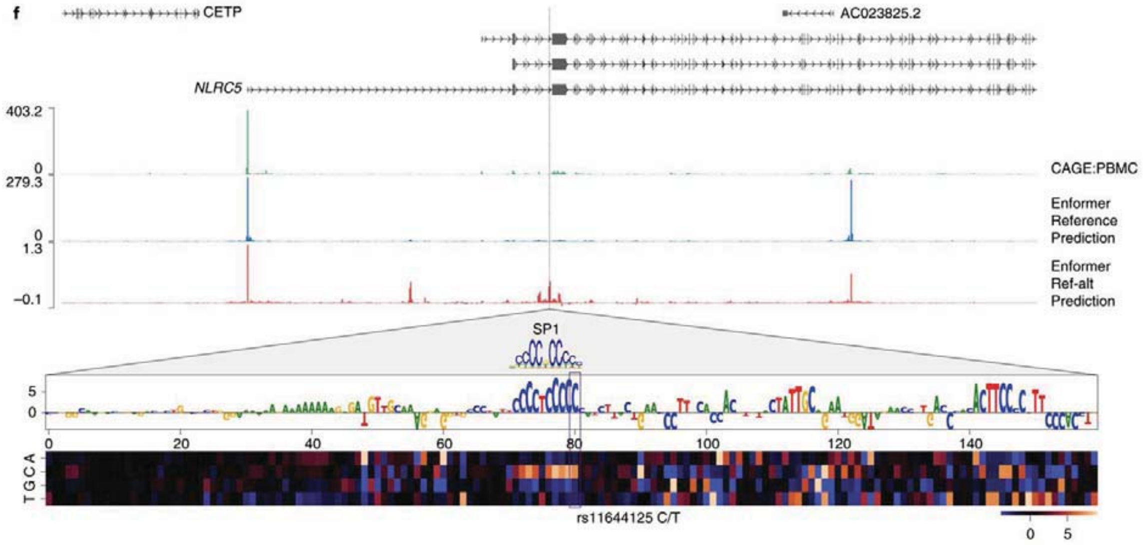
3. Training/test/validation dataset

- Training, testing, and validation dataset (5,313 human and 1,613 mouse)
- Trained, evaluated, and tested on the same targets, genomic intervals, and using Poisson negative log-likelihood loss function to minimize residual between predicted and observed values
- Sequences data: reference genomes of human (GRCh38) and mouse (mm10) divided into 1Mb regions
  - 34,021 training, 2,213 validation, and 1,937 test sequences for the human genome
  - 29,295 training, 2,209 validation, and 2,017 test sequences for the mouse genome
- Available to train simultaneously for human and mouse genome

Function prediction of noncoding variants using enformer

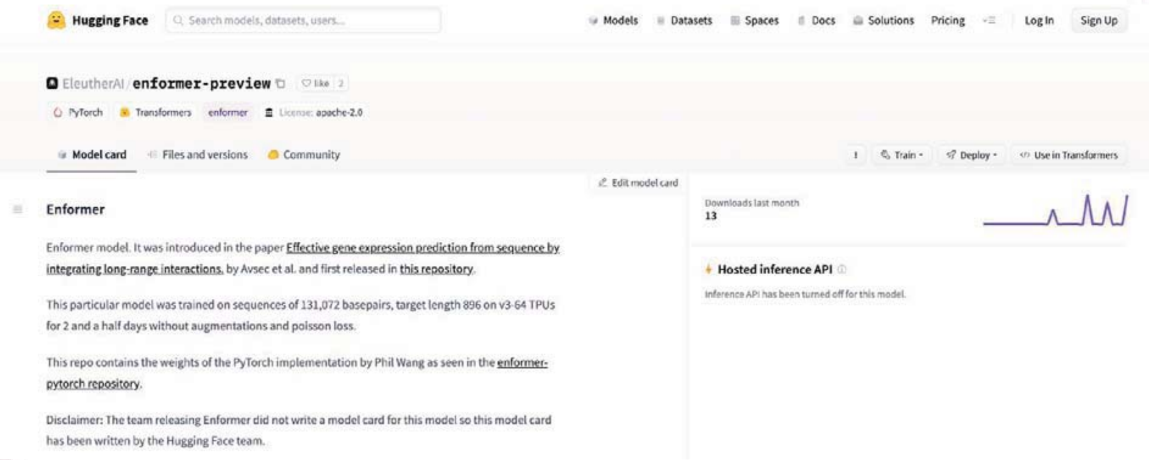


Function prediction of noncoding variants using enformer



The variant rs11644125 is associated with changes in monocyte and lymphocyte blood cell counts

Enformer pre-trained model



Enformer

Enformer model. It was introduced in the paper Effective gene expression prediction from sequence by integrating long-range interactions. by Avsec et al. and first released in this repository.

This particular model was trained on sequences of 131,072 basepairs, target length 896 on v3-64 TPUs for 2 and a half days without augmentations and poisson loss.

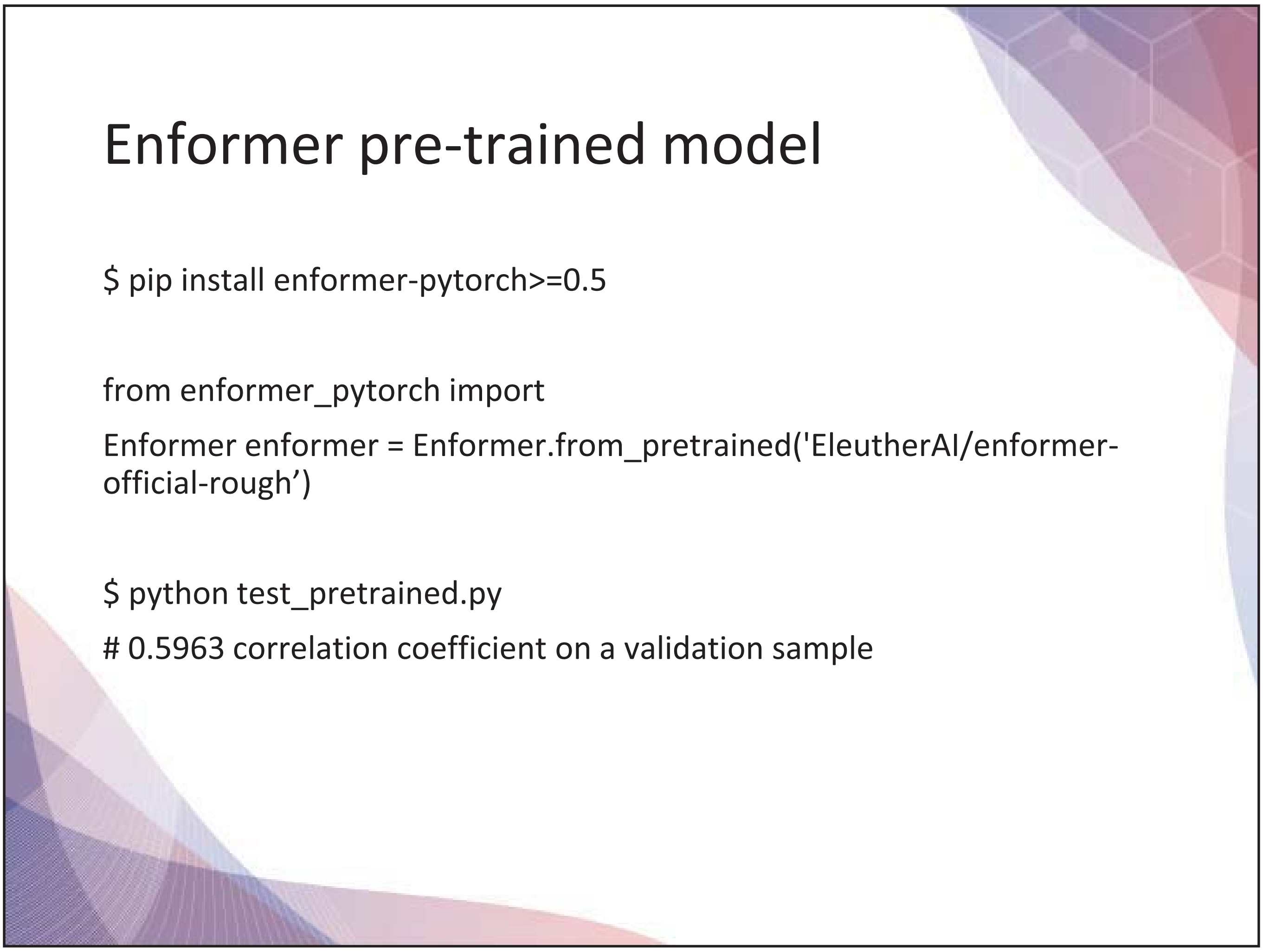
This repo contains the weights of the PyTorch implementation by Phil Wang as seen in the enformer-pytorch repository.

Disclaimer: The team releasing Enformer did not write a model card for this model so this model card has been written by the Hugging Face team.

Hosted inference API

Inference API has been turned off for this model.

- Pre-trained model https://huggingface.co/EleutherAI/enformer-preview
- Predict variant impact via Python

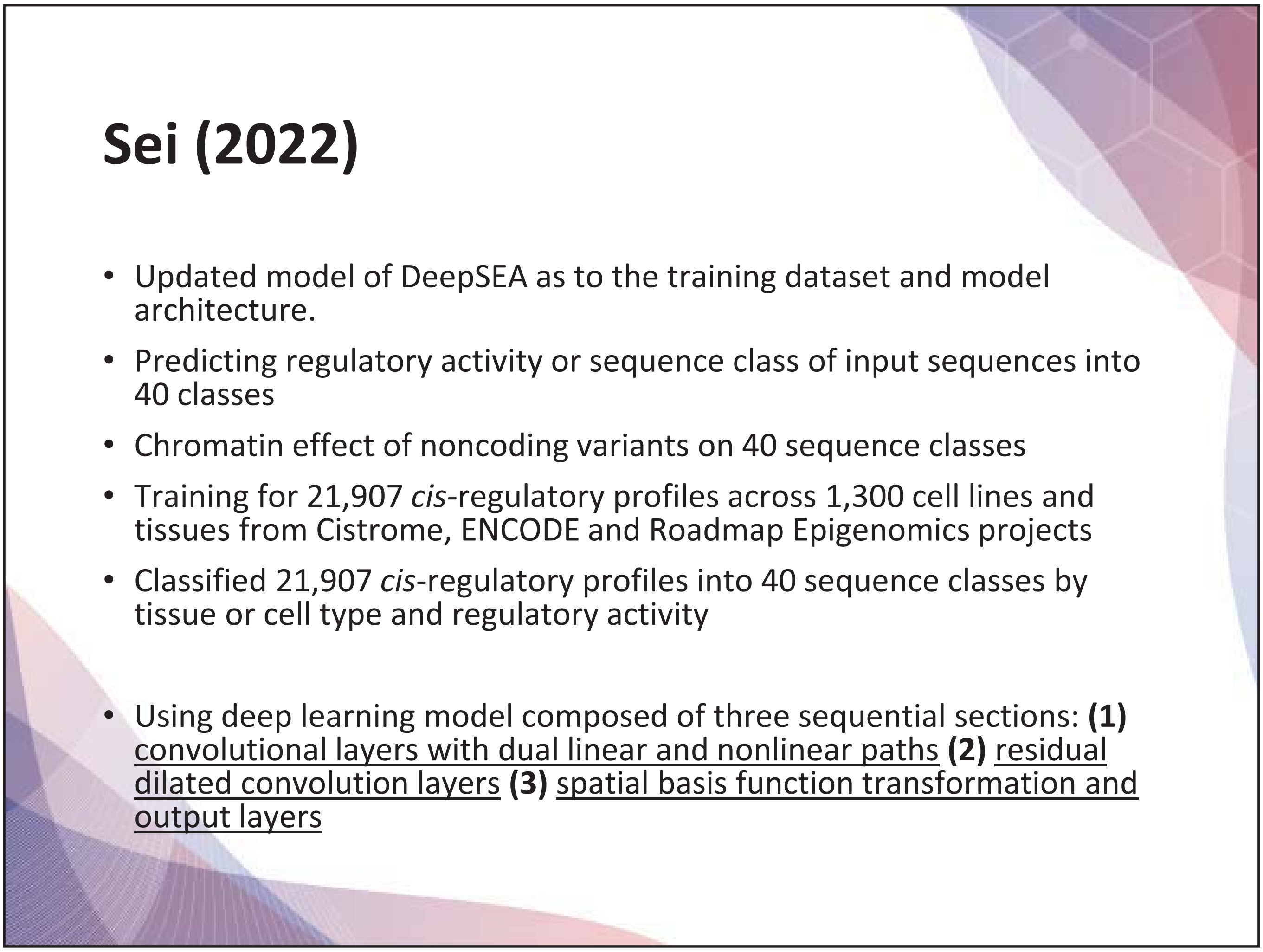


Enformer pre-trained model

```
$ pip install enformer-pytorch>=0.5
```

```
from enformer_pytorch import
Enformer enformer = Enformer.from_pretrained('EleutherAI/enformer-official-rough’)
```

```
$ python test_pretrained.py
# 0.5963 correlation coefficient on a validation sample
```

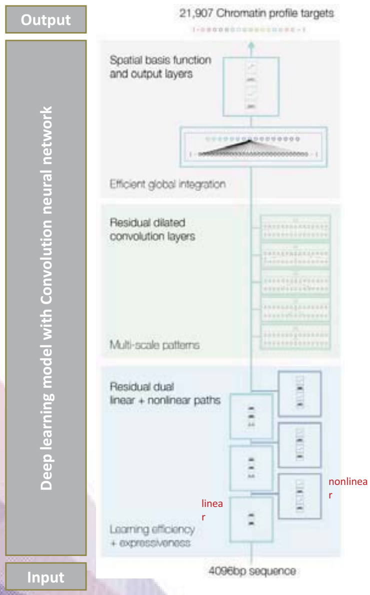


Sei (2022)

- Updated model of DeepSEA as to the training dataset and model architecture.
- Predicting regulatory activity or sequence class of input sequences into 40 classes
- Chromatin effect of noncoding variants on 40 sequence classes
- Training for 21,907 *cis*-regulatory profiles across 1,300 cell lines and tissues from Cistrome, ENCODE and Roadmap Epigenomics projects
- Classified 21,907 *cis*-regulatory profiles into 40 sequence classes by tissue or cell type and regulatory activity

- Using deep learning model composed of three sequential sections: **(1)** convolutional layers with dual linear and nonlinear paths **(2)** residual dilated convolution layers **(3)** spatial basis function transformation and output layers

Sei (2022)


Output
Deep learning model with Convolution neural network
Input
21,907 Chromatin profile targets
Spatial basis function and output layers
Efficient global integration
Residual dilated convolution layers
Multi-scale patterns
Residual dual linear + nonlinear paths
Learning efficiency + expressiveness
nonlinear
linear
4096bp sequence

Probability predictions of 21,907 chromatin features and classifying label of sequence into 40 classes

(3) Spatial basis function transformation and output layers

- Integrating information across spatial location with much higher efficient than fully connected layer
- Spatial basis enables reducing dimensionality while preserving sequence patterns

(2) Residual dilated convolution layers

- Dilated convolution further expand receptive fields (kernel sizes), added dilation rate that is interval between kernels
- Enabling the expansion of the receptive field without reducing spatial resolutions

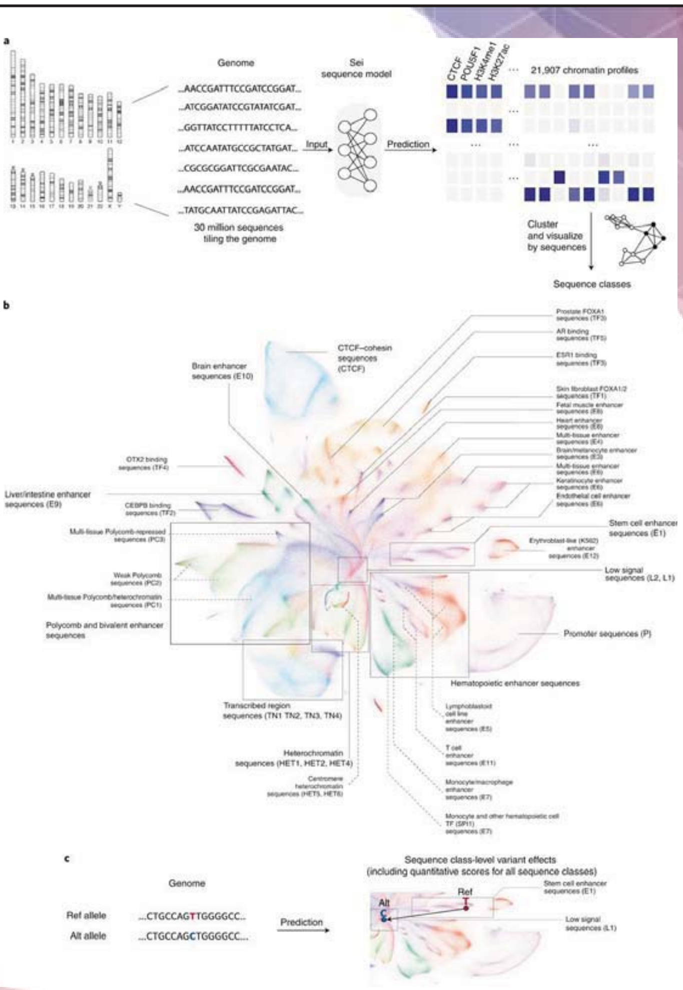
(1) Convolutional network with dual linear and nonlinear paths

- Linear convolution block: fast and statistically efficient training
- Nonlinear convolution block: strong representation power and the capability to learn complex interaction
- Nonlinear blocks are stacked on top of linear blocks with residual connection

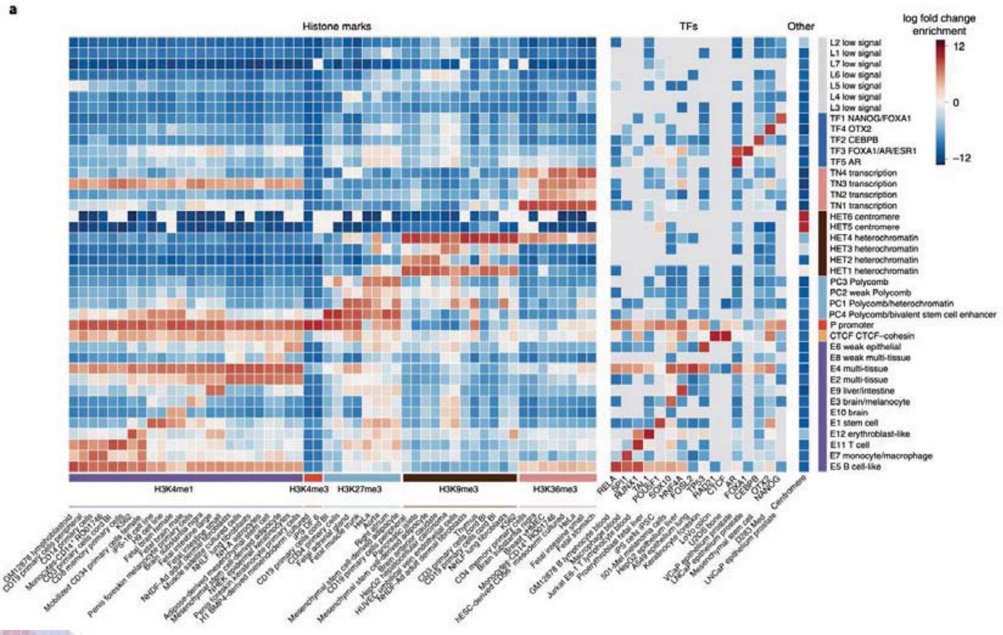
4-kb length sequence with/without variants, transformed to one-hot encoded matrix

Chen et al. 2022, Nature Genetics

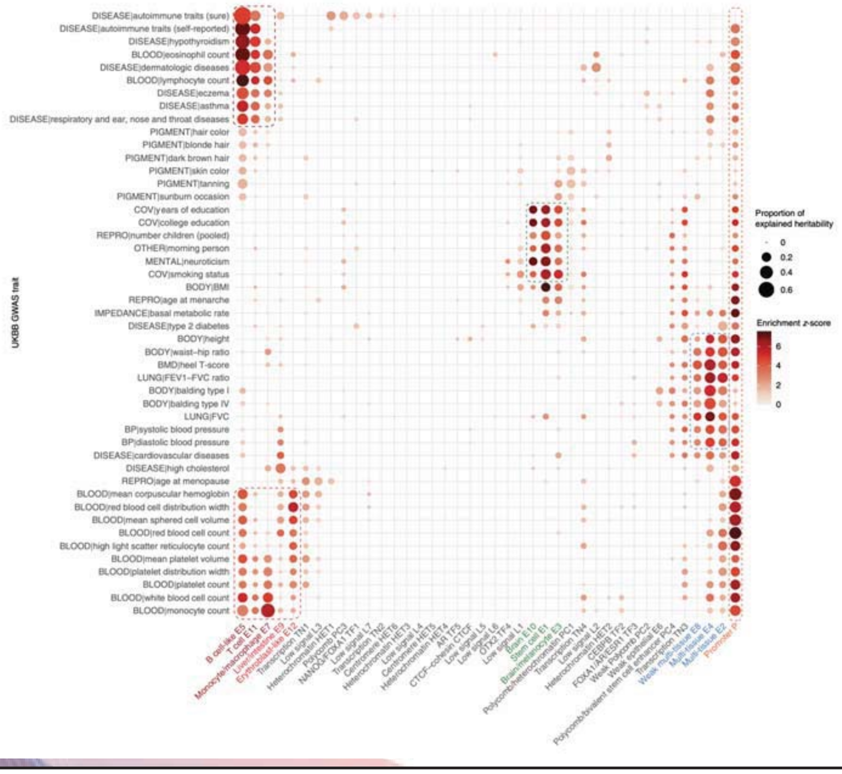
Sei (2022)

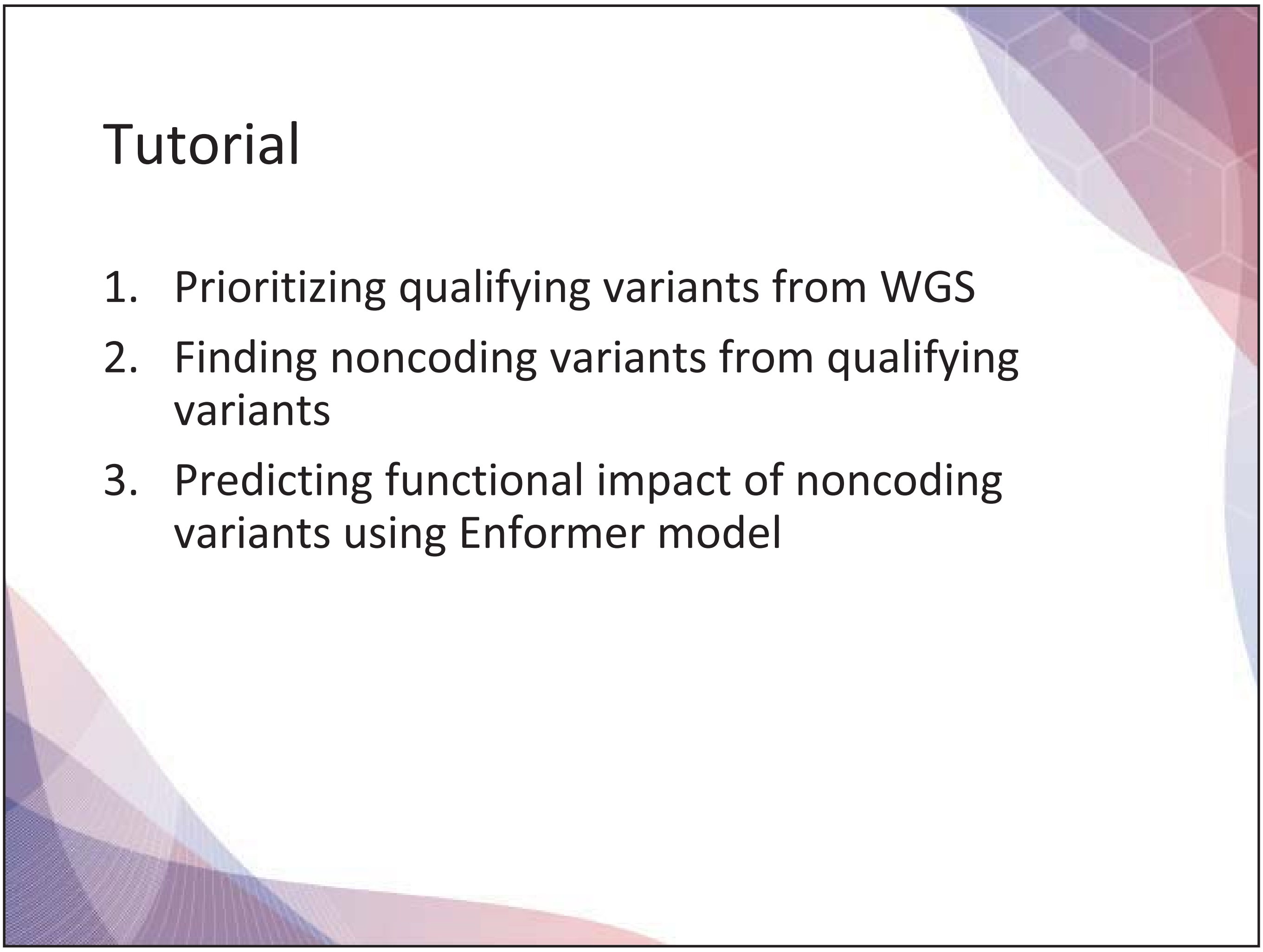


Sei (2022)



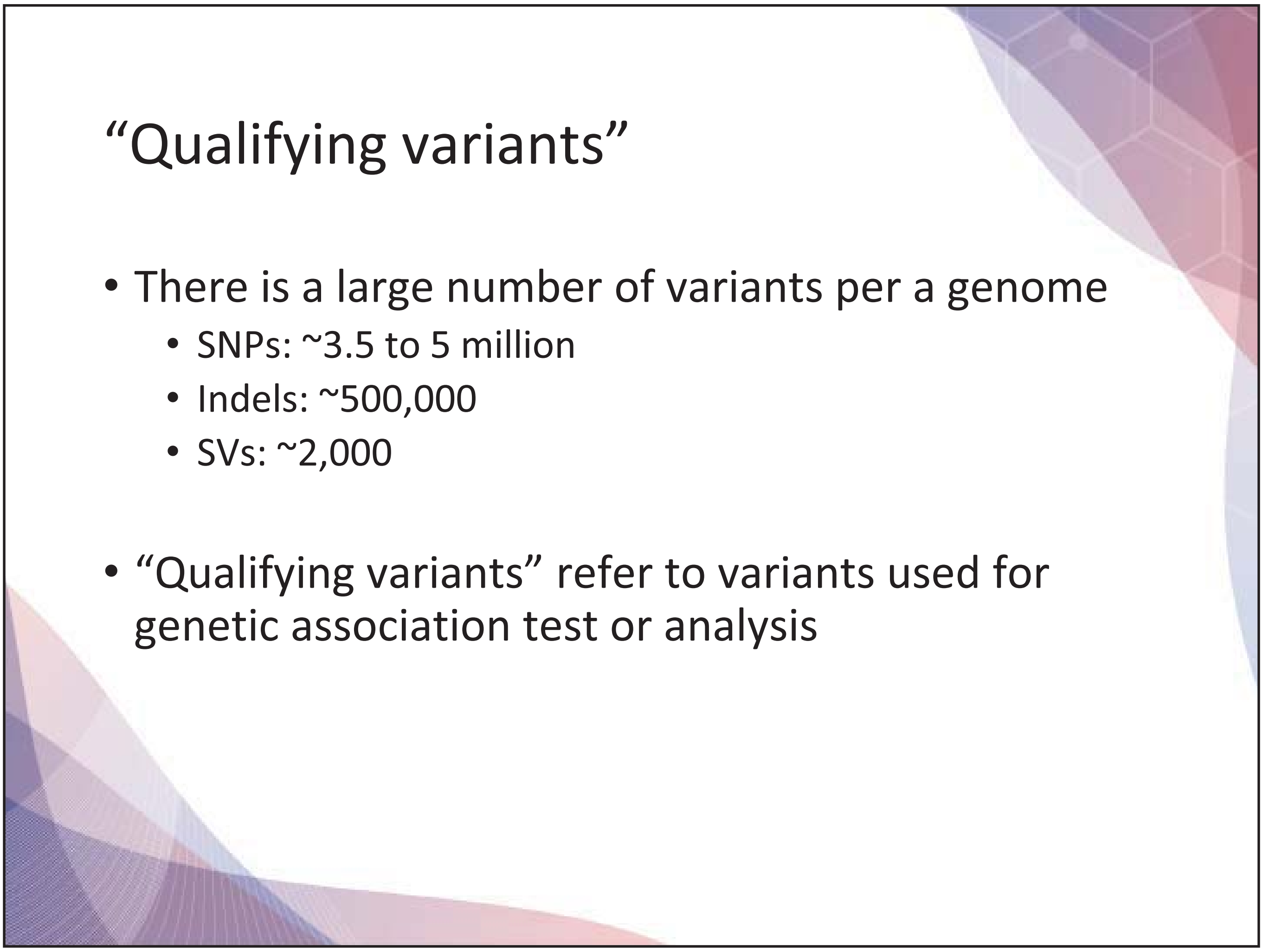
Sei (2022)





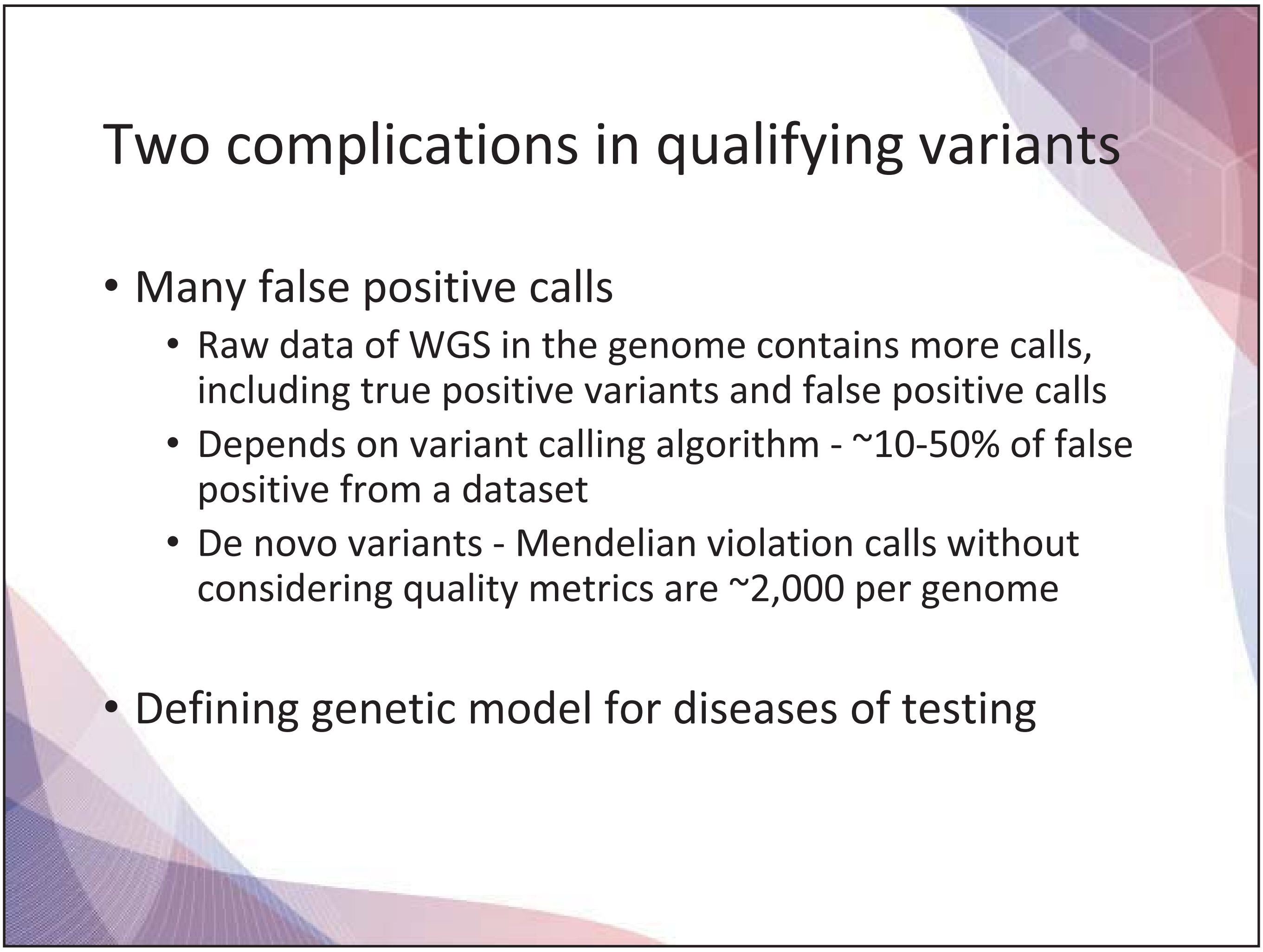
Tutorial

1. Prioritizing qualifying variants from WGS
2. Finding noncoding variants from qualifying variants
3. Predicting functional impact of noncoding variants using Enformer model



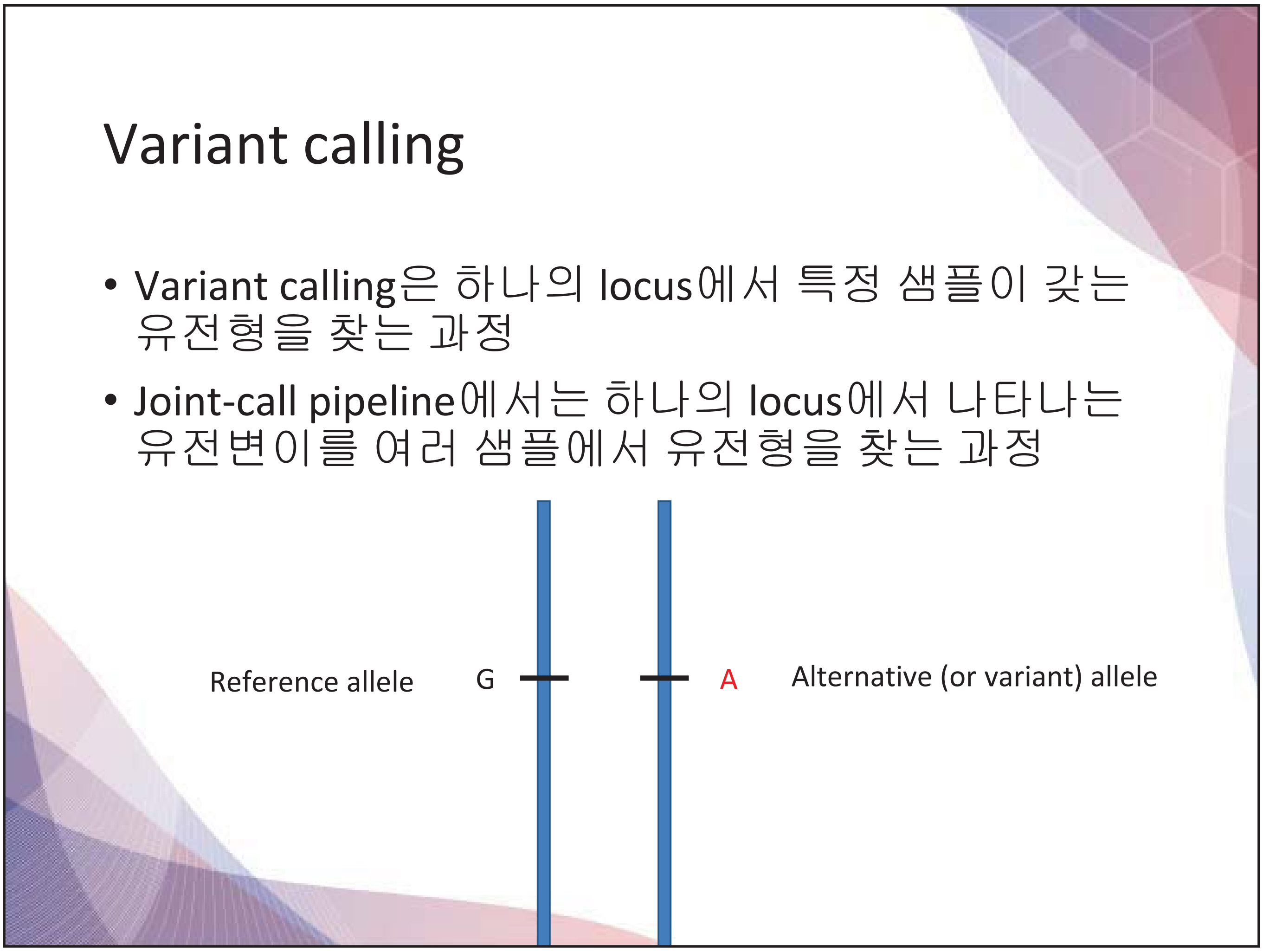
“Qualifying variants”

- There is a large number of variants per a genome
  - SNPs: ~3.5 to 5 million
  - Indels: ~500,000
  - SVs: ~2,000
- “Qualifying variants” refer to variants used for genetic association test or analysis



Two complications in qualifying variants

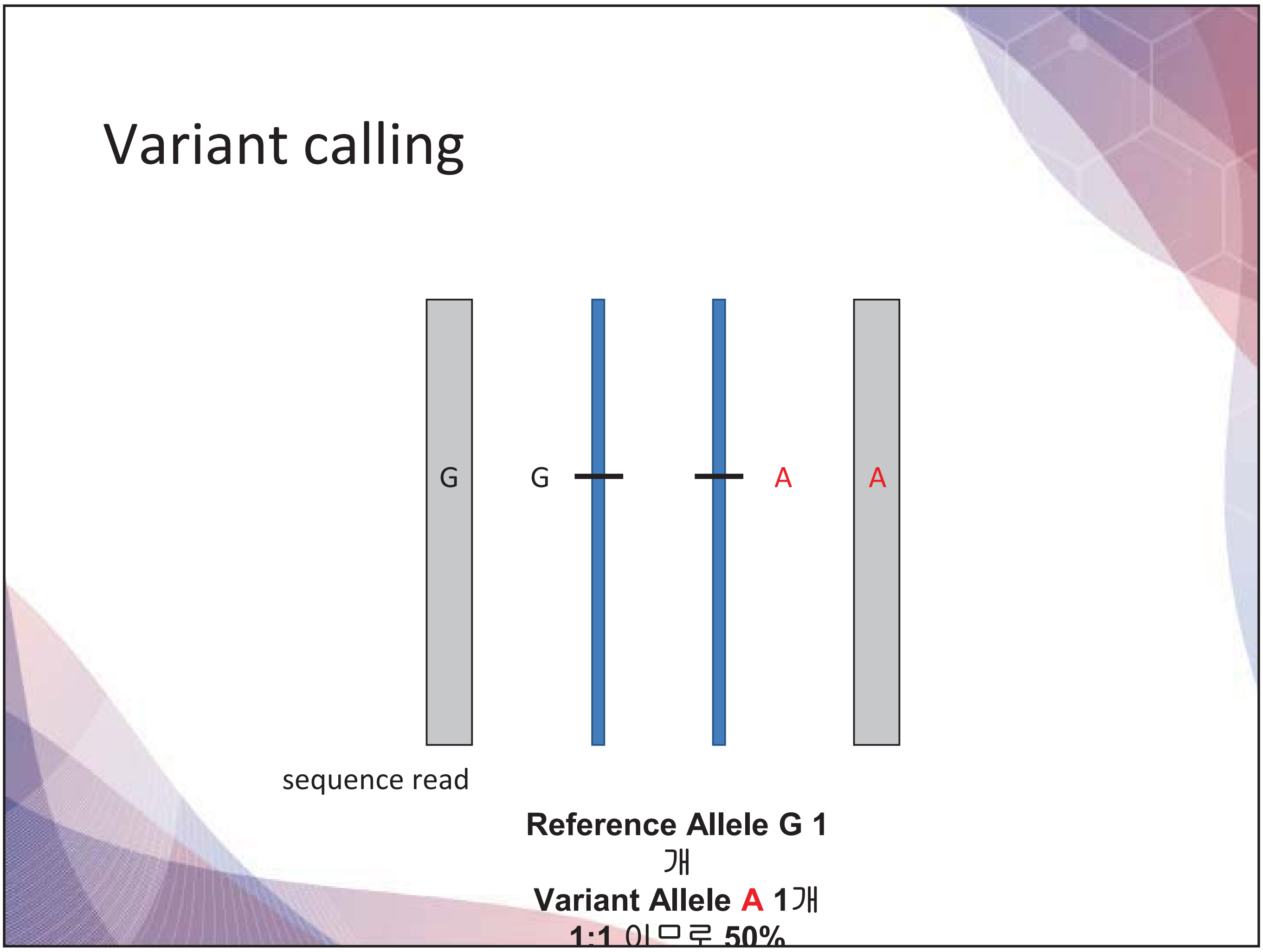
- Many false positive calls
  - Raw data of WGS in the genome contains more calls, including true positive variants and false positive calls
  - Depends on variant calling algorithm - ~10-50% of false positive from a dataset
  - De novo variants - Mendelian violation calls without considering quality metrics are ~2,000 per genome
- Defining genetic model for diseases of testing



Variant calling

- Variant calling은 하나의 locus에서 특정 샘플이 갖는 유전형을 찾는 과정
- Joint-call pipeline에서는 하나의 locus에서 나타나는 유전변이를 여러 샘플에서 유전형을 찾는 과정

Reference allele
G
A
Alternative (or variant) allele



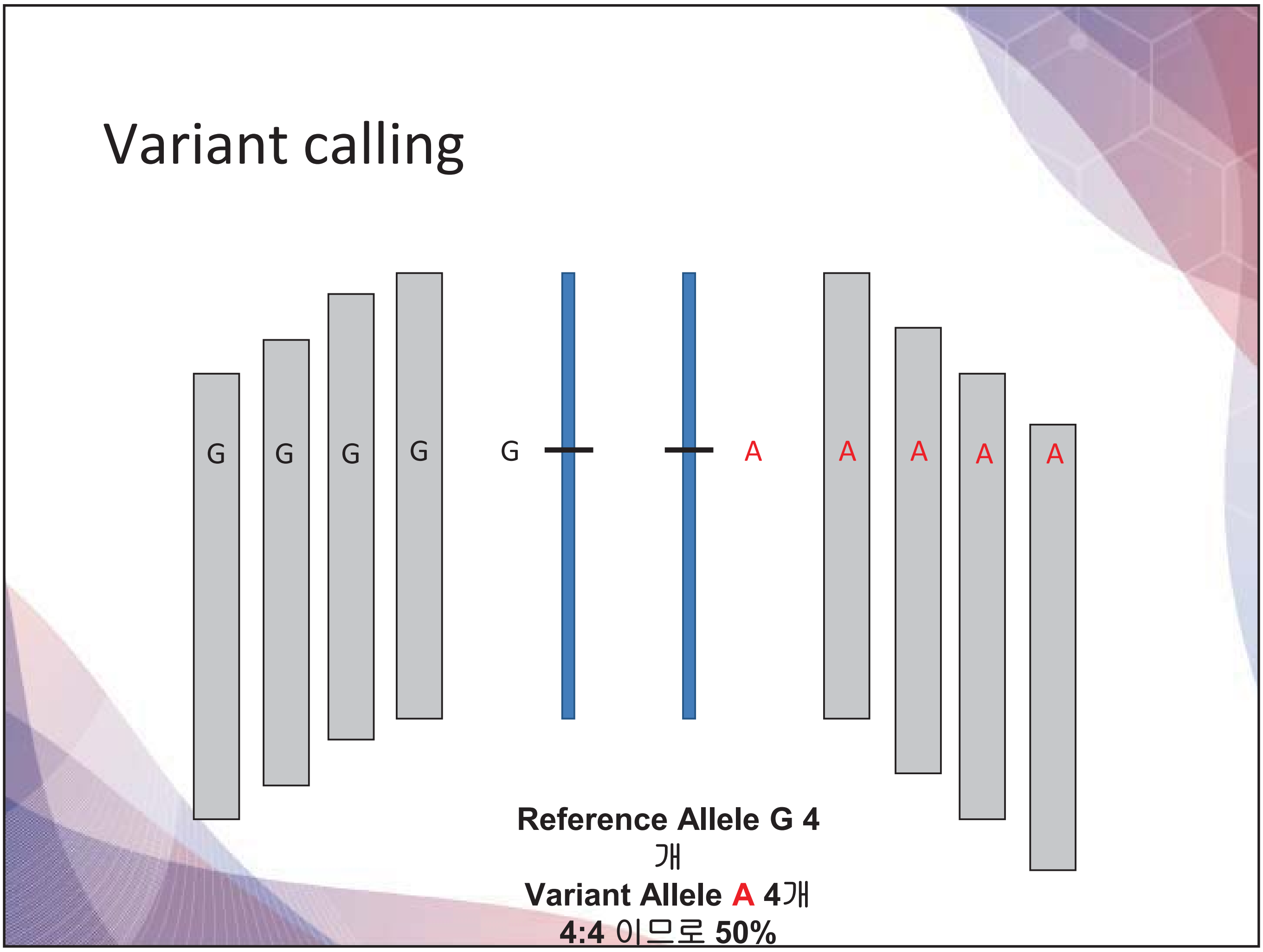
Variant calling

sequence read

Reference Allele G 1 개

Variant Allele A 1개

1:1 이므로 50%

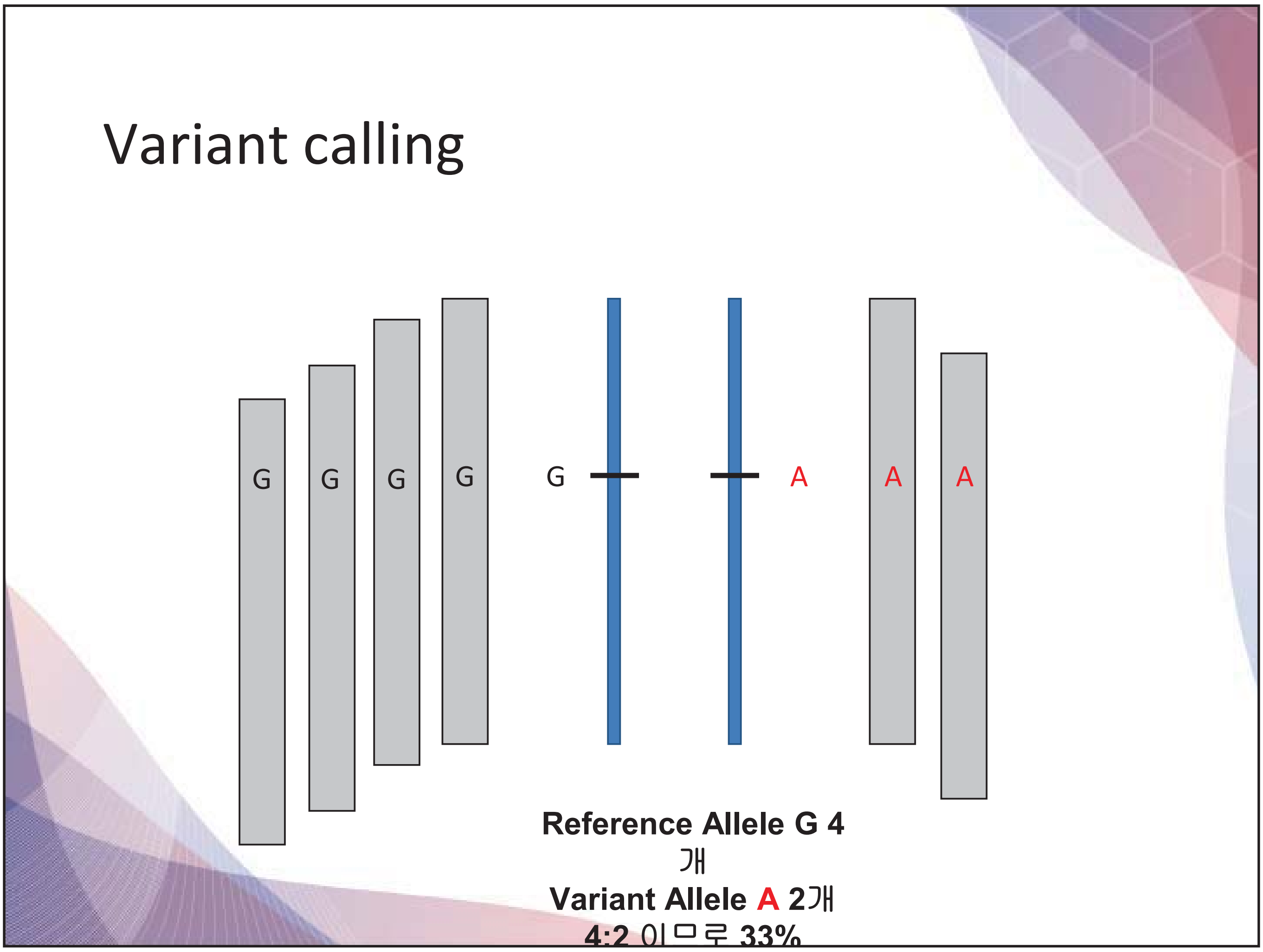


Variant calling

Reference Allele G 4 개

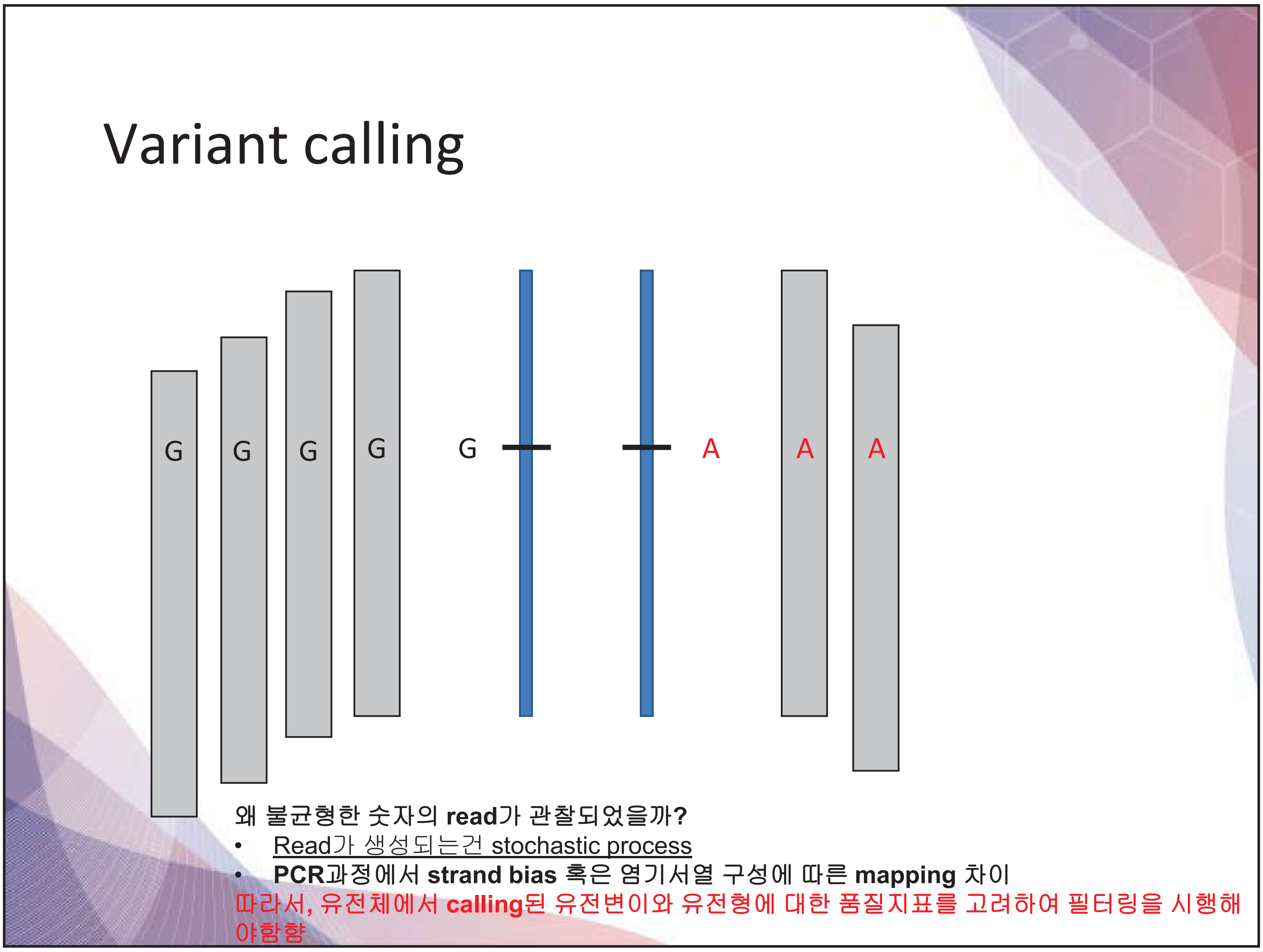
Variant Allele A 4개

4:4 이므로 50%



Variant calling

Reference Allele G 4
개
Variant Allele A 2개
4:2 이므로 **33%**

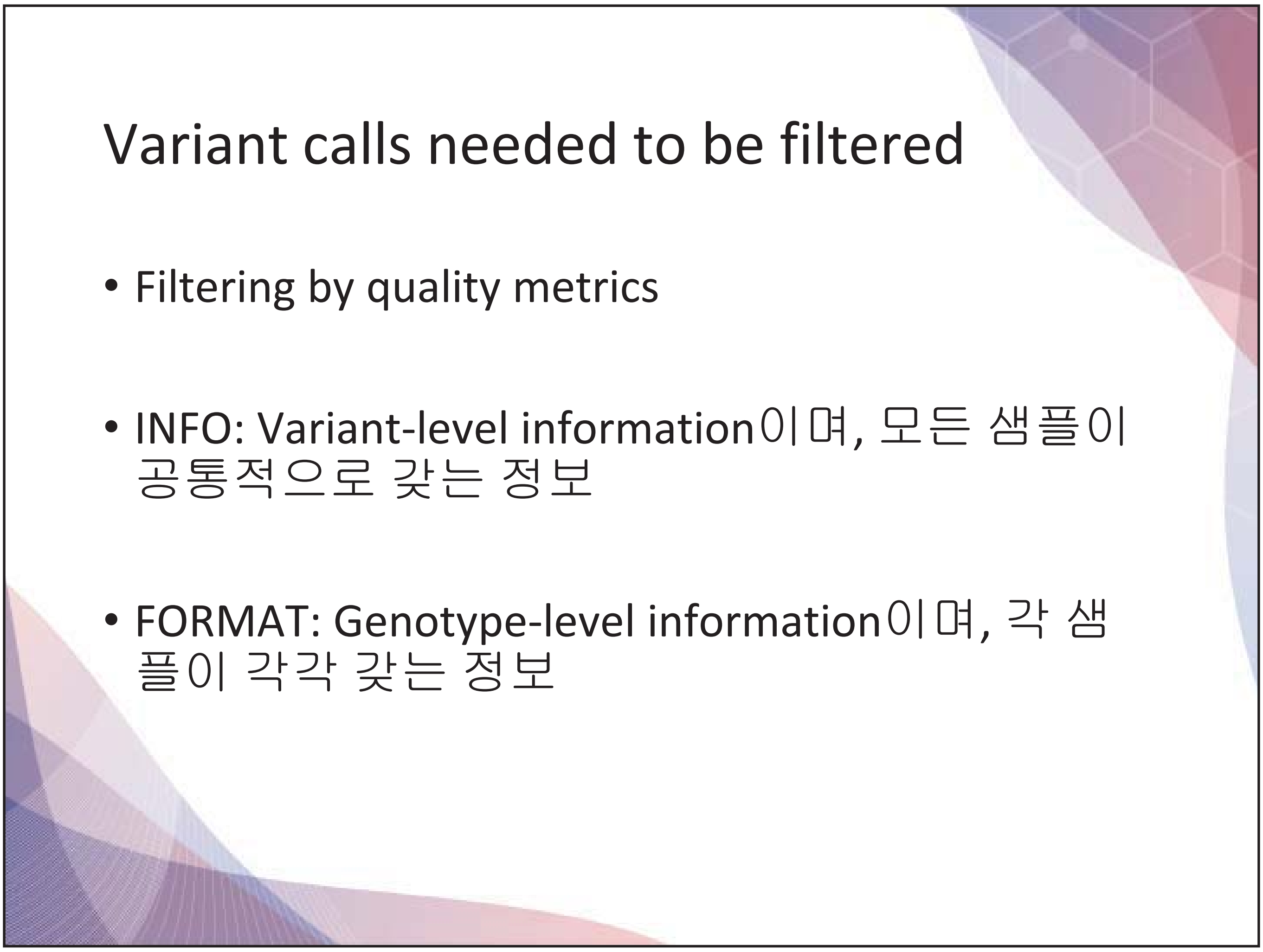


Variant calling

왜 불균형한 숫자의 **read**가 관찰되었을까**?**

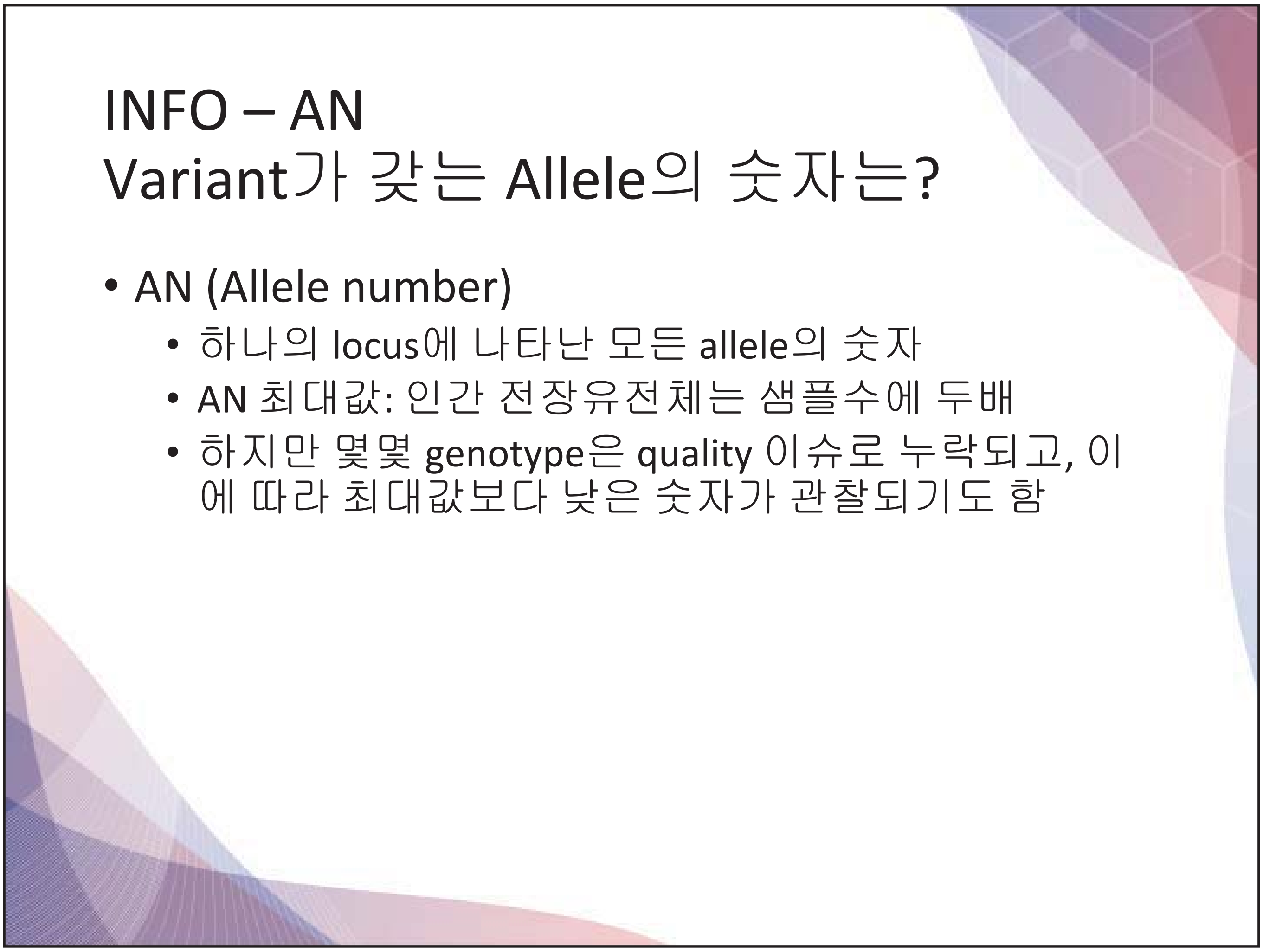
- Read가 생성되는건 stochastic process
- **PCR**과정에서 **strand bias** 혹은 염기서열 구성에 따른 **mapping** 차이

따라서**,** 유전체에서 **calling**된 유전변이와 유전형에 대한 품질지표를 고려하여 필터링을 시행해야함함



Variant calls needed to be filtered

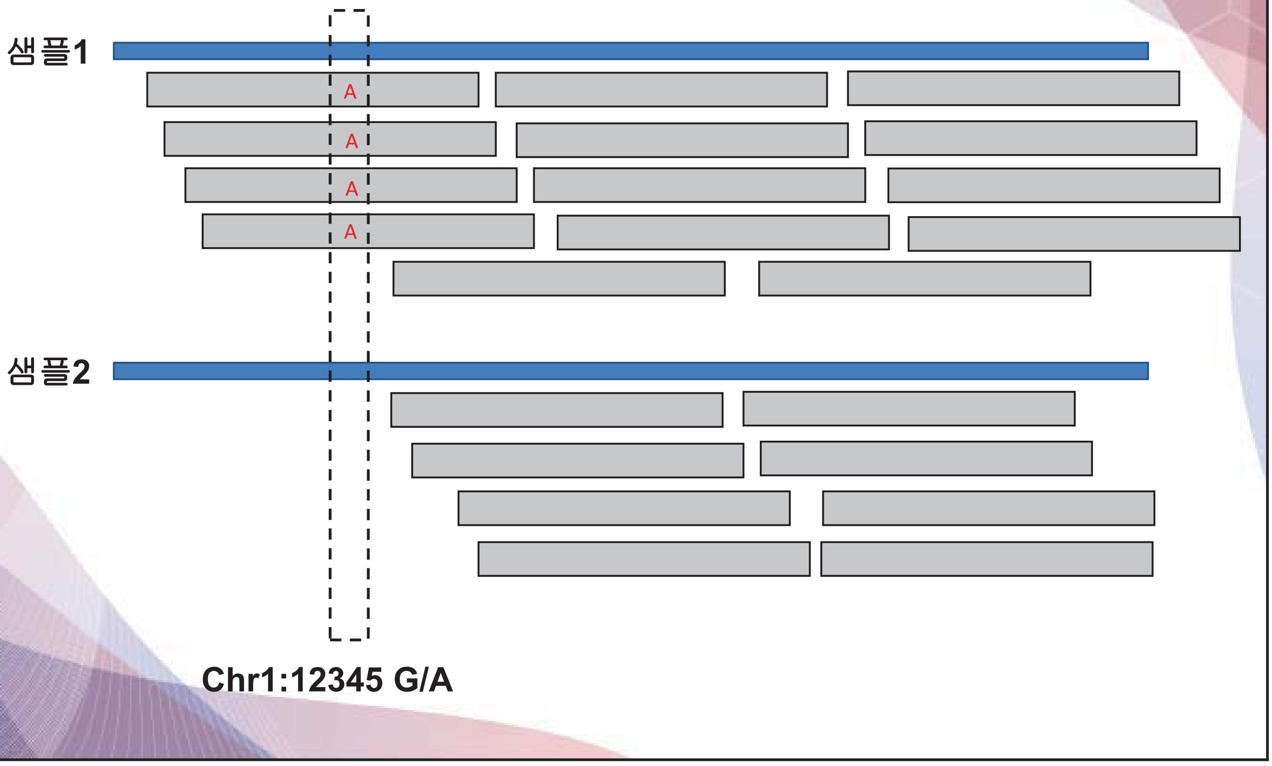
- Filtering by quality metrics
- INFO: Variant-level information이며, 모든 샘플이 공통적으로 갖는 정보
- FORMAT: Genotype-level information이며, 각 샘플이 각각 갖는 정보



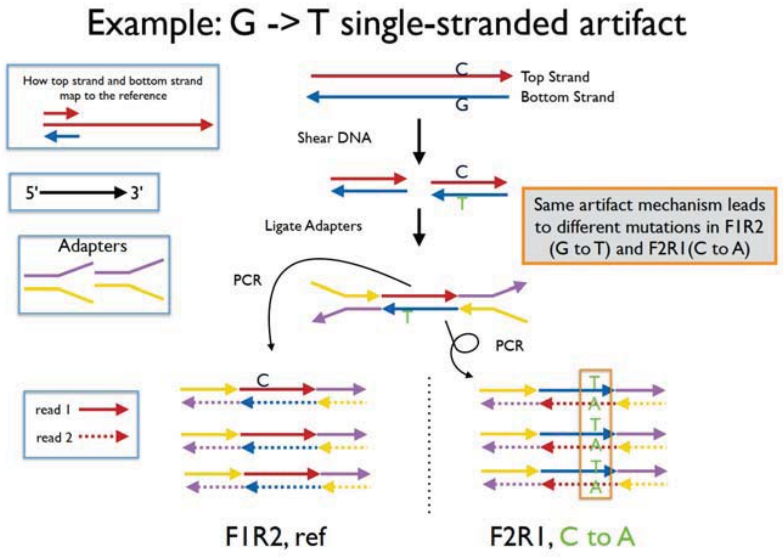
INFO – AN
Variant가 갖는 Allele의 숫자는?

- AN (Allele number)
  - 하나의 locus에 나타난 모든 allele의 숫자
  - AN 최대값: 인간 전장유전체는 샘플수에 두배
  - 하지만 몇몇 genotype은 quality 이슈로 누락되고, 이에 따라 최대값보다 낮은 숫자가 관찰되기도 함

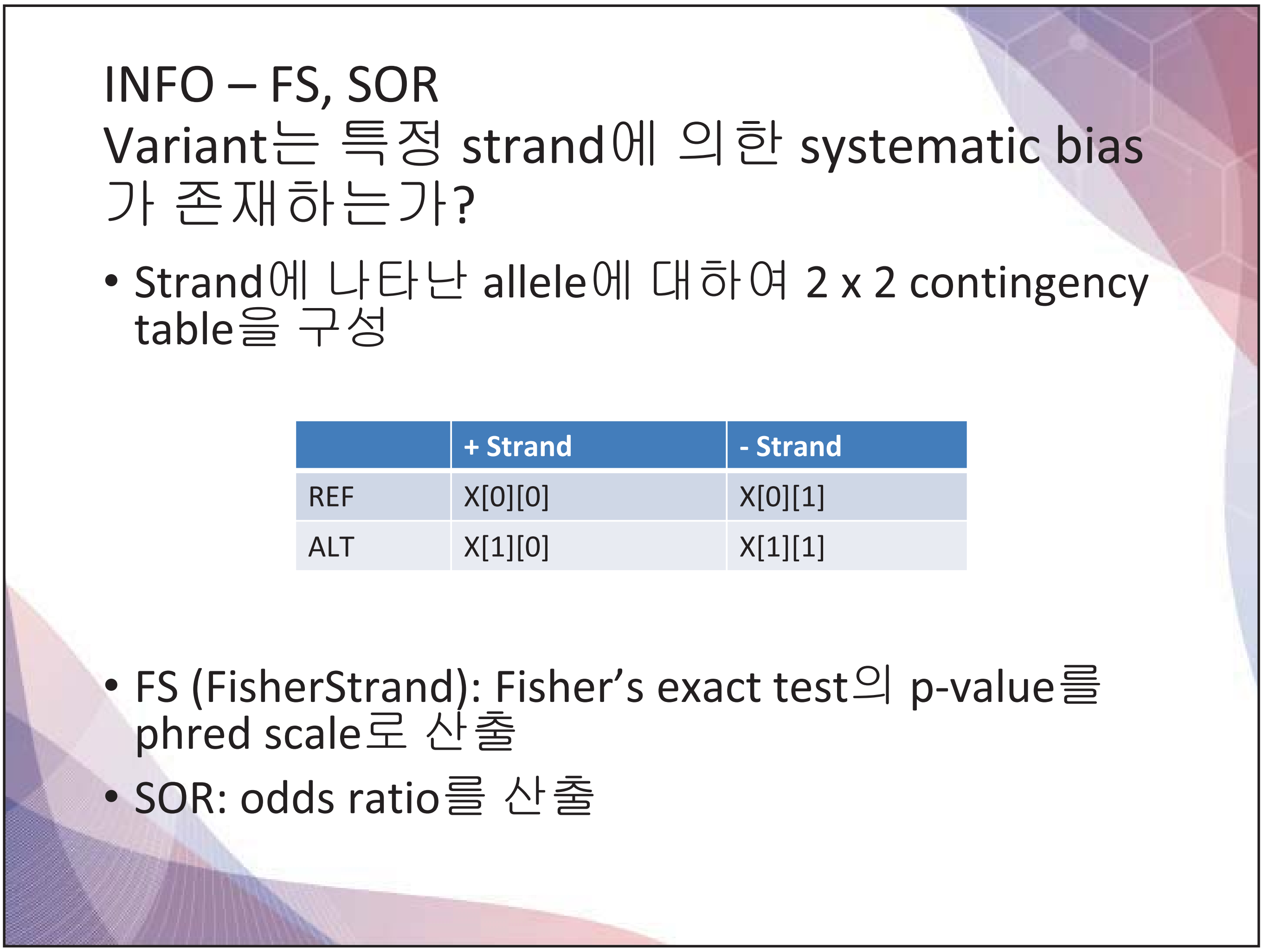
INFO – AC, AN
Variant가 갖는 Allele의 숫자는?



INFO – FS, SOR
Variant는 특정 strand에 의한 systematic bias 가 존재하는가?


Example: G -> T single-stranded artifact
How top strand and bottom strand map to the reference
5' 3'
Adapters
read 1
read 2
C
G
Top Strand
Bottom Strand
Shear DNA
C
T
Same artifact mechanism leads to different mutations in F1R2 (G to T) and F2R1(C to A)
Ligate Adapters
PCR
T
PCR
C
T A T A T A
F1R2, ref
F2R1, C to A

https://gatk.broadinstitute.org/hc/en-us/community/posts/360075017111-strand-bias-and-orientation-bias

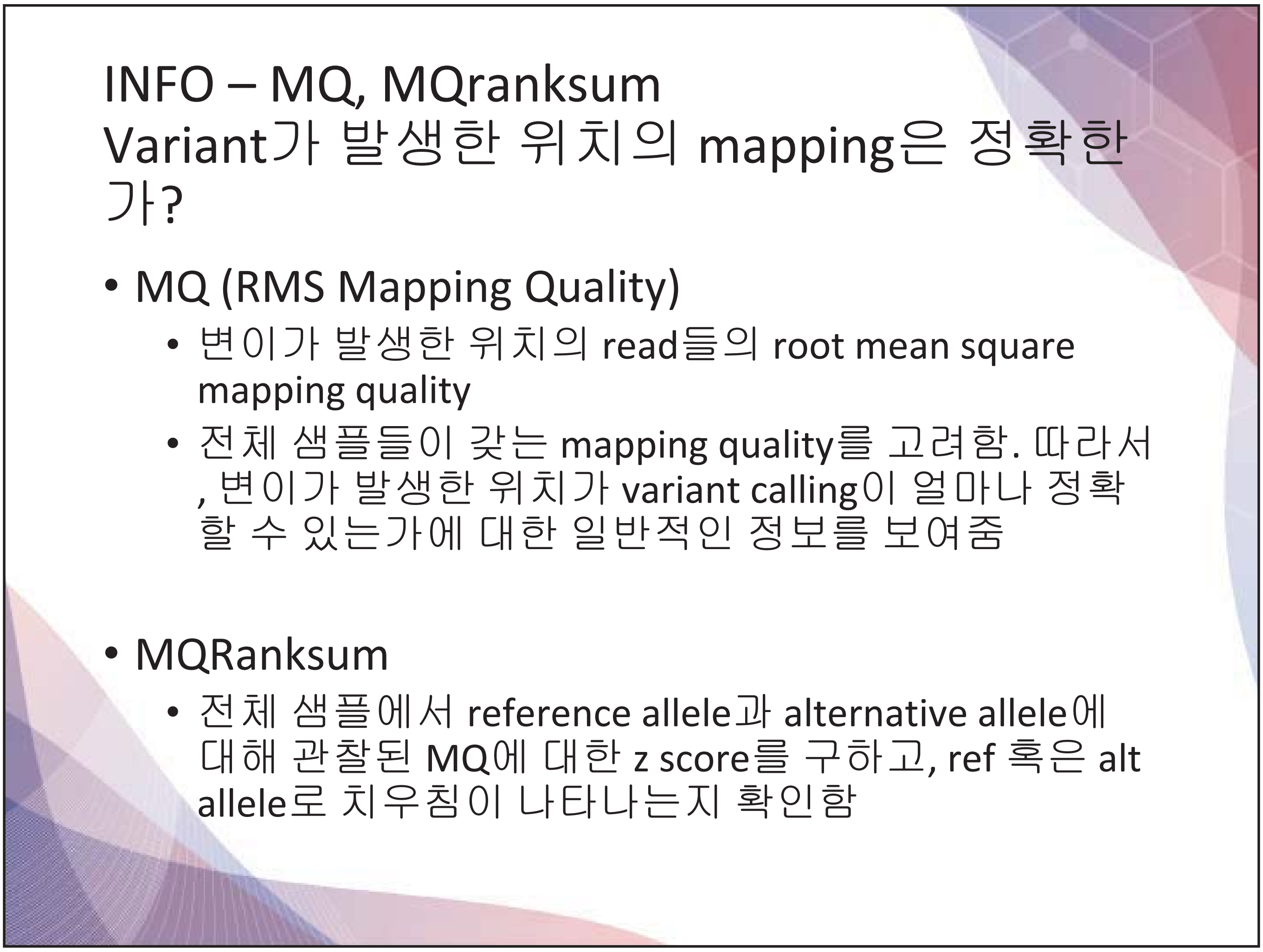


INFO – FS, SOR Variant는 특정 strand에 의한 systematic bias 가 존재하는가?

- Strand에 나타난 allele에 대하여 2 x 2 contingency table을 구성

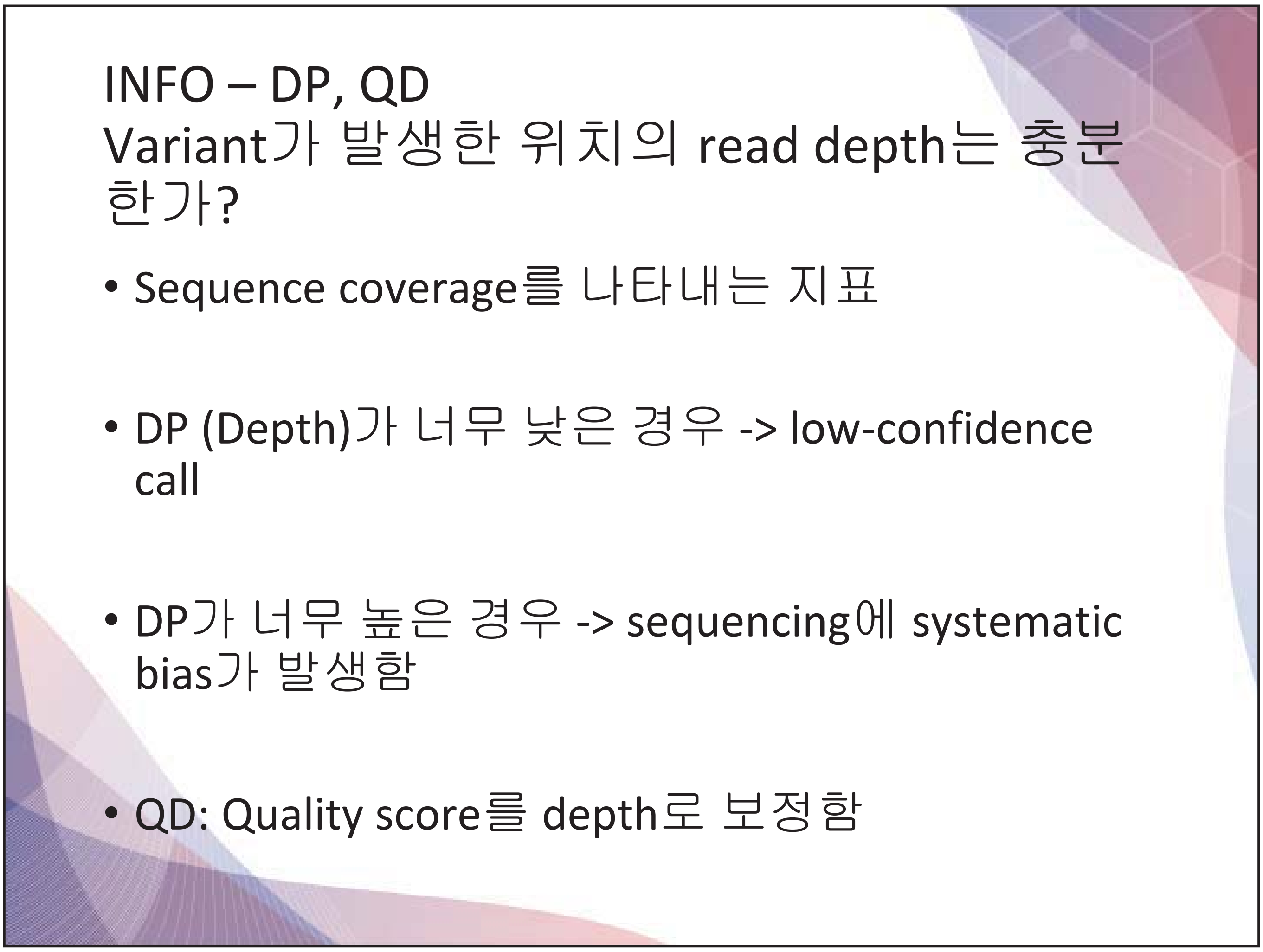
	+ Strand	- Strand
REF	X[0][0]	X[0][1]
ALT	X[1][0]	X[1][1]

- FS (FisherStrand): Fisher's exact test의 p-value를 phred scale로 산출
- SOR: odds ratio를 산출



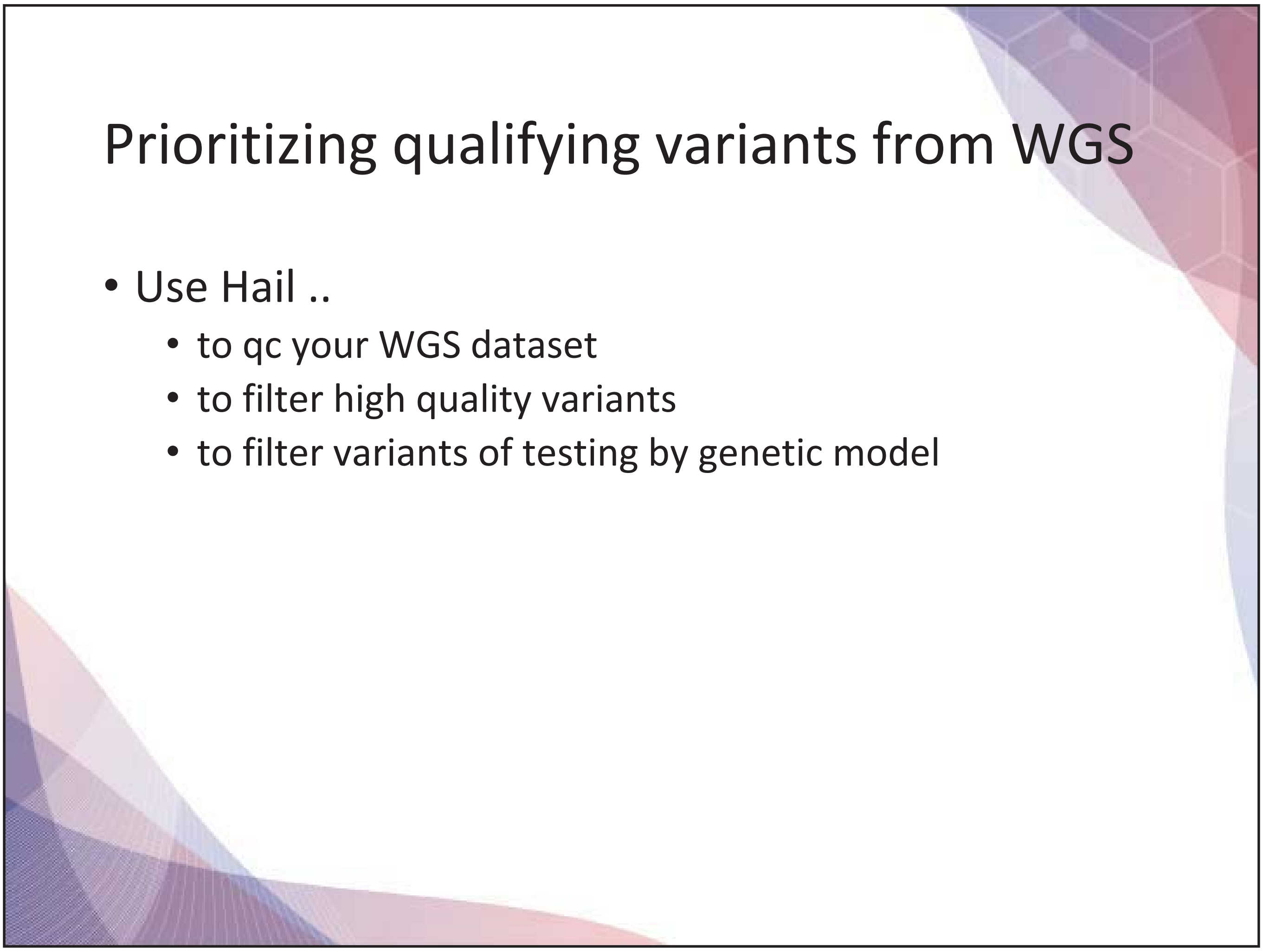
INFO – MQ, MQranksum Variant가 발생한 위치의 mapping은 정확한가?

- MQ (RMS Mapping Quality)
  - 변이가 발생한 위치의 read들의 root mean square mapping quality
  - 전체 샘플들이 갖는 mapping quality를 고려함. 따라서 , 변이가 발생한 위치가 variant calling이 얼마나 정확할 수 있는가에 대한 일반적인 정보를 보여줌
- MQRanksum
  - 전체 샘플에서 reference allele과 alternative allele에 대해 관찰된 MQ에 대한 z score를 구하고, ref 혹은 alt allele로 치우침이 나타나는지 확인함



INFO – DP, QD Variant가 발생한 위치의 read depth는 충분한가?

- Sequence coverage를 나타내는 지표
- DP (Depth)가 너무 낮은 경우 -> low-confidence call
- DP가 너무 높은 경우 -> sequencing에 systematic bias가 발생함
- QD: Quality score를 depth로 보정함



Prioritizing qualifying variants from WGS

- Use Hail ..
  - to qc your WGS dataset
  - to filter high quality variants
  - to filter variants of testing by genetic model

Introduction to HAIL

hail is an open-source Python library for genomic data manipulation and analysis.

- “Parallelization”, ”Scalable", "GWAS"
- Specifically applied on post-VCF analysis
- Used with large-scale datasets like gnomAD, UKBB, FINJEN, TOPMED and etc…

Computational Basis of Hail

1. Written in python, based on Apache Spark
2. Can run on Laptop/Desktop, Server, High performance cluster (HPC), Cloud (AWS, GCP…)
3. Similar to PySpark, but much more specific for genomic data
4. Implements a genomic dataframe ”MatrixTable”
5. As integrated with Spark, it can leverage SQL processing and machine learning algorithms

Apache Spark

What is Apache

1. Hail is an open-source big data framework which enbles the parallel processing

1. Written in Scala. can be accessed with Java, Python, Scala and R (API). SQL로도 데이터 처리 가능.

- Structured Query Language (SQL)은 database가 이해할 수 있는 언어로 원하는 데이터를 데이터베이스에 요청할 때 사용

3. Big data software 중 가장 활발히 사용되고 있음

Main Features of

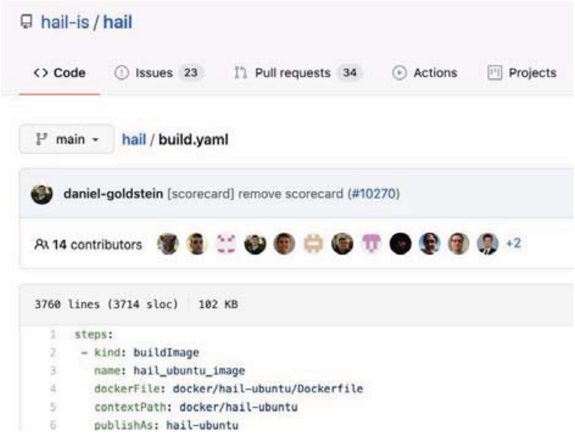
1. Spark RDD / Dataframe / Dataset

- Resilient distributed datasets (RDDs)는 추상적인 intermediate dataset으로 각종 API는 raw data가 아닌 RDD를 변환시켜 사용
- 그 외 dataframe과 dataset의 형태로도 사용 가능

2. Spark Streaming

- RDD 혹은 그 외 형태로 데이터를 실시간 스트리밍 처리
- 반복, 연속성을 바탕으로 변환 처리의 고속화 라는 강점을 가지고 있음

3. Spark Machine Learning



hail-is / hail

hail / build.yaml

daniel-goldstein [scorecard] remove scorecard (#10270)

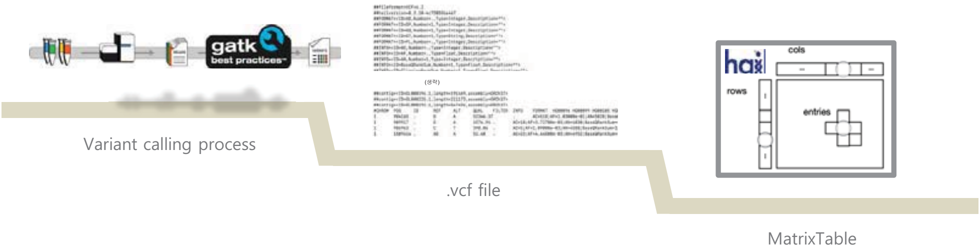
14 contributors +2

3760 lines (3714 sloc) 102 KB

```
steps:
 - kind: buildImage
   name: hail_ubuntu_image
   dockerFile: docker/hail-ubuntu/Dockerfile
   contextPath: docker/hail-ubuntu
   publishAs: hail-ubuntu
```

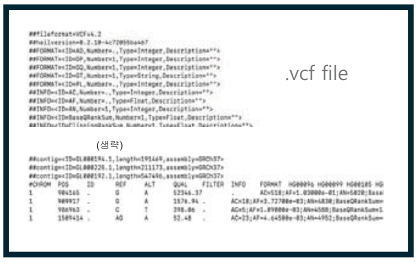
- Hail github에 올라와 있는 'build.yaml' 파일로 GCP에서 hail set up할 시 사용
- Configuration (환경 설정) 정보를 담고 있음
- AWS에서 hail을 설치할 때도 이와 같은 yaml 파일을 가지고 set up 진행

MatrixTable: Genomic Dataframe



다른 Data Science Library의 **DataFrame**은
Observation(행) x Variable(열) 2-dimension 형태로,
genomic data (VCF)를 담기 어렵다.

➡□ **Hail**에서는 `MatrixTable`이라는 **Genomic Dataframe class**를 고안함.

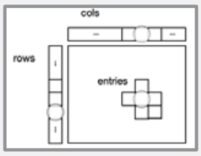


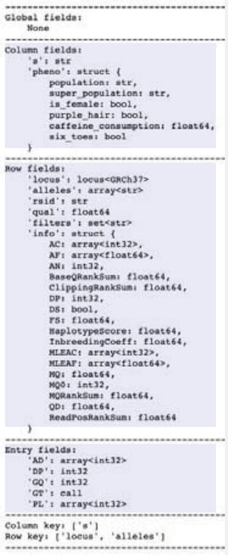
```
##fileformat=VCFv4.2
##hailversion=0.2.10-4c72055ba467
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="">
##FORMAT=<ID=GT,Number=1,Type=String,Description="">
##FORMAT=<ID=PL,Number=.,Type=Integer,Description="">
##INFO=<ID=AC,Number=.,Type=Integer,Description="">
##INFO=<ID=AF,Number=.,Type=Float,Description="">
##INFO=<ID=AN,Number=1,Type=Integer,Description="">
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="">
##INFO=<ID=ClippingRankSum,Number=1,Type=Float,Description="">

                (생략)

##contig=<ID=GL000194.1,length=191469,assembly=GRCh37>
##contig=<ID=GL000225.1,length=211173,assembly=GRCh37>
##contig=<ID=GL000192.1,length=547496,assembly=GRCh37>
#CHROM  POS     ID  REF ALT QUAL     FILTER INFO   FORMAT  HG00096 HG00099 HG00105 HG
1       904165  .   G   A   52346.37        .      AC=518;AF=1.03000e-01;AN=5020;Base
1       909917  .   G   A   1574.94  .      AC=18;AF=3.72700e-03;AN=4830;BaseQRankSum=
1       986963  .   C   T   398.86   .      AC=5;AF=1.09000e-03;AN=4588;BaseQRankSum=1
1       1509414 .   AG  A   52.48    .      AC=23;AF=4.64500e-03;AN=4952;BaseQRankSum=
```

.vcf file

MatrixTable은 3개의 Table(=DataFrame)이 연결된 구조
Row Table + Column Table + Entry Table

MatrixTable



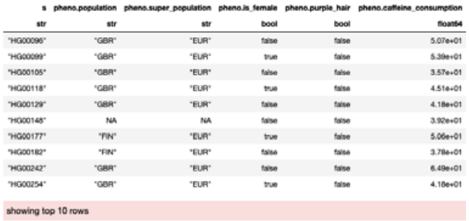


```
----------------------------------------
Global fields:
    None
----------------------------------------
Column fields:
    's': str
    'pheno': struct {
        population: str,
        super_population: str,
        is_female: bool,
        purple_hair: bool,
        caffeine_consumption: float64,
        six_toes: bool
    }
----------------------------------------
Row fields:
    'locus': locus<GRCh37>
    'alleles': array<str>
    'rsid': str
    'qual': float64
    'filters': set<str>
    'info': struct {
        AC: array<int32>,
        AF: array<float64>,
        AN: int32,
        BaseQRankSum: float64,
        ClippingRankSum: float64,
        DP: int32,
        DS: bool,
        FS: float64,
        HaplotypeScore: float64,
        InbreedingCoeff: float64,
        MLEAC: array<int32>,
        MLEAF: array<float64>,
        MQ: float64,
        MQ0: int32,
        MQRankSum: float64,
        QD: float64,
        ReadPosRankSum: float64
    }
----------------------------------------
Entry fields:
    'AD': array<int32>
    'DP': int32
    'GQ': int32
    'GT': call
    'PL': array<int32>
----------------------------------------
Column key: ['s']
Row key: ['locus', 'alleles']
----------------------------------------
```

1. Column table
(= Sample info
e.g. phenotype)

2. Row table
(= VCF INFO)

3. Entry table
(= VCF FORMAT)



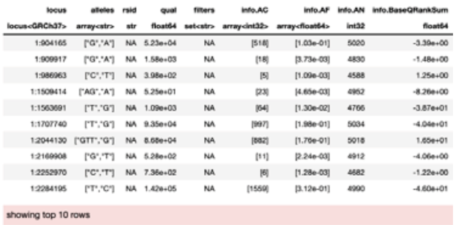
s	pheno.population	pheno.super_population	pheno.is_female	pheno.purple_hair	pheno.caffeine_consumption
str	str	str	bool	bool	float64
"HG00096"	"GBR"	"EUR"	false	false	5.07e+01
"HG00099"	"GBR"	"EUR"	true	false	5.39e+01
"HG00105"	"GBR"	"EUR"	false	false	3.57e+01
"HG00118"	"GBR"	"EUR"	true	false	4.51e+01
"HG00129"	"GBR"	"EUR"	false	false	4.18e+01
"HG00148"	NA	NA	false	false	3.92e+01
"HG00177"	"FIN"	"EUR"	true	false	5.06e+01
"HG00182"	"FIN"	"EUR"	false	false	3.78e+01
"HG00242"	"GBR"	"EUR"	false	false	6.49e+01
"HG00254"	"GBR"	"EUR"	true	false	4.18e+01

showing top 10 rows

Column table

Sample(행) x Sample Info(열)

The cols table stores column fields that have one value per column (sample), like the sample ID and phenotype information(e.g. population, sex, age).



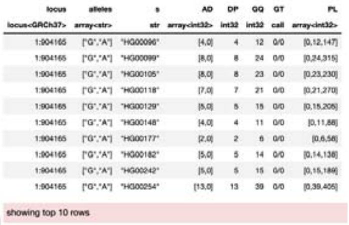
locus	alleles	rsid	qual	filters	info.AC	info.AF	info.AN	info.BaseQRankSum
locus<GRCh37>	array<str>	str	float64	set<str>	array<int32>	array<float64>	int32	float64
1:904165	["G","A"]	NA	5.23e+04	NA	[518]	[1.03e-01]	5020	-3.39e+00
1:909917	["G","A"]	NA	1.58e+03	NA	[18]	[3.73e-03]	4830	-1.48e+00
1:986963	["C","T"]	NA	3.98e+02	NA	[5]	[1.09e-03]	4588	1.25e+00
1:1509414	["AG","A"]	NA	5.25e+01	NA	[23]	[4.65e-03]	4952	-8.26e+00
1:1563691	["T","G"]	NA	1.09e+03	NA	[64]	[1.30e-02]	4766	-3.87e+01
1:1707740	["T","G"]	NA	9.35e+04	NA	[997]	[1.98e-01]	5034	-4.04e+01
1:2044130	["GTT","G"]	NA	8.68e+04	NA	[882]	[1.76e-01]	5018	1.65e+01
1:2169908	["G","T"]	NA	5.28e+02	NA	[11]	[2.24e-03]	4912	-4.06e+00
1:2252970	["C","T"]	NA	7.36e+02	NA	[6]	[1.28e-03]	4682	-1.22e+00
1:2284195	["T","C"]	NA	1.42e+05	NA	[1559]	[3.12e-01]	4990	-4.60e+01

showing top 10 rows

Row table

Variant(행) x Variant annotations(열)

The rows table stores row fields that have one value per row (variant) like locus, alleles, and variant annotations (e.g. AC, AF).



locus	alleles	s	AD	DP	GQ	GT	PL
locus<GRCh37>	array<str>	str	array<int32>	int32	int32	call	array<int32>
1:904165	["G","A"]	"HG00096"	[4,0]	4	12	0/0	[0,12,147]
1:904165	["G","A"]	"HG00099"	[8,0]	8	24	0/0	[0,24,315]
1:904165	["G","A"]	"HG00105"	[8,0]	8	23	0/0	[0,23,230]
1:904165	["G","A"]	"HG00118"	[7,0]	7	21	0/0	[0,21,270]
1:904165	["G","A"]	"HG00129"	[5,0]	5	15	0/0	[0,15,205]
1:904165	["G","A"]	"HG00148"	[4,0]	4	11	0/0	[0,11,88]
1:904165	["G","A"]	"HG00177"	[2,0]	2	6	0/0	[0,6,58]
1:904165	["G","A"]	"HG00182"	[5,0]	5	14	0/0	[0,14,138]
1:904165	["G","A"]	"HG00242"	[5,0]	5	15	0/0	[0,15,189]
1:904165	["G","A"]	"HG00254"	[13,0]	13	39	0/0	[0,39,405]

showing top 10 rows

Entry table

Variant, Sample(행) x Genotype fields(열)

The entries are a two-dimensional structured matrix that can contain genotype fields like GT, DP, GQ, AD, and PL.

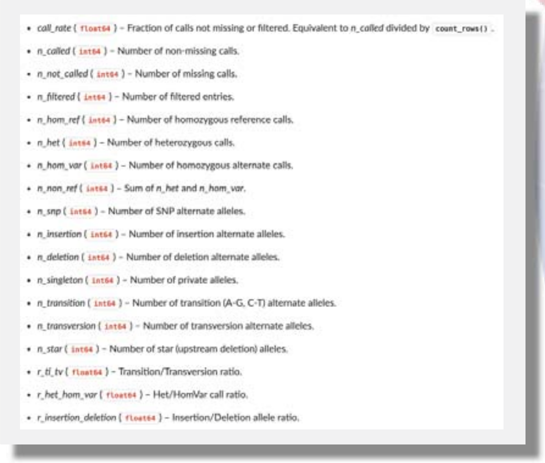
Quality Control
Variant level

```
>>> dataset = hl.sample_qc(dataset, name='sample_qc')
```

Computes summary statistics per sample from a genetic matrix and stores the results as a new column-indexed struct field

❶ From depth (DP) & genotype quality (GQ) -> stats

❷ From genotype (GT) -> following fields



- *call_rate* (`float64`) – Fraction of calls not missing or filtered. Equivalent to *n_called* divided by `count_rows()` .
- *n_called* (`int64`) – Number of non-missing calls.
- *n_not_called* (`int64`) – Number of missing calls.
- *n_filtered* (`int64`) – Number of filtered entries.
- *n_hom_ref* (`int64`) – Number of homozygous reference calls.
- *n_het* (`int64`) – Number of heterozygous calls.
- *n_hom_var* (`int64`) – Number of homozygous alternate calls.
- *n_non_ref* (`int64`) – Sum of *n_het* and *n_hom_var*.
- *n_snp* (`int64`) – Number of SNP alternate alleles.
- *n_insertion* (`int64`) – Number of insertion alternate alleles.
- *n_deletion* (`int64`) – Number of deletion alternate alleles.
- *n_singleton* (`int64`) – Number of private alleles.
- *n_transition* (`int64`) – Number of transition (A-G, C-T) alternate alleles.
- *n_transversion* (`int64`) – Number of transversion alternate alleles.
- *n_star* (`int64`) – Number of star (upstream deletion) alleles.
- *r_ti_tv* (`float64`) – Transition/Transversion ratio.
- *r_het_hom_var* (`float64`) – Het/HomVar call ratio.
- *r_insertion_deletion* (`float64`) – Insertion/Deletion allele ratio.

```
>>> filtered_dataset = dataset.filter_cols((dataset.sample_qc.dp_stats.mean > 20) &
                                           (dataset.sample_qc.r_ti_tv > 1.5))
```

Filter **samples** using generated sample QC statistics

Quality Control

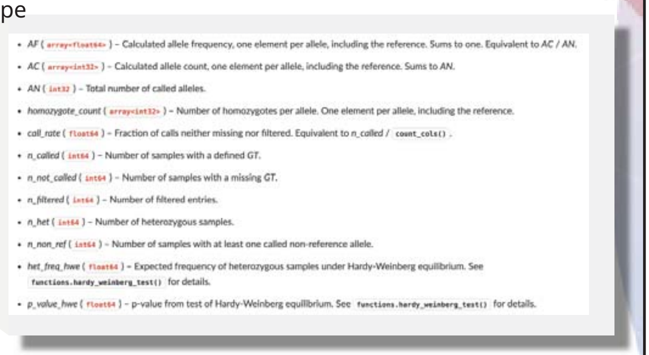
Variant level

```
>>> dataset_result = hl.variant_qc(dataset)
```

Computes summary statistics per variant from the genotype data and
stores the results as a new row-indexed struct field

1 From depth (DP) & genotype quality (GQ) -> stats

2 From genotype (GT) -> following fields



- *AF* (`array<float64>`) – Calculated allele frequency, one element per allele, including the reference. Sums to one. Equivalent to *AC* / *AN*.
- *AC* (`array<int32>`) – Calculated allele count, one element per allele, including the reference. Sums to *AN*.
- *AN* (`int32`) – Total number of called alleles.
- *homozygote_count* (`array<int32>`) – Number of homozygotes per allele. One element per allele, including the reference.
- *call_rate* (`float64`) – Fraction of calls neither missing nor filtered. Equivalent to *n_called* / `count_cols()` .
- *n_called* (`int64`) – Number of samples with a defined GT.
- *n_not_called* (`int64`) – Number of samples with a missing GT.
- *n_filtered* (`int64`) – Number of filtered entries.
- *n_het* (`int64`) – Number of heterozygous samples.
- *n_non_ref* (`int64`) – Number of samples with at least one called non-reference allele.
- *het_freq_hwe* (`float64`) – Expected frequency of heterozygous samples under Hardy-Weinberg equilibrium. See `functions.hardy_weinberg_test()` for details.
- *p_value_hwe* (`float64`) – p-value from test of Hardy-Weinberg equilibrium. See `functions.hardy_weinberg_test()` for details.

Filter **variants** using generated variant QC statistics

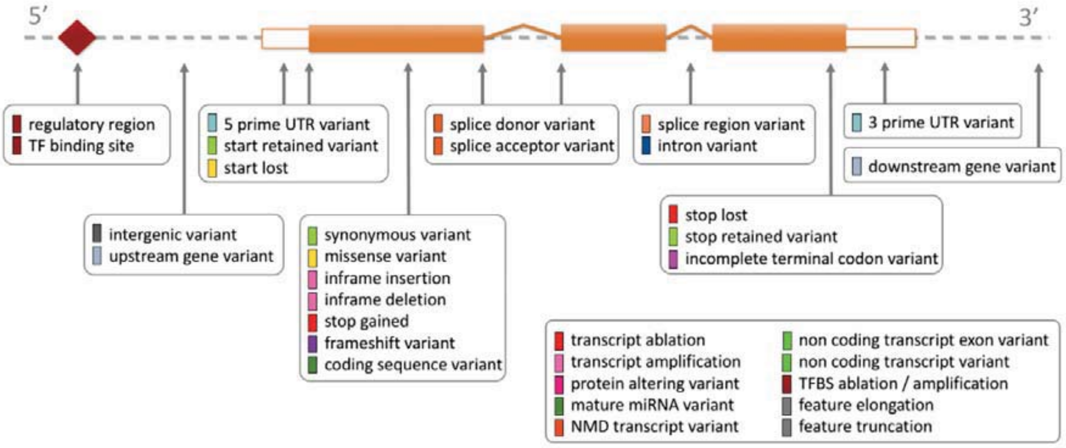
- GATK filter algorithm 통과
- Monoallelic variants
- Low complexity region 제외
- …

Finding noncoding variants from qualifying variants

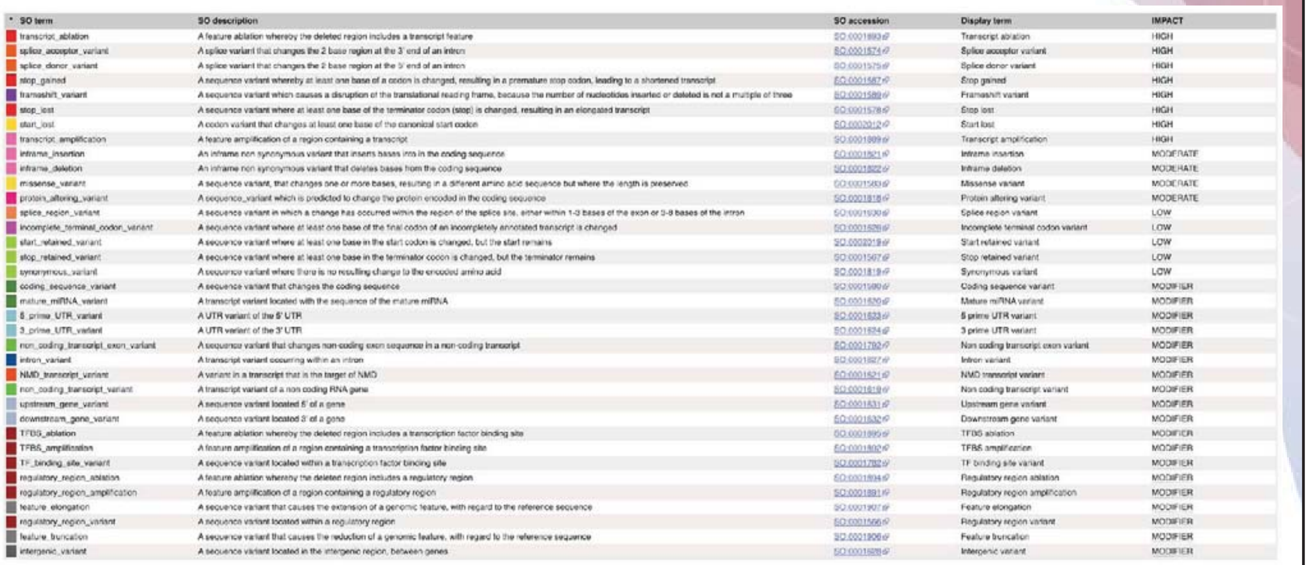
- Variant annotation: find variant information from related databases
  - Genomic positions
  - Genes, transcripts
  - Codon changes
  - Functional scores

- Various tools for variant annotations
  - VEP, ANNOVAR, SNPEFF

VEP: Calculated variant consequences

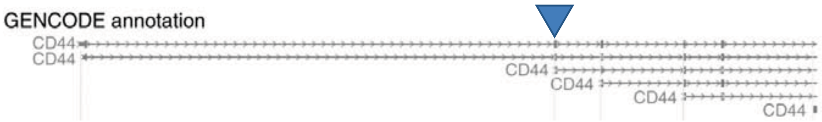


VEP: Calculated variant consequences



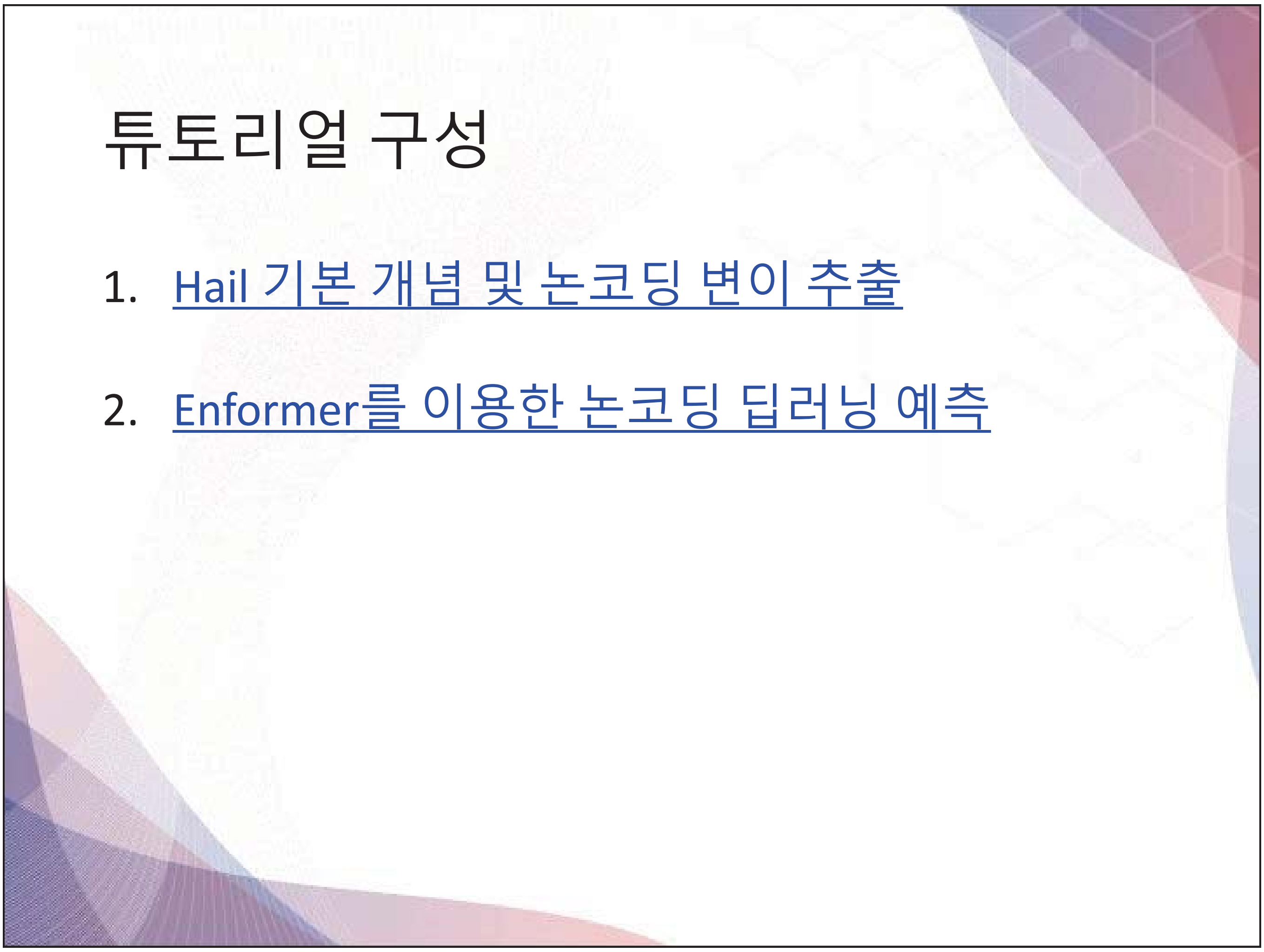
*	SO term	SO description	SO accession	Display term	IMPACT
🟥	transcript_ablation	A feature ablation whereby the deleted region includes a transcript feature	SO:0001893	Transcript ablation	HIGH
🟧	splice_acceptor_variant	A splice variant that changes the 2 base region at the 3' end of an intron	SO:0001574	Splice acceptor variant	HIGH
🟧	splice_donor_variant	A splice variant that changes the 2 base region at the 5' end of an intron	SO:0001575	Splice donor variant	HIGH
🟥	stop_gained	A sequence variant whereby at least one base of a codon is changed, resulting in a premature stop codon, leading to a shortened transcript	SO:0001587	Stop gained	HIGH
🟪	frameshift_variant	A sequence variant which causes a disruption of the translational reading frame, because the number of nucleotides inserted or deleted is not a multiple of three	SO:0001589	Frameshift variant	HIGH
🟥	stop_lost	A sequence variant where at least one base of the terminator codon (stop) is changed, resulting in an elongated transcript	SO:0001578	Stop lost	HIGH
🟨	start_lost	A codon variant that changes at least one base of the canonical start codon	SO:0002012	Start lost	HIGH
🟪	transcript_amplification	A feature amplification of a region containing a transcript	SO:0001889	Transcript amplification	HIGH
🟪	inframe_insertion	An inframe non synonymous variant that inserts bases into in the coding sequence	SO:0001821	Inframe insertion	MODERATE
🟪	inframe_deletion	An inframe non synonymous variant that deletes bases from the coding sequence	SO:0001822	Inframe deletion	MODERATE
🟨	missense_variant	A sequence variant, that changes one or more bases, resulting in a different amino acid sequence but where the length is preserved	SO:0001583	Missense variant	MODERATE
🟥	protein_altering_variant	A sequence_variant which is predicted to change the protein encoded in the coding sequence	SO:0001818	Protein altering variant	MODERATE
🟧	splice_region_variant	A sequence variant in which a change has occurred within the region of the splice site, either within 1-3 bases of the exon or 3-8 bases of the intron	SO:0001630	Splice region variant	LOW
🟪	incomplete_terminal_codon_variant	A sequence variant where at least one base of the final codon of an incompletely annotated transcript is changed	SO:0001626	Incomplete terminal codon variant	LOW
🟩	start_retained_variant	A sequence variant where at least one base in the start codon is changed, but the start remains	SO:0002019	Start retained variant	LOW
🟩	stop_retained_variant	A sequence variant where at least one base in the terminator codon is changed, but the terminator remains	SO:0001567	Stop retained variant	LOW
🟩	synonymous_variant	A sequence variant where there is no resulting change to the encoded amino acid	SO:0001819	Synonymous variant	LOW
🟩	coding_sequence_variant	A sequence variant that changes the coding sequence	SO:0001580	Coding sequence variant	MODIFIER
🟩	mature_miRNA_variant	A transcript variant located with the sequence of the mature miRNA	SO:0001620	Mature miRNA variant	MODIFIER
🟦	5_prime_UTR_variant	A UTR variant of the 5' UTR	SO:0001623	5 prime UTR variant	MODIFIER
🟦	3_prime_UTR_variant	A UTR variant of the 3' UTR	SO:0001624	3 prime UTR variant	MODIFIER
🟩	non_coding_transcript_exon_variant	A sequence variant that changes non-coding exon sequence in a non-coding transcript	SO:0001792	Non coding transcript exon variant	MODIFIER
🟦	intron_variant	A transcript variant occurring within an intron	SO:0001627	Intron variant	MODIFIER
🟥	NMD_transcript_variant	A variant in a transcript that is the target of NMD	SO:0001621	NMD transcript variant	MODIFIER
🟩	non_coding_transcript_variant	A transcript variant of a non coding RNA gene	SO:0001619	Non coding transcript variant	MODIFIER
🟦	upstream_gene_variant	A sequence variant located 5' of a gene	SO:0001631	Upstream gene variant	MODIFIER
🟦	downstream_gene_variant	A sequence variant located 3' of a gene	SO:0001632	Downstream gene variant	MODIFIER
🟥	TFBS_ablation	A feature ablation whereby the deleted region includes a transcription factor binding site	SO:0001895	TFBS ablation	MODIFIER
🟥	TFBS_amplification	A feature amplification of a region containing a transcription factor binding site	SO:0001892	TFBS amplification	MODIFIER
🟥	TF_binding_site_variant	A sequence variant located within a transcription factor binding site	SO:0001782	TF binding site variant	MODIFIER
🟥	regulatory_region_ablation	A feature ablation whereby the deleted region includes a regulatory region	SO:0001894	Regulatory region ablation	MODIFIER
🟥	regulatory_region_amplification	A feature amplification of a region containing a regulatory region	SO:0001891	Regulatory region amplification	MODIFIER
⬛	feature_elongation	A sequence variant that causes the extension of a genomic feature, with regard to the reference sequence	SO:0001907	Feature elongation	MODIFIER
🟥	regulatory_region_variant	A sequence variant located within a regulatory region	SO:0001566	Regulatory region variant	MODIFIER
⬛	feature_truncation	A sequence variant that causes the reduction of a genomic feature, with regard to the reference sequence	SO:0001906	Feature truncation	MODIFIER
⬛	intergenic_variant	A sequence variant located in the intergenic region, between genes	SO:0001628	Intergenic variant	MODIFIER

VEP: Calculated variant consequences



- Priority between coding and noncoding regions
- Variant A is predicted as coding variant on transcript A but is predicted as noncoding variant on transcript B
- Coding variants are likely to have strong impact than noncoding variants

논코딩 딥러닝 분석 코드 튜토리얼



튜토리얼 구성

1. Hail 기본 개념 및 논코딩 변이 추출
2. Enformer를 이용한 논코딩 딥러닝 예측