KSBi-BIML 2026

Bioinformatics & Machine Learning(BIML) Workshop for Life Scientists

생명정보학 & 머신러닝 워크샵 (온라인)

DNA Foundation Model 이론 및 실습

안준용 _ 고려대학교

KSBI KOREAN SOCIETY FOR BIOINFORMATICS | 한국생명정보학회

본 강의 자료는 한국생명정보학회가 주관하는 BIML 2026 워크샵을 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 행위자 본인에게 있음**을 알립니다.

KSBi-BIML 2026

Bioinformatics & Machine Learning (BIML) Workshop for Life Scientists

한국생명정보학회가 주최하는 BIML-2026 동계 Bioinformatics &Machine Learning 교육 워크숍에 여러분을 초대합니다.

BIML 워크숍은 생명정보학 연구자들이 최신 AI바이오 분야의 인공지능 기반 분석 기술과 바이오 데이터 분석 기법을 이론과 실습을 통해 체계적으로 배울 수 있는 전문 교육 프로그램입니다. 2015년에 시작된 BIML 워크숍은 올해로 12년 차를 맞이하며, 국내 생명정보학 분야의 최초이자 최고 수준의 교육 프로그램으로 자리 잡았습니다. 이번 워크숍은 크게 인공지능바이오(AI바이오) 분야와 디지털바이오 분야, 두 분야로 구성됩니다.

AI바이오 분야에서는 생명정보 분석에 폭넓게 응용되고 있는 다양한 인공지능 기반 자료 모델링 기법을 다룰 예정입니다. 특히, 인공지능 심층학습을 활용한 단백질 구조 예측, 유전체 분석, 신약 개발에 대한 이론 및 실습 강의가 진행됩니다.

또한 디지털바이오 분야에서는 단일세포오믹스, 공간오믹스, 멀티오믹스, 메타오믹스에 대한 강의도 마련되어 있어, 연구자들의 분석 역량 강화에 실질적인 도움을 줄 것으로 기대됩니다.

또한 2024년부터 추가된 의료정보 자료 분석을 다루는 강의를 올해도 지속해서 운영하고자 합니다. 이는 최근 의료정보 자료 분석에 관한 연구 수요 증가를 반영한 것으로, 관련 연구를 수행하는 의과학자 및 의료정보 연구자들에게 유용한 지침을 제공할 것입니다.

또한, 올해도 생명정보학 기술의 다양화에 발맞춰 온라인 강좌를 대폭 확대했습니다. 올해는 무료 강좌 10개를 포함한 총 40개 이상의 강좌가 개설되며, 연구 주제에 맞는 강좌 추천과 강연료 할인 혜택도 제공합니다.

BIML-2026는 국내 주요 연구 중심 대학의 전임 교수 및 각 분야 최고 전문가들의 강의로 구성되어 있으며, 기초 이론부터 최신 연구 동향까지 아우르는 심도 있는 교육의 장이 될 것으로 확신합니다.

여러분의 많은 관심과 참여를 기대합니다!

2026년 2월

한국생명정보학회장 류 성 호

강의개요

DNA Foundation Model 이론 및 실습

생물학 텍스트와 DNA 서열을 컴퓨터는 어떻게 이해할 수 있을까? 수억 편의 논문과 수십억 염기쌍의 게놈 데이터에서 의미 있는 패턴을 어떻게 찾아낼 수 있을까? 자연어 처리에서 발전한 언어 모델 기술은 생물정보학 연구의 방식을 근본적으로 변화시키고 있다.

본 강의에서는 텍스트와 DNA 서열 분석을 위한 AI 언어 모델의 핵심 원리를 다룬다. 단어를 벡터로 변환하는 Embedding 기법, 문맥을 이해하는 Attention 메커니즘, DNA 서열 학습하는 Foundation Model의 개념을 설명하며, 이를 DNA 유전체 변이 분석에 적용한 최신 모델들을 소개한다. 이를 통해 유전 변이 효과를 예측하고, 장거리 조절 상호작용을 분석할 수 있는 실무 능력을 갖추는 것을 목표로 한다.

강의는 다음의 내용을 포함한다:

- 언어 모델의 기본 원리와 생물학 텍스트 처리
- 대규모 사전학습 모델과 전이학습 방법론
- DNA 서열을 위한 언어 모델 설계
- 장거리 게놈 분석을 위한 차세대 아키텍처

* 참고강의교재:

  AI for Genomic Science (chaek.org) https://chaek.org/books/ai-for-genomic-science

* 교육생준비물:

  노트북 (인터넷 연결 필요 – 구글 코랩 실습을 위한 구글 계정)

* 강의 난이도: 중급

* 강의: 안준용 교수 (고려대학교 바이오시스템의과학부)

Curriculum Vitae

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

▶ **Personal Info**

Name	Joon-Yong An
Title	Associate Professor
Affiliation	Korea University

▶ **Contact Information**

Address	145 Anam-ro, Seongbuk-gu, Seoul, South Korea
Email	joonan30@korea.ac.kr

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
2019-2022	Assistant Professor, Korea University
2022-	Associate Professor, Korea University

Selected Publications (3 maximum)

1. Kim SW*, Lee H*, (...), Kim EJ**, Werling DM**, Yoo HJ**, An JY**, Whole genome sequencing analysis identifies sex differences of familial pattern contributing to phenotypic diversity in autism. Genome Medicine, 2024
2. Song KJ*, Choi S*, Kim K*, Hwang HS*, (...), Na S**, Jang SJ**, An JY**, Kim KP**, Proteogenomic Analysis of a Korean Cohort Reveals Lung Cancer Subtypes Predictive of Metastasis, Chromosome Instability, and Tumor Microenvironment, Nature Communications, 2024
3. Kim Y*, Jeong M*, (...), An JY**, CWAS-Plus: Estimating category-wide association of rare noncoding variation from whole-genome sequencing data with cell-type-specific functional data, Briefings in Bioinformatics, 2024

KSBi-BIML 2026

DNA Foundation Model

고려대학교 바이오시스템의과학부
안준용

DNA Foundation Model 소개

학습 목표: DNA를 언어처럼 다루는 AI 모델의 개념을 이해하고, 최근 급성장하는 DNA Language Model의 현황을 파악합니다.

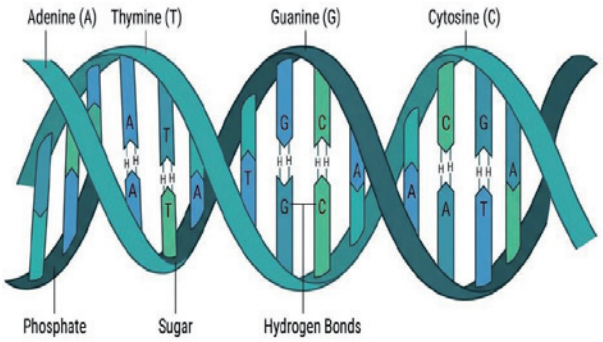
DNA 이중나선 구조

생명의 설계도를 이루는 분자

DNA는 이중나선 구조로 이루어져 있으며, 각 나선은 인산-당-염기 단위로 구성됩니다. 아데닌(A), 티민(T), 구아닌(G), 시토신(C)의 4가지 염기가 특정한 규칙에 따라 결정됩니다.

핵심 원리

A는 T와, G는 C와 짝을 이루며, 이 보완적인 결합은 DNA의 복제와 유전 정보 전달을 가능하게 합니다.



DNA의 이중나선 구조는 유전 정보를 안정적으로 보관하고, 정확하게 복사하며, 필요시 유전자 발현을 통한 단백질 합성에 활용됩니다. 이는 모든 생명체의 기본적인 원리입니다.

DNA Language Model

DNA를 언어처럼 다루는 AI 모델

Biological Sequence

Tokenization & Embedding

AI Model Architecture

유전자 기능 예측

서열 데이터만으로 유전자의 생물학적 기능을 추론

변이 영향 탐지

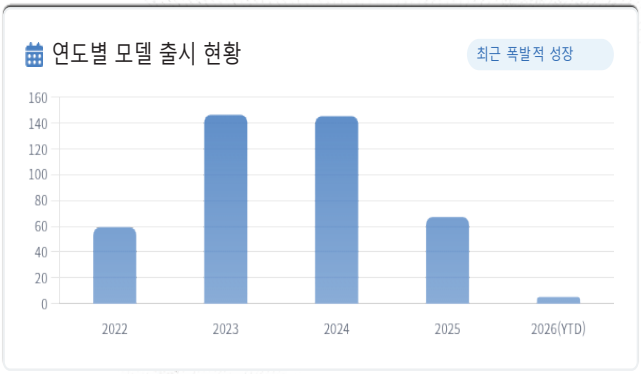
단일 염기 변이가 질병에 미치는 영향을 AI로 분석

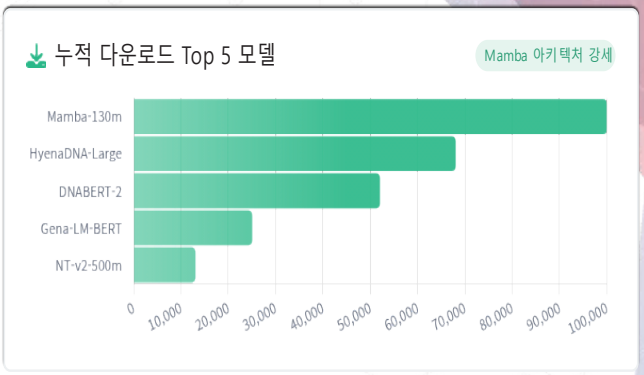
새로운 서열 생성

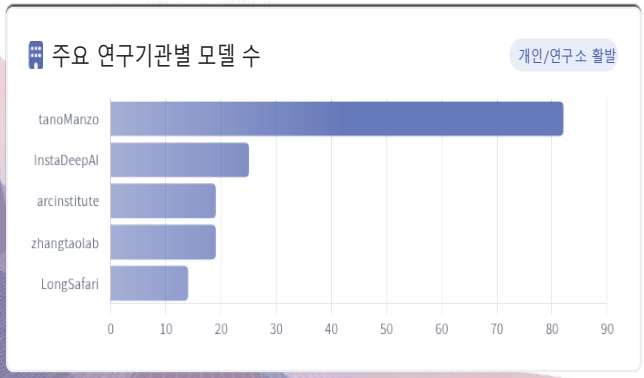
자연어 생성처럼 목적에 맞는 새로운 DNA 서열 설계

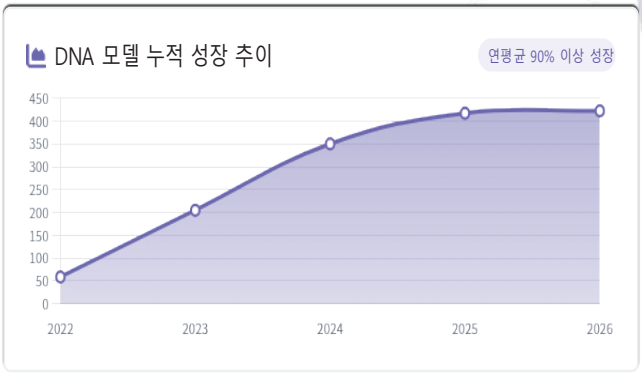
DNA Language Model의 급성장

Data Source: Hugging Face (2026.01)









Language Model 기본 원리

학습 목표: 시퀀스 확률 계산과 조건부 확률의 개념을 이해하고, DNA와 자연어의 차이점을 통해 DNA 모델링의 특수성을 학습합니다.

Language Model의 기본 원리

정의

시퀀스의 확률을 계산하고
다음 요소를 예측하는 AI 모델

DNA 시퀀스 시각화

A T G C

DNA는 4개의 염기로 구성된 문자열

핵심 기능

시퀀스 확률 계산 $P(x_1, ..., x_n)$

다음 요소 예측 $P(x_{n+1}|x_1...x_n)$

30억 염기
인간 게놈

ATGC
문자열 처리

모델 적용
Language Model

확률 기반 예측의 원리

모델은 주어진 문맥(Context)을 보고 가장 그럴듯한 다음 단어를 선택합니다

Context (문맥)

DNA는 _____

Option A

"유전정보를"

예측 확률 (Probability) 98.5%

문맥상 자연스러움 (생물학적 사실)

Option B

"사과를"

예측 확률 (Probability) 0.01%

문맥상 부자연스러움 (의미 불일치)

수학적 표현
조건부 확률 (Conditional Probability)

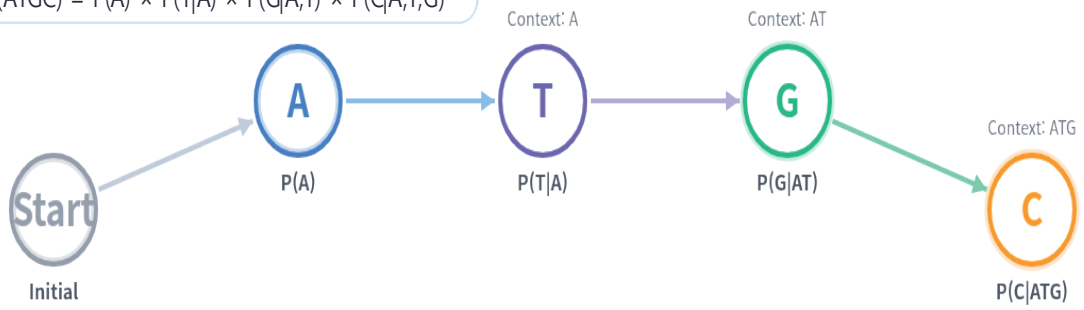
P(next | context)

모델은 수많은 데이터를 통해 이 확률 분포를 학습합니다.

조건부 확률 (Conditional Probability)

시퀀스의 확률 분해와 문맥의 확장

$P(ATGC) = P(A) \times P(T|A) \times P(G|A,T) \times P(C|A,T,G)$



오른쪽으로 갈수록 문맥(Context)이 누적됩니다

Chain Rule

긴 시퀀스의 확률을 단계별 조건부 확률의 곱으로 분해하여 계산의 복잡도를 낮춥니다.

Context의 역할

이전 단계의 모든 정보가 다음 단계의 예측 조건이 되어, 생물학적 문맥을 반영합니다.

DNA와 자연어의 차이점

단어 경계의 유무와 연속성

공백 유무

자연어: 있음 vs DNA: 없음

단어 경계

자연어: 명확함 vs DNA: 모호함

Tokenization

연속적인 DNA 서열을 인위적으로 잘라 '단어'처럼 처리하는 과정 필요

Tokenization

학습 목표: Single nucleotide, k-mer, BPE 등 다양한 토큰화 방법의 원리와 장단점을 비교하고, 각 방법의 생물학적 의미를 이해합니다.

Tokenization
DNA를 '단어'로 나누기

토큰화의 목적

DNA 시퀀스를 의미 있는 단위로 분할하고 모델이 처리할 수 있는 형태로 변환

주요 방법

Single nucleotide: A, T, G, C 각각

k-mer: 고정 길이 조합 (예: 6-mer)

BPE (Byte Pair Encoding): 데이터 기반 가변 길이

Single nucleotide

가장 간단한 방법

A T G C

k-mer

고정 길이 k의 연속 부분시퀀스

ATG TGC GCA CAT

BPE

데이터 기반 가변 길이 토큰

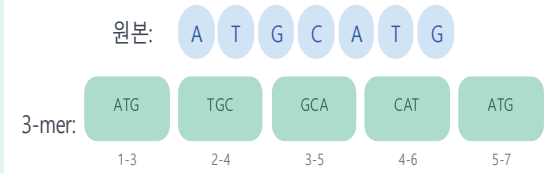
A AT ATG ATGC

k-mer Tokenization

k-mer란?

길이가 k인 모든 연속 부분시퀀스

실시간 k-mer 생성



장점

구현 간단

지역적 패턴 포착

단점

정보 중복 (오버랩)

어휘 크기 폭발: 4^k

k=6 → 4,096개 토큰

BPE (Byte Pair Encoding)

1. 각 뉴클레오타이드 A, T, G, C
2. 빈번한 쌍 찾기 최빈도 연속
3. 새 토큰 병합 하나의 단위로
4. 반복 어휘 확장

구체적 예시

단계 1	단계 2	단계 3
A, T, G, C	AT (빈번)	[AT]G → [ATG]

장점

데이터에서 자동 학습
가변 길이 토큰
효율적 어휘 크기

대표 모델

- GROVER
- DNABERT-2

Vocabulary (어휘)

정의

모델이 알고 있는 모든 토큰의 집합 (사전)

예시

```
Vocab = {
  0: "A",
  1: "T",
  2: "G",
  3: "C",
  4: "AT",
  5: "ATG",
...
}
```

어휘 크기의 중요성

↓ 너무 작음

표현력 부족 → 다양한 DNA 패턴을 학습할 수 없음

↑ 너무 큼

계산 비용 증가 → 학습 시간 및 메모리 사용량 급증

최적의 크기

적절한 균형점 찾기: 표현력과 효율성을 모두 만족

k-mer 토큰의 한계: 왜 고정 길이는 부족한가

1. 조합 폭증

4^k 기하급수적 증가 (k=8 → 65,536개): 희소성 문제, 데이터 요구량 급증

2. 변이에 취약

1bp 변화로 완전히 다른 토큰: OOV(Out-of-Vocabulary) 문제, 일반화 저하

3. 정보 중복

슬라이딩 윈도우로 중첩 발생: 95% 이상의 토큰이 5bp 이상 중복

4. 긴 문맥 처리 한계

장거리 상호작용 포착 어려움: 고정된 k-mer로는 enhancer-promoter 연결 불가

5. 도메인 확장성 제약

종·조건 변화에 따른 분포 이동: 새로운 종에 대한 일반화 어려움

해결책: 가변 길이 토큰화(BPE) 또는 단일 염기 토큰화 고려 필요

Single-nucleotide 토큰

정의

A, C, G, T 4개의 단일 염기를 각각 하나의 토큰으로 인식

장점

최고의 해상도 - SNP/indel 정밀 표현 가능
유연한 조합 학습 - 염기 간 관계 자유롭게 학습
미세한 변이 감지 - 단일 염기 변화도 즉각 반영

vocabulary 크기: 4 (최소)

단점

시퀀스 길이 증가	1kb = 1,000 tokens
메모리 사용량 ↑	수십만 tokens 처리
학습 시간 ↑	더 많은 연산 필요

계산적 고려사항

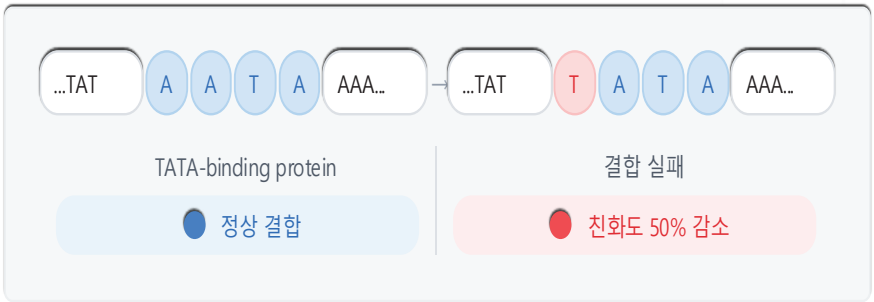
더 긴 context window 필요
효율적 attention 메커니즘 요구 압축 기법 적용 고려

적합: 변이 효과 예측, 세밀한 motif 경계 학습

Single-base 토큰의 생물학적 의미

Splice Site: GT-AG 모티프

TF 결합 부위: TATA Box



통계적 토큰화

개념

데이터에서 자주 함께 등장하는 서열을 자동으로 묶어 토큰을 생성하는 방식

핵심 원리

- 빈도 기반 자동 병합
- 가변 길이 토큰 생성
- 데이터 적응형 Vocabulary

작동 원리

1. 초기화: A, C, G, T 4개로 시작
2. 빈도 집계: 연속 쌍 빈도 계산
3. 병합: 최빈 쌍 → 새 토큰 생성
4. 반복: 목표 Vocab 크기까지 반복

장점

- 표현 효율 ↑
- 데이터 적응
- 가변 길이
- 사전 지식 ↓

BPE 토큰화 애니메이션

1 초기화 (V0 = {A, C, G, T}) → 2 빈도 집계 (빈발 쌍 찾기) → 3 병합 (최빈 쌍 → 신규 토큰) → 4 반복 (목표 vocab까지) → 5 토크나이즈 (최장 일치 분해)

BPE 학습 과정: ATGCATGCAT

초기 상태	1차 병합	2차 병합	3차 병합	최종 토큰
A T G C A T G C A T	AT G C AT G C AT	ATG C ATG C AT	ATGC ATGC AT	ATGC ATGC AT

장점

- 데이터에서 자동 학습
- 가변 길이 효율성
- 계층적 패턴 포착

주요 용도

- GROVER — 인간 게놈
- DNABERT-2 — 효율성↑

k-mer

장점
- ✓ 모티프 직접 표현
- ✓ 해석 용이

단점
- × 4^k 차원 폭증
- × 변이에 취약

적합
경량 분류기,
짧은 모티프 중심 과제

Single-base

장점
- ✓ 최고 해상도
- ✓ 변이 민감도 최상

단점
- × 시퀀스 길이 증가
- × 계산 비용 상승

적합
단일 변이 효과,
미세 조절 분석

BPE

장점
- ✓ 가변 길이 (효율성↑)
- ✓ 데이터 적응형

단점
- × 생물학적 직관성 낮음
- × 토큰 경계 불명확

적합
대규모 Foundation Model,
폭넓은 전이 학습

Token 선택의 방식

국소 vs 전역
모티프 강조(k-mer) ↔ 문맥 유연성(single/BPE)

고정 길이 vs 가변 길이
단순 처리 ↔ 효율적 압축

해석성 vs 표현력
직관(k-mer) ↔ 적응/용량(BPE)

사전 지식 vs 데이터 주도
규칙 주입 ↔ 분포 학습

안정성 vs 민감도
변이 둔감(k-mer) ↔ 민감(single)

핵심 요약

Token 선택은 근본적으로 "무엇을 의미 단위로 볼 것인가"를 결정합니다.
이 선택이 모델의 학습 방식과 해석 가능성을 모두 결정합니다.

Token 파트 핵심 정리

핵심 메시지

Token은 모델의 시각입니다. 선택이 모든 것을 결정합니다.

잊지 마세요!

Token = 모델이 세상을 보는 창
모든 학습은 Token 선택에서 시작됩니다

실전 체크리스트

- 변이 민감도 필요?
- 해석성 요구
- 메모리 제약
- 처리 속도
- Vocab 크기
- 목표 과제

권장 기본값

Foundation Model	BPE
Variant Effect	Single-base
경량/모티프	k-mer

실무 선택 가이드

Variant Effect Prediction
단일 염기 변이 효과 예측

Token	Single-base
Architecture	Transformer
Loss	MLM

대표 모델: DNABERT-2, Nucleotide Transformer

Sequence Generation
새로운 DNA 서열 생성

Token	BPE
Architecture	Autoregressive
Loss	CLM

대표 모델: EVO, HyenaDNA

Transfer Learning
다른 종/작업으로 전이

Token	BPE
Architecture	Transformer
Loss	MLM

대표 모델: NT, DNABERT-2, GPN

Long-range Interaction
장거리 조절 요소 탐지

Token	Single-base
Architecture	SSM
Loss	MLM

대표 모델: HyenaDNA, Caduceus, EVO

Embedding & Learning Objectives

학습 목표: 토큰을 벡터로 변환하는 임베딩의 원리와 MLM, Autoregressive, Supervised Learning 등 다양한 학습 방법을 이해합니다.

Embedding의 기본 개념

정의

토큰을 D차원 실수 벡터로 표현하여 신경망이 처리 가능하게 만드는 표현 학습

Embedding 차원

- 256 : 경량 모델
- 512 : 일반적 설정
- 768 : 대형 모델
- 1024 : 초대형 모델

변환 과정

의미

토큰의 의미를 좌표로 표현합니다. 공간에서의 거리와 방향이 유사성과 관계를 나타냅니다.

벡터 표현의 중요성

거리·방향·연속성으로 DNA의 의미를 기하학적으로 표현합니다

유사도 (Similarity)

벡터 간 각도(θ)가 작을수록 생물학적 기능이 유사합니다. 코사인 유사도로 이를 정량화합니다.

방향성 (Directionality)

공간 상의 특정 방향이 생물학적 속성(예: GC 함량, 모티프 종류)을 나타냅니다.

연속성 (Continuity)

이산적인 DNA 서열을 연속 공간으로 매핑하여, 미분 가능한 최적화와 보간이 가능해집니다.

중간 표현의 가치

임베딩은 최종 출력 이전의 풍부한 특징 벡터

예측보다 더 근본적인 표현 학습

범용 표현 (General-purpose)

하나의 임베딩으로 다양한 태스크 재사용 가능

예측과의 대비

최종 출력 vs 중간 표현

MLM: 4-way 염기 확률 → 임베딩: 문맥적 의미 공간

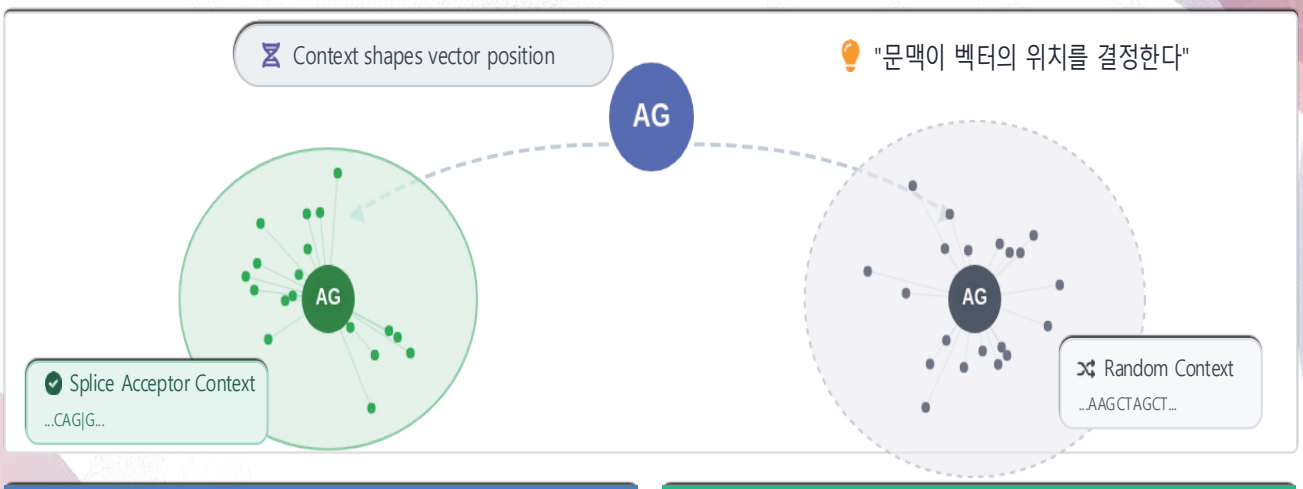
태스크 비종속적

잘못된 예측도 유용한 표현을 제공할 수 있음

실무 활용 → **모델 재사용**
임베딩 동결 + 선형 프로브 → **빠른 수렴**
소량 라벨로도 높은 성능

Contextual Embedding 같은 토큰, 다른 문맥 = 다른 벡터

동일한 서열이라도 주변 환경(Context)에 따라 의미가 달라집니다



Attention Mechanism

Transformer가 주변 토큰들의 정보를 집계(Aggregation)하여 현재 토큰의 벡터를 갱신합니다. 어텐션 가중치에 따라 정보가 동적으로 조합됩니다.

생물학적 의미

위치 의존성(프로모터 vs 인트론)과 조합 효과를 반영합니다. 동일한 "AG" 서열도 주변 모티프와의 관계에 따라 기능적 역할이 결정됩니다.

문맥이 임베딩을 바꾸는 생물학적 이유

위치 의존성

프로모터, 인트론, 인터제닉 등 게놈 맥락에 따라
동일 서열의 기능이 완전히 달라집니다

조합 규칙

전사인자 모티프의 순서, 간격, 동시성이 기능 결정

한 모티프만으로는 enhancer 기능 예측 불가능

장거리 상호작용

enhancer-promoter 상호작용

enhancer — promoter

수만~수십만 bp 떨어져도 3D 접촉 가능

3D 구조와 후성유전학

chromatin 구조, histone modification

기능적 의미는 1차 서열을 넘어서는 요소들에 의해 결정

서열 조합/분포

학습 내용

- 함께 출현 패턴
- 금지 조합 규칙
- 자연스러운 분포

예시

"ATG" 다음에 "CG"가 오는 확률이 높은 이유

"TA" 반복은 불안정하므로 드물게 출현

기능적 제약/보존

학습 내용

- 필수 염기 위치
- 진화적 보존
- 변이 허용도

예시

Splice site의 GT/AG는 거의 변하지 않음

TF 결합 부위의 핵심 염기는 보존됨

반복/특이성/문법

학습 내용

- 반복 서열 시그니처
- 특이적 조절 요소
- DNA 문법 규칙

예시

Alu 반복을 위한 특별한 벡터 패턴

Enhancer의 특이적 조합 규칙 학습

모든 정보는 수백~수천 차원의 벡터 하나에 압축됩니다

Embedding의 재사용

다운스트림 태스크

프로모터 분류

발현 수준 회귀

요소 클러스터링

검색 및 유사도

임베딩 간 거리로 유사한 서열 패턴을 찾아냄

변이 효과 분석

WT: ATGCGT vs Mut: ATG< A GT

ΔEmbedding 거리 계산

||Emb(WT) - Emb(Mut)|| = 0.82

값이 클수록 기능적 영향 ↑

전이 전략

고정 임베딩 + 선형 헤드	빠른 실험
부분 미세조정	균형
전면 미세조정	최대 성능

블랙박스의 한계

- 각 차원의 의미가 불명확
- 직접적인 해석 어려움
- 고차원 공간의 복잡성

해석 도구들

- t-SNE / UMAP (2D/3D 시각화)
- Attention Weight (중요 위치)
- Saliency/Gradient (특징 탐색)

간접적 해석: 모델이 학습한 규칙을 복원

임베딩 학습의 목적

보편적 표현 학습

DNA 서열의 일반적 패턴을 학습하여
종과 조건을 넘나드는 범용적 이해

종 간 전이 가능성

인간 게놈 → 생쥐 게놈으로의 전이

보존된 패턴은 여러 종에서 유효

멀티태스크 전이 학습

- Promoter 인식
- Splice site 예측
- Enhancer 발견
- Variant 효과

하나의 임베딩으로 다양한 생물학적 과제 해결

Foundation Model의 가치

대규모 사전학습

- 수십억 염기로 일반적 패턴 학습
- 소량의 라벨로도 빠른 적응
- 다운스트림 태스크에 즉시 활용

Single-base

4개 토큰 최소 단위, 해상도 최고

✓ SNP/indel 정밀 표현

× 시퀀스 길이 ↑, 계산 비용 ↑

A → [0.1, -0.3, ...]
문맥에 따라 임베딩이 다양하게 변화

6-mer

4,096개 토큰 국소 모티프 직접 표현

✓ 생물학적 직관성 우수

× 변이에 민감, vocab 고정

"ATGCGT" → [0.2, 0.5, ...]
비교적 안정적 의미, but 변이 시 완전히 새 토큰

BPE

가변 길이 데이터 기반 자동 학습

✓ 빈도 적응, 효율성 ↑

× 생물학적 직관성 낮음

"ATG" + "CGT" → [0.3, -0.1, ...]
자주 나타나는 패턴을 압축, but 해석 어려움

Token 설계가 임베딩 공간 구조를 결정 → Embedding 공간의 기하학적 구조가 달라진다

Embedding 파트 핵심 정리

핵심 메시지

Embedding = 의미를 담은 수학적 공간
문맥적·연속적 표현이 일반화를 만든다

Foundation Model의 핵심

한 번 학습한 embedding을 다양한 과제에 재사용 가능

실무 체크리스트

- 임베딩 차원 D
  용량 vs 과적합 균형
- 전이 전략
  동결/부분/전면 미세조정
- 포지셔널 인코딩
  절대/상대 위치 선택
- 유사도 측정
  코사인/유클리디안
- 풀링 방식
  CLS/평균/어텐션
- 컨텍스트 창
  학습 및 추론 시 고려

다음 단계 질문: 임베딩은 어떻게 학습되는가?

핵심 질문

어떤 Loss가 '좋은 임베딩'을 정의하는가?

Loss function은 임베딩이 가져야 할 성질을 암시적으로 결정합니다.

로드맵

1. Self-supervised 개념
2. MLM(마스킹) 메커니즘
3. Autoregressive(다음 토큰) 대비
4. Loss가 임베딩 기하에 미치는 영향

핵심 인사이트

"좋은 임베딩"은 목적이 아니라 결과입니다. 목표는 자연스러운 서열 분포를 학습하는 것이고,
그 과정에서 의미 있는 표현이 자연스럽게 생성됩니다.

학습의 역할

정의

'좋은 예측'의 기준을 설정해
파라미터를 조정하는 목적 함수

핵심 기능

- 무엇을 중요시할지 결정
- 주목하는 패턴이 달라짐
- 모델 행동의 방향성을 제시

Loss의 효과

보상	벌점
자연스러운 예측	부자연스러운 예측

실무 포인트

목표가 데이터 활용 방식을 규정

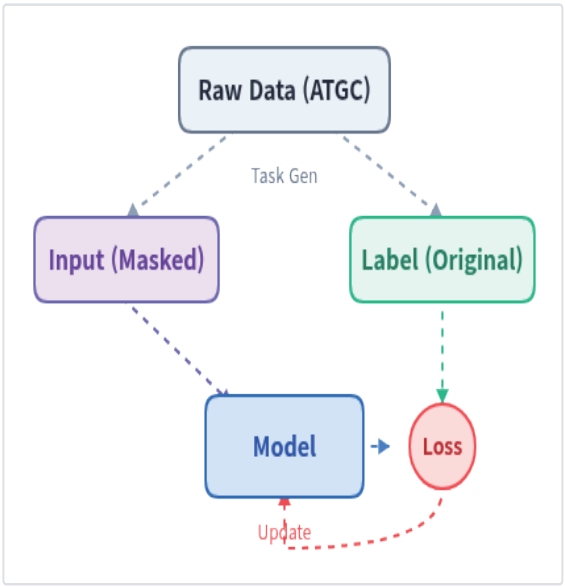
모델의 일반화 범위를 결정

모델의 정체성을 형성

Self-supervised Learning

데이터가 스스로 학습 과제를 생성하는 메커니즘

Self-Supervision Loop



핵심 원리: 원본 데이터(Raw Data)를 변형하여 입력(Input)과 정답(Label)을 자동으로 생성하고, 모델이 이 관계를 복원하며 학습합니다.

패러다임 비교: 문맥의 방향성

MLM

Masked Language Model

양방향(Bidirectional) 문맥을 모두 참조하여 빈칸을 채움

AR

Autoregressive

과거(Left) 문맥만 참조하여 미래(Next)를 예측

MLM (Masked Language Modeling)

DNA 서열 마스킹 과정

원본: ATGCA	→	AT [MASK] CA
원본: TATAAA	→	TAT [MASK] A
원본: GGTTCG	→	GGT [MASK] CG

문맥 활용 예시

...[TATA]...	→	TATA box
...AG[GT]...	→	splice donor
...G[ATA]...	→	promoter motif

1 데이터 준비
전체 게놈 서열 수집

2 마스킹
15% 토큰 랜덤 마스킹

3 문맥 활용
양방향 문맥으로 예측

장점

양방향 문맥 이해 (Bidirectional)

자연스러운 서열 패턴 학습

전역 의존성(Long-range dependency) 포착

대표 모델

DNABERT — Foundation

Nucleotide Transformer — SOTA

서열 조합/분포

학습 내용

- 함께 출현 패턴
- 금지 조합
- 자연스러움 통계

예시

CG가 TA보다 흔함,
CpG 제약, codon bias

의미

진화적 선호,
돌연변이 제약 반영

기능적 제약/보존

학습 내용

- 필수 염기
- 선택압
- 변이 허용도

예시

splice GT-AG, start codon ATG,
active site

의미

기능적 중요성
= 낮은 엔트로피 = 높은 확신

반복/특이성/문법

학습 내용

- 반복서열 시그니처
- 조절 요소 특이성
- DNA 문법

예시

Alu, LINE, LTR,
promoter motif, TATA-box

의미

유전자 조절, 게놈 구조,
진화적 유산

간접 학습의 힘

- 기능 라벨 없이도기능의 '흔적'을 학습
- 진화적 보존 →자연스러운 패턴 포착

"분포가 기능을 반영한다"

일반화 능력

- 종·조건 변화에도유지되는 보편 패턴
- 수조 염기로 학습 →미세한 패턴까지 포착

"보편적 서열 이해 획득"

Supervised Learning
특정 과제 최적화, 범용성 제한

→

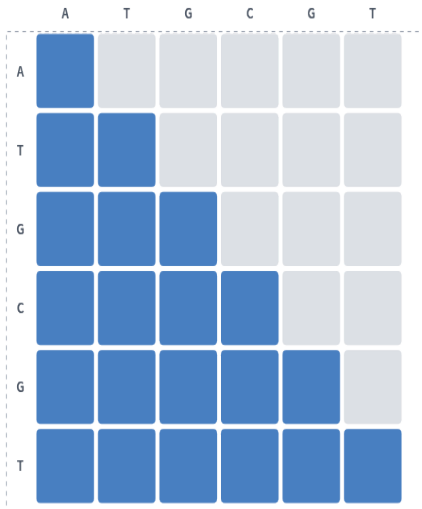
MLM (Masked LM)
범용 표현 학습, 전이 용이

Autoregressive Language Modeling

Causal Masking을 통한 단방향 순차 생성 프로세스

Causal Masking Matrix

Look-back only



순차 생성 시뮬레이션

Step-by-step

장점

자연스러운 생성 (Generative)

임의 길이 시퀀스 확장 가능

대표 모델

Evo DNABERT-1

Cross-Entropy Loss

예측 분포와 실제 분포의 차이를 기하학적으로 측정

$$L = -\Sigma_y p^*(y) \log q_\theta(y)$$

Ground Truth (p*)

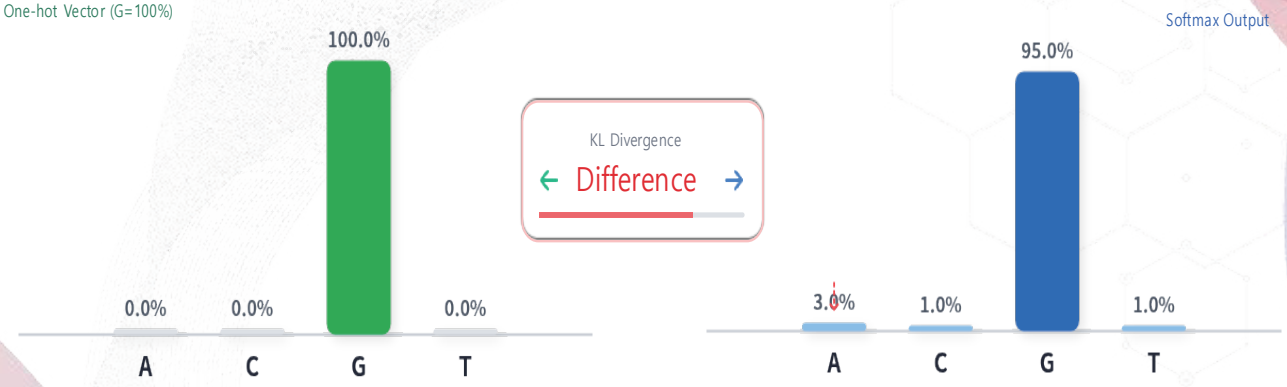
실제 분포

One-hot Vector (G=100%)

Model Prediction (qθ)

모델 예측 분포

Softmax Output



실제 분포 (Target)

특정 위치의 정답 염기는 하나뿐입니다.

(예: G=1.0, 나머지는 0.0)

모델 분포 (Prediction)

모델은 모든 염기에 대해 확률을 할당합니다.

(예: G=0.95, A=0.03...)

Loss의 역할

예측 분포를 실제 분포로 '밀어넣는' 힘.

불확실성(Entropy)을 최소화합니다.

Supervised Learning

- 특정 과제 최적화 (분류/회귀)
- 높은 태스크 성능
- 전이 범위 제한

"이 서열은 promoter인가?"

· 명확한 정답 라벨 필요
· 과제 특화 모델

Self-supervised Learning

- 서열 이해 최적화
- 범용 임베딩
- 라벨 효율↑, 전이 용이

"이 서열이 자연스러운가?"

· 라벨 없이도 학습 가능
· 범용 표현 획득

선택 가이드 → 라벨 풍부·명확 과제 →Supervised → 범용 표현·저라벨→Self-supervised

Loss와 임베딩의 연결

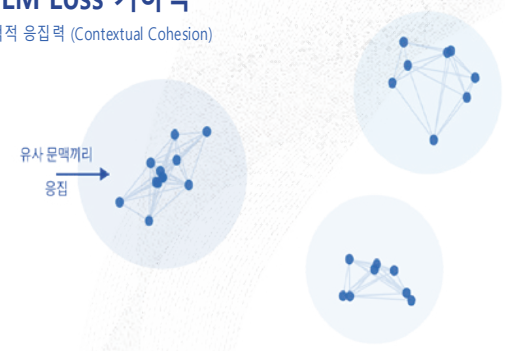
학습 목표(Loss)가 벡터 공간의 형태를 결정합니다

"Loss는 임베딩 공간의 기하학을 조각한다"

Scenario A

MLM Loss 기하학

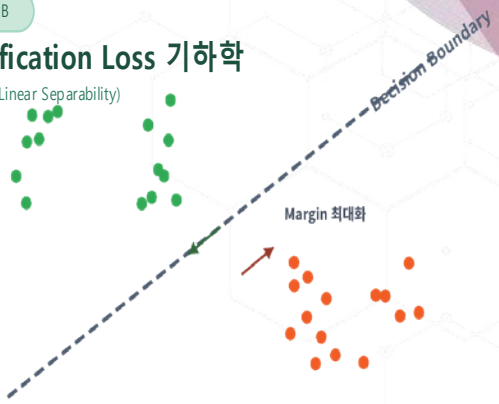
문맥적 응집력 (Contextual Cohesion)



Scenario B

Classification Loss 기하학

선형 분리 (Linear Separability)



MLM Loss의 영향

- ✓ 문맥 예측 가능성 강조 (Predictability)
- ✓ 인접/유사 토큰끼리 가까운 거리 형성
- ✓ 데이터의 전역적 분포 패턴 학습

분류 Loss의 영향

- ✓ 클래스 간 경계(Margin) 최대화
- ✓ 선형 분리가 가능한 공간 구조 형성
- ✓ 명확한 결정 경계(Decision Boundary) 생성

MLM

언제 사용?

- ✓ 범용 이해/전이
- ✓ 대규모 사전학습
- ✓ 라벨 없는 데이터

장점

- + 양방향 문맥 활용
- + 범용 표현 생성

예시

DTI, Nucleotide Transformer

→ Autoregressive

언제 사용?

- ✓ 생성/설계 탐색
- ✓ 시퀀스 샘플링
- ✓ 생성적 모델

장점

- + 자연스러운 생성
- + 임의 길이 샘플링

예시

Evo, Mamba

Supervised

언제 사용?

- ✓ 명확한 라벨
- ✓ 명확 성능 지표
- ✓ 특정 과제 최적화

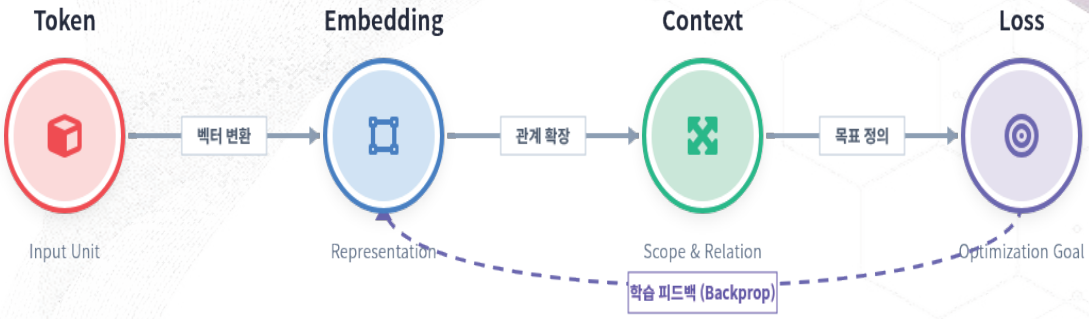
장점

- + 높은 태스크 성능
- + 직관적 목표

예시

DeepSEA, GBT

개념 연결: Token >Embedding > Context > Loss



시나리오 1: 변이 예측 (Variant Prediction)

Single-base Token | Dense Embedding | Local Context | Classification

시나리오 2: 범용 표현 (General Representation)

BPE Token | Large Embedding | Global Context | MLM Loss

Embedding 해석과 MLM Loss

학습 목표: Embedding 공간의 기하학적 의미를 해석하고, MLM Loss가 모델 학습과 아키텍처 설계에 미치는 영향을 이해합니다.

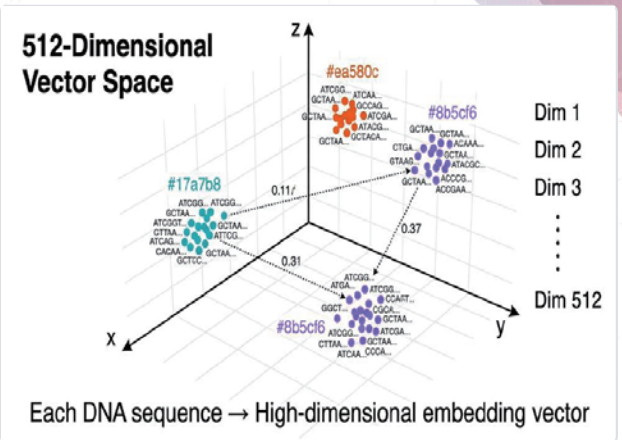
Embedding 해석의 출발점

수백 차원의 벡터 공간에 담긴 생물학적 의미

Embedding은 수백 차원의 숫자 벡터입니다. 그 자체로는 직관적이지 않지만, 이 고차원 공간의 구조는 DNA 서열 간의 생물학적 관계와 기능을 반영하고 있습니다.

핵심 원리

비슷한 기능을 하는 서열은 공간상에서 가까이 모이고(Clustering), 다른 기능은 멀리 떨어집니다. 이 공간의 기하학적 분포를 분석하면 모델이 학습한 생물학적 문법을 이해할 수 있습니다.



Embedding 해석은 블랙박스를 여는 핵심 도구입니다. 시각화된 고차원 공간의 클러스터링 패턴을 통해 서열의 기능적 유사성과 잠재된 생물학적 제약을 직관적으로 파악할 수 있습니다.

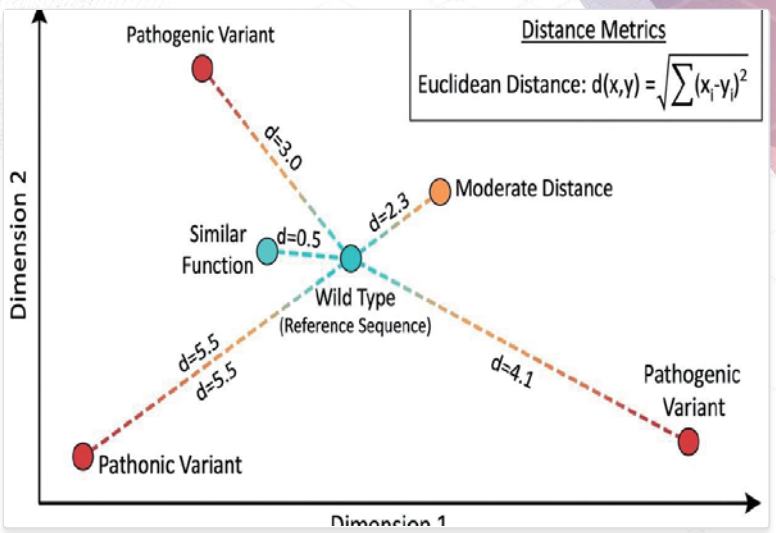
거리(distance)의 의미

유사도와 기능적 관계의 척도

Embedding 공간에서 두 벡터 간의 거리는 단순한 수치가 아닌 생물학적 유사도를 나타냅니다. 거리가 가까울수록 기능적으로 유사하거나 같은 역할을 수행하며, 멀수록 서로 다른 특성을 가집니다.

핵심 개념

DNA Embedding에서는 단순한 유클리드 거리보다 벡터의 방향성을 고려한 코사인 유사도(Cosine Similarity)가 더 자주 사용됩니다.



유사도 측정 지표

두 서열이 공간상에서 얼마나 가까운지를 정량화합니다. 유클리드 거리는 절대적 위치 차이를, 코사인 유사도는 패턴의 방향적 일치도를 측정합니다.

기능적 클러스터링

기능이 같은 서열들은 자연스럽게 뭉칩니다. 예: 여러 Splice acceptor site나 Start codon(ATG, GTG)들은 서로 가까운 클러스터를 형성합니다.

생물학적 구분

반대로 기능이 다른 요소들은 멀리 떨어져야 합니다. Promoter와 Terminator의 임베딩 거리가 멀다는 것은 모델이 그 기능적 차이를 학습했음을 의미합니다.

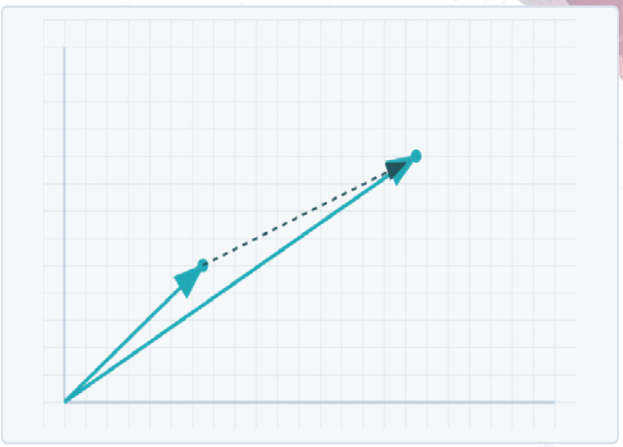
방향(direction)의 의미

Embedding 벡터 연산과 생물학적 속성

Embedding 공간에서 두 점 사이의 '방향'은 단순한 위치 차이가 아닌, 생물학적 속성의 구체적인 변화를 의미합니다. 특정 축으로의 이동은 안정성, 기능성, 혹은 병원성과 같은 특성의 증감과 연결됩니다.

벡터 연산 (Vector Arithmetic)

자연어 처리의 "King - Man + Woman = Queen" 예시처럼, DNA에서도 [Wild Type] + [Pathogenic Vector] = [Disease Variant]와 같은 연산이 성립하여 유전자의 기능적 진화를 예측할 수 있습니다.



위 그림은 임베딩 공간 내에서 정상 유전자(Wild Type)가 질병 유발 변이(Variant)로 변화할 때, 특정 '방향 벡터'가 그 병원성(Pathogenicity)의 특성을 나타냄을 보여줍니다.

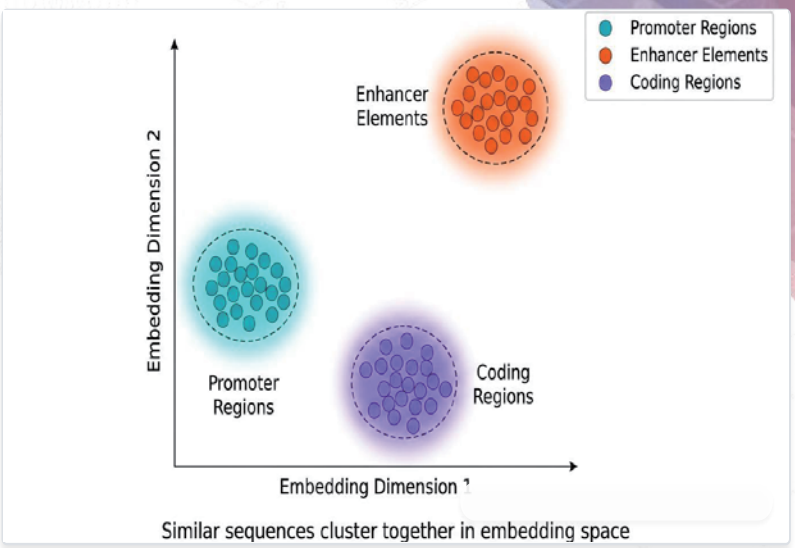
군집(clustering)의 의미

Embedding 공간의 그룹화

Embedding을 2D나 3D로 시각화하면, 모델이 학습한 데이터가 무작위로 분포하지 않고 특정한 패턴을 형성하는 것을 볼 수 있습니다. Promoter, Enhancer 등 기능적으로 유사한 서열들이 서로 가까이 모여 자연스러운 군집(Cluster)을 형성합니다.

핵심 원리

이러한 군집은 모델이 명시적인 라벨링 없이도 DNA 서열의 기능적 문법과 의미를 스스로 파악했음을 증명합니다. 데이터의 구조가 곧 모델의 지식이 됩니다.



Similar sequences cluster together in embedding space

기능적 군집화 (Functional Grouping)

비슷한 생물학적 기능을 수행하는 서열들은 Embedding 공간에서 물리적으로 인접합니다. 거리는 곧 기능적 유사성을 의미합니다.

비지도 학습 (Unsupervised Learning)

"이것은 Promoter다"라고 가르치지 않아도, MLM Loss를 최적화하는 과정에서 모델은 서열의 맥락을 통해 스스로 범주를 구분합니다.

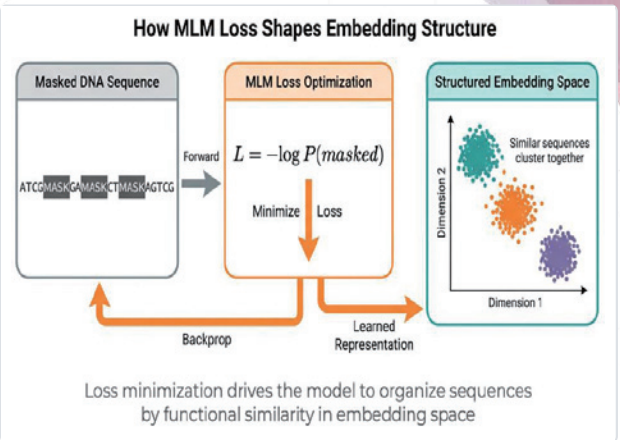
Loss가 구조를 결정하는 방식

MLM loss가 embedding 공간의 기하학적 구조를 형성합니다

MLM은 "문맥이 비슷하면 토큰도 비슷하다"를 학습합니다. 같은 문맥에 나타나는 토큰들은 서로 바뀌어도 loss가 크게 증가하지 않아야 합니다.

학습 메커니즘

예를 들어, "CAG|G..."와 "CAG|A..." 둘 다 가능한 splice site라면 G와 A의 해당 위치 embedding이 어느 정도 가까워야 loss를 낮출 수 있습니다. Loss 최소화 과정에서 embedding 공간이 재구성됩니다.



Loss minimization drives the model to organize sequences by functional similarity in embedding space

좋은 예측을 위해 필요한 구조가 자동으로 형성됩니다. 이런 방식으로 embedding은 문맥적 유사성을 반영하는 공간이 됩니다.

Variant 효과 해석과 Embedding

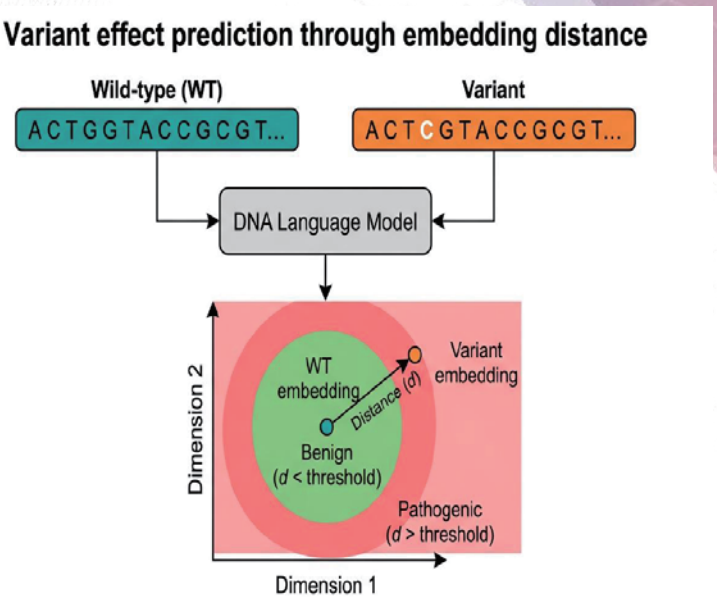
벡터 공간에서의 거리 차이를 통한 예측

해석 원리

DNA 언어모델은 서열을 고차원 벡터 공간(Embedding Space)으로 매핑합니다. Wild-type과 Variant 서열 간의 Embedding 거리가 클수록 해당 변이가 단백질 기능이나 유전자 발현에 큰 영향을 미칠 가능성(Pathogenicity)이 높습니다.

핵심 인사이트

단순한 염기 변화(A→T)라도 문맥에 따라 Embedding 변화량이 다릅니다. 중요 위치(Splice site, Start codon)의 변이는 큰 벡터 이동을 유발하며, 이는 모델이 생물학적 문맥을 이해하고 있음을 보여줍니다.



Distance-based Prediction

두 벡터 사이의 유클리드 거리(L2 norm)나 코사인 유사도를 계산하여 변이의 영향력을 단일 스칼라 값으로 정량화합니다.

Direction-based Analysis

벡터가 이동한 방향을 분석하여 변이가 어떤 기능적 속성(예: 소수성 변화, 구조 변형)으로 변화했는지 정성적으로 해석합니다.

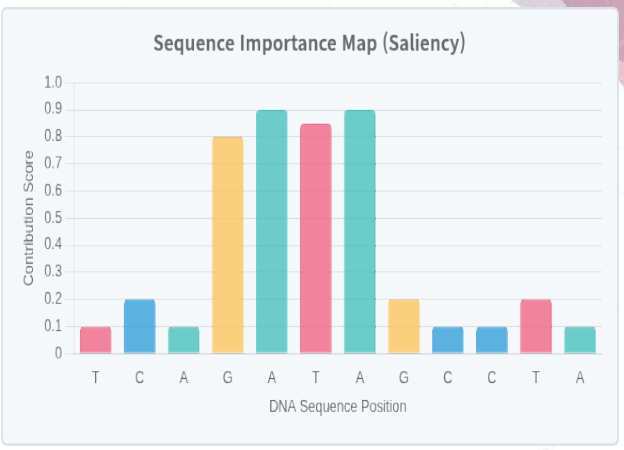
Attribution: 어떤 위치가 중요한가

모델의 예측 근거를 역추적하는 해석 기법

Attribution 분석(기여도 분석)은 딥러닝 모델이 특정 예측 결과를 도출할 때, 입력된 DNA 서열의 어느 위치(Nucleotide)가 결정적인 역할을 했는지 식별하는 과정입니다. 이를 통해 블랙박스 모델 내부를 들여다보고, 모델이 생물학적으로 유의미한 모티프(Motif)를 학습했는지 검증할 수 있습니다.

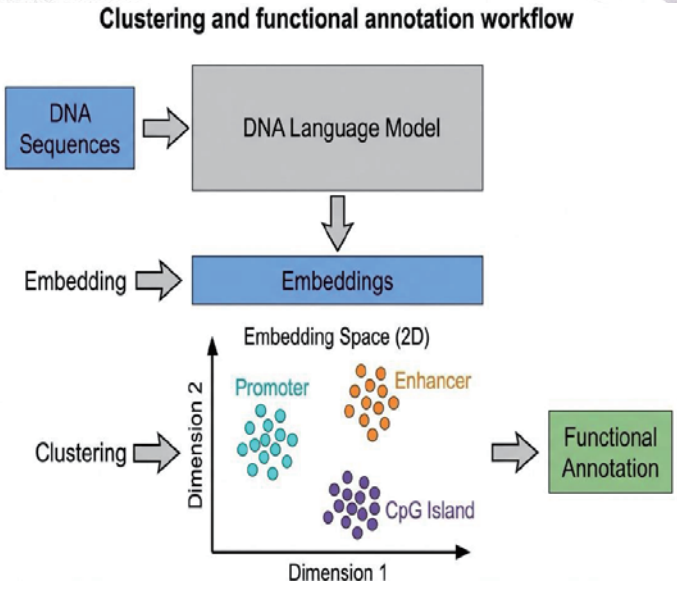
핵심 분석 기법

- In silico Mutagenesis: 가상의 돌연변이를 생성하여 예측 변화 관찰
- Saliency Maps: Gradient 기반으로 입력 위치별 민감도 측정
- Integrated Gradients: 기준점 대비 기여도를 적분하여 정확도 향상



Attribution Score 예시: 위 그래프는 DNA 서열의 각 염기가 모델 예측에 기여한 중요도를 나타냅니다. 높은 막대는 해당 위치의 염기가 모델의 의사결정에 핵심적인 역할을 수행함을 의미하며, 이는 전사 인자 결합 부위 등 기능적 요소와 일치하는 경우가 많습니다.

Clustering과 Functional Annotation



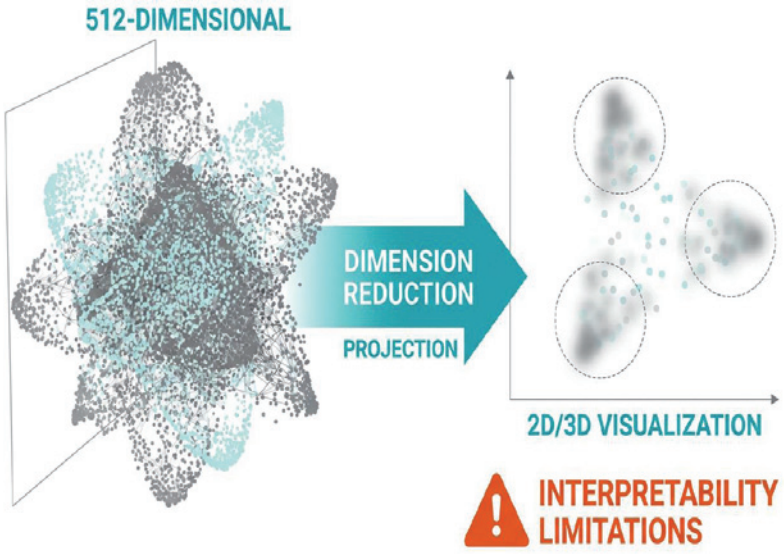
Key Insight

모델에게 명시적으로 기능을 가르치지 않았지만, MLM(Masked Language Modeling) Loss를 줄이는 학습 과정에서 자연스럽게 서열의 기능적 그룹화가 형성됩니다. 이는 딥러닝 모델이 생물학적 문맥을 이해하고 있음을 시사합니다.

해석 가능성의 한계

고차원 임베딩의 블랙박스 특성

DNA 언어모델의 임베딩은 수백 차원의 고차원 공간에 존재합니다. 이 복잡한 공간을 우리가 이해할 수 있는 저차원(2D/3D)으로 투영하는 과정에서 필연적으로 정보의 손실과 왜곡이 발생합니다.



핵심: 모델이 학습한 생물학적 문맥은 고차원 공간에 분산되어 있어, 단순한 시각화로는 그 의미를 온전히 파악하기 어렵습니다.

1. 차원의 의미 불명확

512차원 중 특정 차원이 구체적으로 어떤 생물학적 의미를 갖는지 알 수 없습니다. 비선형 변환으로 인해 인간의 직관이 통하지 않는 영역입니다.

2. 상관-인과 혼동

Attention 가중치가 높다고 해서 반드시 기능적 인과관계를 의미하지 않습니다. 단순한 통계적 상관관계와 실제 생물학적 메커니즘을 혼동하지 않도록 주의해야 합니다.

3. 시각화의 한계

t-SNE나 UMAP 등으로 차원을 축소하면 전체적인 구조는 보존될 수 있으나, 세부적인 거리 관계나 밀도 정보가 왜곡되어 잘못된 해석을 유도할 수 있습니다.

Embedding 해석 핵심 정리

Embedding Space란?

단순한 숫자 벡터가 아닌, 생물학적 의미를 담고 있는 고차원 공간입니다.

MLM loss를 줄이는 과정에서 자연스럽게 의미 있는 구조가 형성됩니다. 이는 모델 이해의 핵심이자 다양한 downstream 분석의 출발점입니다.

한계 인식

차원의 의미가 불명확하고 비선형성으로 인해 직관이 깨질 수 있으므로, 조심스럽고 비판적인 접근이 필요합니다.

Loss가 구조를 결정한다

Masked Language Modeling (MLM)은 단순한 빈칸 채우기가 아닙니다. 수백만 염기(Mb) 떨어진 문맥을 이해하면서도(Long-range), 동시에 단 하나의 염기 차이도 구분해야 하는(Single-base) 가혹한 최적화 문제입니다.

Key Insight

Objective Function(Loss)의 요구사항이 결국 아키텍처의 진화 방향을 강제합니다.

1. Objective (MLM Loss)

모델이 풀어야 할 궁극적인 과제

요구사항: "전체 유전체 문맥(Long-range)을 보면서 개별 염기(Single-base)를 정확히 맞추시오"

2. Architecture Choice

Loss를 최소화하기 위한 구조적 해법

Transformer	SSM / Mamba	U-Net
Global Attention	Linear State	Hierarchy
Cost: High ($O(n^2)$)	Cost: Low ($O(n)$)	Cost: Medium

3. Performance

구조적 선택에 따른 최종 결과

ACCURACY	SCALE
Variant Effect	1M+ Context

왜 이제 Loss를 다시 보는가?

구조(Architecture)는 중요하지만, 그것은 도구일 뿐입니다. 모델이 실제로 무엇을 학습할지는 Objective Function (Loss)이 결정합니다.

> "같은 구조라도 Loss가 다르면 전혀 다른 모델이 됩니다."
>
> — 학습의 방향성을 결정하는 핵심 키

1. MLM Loss의 정의 (The Demand)

"문맥을 보고 빈칸을 채워라"

요구사항: 수만 염기 떨어진 Enhancer와 Promoter의 관계를 파악해야만 현재 위치의 [MASK]를 맞출 수 있음.

2. 구조적 진화 (The Evolution)

Loss를 최소화하기 위한 필연적 선택

진화 압력: Long-range 문맥을 못 보면 Loss가 줄지 않음 → Attention ($O(n^2)$)의 한계 봉착 → SSM/Mamba ($O(n)$)로 진화 강제됨

3. 임베딩 공간 형성 (The Result)

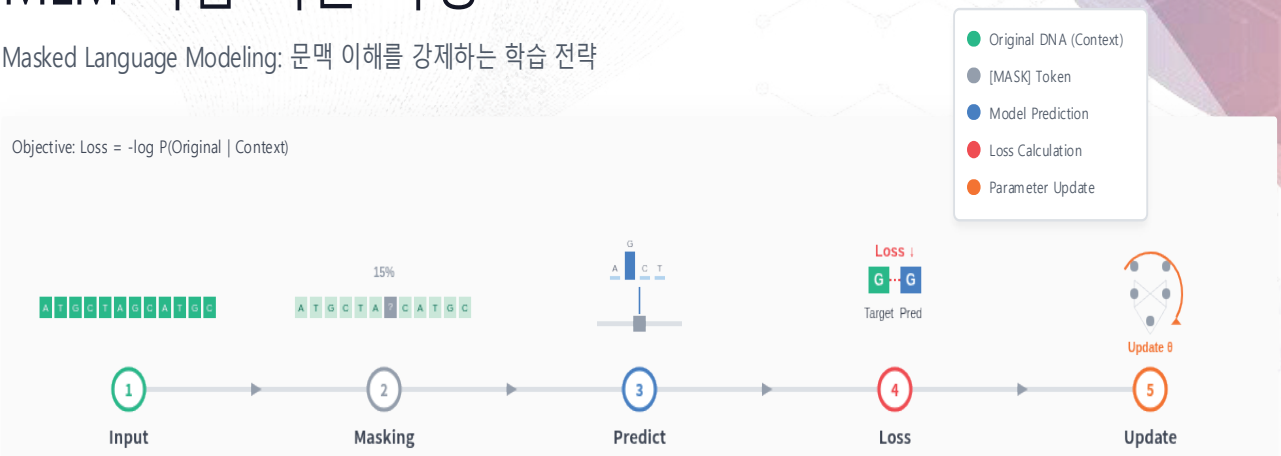
학습된 표현(Representation)의 기하학

결과물: 기능적으로 유사한 유전자들이 공간상에 가깝게 배치됨. (Sequence Similarity가 아닌 Functional Similarity 학습)

MLM 학습 기본 과정

Masked Language Modeling: 문맥 이해를 강제하는 학습 전략

Objective: Loss = -log P(Original | Context)



1. Masking Strategy

전체 염기 중 15%를 무작위로 가려(Masking), 모델이 주변 문맥을 통해 이를 추론하도록 강제합니다.

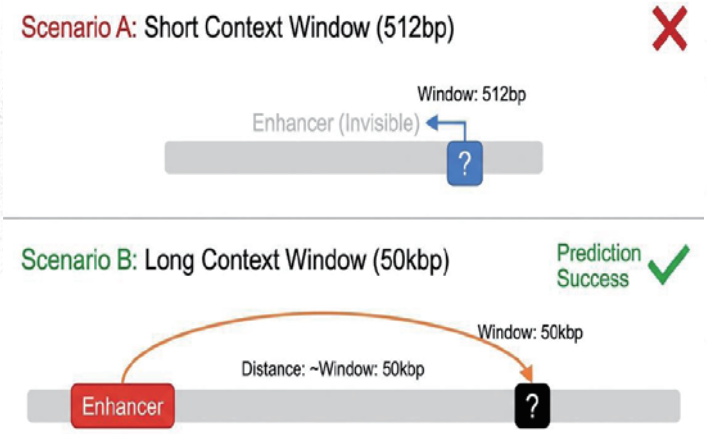
2. Loss Calculation

모델의 예측 확률 분포와 실제 원본 염기(Ground Truth)를 비교하여 Cross Entropy Loss를 계산합니다.

3. Model Update

계산된 Loss를 기반으로 역전파(Backpropagation)를 수행하여 모델의 파라미터를 업데이트합니다.

Short Context로는 해결되지 않는 경우



Short Context (Failure)

512bp의 좁은 시야로는 주변 서열만 보여 평범해 보입니다. 핵심 단서인 Enhancer가 시야 밖에 있어 예측에 실패합니다.

Long Context (Success)

50kb+의 넓은 시야로 멀리 떨어진 Enhancer를 포착합니다. 조절 요소와 타겟 유전자의 관계를 파악하여 정확히 예측합니다.

The Distance Gap

생물학적 상호작용은 수십~수백 kb 거리에서 발생하지만, 기존 모델(Transformer)은 계산 비용 문제로 이를 보지 못했습니다.

Loss가 Context 확장을 강제하는 메커니즘

핵심: "벌점(Loss)을 피하려면 더 넓게 봐야 한다"

Penalty Signal

문맥이 부족하면 모델은 틀린 예측을 반복하고, 높은 Loss(벌점)를 지속적으로 받게 됩니다.

Expansion Pressure

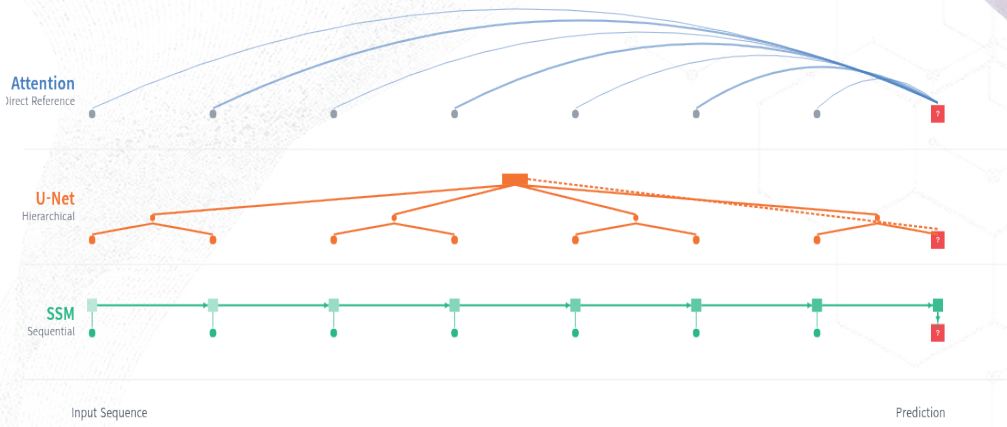
Loss를 줄이기 위한 유일한 방법은 정보량을 늘리는 것입니다. 이는 아키텍처가 Context Window를 넓히도록 진화를 강제합니다.

New Architecture

결과적으로 SSM, U-Net과 같이 초장거리(Long-range) 문맥을 효율적으로 처리하는 새로운 아키텍처가 등장합니다.

아키텍처별 MLM Loss 만족 방식

같은 Loss를 목표로 하지만, 정보를 수집하고 전달하는 경로는 서로 다릅니다.



Attention (Direct) U-Net (Hierarchical) SSM (Sequential) Target [MASK]

Attention

$O(n^2)$

모든 토큰 간의 관계를 직접 계산하여 [MASK]에 필요한 정보를 즉시 가져옵니다.

최고 표현력 높은 계산 비용

U-Net

정보를 계층적으로 압축하고(Encoder) 다시 복원(Decoder)하여 [MASK]를 예측합니다.

효율성/표현력 균형 복잡한 구조

SSM / Mamba

$O(n)$

과거 정보를 순차적으로 State에 압축하여 전달하며, 최종 State로 [MASK]를 예측합니다.

선형 복잡도 정보 선택 의존

U-Net이 Loss를 만족시키는 방식

계층적 요약으로 장거리 문맥(Enhancer)을 통합하고, Skip Connection으로 세부 정보를 보존합니다.

Hierarchical Compression

Encoder가 해상도를 점진적으로 줄여(Pooling), 50kb 떨어진 Enhancer 정보를 Bottleneck의 함축적 특징으로 압축합니다.

Skip Connections

압축 과정에서 손실될 수 있는 위치 정보와 세부 패턴을 Decoder로 직접 전달하여 Single-base 해상도를 유지합니다.

Context Integration

Decoder는 Bottleneck의 전역 문맥(Global Context)과 Skip Connection의 지역 정보(Local Detail)를 결합해 정확한 [MASK]를 예측합니다.

SSM이 Loss를 만족시키는 방식

과거 정보를 State로 압축하고, 중요한 정보만 선별적으로 기억하여 [MASK]를 예측합니다.

Selective Memory

중요한 정보는 State에 강하게 남기고, 불필요한 정보는 빠르게 망각하여 메모리를 효율화합니다.

Linear Complexity

입력 길이가 늘어나도 State 크기는 고정(O(1))되어 있어 계산 복잡도가 선형(O(N))입니다.

Predictive Power

압축된 State 정보를 통해 멀리 떨어진 문맥 정보를 복원하여 정확한 [MASK] 예측을 수행합니다.

주요 모델 사례

학습 목표: DNABERT-2, Nucleotide Transformer, HyenaDNA, EVO, Caduceus 등 대표 모델들의 특징과 성능을 비교 분석합니다.

gLM 모델 타임라인 (2020-2025)

2020-21	2022	2023		2024	2025
DNABERT	DNABERT-2	HyenaDNA	EVO	Caduceus	GPN-MSA
6-mer BERT-MLM	BPE 도입 효율성↑	SSM 계열 장문맥	생성/설계 AR 모델	Mamba SSM 131kb	MSA 결합 보존 강화

발전 트렌드

Tokenization: k-mer → BPE → Single-base

Architecture: CNN → Transformer → SSM

Context: 1kb → 12kb → 131kb+

핵심 성과

DNABERT-2: BPE 효율성 혁신

HyenaDNA: 100kb+ 장거리 학습

EVO: 대규모 서열 생성 가능

DNA Language Model 발전의 역사

단어 임베딩의 시작부터 생성형 AI와 멀티모달 통합 모델까지의 여정

EMBEDDING · BERT/ENCODER · LONG-CONTEXT · GENERATIVE/SSM

2017-19 태동기

- dna2vec — Word2Vec 기반 시초
- NeuroGen
- MotifBERT (초기)

2020-21 BERT 시대

- DNABERT — k-mer Masked LM
- BigBird-Genomics — Sparse Attention
- GeneTransformer
- Genome-BERT
- LOGO
- MuLan-Methyl

2022 다양화

- GenSLM — SARS-CoV-2 → General
- GPN — Genomic Pre-trained
- FloraBERT (식물)
- Trans-Omics
- CodonBERT

2023 폭발적 성장 & SCALE-UP

- Nucleotide Trans. — 2.5B Params
- HyenaDNA — 1M Context
- DNABERT-2 — BPE, ALiBi, 효율성↑
- GENA-LM
- ProkBERT
- GPN-MSA
- SpliceBERT
- DARWIN
- Uni-RNA

2024 - 26.01 GENERATIVE & SOTA

- Evo / Evo 2 — 7B, Generative, Arc Inst.
- Caduceus — Bi-Mamba, RC Equiv.
- OmniGenome / Generator — Multi-modal & Generative Focus (NEW)
- DNABERT-3
- GROVER
- gLM2
- HybriDNA
- ATGC-Gen
- RegLM, ViraLM, Gene42...
- Mistral-DNA
- AgroNT / PlantCaduceus
- SegmentNT
- PlantBiMoE
- ConvNova

TOKENIZATION	ARCHITECTURE	CONTEXT LENGTH	KEY TASK
k-mer → BPE → Single-base / VQ	CNN/LSTM → Transformer → SSM / Hybrid	< 512bp → 4kb ~ 32kb → 131kb ~ 1M+	Classification → Interpretation → Generation & Design

DNABERT-2: BPE 기반 효율적 사전학습

핵심 혁신

BPE(Byte Pair Encoding) 도입으로 k-mer 한계 극복

주요 특징

가변 길이 토큰: 효율적 컨텍스트 활용

OOV(Out-of-Vocabulary) 단어 감소

학습/추론 효율성 향상

구체적 사양

Tokenization	Architecture
BPE Byte Pair Encoding	Transformer Encoder-only
Loss	**Data**
MLM Masked Language Modeling	Human + Multi-species 프리트레인

주요 성과

모티프/조절요소 인식 향상

프로모터/엔핸서 분류 성능

↑

소량 라벨로도 효과적 미세조정

⚠ 한계: 초장거리 상호작용(>100kb) 직접 모델링 제한

Nucleotide Transformer (NT)

다종 사전학습과 12kb 긴 컨텍스트

다종 게놈 데이터로 사전학습한 대규모 Transformer 모델

성과: 전이학습 향상

- SegmentNT: 유전자/CRE 주석 SOTA 달성
- AgroNT: 식물 크로마틴/발현 예측 향상
- ⚠ 인간 변이 LLR: 기존 보존 지표 대비 제한적

주요 특징

Tokenization	Architecture
초기 k-mer ↔ v2는 12kb 확장	Transformer + MLM
Data	**Transfer**
Human + 1000 Genomes + Multi-species	다종 전이학습

목표 및 용도

- 범용 임베딩 추출
- 이종 도메인 전이
- 다종 게놈 분석
- 크로마틴/발현 예측

HyenaDNA: SSM 기반 100kb+ 초장거리 문맥 처리

핵심 혁신

State-Space Model (SSM) 기반,
Transformer의 O(n²) 제약을 극복

장거리 의존성 포착

Enhancer-Promoter 상호작용 (수만~수십만 bp)

인트론-엑손, 3D 게놈 구조 간접 반영

준선형 스케일 O(n) → 100kb+ 문맥 처리

성능 비교

100kb+	O(n)	131kb
최대 문맥	복잡도	Fine-tune
엔핸서-프로모터	인트론-엑손	3D 게놈 구조

기술 특징

- Memorization 효율 — RNN 대비 효과적
- 빠른 학습 — 선형 스케일
- Bidirectional — 양방향 처리
- Context Scaling — 100kb+ 확장

EVO: 생성·설계 지향 AR 하이브리드

Architecture	Training	Generation
Hybrid SSM+Transformer	Autoregressive (AR)	Next-token Prediction

모델 특징

CRISPR-Cas 생성

- Cas subtype-specific prompt
- Novel CRISPR systems

대규모 생성

- 650 Mbp sequences
- 20 generated sequences

한계사항

- 자연 게놈과 구성 차이
- 보편 마커 유전자 부족
- 평가 재현성 개선 필요

활용 분야

- 조절서열 설계
- 유전요소 조건부 생성
- 합성 생물학 도구

GPN-MSA

- 단일 염기 + Dilated CNN/Transformer 변형
- MLM + 전장 MSA 결합
- 보존 신호로 LLR 기반 변이 제약 예측 SOTA

강점: 보존 신호 강화, MSA 기반 진화적 제약 활용

한계: 전장 MSA 가용 종/영역 제한

Caduceus (Mamba)

- 단일 염기 + SSM(Mamba) 기반
- 131 kb 문맥 처리, 역상보 등가성
- 초장거리 상호작용, 효율적 시퀀스 처리

강점: 초장거리 상호작용, 선형 스케일, 100kb+ 효율적 처리

한계: 데이터/튜닝 없이는 지역 모티프 해석성 보완 필요

보존 강화: MSA 기반 진화적 제약 활용 ⇄ 초장문맥: SSM으로 131kb+ 효율적 처리

Transformer

대표 모델
- DNABERT-2
- Nucleotide Transformer

특징
- ✓ BPE/k-mer 주로 사용
- ✓ Context: ~1–12 kb
- ✓ Loss: MLM

강점
- ✓ 전이성/효율/생태계 풍부

용도

범용 임베딩, 분류·주석, 미세조정

SSM/Hybrid

대표 모델
- HyenaDNA
- Caduceus

특징
- ✓ 단일 염기
- ✓ Context: 100 kb+
- ✓ Loss: 주로 MLM

강점
- ✓ 장거리 의존성 학습
- ✓ 선형 스케일

용도

장거리 조절, 3D 게놈 신호 간접 반영

생성·보존

대표 모델
- EVO
- GPN-MSA

특징
- + Objective: AR 생성(EVO)
- + 보존 강화 LLR(GPN-MSA)

강점
- ✓ 조건부 생성(EVO)
- ✓ 변이 제약 SOTA(GPN-MSA)

한계

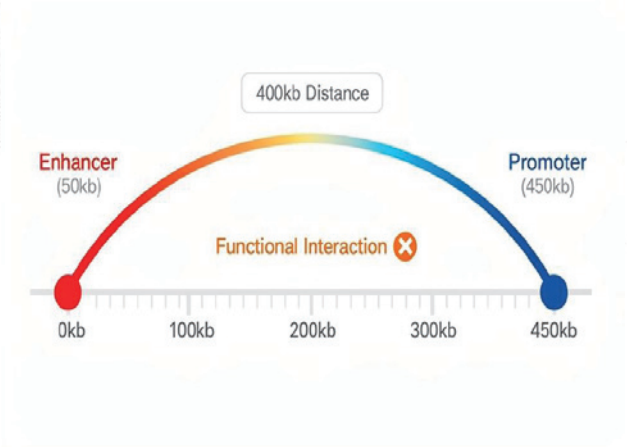
평가 재현성(EVO), MSA 의존성(GPN-MSA)

Long-range Context

학습 목표: Long-range context의 생물학적 필요성을 이해하고, 계층적 구조와 효율적 아키텍처를 통한 장거리 의존성 처리 방법을 학습합니다.

Long-range context란 무엇인가

수 kb에서 Mb에 이르는 넓은 서열 범위를 동시에 고려하며, 단순한 길이가 아닌 기능적 연결(Functional Link)을 통해 멀리 떨어진 유전자 조절 요소들이 상호작용하는 맥락을 의미합니다.



광범위한 스케일 (Scope)

수 kb 모티프부터 Mb 단위 염색질 도메인까지 다양한 스케일의 정보를 통합

기능적 연결 (Functional Link)

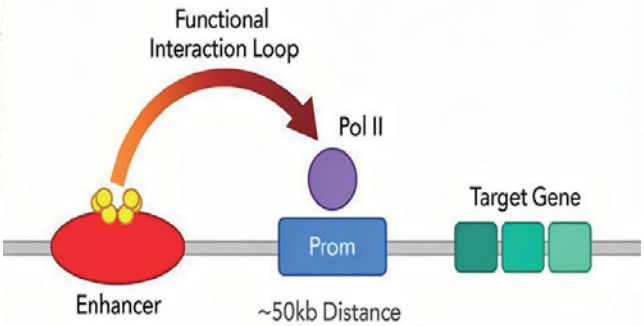
물리적으로 멀어도 3D 공간에서 접촉하여 직접적인 상호작용 루프 형성

생물학적 의미 (Impact)

유전자 발현 정밀 조절, 조직 특이적 활성, 비전사 영역 변이의 질병 연관성 해석

DNA에서 장거리 문맥이 필요한 생물학적 이유

핵심: 유전자 발현은 프로모터 주변 정보만으로 결정되지 않음



장거리 조절 (Long-range Regulation)

Enhancer는 유전자로부터 수십 kb에서 수 Mb 떨어진 위치에 존재하며, 거리에 상관없이 유전자의 스위치 역할을 수행합니다.

Chromatin Loop (3D 구조)

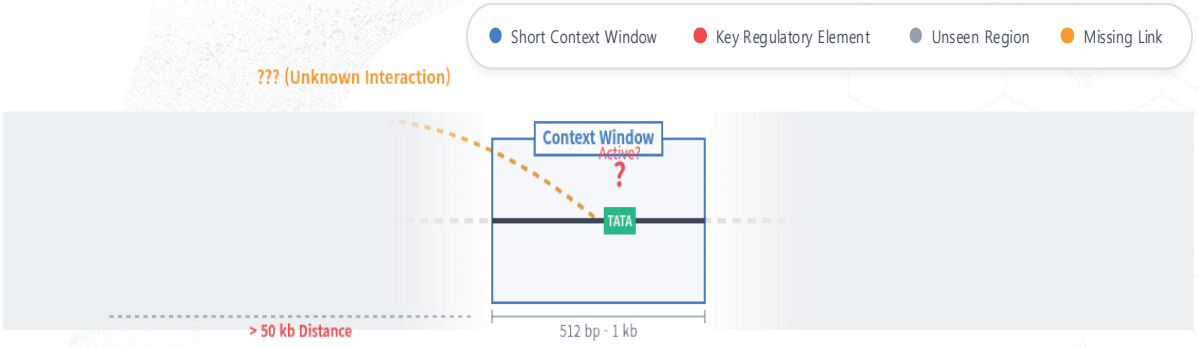
DNA 사슬이 3차원 공간에서 접혀(Looping) 물리적 거리를 단축시킵니다. 이는 선형 거리와 기능적 거리가 다름을 의미합니다.

전사 활성화 (Activation)

Enhancer에 결합한 전사 인자들이 Promoter와 접촉하여 RNA 중합효소를 모집하고 유전자 발현을 시작합니다.

Short context 모델이 놓치는 것

핵심: 국소 패턴(Local Pattern)만 보고 전체 조절 맥락을 놓침



국소 패턴 편향 (Local Bias)

모델이 주변 수백 bp만 학습하여 Motif(단어)는 인식하지만, 그것이 놓인 전체 문맥(문장)은 이해하지 못합니다.

장거리 조절 정보 누락

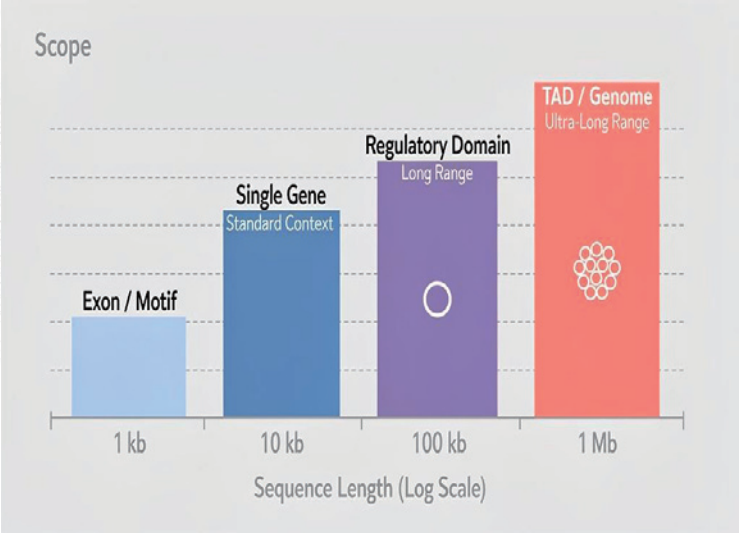
50kb 이상 떨어진 Enhancer나 Silencer의 신호를 받지 못해, 유전자의 실제 활성 상태를 잘못 판단합니다.

예측 실패 사례

TATA box가 있어도 조직 특이적 비활성 상태인 경우를 구별하지 못해 위양성(False Positive) 예측이 발생합니다.

왜 Mb 단위까지 고려해야 하는가

유전자 하나가 수백 kb에 이르는 조절 영역을 가집니다



1kb - 10kb: 기본 단위

전통적인 NLP 모델의 Context Window 수준.

단일 유전자나 국소적인 Motif 패턴(단어)만 파악 가능합니다.

100kb: 조절 상호작용

Enhancer와 Promoter가 만나는 실제 기능적 범위.

이 범위를 보지 못하면 유전자의 발현 여부를 예측할 수 없습니다.

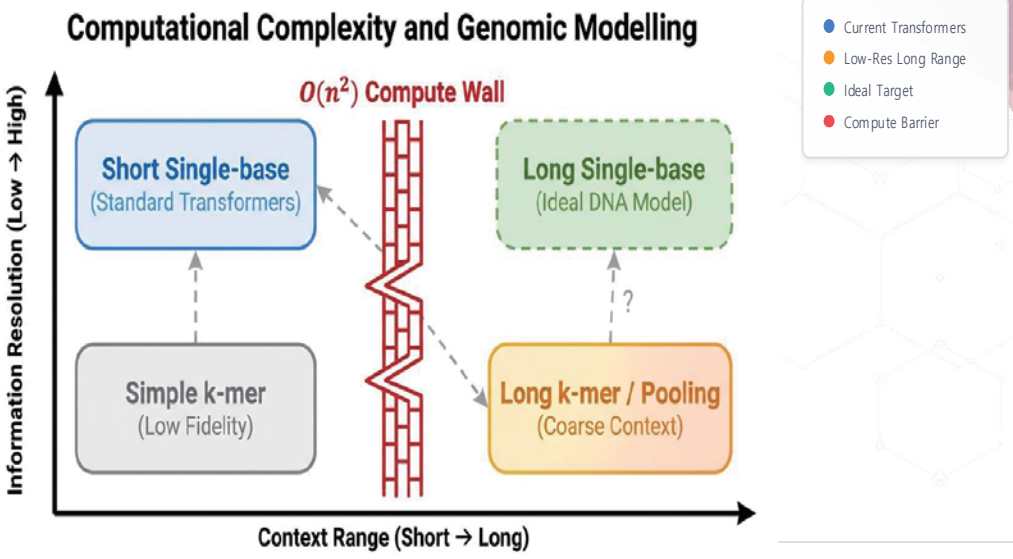
1Mb: 구조적 도메인 (TAD)

염색체가 접혀서 형성하는 3차원 구조적 구획.

전체적인 게놈의 문법과 격리(Insulation) 규칙이 결정되는 스케일입니다.

Long-range context의 핵심 문제

단순 길이 확장이 아닌, '정밀도'와 '범위'의 동시 확보가 난제



정보 폭발 (Information Explosion)

범위를 Mb 단위로 늘리면 토큰 수가 폭발적으로 증가합니다. 단순히 Context Window만 늘리는 것은 불가능합니다.

계산 비용 (Computational Cost)

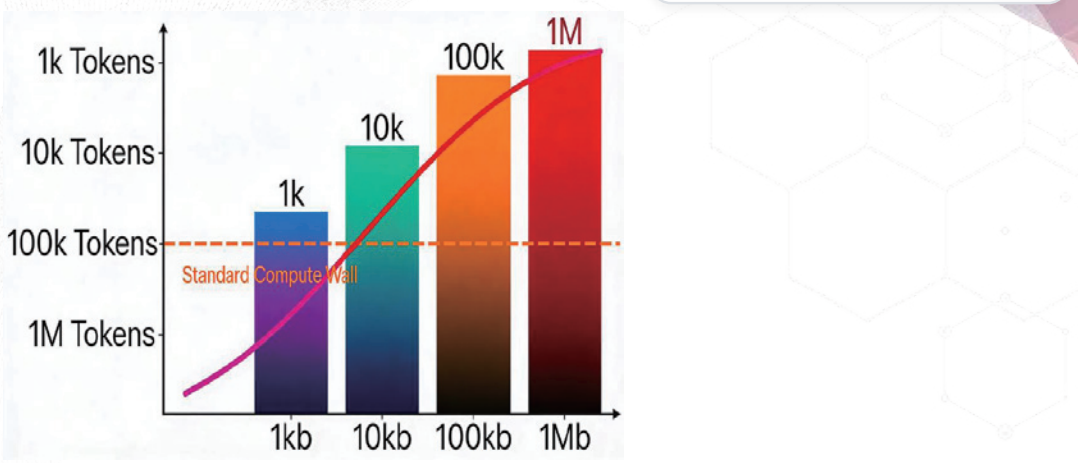
Single-base 해상도를 유지하면서 범위를 넓히면 연산량이 제곱($O(n^2)$)으로 증가하여 하드웨어 한계에 부딪힙니다.

선택적 처리 (Selective Processing)

모든 정보를 다 가져가되, 중요한 정보만 선별하는 새로운 메커니즘(SSM 등)이 필요합니다.

모든 정보를 동일하게 볼 수 없는 이유

핵심: 1Mb 서열 = 100만 토큰 = 계산 불가능한 비용 폭발



100만 개의 토큰

Single-base 해상도로 1Mb를 처리하려면 100만 개의 토큰이 생성됩니다. 이는 일반 언어 모델의 수천 배에 달하는 길이입니다.

O(n²) 연산 비용

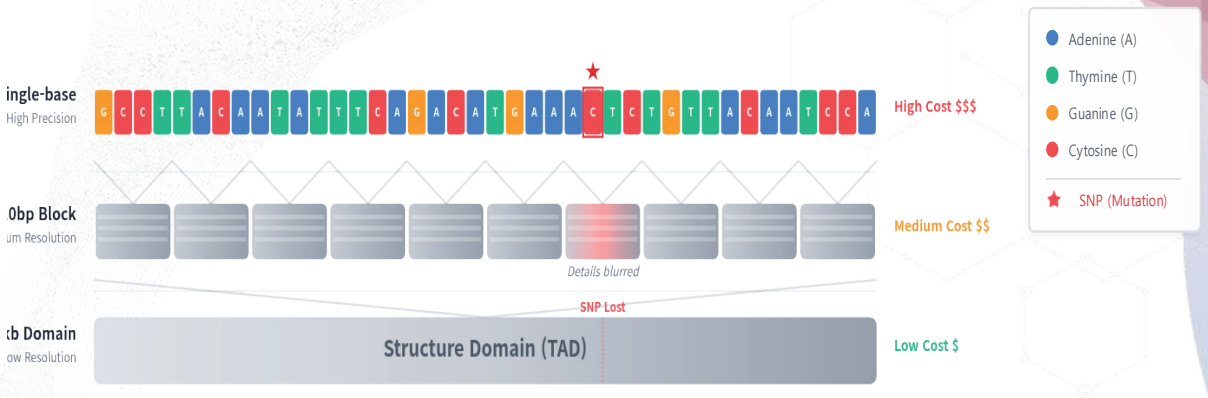
표준 Attention은 토큰 수의 제곱에 비례하여 연산량이 증가합니다. 100만 토큰의 제곱은 1조(10^{12}) 번의 상호작용 계산을 의미합니다.

하드웨어 물리적 한계

100만×100만 크기의 Attention Matrix를 저장하려면 수 TB의 GPU 메모리가 필요하여, 현존하는 하드웨어로는 처리가 불가능합니다.

해상도(Resolution)의 단계별 비교

핵심: 해상도는 정밀도(Precision)와 계산 비용(Cost)의 트레이드오프입니다



Single-base (1bp)

최고 정밀도, 최고 비용. 단 하나의 염기 변이(SNP)도 정확히 포착하지만, 1Mb 처리 시 100만 토큰의 막대한 연산이 필요합니다.

Region Level (10bp)

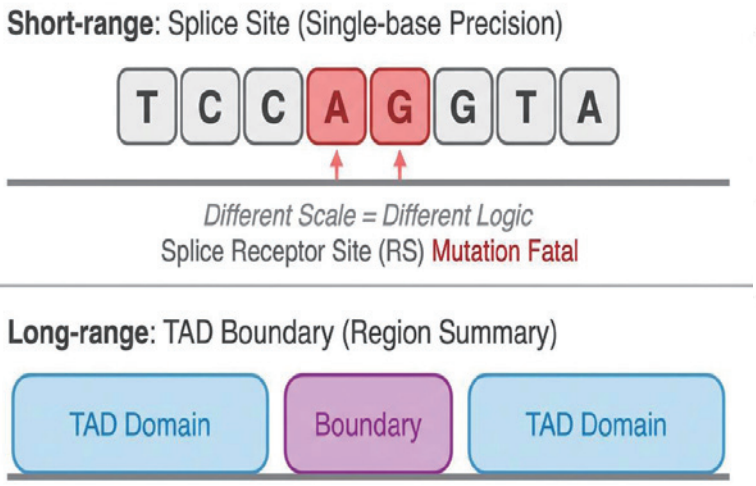
균형점 (Trade-off). 여러 염기를 묶어 처리 효율을 높이지만, 개별 염기 수준의 미세한 돌연변이 정보는 희석됩니다.

Domain Level (1kb)

최저 정밀도, 최고 효율. 전체적인 구조적 특징(TAD 등)은 파악하기 쉽지만, 서열 내의 구체적인 유전 정보는 대부분 소실됩니다.

Hierarchy가 필요한 이유

거리에 따라 중요한 정보의 단위가 달라집니다 (Scale-dependent Features)



Short-range: Precision

Splice site(AG)와 같은 국소적 기능은 단 하나의 염기 차이가 치명적입니다. Single-base 해상도가 필수적입니다.

Long-range: Context

TAD와 같은 거대 구조는 내부 서열보다 경계의 존재 여부가 중요합니다. 지역적 요약(Pooling)이 효율적입니다.

Hierarchy Solution

두 요구사항을 모두 만족하기 위해, U-Net과 같은 계층적 모델은 정밀도와 문맥을 동시에 포착합니다.

왜 계층(Hierarchy)이 필요한가?

해상도 딜레마

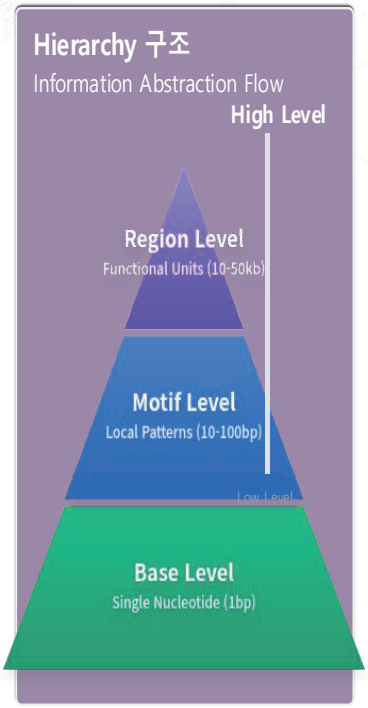
모든 염기를 동일하게 처리하면 계산량이 폭발합니다. 30억 개 염기를 1:1로 보는 것은 불가능에 가깝습니다.

문맥 의존성

같은 'A' 염기도 위치에 따라 의미가 다릅니다. 주변 문맥(Context)이 염기의 기능을 정의합니다.

효율적 압축

중요한 정보(Functional)는 남기고 반복 서열(Redundant)은 요약하여 모델 효율을 극대화합니다.



각 수준의 구체 예시

Top: Region Level

TAD Boundary

10kb~ 단위. 염색질 루프를 형성하여 유전자 발현 영역을 물리적으로 격리

Middle: Motif Level

TATA Box

10~100bp 단위. 전사 개시 위치를 지정하는 핵심 패턴 서열

Bottom: Base Level

SNP (rs123456)

1bp 단위. 단 하나의 염기 차이가 질병 감수성을 결정

Context Window의 정의와 의미

모델이 한 번에 "기억"하고 처리할 수 있는 시야의 범위 (Scope of Attention)

모델의 시야 (Vision)

Context Window는 모델이 동시에 '보고' 상관관계를 계산할 수 있는 최대 범위입니다.

윈도우 밖의 정보는 직접적으로 참조할 수 없습니다.

슬라이딩 메커니즘

전체 유전체(30억 염기)처럼 긴 데이터는 윈도우 단위로 잘라서 처리합니다.

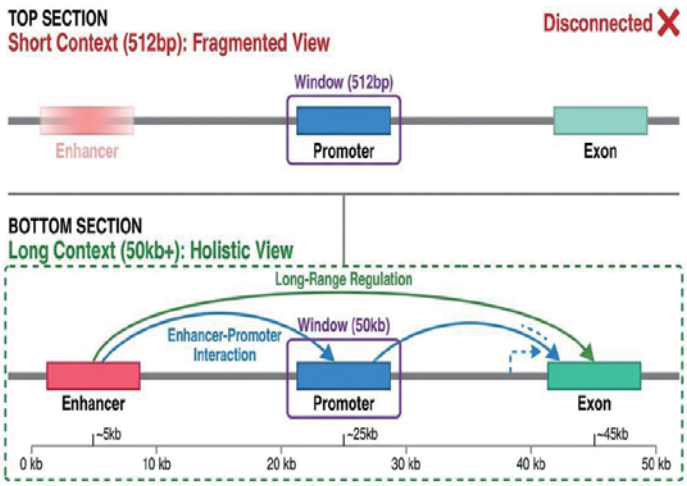
이때 경계면의 정보 손실을 막기 위해 윈도우를 겹쳐서 (Overlap) 이동합니다.

Window Size의 딜레마

윈도우가 작으면(512bp) 장거리 연결을 놓치고,

너무 크면(1Mb) 계산 비용(메모리/시간)이 감당할 수 없을 만큼 커집니다.

짧은 Context의 한계



Partial Gene View

512bp는 유전자 하나의 길이보다 짧습니다. 프로모터만 보고 Downstream Exon과의 연결성을 놓쳐 파편화된 정보만 학습합니다.

Missing Regulation

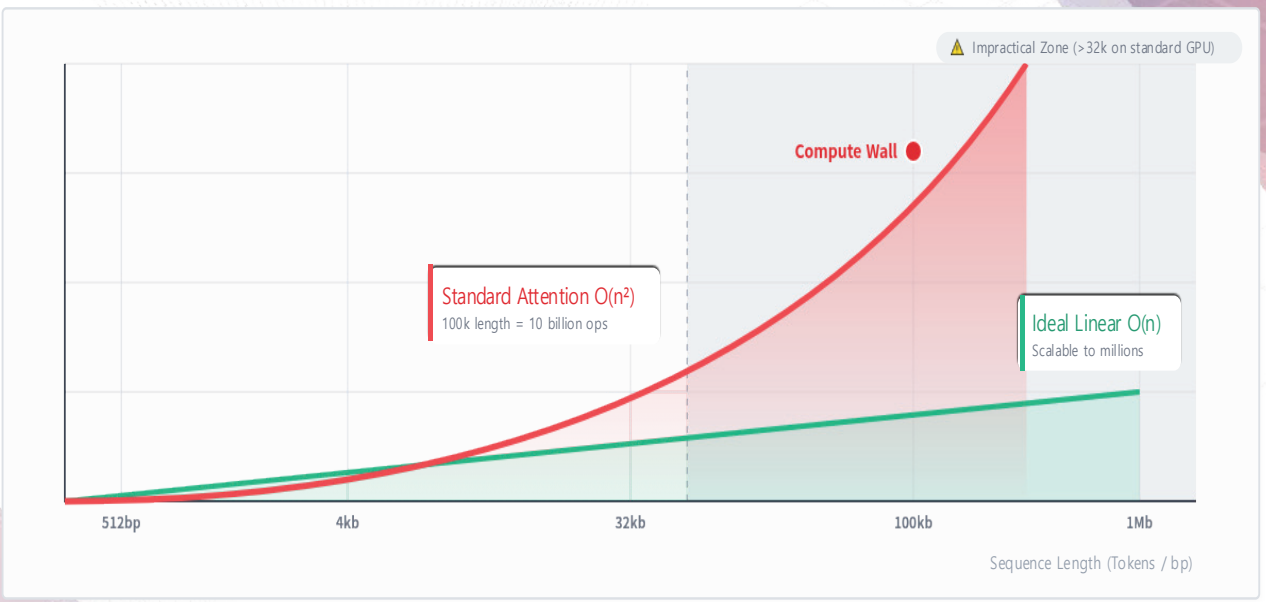
Enhancer가 50kb 떨어져 있을 때, 이를 유전자와 동시에 보지 못합니다. 발현 조절의 핵심 인과관계를 파악할 수 없습니다.

Structural Blindness

수십만~백만 염기 규모의 TAD(위상학적 구조)를 전혀 파악할 수 없어, 3차원적 유전체 상호작용을 놓치게 됩니다.

Context를 확장하는 것의 어려움

시퀀스 길이가 길어질수록 Attention 연산 비용은 기하급수적으로 폭발합니다 (Quadratic Bottleneck).



메모리 폭발 (Memory Explosion)

Attention Matrix 크기는 길이의 제곱(N^2)으로 커집니다.

100k 토큰 처리 시 40GB+ 메모리가 필요하여 일반 GPU에서 OOM 발생.

학습 시간의 기하급수적 증가

입력 길이가 2배 늘어나면 연산량은 4배 증가합니다.

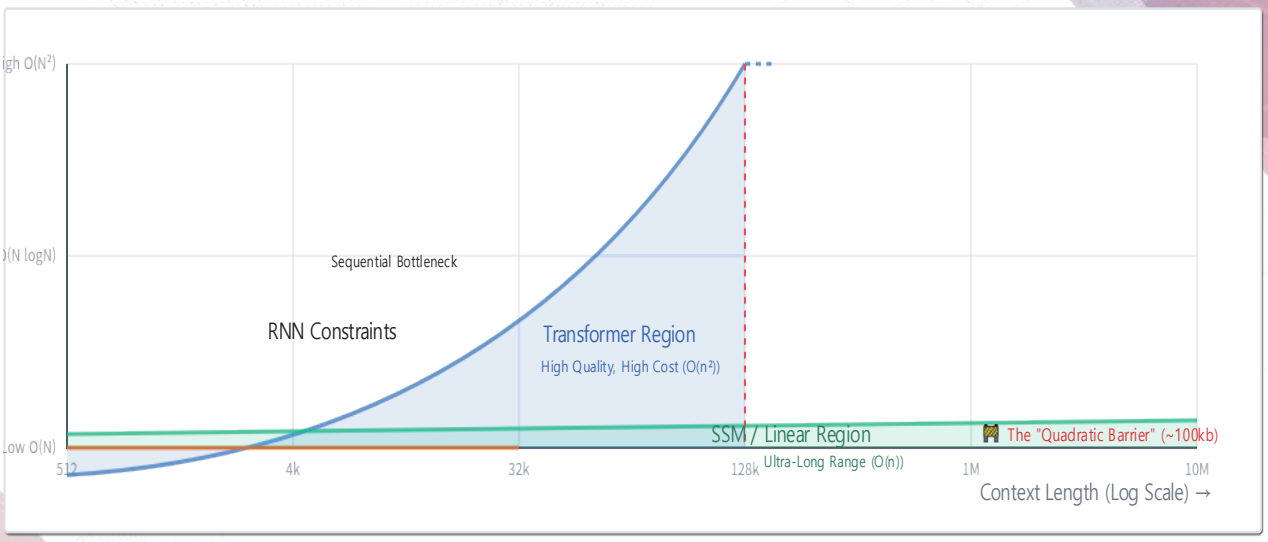
전체 유전체(3B) 학습은 기존 Attention 구조로는 불가능에 가깝습니다.

하드웨어 물리적 한계

최신 H100 GPU조차도 Quadratic Attention의 연산량을 감당하기 어렵습니다. 이는 단순 스펙 업그레이드가 아닌 알고리즘 혁신을 요구합니다.

Context와 모델 설계의 관계

Context window는 단순 하이퍼파라미터가 아닌, 아키텍처 선택을 결정하는 핵심 제약입니다.



Transformer

Context:	Limited (~128k)
Complexity:	Quadratic $O(N^2)$

핵심 한계: 길이가 늘어나면 메모리와 계산량이 폭발하여 DNA 전체(3B) 처리가 불가능

RNN / LSTM

Context:	Short (~1k)
Complexity:	Linear O(N)

핵심 한계: 순차적 처리로 병렬화가 어렵고, 장거리 정보가 희석되는 "망각" 문제 발생

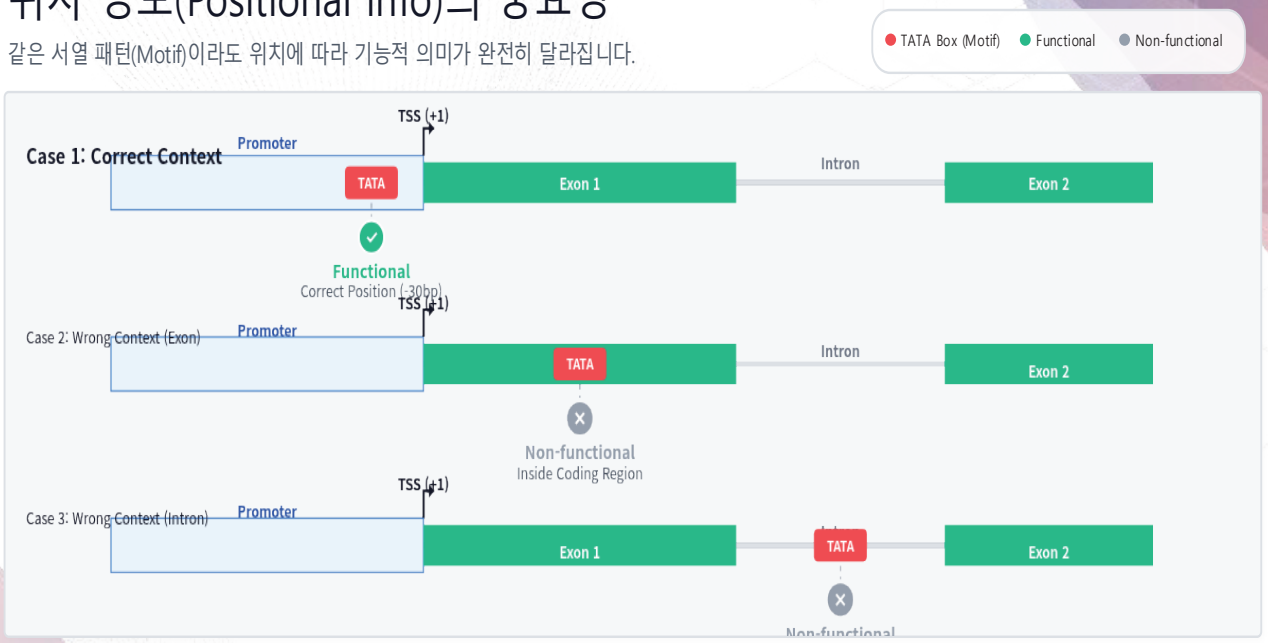
SSM / Mamba

Context:	Ultra-Long (~1M+)
Complexity:	Linear O(N)

핵심 강점: Transformer 수준의 성능을 유지하면서 선형 복잡도로 전체 유전체 처리 가능

위치 정보(Positional Info)의 중요성

같은 서열 패턴(Motif)이라도 위치에 따라 기능적 의미가 완전히 달라집니다.



Case 1: Promoter Region

위치: -30bp (Upstream) TATA box가 전사 시작점(TSS) 앞의 정확한 위치에 있을 때만 RNA Polymerase를 유도하여 유전자를 활성화합니다.

Case 2: Exon Region

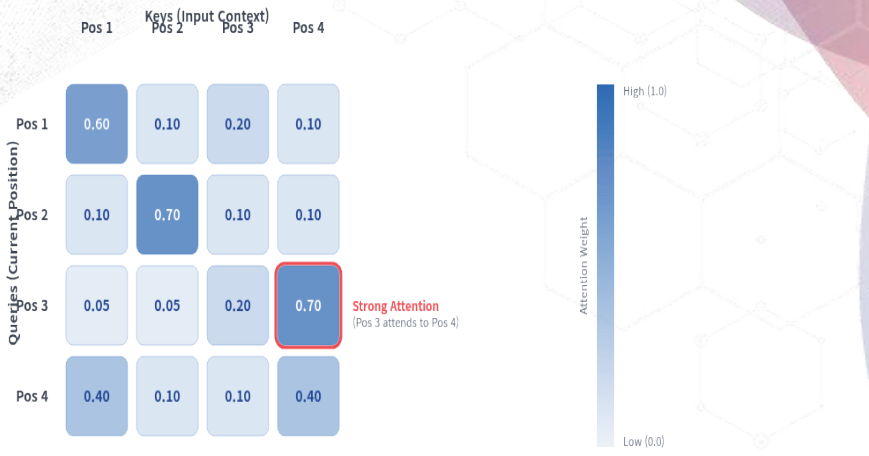
위치: Inside Exon (Coding) 단백질 코딩 영역 내에 동일한 서열(TATAAA)이 나타나면, 이는 조절 신호가 아닌 아미노산(Tyr-Lys...) 서열로 번역될 뿐입니다.

Case 3: Intron Region

위치: Inside Intron 비코딩 영역인 인트론 내에 존재하면, Splicing 과정에서 제거되거나 무시됩니다. "어디에 있는가"가 기능을 결정합니다.

Attention 메커니즘 기초 시각화

모든 위치가 다른 모든 위치와 직접 소통하여(All-to-All) 전역적 맥락을 파악합니다.



1. Query-Key-Value (Q, K, V)

입력 벡터를 세 가지 역할로 변환합니다. Query는 질문을 던지고, Key는 그 질문에 대한 매칭 정보를 담고 있습니다.

2. Score & Softmax

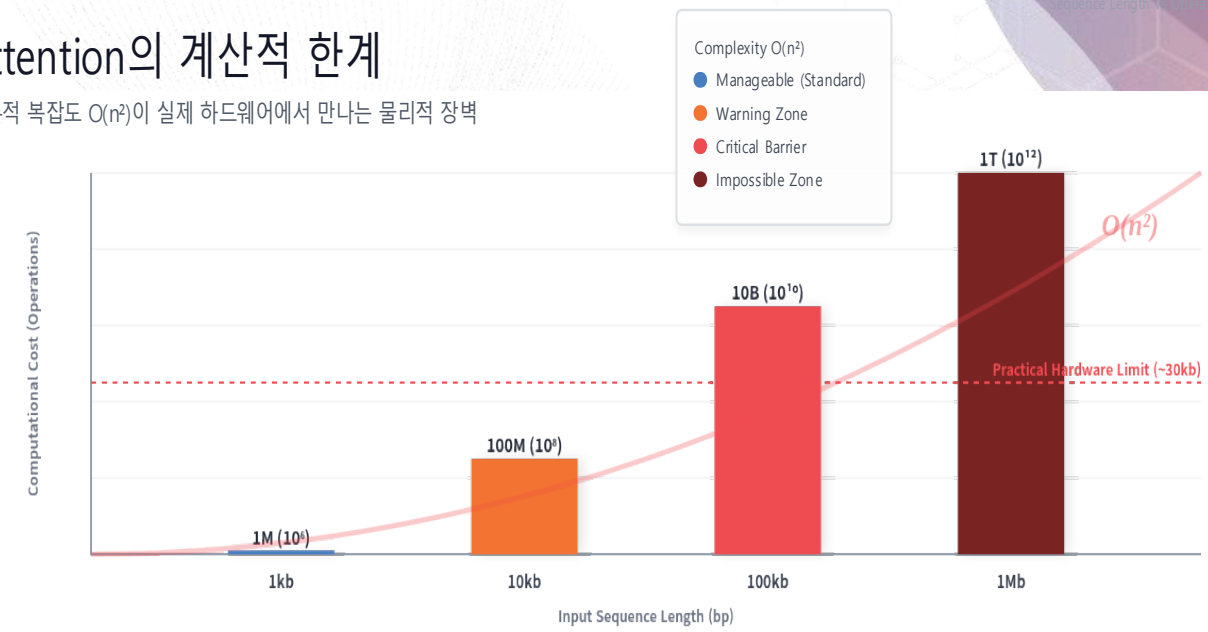
모든 Q와 K의 내적(Dot Product)을 통해 유사도 점수를 계산하고, Softmax 함수로 확률(Attention Weight)로 변환합니다.

3. Weighted Sum (Output)

계산된 확률을 가중치로 사용하여 Value(V)들의 합을 구합니다. 중요한 정보(높은 가중치)만 선택적으로 모아 Output을 만듭니다.

Attention의 계산적 한계

이론적 복잡도 O(n²)이 실제 하드웨어에서 만나는 물리적 장벽



GPU 메모리 폭발

메모리 사용량이 길이의 제곱으로 증가합니다. 100kb 시퀀스 처리 시 수십 GB 메모리 필요하여 일반 GPU에서는 OOM(Out of Memory) 발생.

학습 시간의 비현실성

1Mb(백만 염기) 처리 시 Attention 연산만 1조 번(10¹²) 발생. 단일 스텝 학습에도 엄청난 시간이 소요되어 모델 개발이 불가능합니다.

현실적 타협: Short Context

이러한 한계로 인해 대부분의 초기 DNA 모델은 512~4,096bp로 입력을 제한했습니다. 결과적으로 생물학적으로 중요한 장거리 상호작용을 놓치게 됩니다.

U-Net 접근의 출발점

⚠ 문제 정의

O(n²) 복잡도

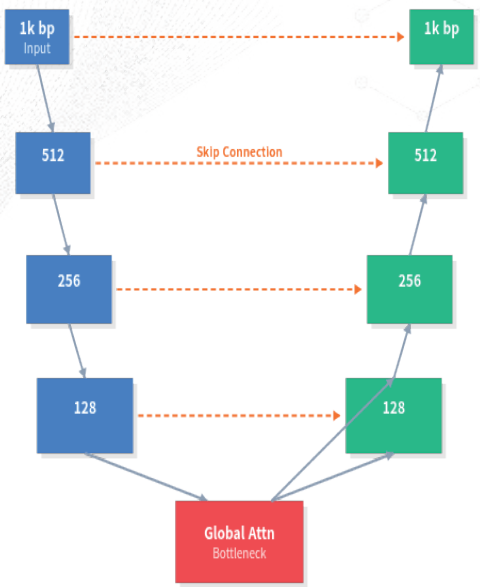
Attention의 계산적 한계

1Mb(백만) 염기를 처리하려면 1조 번의 연산이 필요합니다. 표준 Transformer로는 메모리와 계산량이 감당 불가능합니다.

🚫 100% 해상도 유지 불가

모든 위치를 처음부터 끝까지 고해상도로 유지하며 긴 문맥을 보는 것은 비효율적입니다.

INPUT Sequence — U-Net Architecture — OUTPUT Prediction



핵심 메커니즘

계층적 처리

Encoder

정보를 단계적으로 압축합니다. Local → Global 해상도는 낮아지지만, 수용 영역(Receptive Field)은 넓어집니다.

Bottleneck Attention

Center

가장 압축된 상태에서 Global Attention을 수행합니다. 1/1000 연산량으로 전체 문맥을 효율적으로 파악합니다.

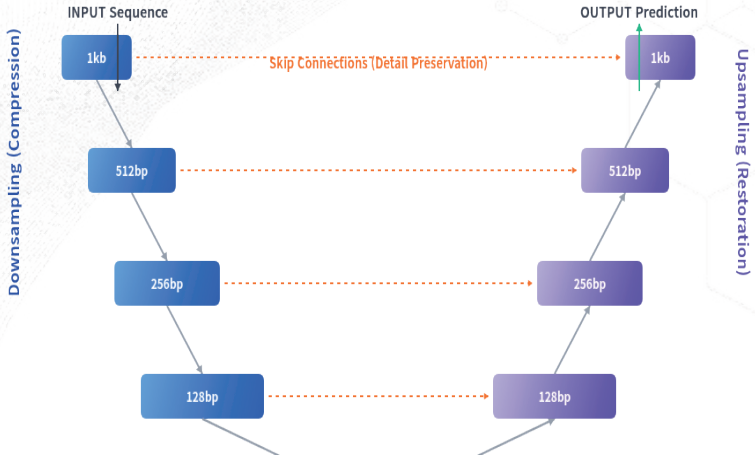
세밀한 복원

Decoder

Skip Connection을 통해 손실된 위치 정보를 복구하여 Single-base 해상도를 회복합니다.

U-Net 구조의 동작 방식

계층적 압축(Encoder)과 복원(Decoder)으로 효율과 정밀도를 동시에 확보합니다.



1. Encoder (Compression)

입력을 단계적으로 압축(Pooling)하여 해상도를 줄입니다. 넓은 수용 영역(Receptive Field)을 확보하여 전역적인 패턴을 학습합니다.

2. Bottleneck (Integration)

가장 압축된 상태에서 장거리 문맥(Long-range Context)을 통합합니다. 1Mb의 거대한 정보가 수천 개의 벡터로 요약되어 효율적으로 처리됩니다.

3. Decoder (Restoration)

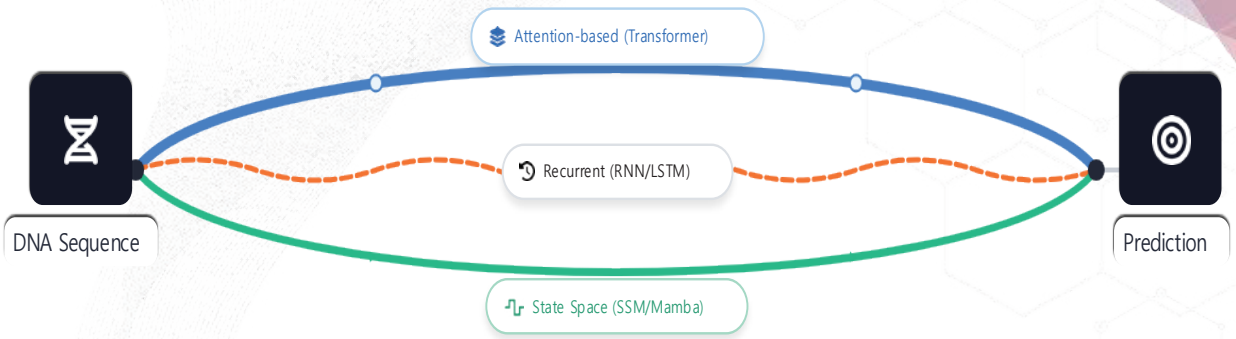
압축된 정보를 Skip Connection과 결합하여 원래 해상도로 복원합니다. 손실된 세부 위치 정보를 되살려 Single-base 정밀도를 유지합니다.

아키텍처

학습 목표: Transformer, U-Net, SSM(Mamba, Hyena) 등 다양한 아키텍처의 작동 원리와 장단점을 비교하고, 태스크에 맞는 선택 기준을 이해합니다.

아키텍처의 역할

입력은 같지만, 목표에 도달하는 경로는 다릅니다 (Same Goal, Different Paths)



Transformer
Attention Is All You Need

복잡도	Quadratic O(n²)
강점	최고의 표현력, 병렬화
약점	긴 문맥에서 비용 폭발

RNN / LSTM
Sequential Processing

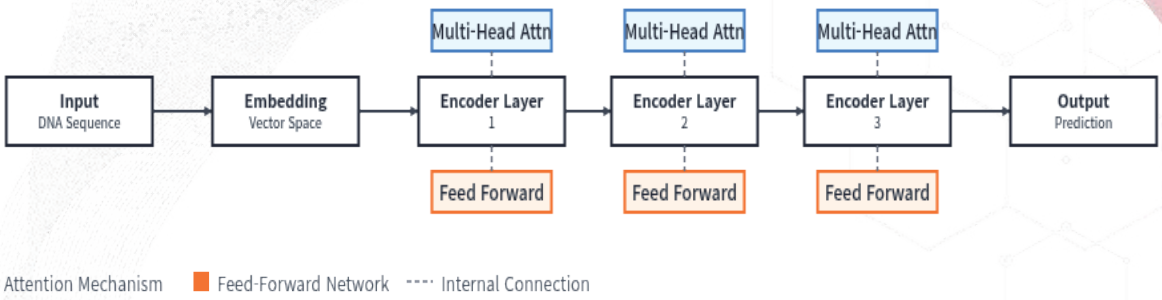
복잡도	Linear O(n)
강점	효율적 추론, 무한 길이
약점	병렬화 불가, 장기 기억 소실

SSM / Mamba
Selective State Space

복잡도	Linear O(n)
강점	병렬 학습 + 빠른 추론
약점	상태 공간 제약, 구현 난이도

Transformer Architecture

Multi-layer Encoder-Decoder 구조



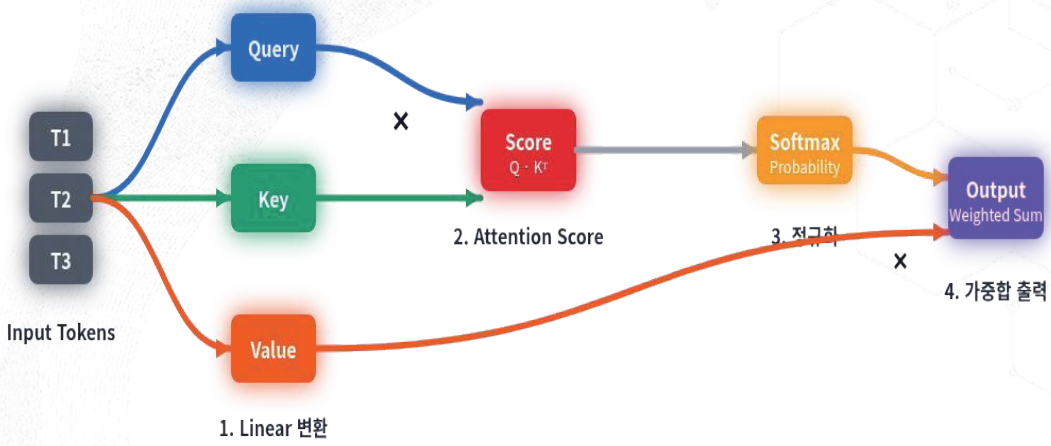
계층적 구조
N개의 동일한 Encoder Layer를 쌓아 구성. 각 Layer는 Attention과 FFN을 포함하며, 잔차 연결로 깊은 학습 가능

Self-Attention 핵심
모든 토큰 간 상호작용을 동시에 계산. Query-Key-Value 메커니즘으로 문맥 정보를 직접적으로 포착

O(n²) 복잡도
모든 토큰 쌍을 계산하므로 입력 길이에 따라 계산량이 제곱으로 증가. 긴 시퀀스에서 메모리/시간 문제 발생

Self-Attention 메커니즘

Q-K-V 연산으로 모든 토큰 간 관계를 계산



Query-Key-Value 변환

입력 토큰(T)을 Query, Key, Value 세 가지 벡터로 변환합니다. Query는 질문, Key는 인덱스, Value는 실제 컨텐츠 역할을 수행합니다.

Score 계산 및 정규화

Query와 Key의 내적(Dot Product)으로 연관성을 계산하고, Softmax 함수를 통해 확률값(0~1)으로 변환하여 '어디에 집중할지' 결정합니다.

최종 Context 도출

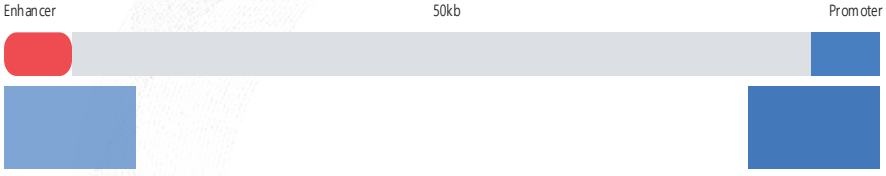
계산된 Attention 가중치를 Value 벡터에 곱해 합칩니다. 이로써 문맥 정보가 반영된 새로운 토큰 표현(Representation)이 완성됩니다.

Attention의 생물학적 의미

Splice Site 경계 학습

Attention이 Exon-Intron 경계에 집중

Enhancer-Promoter 장거리 상호작용



Attention이 50kb 떨어진 조절 요소 간 연결 학습

Attention이 DNA의 기능적 패턴을 자동 학습

3D 게놈 구조의 반영 가능성

Splice site 학습

Exon-Intron 경계 인식, 정확한 접합 위치 예측

Enhancer 인식

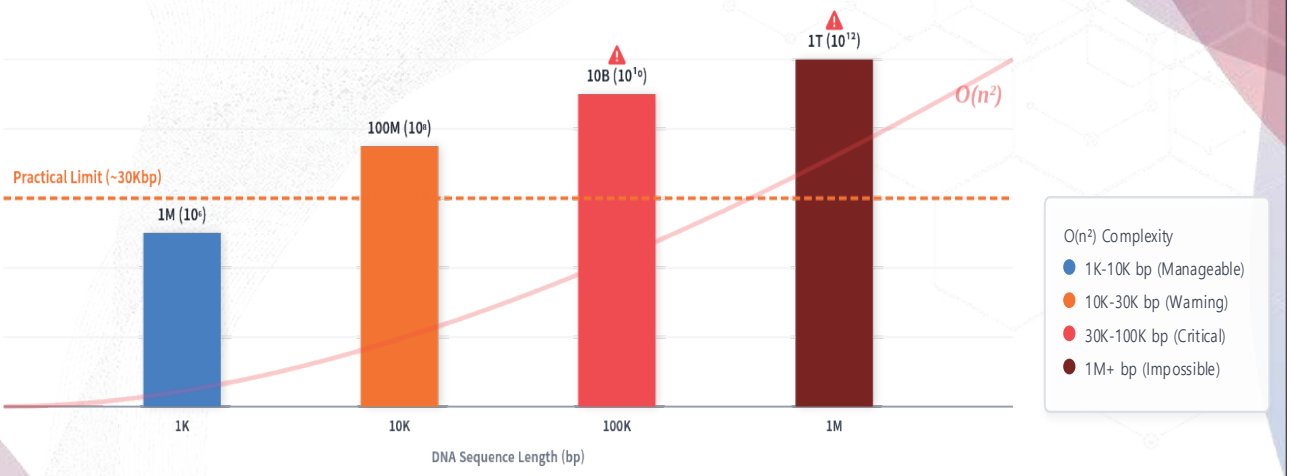
장거리 조절 요소의 기능적 중요도 파악

기능적 맥락 파악

DNA의 3D 구조와 기능적 연관성 학습

Attention의 계산적 한계 - Transformer

$O(n^2)$ 계산 복잡도가 만드는 현실적 장벽



GPU 메모리 폭발

Attention 계산 시 메모리 사용량이 길이의 제곱으로 증가. 100Kbp 처리 시 수십 GB 메모리 필요, 일반 GPU로는 불가능.

학습 시간 비현실성

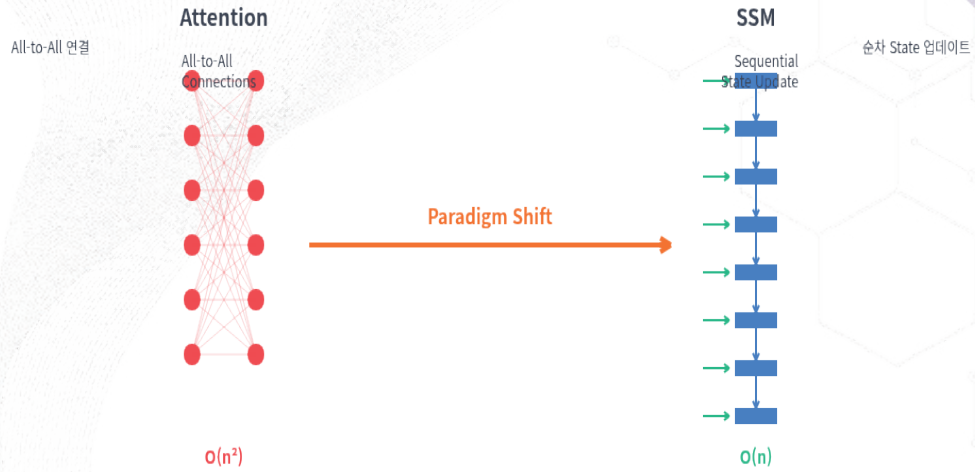
1Mbp(백만 염기) 처리 시 1조 번(10^{12}) 연산 발생. 단일 스텝 학습에도 수십시간 소요, 모델 개발 불가능.

현실적 타협: Short Context

이러한 한계로 인해 대부분의 DNA 모델은 512-4Kbp로 제한. 결과적으로 장거리 상호작용(50kb+)을 놓치게 됩니다.

SSM의 등장 배경

Attention의 계산 비용을 근본적으로 회피하는 새로운 접근



복잡도 차이

Attention은 $O(n^2)$로 증가하지만, SSM은 O(n) 선형 복잡도를 유지. 1Mbp 처리 시 1조→100만 연산으로 감소.

정보 처리 방식

Attention은 모든 위치를 동시에 비교, SSM은 순차적 State 업데이트. 흐름 중심 vs 상태 중심 접근.

제어 이론 기원

SSM은 제어 이론과 신호 처리에서 유래. 연속 시간 시스템을 이산 시간으로 변환하는 수학적 framework.

SSM의 정보 처리 방식

SSM은 정보를 수직적인 시간의 흐름(Time-flow) 속에서 처리합니다. 중요 정보는 State Vector를 확장하여 보존하고, 노이즈는 흘려보냅니다.

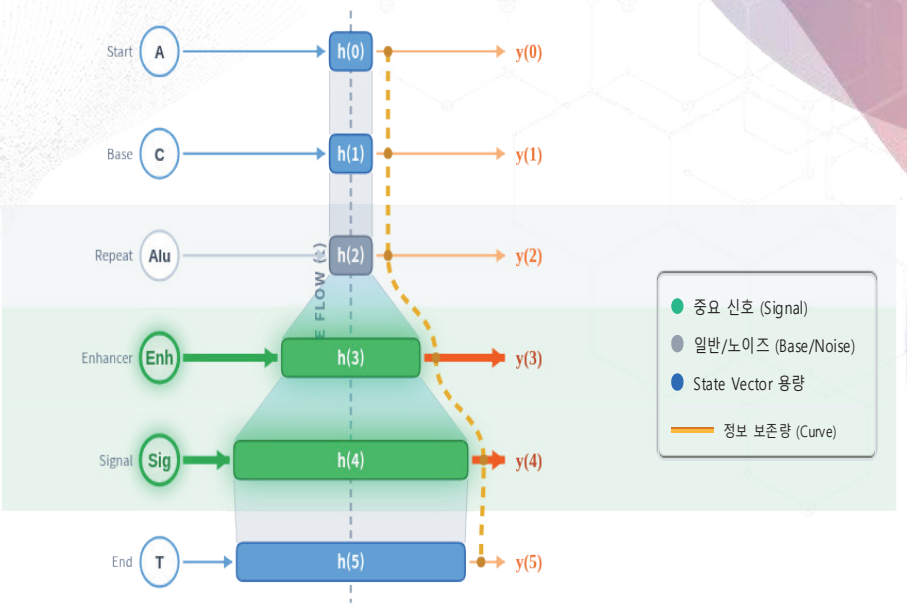
$$h(t+1) = Ah(t) + Bx(t)$$

$$y(t) = Ch(t)$$

상태 $h(t)$는 과거의 요약본입니다.

행렬 A가 이전 기억 유지율을 결정합니다.

- 순차적 처리 (Sequential)
- 상태 압축 (State Compression)



State 메커니즘

과거의 모든 문맥을 하나의 고정된 크기 벡터 h(t)에 압축하여 전달합니다.

선형 효율성

입력 길이에 비례하는 O(n) 복잡도로 긴 시퀀스를 효율적으로 처리합니다.

정보 선별

중요한 신호(Enhancer)는 강하게, 반복 서열(Alu)은 약하게 기억합니다.

SSM과 DNA의 궁합: 선택적 기억(Selective Memory)

DNA 구조적 특징

DNA는 정보 밀도가 불균일합니다. 90% 이상의 비기능성 서열과 흩어진 핵심 신호가 섞여 있습니다.

반복 서열 (Repeats)

Alu, LINE 등 (~45%)

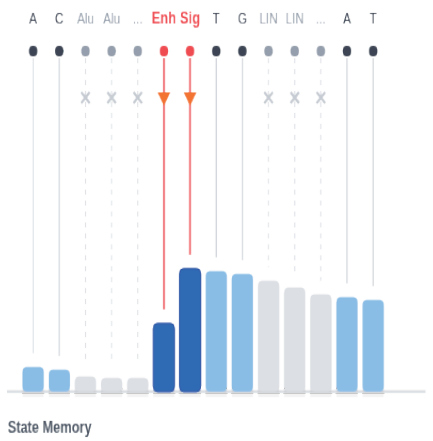
단순 반복 패턴으로 정보량이 낮음. 모델이 굳이 오래 기억할 필요가 없는 노이즈에 가깝습니다.

기능 신호 (Signals)

Promoter, Enhancer

유전자 발현을 조절하는 핵심 스위치. 수십만 염기 떨어져 있어도 반드시 기억해야 합니다.

SSM State Vector의 동적 업데이트



"Gating 메커니즘이 중요도는 높이고, 노이즈는 차단합니다"

SSM의 핵심 강점

자동 중요도 판별

입력 내용에 따라 실시간으로 A, B, C 파라미터를 조정하여 정보 가치를 평가합니다.

효율적 메모리 압축

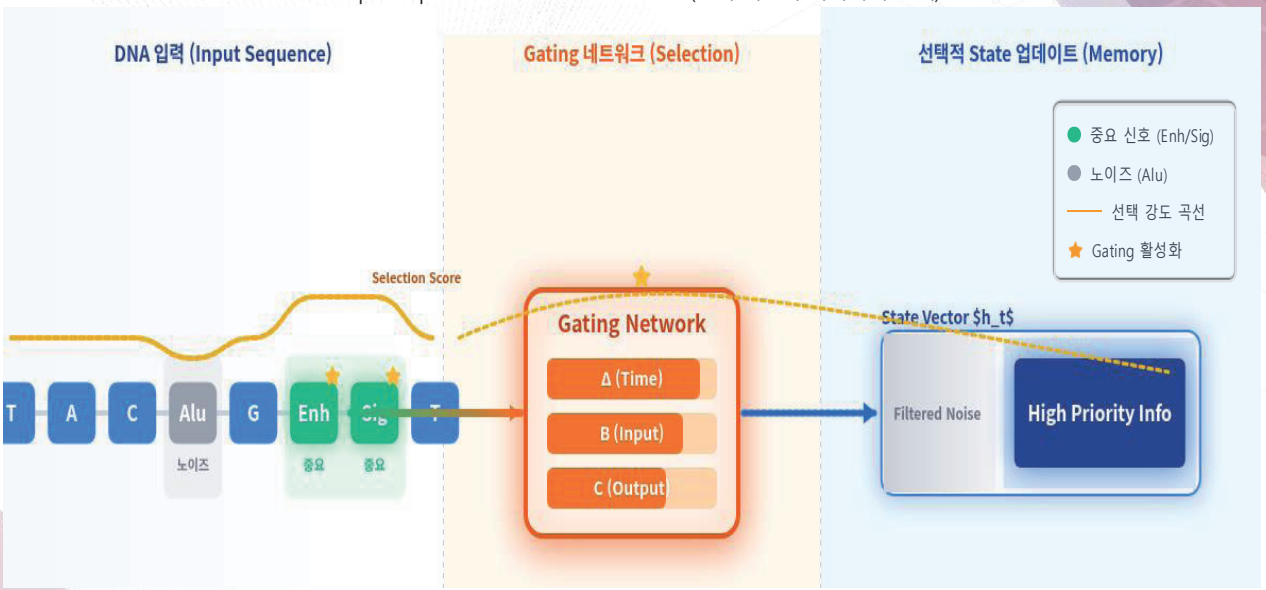
불필요한 반복 서열은 State에 기록하지 않고 흘려보내 저장 공간을 절약합니다.

초장거리 문맥 유지

중요 신호(Enhancer)는 감쇠 없이 끝까지 보존되어 1Mb 거리에서도 복원 가능합니다.

Mamba: Selective SSM의 혁신

Input-dependent Parameter Selection (입력 의존적 파라미터 선택)



Input-dependent

입력 서열(x_t)에 따라 파라미터($B, C, ₩Delta$)를 매 스텝 동적으로 생성

Selective Memory

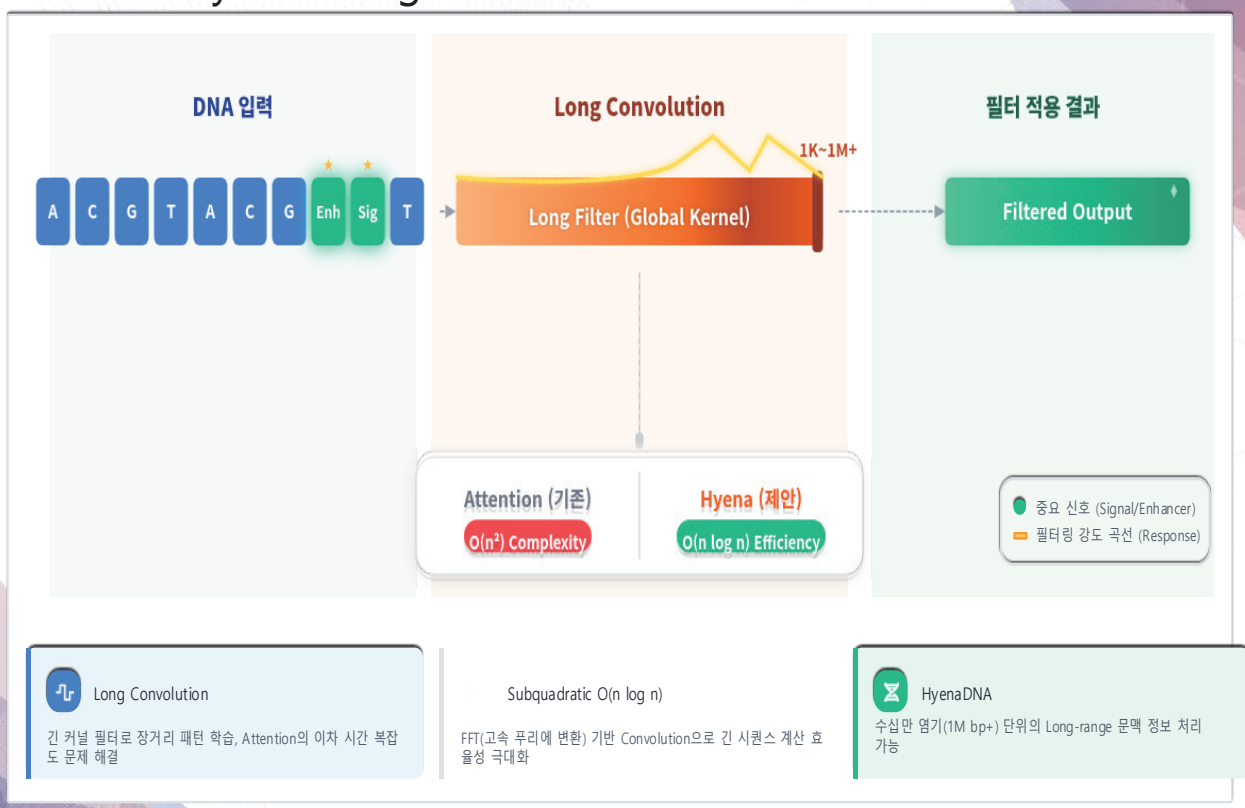
$₩Delta$(Step Size)를 조절하여 중요 정보는 오래 기억하고 노이즈는 망각

Hardware-aware

GPU SRAM을 활용한 Kernel Fusion으로 Recurrent 연산 속도 극대화

Hyena: Long Convolution 접근

Implicit Convolution for Long-range Dependencies



Long Convolution

긴 커널 필터로 장거리 패턴 학습, Attention의 이차 시간 복잡도 문제 해결

Subquadratic O(n log n)

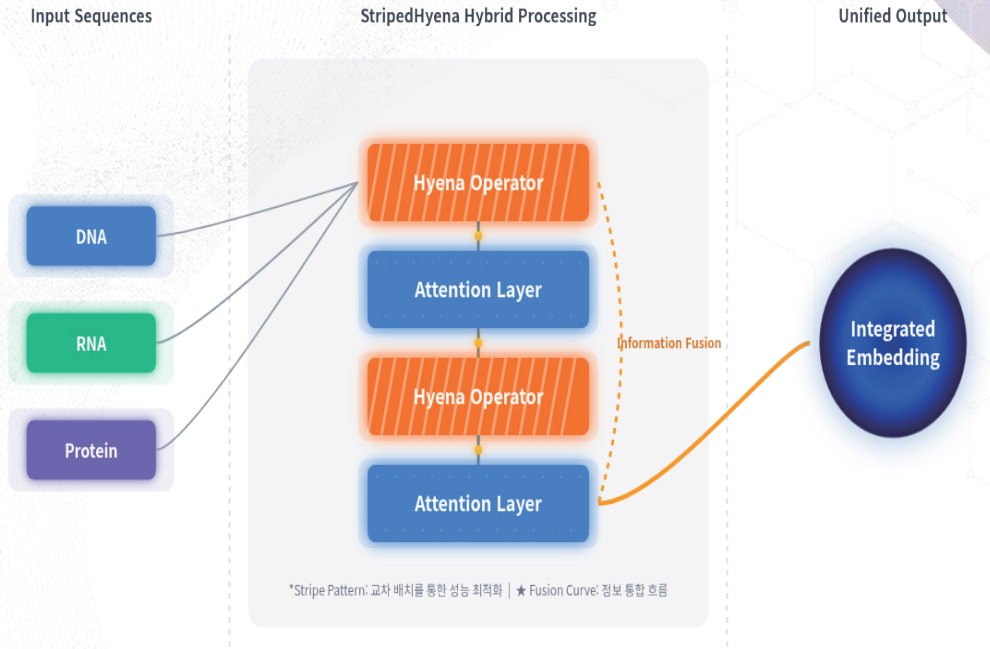
FFT(고속 푸리에 변환) 기반 Convolution으로 긴 시퀀스 계산 효율성 극대화

HyenaDNA

수십만 염기(1M bp+) 단위의 Long-range 문맥 정보 처리 가능

Evo: StripedHyena Hybrid Architecture

DNA·RNA·Protein 통합 처리를 위한 하이브리드 구조



Hybrid Strategy

Hyena Layer와 Attention Layer를 교대 배치하여 장단점 극대화

Multimodal Integration

DNA, RNA, 단백질을 하나의 임베딩 공간으로 통합 처리

131kb Ultra-long Context

최대 131kb의 시퀀스를 단일 모델로 처리하는 확장성

Caduceus DNA 특화 설계

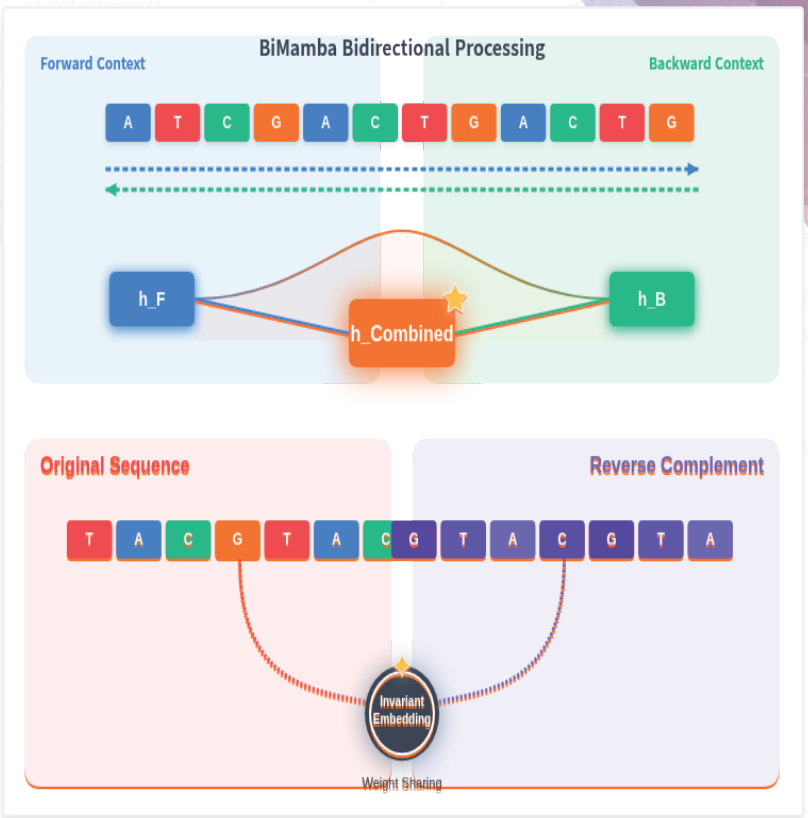
DNA의 고유한 특성을 구조에 반영한 모델입니다. BiMamba를 사용하여 양방향 문맥을 활용하고, RC Equivariance를 구조적으로 보장하여 학습 효율을 극대화합니다.

BiMamba

Forward 및 Backward 방향의 문맥을 동시에 학습하여 정보 손실 최소화 및 융합

RC Equivariance

DNA 이중나선 구조의 상보적 대칭성을 모델 아키텍처에 내재화 (Weight Sharing)



Attention, U-Net, SSM 구조적 차이

세 가지 아키텍처는 근본적으로 다른 철학을 가집니다. Attention은 모든 위치를 동등하게 보려 하며, U-Net은 계층적으로 요약하고, SSM은 상태로 정보를 전달합니다.

핵심 포인트

Attention: 높은 표현력이나 계산 비용 큼

U-Net: 다중 해상도로 효율성과 정밀도 균형

SSM: 선형 복잡도로 긴 시퀀스 처리에 최적

비교 항목	Attention	U-Net	SSM/Mamba
계산 복잡도	$O(n^2)$ 길이 제곱 비례	Multi-scale 계층적 감소	$O(n)$ 선형 비례
장거리 처리	직접 참조 모든 위치 쌍 연결	계층적 요약 Down/Up sampling	순차적 상태 과거 정보 요약 전달
메모리 사용	매우 높음 Attention Matrix 저장	중간 Feature Map 저장	낮음 고정 크기 State
주요 특징	높은 표현력,전역 문맥 파악	위치 정보 보존,다중 해상도	효율적 추론,무제한 길이

아키텍처 선택의 철학

아키텍처 선택은 기술적 결정이 아닌,데이터를 바라보는 관점의 차이입니다.

"모든 위치를 동등하게 볼 것인가?" "계층적 구조가 필요한가?" "순차적 처리가 적합한가?"

	Attention	U-Net	SSM
모든 위치동등?	✓ YES	✗ NO	✗ NO
계층적압축?	✗ NO	✓ YES	PARTIAL
순차적처리?	✗ NO	✗ NO	✓ YES

정보 처리 철학

Attention은 전체를 한 번에 보고, SSM은 필요한 것만 선택적으로 기억합니다.

계산 효율 우선순위

$O(n^2)$의 한계를 극복하기 위해 $O(n \log n)$을 거쳐 $O(n)$ 선형 복잡도로 진화했습니다.

생물학적 가정

DNA 서열의 본질적 특성(조장거리 문맥, 순차성)을 반영한 구조적 선택입니다.

Hybrid 접근의 등장

Hyena + Attention

대부분은 효율적인 Hyena로 처리하고, 핵심 정보 통합만 Attention을 사용하여 O(N log N) 효율성 달성

Mamba + Attention

교대 배치(Interleaving)로 SSM의 상태 유지력과 Attention의 정밀도 균형 확보

U-Net + SSM

멀티해상도 계층 구조 내부에 선형 처리 메커니즘 통합

아키텍처와 생물학적 가정

아키텍처는 DNA의 본질적 특성에 대한 가정을 반영합니다

Attention 가정	U-Net 가정	SSM 가정
"All-to-All Relevance"	"Scale-dependent Importance"	"Signal in Noise"
모든 위치가 잠재적으로 관련	정보가 계층적으로 조직됨	중요한 정보만 선택적으로 유지
균일성 가정	계층적 가정	선택적 가정

아키텍처 발전 방향

DNA 언어 모델 아키텍처의 미래 발전 로드맵: Context 확장과 계산 효율성의 동시 달성

아키텍처 파트 핵심 정리

Transformer 한계 극복

O(n²) 계산 복잡도를 해결하기 위한 새로운 아키텍처 탐색

U-Net/SSM/Mamba 등장

계층적 요약, 순차 상태, 선택적 기억으로 효율적 처리

Hybrid 접근

여러 아키텍처의 장점을 결합한 혁신적 구조

생물학적 가정 반영

DNA의 특성을 반영한 구조적 설계 선택

핵심 과제: O(n²) 계산 복잡도 극복

해결 방향: 다양한 아키텍처 활용

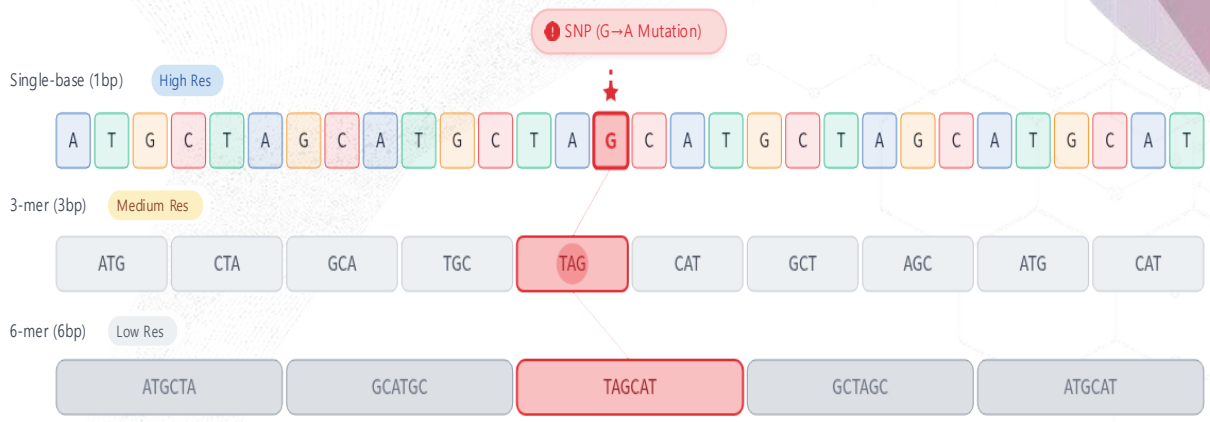
미래 전망: Hybrid + Ultra-long Context

Single-base 해상도

학습 목표: Single-base 해상도의 중요성과 Long-range context와의 트레이드오프를 이해하고, 두 목표를 동시에 달성하는 최신 모델의 전략을 학습합니다.

Single-base 해상도란 무엇인가

염기 하나가 하나의 결정 단위 (Maximum Precision)



Single-base

Resolution: 1 bp (Maximum)

Cost: High

모든 단일 염기 변이(SNP)를 정확하게 포착. 생물학적으로 가장 정밀한 정보를 제공하지만 계산 비용이 높음.

k-mer (3~6bp)

Resolution: 3-6 bp

Cost: Medium

주변 문맥을 포함하지만, 개별 염기의 위치 정보가 흐려짐(blurring). 변이의 정확한 위치 특정이 어려울 수 있음.

Summary / Patch

Resolution: >10 bp

Cost: Low

높은 압축률로 긴 서열 처리에 유리하나, 미세한 유전적 차이(Point mutation)를 놓칠 위험이 가장 큼.

SNP 영향: 단일 염기 변이의 중요성

정상 서열

Splice donor site

A G G T A G T A A G T

Exon → | → Intron

✓ 정상 기능

돌연변이 서열

Splice donor site

A G G T A A T A A G T

Exon → | → Intron

✕ 기능 상실

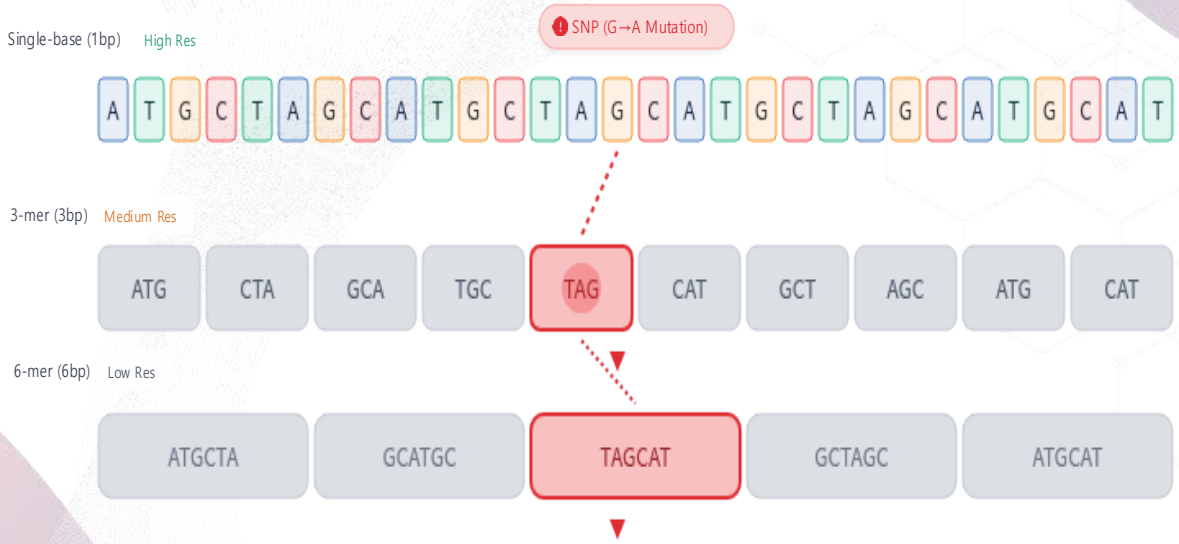
SNP 한 개로 Splice site 파괴

유전 질환 유발 가능

Single-base 해상도 필수

k-mer 해상도의 구조적 한계

k-mer는 정보를 묶어 처리하면서 위치 정보를 흐리게 만듭니다



Single-base

염기 단위로 정확한 위치 파악 가능. SNP 정확히 포착. 생물학적 정밀도 최대.

3-mer

토큰 내부에서 SNP 위치 모호. 경계 불명확. 변이 탐지 정확도 저하.

6-mer

정보가 크게 뭉개짐. SNP 위치 완전히 상실. 생물학적 의미 해석 어려움.

해상도를 낮춘다는 것의 의미

정보 요약은 항상 손실을 동반합니다

고해상도 DNA

Full Information

COMPRESSION / SUMMARIZATION

저해상도 DNA

Lost Details

세부 구조 손실

개별 염기의 정확한 위치와 순차적 정보가 사라집니다. 정밀한 변이 분석이 불가능해집니다.

SNP/Motif 경계 모호화

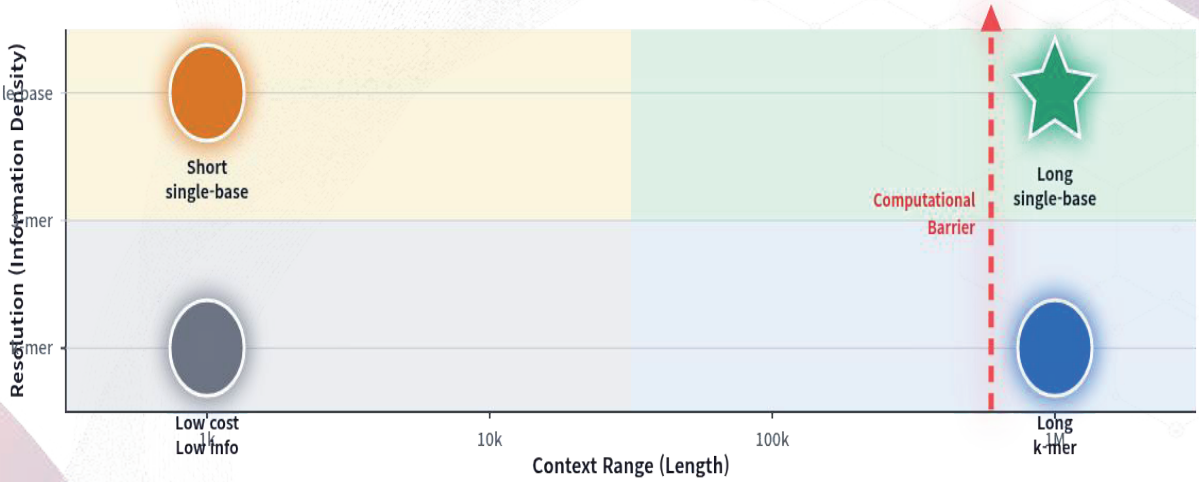
k-mer 내부의 변이는 토큰 단위로 묶여 구분이 어렵습니다. 경계를 넘나드는 패턴은 흐려집니다.

생물학적 의미 누락

단백질 결합 부위, 스플라이스 신호 등 정밀한 기능적 요소들이 토큰 단위로 묶여 인식이 어려워집니다.

Long-range vs Single-base 충돌

두 목표의 동시 달성이 어려운 이유



토큰 수 폭발

1Mb × single-base = 100만 토큰. Attention의 $O(n^2)$ 복잡도로 계산 불가능

$O(n^2)$ 복잡도

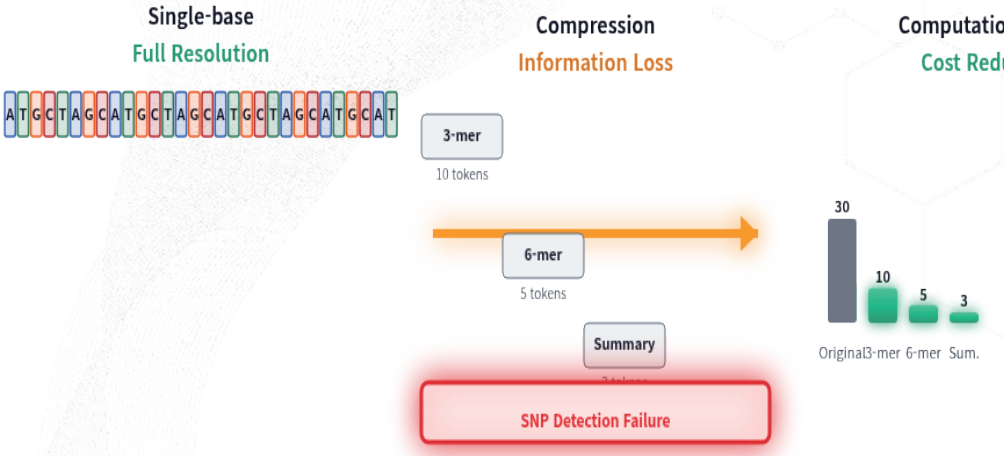
n=1M일 때 계산량 1조 연산. GPU 메모리와 시간 모두 초과

GPU 메모리 한계

100만 토큰을 위한 Attention 행렬은 현실적으로 100GB 이상 필요

선택지 1: 해상도를 포기하는 접근

효율을 위한 해상도 희생



효율성 증가

토큰 수 감소로 계산량 대폭 감소. GPU 메모리 절약, 학습 시간 단축.

변이 누락

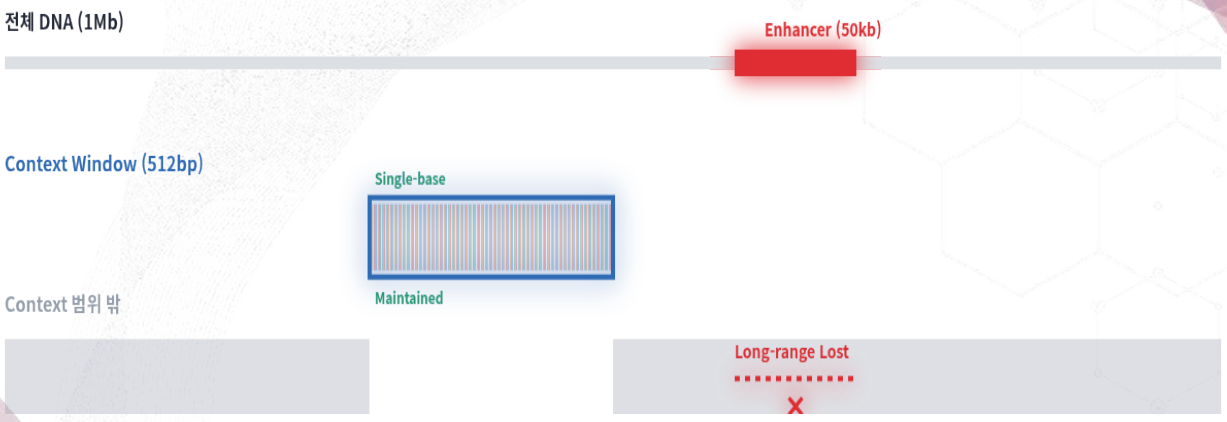
SNP, InDel 같은 단일 염기 변이를 직접 탐지 불가. 정밀한 변이 분석 어려움.

간접 표현

변이 효과는 간접적으로만 표현. 정밀한 기능 예측에는 한계가 있음.

선택지 2: 컨텍스트를 포기하는 접근

해상도는 유지하되, 입력 범위를 제한합니다



빠른 계산

512bp 이하의 짧은 입력으로 계산량을 최소화. $O(n^2)$ 복잡도에서도 실용적입니다.

국소 정확

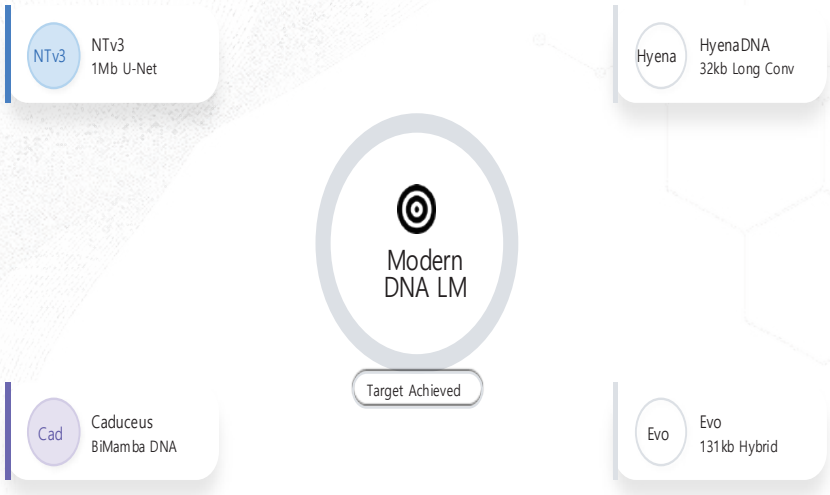
염기 단위 해상도 유지로 Splice site, promoter motif 등 국소 feature를 정확히 포착합니다.

장거리 실패

50kb+ 떨어진 Enhancer-promoter 상호작용, TAD 구조, 유전자 환경 등은 학습할 수 없습니다.

현대 DNA 모델의 목표

정밀도와 범위를 동시에 달성하는 최첨단 접근



U-Net 계층 설계

Skip Connections로 고해상도 정보 보존, 계층적 압축으로 장거리 문맥 학습

Long Convolution

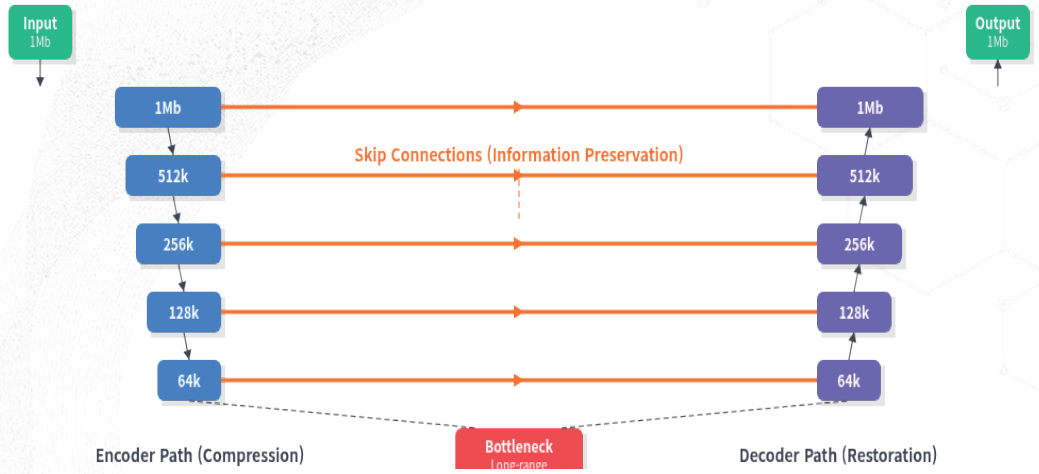
Hyena 접근: 긴 컨볼루션 필터로 효율적이고 확장 가능한 문맥 처리

Selective SSM

Mamba 접근: 선택적인 State 업데이트로 중요 정보만 유지, 계산 효율성 극대화

NTv3의 single-base 유지 전략

1Mb Context + Single-base Resolution의 동시 달성



점진적 압축 (Encoding)

Encoder는 1Mb 입력에서 시작해 5단계에 걸쳐 정보를 압축합니다. 고차원 문맥 정보는 추출되지만 해상도는 감소합니다.

→ Skip Connections

각 단계의 고해상도 정보를 Decoder로 직접 전달합니다. 이를 통해 압축 과정에서 손실될 수 있는 미세 위치 정보를 보존합니다.

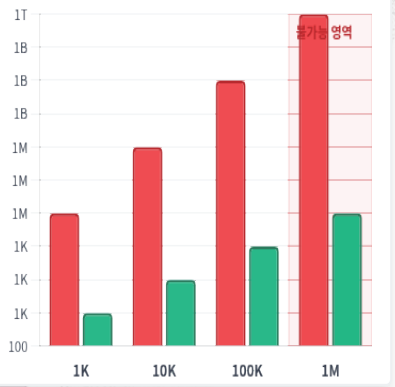
해상도 복원 (Decoding)

Decoder는 압축된 문맥 정보와 Skip connection의 위치 정보를 결합하여 최종적으로 1Mb Single-base 출력을 생성합니다.

모델별 계산 비용 종합 비교

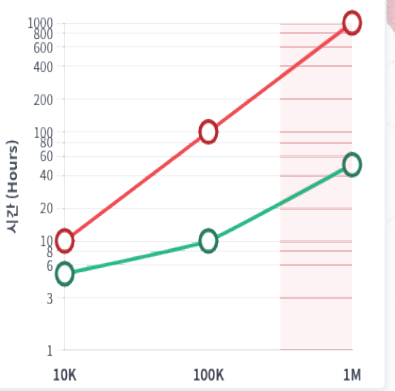
복잡도-메모리-시간의 3중 장벽과 SSM의 돌파구

계산 복잡도 (O)



GPU 메모리

학습 시간



Transformer 장벽

100kb 이상에서 O(n²) 복잡도로 폭발. 1Mb 처리에 수천 배의 리소스가 필요하여 사실상 실행 불가능.

SSM 돌파구

Mamba/Hyena는 O(n) 선형 확장. 1Mb 시퀀스도 단일 GPU 메모리 내에서 50시간 이내 학습 가능.

실용성 격차

100kb 기준 Transformer 대비 100배 연산 효율, 20배 메모리 절감, 10배 빠른 속도 달성.

HyenaDNA와 Evo의 선택

정밀도와 범위, 둘 다 포기하지 않는다 (Single-base + Ultra-long Context)

HyenaDNA

Stanford Hazy Research

CONTEXT RANGE

32kb ~ 1Mb — Ultra-long

TOKENIZATION

Single-nucleotide — Max Precision

ARCHITECTURE

Long Convolution (FFT)

CORE INNOVATION

Attention의 O(N²) 병목을 FFT 기반 컨볼루션 O(N log N)으로 대체하여 효율성 확보

Evo

Arc Institute

CONTEXT RANGE

131kb ~ 1Mb — Ultra-long

TOKENIZATION

Single-nucleotide — Max Precision

ARCHITECTURE

StripedHyena (Hybrid)

CORE INNOVATION

Hyena 연산자와 Attention을 하이브리드로 결합하여 추론 품질과 긴 문맥 처리 동시 달성

어떻게 달성했는가?

기존 Attention의 2차 복잡도 장벽을 Sub-quadratic 아키텍처(Long Convolution, SSM)로 대체했습니다.

무엇을 거부했는가?

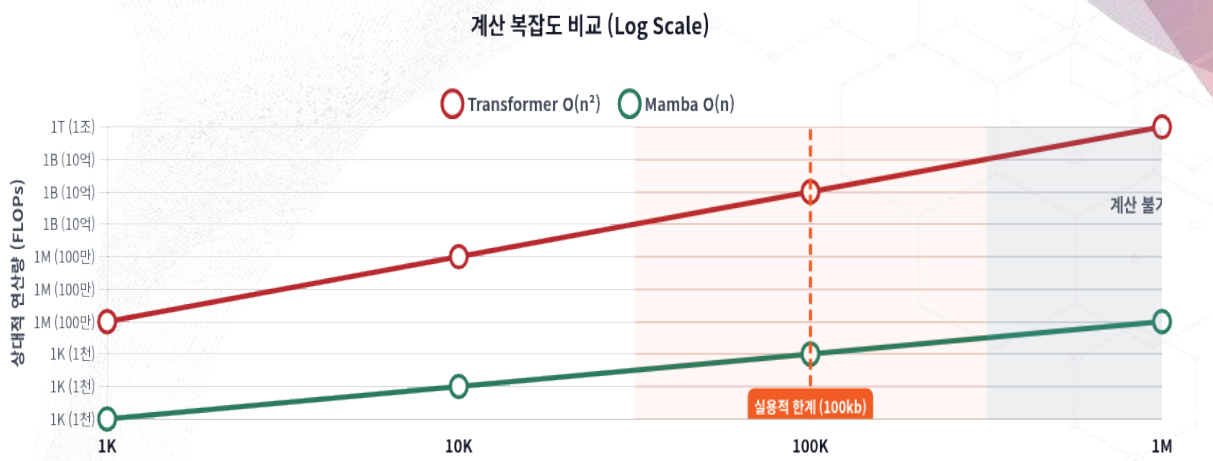
대다수 모델이 선택한 '해상도 타협(k-mer)'과 'Context 축소'를 모두 거부했습니다.

결과는 무엇인가?

단일 염기 변이(SNP)가 수십만 염기 떨어진 유전자 발현에 미치는 영향을 직접 모델링합니다.

Transformer vs Mamba 복잡도 비교

이론적 복잡도의 실제 차이와 계산 불가능 영역



O(n²) 폭발

Transformer는 1M 토큰에서 1조 번의 연산을 수행합니다. 시퀀스 길이가 2배 증가할 때마다 계산량이 4배로 폭증하여 100kb를 넘어서면 물리적 한계에 직면합니다.

O(n) 선형

Mamba/SSM은 1M 토큰에서 100만 번의 연산만 수행합니다. 시퀀스 길이에 정비례하여 선형적으로 증가하므로 수백만 토큰까지 안정적으로 확장 가능합니다.

실용성 격차

100kb 이상에서 실용적 차이가 극명하게 발생합니다. 1M 토큰에서는 이론적으로 1만 배 이상의 효율성 격차가 나타나며 이는 학습 가능 여부를 결정합니다.

GPU 메모리 사용량 비교

DNA 모델의 메모리 사용량은 시퀀스 길이에 따라 급증합니다. Transformer는 메모리 장벽에 부딪히지만, SSM 계열은 효율적으로 처리합니다.

Transformer의 메모리 장벽

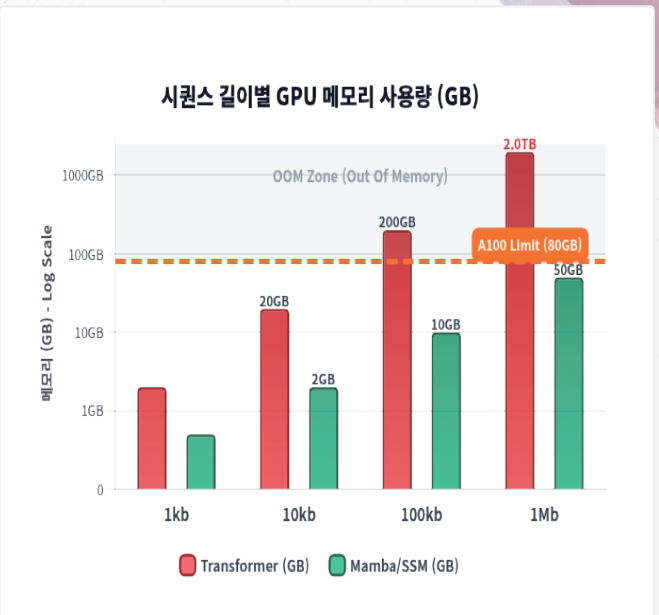
1Mb 처리 시 수백 GB 필요. 일반적인 A100 GPU(80GB) 용량을 초과하여 학습이 불가능합니다.

Linear 모델의 혁신적 효율성

SSM, Hyena, Evo는 1Mb Context 처리 시 8-50GB 수준으로, 단일 GPU 학습이 가능합니다.

연구 접근성의 대중화

수백 대의 GPU 클러스터가 없어도, 일반 연구실 단위에서 Long-range 모델 실험이 가능해집니다.



학습 시간 비교 분석

HyenaDNA는 시퀀스 길이가 증가해도 학습 시간이 선형적으로 유지되는 반면, Transformer는 기하급수적으로 폭발합니다.

핵심 인사이트

500kb 길이에서 Transformer는 사실상 학습이 불가능한 영역(수천 시간)에 진입하지만, HyenaDNA는 50시간 내에 처리가 가능하여 연구의 가능성을 열어줍니다.

Length	Transformer		HyenaDNA
1kb Short	1.0 h	›	0.5 h
10kb Medium	10.0 h	»	2.0 h
100kb Long	100.0 h	10x	10.0 h
500kb Ultra-Long	⚠ 수천 시간 Not Feasible	∞	50.0 h

Source: Nguyen et al., 2023, "HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution"

모델 비교 핵심 질문

모델 선택 체크리스트

질문 / 모델	NTv3	HyenaDNA	Evo	Caduceus
Single-base 해상도 유지?	✓	✓	✓	✓
Long-range Context 확보?	△	✓	✓	✓
복잡도 O(n) 실행 가능?	✗	✓	✓	✓
GPU 메모리 80GB 이하?	✗	✓	✓	✓
학습 시간 수용 가능?	✗	✓	✓	✓

⚠ 필수 기준

Single-base 해상도는 생물학적 정확성을 위한 필수 조건입니다. 이를 포기하는 모델은 변이 해석에 한계가 있습니다.

✓ 선택 기준

Context 길이와 계산 효율성은 모델 선택의 핵심 기준입니다. Long-range 처리 능력은 연구 목적에 따라 달라집니다.

⚖ 사용 사례

변이 효과 예측에는 NTv3, 대규모 분석에는 HyenaDNA/Evo, DNA 특화 작업에는 Caduceus가 적합합니다.

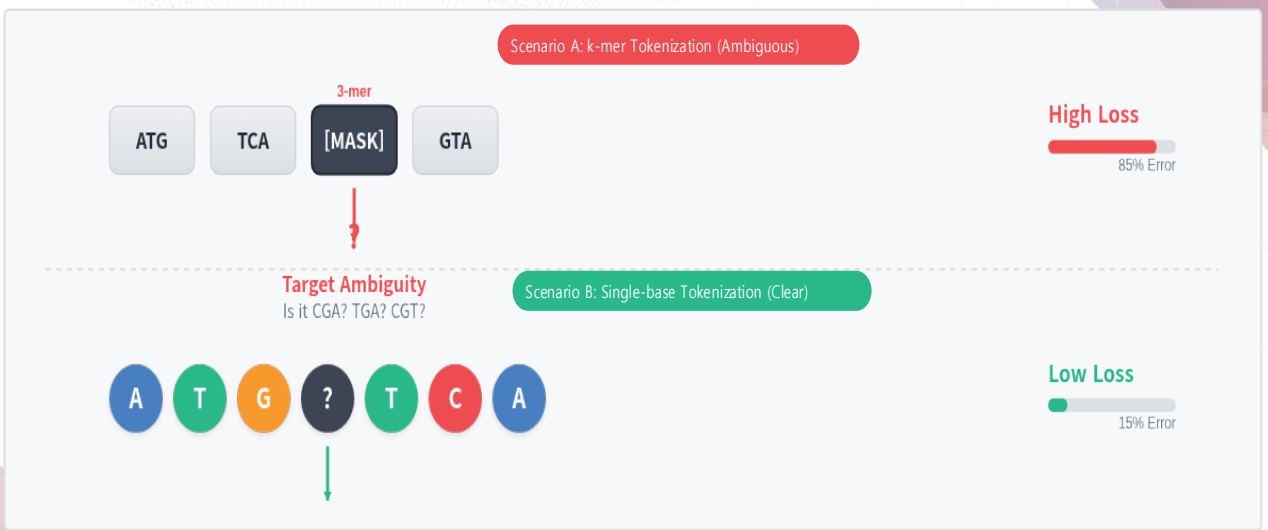
왜 모든 정보를 기억할 필요가 없는가

Loss는 모든 과거 정보를 요구하지 않습니다. 예측에 필요한 요약 정보만 있으면 됩니다.

예를 들어 현재 위치가 exon인지 예측할 때, 10kb 전의 모든 염기를 알 필요는 없습니다.
"10kb 전쯤에 promoter가 있었다" 정도의 요약 정보면 충분합니다. 불필요한 정보는 오히려 노이즈가 됩니다.

Single-base 해상도와 Loss의 관계

MLM의 목표는 가려진 '염기 하나'를 정확히 맞히는 것입니다. 따라서 염기 단위 표현이 필수적입니다.



MLM Objective

MLM은 가려진(Masked) 위치의 원래 정보가 무엇인지 정확하게 복원하는 것을 목표로 학습합니다.

1:1 Correspondence

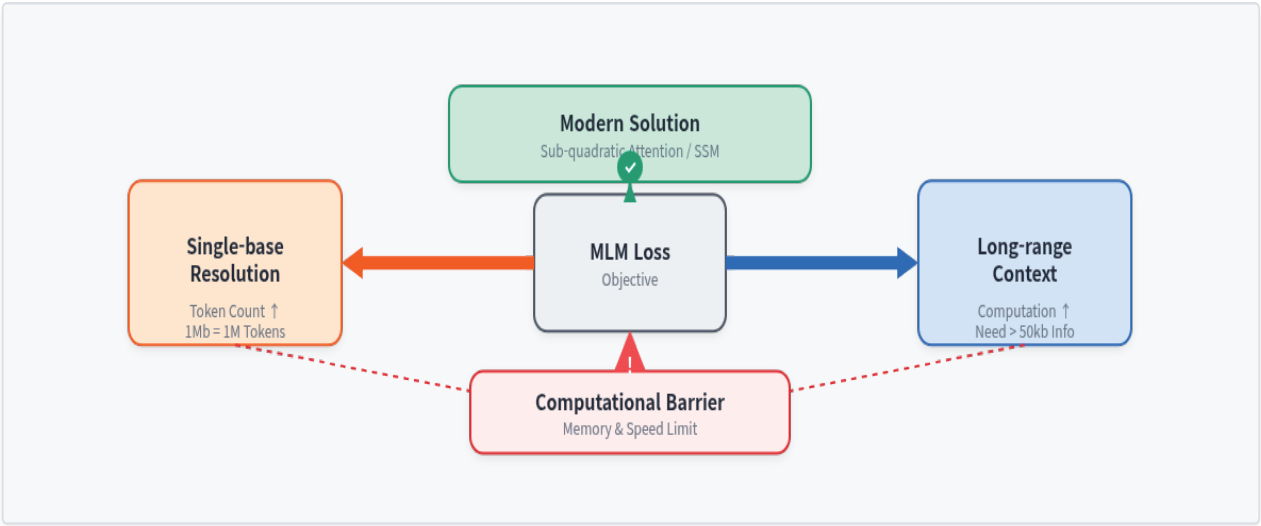
토큰과 예측 대상(염기)이 1:1로 대응되어야 Loss가 명확한 피드백을 줄 수 있습니다.

Precise Signal

Single-base 해상도는 모호함을 제거하여 모델이 더 날카롭고 정확한 신호로 학습하도록 유도합니다.

Long-range와 Single-base를 동시에 요구하는 Loss

MLM은 '정밀도(Precision)'와 '문맥(Context)'이라는 상충되는 두 가지 요소를 동시에 요구합니다.



Single-base Requirement

가려진 염기를 정확히 맞추기 위해 모든 단일 위치를 개별 토큰으로 처리해야 합니다. (토큰 수 급증)

Long-range Requirement

정답을 추론하기 위해 수십 kb 떨어진 Enhancer/Promoter 정보를 참조해야 합니다. (연산량 급증)

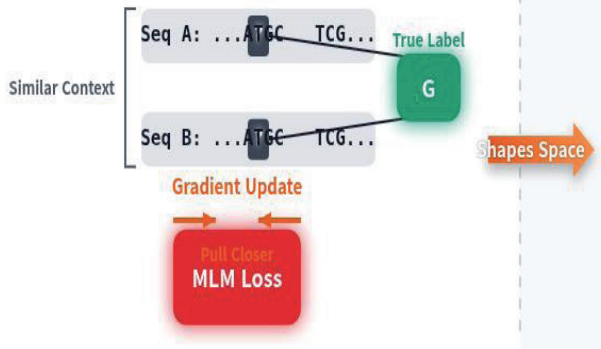
Modern Solution

최신 모델(Hyena, Mamba 등)은 이 충돌을 O(N log N) 또는 O(N) 복잡도로 해결하려 시도합니다.

Loss가 Embedding 공간을 만드는 방식

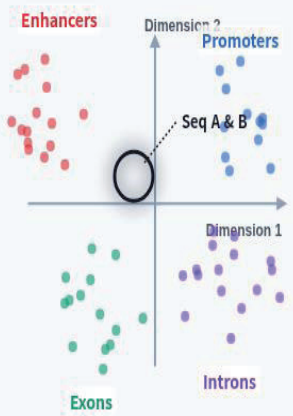
MLM Loss는 '비슷한 문맥'을 가진 토큰들을 Embedding 공간 상에서 '가까운 위치'로 당깁니다.

1. Loss & Update Mechanism



Shapes Space

2. Resulting Embedding Space



Loss Signal

같은 문맥에서 같은 정답을 요구하면, Loss는 두 입력의 벡터 표현을 서로 닮게 만들도록 신호를 보냅니다.

Gradient Update

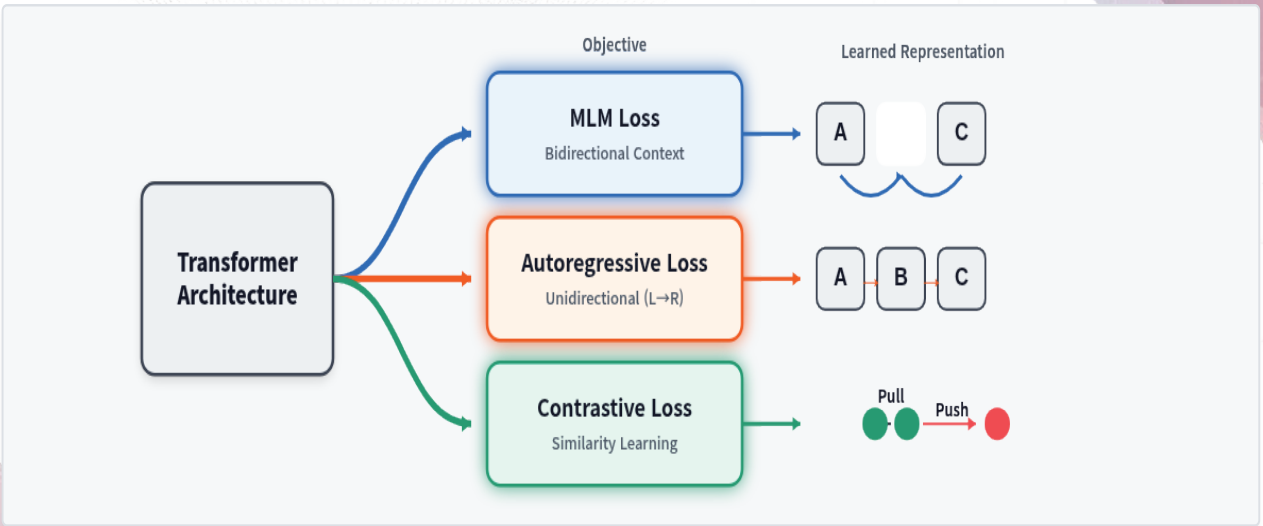
Backpropagation 과정에서 Gradient는 유사한 기능을 가진 서열들의 Embedding 벡터 거리를 좁힙니다.

Functional Clustering

결과적으로 Enhancer, Promoter 등 기능적으로 유사한 역할을 하는 DNA 요소들이 공간상에 군집을 형성합니다.

모델 비교에서 Loss를 봐야 하는 이유

같은 구조라도 Loss Function이 다르면 완전히 다른 모델이 됩니다. Loss가 모델의 정체성을 결정합니다.



논문 읽기의 시작점

아키텍처 그림보다 Loss 수식을 먼저 확인해야 모델이 무엇을 학습하려 했는지 정확히 파악할 수 있습니다.

성능 비교의 한계

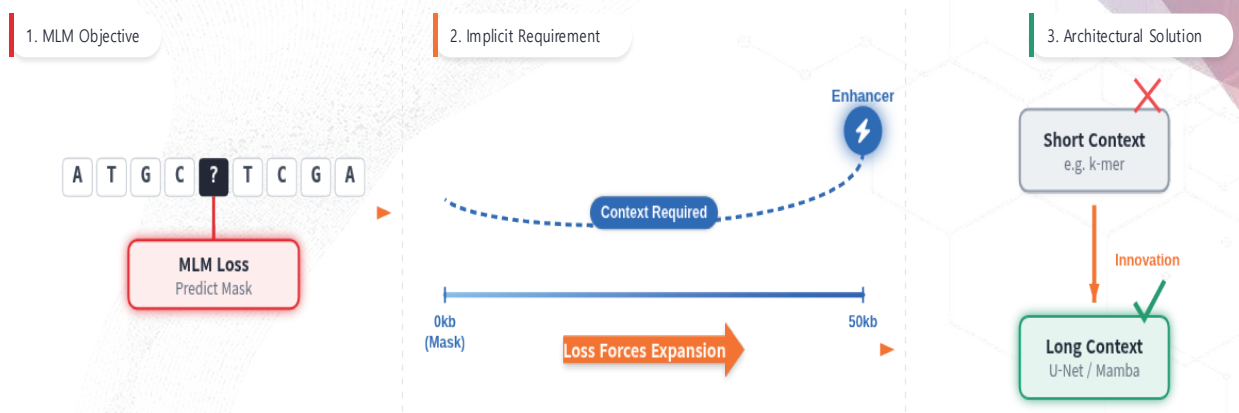
단순 수치 비교는 위험합니다. 학습 목적(Loss)이 다른 모델은 서로 다른 "언어"를 배우고 있는 것과 같습니다.

구조 선택의 근거

특정 구조(예: Attention)를 선택한 이유는 결국 특정 Loss(예: MLM)를 최소화하기 가장 유리했기 때문입니다.

Long-range 구조는 Loss의 요구다

MLM Loss의 '단순한 목표'가 역설적으로 '복잡한 아키텍처 혁신'을 강제하는 원동력이 됩니다.



Loss Objective

MLM은 겉보기에 단순히 가려진 염기([MASK]) 하나를 맞추는 것을 목표로 합니다.

Long-range Necessity

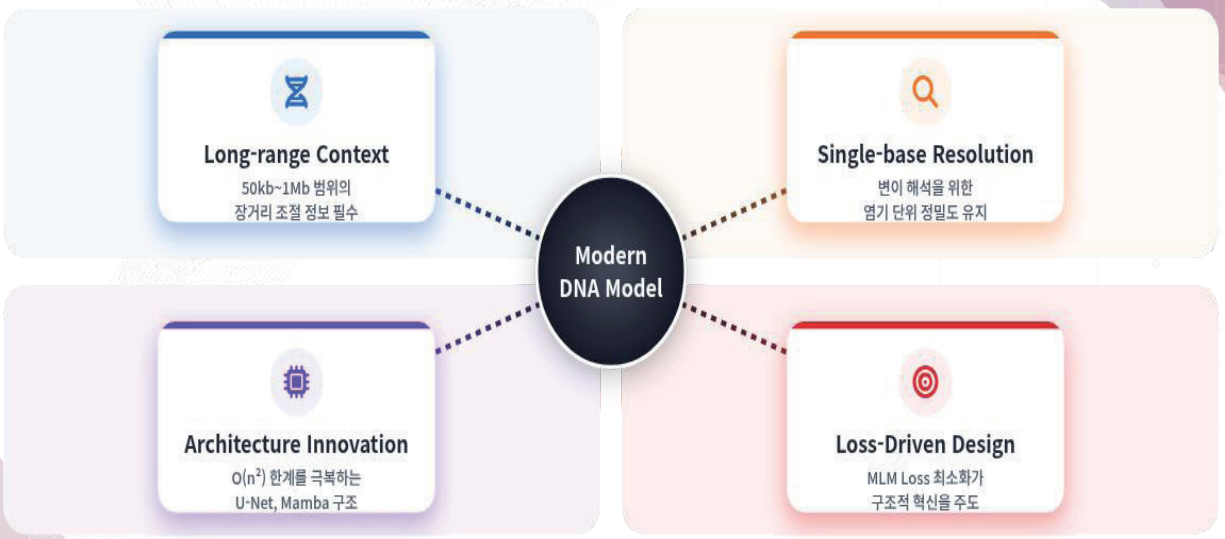
하지만 정답을 알기 위해서는 50kb 이상 떨어진 Enhancer의 정보가 필수적입니다.

Structural Innovation

이 간극을 메우기 위해 U-Net, SSM, Mamba 같은 구조적 혁신이 필연적으로 등장합니다.

지금까지 개념 연결

DNA 언어모델을 구성하는 핵심 개념들의 논리적 인과 관계와 흐름을 정리합니다.



입력과 표현 (Input)

DNA를 단일 염기(Token)로 보고 이를 고차원 벡터(Embedding)로 변환하여 의미 공간을 형성합니다.

범위와 구조 (Processing)

생물학적 필연성으로 넓은 문맥(Context)이 요구되며, 이를 효율적으로 처리할 아키텍처가 필요합니다.

목표와 최적화 (Goal)

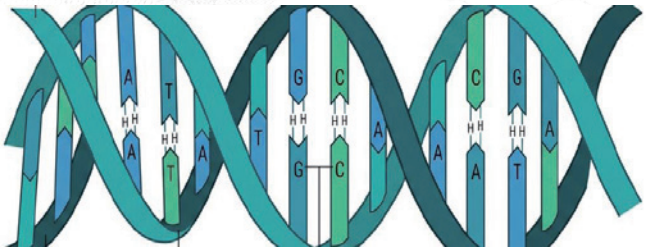
결국 모든 설계와 선택은 학습의 목표인 Loss 함수를 최소화하기 위한 수단으로 귀결됩니다.

학습 과정 모니터링

학습 목표: Loss curve, Learning rate, Gradient 등 학습 지표를 모니터링하고, Overfitting/Underfitting을 조기에 발견하는 방법을 익힙니다.

학습 과정 모니터링의 핵심

Learning Curves로 문제 조기 발견하기



학습 곡선은 모델의 학습 상태를 한눈에 보여줍니다.

Training loss와 Validation loss의 추이를 통해 Overfitting, Underfitting, 학습률 문제 등을 조기에 발견할 수 있습니다.

모니터링 포인트

1. Training loss: 모델이 학습 데이터를 얼마나 잘 학습하는지
2. Validation loss: 모델이 새로운 데이터에 얼마나 잘 일반화되는지
3. Loss gap: 두 곡선의 간격이 너무 벌어지면 Overfitting 신호

Loss curve: 가장 기본적인 지표

학습 진행에 따른 loss 변화를 추적합니다

Loss curve는 시간(epoch, step)에 따른 loss 변화를 보여줍니다. Training loss와 validation loss를 함께 그려 모델의 학습 상태를 진단합니다.

이상적인 패턴

빠르게 떨어지다가 점차 완만해지며, 최종적으로 수렴하는 형태입니다. Loss curve의 모양은 학습 상태를 말해주는 가장 기본적인 출발점입니다.

Training loss는 학습 데이터에서의 loss, validation loss는 보지 못한 데이터에서의 loss를 나타냅니다. 두 loss의 관계가 핵심 지표입니다.

Training vs Validation Loss

Training Loss

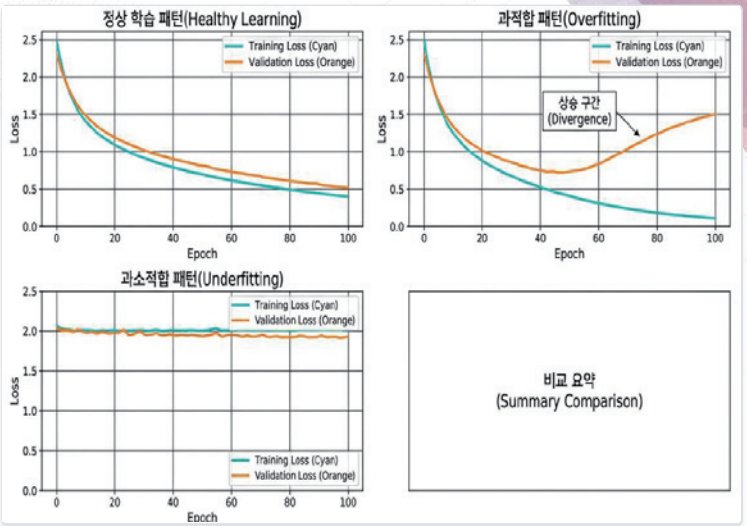
모델이 학습 데이터를 얼마나 잘 학습했는지(암기했는지)를 나타내는 지표입니다. 학습이 진행됨에 따라 지속적으로 감소해야 합니다.

Validation Loss

모델이 보지 못한 새로운 데이터에 대한 성능을 나타냅니다. 실제 일반화 능력을 평가하는 핵심 지표입니다.

핵심 인사이트

두 Loss의 차이(Gap)가 벌어지기 시작하는 지점이 바로 과적합(Overfitting)의 시작입니다.



정상 학습
Train/Val loss가 함께 감소하며 수렴. 가장 이상적인 형태.

과적합 (Overfitting)
Train loss 감소, Val loss 증가. 일반화 실패 신호.

과소적합 (Underfitting)
둘 다 높게 유지됨. 모델 용량 부족 또는 학습 부족.

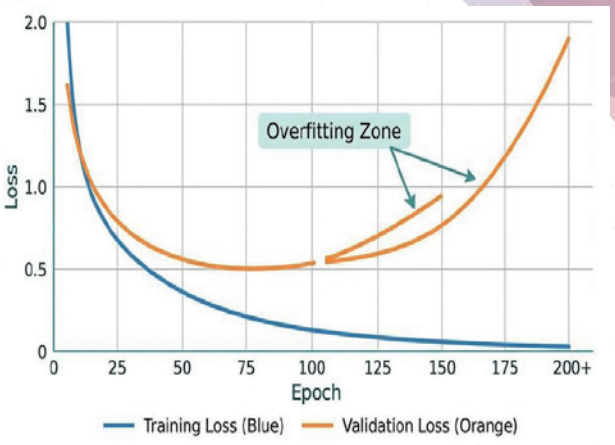
Overfitting 패턴

과적합의 정의

Overfitting은 학습 데이터에만 과도하게 맞춰진 상태입니다. Training loss는 계속 감소하지만, validation loss는 증가하거나 정체합니다. 모델이 학습 데이터의 특이성과 노이즈까지 외우고 있습니다.

주요 특징

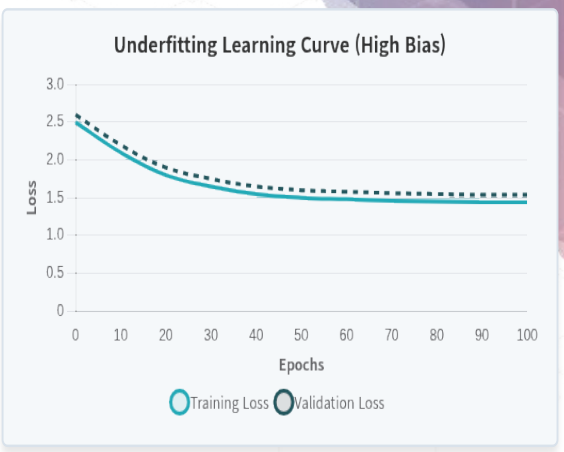
유전체 데이터는 반복 서열이 많아 외우기 쉽습니다. 예: Alu element 같은 반복 서열을 통째로 암기할 수 있습니다. 이는 일반화 능력을 잃어가고 있습니다.



원인: 모델이 너무 크거나, 학습 데이터가 너무 적거나, 학습을 너무 오래 했거나, regularization이 부족하거나. 문제를 조기에 발견하고 해결해야 합니다.

Underfitting 패턴

Underfitting은 모델이 데이터의 패턴을 충분히 학습하지 못한 상태입니다. Training loss와 validation loss가 모두 높게 유지되며, 학습이 더 진행되어도 loss가 잘 안 떨어집니다.



주요 특징

정의: 모델의 표현력이 부족하거나, 학습이 충분하지 않거나, learning rate가 너무 낮은 것이 원인일 수 있습니다.

해결 방법

모델을 키우거나, 더 오래 학습하거나, learning rate를 높이거나, 데이터 증강을 통해 해결할 수 있습니다. DNA 언어모델에서는 underfitting보다 overfitting이 더 흔한 문제입니다.

Loss Curve Pattern (High Bias)

그래프와 같이 Training Loss와 Validation Loss가 모두 높은 상태에서 평탄화(Plateau)됩니다. 이는 모델이 데이터의 복잡도를 담아내지 못하고 있음을 의미합니다.

Validation Performance

두 곡선의 간격이 좁지만 값 자체가 높다면, 일반화 성능보다는 모델의 기본 성능 부족을 의심해야 합니다.

Epoch, Step, Batch의 개념

Batch (배치)

모델이 한 번에 학습하는 데이터의 작은 묶음입니다. 예 : 100만 개 중 32개씩 묶음

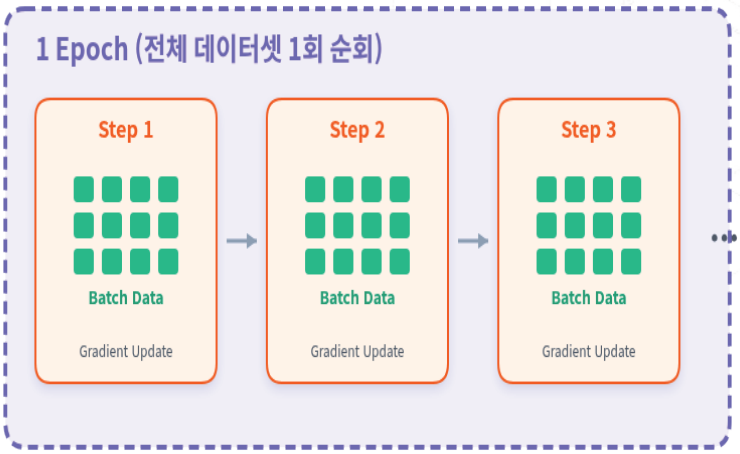
Step (스텝)

배치 하나를 학습하고 가중치(Weight)를 1회 업데이트하는 과정입니다.

Epoch (에포크)

전체 데이터셋을 한 바퀴 모두 순회하여 학습한 상태입니다.

총 데이터 N ÷ Batch Size B
= Steps per Epoch



Learning rate의 역할

Learning rate는 파라미터 업데이트의 크기를 조절합니다. Gradient가 방향을 알려주면, learning rate가 그 방향으로 얼마나 갈지 결정합니다.

너무 큰 Learning rate

Learning rate가 너무 크면 불안정합니다. Loss가 진동하거나 발산할 수 있습니다. 파라미터가 최적점을 지나쳐 과도하게 움직입니다.

너무 작은 Learning rate

Learning rate가 너무 작으면 학습이 느립니다. Loss가 아주 천천히 감소하거나 국소 최적에 갇힐 수 있습니다. 수렴에 매우 오랜 시간이 걸립니다.

적절한 learning rate는 데이터, 모델, task에 따라 다릅니다. 일반적으로 1e-3, 1e-4, 1e-5 같은 값을 시도합니다. DNA 언어모델은 보통 1e-4 수준을 사용합니다.

Learning Rate Schedule 패턴

Warmup과 Cosine Decay의 이상적인 결합

Warmup Phase

학습 초기 불안정성을 방지하기 위해 0에서 목표치까지 선형적으로 증가시킵니다. Gradient가 튀는 것을 막고 안정적인 출발을 보장합니다.

Cosine Decay Phase

목표 Learning rate 도달 후, 코사인 곡선을 그리며 0에 가깝게 감소합니다. 학습 후반부에 파라미터를 미세하게 조정하여 최적해(Global Minima)에 수렴하도록 돕습니다.

[그림] Transformer 모델 학습의 표준적인 Learning Rate Schedule 곡선

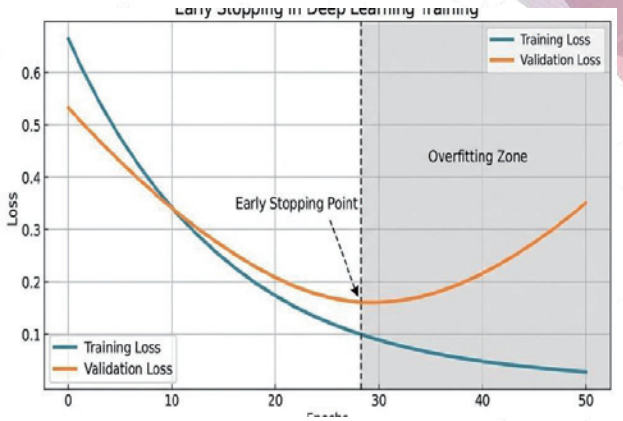
Early stopping: 언제 멈출 것인가

학습 종료 시점 결정 전략

Early stopping은 validation loss가 더 이상 개선되지 않을 때 학습을 중단하는 전략입니다. Validation loss를 모니터링하며 일정 epoch 동안(예: 10 epoch) 개선이 없으면 학습을 중단합니다.

핵심 목적

Overfitting을 방지하고, 가장 좋았던 모델(validation loss가 최저였던 시점)을 저장하여 시간과 자원을 절약합니다. DNA 언어모델은 학습에 오래 걸리므로 특히 중요합니다.



Early stopping은 validation loss가 증가하기 시작하는 시점에서 학습을 중단하여, 최적의 성능을 유지하면서도 과도한 학습을 방지합니다.

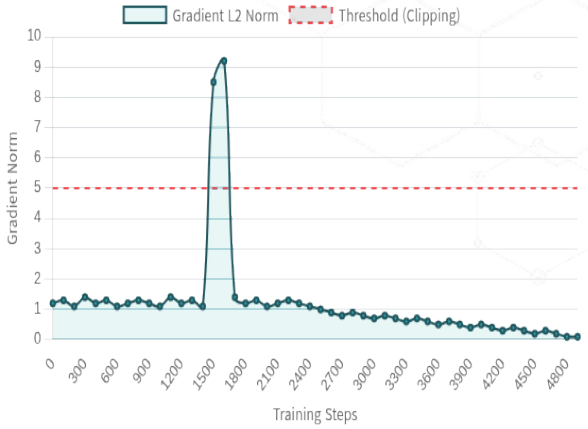
Gradient 모니터링의 중요성

학습 안정성을 진단하는 핵심 지표

딥러닝 모델, 특히 Transformer 기반 DNA 언어모델 학습 시 Gradient Norm(기울기 크기) 모니터링은 필수적입니다. Gradient가 너무 작으면 학습이 정체(Vanishing)되고, 너무 크면 발산(Exploding)하여 모델이 망가질 수 있습니다.

핵심 진단 포인트

급격한 Gradient Spike는 나쁜 배치 데이터나 과도한 Learning rate를 시사합니다. 이를 방지하기 위해 Gradient Clipping 을 적용하여 학습 안정성을 확보해야 합니다.



Gradient Norm 추이 예시: Spike 발생 및 Clipping을 통한 안정화

Loss가 감소하지 않을 때

체계적인 문제 해결 프로세스

모델 학습 중 Loss가 줄어들지 않거나 진동하는 경우, 무작위로 파라미터를 수정하기보다 체계적인 진단 순서를 따르는 것이 효율적입니다.

가장 흔한 원인인 Learning Rate부터 시작하여 모델 구조, 데이터 품질 순으로 점검 범위를 확장해 나갑니다.

핵심 진단 원칙

한 번에 하나의 변수만 변경하며 실험 결과를 기록하세요. 여러 요소를 동시에 수정하면 원인 파악이 불가능해집니다.

1단계: Learning Rate 확인

가장 빈번한 실패 원인입니다. 너무 높으면 발산하고, 너무 낮으면 학습이 정체됩니다.

긴급 점검 | 권장: 1e-3 ~ 1e-5

2단계: 모델 구조 점검

모델의 용량(Capacity) 문제입니다. 데이터 복잡도에 비해 모델이 너무 단순하거나 복잡하지 않은지 확인합니다.

구조 최적화 | Layer 수 / Hidden dim 조절

3단계: 데이터 품질 확인

데이터 자체의 문제입니다. 잘못된 라벨링, 노이즈, 전처리 오류가 모델 학습을 방해할 수 있습니다.

품질 검증 | Label Noise / Outlier 제거

Confusion Matrix와 분류 모니터링

Confusion Matrix란?

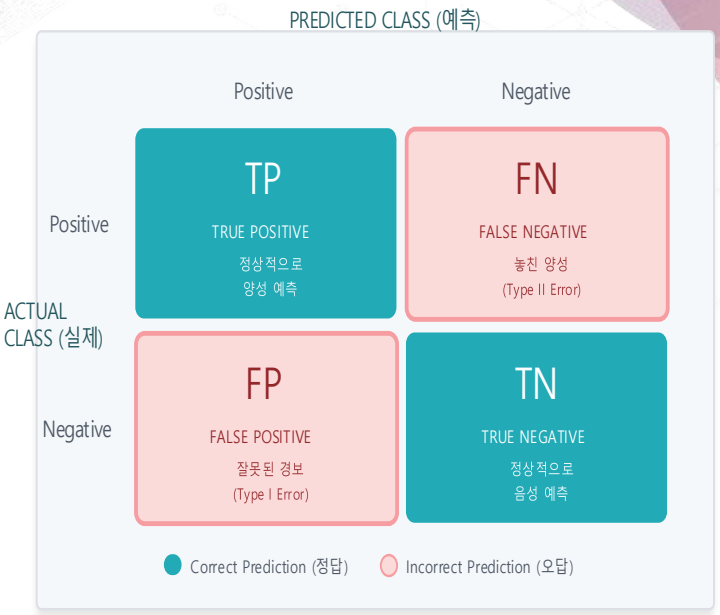
실제 클래스와 예측 클래스를 행과 열로 배치한 2×2 교차표입니다. 모델의 예측 성능을 네 가지 경우(TP, TN, FP, FN)로 상세히 분해하여 보여줍니다.

분석의 핵심

단순 Accuracy가 숨길 수 있는 편향을 드러냅니다. 특히 질병 진단이나 희귀 변이 예측처럼 데이터 불균형이 심한 경우, 어떤 유형의 오류(FP vs FN)가 많은지 파악하는 데 필수적입니다.

학습 모니터링

학습이 진행됨에 따라 대각선(TP, TN)의 비중이 높아지고 비대각선(FP, FN)이 감소하는 패턴을 시각적으로 확인하여 성능 개선을 검증합니다.



Precision-Recall Tradeoff

상충관계의 본질

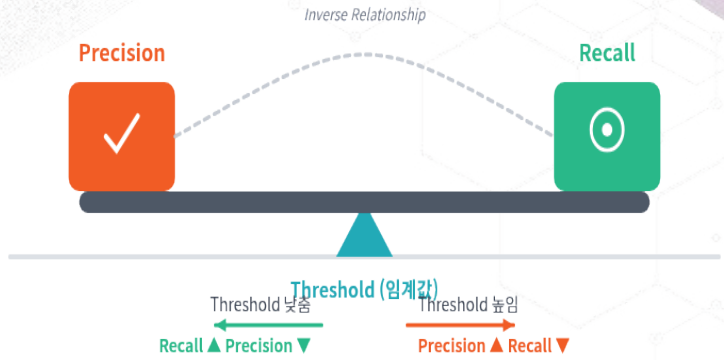
Precision과 Recall은 시소처럼 반대로 움직입니다. 하나를 얻으려 하면 다른 하나를 희생해야 하는 구조적 관계입니다.

Threshold의 역할

Threshold(임계값)는 이 균형을 조절하는 지지점입니다. 임계값을 어디에 두느냐에 따라 모델의 성격이 결정됩니다.

최적의 선택

완벽한 수치는 없습니다. 비즈니스 목적과 실패 비용(False Positive vs False Negative)을 고려하여 선택해야 합니다.



Screening (대규모 검사) Precision 우선

수많은 후보 중 진짜를 찾아내는 단계.
비용 절감을 위해 확실한 것만 선별합니다.

Diagnosis (진단) Recall 우선

질병이나 위험 요소를 놓치지 않는 것이 핵심
약간의 오탐을 감수하더라도 포괄적으로 찾습니다
.

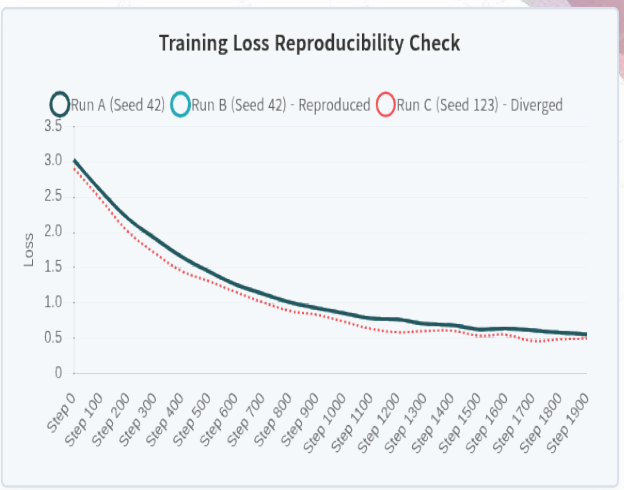
학습 재현성 (Reproducibility)

실험의 신뢰성 확보

DNA 언어모델 학습은 비용이 높고 복잡하므로 재현성 확보가 필수적입니다. 동일한 코드와 데이터로 언제나 같은 결과(Loss curve)를 얻어야 모델의 개선 여부를 정확히 판단할 수 있습니다.

재현성 확보를 위한 3대 요소

Seed 고정: Python, PyTorch, CUDA 난수 제어
환경 통제: 하드웨어 및 라이브러리 버전 고정
버전 관리: 코드 및 데이터셋의 스냅샷 저장



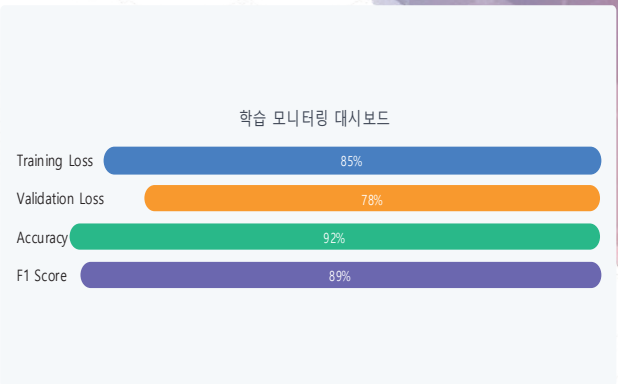
Seed 고정 시 Loss Curve의 완벽한 일치 (Run A vs Run B)

학습 과정 파트 핵심 정리

Loss curve는 기본 지표이며, training vs validation loss의 관계로 과적합/과소적합을 진단합니다.

Learning rate와 schedule이 학습 동역학을 좌우하며, early stopping과 gradient 모니터링으로 문제를 조기에 발견합니다.

Confusion matrix, precision, recall, F1, AUPRC, AUROC 등 다각도의 지표를 함께 모니터링해야 합니다.



Precision	Recall
0.91	0.87

AUPRC	AUROC
0.94	0.96

핵심 요약

학습 과정 모니터링은 모델 개발의 필수 과정입니다. Loss curve 분석만으로도 대부분의 학습 문제를 진단할 수 있으며, 적절한 early stopping과 regularization을 통해 최적의 모델을 얻을 수 있습니다.

모델 평가 지표

학습 목표: Precision, Recall, F1, AUROC, AUPRC 등 평가 지표의 의미를 이해하고, 문제 특성에 따라 적합한 지표를 선택할 수 있습니다.

모델 평가 지표 개념

정확한 평가를 위한 핵심 지표

모델 성능을 평가할 때는 단순한 정확도(Accuracy)만으로는 충분하지 않습니다. 특히 DNA 언어모델은 불균형 데이터를 다루기 때문에 여러 지표를 종합적으로 고려해야 합니다.

핵심 지표

Precision, Recall, F1-score, AUROC, AUPRC 등의 지표를 함께 사용하여 모델의 성능을 멀티차원적으로 평가합니다.

Precision

TP/(TP+FP)

정밀도: 양성으로 예측한 것 중 실제 양성 비율

Recall

TP/(TP+FN)

재현율: 실제 양성 중 양성으로 예측한 비율

F1-Score

2PR/(P+R)

정밀도와 재현율의 조화평균

AUROC

AUC

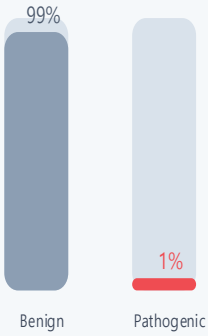
ROC 곡선 아래 면적

각 지표는 서로 다른 관점에서 모델의 성능을 평가합니다. DNA 언어모델에서는 희귀 변이나 조절 요소를 예측하는 데 특히 중요합니다.

Accuracy의 함정

⚠Class Imbalance 상황에서의 Misleading Metric

데이터 분포 (Ground Truth)



극단적 Class Imbalance

모든 샘플을 Benign으로 예측

☣ 모든 Pathogenic 놓침!

중요한 변이를 하나도 찾지 못함

Recall = 0%

재현율이 0인 무의미한 모델

✕ 실질적 가치 없음

동전 던지기보다 못한 진단 가치

99% Accuracy ≠ 좋은 모델. DNA 데이터처럼 불균형이 심한 경우(Benign >>> Pathogenic), Accuracy는 모델 성능을 심각하게 왜곡할 수 있습니다.

Confusion matrix 다시 보기

모든 성능 지표의 기초가 되는 2×2 표

Confusion matrix는 예측값과 실제값의 일치 여부를 보여주는 가장 기본적인 도구입니다. 이 표의 네 가지 구성 요소를 정확히 이해해야 정확도, 정밀도, 재현율 등의 파생 지표를 올바르게 계산할 수 있습니다.

핵심 구성 요소

- True Positive (TP): 질병을 질병으로 정확히 예측
- True Negative (TN): 정상을 정상으로 정확히 예측
- False Positive (FP): 정상을 질병으로 오진 (1종 오류)
- False Negative (FN): 질병을 정상으로 오진 (2종 오류)

Confusion Matrix 구조 및 계산

	Actual Positive (1)	Actual Negative (0)
Predicted Positive (1)	TP 정답을 맞춤(Positive)	FP 틀리게 예측(과잉 탐지)
Predicted Negative (0)	FN 틀리게 예측(놓침)	TN 정답을 맞춤(Negative)

Accuracy	Precision	Recall
(TP+TN) / Total	TP / (TP+FP)	TP / (TP+FN)

Precision(정밀도)의 의미

"내가 Positive라고 한 것 중 진짜는?"

$$\text{Precision} = TP / (TP + FP)$$

예측된 양성(Positive Prediction) 중 실제 양성 비율

핵심 개념

모델이 "이것은 병원성 변이(Pathogenic)입니다"라고 경고했을 때, 그 경고가 얼마나 신뢰할 수 있는지를 나타냅니다. Precision이 낮으면 '양치기 소년'처럼 잘못된 경보(False Alarm)가 많아집니다.

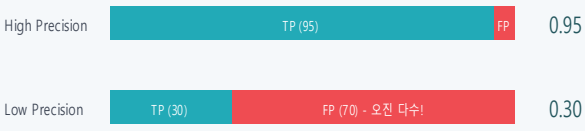
위험 요소: False Positive (FP)

Precision을 낮추는 주범은 FP입니다. 정상 변이를 병원성으로 오진하면, 불필요한 추가 검사와 환자의 불안을 초래합니다. 스팸 필터에서 정상 메일을 스팸으로 분류하는 것과 같습니다.

Confusion Matrix에서의 위치

	Predict: Positive (P)
Actual: P	TP 올바른 예측
Actual: N	FP 잘못된 경보!

Precision 시나리오 비교



↑ False Positive(FP)가 많아지면 Precision은 급격히 하락 ↓

Recall(재현율)의 의미

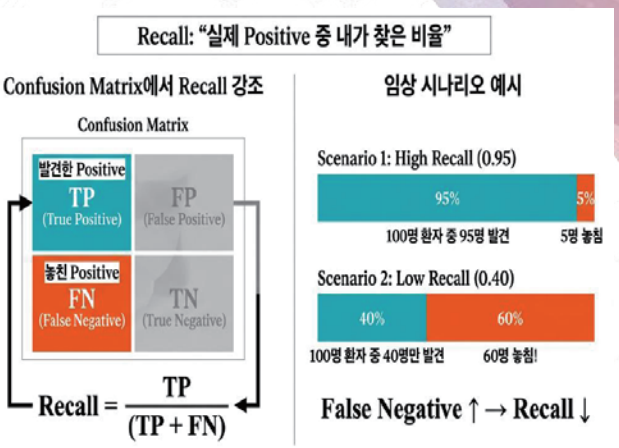
Recall (또는 Sensitivity)은 "실제 Positive 중 Positive로 예측한 비율"입니다. Recall = TP / (TP + FN)로 계산되며, 모델이 실제 중요한 케이스를 얼마나 놓치지 않고 잡아내는지를 평가합니다.

핵심 의미

"실제 pathogenic variant를 얼마나 잘 찾아냈는가"를 측정합니다. Recall이 높으면 False Negative(FN)가 적어, 중요한 변이를 놓치는 위험이 줄어듭니다.

임상적 중요성

Clinical application에서는 Recall이 매우 중요합니다. 희귀 질환이나 암 관련 변이를 놓치는 것(False Negative)이 환자에게 치명적일 수 있기 때문입니다.

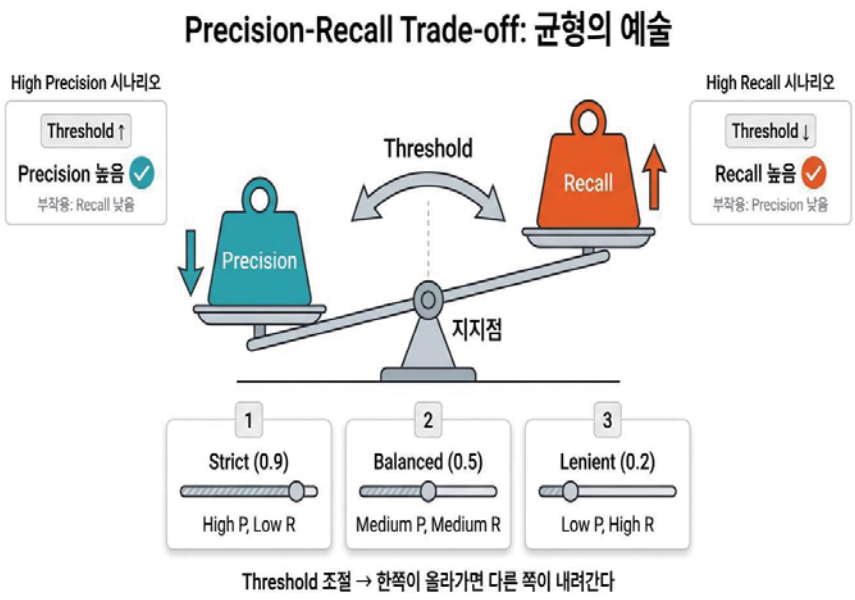


Confusion Matrix에서 Recall은 실제 Positive 행(Row)의 정확도를 의미합니다. 좌측 다이어그램처럼 Recall은 '놓친 Positive(FN)'에 민감하며, 우측 그래프는 낮은 Recall이 임상적으로 얼마나 많은 환자를 놓치게 되는지(High False Negative)를 시각적으로 보여줍니다.

Precision-Recall Trade-off

균형의 예술: Threshold 조절

Precision과 Recall은 마치 시소와 같아서, 하나의 성능을 높이려 하면 다른 하나는 필연적으로 낮아지는 상충 관계(Trade-off)를 가집니다. 이는 모델의 분류 기준인 Threshold(임계값) 설정에 따라 결정됩니다.



F1 Score: 조화평균의 힘

Precision과 Recall의 균형잡힌 지표

Harmonic Mean Calculation

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

"산술평균보다 작은 값에 더 민감하게 반응합니다"

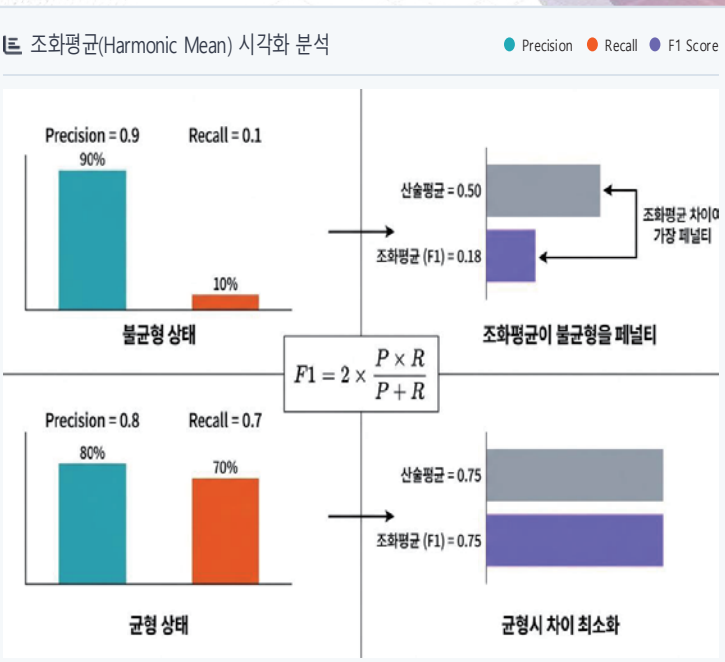
Imbalance 페널티

Precision이나 Recall 중 하나라도 매우 낮으면 F1 Score는 급격히 낮아져 모델의 결함을 드러냅니다.

불균형 데이터 필수 지표

Accuracy의 함정을 피하고 소수 클래스(Positive)에 대한 성능을 올바르게 평가할 수 있습니다.

조화평균(Harmonic Mean) 시각화 분석



위: 불균형 상태(P=0.9, R=0.1)에서는 산술평균(0.5)과 달리 F1(0.18)이 크게 낮아짐
아래: 균형 상태(P=0.8, R=0.7)에서는 산술평균과 F1이 유사한 값을 가짐

AUROC (Area Under ROC Curve)

AUROC는 다양한 Threshold 설정에서 모델의 분류 성능을 종합적으로 평가하는 지표입니다. ROC 곡선 아래의 면적을 계산하여 0.5(랜덤)에서 1.0(완벽) 사이의 값으로 나타냅니다.

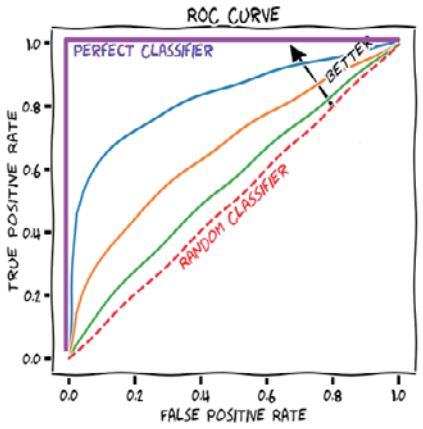
ROC 곡선이란?

X축:False Positive Rate (1-Specificity)

Y축:True Positive Rate (Sensitivity)
이 두 지표의 트레이드오프 관계를 시각화한 곡선입니다.

해석 방법

- AUROC = 0.5 : 무작위 추측 (Random Guess)
- AUROC > 0.8 : 우수한 분류 성능
- 의미: 랜덤한 Positive 샘플이 랜덤한 Negative 샘플보다 더 높은 점수를 받을 확률입니다.



곡선이 좌상단(Top-Left)에 가까울수록 더 우수한 성능을 의미합니다.

AUPRC (Area Under Precision-Recall Curve)

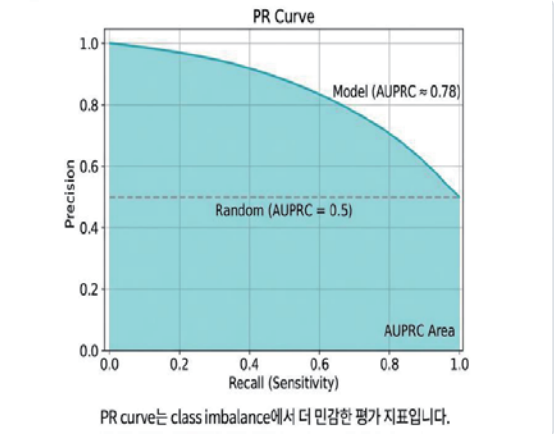
Precision-Recall curve는 Recall(x축)과 Precision(y축)의 관계를 그린 것입니다. AUPRC는 이 곡선 아래 면적으로, 0에서 1 사이의 값을 가집니다.

Imbalance 상황 평가

AUPRC는 class imbalance가 심한 상황에서 AUROC보다 더 정직한 지표입니다. Positive class가 극소수일 때 AUROC는 과대평가될 수 있지만, AUPRC는 그렇지 않습니다.

False Positive 민감도

AUPRC는 False Positive의 영향을 더 민감하게 반영합니다. 이는 imbalance가 심한 문제에서 특히 중요합니다.



PR curve는 class imbalance에서 더 민감한 평가 지표입니다.

DNA variant, regulatory element 예측처럼 imbalance가 심한 문제에서는 AUPRC를 우선적으로 사용해야 합니다. 하지만 AUPRC는 직관적으로 이해하기 어렵고, 비교 시 주의가 필요합니다.

AUROC vs AUPRC: 언제 무엇을 쓰는가

차이점

Imbalance 민감도: Class가 50:50으로 균형 잡혀 있으면 두 지표는 비슷합니다. 하지만 불균형이 심해질수록 (1:99 등) 두 지표의 차이는 커집니다.

Warning Signal

AUROC는 0.9로 매우 높지만 AUPRC는 0.2로 매우 낮을 수 있습니다. 이는 모델이 실제로는 성능이 좋지 않은데 AUROC 수치에 속을 수 있는 위험한 상황입니다.

Positive Class 비율은?

≥ 10% (균형/경미)	< 10% (불균형)	< 1% (극심한 불균형)
AUROC	AUPRC	AUPRC
사용 가능 (안전)	사용 권장 (Better)	필수 사용 (Must)
DNA 예시 일반적인 유전자 분류 , CpG island 등	DNA 예시 Enhancer (2-5%), TFBS (Transcription Factor Binding Site)	DNA 예시 Pathogenic Variant (0.1%), Splice Site (극소수)

핵심 원칙: 대부분의 DNA 응용(Variant, Promoter 등)은 심한 Class Imbalance를 가지므로, 습관적으로 AUPRC를 우선 확인하는 것이 안전합니다.

Specificity와 Sensitivity

모델 성능 평가의 두 축: 민감도와 특이도

Sensitivity (민감도)

Recall = TP / (TP + FN)

"실제 Positive 중
얼마나 잘 찾아냈는가"

↑ 높을수록 False Negative(놓친 것) 적음

Q 발견 중심의 지표 (재현율과 동일)

Clinical: 암 검진 등 질병을 놓치지 않아야 할 때 중요 (Screening)

Specificity (특이도)

TNR = TN / (TN + FP)

"실제 Negative를
얼마나 Negative로 맞췄는가"

↑ 높을수록 False Positive(오경보) 적음

▼ 정확성 중심의 지표

Clinical: 확진 판정 시 오진을 방지하기 위해 중요 (Confirmation)

Trade-off 관계

상충 관계: Threshold를 낮추면 Sensitivity는 증가하지만, Specificity는 감소합니다. 목적에 따른 균형점 선택이 필요합니다.

Class imbalance 문제의 심각성

1. DNA 데이터의 본질적 불균형

Promoter는 전체 게놈의 0.1% 미만, Pathogenic variant는 전체 variant의 1% 미만, Splice site는 유전자 내에서도 극소수입니다. 이런 불균형은 DNA 데이터의 본질적 특성입니다.

2. Naive 모델의 경향

모델은 loss를 최소화하기 위해 "모두 Negative"로 예측하는 쉬운 방법을 선택합니다. 이런 모델도 높은 accuracy를 달성하지만 전혀 쓸모가 없습니다.

3. 대응 전략

Class weight, focal loss, oversampling 등으로 대응해야 하지만, 근본적으로 어려운 문제입니다. 평가 지표 선택이 매우 중요합니다.

Baseline과의 비교

성능 평가의 기준 설정

모델 성능은 절대값이 아니라 적절한 baseline과 비교해야 의미가 있습니다. 특히 Class Imbalance 상황에서는 단순 Accuracy가 아닌, Random Guess나 Majority Class 예측과 비교하여 실제 정보 획득량을 평가해야 합니다.

핵심 원칙

"AUROC 0.85"라는 절대 수치보다 "Baseline 0.5 대비 0.85"라는 상대적 향상이 훨씬 정보가 많습니다. 항상 가장 단순한 모델(Naive Baseline)을 먼저 설정하고 이를 넘어서는지 확인하세요.

1. Random Guess

Worst Case

무작위로 예측하는 경우입니다. 모델이 최소한 학습을 했는지 판단하는 절대적 하한선입니다.

AUROC ≈ 0.50 | Acc ≈ 50%

2. Majority Baseline

Naive

항상 다수 클래스(예: Negative)로만 예측하는 경우입니다. Imbalance 시 Accuracy는 높지만 F1은 0에 가깝습니다.

Acc ≈ 99% | F1 ≈ 0.0 | AUPRC ≈ Low

3. Your Model

Target

Baseline을 넘어선 유의미한 패턴 학습 결과입니다. 모든 지표에서 균형 잡힌 성능을 보여야 합니다.

AUROC > 0.85 | F1 > 0.70 | AUPRC > Baseline

Imbalance 데이터(예: 1:99)에서는 Majority Baseline이 Accuracy 99%를 달성하므로, Accuracy는 성능 지표로 무의미할 수 있습니다.

Cross-validation의 중요성

단일 검증의 한계

단일 Train/Test 분할은 데이터 분포의 편향으로 인해 모델 성능을 과대평가하거나 신뢰하기 어렵게 만듭니다. 특히 DNA 데이터에서는 단순 무작위 분할 시 서열 유사성(Sequence Similarity)으로 인한 Information Leakage가 발생할 위험이 매우 높습니다.

Standard K-fold

데이터를 K개로 나누어 교차 검증을 수행함으로써, 성능의 평균과 표준편차를 확인하여 통계적 신뢰성을 확보합니다.

Homology-aware

서열 유사도(Homology)를 기준으로 데이터를 클러스터링하여 분할함으로써 Leakage를 원천 차단합니다.

실전에서의 지표 선택 원칙

문제 특성에 따른 지표 선택 가이드

문제 특성 파악

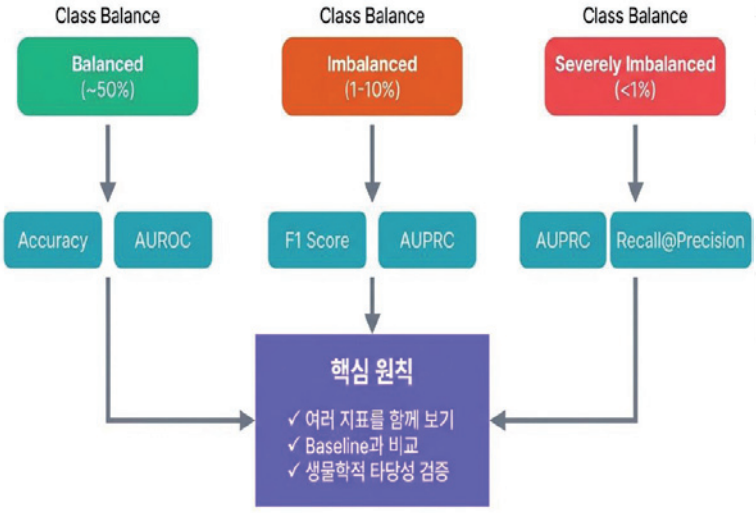
Class balance 체크:

Positive class가 전체 데이터의 몇 %를 차지하는지 확인하는 것이 첫 단계입니다.

응용 목적 파악

Cost Matrix 고려:

FN(놓침)이 더 치명적인지, FP(오탐)가 더 치명적인지, 혹은 둘 다 중요한지 평가합니다.



Foundation Model 활용

학습 목표: Zero-shot과 Fine-tuning의 차이를 이해하고, Pretrained 모델을 활용한 실전 응용 방법과 고려사항을 학습합니다.

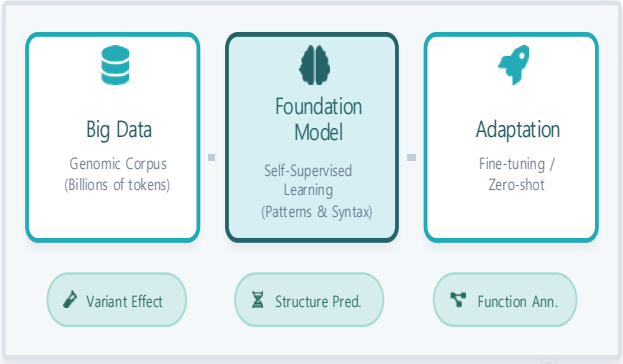
Foundation model이란

대규모 데이터로 사전학습된 범용 모델

Foundation model은 수십억 개의 DNA 서열로 사전 학습(Pre-training)을 수행하여 언어의 패턴을 익힌 거대 모델입니다. 특정 목적을 위해 만들어지지 않았지만, 미세 조정(Fine-tuning)을 통해 다양한 생물학적 문제를 해결할 수 있습니다.

핵심 특징

학습된 표현(embedding)과 파라미터를 다양한 응용에 재사용할 수 있습니다. 대표적인 DNA 모델로는 Nucleotide Transformer, Evo, DNABERT 등이 있으며, 이는 유전체 연구의 기반이 됩니다.



Foundation model의 진정한 가치는 'One Model, Many Tasks'에 있습니다. 방대한 데이터에서 보편적인 유전 문법을 학습한 모델은 최소한의 추가 학습만으로도 수백 가지의 다운스트림 태스크에서 뛰어난 성능을 발휘합니다.

Pretrained 모델의 출력 유형

Embedding (벡터 표현)

각 위치 또는 전체 서열의 수백 차원 벡터 표현으로, 다양한 downstream 분석의 기초가 됩니다. 유사성 분석, 클러스터링, 분류 등에 활용됩니다.

Per-base Probability

각 위치에 A, C, G, T가 올 확률을 예측하며, MLM 방식으로 학습된 모델의 기본 출력입니다.

Contact Map (3D 구조)

서열 내 두 위치 간 3D 접촉 확률을 예측합니다. AlphaFold-style 접근으로 RNA 구조나 chromatin loop를 예측할 수 있습니다.

Gene Expression Score

유전자 발현 수준을 예측하며, promoter, enhancer 활성도를 수치화합니다
.

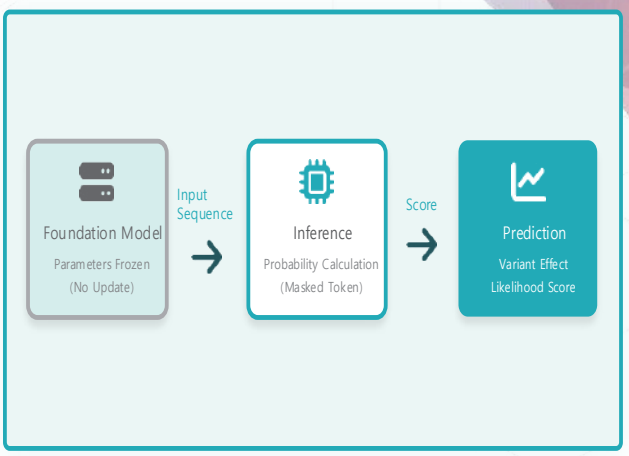
Regulatory Activity

Enhancer, promoter, silencer 등의 조절 기능을 예측하고, TF binding affinity를 추정합니다.

Zero-shot Prediction

추가 학습 없이 바로 예측하기

Zero-shot은 모델의 파라미터를 업데이트하지 않고, 사전 학습된 지식(Pre-trained Knowledge)만을 활용해 새로운 태스크를 수행하는 방법입니다. DNA 언어모델은 수십억 개의 서열에서 학습한 문맥 정보를 바탕으로, 특정 변이(Variant)가 생물학적으로 자연스러운지 즉시 평가할 수 있습니다.



핵심 메커니즘

모델은 '마스크 된 위치에 원래 무엇이 와야 하는지'를 확률로 계산합니다. 이 확률값(Likelihood)을 스코어로 변환하여, 별도의 Fine-tuning 없이도 병원성 예측이나 기능 평가를 수행합니다.

프로세스 요약: 거대 Foundation Model을 'Frozen(동결)' 상태로 두고, 입력 서열에 대한 모델의 내재적 확률 분포를 계산하여 결과를 도출합니다. 이는 데이터 레이블링 비용과 학습 시간을 '0'으로 만듭니다.

왜 zero-shot이 작동하는가

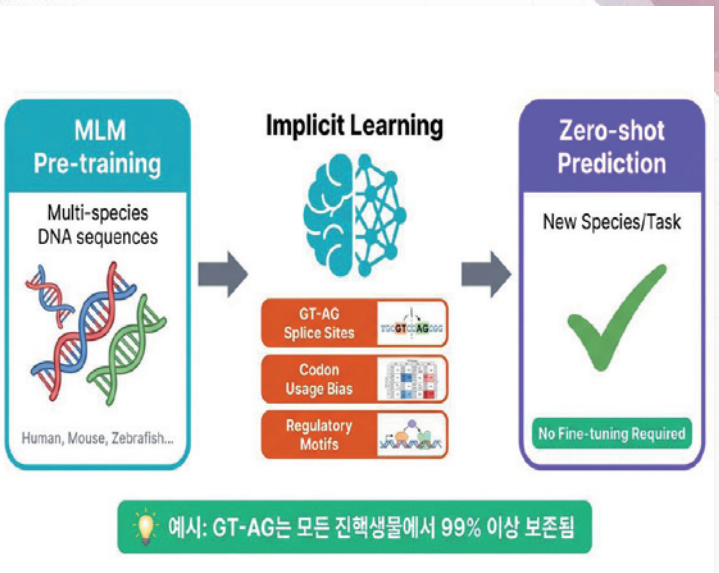
MLM 학습은 서열의 자연스러움을 학습합니다. 이 과정에서 생물학적 제약이 간접적으로 학습되며, 이러한 보편적 원리는 다양한 종에서 공통적으로 작용합니다.

보편적 원리 학습

MLM은 명시적인 지도 없이도 Splice site, Codon usage 등의 보존된 패턴을 자연스럽게 포착합니다. 이는 마치 어린아이가 문법을 배우듯 DNA의 언어를 습득하는 과정입니다.

Zero-shot 일반화

학습된 보편적 원리는 새로운 종이나 태스크에도 그대로 적용됩니다. 따라서 추가 학습(Fine-tuning) 없이도 즉시 추론이 가능합니다.



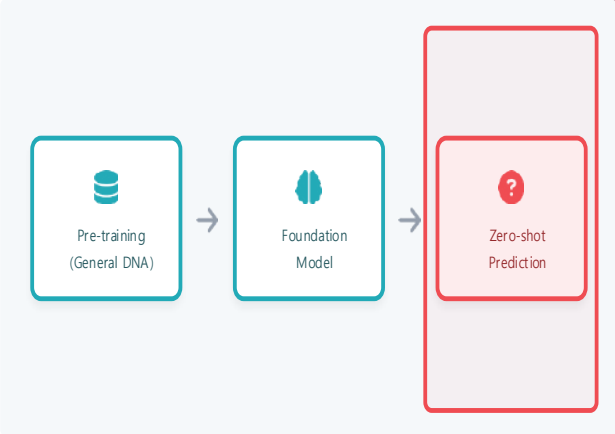
Zero-shot의 한계

성능 제약

Zero-shot은 편리하지만 최적 성능을 보장하지 않습니다. Task-specific 정보를 활용하지 못하며, 학습 데이터 분포와 많이 다른 태스크에서는 성능 저하가 클 수 있습니다.

주요 한계점

1. 특정 질병 관련 variant 예측 시 해당 질병의 clinical data가 없으면 정확도가 떨어집니다.
2. 해석이 어려워 예측 이유를 명확히 설명하기 힘들 수 있습니다.
3. Fine-tuning보다 일반적으로 낮은 성능을 보입니다.



실전에서는 zero-shot으로 시작하여 가능성을 확인한 후, 필요시 fine-tuning으로 개선하는 것이 일반적입니다. (Visualization inspired by DNA language model concepts)

Fine-tuning이란

태스크 특화 추가 학습

Pretrained model의 파라미터를 초기값으로 사용하고, task-specific data로 추가 학습을 진행합니다. DNABERT를 promoter 인식 task로 fine-tuning하는 것이 대표적 예시입니다.

핵심 장점

소량의 labeled data(수천~수만 샘플)로도 높은 성능을 얻을 수 있으며, 학습이 빠르고 from scratch보다 훨씬 효율적입니다.

Foundation model이 이미 좋은 표현을 가지고 있기 때문에, 적은 데이터로도 task에 최적화된 모델을 만들 수 있습니다.

Zero-shot vs Fine-tuning 비교

프로젝트의 목표와 데이터 가용성에 따라 최적의 전략을 선택하세요. Zero-shot은 속도와 효율성을, Fine-tuning은 성능과 최적화를 제공합니다.

선택 가이드

Zero-shot: 데이터 부족, 빠른 프로토타이핑, 범용적 태스크 Fine-tuning: 충분한 데이터, 최고 성능 요구, 특화된 도메인

- 즉시성 & 일반화
- 최적화 & 고성능

Zero-shot

Pretrained Model
기본 모델 그대로 사용

↓

Instant Inference
No Training

↓

Direct Application
즉시 결과 도출

장점
- 빠른 프로토타이핑
- 라벨 데이터 불필요
- 즉시 사용 가능 (Instant)

제약사항
- 최적 성능 미보장
- 태스크 특화 부족

Fine-tuning

Pretrained Model
기본 모델 준비

↓

Weight Update
Training Required

↓

Optimized Model
태스크 특화 모델

장점
- 최고 성능 달성 (SOTA)
- 도메인/태스크 완벽 적응
- 맞춤형 결과물

제약사항
- 라벨 데이터 필수
- 학습 시간 및 비용 발생
- Overfitting 위험

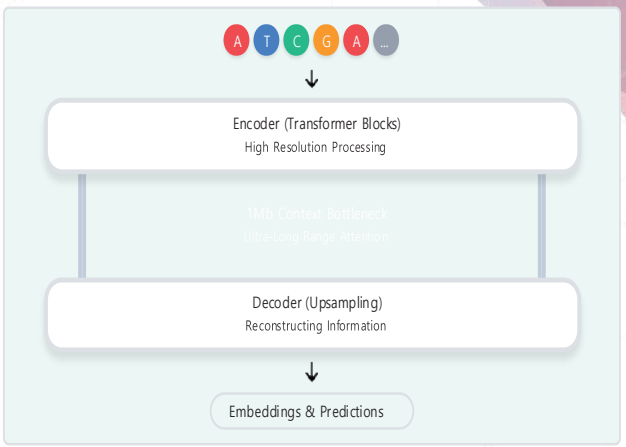
전략: Zero-shot으로 빠르게 검증 후, 필요시 Fine-tuning으로 성능 고도화

Nucleotide Transformer v3의 특징

2024년 공개된 최신 DNA Foundation Model

Nucleotide Transformer v3 (NTv3)는 U-Net 아키텍처를 활용하여 1Mb context window를 지원하는 최신 foundation model입니다. Single-base resolution을 유지하면서도 ultra-long range 문맥을 처리할 수 있습니다.

- Multi-species 학습: 동물, 식물, 미생물 등 다양한 종의 게놈 데이터로 학습
- 대규모 학습300B (3천억) 개 염기 서열로 사전 학습
- 다양한 출력:
  Per-base prediction, embedding extraction, contact map 제공
- 응용 분야:
  Genomic element prediction, variant effect, RNA structure 예측



핵심 성과

NTv3는 현재 가장 강력한 DNA foundation model 중 하나로, 다양한 게놈 태스크에 대해 state-of-the-art 성능을 달성하고 있습니다.

Evo 모델의 특징과 활용

7B/40B 파라미터, 생성 특화 모델

Evo는 Arc Institute와 Stanford에서 개발한 대규모 DNA 언어 모델입니다. StripedHyena 아키텍처를 사용하여 ultra-long context(131kb~1Mb)를 처리하며, 단순한 이해를 넘어선 '생성' 능력을 갖추고 있습니다.

핵심 특징: Generative Design

DNA, RNA, 단백질 서열을 통합 학습한 멀티모달 모델로, Autoregressive 방식을 통해 새로운 유전자 서열을 처음부터 설계할 수 있습니다.

Evo는 기존의 Transformer 한계를 넘어 긴 문맥을 학습함으로써, 합성 CRISPR 시스템 생성, 기능성 유전자 설계, Long-range regulatory element 예측 등 실제 생물학적 설계에 활용됩니다.

모델 선택 가이드

태스크별 최적 모델 선택 전략

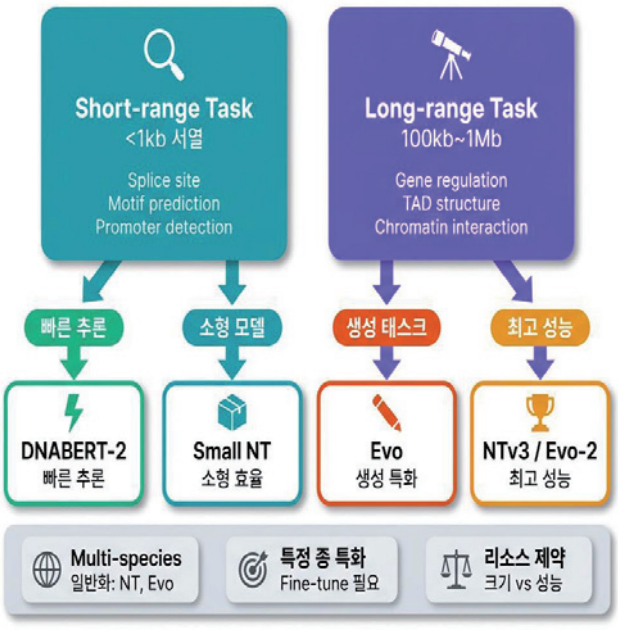
핵심 원칙

프로젝트의 목적에 맞는 모델을 선택하는 것은 성능 최적화의 첫걸음입니다. 서열의 길이와 분석 목적에 따라 적합한 아키텍처가 달라집니다.

Short-range Task는 국소적 패턴 인식에 강한 모델을, Long-range Task는 장거리 상호작용을 포착하는 모델을 선택해야 합니다.

트레이드오프

모델 크기와 추론 속도, 그리고 성능 사이의 균형을 고려하여 리소스 제약 내에서 최선의 선택을 하세요.



Embedding 추출과 활용

Foundation model의 핵심 활용법

DNA 언어모델을 활용하는 가장 기본적이고 강력한 방법은 모델의 중간 층에서 Embedding(임베딩)을 추출하는 것입니다.

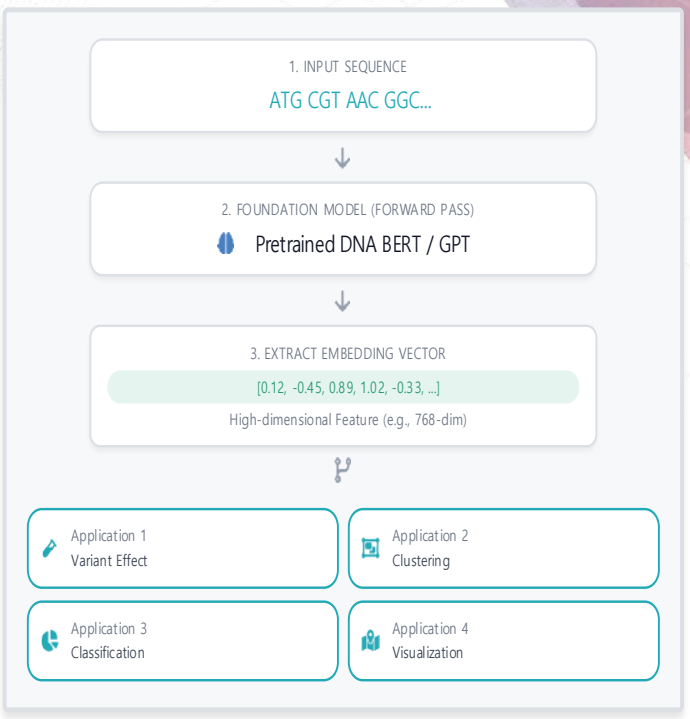
단순한 염기서열(A,T,G,C)을 모델에 통과시켜 얻은 고차원 벡터는 생물학적 문맥과 의미를 함축하고 있어, 다양한 분석의 입력값(Feature)으로 즉시 활용 가능합니다.

활용 포인트

간편함: 복잡한 추가 학습(Training) 없이 Pretrained 모델만으로 수행 가능

범용성: 하나의 임베딩으로 분류, 군집화, 시각화 등 여러 태스크 해결

해석 가능성: 벡터 공간 분석을 통해 모델이 학습한 생물학적 특징 파악



API와 웹 서비스 활용

Foundation model의 크기가 커짐에 따라, 로컬 환경 구축 대신 API를 활용하는 방식이 보편화되고 있습니다.

API 방식은 초기 진입 장벽을 낮추고 빠른 실험을 가능하게 하지만, 데이터 보안과 비용 측면에서 고려해야 할 점들이 있습니다.

프로젝트의 규모와 데이터의 민감도에 따라 가장 적합한 방식을 선택하는 것이 중요합니다.

API 기반

Cloud Service (원격 접근)

User → Internet → Cloud API

장점

- ✓ GPU 자원 불필요
- ✓ 복잡한 설치 없음
- ✓ 최신 모델 자동 적용

제약사항

- × 인터넷 연결 필수
- × 호출 비용 발생
- × 데이터 Privacy 이슈

로컬 설치

On-premise (직접 설치)

User → Local Server → GPU

장점

- ✓ 완전한 제어권/보안
- ✓ 호출 비용 없음 (무제한)
- ✓ 오프라인 환경 가능

제약사항

- × 고성능 GPU 필수
- × 환경 설정 복잡
- × 유지보수 부담

선택 가이드: 소규모 실험 및 빠른 시작은 API , 대량 처리 및 민감 데이터는 로컬 설치를 권장합니다.

해석 시 주의점

Foundation model의 예측 결과는 강력하지만, 완벽하지 않습니다. 모델의 예측을 최종 결론이 아닌 연구의 출발점으로 활용해야 하며, 항상 비판적인 시각으로 검증하는 과정이 필수적입니다.

모델 한계
Prediction Limitations

- 예측은 가설이지 증명이 아님
- 100% 신뢰 불가, 오류 가능성

편향성 인식
Bias Awareness

- 학습 데이터의 bias 반영
- 특정 Population 편향 주의

Wet Lab 검증
Experimental Validation

- 반드시 실험적 확인 필요
- In silico ≠ In vivo 환경 차이

생물학적 타당성
Biological Plausibility

- 생물학적 메커니즘 확인
- 기존 지식과 대조 및 검토

핵심 원칙

중요한 발견은 반드시 Wet Lab으로 확인해야 합니다. Computational prediction은 가설이지 증명이 아닙니다.

핵심 원칙: 모델 출력을 '출발점' 으로 활용, 항상 비판적 검증 이 필요합니다.

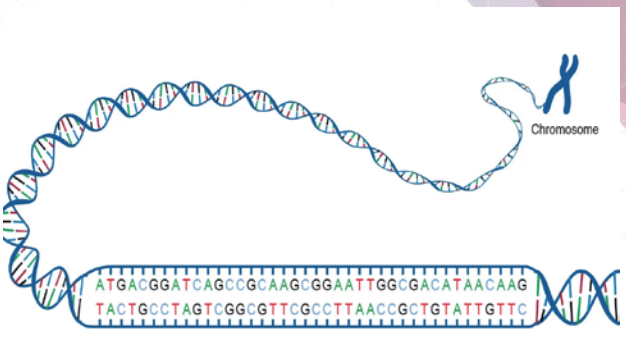
활용 사례:
Variant effect prediction

Embedding 거리 기반 pathogenicity 예측

Variant effect prediction은 embedding 차이로 수행할 수 있습니다. Wild-type 서열과 variant 서열의 embedding을 각각 구하여 두 embedding 사이의 거리를 계산합니다.

예측 원리

거리가 크면 변이의 효과가 크다고 해석하며, 거리가 작으면 변이가 미미한 영향만 준다고 판단합니다. Embedding은 문맥을 반영하므로 같은 염기 변화라도 위치에 따라 다른 거리를 갖습니다.



중요한 위치(splice site, start codon)의 변이는 큰 embedding 변화를, 중립적 위치의 변이는 작은 변화를 일으킵니다. 이를 통해 variant pathogenicity를 간접적으로 평가할 수 있습니다.

활용 사례: Regulatory Element 발견

Genome-wide 스캔으로 새로운 조절 요소 탐색

Foundation model은 전체 게놈을 sliding window 방식으로 스캔하여, 기존에 알려지지 않은 기능적 요소를 발굴하는 강력한 도구입니다.

핵심 탐색 전략

모델의 Embedding 또는 Activation 패턴을 분석하여 Promoter-like cluster를 형성하거나 Attention이 집중되는 영역을 감지합니다. 이들은 잠재적인 조절 요소(Candidate)로 간주됩니다.

핵심 가치: Annotation이 없는 non-coding region에서 Enhancer, Silencer, Insulator 등 새로운 regulatory element를 발견할 수 있습니다.

활용 사례: Synthetic sequence 설계

Autoregressive 생성을 통한 기능적 서열 설계

Evo 같은 generative model은 조건부로 새로운 DNA 서열을 생성합니다. 원하는 특성(강한 promoter, 특정 TF 결합, 코돈 최적화 등)을 조건으로 주면, 한 염기씩 확률적으로 생성합니다.

핵심 원리

생성된 서열의 기능을 computational로 예측하고, 유망한 후보를 실험실에서 합성하여 검증합니다. 이는 합성 생물학, 치료제 개발, CRISPR guide RNA 설계에 활용됩니다.

핵심 원칙:In silico 디자인과 실험 검증의 반복 사이클 (Design-Build-Test-Learn)로최적 서열을 도출합니다.

Cross-species comparative analysis

다양한 종의 유전체를 비교 분석하여 보존된 유전 요소를 식별하고, 종 간 유전자 기능의 차이를 이해합니다. 이는 진화적 관계와 생물학적 기능을 파악하는 핵심 도구입니다.

비교 유전체학

단일 종 분석의 한계를 넘어, 다종 게놈 비교를 통해 유전자 가족의 확장·축소와 종 특이적 적응을 분석합니다.

다종 데이터 수집 (Data Collection)
- Human, Mouse, Zebrafish
- 다양한 계통 게놈 확보

서열 정렬 (Alignment)
- Ortholog 매핑
- Multiple Alignment

보존 분석 (Conservation)
- PhyloP/PhastCons 점수
- Constraint 영역 식별

기능 주석 (Annotation)
- GO term 매핑
- 기능 및 Pathway 예측

진화적 추론 (Inference)
- 선택압(dN/dS) 계산
- 진화 모델 검증

핵심 원칙**진화적 보존** 은 생물학적 중요성의 강력한 증거입니다. 수억 년의 자연선택에서 살아남은 서열은 생명 유지에 필수적인 기능을 수행할 가능성이 매우 높습니다.

모델 접근 방법: Hugging Face

쉽게 사용 가능한 DNA Foundation Model

Hugging Face는 DNA foundation model의 허브입니다. Transformers 라이브러리를 통해 몇 줄의 코드로 최신 모델을 사용할 수 있습니다.

주요 장점

- 설치 및 환경 설정 불필요
- 최신 버전 자동 사용
- GPU가 있으면 더 빠른 추론 가능
- CPU로도 작은 서열은 처리 가능

Hugging Face Hub에서는 다양한 DNA 모델을 비교하고, 커뮤니티 리뷰를 확인하며, 직접 다운로드 없이 온라인으로 테스트할 수 있습니다.

주의사항

- 인터넷 연결 필요
- 대량 처리 시 비용 발생 가능
- 데이터 프라이버시 고려 필요
- 상업적 사용 시 라이선스 확인

학습 비용과 자원

Foundation model 학습의 경제적 실태

Foundation model을 직접 학습하는 것은 매우 비쌉니다. 수천~수만 GPU-hour가 필요하며, Evo-2 (40B) 같은 대형 모델 학습에는 수백 개 GPU로 몇 주가 걸립니다. 비용은 수십만~수백만 달러에 달할 수 있습니다.

대규모 학습 비용

따라서 대부분의 연구자는 Pretrained model을 사용합니다. Fine-tuning은 1-8개 GPU로 수 시간에서 며칠이면 가능하여 훨씬 경제적입니다.

처음부터 학습 (From Scratch)

$ 매우 고비용

- 수천만~수억 GPU-hours 소요
- 전문 인력 팀 및 대규모 클라우드 인프라 필요
- 대부분의 기업/연구소만 가능

미세 조정 (Fine-tuning)

$ 중간 비용

- 1-8개 GPU 필요, 수 시간~며칠 소요
- 대부분의 대학 연구실에서 수행 가능
- Task-specific 데이터 준비 필요

즉시 사용 (Zero-shot Inference)

✓ 최저 비용

- 추가 학습 불필요, 즉시 사용 가능
- CPU로도 처리 가능 (작은 모델)
- 인터넷 연결만 필요 (API 사용 시)

핵심 원칙: Pretrained model 활용으로 비용 효율성을 극대화하고, Zero-shot 으로 시작해 필요시 Fine-tuning 을 검토하는 것이 경제적입니다.

향후 발전 방향

DNA 언어모델은 단순히 크기를 키우는 것을 넘어, 생물학적 이해의 깊이와 폭을 모두 확장하는 방향으로 진화하고 있습니다.

Context 확장

현재의 킬로베이스(kb) 단위를 넘어 메가베이스(Mb), 전체 게놈 수준 처리를 목표로 합니다.

Multi-modal 통합

DNA 서열뿐만 아니라 RNA, 단백질 구조, 후성유전체 데이터 등을 통합하여 포괄적 모델을 구축합니다.

최종 목표

전체 유기체 수준의 시뮬레이션(Full organism simulation)을 통해 생명 현상을 디지털 재현합니다.

현재 (Current)

기반 구축 단계

- Nucleotide Transformer
- Evo (수백 kb context)
- 기초 문법 학습 완료

단기 (Near-term)

확장 및 통합 단계

- 1-10 Mb Context 확장
- Multi-modal 통합
- 유전자 조절 기작 이해

장기 (Long-term)

완전 시뮬레이션

- Genome-scale Model
- Full Organism Sim
- 표현형(Phenotype) 예측

Multi-species 학습의 핵심 가치 요약

진화적 신호 활용

수억 년의 자연선택이 필터링한 서열 정보를 활용하여 기능적 제약을 학습합니다. 보존된 영역은 중요하고, 변이가 많은 영역은 제약이 약함을 자동으로 파악합니다.

일반화 능력 향상

다양한 종의 데이터로 학습하면 새로운 종에 대한 예측 능력이 향상됩니다. Zero-shot prediction이 가능해지며, foundation model의 핵심 가치를 제공합니다.

자연스러운 Regularization

한 종에만 특화된 패턴은 다른 종에서 높은 loss를 유발하여 과적합을 방지합니다. 종을 넘나드는 보편적 원리만 낮은 loss로 보상받도록 합니다.

생물학적 발견 가능

Multi-species 학습은 인간 게놈에만 존재하지 않는 새로운 기능적 요소를 발견할 수 있습니다. 진화적으로 보존된 미지의 regulatory element를 탐색합니다.

향후 연구 방향 및 실무 가이드

학습 목표: DNA 언어모델의 주요 도전과제와 향후 연구 방향을 파악하고, 실무 적용을 위한 모델 선택 및 평가 가이드라인을 익힙니다.

Embedding의 생물학적 의미

수백 차원의 벡터가 담은 생명의 언어

Embedding은 단순한 숫자 벡터가 아닙니다. 이 고차원 공간은 생물학적 의미를 담고 있는 지도와 같습니다.

MLM 학습을 통해 모델은 DNA 서열 간의 복잡한 관계를 기하학적 구조로 변환하여 저장합니다.

핵심 통찰

블랙박스(Blackbox) 내부에서 자연스럽게 형성된 이 구조는, 우리가 생물학적 기능을 이해하는 데 있어 가장 강력한 해석 도구입니다.

Distance (거리) 유사도

벡터 공간에서의 거리는 생물학적 기능의 유사도를 의미합니다.

예시: Promoter variants 클러스터링

Direction (방향) 의미축

특정 방향으로의 이동은 생물학적 속성(예: 안정성, 활성)의 변화를 나타냅니다.

예시: Evolution / Function axis

Clustering (군집) 기능 범주

유사한 기능을 가진 서열들은 공간 상에서 밀집하여 고유한 영역을 형성합니다.

예시: Regulatory elements vs Coding

이러한 Embedding 구조는 Variant effect prediction ,Attribution ,Functional annotation 등 Downstream 분석의 출발점이 됩니다.

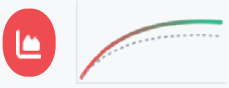
학습 과정 모니터링의 중요성

최종 성능 지표만으로는 학습 과정에서 발생하는 미묘한 문제를 감지할 수 없습니다. DNA 언어모델은 학습 비용이 높기 때문에, 초기 단계에서의 이상 징후 포착이 프로젝트의 성패를 좌우합니다.

핵심 통찰

"문제를 조기에 발견하면 시간과 자원을 절약할 수 있습니다."

Loss Curve와 Gradient 등의 동역학(Dynamics) 지표는 모델 내부의 상태를 보여주는 MRI와 같습니다.



Loss Curve 학습 추세

가파른 하강 후 안정화되는 패턴(L자형)이 이상적입니다. Train/Valid 차이로 Overfitting을 감지합니다.

Learning Rate 스케줄링

Warmup으로 초기 불안정성을 잡고, Cosine Decay로 수렴을 돕습니다. 적절한 LR 탐색이 필수입니다.

Gradient Norm 안정성

기울기 폭발(Exploding)이나 소실(Vanishing) 없이 일정 범위 내에서 유지되는지 확인해야 합니다.

성공적인 학습의 징후: Loss는 지속적으로 감소하고, Gradient Norm은 튀지 않으며, Validation 성능이 꾸준히 향상되는 상태를 유지해야 합니다.

성능 지표 선택의 3대 원칙

1. 문제 특성 파악

Class balance 비율을 먼저 확인합니다. Imbalance 정도에 따라 AUROC vs AUPRC를 선택합니다.

2. 응용 목적 정의

FN이 치명적인지, FP가 치명적인지 명확히 합니다. Precision vs Recall 중요도를 결정합니다.

3. 여러 지표 종합

단일 지표에 의존하지 않고, Precision, Recall, F1, AUPRC를 함께 제시합니다.

Pretrained 모델의 활용 전략

범용 모델을 효과적으로 활용하는 방법

Foundation model은 대규모 데이터로 사전 학습된 범용 모델로, 다양한 태스크에 즉시 재사용하거나 맞춤화할 수 있습니다.

프로젝트의 목표, 가용 데이터, 리소스 상황에 따라 최적의 활용 경로를 선택하는 것이 성공의 핵심입니다.

전략 수립 가이드

목적에 따라 Zero-shot으로 빠르게 가능성을 검증한 후, 성능 극대화가 필요할 때 Fine-tuning으로 전환하는 단계적 접근을 권장합니다.

1. Zero-shot (즉시 사용)

Instant Use

추가 학습 없이 사전 학습된 지식을 그대로 사용하여 예측을 수행합니다. 모델의 범용적인 이해력을 활용합니다.

✓비용 최소 ✓즉시 실행 특화 성능 한계 탐색적 분석 / POC

2. Fine-tuning (맞춤 학습)

Custom Training

특정 태스크(Task-specific) 데이터로 모델 파라미터를 미세 조정하여 해당 도메인에 최적화된 성능을 확보합니다.

✓최고 성능 달성 데이터/자원 필요 Production 시스템

3. API 활용 (클라우드)

Cloud Service

Hugging Face Inference API 등 외부 서비스를 호출하여 모델을 직접 운용하는 부담 없이 기능을 활용합니다.

✓인프라 불필요 데이터 보안 이슈 웹 서비스 / 중규모

DNA 언어모델의 실전 적용 전략

실전 적용 전략

DNA 언어모델의 실제 적용은 단순한 모델 선택이 아닌, 목적에 따른 전략적 접근이 필요합니다. 데이터 특성, 컴퓨팅 자원, 정확도 요구사항을 종합적으로 고려해야 합니다.

핵심 전략

- Zero-shot로 빠른 프로토타입 → Fine-tuning으로 성능 향상
- Multi-species 모델로 일반화 → 종 특화 파인튜닝
- Embedding 활용으로 다양한 downstream 태스크 지원
- API 활용으로 대규모 모델 접근성 확보

1단계: 프로토타이핑 (Prototyping)
Zero-shot 예측으로 가능성 검증 및 베이스라인 설정

2단계: 모델 최적화 (Optimization)
데이터 특성에 맞춘 Fine-tuning 또는 Embedding 학습

3단계: 시스템 통합 (Integration)
API 배포 및 Downstream 파이프라인 연동

실험적 검증을 통한 단계적 적용이 중요하며, 해석 가능성과 윤리적 고려사항을 함께 고려해야 합니다.

연구 과정에서의 3대 도전과제

1. 데이터 품질과 균형

고품질의 다양한 계통 데이터 확보, 클래스 불균형 해결, 표본 편향 최소화가 핵심 도전입니다. 특히 희귀 변이나 특정 인구 집단 데이터는 충분히 확보하기 어렵습니다.

2. 모델 복잡성과 해석성

딥러닝 모델의 블랙박스 특성, 수백만 개의 파라미터, 복잡한 내부 표현은 생물학적 의미를 도출하는 데 어려움을 줍니다.

3. 실험적 검증의 어려움

생물학적 실험은 시간과 비용이 많이 들며, 예측 결과를 실제로 검증하기 어렵습니다. 계산적 예측과 실험 결과 간 격차를 해결해야 합니다.

데이터 vs 모델 복잡도의 균형

데이터 복잡성 증가

Multi-species 데이터는 전역적 구조(유전자 순서, intron 길이)가 종마다 다릅니다. 모델은 이 차이를 포용하면서도 공통 원리를 찾아야 합니다.

구조적 진화

실제로 multi-species로 학습한 모델(NT, Evo)은 더 깊고 넓은 구조를 갖습니다. 이는 데이터의 복잡성을 다루기 위함입니다.

모델 표현력 요구

따라서 더 유연하고 강력한 표현이 필요합니다. 파라미터 수, context window, 층 수가 증가합니다.

데이터가 구조를 결정

결국 "데이터가 구조를 결정한다"는 원리가 적용됩니다. 복잡한 데이터는 복잡한 모델을 요구합니다.

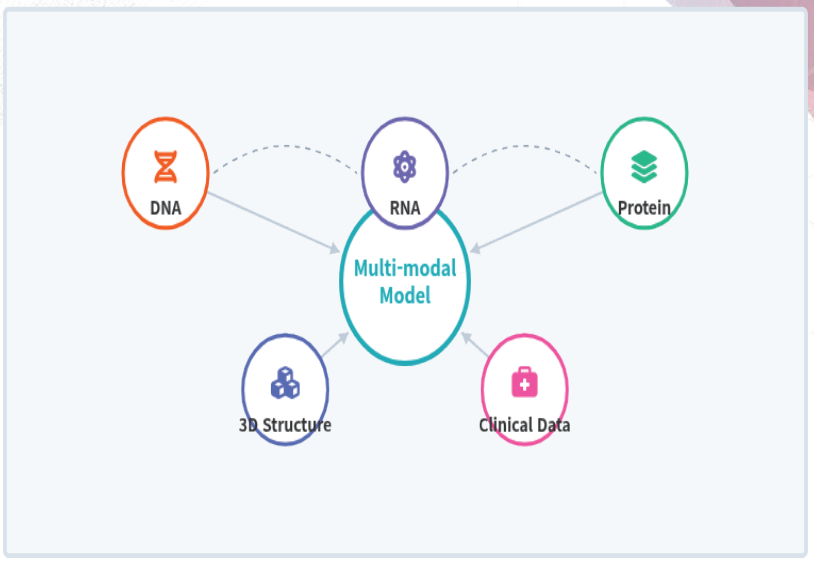
향후 연구 방향

멀티모달 통합

DNA 언어모델의 다음 단계는 DNA, RNA, 단백질, 후성유전체, 3D 구조, 임상 데이터를 통합한 멀티모달 모델을 개발하는 것입니다. 이는 생명현상의 완전한 이해를 위한 필수적인 접근입니다.

통합 전략

각 분야의 특성을 유지하면서도 상호작용을 학습하는 하이브리드 아키텍처가 필요합니다. 이는 단순한 데이터 결합이 아닌, 생물학적 상호작용의 본질을 이해하는 것을 목표로 합니다.



통합 모델은 단일 모델의 한계를 극복하고 종합적인 생물학적 인사이트를 제공하여 질병 이해에 기여할 것입니다.

튜토리얼 실습

이 강의를 통해 습득한 DNA 언어모델의 기초 지식을 바탕으로, 이제 실전 연구 프로젝트로 나아갈 차례입니다.

데이터 수집부터 모델 검증까지, 체계적인 3단계 접근 방식을 통해 시행착오를 줄이고 연구의 완성도를 높일 수 있습니다.

Start Small

거창한 목표보다는 작은 가설 검증부터 시작하세요.
각 단계에서의 피드백 루프가 성공의 핵심입니다.

튜토리얼 실습

- Tutorial 1: DNA Language Model의 이해 (바로가기 링크):
  https://drive.google.com/file/d/1y2nK2krHGUpS-ej268pHnNHSmmU5G0AH/view?usp=sharing
- Tutorial 2: DNA Language Model을 활용한 TFBS 예측 실습 (바로가기 링크):
  https://drive.google.com/file/d/1E0ZgE1C7Yylf7hos2Pmaa1WNMGk7FDgx/view?usp=sharing
- Tutorial 3: 유전 변이 효과 예측과 장거리 상호작용 분석 (바로가기 링크):
  https://drive.google.com/file/d/1X3b6VcQZ88ujRfEEdeckN9HOxuw1c7Xo/view?usp=sharing
- Tutorial 4: NTv3 Post-trained Model for Genome Annotation (바로가기 링크):
  https://colab.research.google.com/drive/16-llUuOu0pUlq2JU7gW-XAv15FDH1FcH?usp=sharing

THANK YOU

DNA Language Model 강의를 마치며