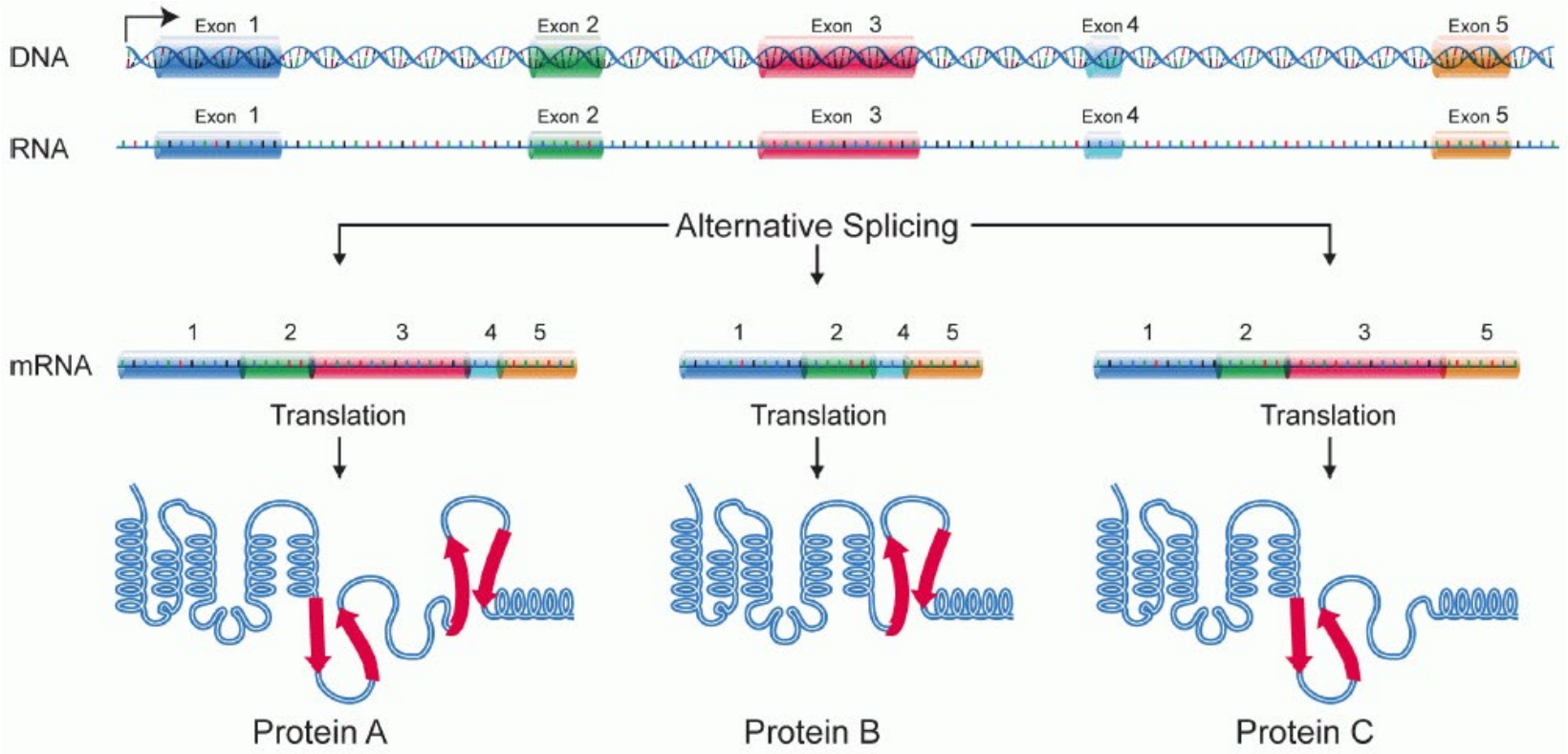
DNA Models

October 17, 2024

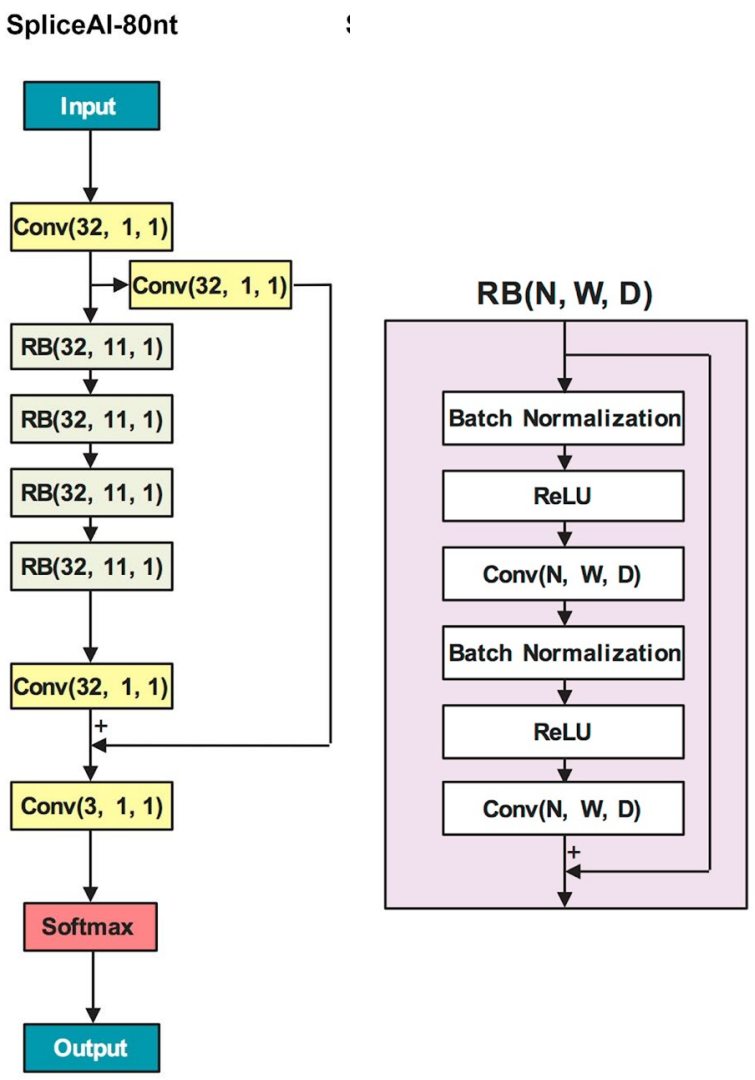
RNA Splicing



RNA Splice Site Prediction

- **Definition**: Predicting RNA splice sites to identify where splicing occurs in pre-mRNA.
- **Importance**: Crucial for understanding gene expression, alternative splicing, and genetic disorders.
- **Splice Sites**:
  - **Donor (5' GT)**
  - **Acceptor (3' AG)**
- **Challenges**: Sequence variability and alternative splicing add complexity.
- **Approaches**:
  - **Machine Learning**: SVM, Random Forest using sequence features.
  - **Deep Learning**: CNNs, RNNs for raw sequence data.
- **Applications**: Gene annotation, disease research (e.g., cancer).
- **Tools**: **SpliceAI**, **GeneSplicer**, **MaxEntScan**.

SpliceAI


SpliceAI-80nt
Input
Conv(32, 1, 1)
Conv(32, 1, 1)
RB(32, 11, 1)
RB(32, 11, 1)
RB(32, 11, 1)
RB(32, 11, 1)
Conv(32, 1, 1)
+
Conv(3, 1, 1)
Softmax
Output
RB(N, W, D)
Batch Normalization
ReLU
Conv(N, W, D)
Batch Normalization
ReLU
Conv(N, W, D)
+

What is a convolution?

- **Definition**: Convolution combines two functions to produce a third, showing how one modifies the other.
- **Mathematical Form**:
  Measures overlap between f and a reversed, shifted g.
- **Purpose in Deep Learning**: Extracts features by applying filters (kernels) to data in CNNs.
- **Filter (Kernel)**: Slides across input, performing element-wise multiplication and summation.
- **Advantages**:
  - **Parameter Sharing**: Fewer parameters, more efficient.
  - **Translation Invariance**: Recognizes features regardless of position.
- **Applications**: Signal processing, images, spatial data.

AMERICAN SOCIETY FOR MICROBIOLOGY | mSystems®

RESEARCH ARTICLE
Applied and Environmental Science

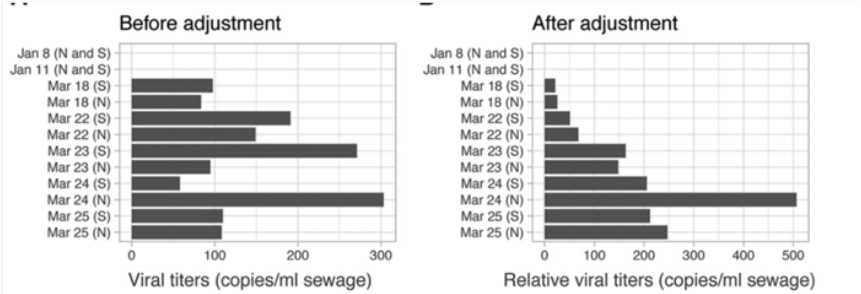
SARS-CoV-2 Titers in Wastewater Are Higher than Expected from Clinically Confirmed Cases

Fuqing Wu,[a,n,o] Jianbo Zhang,[a,n] Amy Xiao,[a,n,o] Xiaoqiong Gu,[b,m] Wei Lin Lee,[b,m] Federica Armas,[b,m] Kathryn Kauffman,[c] William Hanage,[d] Mariana Matus,[e] Newsha Ghaeli,[e] Noriko Endo,[e] Claire Duvallet,[e] Mathilde Poyet,[a,n,o] Katya Moniz,[a,n,o] Alex D. Washburne,[p] Timothy B. Erickson,[f,g] Peter R. Chai,[f,h,i] Janelle Thompson,[j,k,m] Eric J. Alm[a,b,e,l,m,n,o]

```
>Northern-Forward    ------------------------------------------------
>SARS-CoV-2 S gene   GAAATTAATACGACTCACTATAGGGAGGTTTCAAACTTTACTTGCTTTAC  50
>Northern-Reverse    GAAATTAATACGACTCACTATAGGGAGGTTTCAAACTTTACTTGCTTTAC

>Northern-Forward    --------------ACTNCTGNNGNNTCTTCTTNNGGTTGGACAGCTGGT
>SARS-CoV-2 S gene   ATAGAAGTTATTTGACTCCTGGTGATTCTTCTTCAGGTTGGACAGCTGGT  100
>Northern-Reverse    ATAGAAGTTATTTGACTCCTGGTG-NTCTTCTTCAGGTNGAC--------

>Northern-Forward    GCTGCAGCTTATTATGTGGGTTATCTTCAACCTAGGA
>SARS-CoV-2 S gene   GCTGCAGCTTATTATGTGGGTTATCTTCAACCTAGGA              137
>Northern-Reverse    -------------------------------------
```



Example problem

A new virus appears in Boston

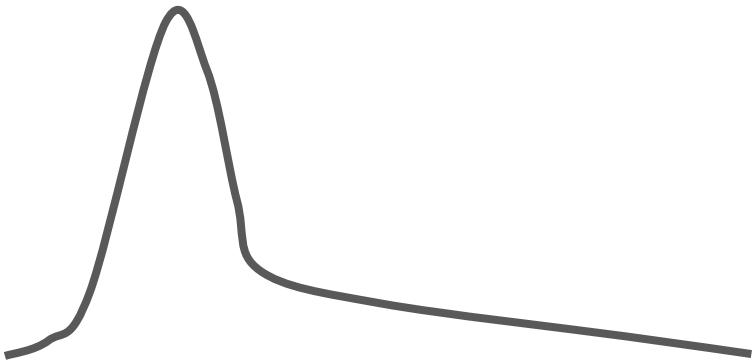
Each day, every infected individual excretes 100 units of the virus in wastewater

Each new infection lasts 5 days.

New infections I(t) = {10, 25, 150, 50, 25, 0, 0, 0, 0, 0}

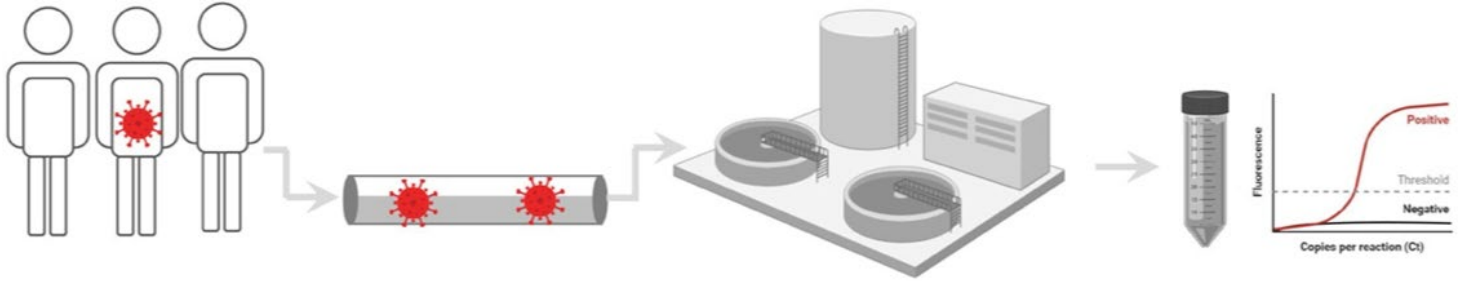
How does the amount of virus in wastewater vary over time V(t)?

How does this change if infected individuals give off 200 units on the first day, and 100 units the next four days?



True shedding function

Observed shedding function



New infection cases $I(t)$
Mean viral shedding $S(t)$

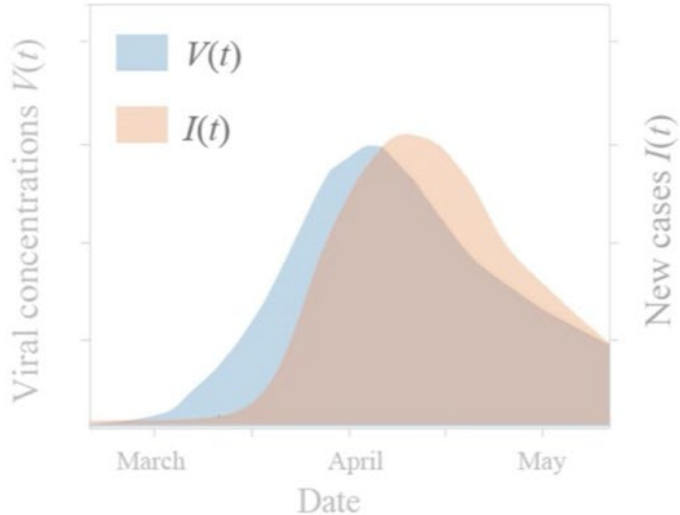
Viral copies in wastewater
$W(t)$

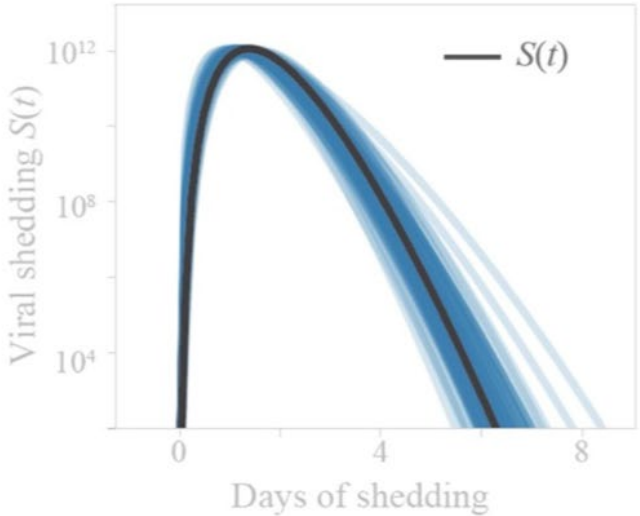
Viral concentrations
$V(t)$

$$V(t) \propto I(t)$$

$$S(t) = Beta(t,\alpha,\beta)$$

$$W(t) = S(t) * I(t)$$





Discrete Convolution

Input - an image or sequence

Kernel - matrix of weights that slide across image

Feature map - output of a convolutional layer

Width - size of kernel matrix

Stride - step size over input data

Padding - additional data (zeros) added to edges around input data

CONVOLUTION IN ACTION
Problem Setup

Padding: None
Stride: 1

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9

9

Kernel

-1	0	1
-1	0	1
-1	0	1

3

3

Convolved Image

?

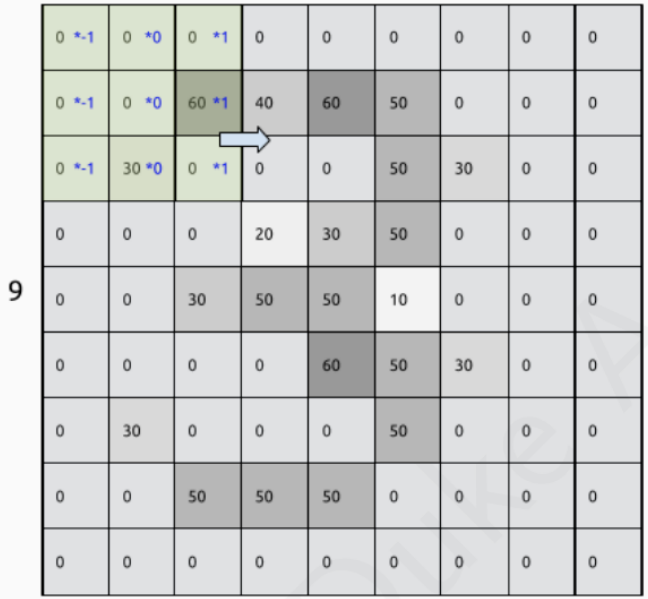
Duke Applied ML

CONVOLUTION IN ACTION

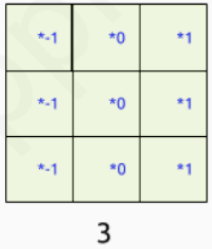
Step 1

Padding: None
Stride: 1

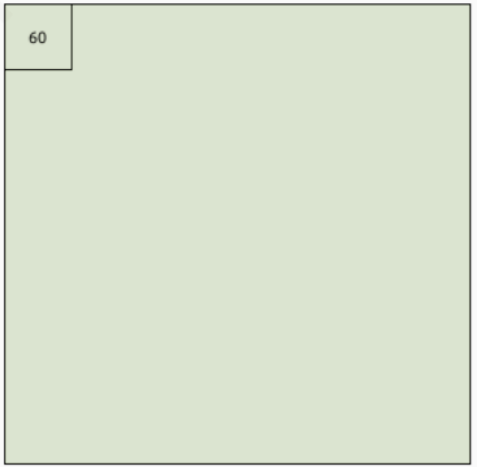
Input Image of Handwritten 3



Kernel



Convolved Image



Calculation

0*(-1) + 0*(-1) + 0*(-1) + 0*(0) + 60*(-1) + 0*(-1) + 30*(0) + 0*(1) = -60

https://dukephoenixproject.notion.site/Deep-Learning-Theory-and-Practice-7281ca43ddeb43debbddd46162fe5ea4

CONVOLUTION IN ACTION

Step 2

Padding: None
Stride: 1

Input Image of Handwritten 3

0	0 *-1	0 *0	0 *1	0	0	0	0	0
0	0 *-1	60 *0	40 *1	60	50	0	0	0
0	30*-1	0 *0	0 *1	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9 (rows) × 9 (columns)

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 × 3

Convolved Image

60	10

Calculation

0(-1) + 0(0) + 0(1) + 0(-1) + 60(0) + 40(1) + 30(-1) + 00(0) + 0(1) = 10

Duke Applied ML

CONVOLUTION IN ACTION

Step 38-42

Padding: None
Stride: 1

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0 *-1	0 *0	0 *1
0	0	50	50	50	0	0 *-1	0 *0	0 *1
0	0	0	0	0	0	0 *-1	0 *0	0 *1

9 (rows) × 9 (columns)

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 × 3

Convolved Image

60	10	0	60	-30	-100	-30
60	30	30	90	-30	-150	-30
30	40	50	40	-50	-110	-30
30	70	110	40	-110	-150	-30
30	20	80	60	-80	-110	-30
50	20	60	50	-80	-100	-30
50	20	0	0	-50	-50	0

Step 42
Calculation

$$0(-1) + 0(0) + 0(1) + 0(-1) + 0(0) + 0(1) + 0(-1) + 0(0) + 0(1) = 0$$

Duke Applied ML

CONVOLUTION IN ACTION

Step 38-42

Padding: None
Stride: 1

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9 (rows) × 9 (columns)

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 × 3

Convolved Image

60	10	0	60	-30	-100	-30
60	30	30	90	-30	-150	-30
30	40	50	40	-50	-110	-30
30	70		40	-110	-150	-30
30	20	80	60	-80	-110	-30
50	20	60	50	-80	-100	-30
50	20	0	0	-50	-50	0

Visualizing the Convolution

Can we explain the activations in the convolved image?

Duke Applied ML

CONVOLUTION IN ACTION

Step 38-42

Padding: None
Stride: 1

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9 (rows) × 9 (columns)

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 × 3

Convolved Image

60	10	0	60	-30	-100	-30
60	30	30	90	-30	-150	-30
30	40	50	40	-50	-110	-30
30	70		40	-110	-150	-30
30	20	80	60	-80	-110	-30
50	20	60	50	-80	-100	-30
50	20	0	0	-50	-50	0

Visualizing the Convolution

The greatest magnitude of activations happen along the **vertical edges.**

Duke Applied ML

CONVOLUTION WITH STRIDE 2 IN ACTION

Step 2

Padding: None
Stride: 2

Input Image of Handwritten 3

0 *-1	0 *0	0 *1	0	0	0	0	0	0
0 *-1	0 *0	60 *1	40	60	50	0	0	0
0 *-1	30 *0	0 *1	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

Stride 2

9

9

Kernel

-1	0	1
-1	0	1
-1	0	1

3

3

Convolved Image

60	

How does changing the stride size affect convolution?

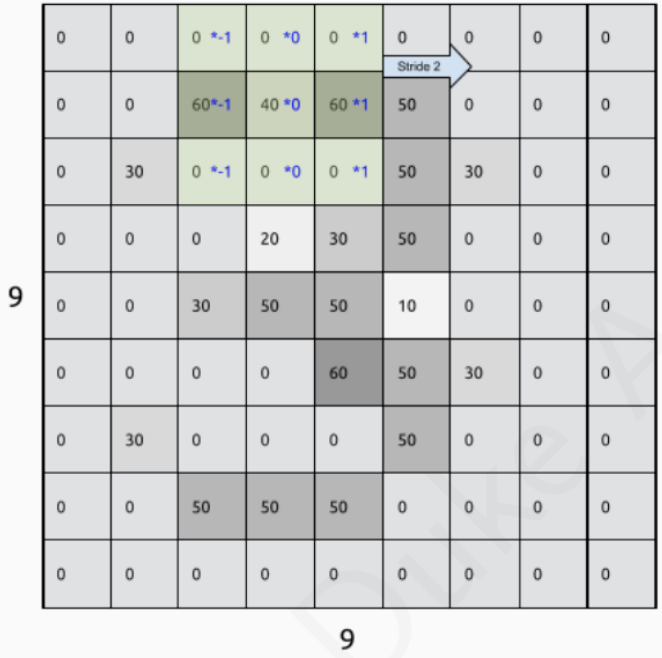
Duke Applied ML

CONVOLUTION WITH STRIDE 2 IN ACTION

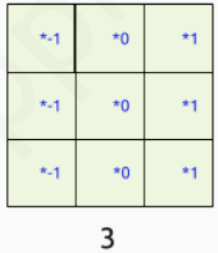
Step 2

Padding: None
Stride: 2

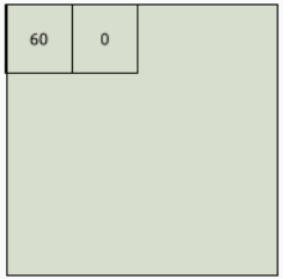
Input Image of Handwritten 3



Kernel



Convolved Image



Duke Applied ML

https://dukephoenixproject.notion.site/Deep-Learning-in-Theory-and-Practice-7281ca43ddeb43debbddd46162fe5ea4

CONVOLUTION WITH STRIDE 2 IN ACTION

Step 7-16

Padding: None
Stride: 2

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9 (rows) × 9 (columns)

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 × 3

Convolved Image for Stride 2

60	0	-30	-30
30	50	-50	-30
30	80	-80	-30
50	0	-50	0

Convolved Image for Stride 1

60	10	0	60	-30	-100	-30
60	30	30	90	-30	-150	-30
30	40	50	40	-50	-110	-30
30	70	110	40	-110	-150	-30
30	20	80	60	-80	-110	-30
50	20	60	50	-80	-100	-30
50	20	0	0	-50	-50	0

How does our convolution with stride 1 relate to the one with stride 2?

Duke Applied ML

CONVOLUTION WITH STRIDE 2 IN ACTION

Step 7-16

Padding: None
Stride: 2

Input Image of Handwritten 3

0	0	0	0	0	0	0	0	0
0	0	60	40	60	50	0	0	0
0	30	0	0	0	50	30	0	0
0	0	0	20	30	50	0	0	0
0	0	30	50	50	10	0	0	0
0	0	0	0	60	50	30	0	0
0	30	0	0	0	50	0	0	0
0	0	50	50	50	0	0	0	0
0	0	0	0	0	0	0	0	0

9

9

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3

3

Convolved Image for Stride 2

60	0	-30	-30
30	50	-50	-30
30	80	-80	-30
50	0	-50	0

Convolved Image for Stride 1

60	10	0	60	-30	-100	-30
60	30	30	90	-30	-150	-30
30	40	50	40	-50	-110	-30
30	70	110	40	-110	-150	-30
30	20	80	60	-80	-110	-30
50	20	60	50	-80	-100	-30
50	20	0	0	-50	-50	0

How does our convolution with stride 1 relate to the one with stride 2?

The values in yellow all match to the activations in the convolution with stride 2.

Duke Applied ML

CONVOLUTION WITH PADDING DIAGRAM

Padding: 1
Stride: 1

Input Image of Handwritten 3

0 *-1	0 *0	0 *1	0	0	0	0	0	0	0	0
0 *-1	0 *0	0 *1	0	0	0	0	0	0	0	0
0 *-1	0 *0	0 *1	60	40	60	50	0	0	0	0
0	0	30	0	0	0	50	30	0	0	0
0	0	0	0	20	30	50	0	0	0	0
0	0	0	30	50	50	10	0	0	0	0
0	0	0	0	0	60	50	30	0	0	0
0	0	30	0	0	0	50	0	0	0	0
0	0	0	50	50	50	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Kernel

*-1	*0	*1
*-1	*0	*1
*-1	*0	*1

3 (rows) × 3 (columns)

Convolved Image with Padding=1

What would be the dimension of the convolved image be?

It is common practice to pad the image **(n-1)/2** where n is the length/width of kernel. Since the kernel above has a length of 3, we set padding = 1.

This ensures the output of the convolution **maintains the dimensions of the input space.**

Duke Applied ML

https://dukephoenixproject.notion.site/Deep-Learning-in-Theory-and-Practice-7281ca43ddeb43debbddd46162fe5ea4

Convolution output size

Lin = size of input image or feature map

K = kernel size

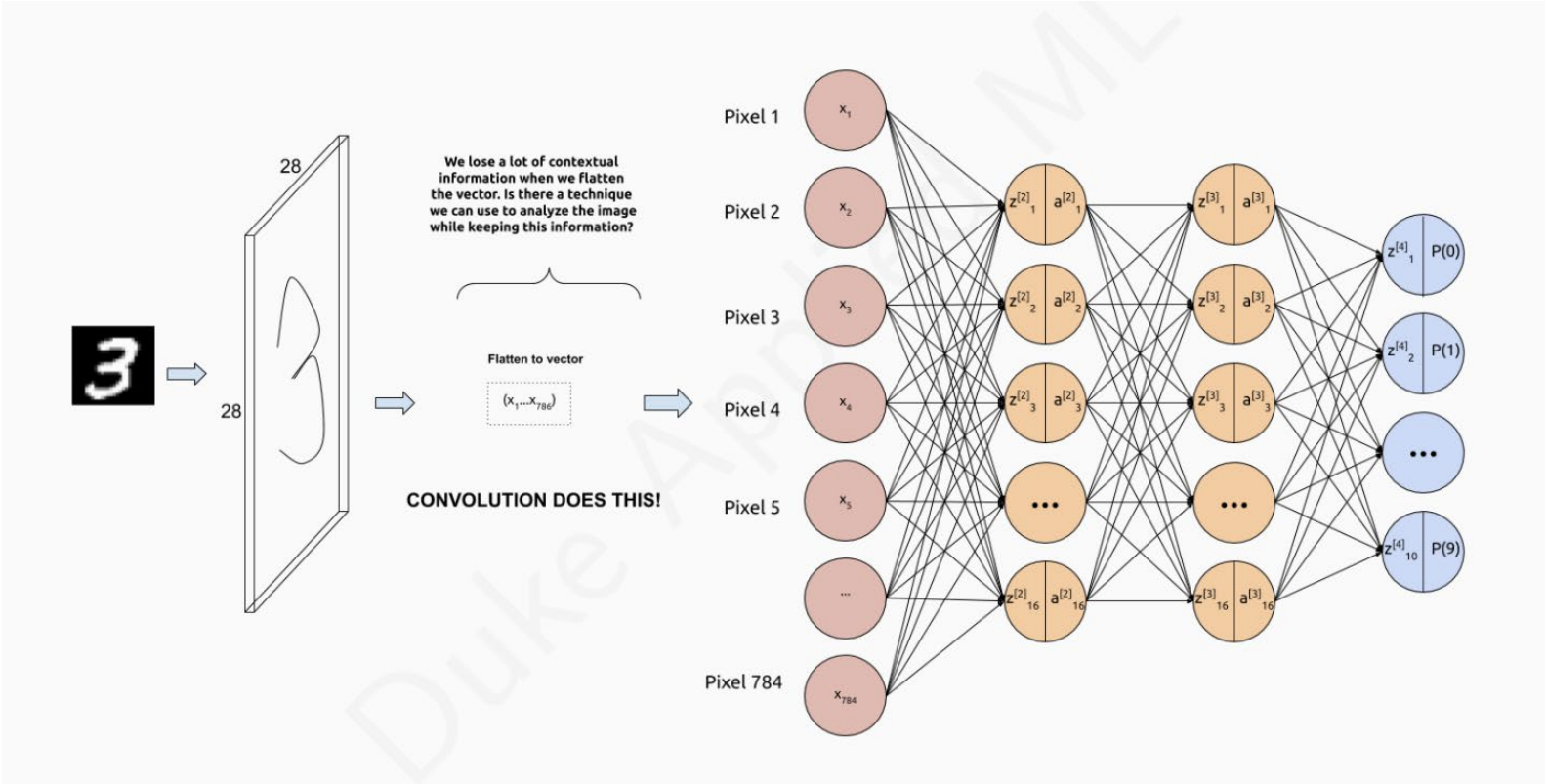
S = stride

P = padding

Lout = size of output feature map

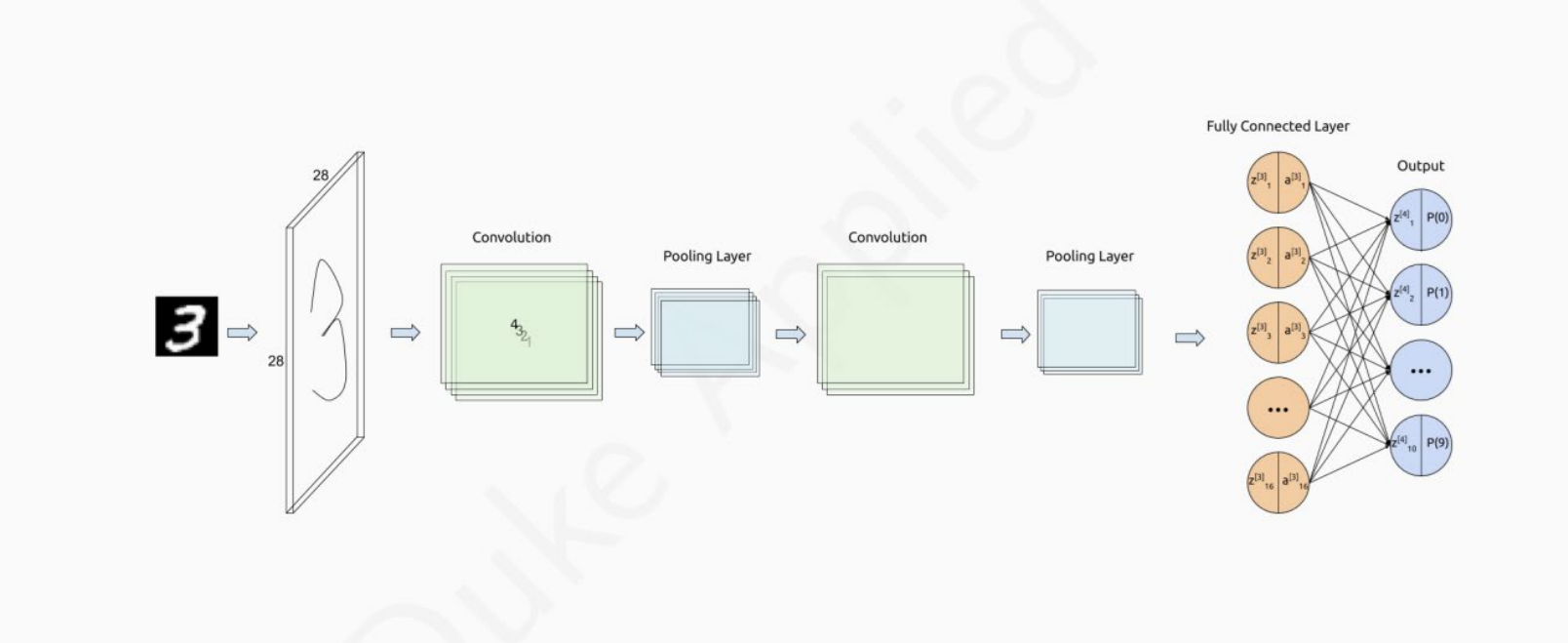
$$\mathbf{Lout = (Lin - K + 2P)/S + 1}$$

Fully connected layers have many parameters



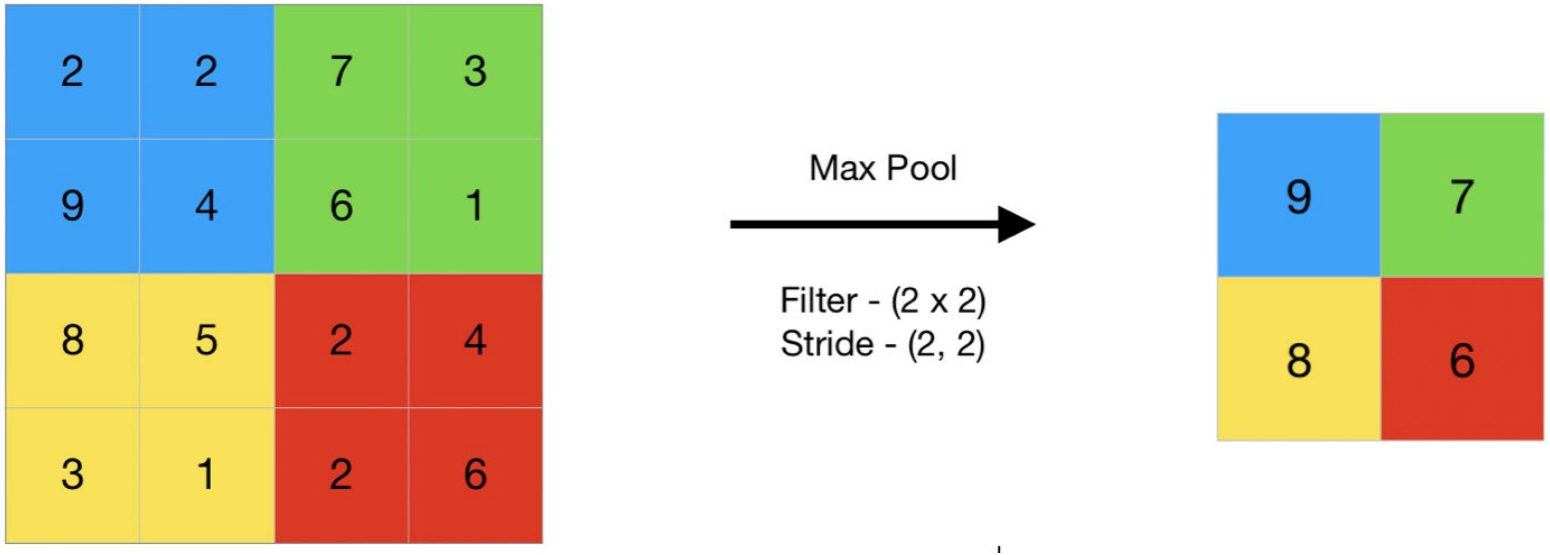
https://dukephoenixproject.notion.site/Deep-Learning-in-Theory-and-Practice-7281ca43ddeb43debbddd46162fe5ea4

Convolutional neural networks



https://dukephoenixproject.notion.site/Deep-Learning-in-Theory-and-Practice-7281ca43ddeb43debbddd46162fe5ea4

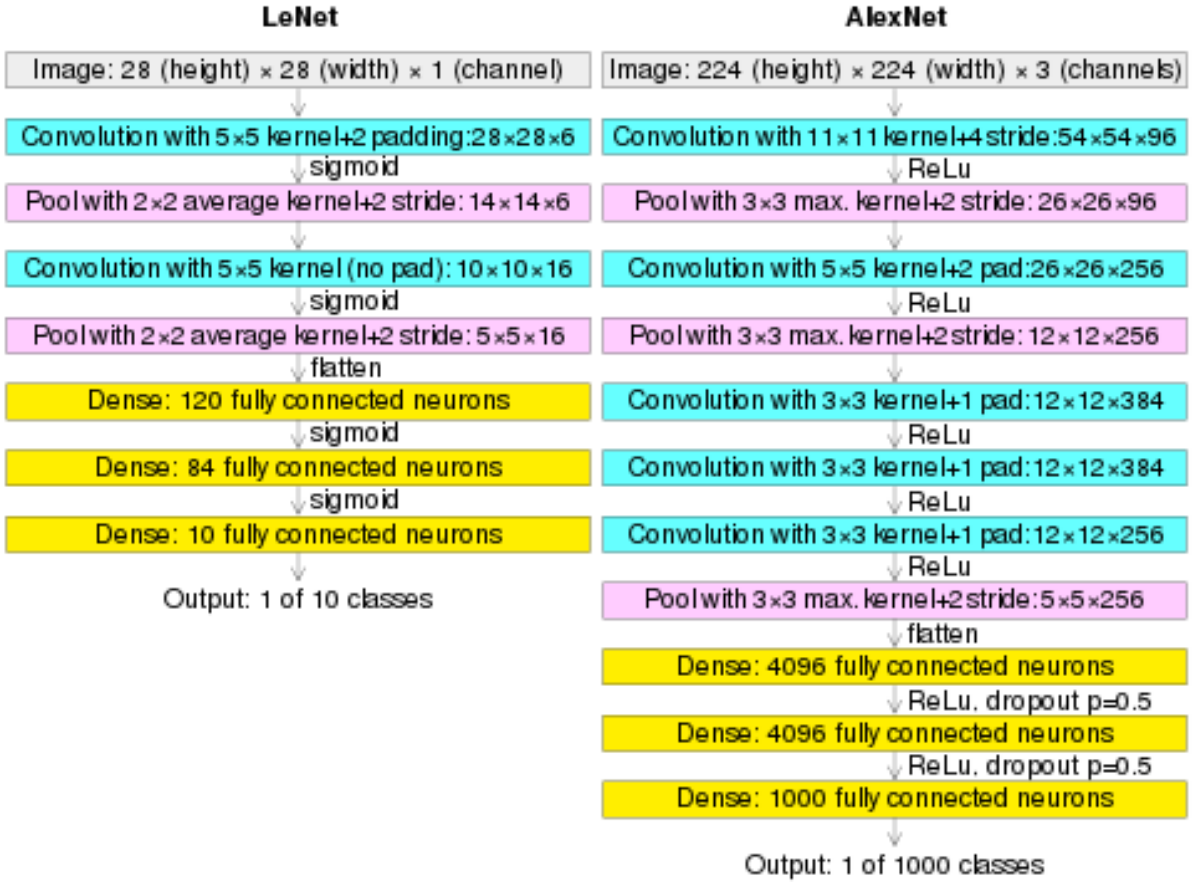
Max Pooling



Convolution

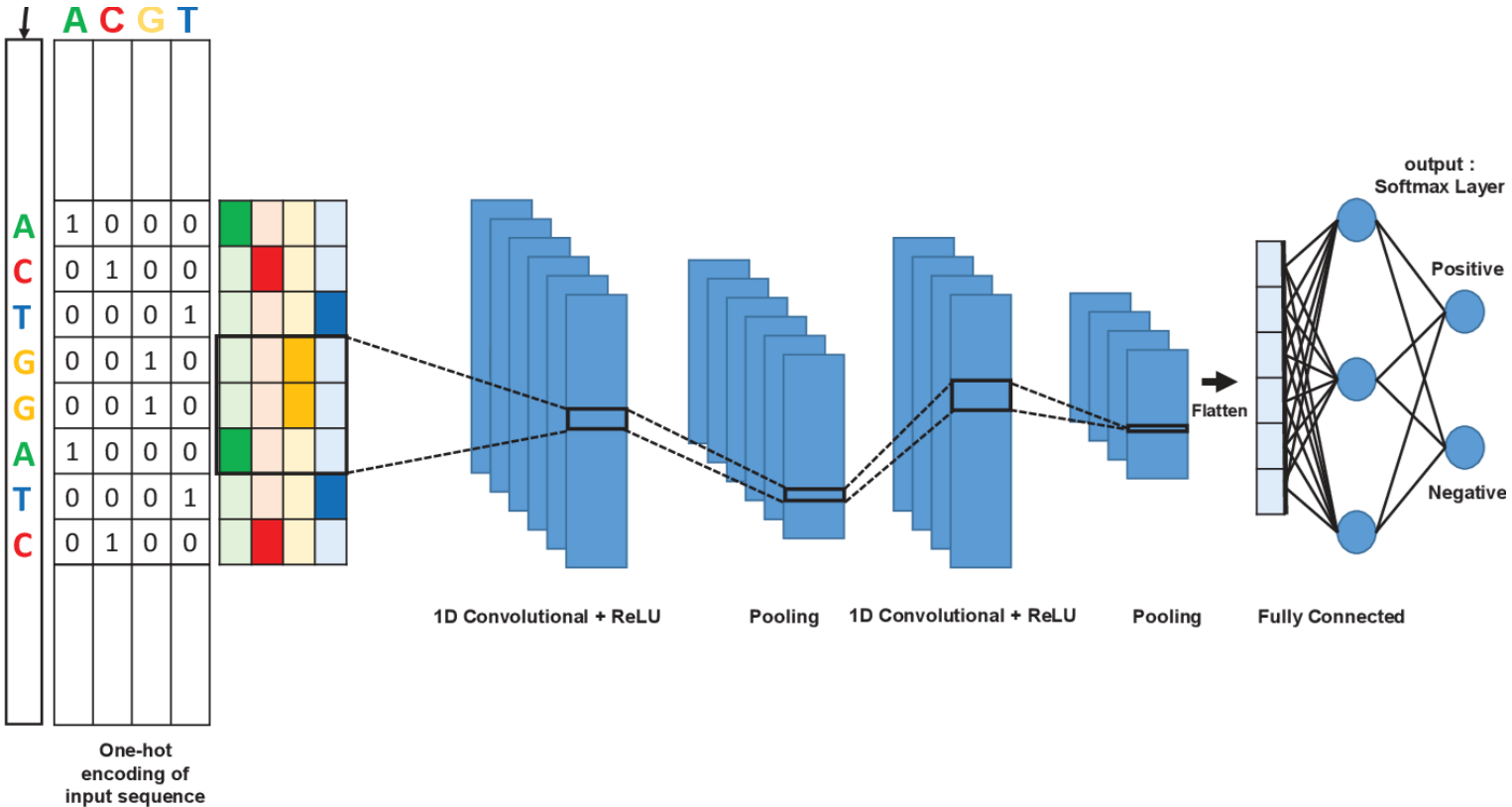
Yann Le Cun 1989 - used back prop to train a CNN on handwritten digits

Alex Krizhevsky 2012 - used a CNN trained on GPUs to win ImageNet contest by a large margin

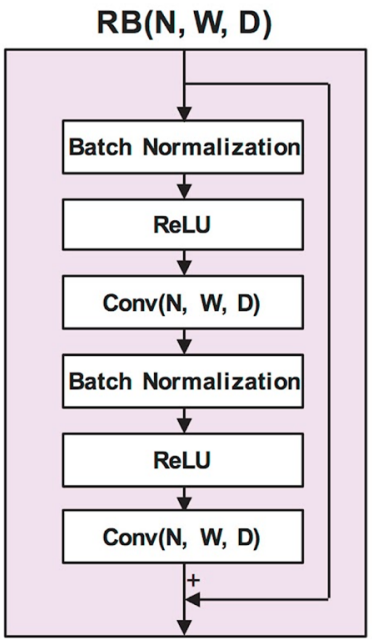


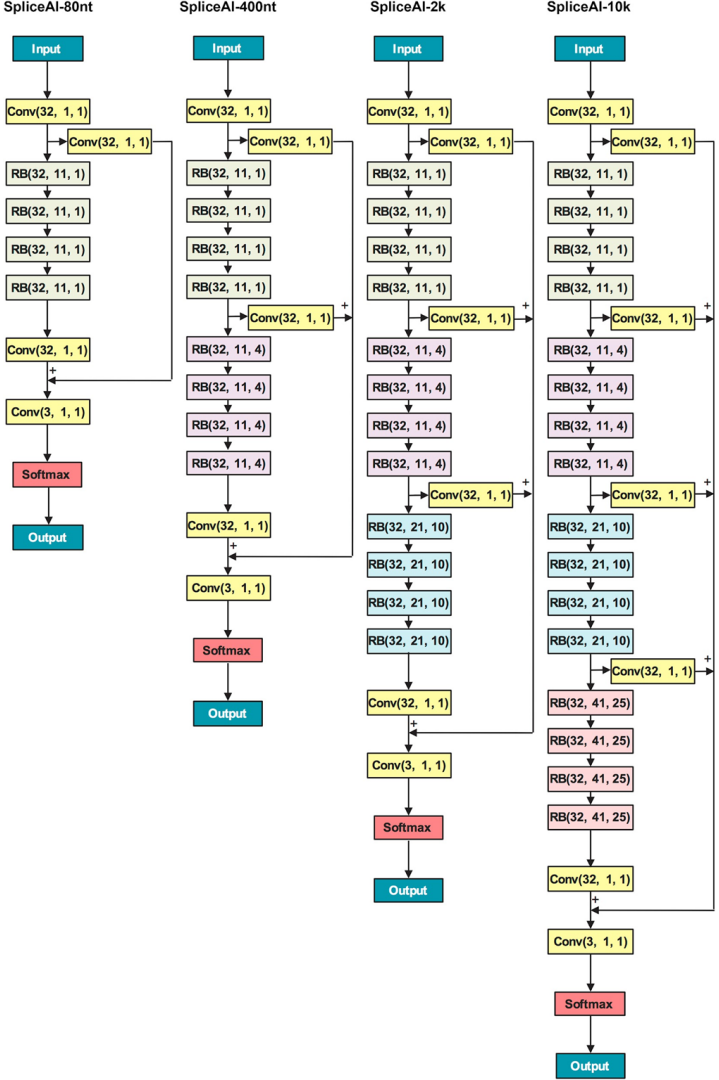
http://wikipedia.org

1d convolution for DNA sequence



SpliceAI

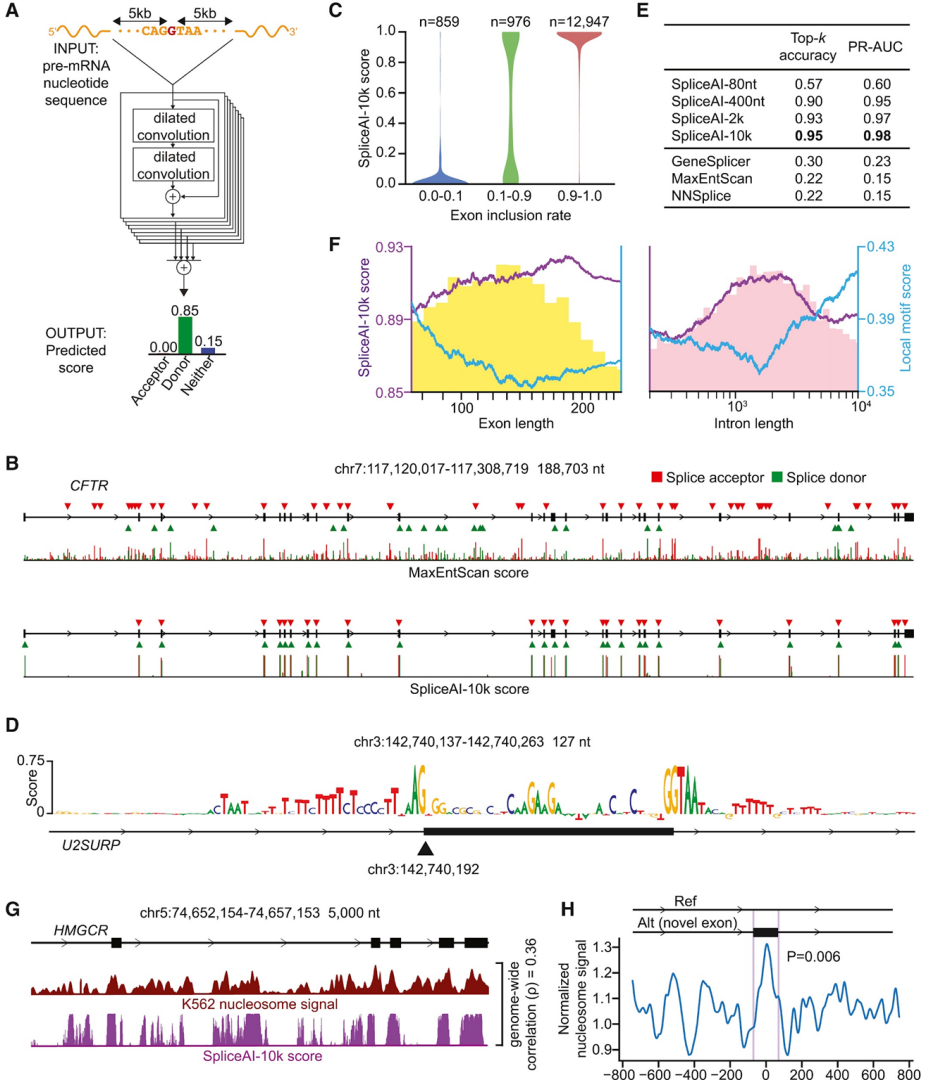

RB(N, W, D)
Batch Normalization
ReLU
Conv(N, W, D)
Batch Normalization
ReLU
Conv(N, W, D)
+


SpliceAI-80nt
SpliceAI-400nt
SpliceAI-2k
SpliceAI-10k
Input
Conv(32, 1, 1)
RB(32, 11, 1)
RB(32, 11, 4)
RB(32, 21, 10)
RB(32, 41, 25)
Conv(3, 1, 1)
Softmax
Output

Dilated Convolutions

- **Definition**: Dilated convolutions apply filters with gaps, expanding the receptive field without increasing parameters or computation.
- **Purpose**: Captures context over larger areas, useful for biological data with long-range dependencies.
- **Mechanism**: Uses a dilation rate to define gaps between filter elements; a rate of 1 is a standard convolution.
- **Advantages**:
  - **Captures Context**: Useful in biological imaging to detect meaningful patterns.
  - **Efficient**: Expands receptive field without extra computational cost.
- **Applications**:
  - **Genomics**: Models motifs over long DNA regions.
  - **Imaging**: Segments microscopy images with full spatial context.

SpliceAI


A
5kb
5kb
CAGGTAA
INPUT:
pre-mRNA
nucleotide
sequence
dilated
convolution
dilated
convolution
OUTPUT:
Predicted
score
0.85
0.00
0.15
Acceptor
Donor
Neither
C
n=859
n=976
n=12,947
SpliceAI-10k score
Exon inclusion rate
0.0-0.1
0.1-0.9
0.9-1.0
E
Top-k accuracy
PR-AUC
SpliceAI-80nt 0.57 0.60
SpliceAI-400nt 0.90 0.95
SpliceAI-2k 0.93 0.97
SpliceAI-10k 0.95 0.98
GeneSplicer 0.30 0.23
MaxEntScan 0.22 0.15
NNSplice 0.22 0.15
F
SpliceAI-10k score
Exon length
Intron length
Local motif score
B
chr7:117,120,017-117,308,719 188,703 nt
CFTR
Splice acceptor
Splice donor
MaxEntScan score
SpliceAI-10k score
D
chr3:142,740,137-142,740,263 127 nt
Score
U2SURP
chr3:142,740,192
G
chr5:74,652,154-74,657,153 5,000 nt
HMGCR
K562 nucleosome signal
SpliceAI-10k score
genome-wide correlation (ρ) = 0.36
H
Ref
Alt (novel exon)
P=0.006
Normalized nucleosome signal

DNA Foundation Models

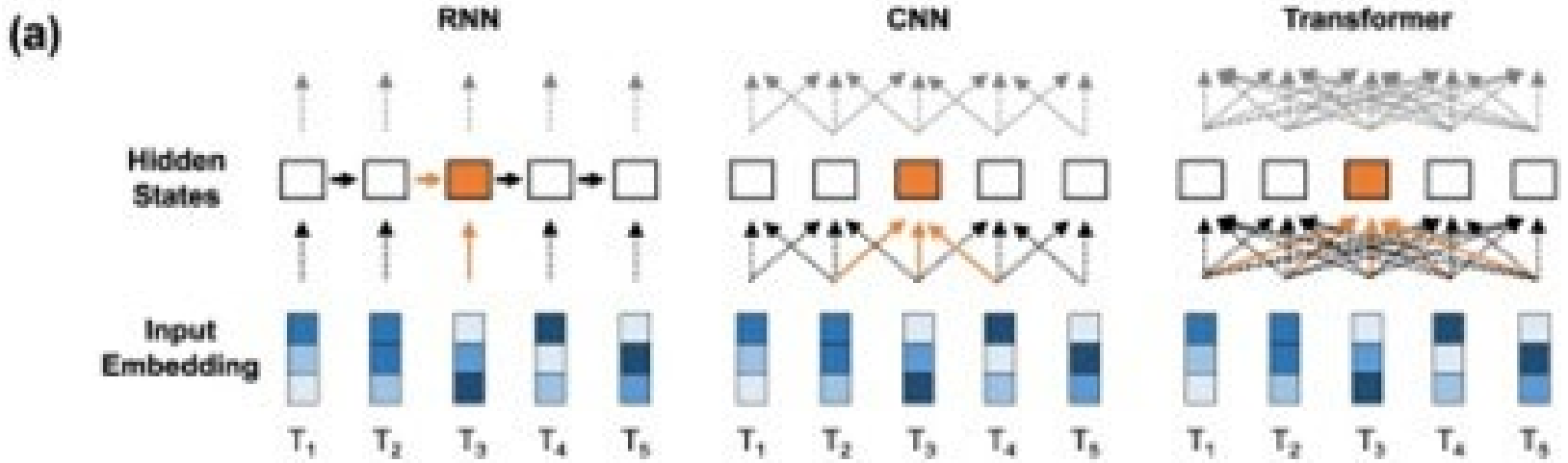
- **What are DNA Foundation Models?**
  - Large-scale neural networks trained on genomic data (similar in concept to language models like GPT).
  - Designed to capture patterns and relationships in DNA sequences, enabling various downstream tasks.
- **Training DNA Foundation Models**
  - Trained on massive amounts of genomic data, including public DNA sequences.
  - Self-supervised learning techniques are often used, such as predicting masked nucleotides, akin to how language models predict masked words.
  - Requires significant computational resources, but pretraining allows use in many tasks with minimal additional data.
- **Applications**
  - **Functional Annotation**: Predicting the function of genes, regulatory elements, or mutations.
  - **Variant Impact**: Identifying pathogenic variants, predicting their impact on gene expression or protein function.
  - **Evolutionary Insights**: Analyzing conservation across species or understanding mutational biases.
- **Advantages over Traditional Methods**
  - **Transferability**: Once trained, the models can be fine-tuned for specific applications, saving time and computational power.
  - **Generalization**: Able to learn general sequence motifs, making them highly adaptable.
  - **Handling Complexity**: Better at recognizing subtle sequence motifs and dependencies, which might be missed by simpler models.
- **Challenges and Limitations**
  - **Data Bias**: Limited by biases present in training data, such as underrepresentation of certain species or populations.
  - **Interpretability**: Hard to interpret model decisions, making biological validation necessary.

Bioinformatics. 2021 Feb 4;37(15):2112–2120. doi: 10.1093/bioinformatics/btab083

DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome

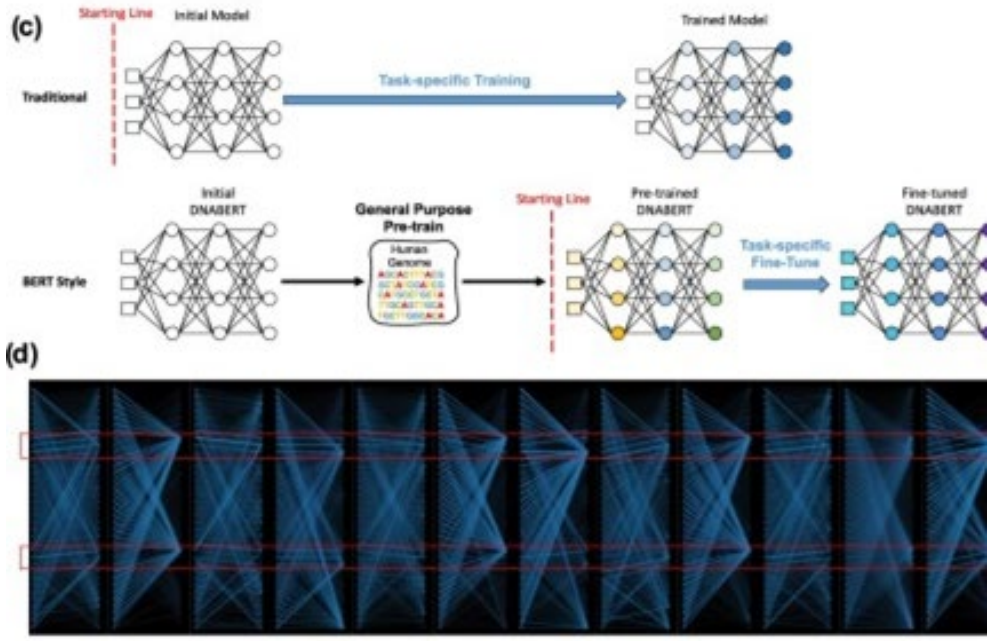
Yanrong Ji [1,a], Zhihan Zhou [2,a], Han Liu [3,✉], Ramana V Davuluri [4,✉]

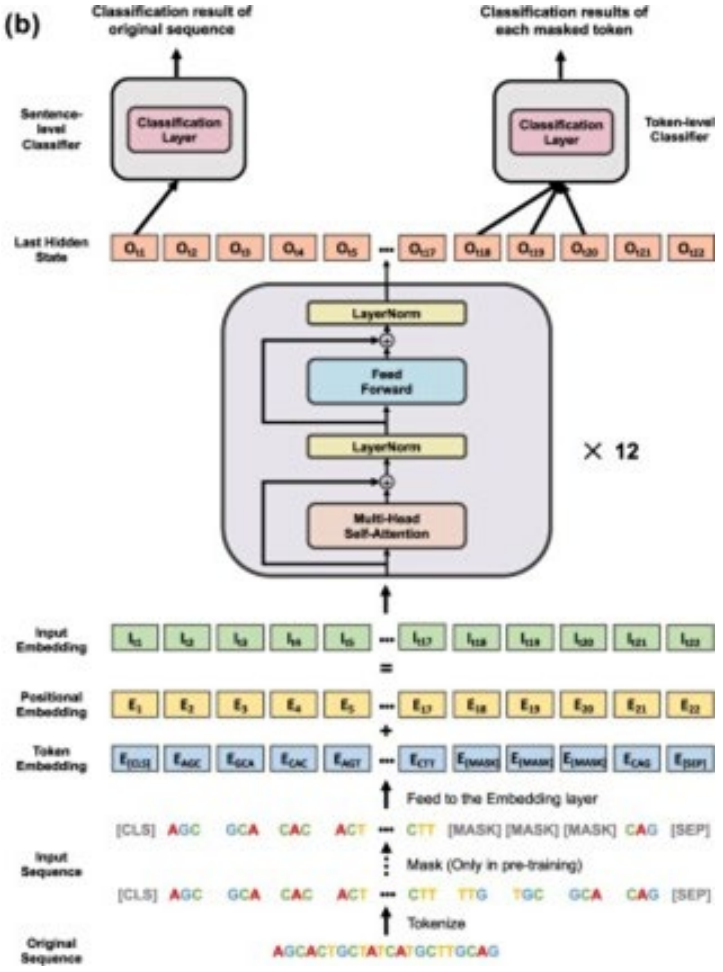
DNABERT



Yanrong et al., 2021 https://pubmed.ncbi.nlm.nih.gov/33538820/

DNABERT

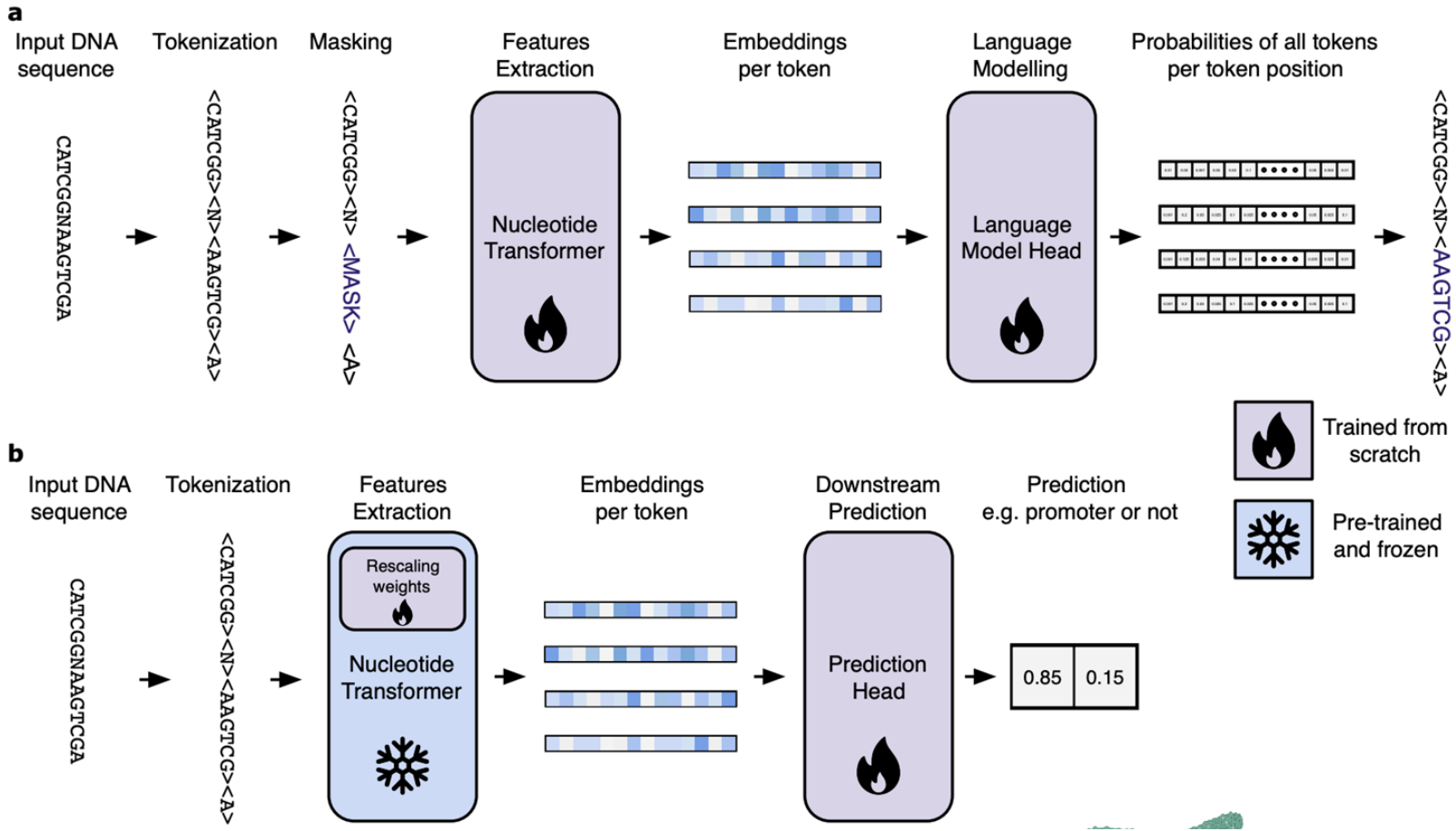


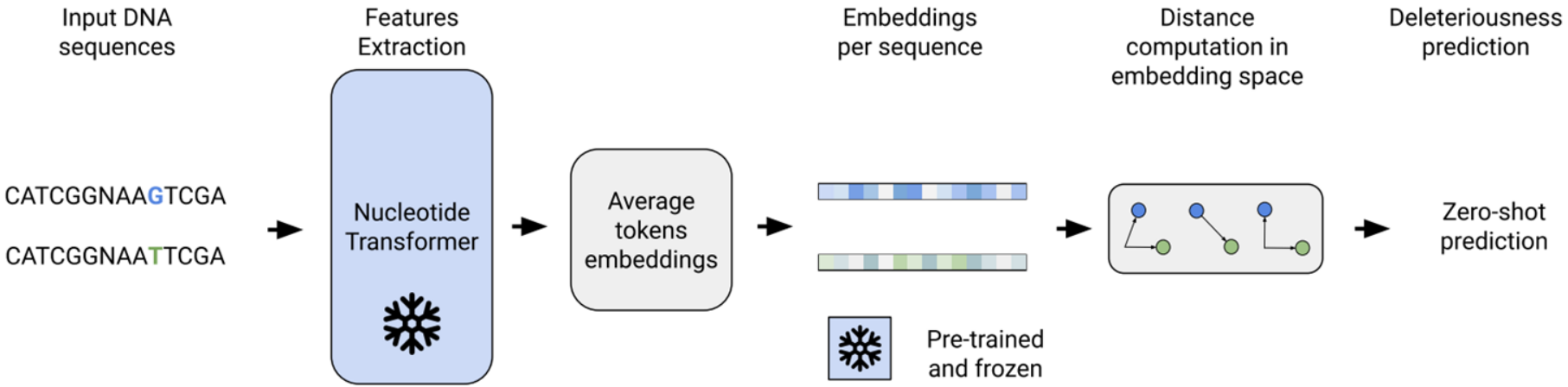


Yanrong et al., 2021 https://pubmed.ncbi.nlm.nih.gov/33538820/

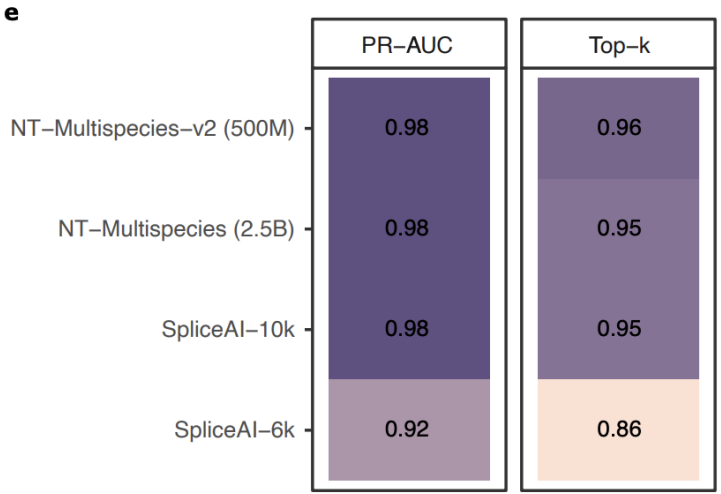
The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics

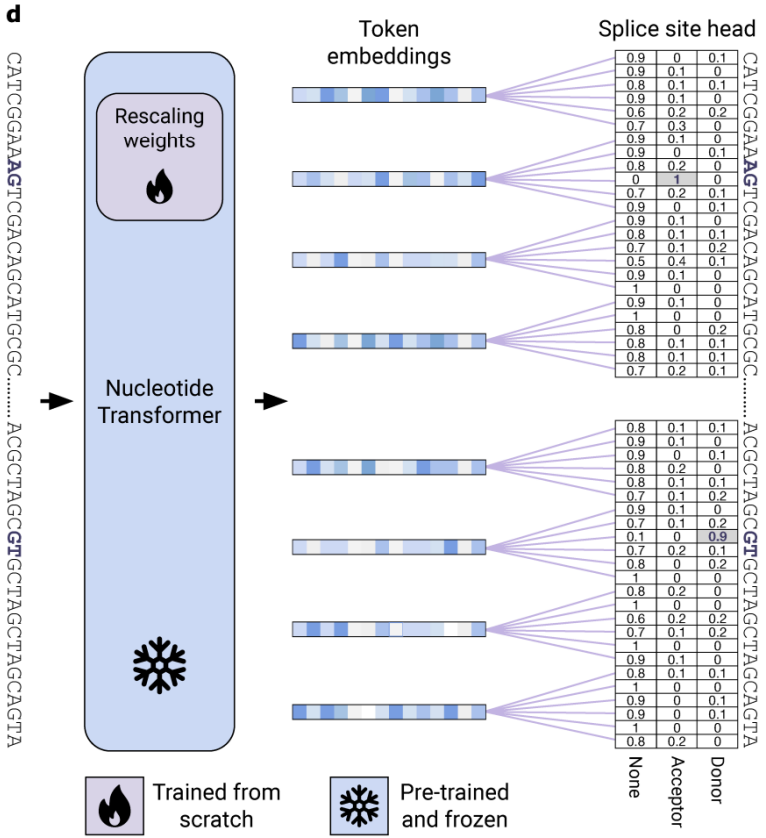
Hugo Dalla-Torre[1] **Liam Gonzalez**[1] **Javier Mendoza-Revilla**[1] **Nicolas Lopez Carranza**[1]
Adam Henryk Grzywaczewski[2] **Francesco Oteri**[1] **Christian Dallago**[2,3] **Evan Trop**[1] **Bernardo P. de Almeida**[1]
Hassan Sirelkhatim[2] **Guillaume Richard**[1] **Marcin Skwark**[1] **Karim Beguir**[1]
Marie Lopez[*,1] **and Thomas Pierrot**[*,1]
[*]Equal supervision, [1]InstaDeep, [2]Nvidia, [3]TUM


a
Input DNA sequence
Tokenization
Masking
Features Extraction
Embeddings per token
Language Modelling
Probabilities of all tokens per token position
CATCGGNAAGTCGA
<CATCGG><N><AAGTCG><A>
<CATCGG><N> <MASK> <A>
Nucleotide Transformer
Language Model Head
<CATCGG><N><AAGTCG><A>
Trained from scratch
Pre-trained and frozen
b
Input DNA sequence
Tokenization
Features Extraction
Embeddings per token
Downstream Prediction
Prediction e.g. promoter or not
CATCGGNAAGTCGA
<CATCGG><N><AAGTCG><A>
Rescaling weights
Nucleotide Transformer
Prediction Head
0.85
0.15

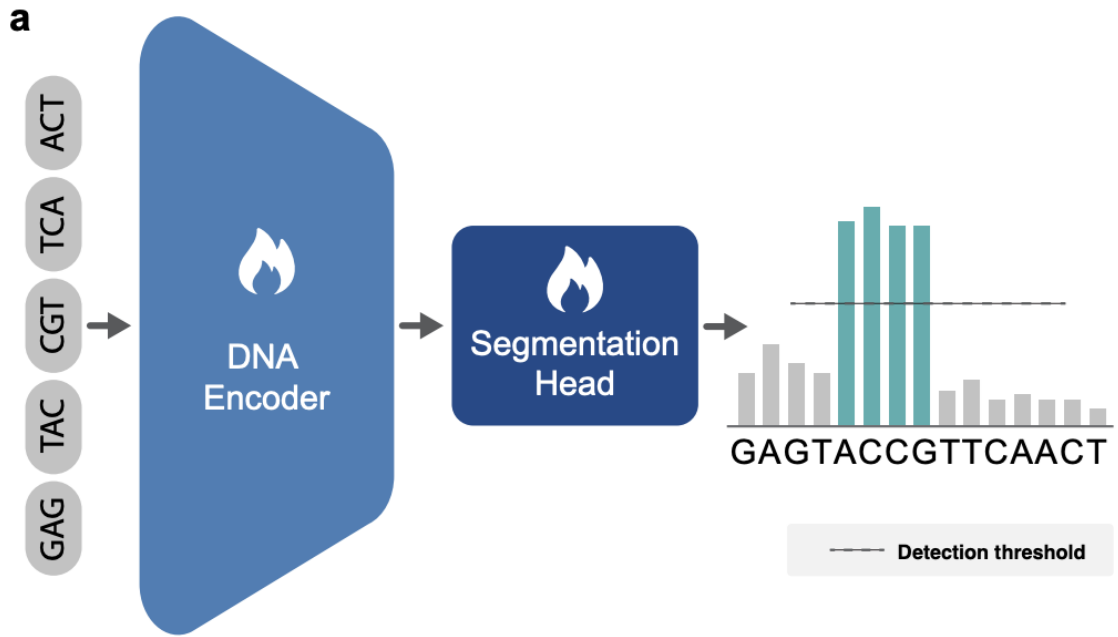

Input DNA sequences
Features Extraction
Embeddings per sequence
Distance computation in embedding space
Deleteriousness prediction
CATCGGNAAGTCGA
CATCGGNAATTCGA
Nucleotide Transformer
Average tokens embeddings
Pre-trained and frozen
Zero-shot prediction

Predicting Splice Sites



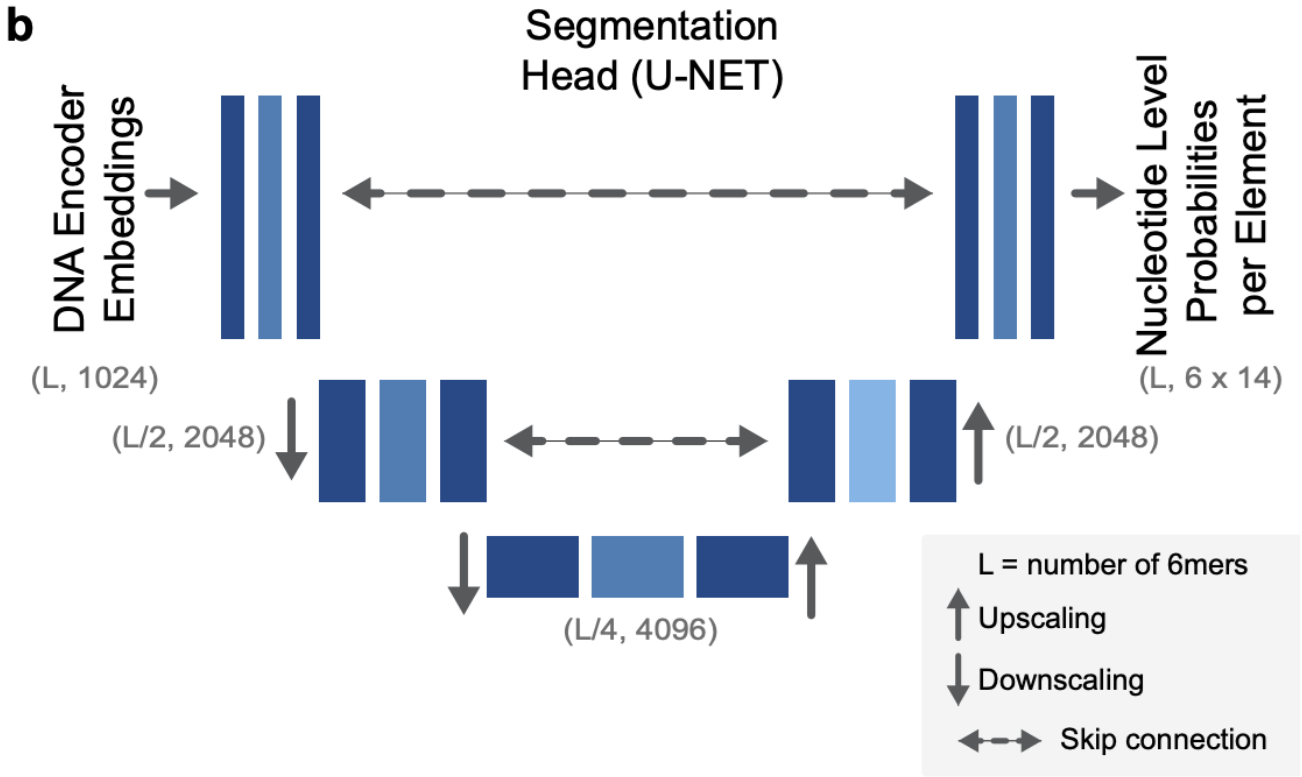


SegmentNT

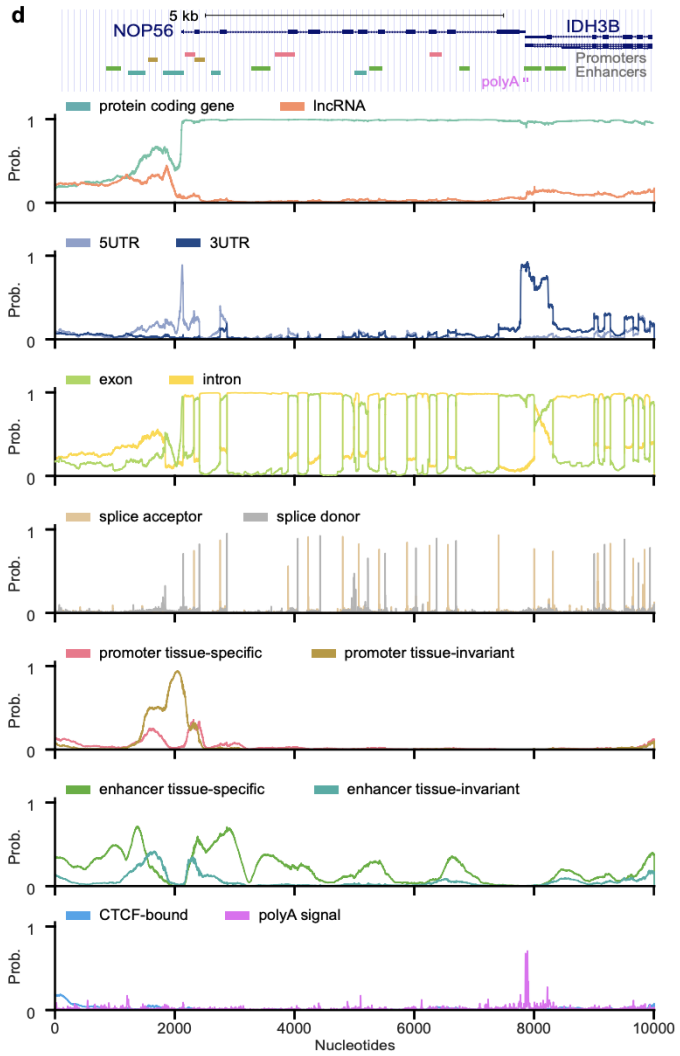


Almeida et al., 2024 https://www.biorxiv.org/content/10.1101/2024.03.14.584712v2.full.pdf

SegmentNT uses a U-NET for sequence segmentation



SegmentNT predictions



HyenaDNA

HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution

Eric Nguyen[*,1], **Michael Poli**[*,1], **Marjan Faizi**[2,*],
Armin W. Thomas[1], Callum Birch Sykes[3], Michael Wornow[1], Aman Patel[1],
Clayton Rabideau[3], Stefano Massaroli[4], Yoshua Bengio[4], Stefano Ermon[1],
Stephen A. Baccus[1,†], Christopher Ré[1,†]

Hyena operator is O(L log L)

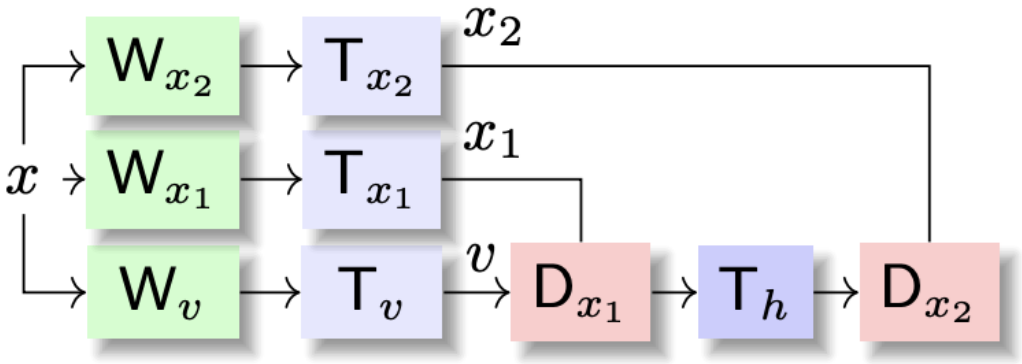
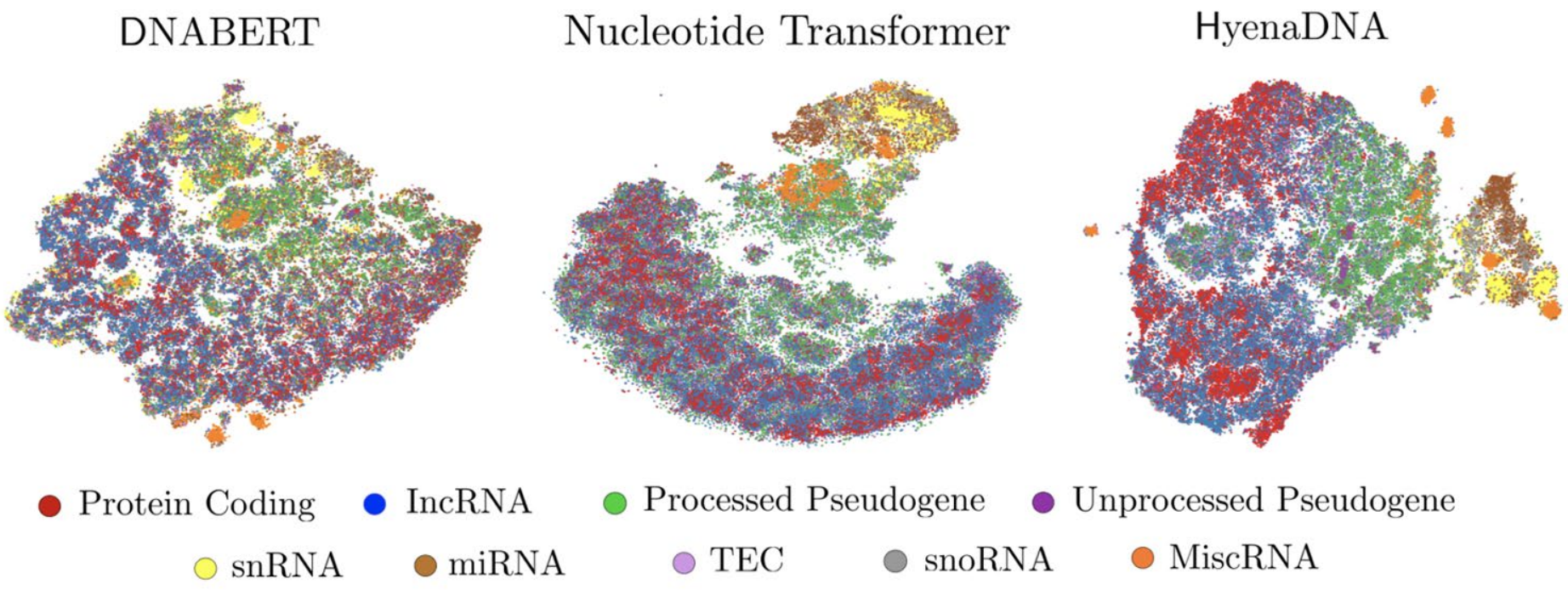
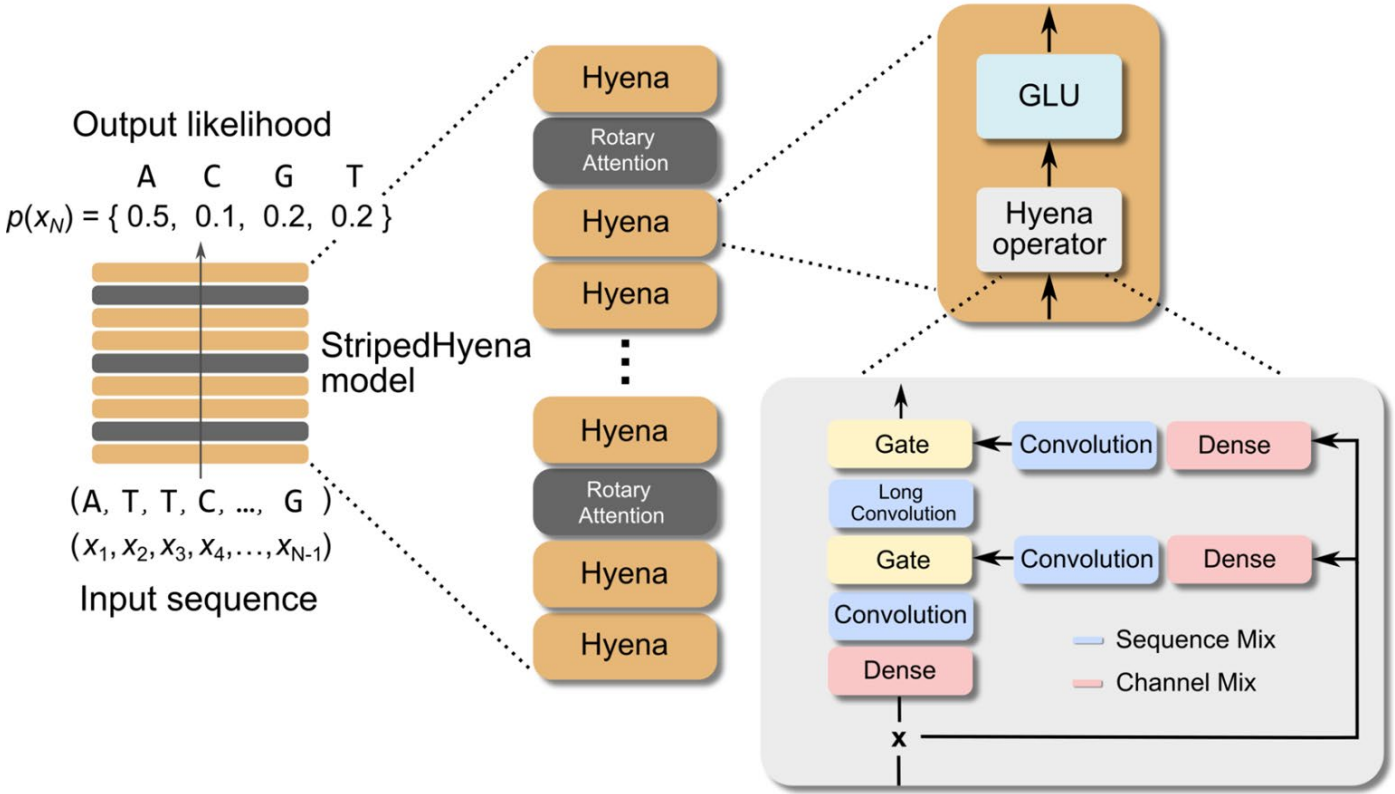


Figure 3.1: The Hyena operator is a combination of long convolutions T and data-controlled gating D, and can be a drop-in replacement for attention.

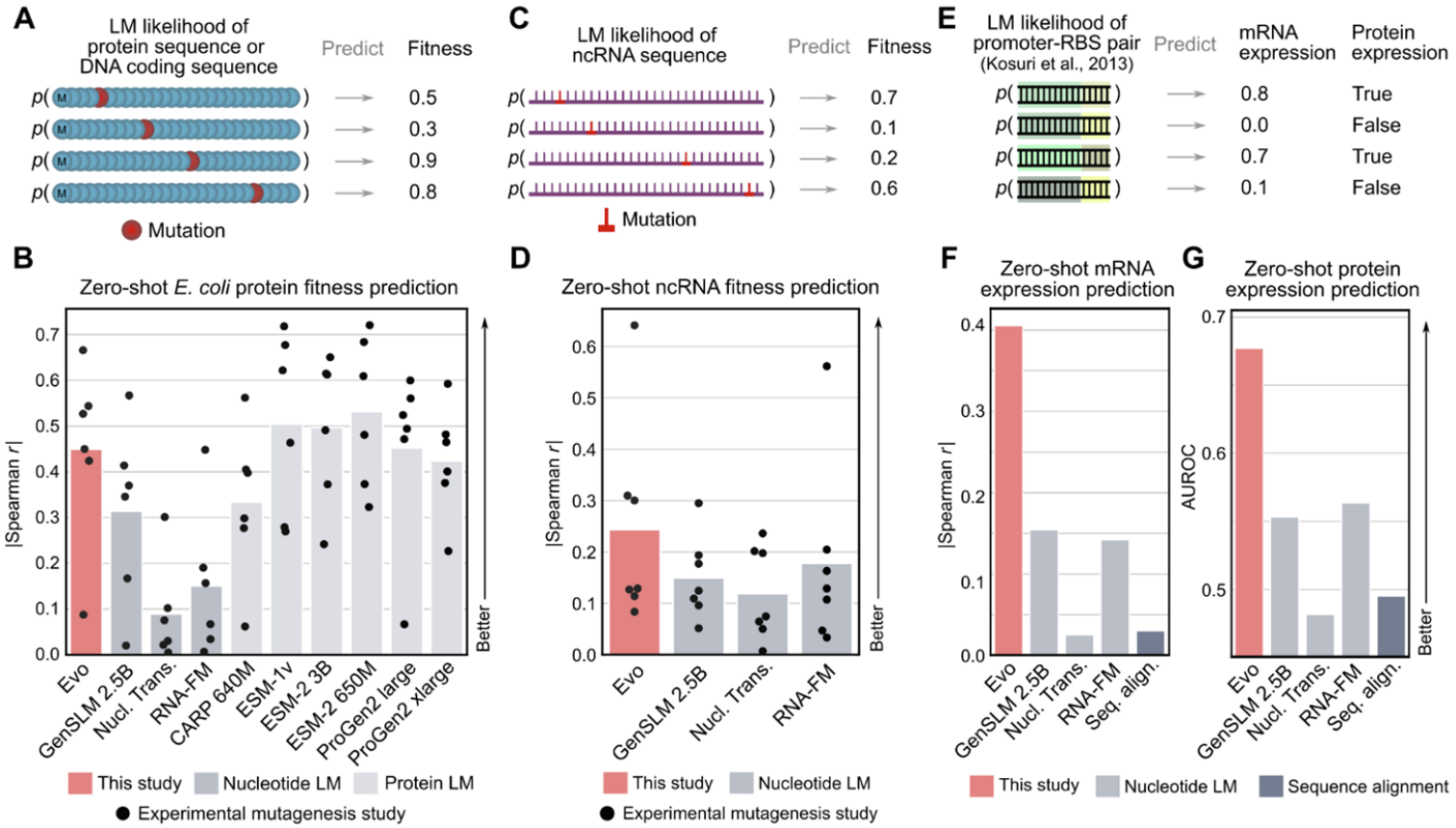

DNABERT
Nucleotide Transformer
HyenaDNA
Protein Coding
lncRNA
Processed Pseudogene
Unprocessed Pseudogene
snRNA
miRNA
TEC
snoRNA
MiscRNA

Evo DNA Foundation Model



https://www.together.ai/blog/evo

Evo DNA Foundation Model



Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation

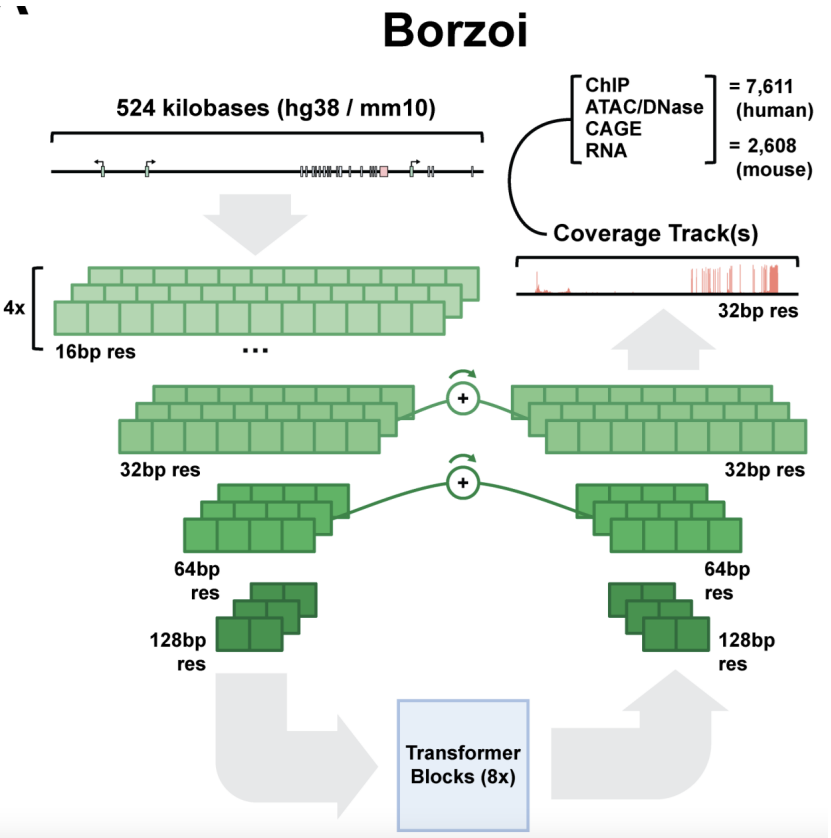
Johannes Linder
Calico Life Sciences LLC
jlinder@calicolabs.com

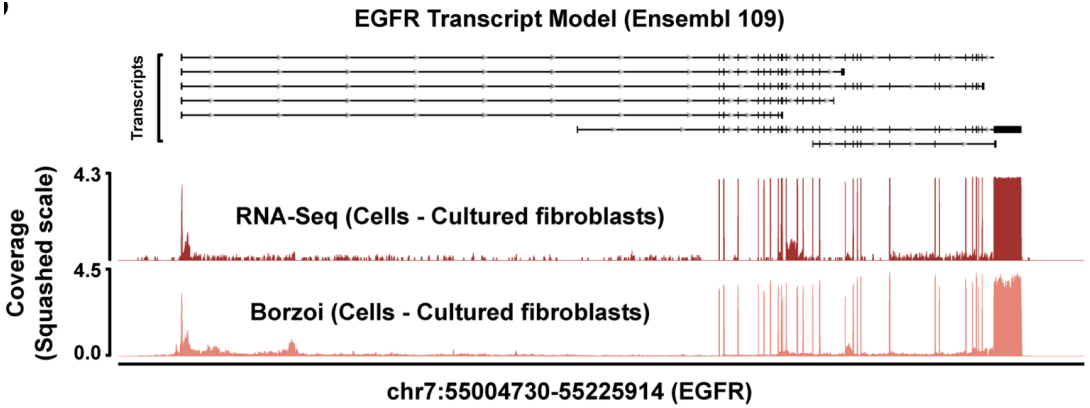
Divyanshi Srivastava
Calico Life Sciences LLC
divyanshi@calicolabs.com

Han Yuan
Calico Life Sciences LLC
yuanh@calicolabs.com

Vikram Agarwal
mRNA Center of Excellence, Sanofi Pasteur Inc.
Vikram.Agarwal@sanofi.com

David R. Kelley
Calico Life Sciences LLC
drk@calicolabs.com


Borzoi
524 kilobases (hg38 / mm10)
ChIP
ATAC/DNase
CAGE
RNA
= 7,611 (human)
= 2,608 (mouse)
Coverage Track(s)
32bp res
4x
16bp res
32bp res
32bp res
64bp res
64bp res
128bp res
128bp res
Transformer Blocks (8x)


EGFR Transcript Model (Ensembl 109)
Transcripts
Coverage (Squashed scale)
4.3
RNA-Seq (Cells - Cultured fibroblasts)
4.5
Borzoi (Cells - Cultured fibroblasts)
0.0
chr7:55004730-55225914 (EGFR)