UiO : Universitetet i Oslo

Machine Learning in Computational Biology: Overview

IN-BIOS5000/IN-BIOS9000

Milena Pavlović
Biomedical Informatics Research Group
Department of Informatics

milenpa@uio.no

Disclaimer

I am a machine learning researcher, not a biologist:
you are the experts there!

Learning aims

- ❏ Key points should be the intuition and high-level understanding of what machine learning is, types of problems it can help solving
- ❏ Machine learning is not a black box: every choice we make has a meaning
- ❏ Overall understanding that there is a data representation component and a machine learning algorithm
- ❏ High-level understanding of machine learning workflow, comparison and uncertainty related to it

Sequencing technologies provide data which can be examined for biological properties

CDR3	V gene	J gene	Species
CAAAERNTGELFF	TRBV28*01	TRBJ2-2*01	HomoSapiens
CAAGVENTGELFF	TRBV5-6*01	TRBJ2-2*01	HomoSapiens
CAAQATNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQDSNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAAQMTNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQNLNTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAARDQRDLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens
CAASDPNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAASEMNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CACQELNTGELFF	TRBV30*01	TRBJ2-2*01	HomoSapiens
CAEGELNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGADSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGDYLNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGDPNTGELFF	TRBV7-9*01	TRBJ2-2*01	HomoSapiens
CAGGDSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGRGNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAGGVNPNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAGQDLNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGQNLNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAGQRANTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAIADANTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAIGDENTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAIGDRNTGELFF	TRBV5-5*01	TRBJ2-2*01	HomoSapiens
CAIGDRSSGEQYF	TRBV5-4*01	TRBJ2-7*01	HomoSapiens
CAIQDLNTGELFF	TRBV13*01	TRBJ2-2*01	HomoSapiens
CAIQESNTGELFF	TRBV10-3*01	TRBJ2-2*01	HomoSapiens
CAIQYANTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAIRTSGMLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens

Motifs and data from VDJdb (Bagaev et al. 2020)

Sequencing technologies provide data which can be examined for biological properties

CDR3	V gene	J gene	Species
CAAAERNTGELFF	TRBV28*01	TRBJ2-2*01	HomoSapiens
CAAGVENTGELFF	TRBV5-6*01	TRBJ2-2*01	HomoSapiens
CAAQATNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQDSNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAAQMTNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQNLNTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAARDQRDLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens
CAASDPNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAASEMNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CACQELNTGELFF	TRBV30*01	TRBJ2-2*01	HomoSapiens
CAEGELNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGADSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGDYLNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGDPNTGELFF	TRBV7-9*01	TRBJ2-2*01	HomoSapiens
CAGGDSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGRGNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAGGVNPNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAGQDLNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGQNLNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAGQRANTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAIADANTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAIGDENTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAIGDRNTGELFF	TRBV5-5*01	TRBJ2-2*01	HomoSapiens
CAIGDRSSGEQYF	TRBV5-4*01	TRBJ2-7*01	HomoSapiens
CAIQDLNTGELFF	TRBV13*01	TRBJ2-2*01	HomoSapiens
CAIQESNTGELFF	TRBV10-3*01	TRBJ2-2*01	HomoSapiens
CAIQYANTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAIRTSGMLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens

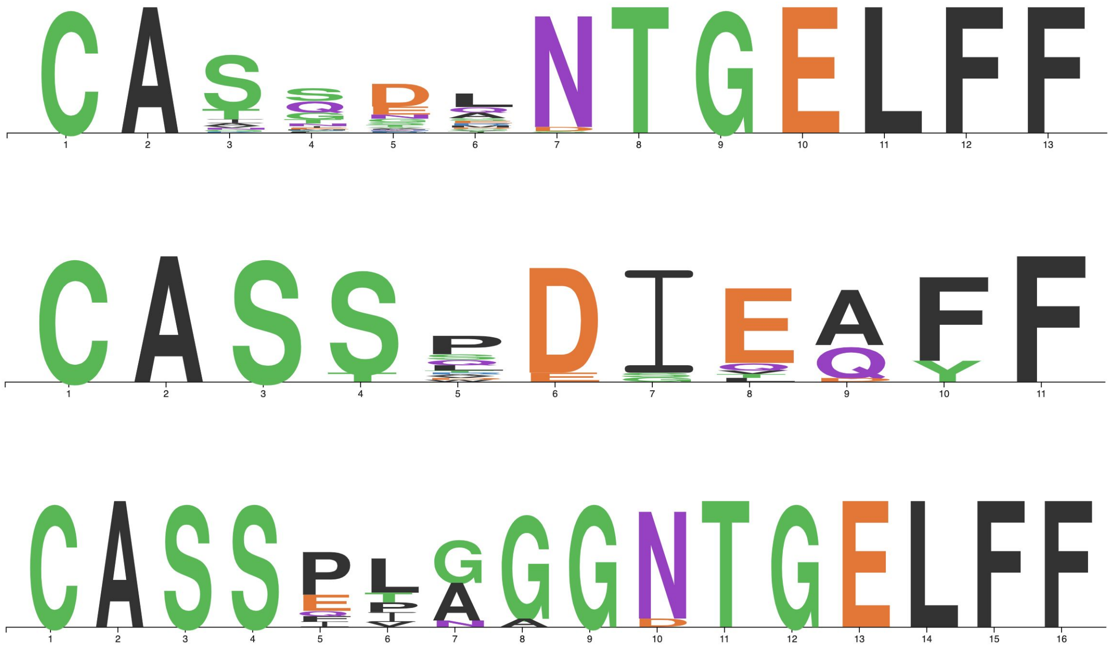
discover motifs in the data $\longrightarrow$

Motifs and data from VDJdb (Bagaev et al. 2020)

Sequencing technologies provide data which can be examined for biological properties

CDR3	V gene	J gene	Species
CAAAERNTGELFF	TRBV28*01	TRBJ2-2*01	HomoSapiens
CAAGVENTGELFF	TRBV5-6*01	TRBJ2-2*01	HomoSapiens
CAAQATNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQDSNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAAQMTNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQNLNTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAARDQRDLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens
CAASDPNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAASEMNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CACQELNTGELFF	TRBV30*01	TRBJ2-2*01	HomoSapiens
CAEGELNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGADSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGDYLNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGDPNTGELFF	TRBV7-9*01	TRBJ2-2*01	HomoSapiens
CAGGDSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGRGNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAGGVNPNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAGQDLNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGQNLNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAGQRANTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAIADANTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAIGDENTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAIGDRNTGELFF	TRBV5-5*01	TRBJ2-2*01	HomoSapiens
CAIGDRSSGEQYF	TRBV5-4*01	TRBJ2-7*01	HomoSapiens
CAIQDLNTGELFF	TRBV13*01	TRBJ2-2*01	HomoSapiens
CAIQESNTGELFF	TRBV10-3*01	TRBJ2-2*01	HomoSapiens
CAIQYANTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAIRTSGMLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens

discover motifs in the data $\longrightarrow$



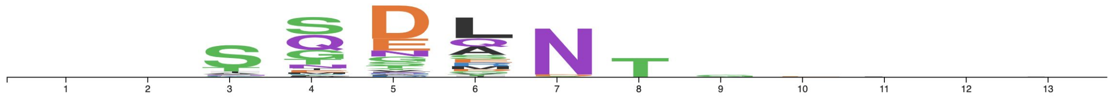
Motifs and data from VDJdb (Bagaev et al. 2020)

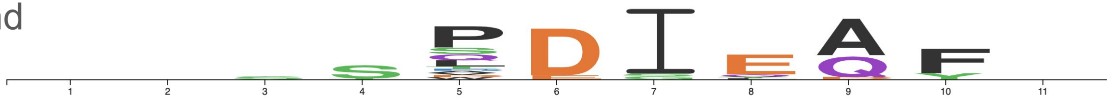
Sequencing technologies provide data which can be examined for biological properties

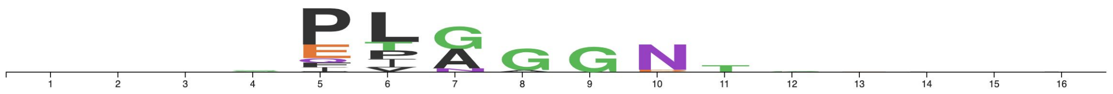
CDR3	V gene	J gene	Species
CAAAERNTGELFF	TRBV28*01	TRBJ2-2*01	HomoSapiens
CAAGVENTGELFF	TRBV5-6*01	TRBJ2-2*01	HomoSapiens
CAAQATNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQDSNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAAQMTNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAAQNLNTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAARDQRDLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens
CAASDPNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAASEMNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CACQELNTGELFF	TRBV30*01	TRBJ2-2*01	HomoSapiens
CAEGELNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGADSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGDYLNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGDPNTGELFF	TRBV7-9*01	TRBJ2-2*01	HomoSapiens
CAGGDSNTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAGGRGNTGELFF	TRBV12-3*01	TRBJ2-2*01	HomoSapiens
CAGGVNPNTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAGQDLNTGELFF	TRBV7-2*01	TRBJ2-2*01	HomoSapiens
CAGQNLNTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAGQRANTGELFF	TRBV19*01	TRBJ2-2*01	HomoSapiens
CAIADANTGELFF	TRBV5-1*01	TRBJ2-2*01	HomoSapiens
CAIGDENTGELFF	TRBV7-8*01	TRBJ2-2*01	HomoSapiens
CAIGDRNTGELFF	TRBV5-5*01	TRBJ2-2*01	HomoSapiens
CAIGDRSSGEQYF	TRBV5-4*01	TRBJ2-7*01	HomoSapiens
CAIQDLNTGELFF	TRBV13*01	TRBJ2-2*01	HomoSapiens
CAIQESNTGELFF	TRBV10-3*01	TRBJ2-2*01	HomoSapiens
CAIQYANTGELFF	TRBV15*01	TRBJ2-2*01	HomoSapiens
CAIRTSGMLNTGELFF	TRBV2*01	TRBJ2-2*01	HomoSapiens

discover motifs and remove genetic background

$\longrightarrow$



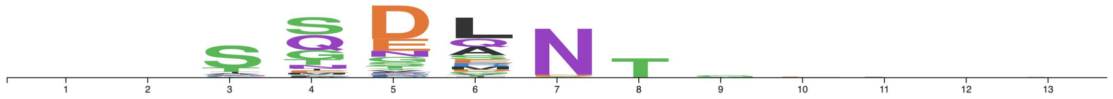
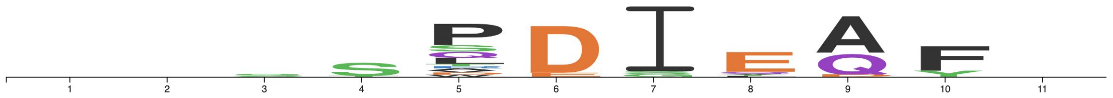
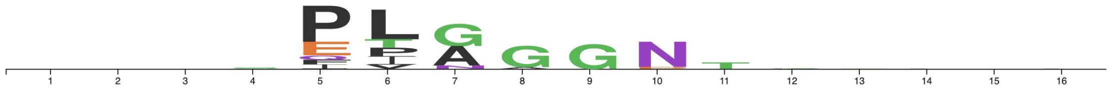




Motifs and data from VDJdb (Bagaev et al. 2020)

Sequencing technologies provide data which can be examined for biological properties

- One way to approach an analysis: make a position weight matrix showing product multinomial distribution of amino acids
- But what if we want to predict if a sequence is specific to a virus or not?

Motifs and data from VDJdb (Bagaev et al. 2020)

Machine learning is a powerful approach to discovering patterns in (biological) data

- A set of methods that allow for making inferences about the data
- Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
.....	

raw labeled data

Machine learning is a powerful approach to discovering patterns in (biological) data

- A set of methods that allow for making inferences about the data
- Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
.....	

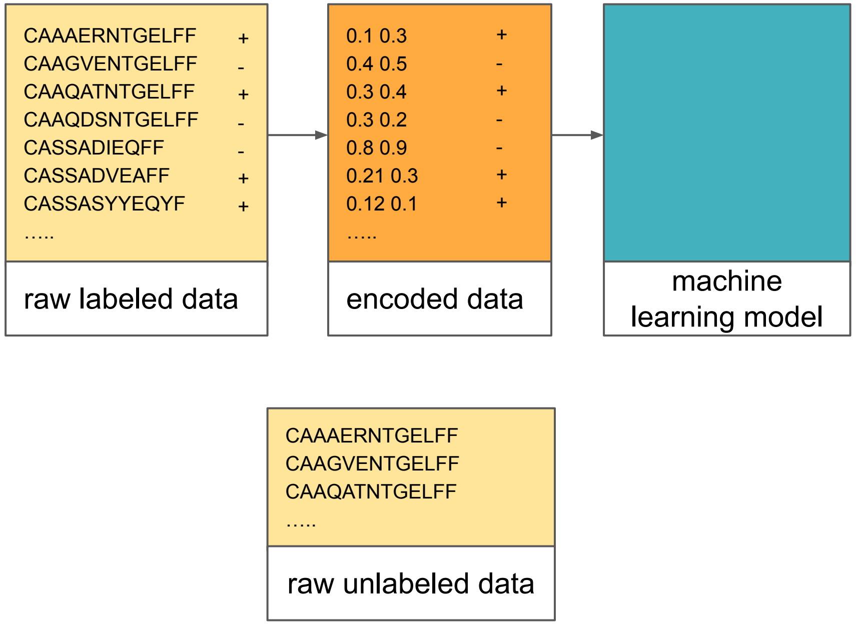
raw labeled data

CAAAERNTGELFF
CAAGVENTGELFF
CAAQATNTGELFF
.....

raw unlabeled data

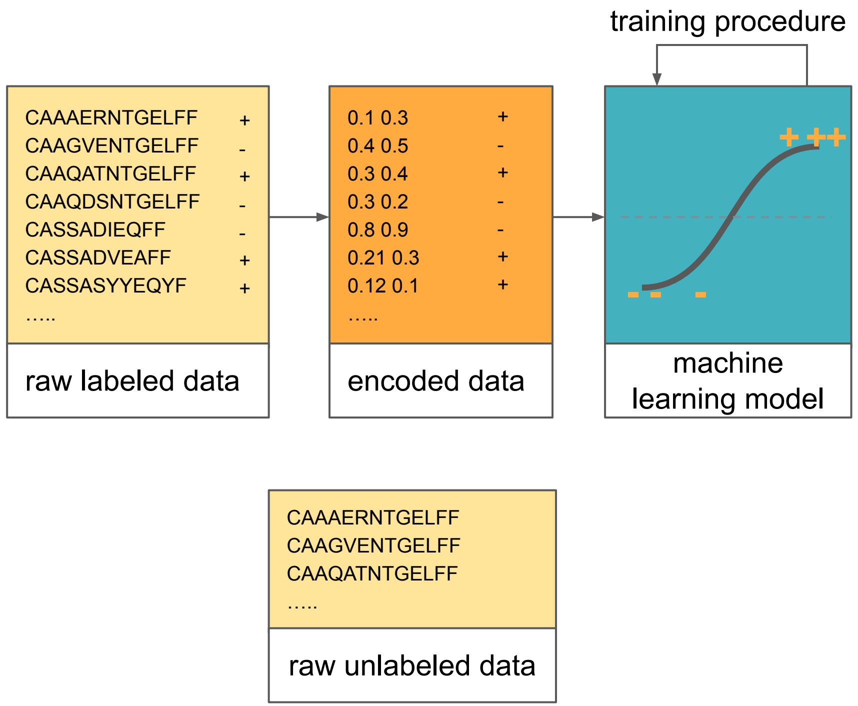
Machine learning is a powerful approach to discovering patterns in (biological) data

- ❏ A set of methods that allow for making inferences about the data
- ❏ Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors



Machine learning is a powerful approach to discovering patterns in (biological) data

- A set of methods that allow for making inferences about the data
- Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors



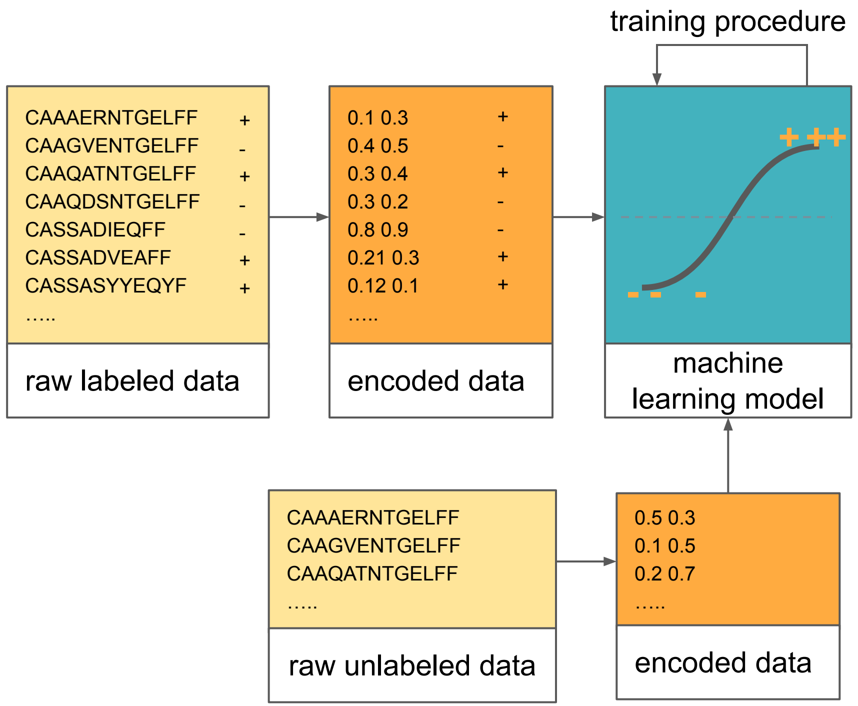
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
.....	

0.1 0.3	+
0.4 0.5	-
0.3 0.4	+
0.3 0.2	-
0.8 0.9	-
0.21 0.3	+
0.12 0.1	+
.....	

CAAAERNTGELFF
CAAGVENTGELFF
CAAQATNTGELFF
.....

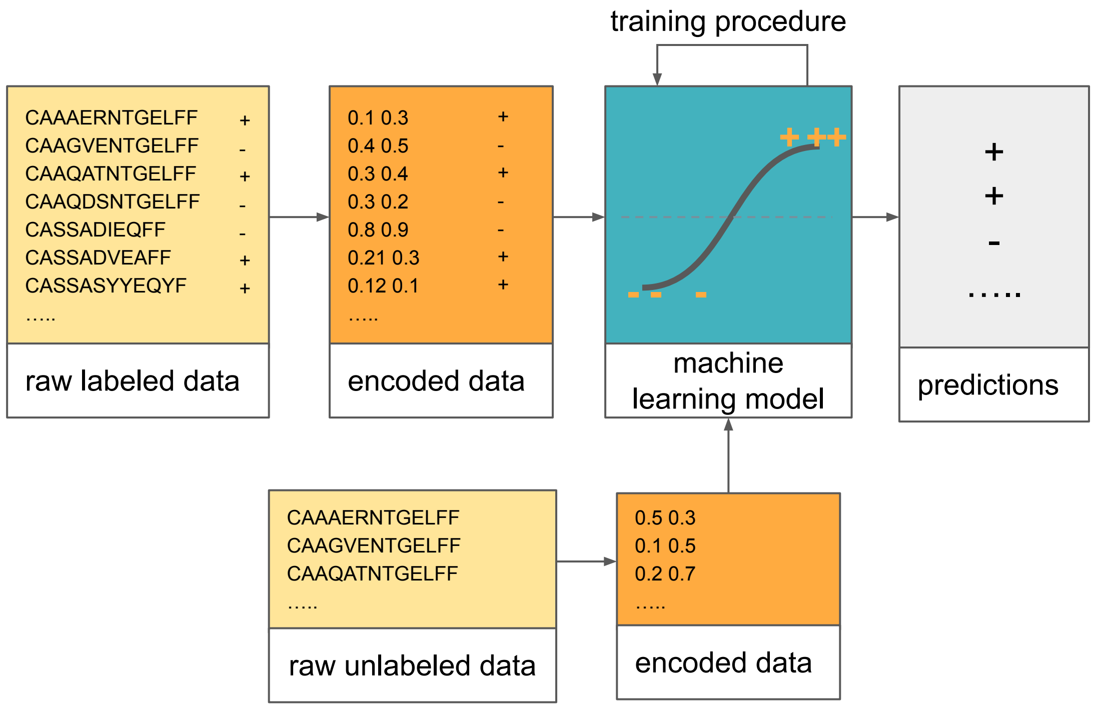
Machine learning is a powerful approach to discovering patterns in (biological) data

- A set of methods that allow for making inferences about the data
- Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors

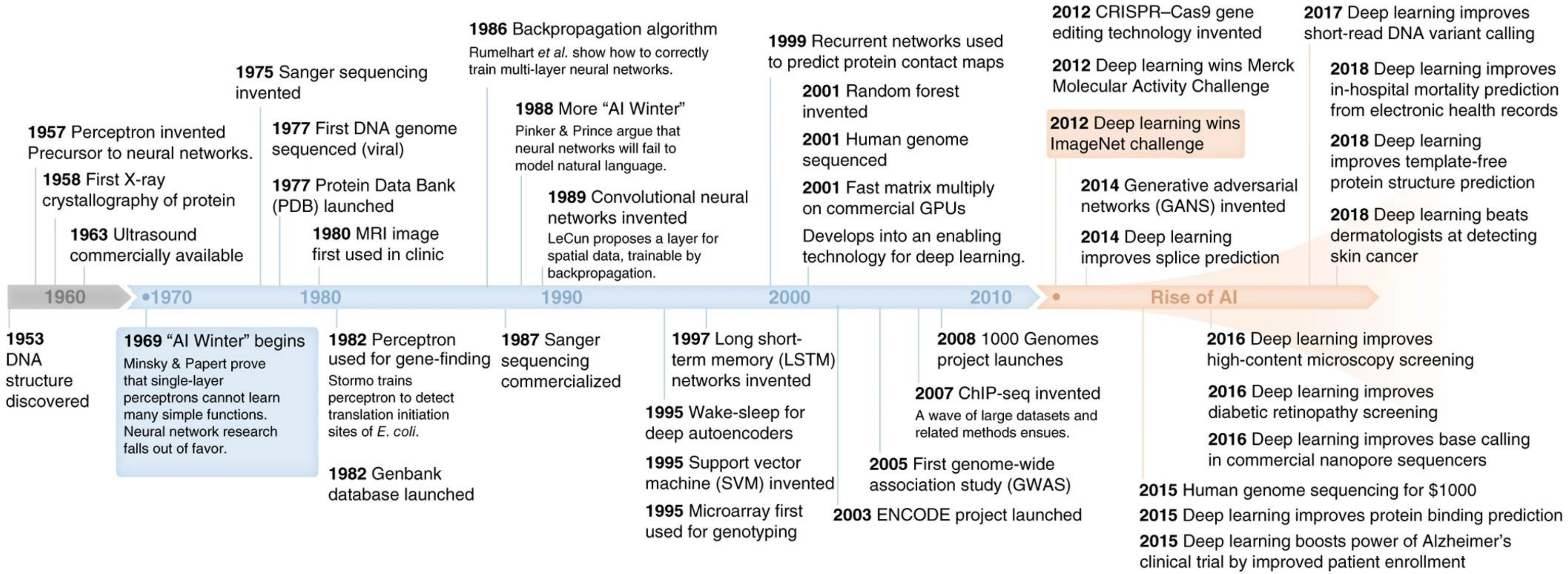


Machine learning is a powerful approach to discovering patterns in (biological) data

- A set of methods that allow for making inferences about the data
- Example: will the receptor bind to the virus or not? - we can fit a logistic regression model on receptor data and then predict binding for new receptors



ML and computational biology development timeline



1953 DNA structure discovered

1957 Perceptron invented
Precursor to neural networks.

1958 First X-ray crystallography of protein

1963 Ultrasound commercially available

1969 "AI Winter" begins
Minsky & Papert prove that single-layer perceptrons cannot learn many simple functions. Neural network research falls out of favor.

1975 Sanger sequencing invented

1977 First DNA genome sequenced (viral)

1977 Protein Data Bank (PDB) launched

1980 MRI image first used in clinic

1982 Perceptron used for gene-finding
Stormo trains perceptron to detect translation initiation sites of *E. coli*.

1982 Genbank database launched

1986 Backpropagation algorithm
Rumelhart *et al.* show how to correctly train multi-layer neural networks.

1987 Sanger sequencing commercialized

1988 More "AI Winter"
Pinker & Prince argue that neural networks will fail to model natural language.

1989 Convolutional neural networks invented
LeCun proposes a layer for spatial data, trainable by backpropagation.

1995 Wake-sleep for deep autoencoders

1995 Support vector machine (SVM) invented

1995 Microarray first used for genotyping

1997 Long short-term memory (LSTM) networks invented

1999 Recurrent networks used to predict protein contact maps

2001 Random forest invented

2001 Human genome sequenced

2001 Fast matrix multiply on commercial GPUs
Develops into an enabling technology for deep learning.

2003 ENCODE project launched

2005 First genome-wide association study (GWAS)

2007 ChIP-seq invented
A wave of large datasets and related methods ensues.

2008 1000 Genomes project launches

2012 CRISPR–Cas9 gene editing technology invented

2012 Deep learning wins Merck Molecular Activity Challenge

2012 Deep learning wins ImageNet challenge

2014 Generative adversarial networks (GANS) invented

2014 Deep learning improves splice prediction

2015 Human genome sequencing for $1000

2015 Deep learning improves protein binding prediction

2015 Deep learning boosts power of Alzheimer's clinical trial by improved patient enrollment

2016 Deep learning improves high-content microscopy screening

2016 Deep learning improves diabetic retinopathy screening

2016 Deep learning improves base calling in commercial nanopore sequencers

2017 Deep learning improves short-read DNA variant calling

2018 Deep learning improves in-hospital mortality prediction from electronic health records

2018 Deep learning improves template-free protein structure prediction

2018 Deep learning beats dermatologists at detecting skin cancer

Wainberg et al. 2018

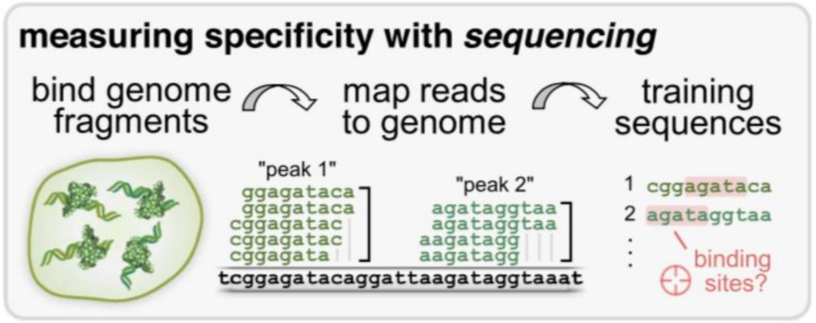
A variety of research questions in computational biology can be tackled with machine learning

Transcription factor binding prediction:

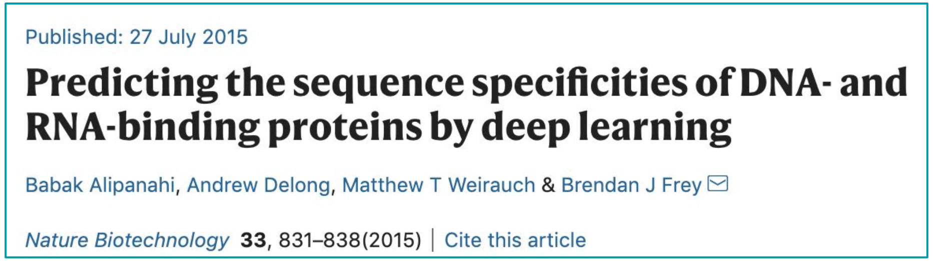
Transcription factors are proteins which bind to certain sites in DNA and regulate transcription of genes

Given a set of DNA sequences for which we know if they will bind or not, how can we predict if a transcription factor will bind to a new DNA sequence?

Classification problem!



Leung et al. 2016



Published: 27 July 2015

Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning

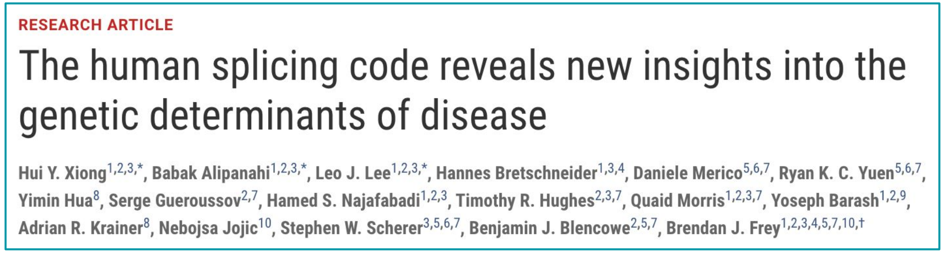
Babak Alipanahi, Andrew Delong, Matthew T Weirauch & Brendan J Frey

Nature Biotechnology **33**, 831–838(2015) | Cite this article

A variety of research questions in computational biology can be tackled with machine learning

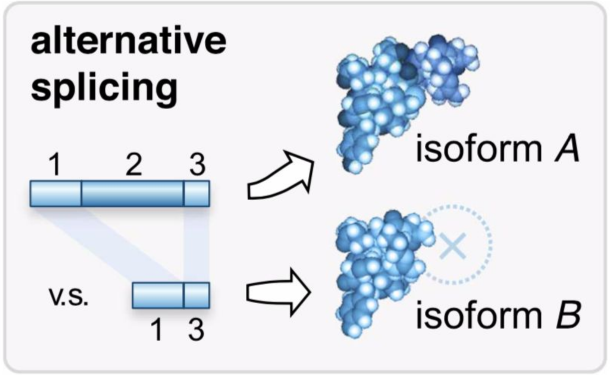
Splicing: processing of precursor RNA that creates messenger RNA by removing non-coding regions (introns) and connects gene-coding regions (exons) together

Predicting how often exons will be included in the transcripts is a regression problem


RESEARCH ARTICLE

The human splicing code reveals new insights into the genetic determinants of disease

Hui Y. Xiong[1,2,3,*], Babak Alipanahi[1,2,3,*], Leo J. Lee[1,2,3,*], Hannes Bretschneider[1,3,4], Daniele Merico[5,6,7], Ryan K. C. Yuen[5,6,7], Yimin Hua[8], Serge Gueroussov[2,7], Hamed S. Najafabadi[1,2,3], Timothy R. Hughes[2,3,7], Quaid Morris[1,2,3,7], Yoseph Barash[1,2,9], Adrian R. Krainer[8], Nebojsa Jojic[10], Stephen W. Scherer[3,5,6,7], Benjamin J. Blencowe[2,5,7], Brendan J. Frey[1,2,3,4,5,7,10,†]



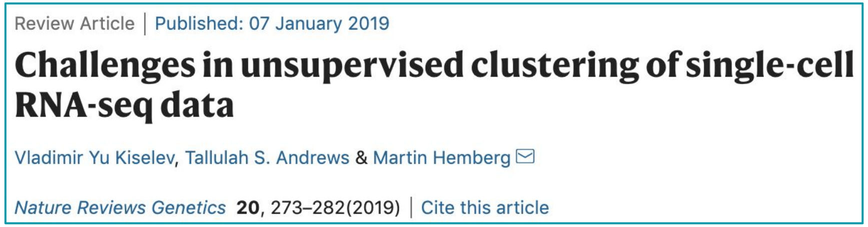
Leung et al. 2016

A variety of research questions in computational biology can be tackled with machine learning

Single-cell RNA-seq clustering for identification of new cell types:

A dimensionality reduction technique is applied to normalized count data

Clustering the data (using e.g., k-means algorithm) can reveal new cell types

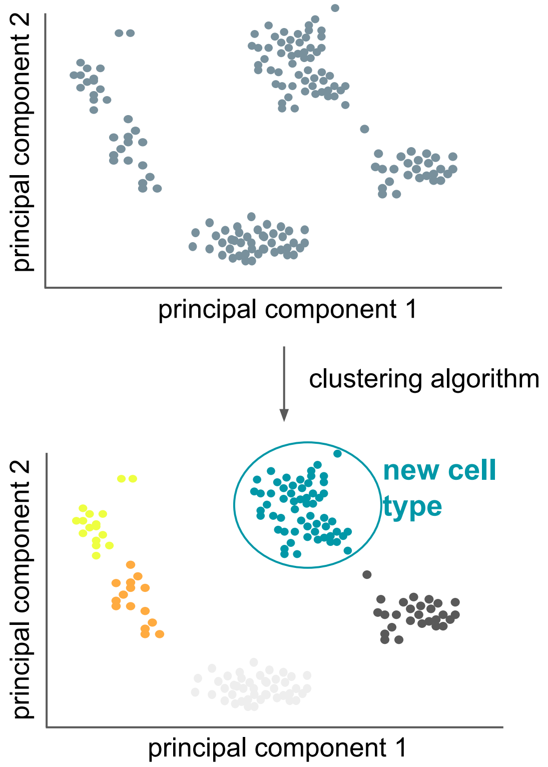


Review Article | Published: 07 January 2019

Challenges in unsupervised clustering of single-cell RNA-seq data

Vladimir Yu Kiselev, Tallulah S. Andrews & Martin Hemberg

Nature Reviews Genetics **20**, 273–282(2019) | Cite this article



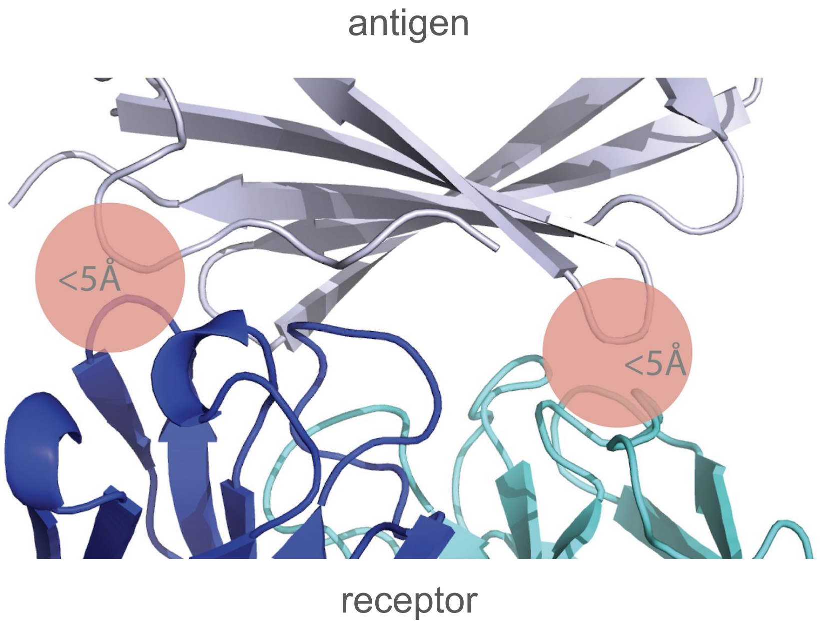
A variety of research questions in computational biology can be tackled with machine learning

Example: antigen binding prediction

Immune receptors (proteins) bind to antigens (e.g., parts of viruses) to help eliminate them

Given a set of receptors known to bind a given antigen, can we predict for the new receptor if it will bind to the antigen?

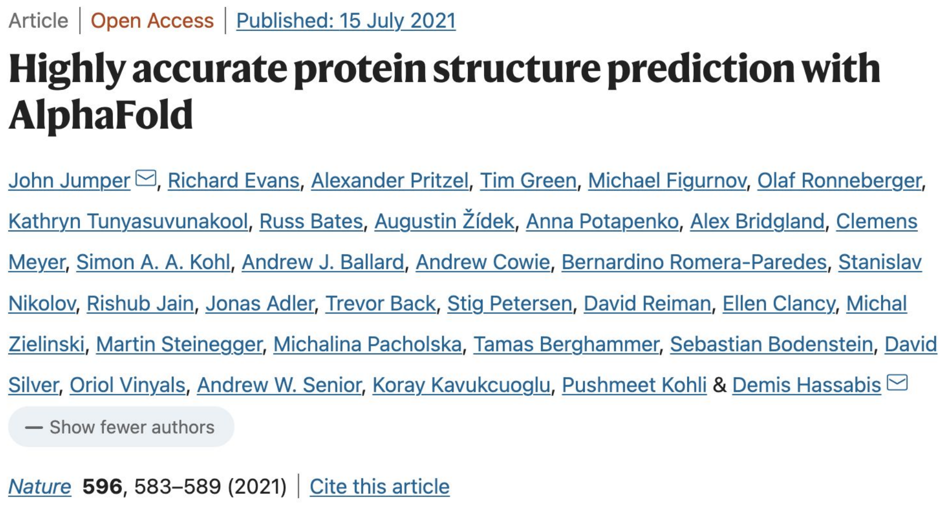
Classification problem!



Akbar et al. 2019

A variety of research questions in computational biology can be tackled with machine learning

Example: protein structure prediction with AlphaFold



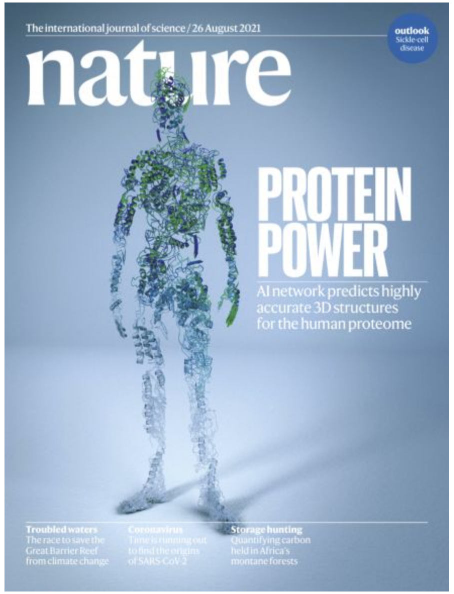
Article | Open Access | Published: 15 July 2021

Highly accurate protein structure prediction with AlphaFold

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli & Demis Hassabis

— Show fewer authors

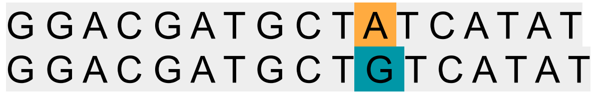
Nature **596**, 583–589 (2021) | Cite this article

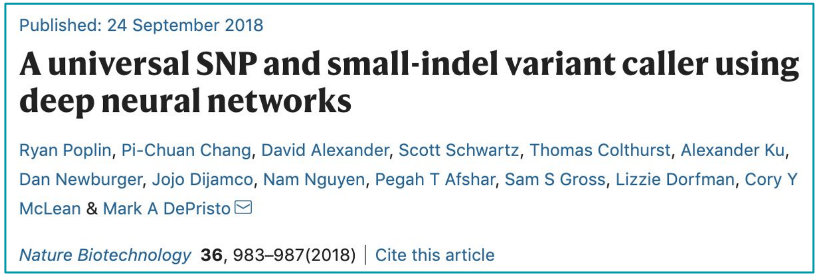

The international journal of science / 26 August 2021
outlook Sickle-cell disease
nature
PROTEIN POWER
AI network predicts highly accurate 3D structures for the human proteome
Troubled waters The race to save the Great Barrier Reef from climate change
Coronavirus Time is running out to find the origins of SARS-CoV-2
Storage hunting Quantifying carbon held in Africa's montane forests

A variety of research questions in computational biology can be tackled with machine learning

Variant calling: finding genetic variants from sequence reads

From a pileup of the reference and read data around each candidate variant, ML models could determine the probabilities for each of the three diploid genotypes

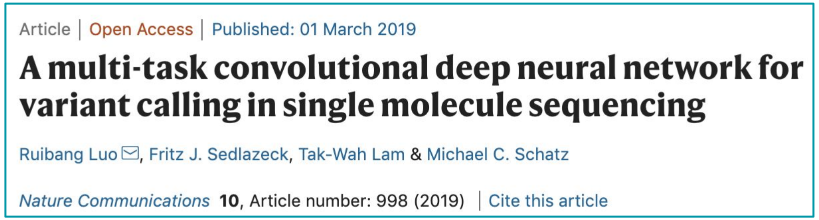

G G A C G A T G C T A T C A T A T
G G A C G A T G C T G T C A T A T


Published: 24 September 2018

A universal SNP and small-indel variant caller using deep neural networks

Ryan Poplin, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan Newburger, Jojo Dijamco, Nam Nguyen, Pegah T Afshar, Sam S Gross, Lizzie Dorfman, Cory Y McLean & Mark A DePristo

Nature Biotechnology **36**, 983–987(2018) | Cite this article

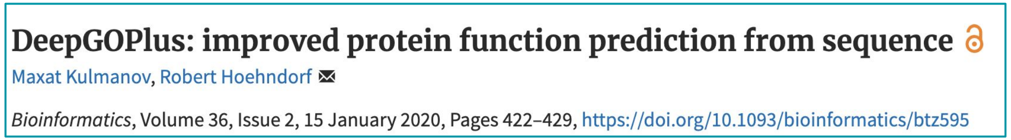

Article | Open Access | Published: 01 March 2019

A multi-task convolutional deep neural network for variant calling in single molecule sequencing

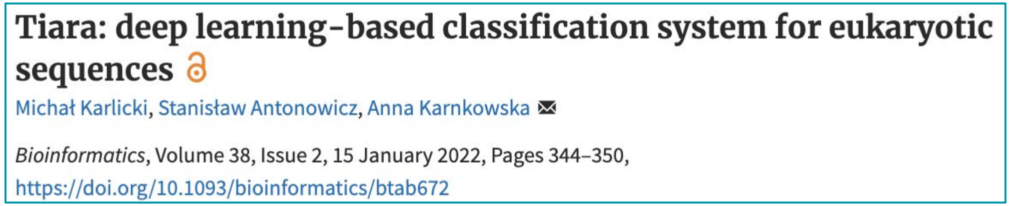
Ruibang Luo, Fritz J. Sedlazeck, Tak-Wah Lam & Michael C. Schatz

Nature Communications **10**, Article number: 998 (2019) | Cite this article

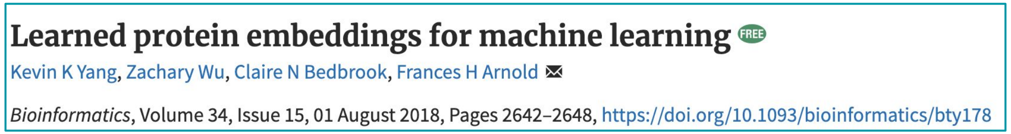
A variety of research questions in computational biology can be tackled with machine learning


DeepGOPlus: improved protein function prediction from sequence
Maxat Kulmanov, Robert Hoehndorf

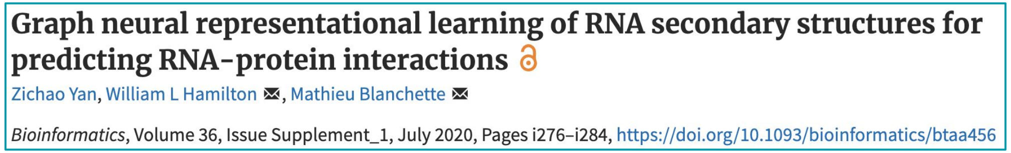
Bioinformatics, Volume 36, Issue 2, 15 January 2020, Pages 422–429, https://doi.org/10.1093/bioinformatics/btz595


Tiara: deep learning-based classification system for eukaryotic sequences
Michał Karlicki, Stanisław Antonowicz, Anna Karnkowska

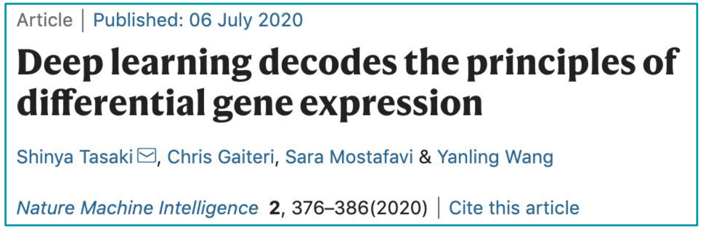
Bioinformatics, Volume 38, Issue 2, 15 January 2022, Pages 344–350, https://doi.org/10.1093/bioinformatics/btab672


Learned protein embeddings for machine learning FREE
Kevin K Yang, Zachary Wu, Claire N Bedbrook, Frances H Arnold

Bioinformatics, Volume 34, Issue 15, 01 August 2018, Pages 2642–2648, https://doi.org/10.1093/bioinformatics/bty178


Graph neural representational learning of RNA secondary structures for predicting RNA-protein interactions
Zichao Yan, William L Hamilton, Mathieu Blanchette

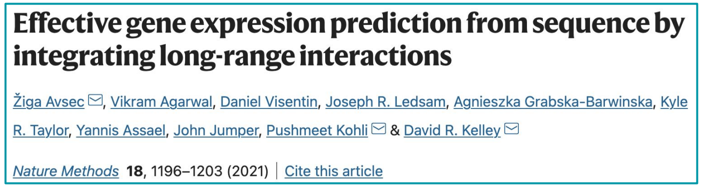
Bioinformatics, Volume 36, Issue Supplement_1, July 2020, Pages i276–i284, https://doi.org/10.1093/bioinformatics/btaa456


Article | Published: 06 July 2020

Deep learning decodes the principles of differential gene expression

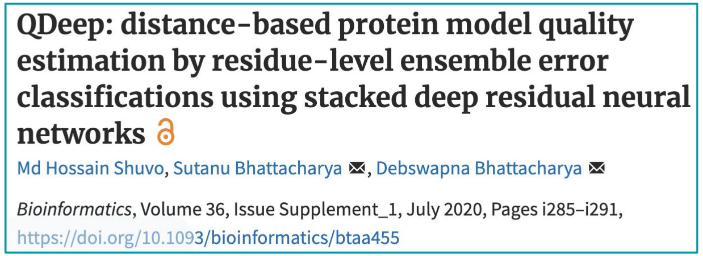
Shinya Tasaki, Chris Gaiteri, Sara Mostafavi & Yanling Wang

Nature Machine Intelligence **2**, 376–386(2020) | Cite this article


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

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli & David R. Kelley

Nature Methods **18**, 1196–1203 (2021) | Cite this article


QDeep: distance-based protein model quality estimation by residue-level ensemble error classifications using stacked deep residual neural networks
Md Hossain Shuvo, Sutanu Bhattacharya, Debswapna Bhattacharya

Bioinformatics, Volume 36, Issue Supplement_1, July 2020, Pages i285–i291, https://doi.org/10.1093/bioinformatics/btaa455

Computational biology poses unique challenges for machine learning

- ❏ Dimensionality & dataset size
- ❏ Signal and noise in the data
- ❏ Unknown ground truth and weakly labeled datasets
- ❏ Selection bias

Keep in the data generation process in mind!

References

Wainberg M, Merico D, Delong A, Frey BJ. Deep learning in biomedicine. *Nature Biotechnology*. 2018;36(9):829-838. doi:10.1038/nbt.4233

Bagaev DV, Vroomans RMA, Samir J, et al. VDJdb in 2019: database extension, new analysis infrastructure and a T-cell receptor motif compendium. *Nucleic Acids Res*. 2020;48(D1):D1057-D1062. doi:10.1093/nar/gkz874

Akbar R, Robert PA, Pavlović M, et al. A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding. *bioRxiv*. Published online November 30, 2019:759498. doi:10.1101/759498

Leung MKK, Delong A, Alipanahi B, Frey BJ. Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets. *Proceedings of the IEEE*. 2016;104(1):176-197. doi:10.1109/JPROC.2015.2494198

Machine learning in computational biology - outline

- Introduction to machine learning:
  - What is machine learning, types of problems, assumptions, workflow, generalization
- Machine learning models and algorithms:
  - Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)
- Data representation:
  - Considerations and examples, one-hot encoding, feature engineering, representation learning
- Model comparison and uncertainty:
  - Model assessment, model selection, uncertainty, cross-validation
- Transparency and reproducibility

Machine learning in computational biology - outline

- **Introduction to machine learning:**
  - **What is machine learning, types of problems, assumptions, workflow, generalization**
- Machine learning models and algorithms:
  - Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)
- Data representation:
  - Considerations and examples, one-hot encoding, feature engineering, representation learning
- Model comparison and uncertainty:
  - Model assessment, model selection, uncertainty, cross-validation
- Transparency and reproducibility

Part 2: Introduction to ML

What is machine learning?

“Machine learning refers to extracting patterns from raw data.”

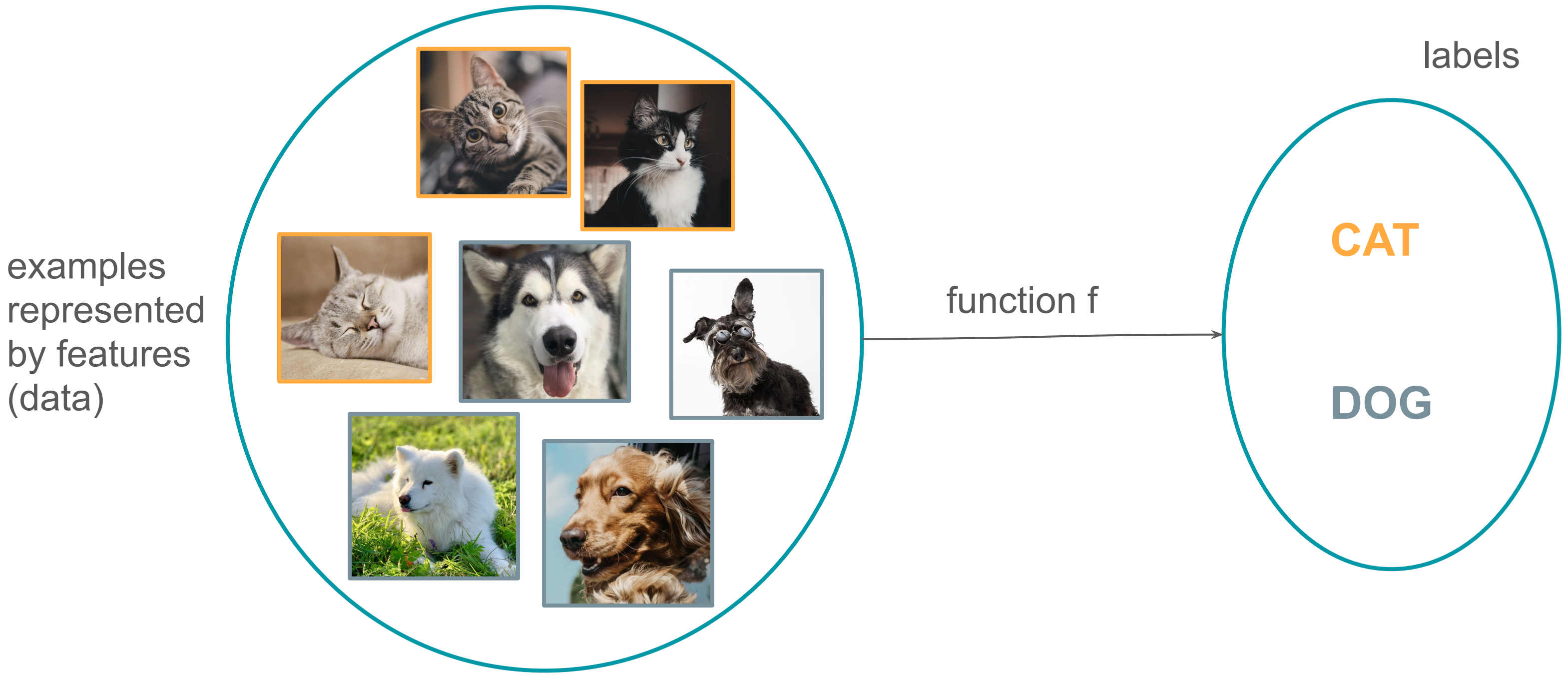
What is machine learning?

“Machine learning refers to extracting patterns from raw data.”

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

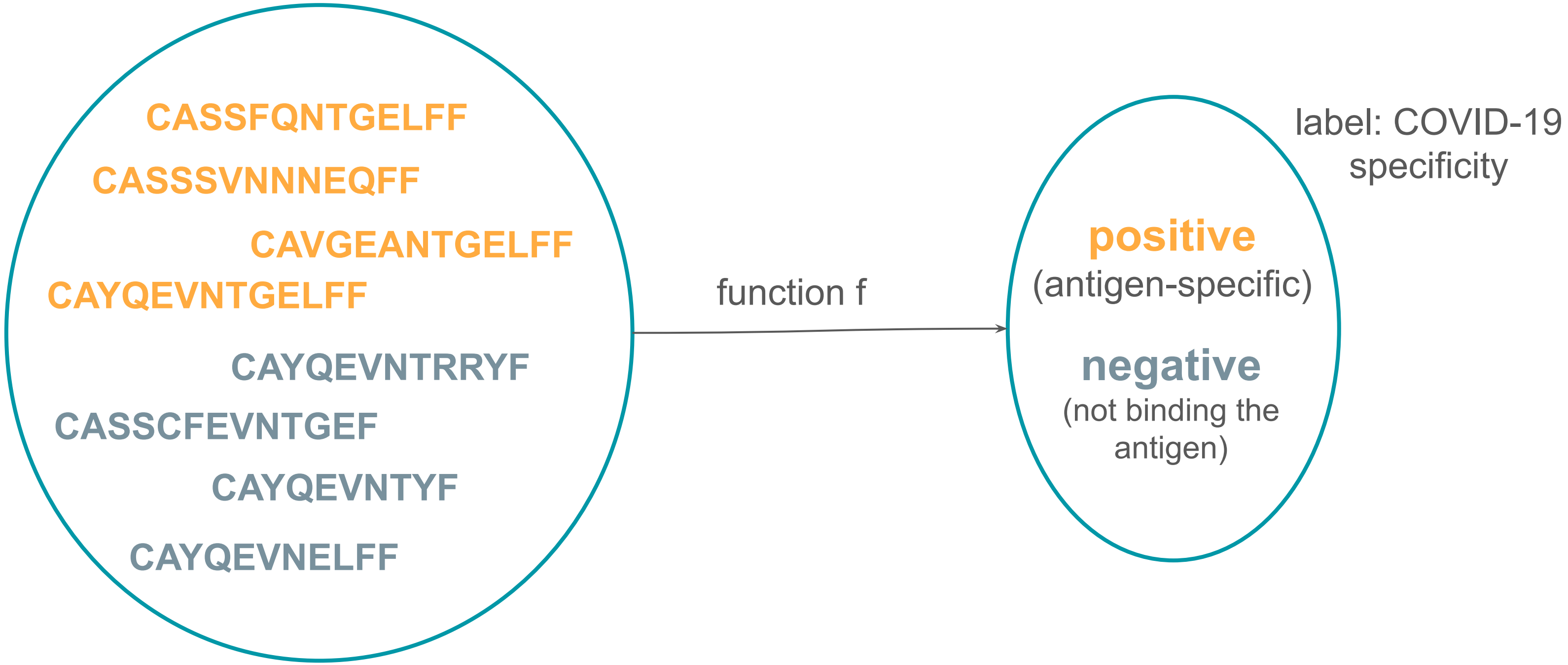
Mitchell 1997

Machine learning as a function approximation task

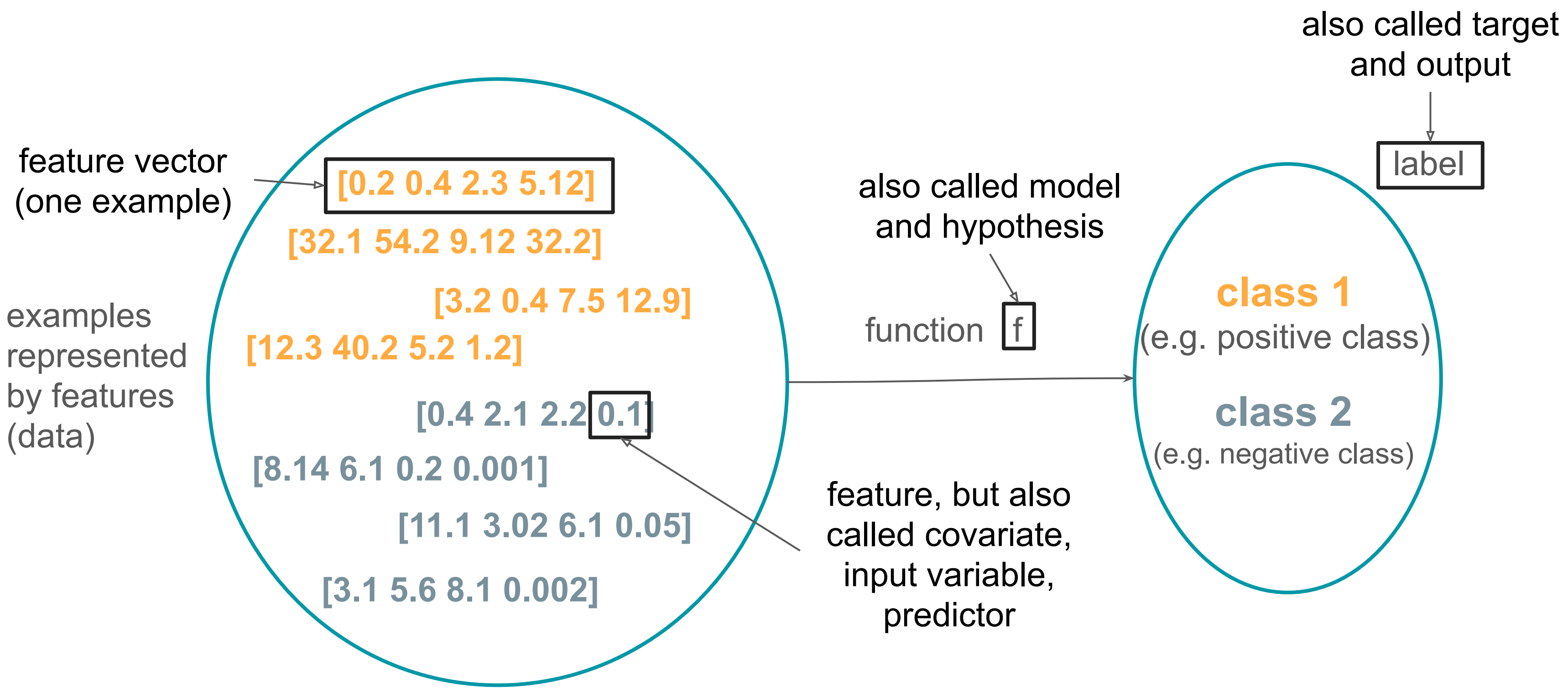


Machine learning as a function approximation task

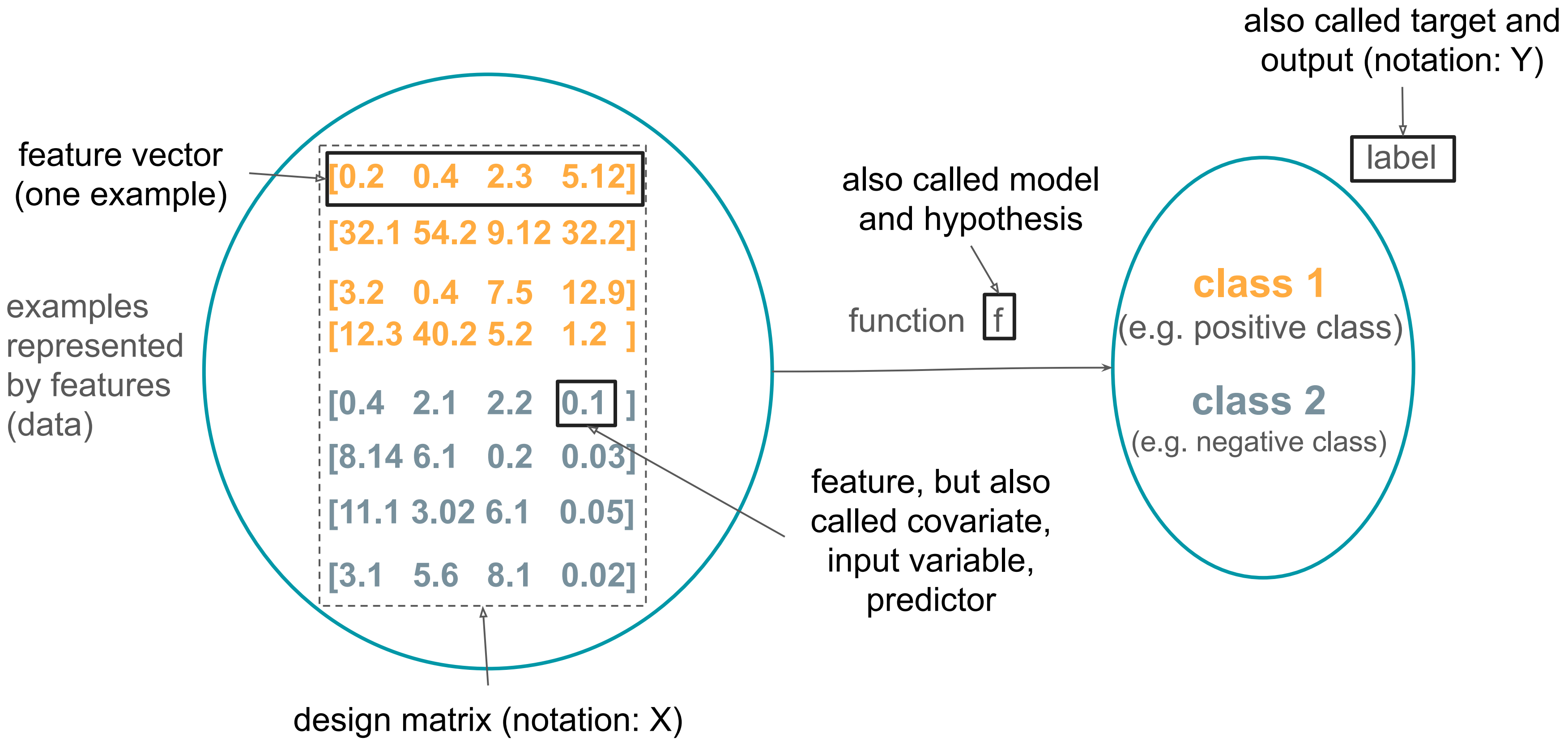
examples represented by features: immune receptor sequences



Machine learning as a function approximation task

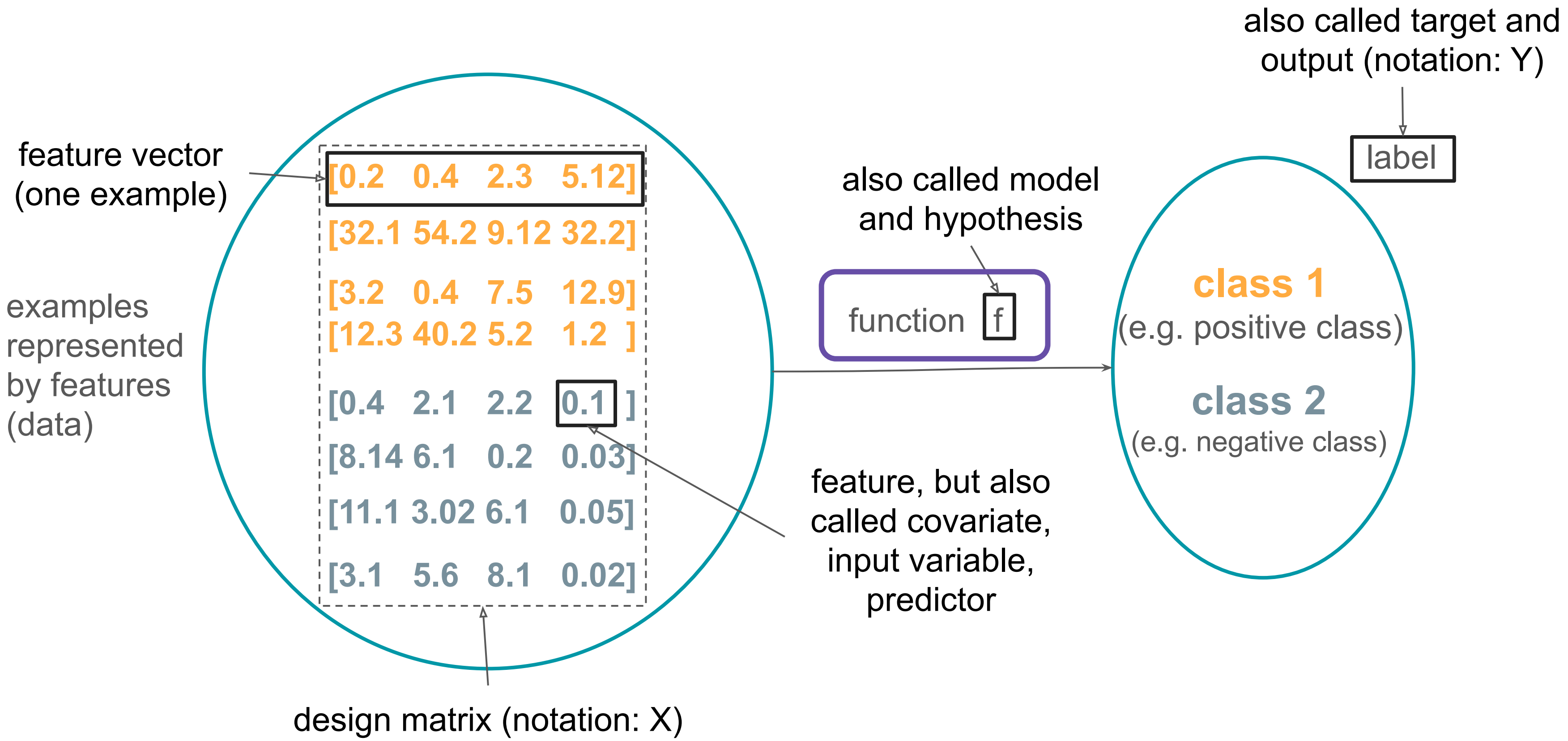


Machine learning as a function approximation task



feature vector (one example)
examples represented by features (data)
[0.2 0.4 2.3 5.12]
[32.1 54.2 9.12 32.2]
[3.2 0.4 7.5 12.9]
[12.3 40.2 5.2 1.2]
[0.4 2.1 2.2 0.1]
[8.14 6.1 0.2 0.03]
[11.1 3.02 6.1 0.05]
[3.1 5.6 8.1 0.02]
design matrix (notation: X)
also called model and hypothesis
function f
feature, but also called covariate, input variable, predictor
also called target and output (notation: Y)
label
class 1
(e.g. positive class)
class 2
(e.g. negative class)

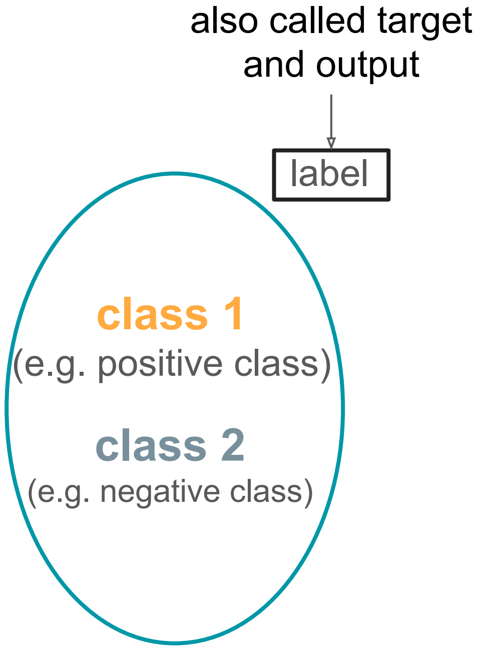
Machine learning as a function approximation task



feature vector (one example)
examples represented by features (data)
[0.2 0.4 2.3 5.12]
[32.1 54.2 9.12 32.2]
[3.2 0.4 7.5 12.9]
[12.3 40.2 5.2 1.2]
[0.4 2.1 2.2 0.1]
[8.14 6.1 0.2 0.03]
[11.1 3.02 6.1 0.05]
[3.1 5.6 8.1 0.02]
design matrix (notation: X)
also called model and hypothesis
function f
feature, but also called covariate, input variable, predictor
also called target and output (notation: Y)
label
class 1
(e.g. positive class)
class 2
(e.g. negative class)

Types of problems in machine learning: what does the function f do

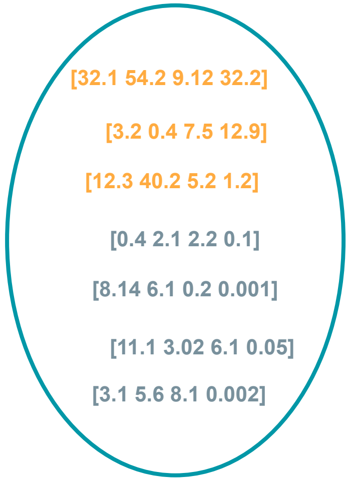
1. **Supervised**: for each example we know the label
    a. Label can be a discrete value - a class (e.g., a receptor is antigen-specific or not, the picture contains a dog or a cat): **classification** or
    b. a continuous value (e.g., binding affinity, house price): **regression**



Types of problems in machine learning: what does the function f do

1. **Supervised**: **classification** and **regression**
2. **Unsupervised**:
   a. the data we have has a lot of features, and we want to see if there is a structure in the data
   b. there is no explicit label
   c. example: there is a set of cells and we want to see if we can group them and see if there are new groups which could indicate new cell types

no label, just data



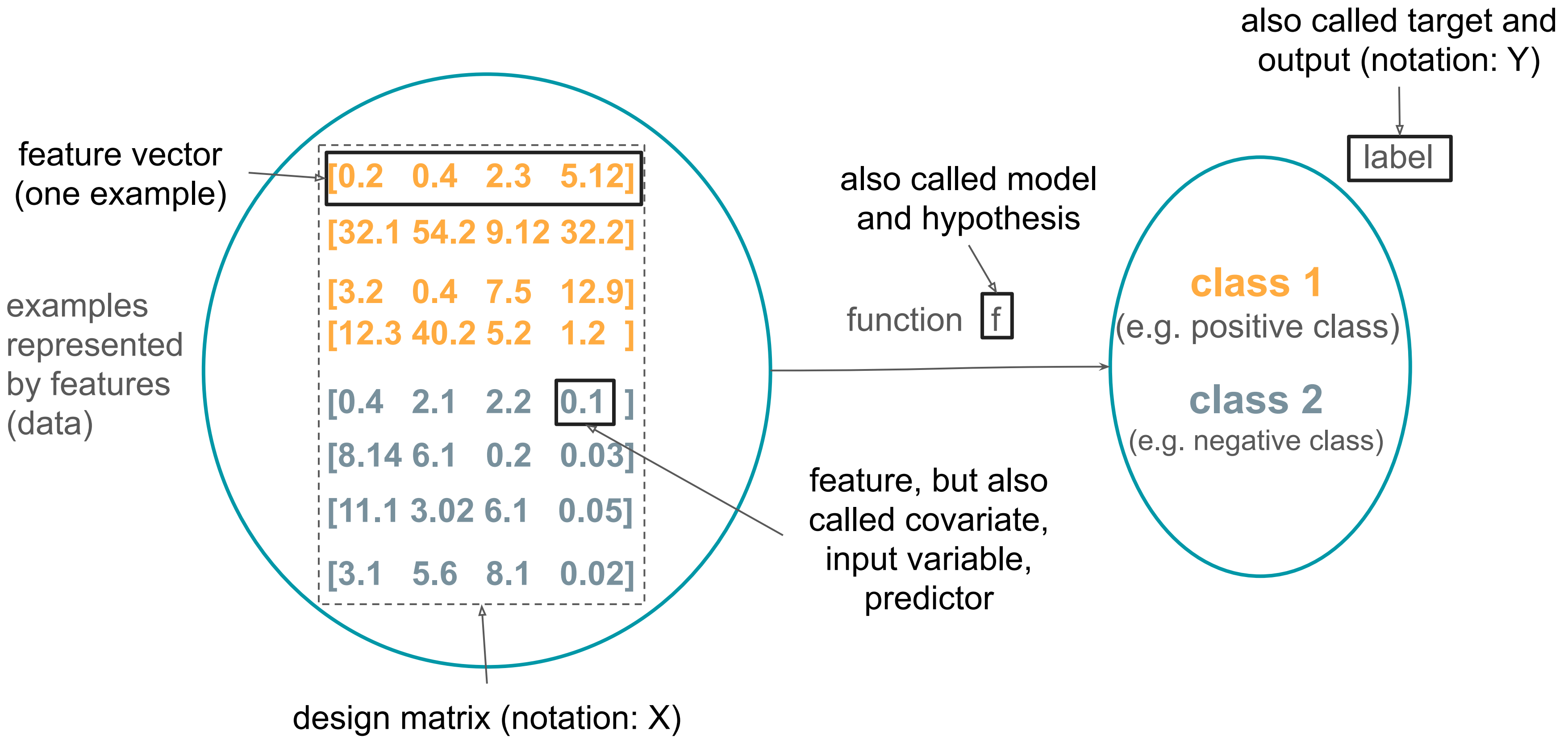
Types of problems in machine learning: what does the function f do

1. **Supervised**: **classification** and **regression**
2. **Unsupervised**
3. **Reinforcement learning**:
    a. dataset is not fixed, the program interacts with environment
    b. used when choosing a sequence of actions: we don’t know the label - don’t know the optimal sequence of actions, but we know how good an action is
    c. example: discover optimal dosing policy for a medication

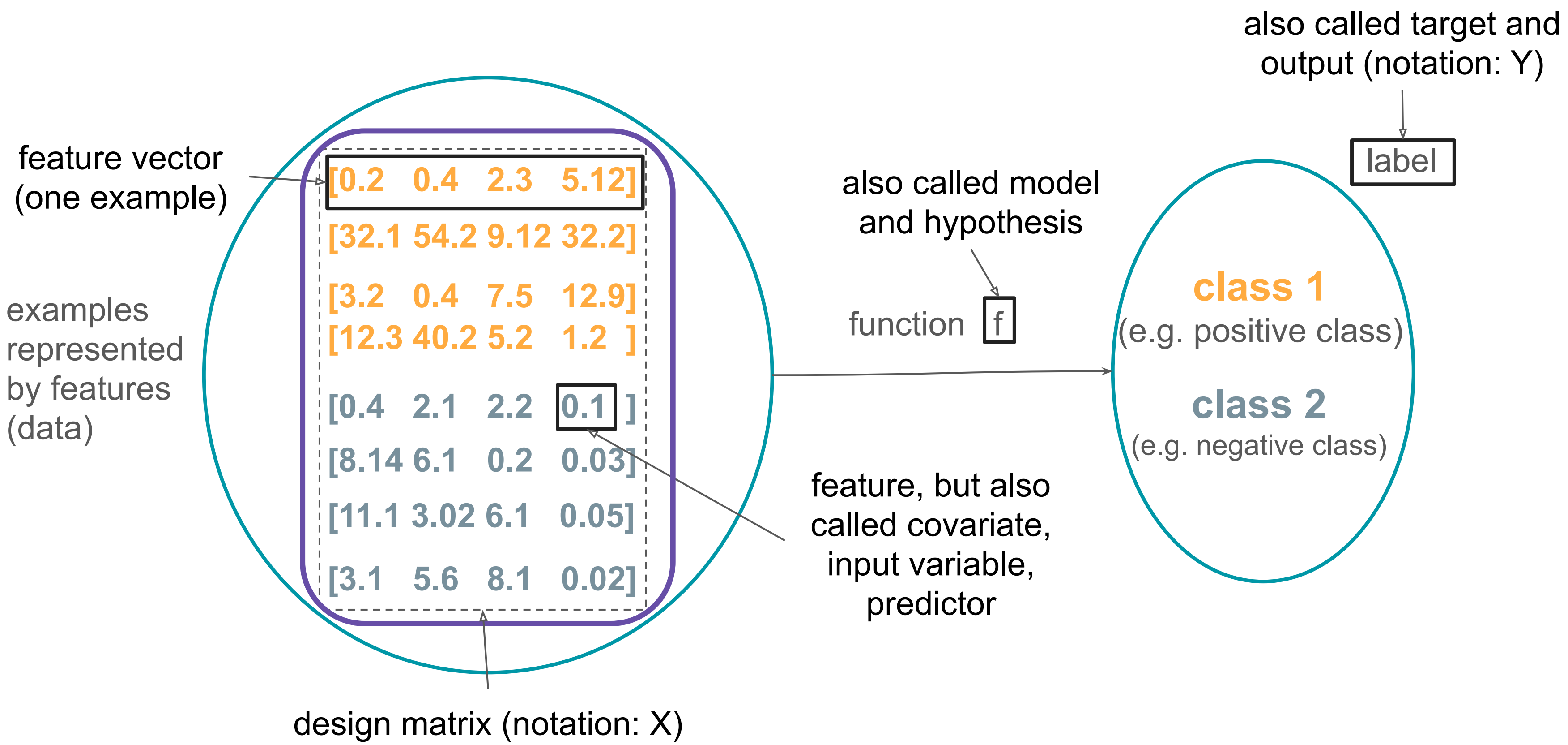
Types of problems in machine learning: what does the function f do

1. **Supervised**: **classification** and **regression**
2. **Unsupervised**
3. **Reinforcement learning**

Machine learning as a function approximation task


also called target and output (notation: Y)
label
feature vector (one example)
[0.2 0.4 2.3 5.12]
[32.1 54.2 9.12 32.2]
[3.2 0.4 7.5 12.9]
[12.3 40.2 5.2 1.2]
[0.4 2.1 2.2 0.1]
[8.14 6.1 0.2 0.03]
[11.1 3.02 6.1 0.05]
[3.1 5.6 8.1 0.02]
examples represented by features (data)
also called model and hypothesis
function
f
class 1
(e.g. positive class)
class 2
(e.g. negative class)
feature, but also called covariate, input variable, predictor
design matrix (notation: X)

Machine learning as a function approximation task


also called target and output (notation: Y)
label
feature vector (one example)
[0.2 0.4 2.3 5.12]
[32.1 54.2 9.12 32.2]
[3.2 0.4 7.5 12.9]
[12.3 40.2 5.2 1.2]
[0.4 2.1 2.2 0.1]
[8.14 6.1 0.2 0.03]
[11.1 3.02 6.1 0.05]
[3.1 5.6 8.1 0.02]
examples represented by features (data)
also called model and hypothesis
function f
class 1
(e.g. positive class)
class 2
(e.g. negative class)
feature, but also called covariate, input variable, predictor
design matrix (notation: X)

What do we assume about the data?

- Data generation process produces the data
  (data generation process results in a probability distribution p_{data})
- We assume:
  - Examples in each dataset are independent of each other
  - When we want to use the machine learning model on some new data to predict a label: these new data come from the same data generation process (same probability distribution)

What do we assume about the data?

- ❏ Data generation process produces the data
  (data generation process results in a probability distribution p_{data})
- ❏ We assume:
  - ❏ Examples in each dataset are independent of each other
  - ❏ When we want to use the machine learning model on some new data to predict a label: these new data come from the same data generation process (same probability distribution)

What do we assume about the data?

- ❑ Data generation process produces the data
  (data generation process results in a probability distribution p_{data})
- ❑ We assume:
  - ❑ Examples in each dataset are independent of each other
  - ❑ When we want to use the machine learning model on some new data to predict a label: these new data come from the same data generation process (same probability distribution)

i.i.d. assumption

$\downarrow$

examples are independent and identically distributed

What do we assume about the data?

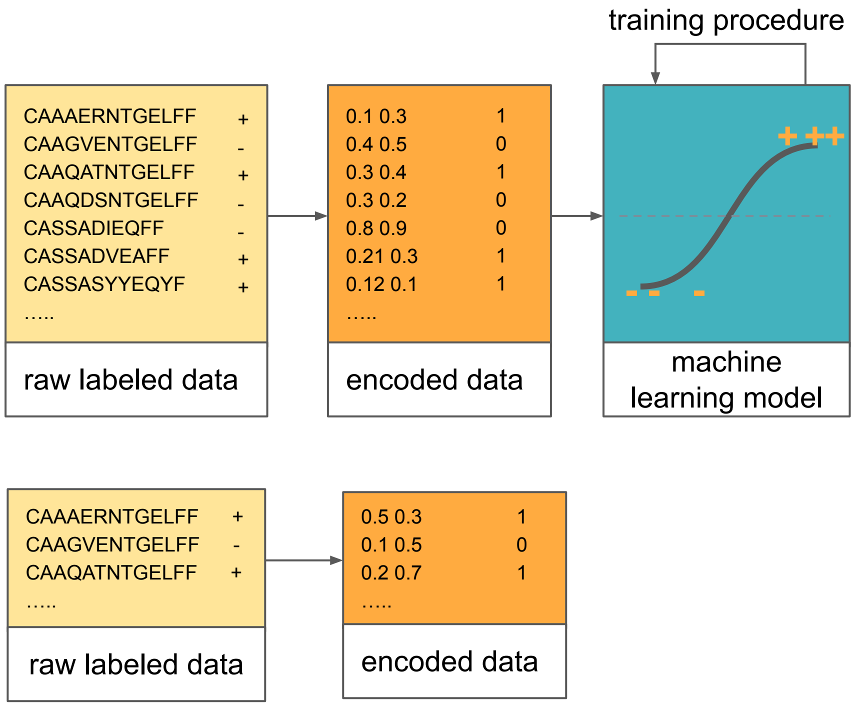
- ❏ Data generation process produces the data
  (data generation process results in a probability distribution p_{data})
- ❏ We assume:
  - ❏ Examples in each dataset are independent of each other
  - ❏ When we want to use the machine learning model on some new data to predict a label: these new data come from the same data generation process (same probability distribution)
- ❏ With these assumptions satisfied (or approximately satisfied), we choose the data representation and estimate the function

i.i.d. assumption

$\downarrow$

examples are independent and identically distributed

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

raw labeled data

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

encoded data

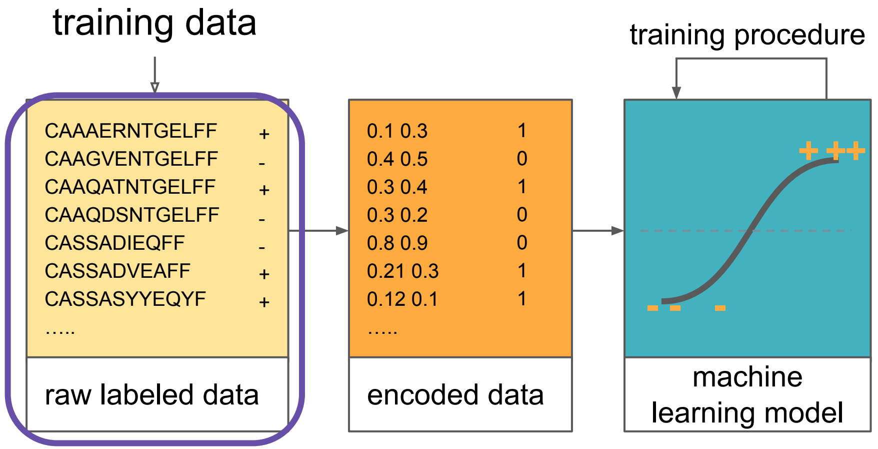
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

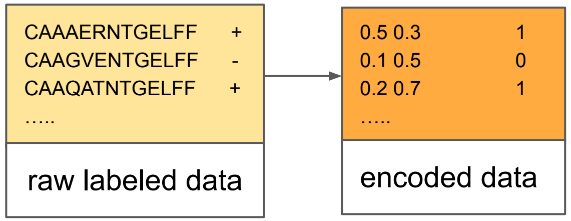
raw labeled data

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

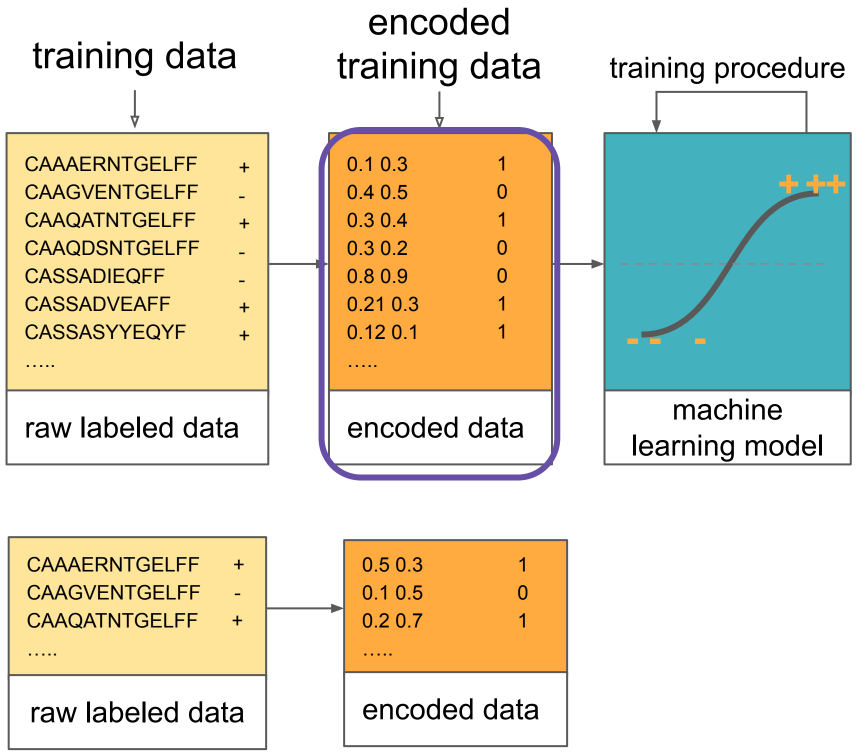
encoded data

Estimating the function (training procedure)





Estimating the function (training procedure)



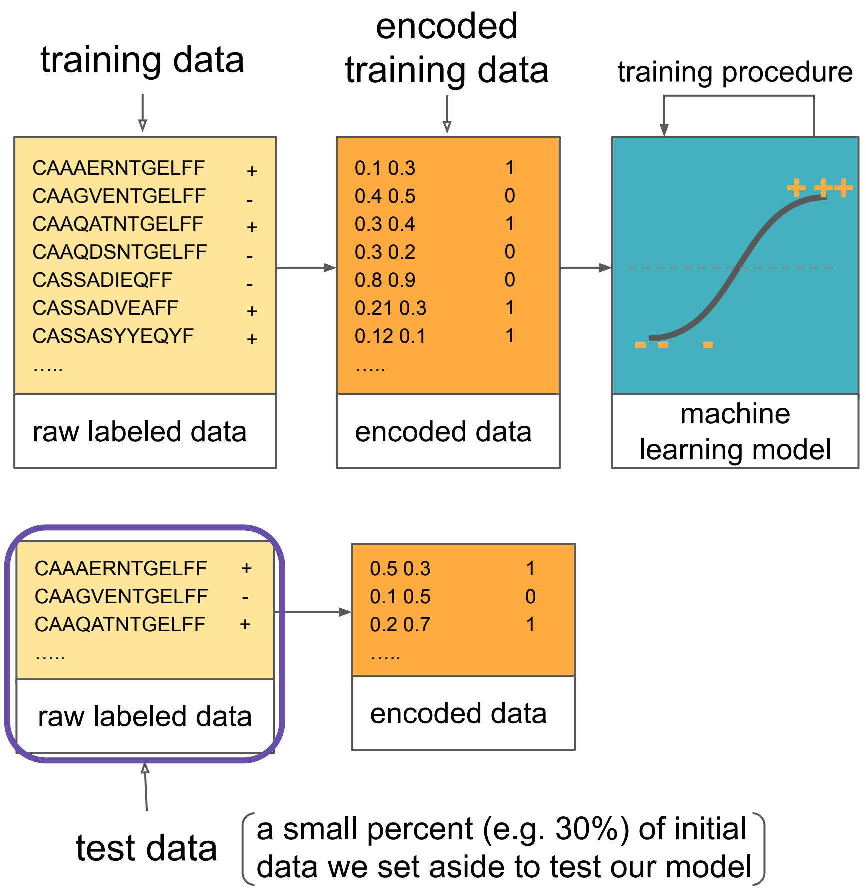
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Estimating the function (training procedure)



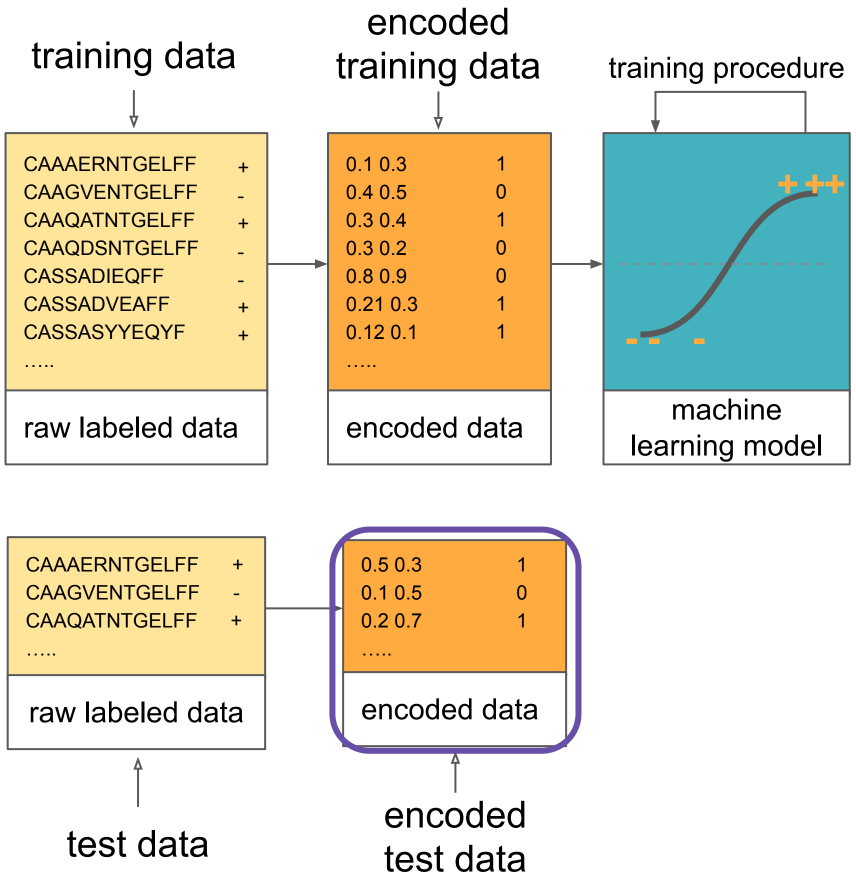
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Estimating the function (training procedure)



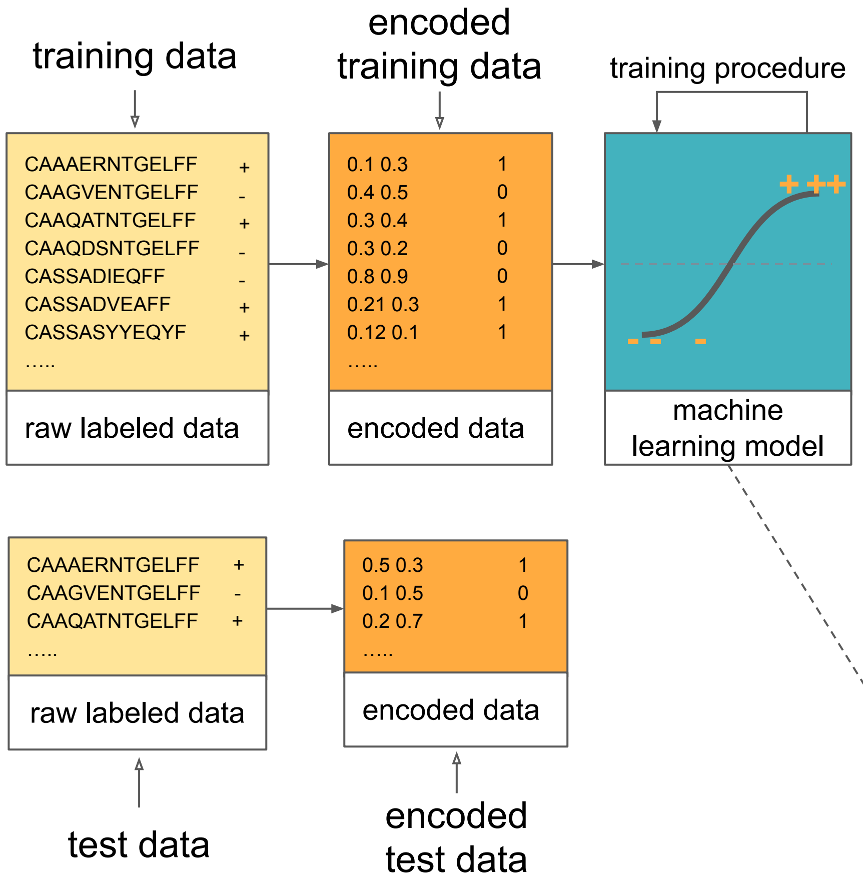
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

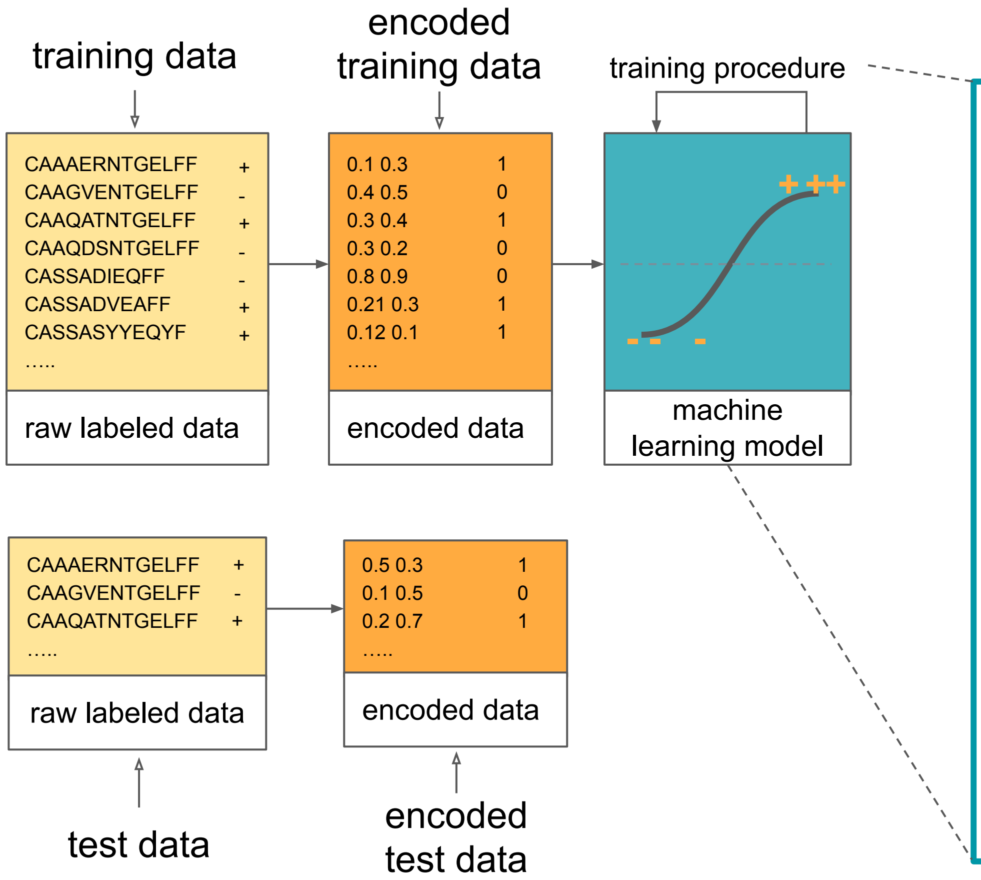
0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

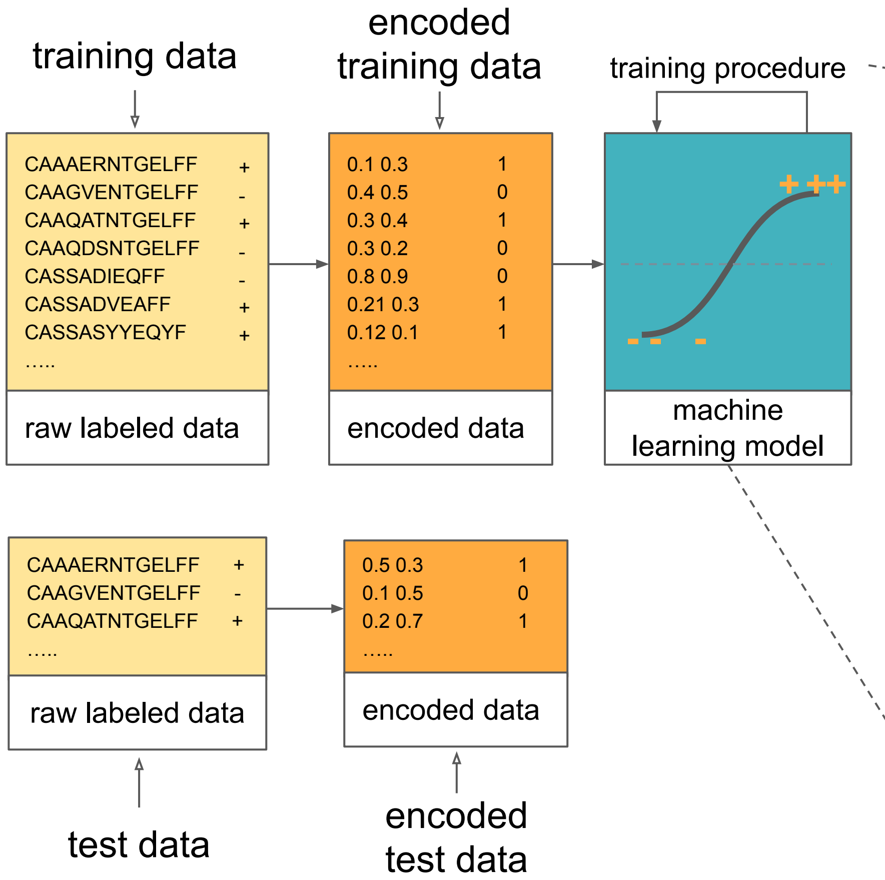
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Training procedure:

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

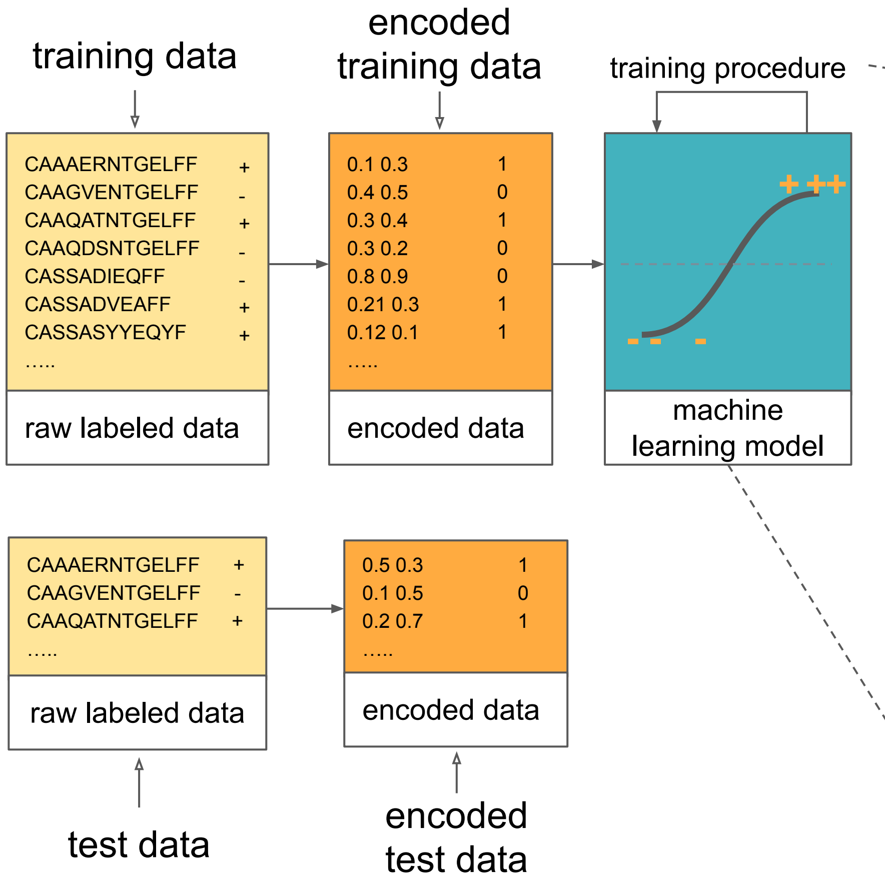
0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
.....	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
.....	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
.....	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
.....	

Task: estimate function f so that f(X) = Y

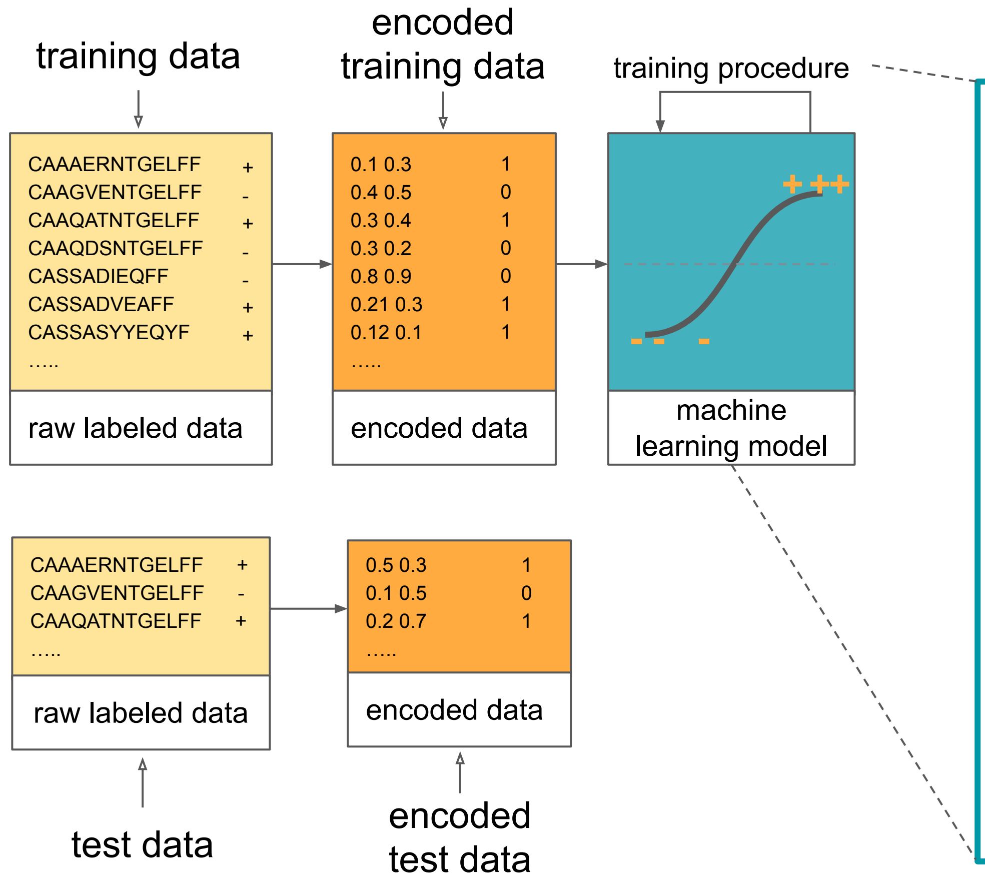
Training procedure:

1. Start with some function f with some parameters
   for example, logistic regression:

$$g(\omega x + b) = \left(1 + e^{-(\omega x + b)}\right)^{-1}$$

$$f(x) = \begin{cases} 1, & g(\omega x + b) \geq 0.5 \\ 0, & g(\omega x + b) < 0.5 \end{cases}$$

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

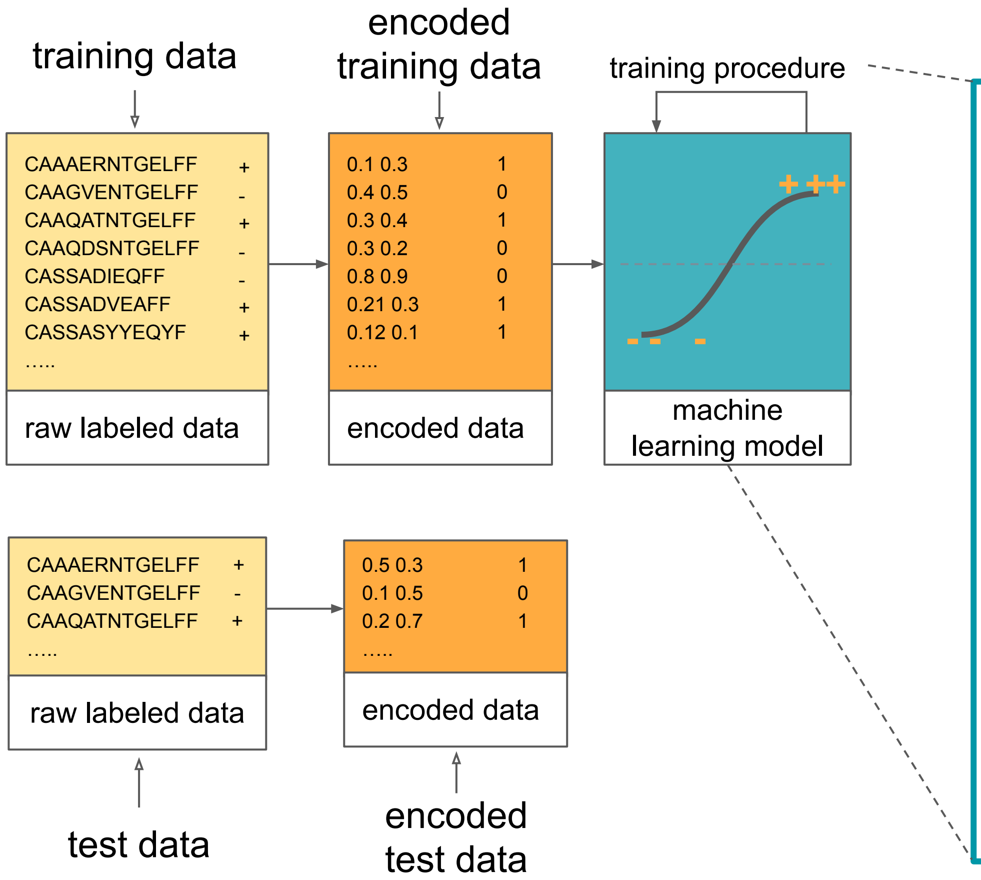
0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
.....	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
.....	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
.....	

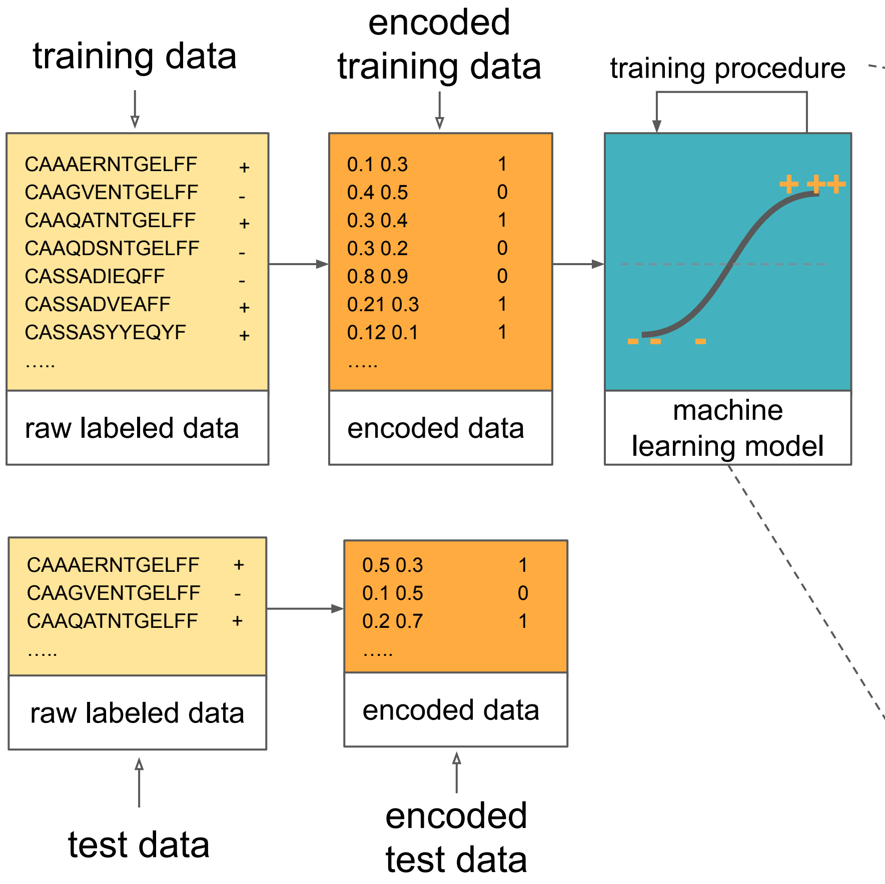
0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
.....	

Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

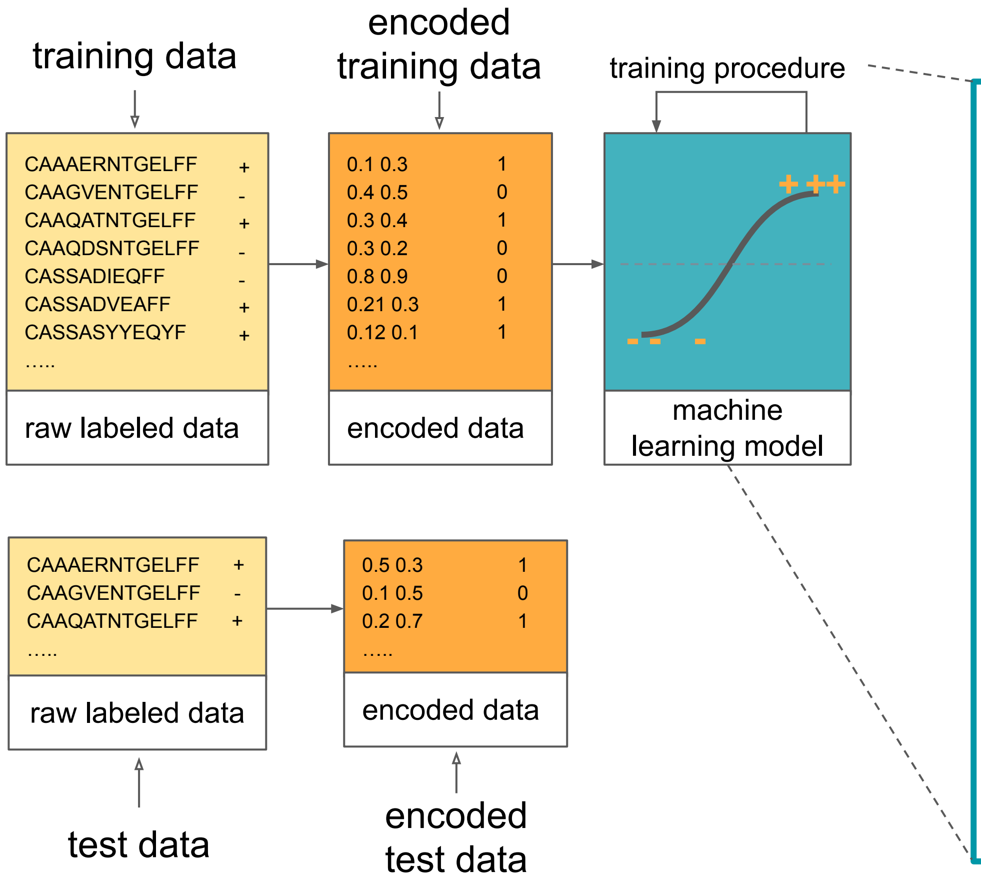
Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:

max number of iterations was not reached and predictions are not good enough

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

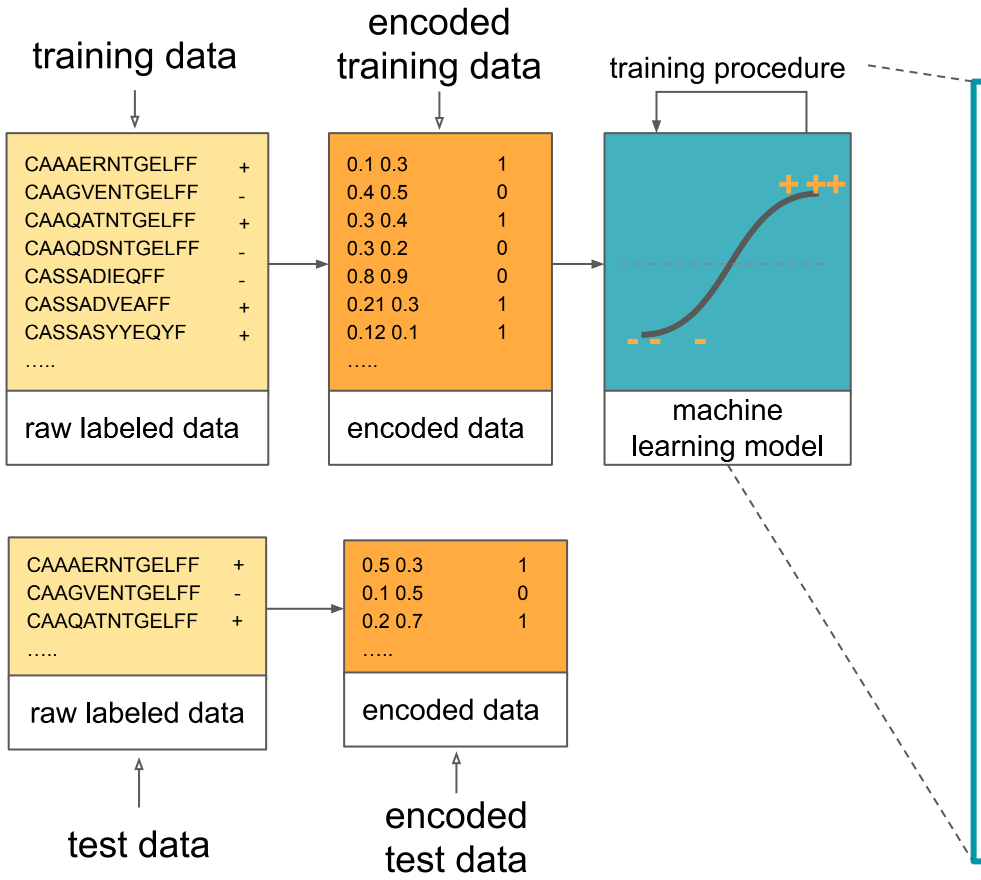
0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:

Estimating the function (training procedure)

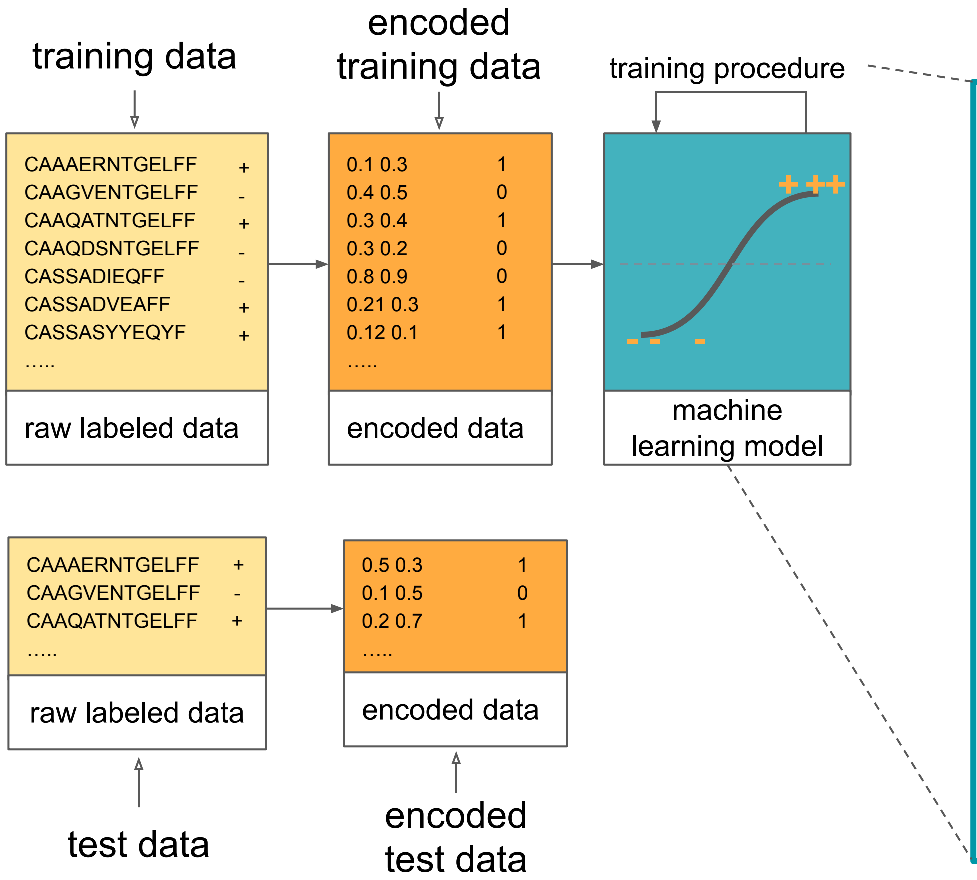


Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:
    a. Predict the label Y from the encoded training data X

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

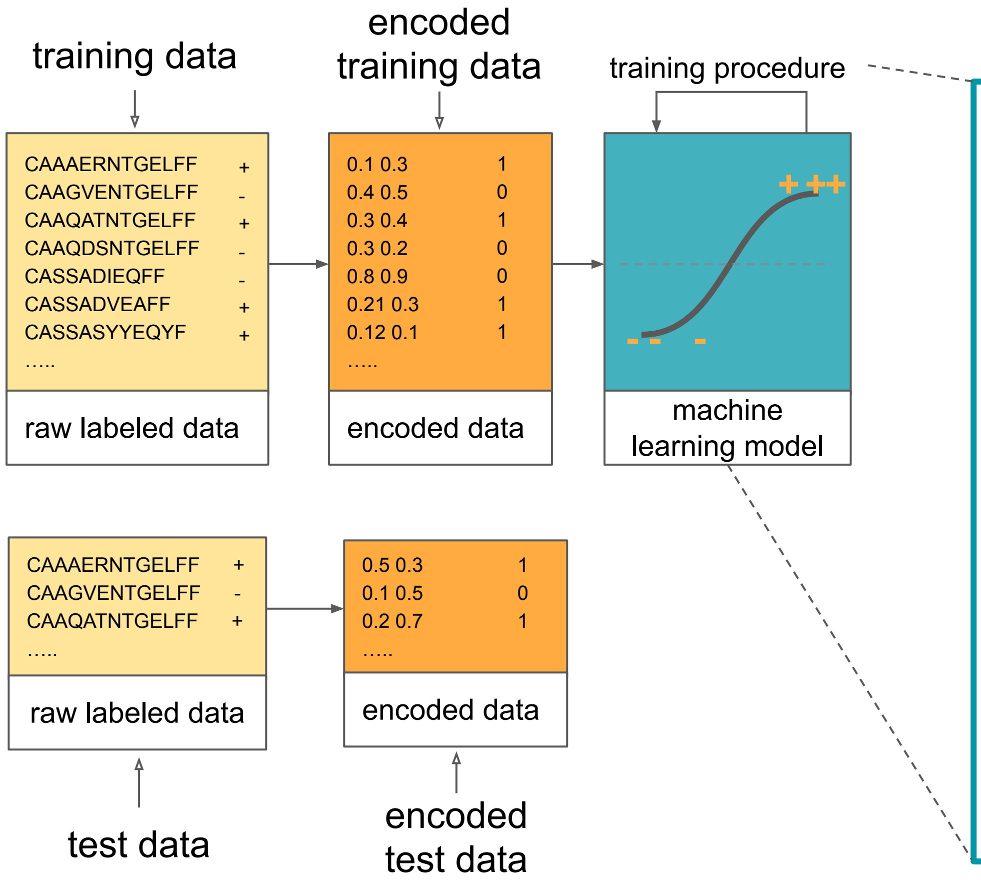
0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:
   a. Predict the label Y from the encoded training data X
   b. Compute the cost function: how much predictions deviate from the label Y

Estimating the function (training procedure)

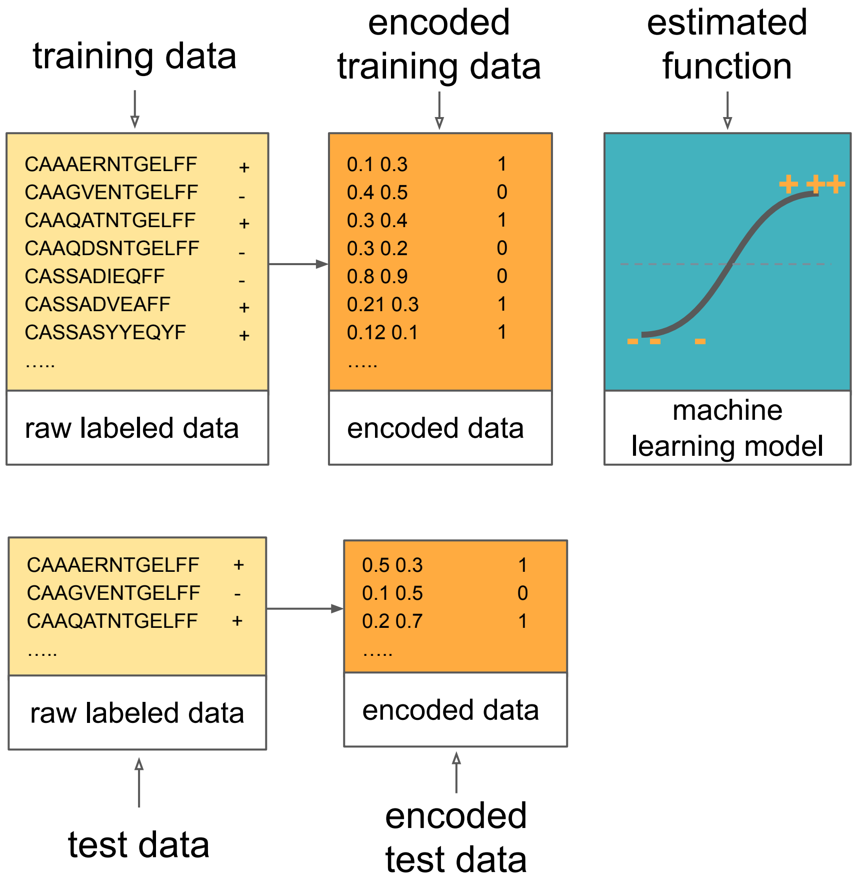


Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters (e.g., logistic regression)
2. While training:
   a. Predict the label Y from the encoded training data X
   b. Compute the cost function: how much predictions deviate from the label Y
   c. Update the parameters of the function f to reduce the cost function so that we get better predictions

Estimating the function (training procedure)



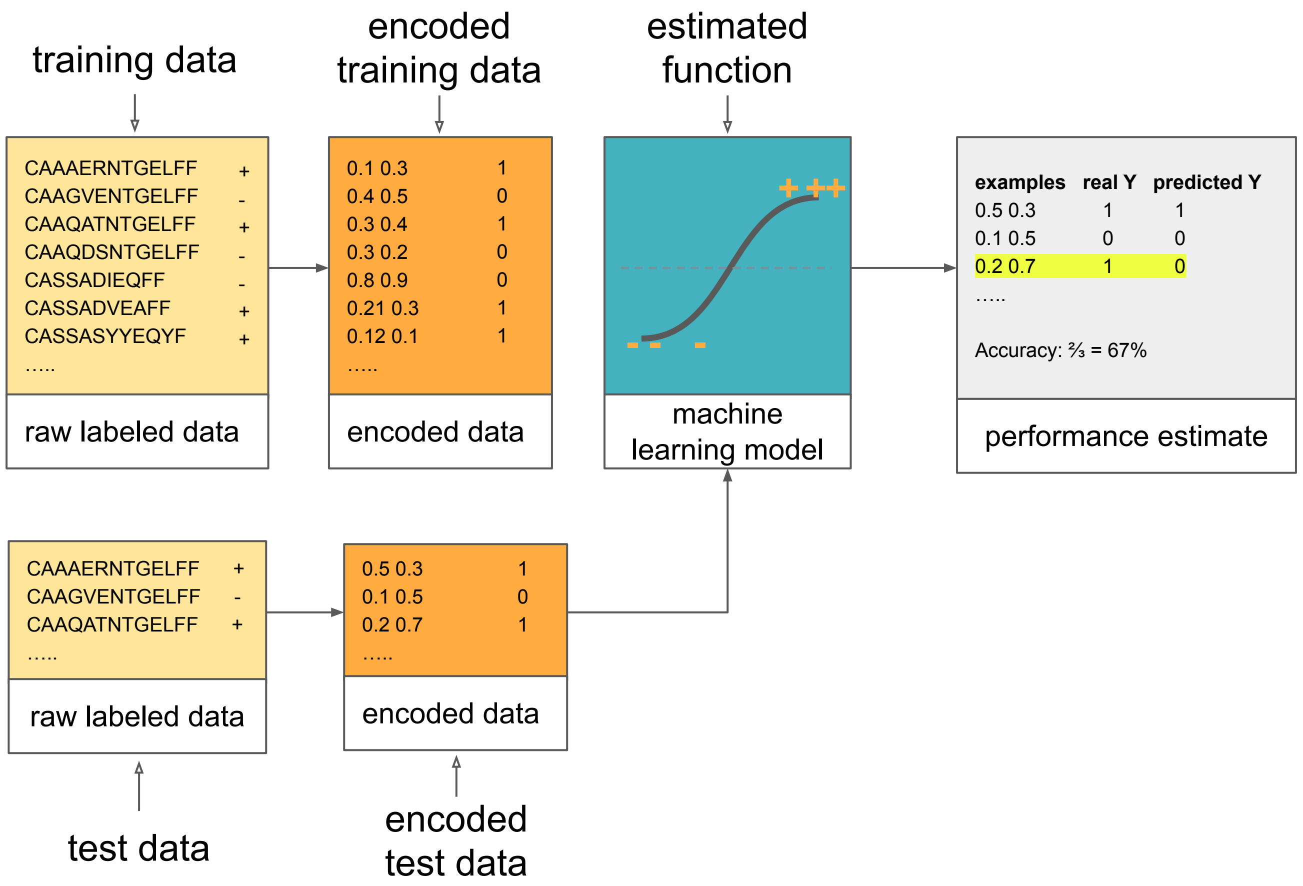
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

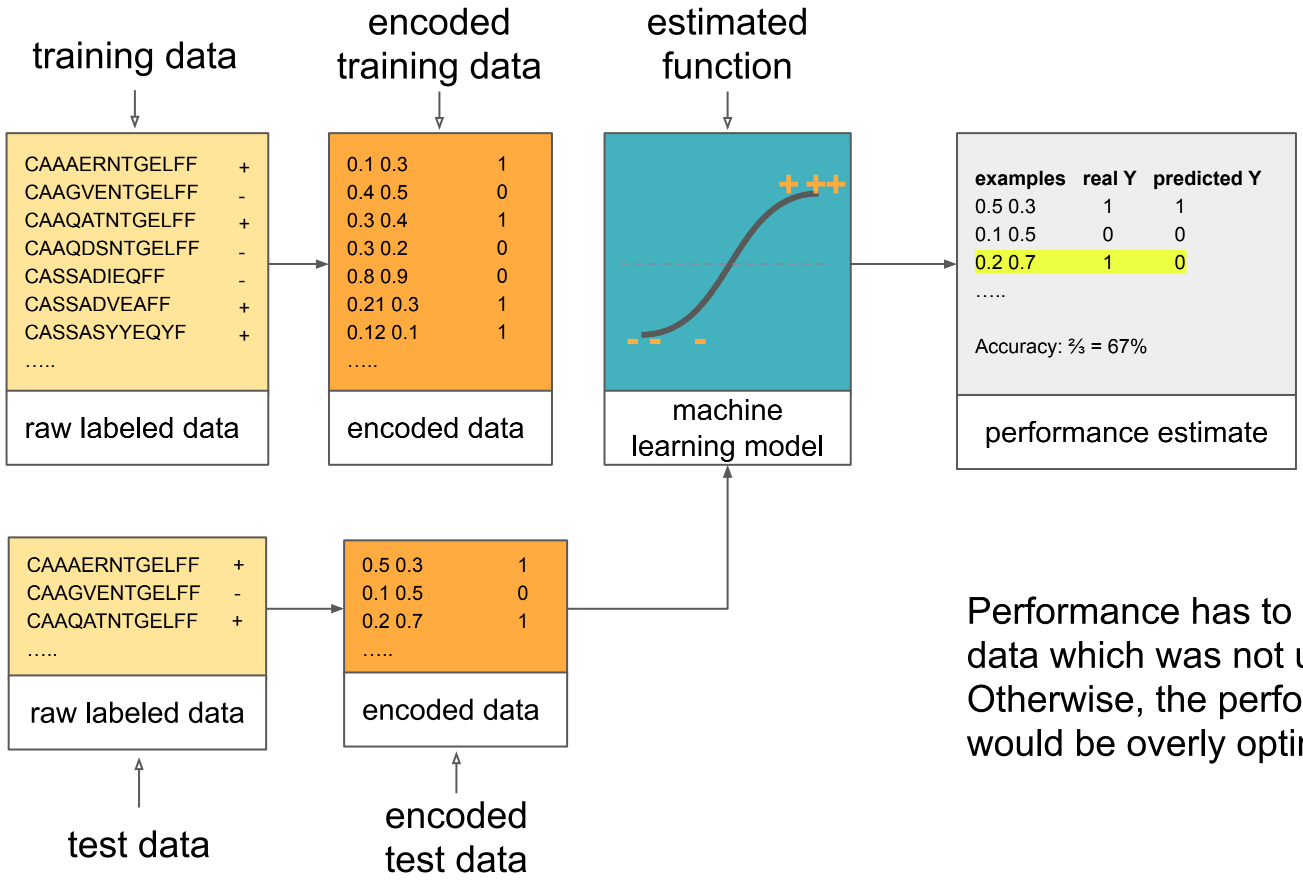
CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

examples	real Y	predicted Y
0.5 0.3	1	1
0.1 0.5	0	0
0.2 0.7	1	0
…..		

Accuracy: ⅔ = 67%

Estimating the function (training procedure)



CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
CAAQDSNTGELFF	-
CASSADIEQFF	-
CASSADVEAFF	+
CASSASYYEQYF	+
…..	

0.1 0.3	1
0.4 0.5	0
0.3 0.4	1
0.3 0.2	0
0.8 0.9	0
0.21 0.3	1
0.12 0.1	1
…..	

examples	**real Y**	**predicted Y**
0.5 0.3	1	1
0.1 0.5	0	0
0.2 0.7	1	0
…..		

Accuracy: ⅔ = 67%

CAAAERNTGELFF	+
CAAGVENTGELFF	-
CAAQATNTGELFF	+
…..	

0.5 0.3	1
0.1 0.5	0
0.2 0.7	1
…..	

Performance has to be estimated on data which was not used during training. Otherwise, the performance estimate would be overly optimistic.

Performance metrics - classification

- ❑ Depends on the problem and the data
- ❑ Classification (label values come from a discrete set):

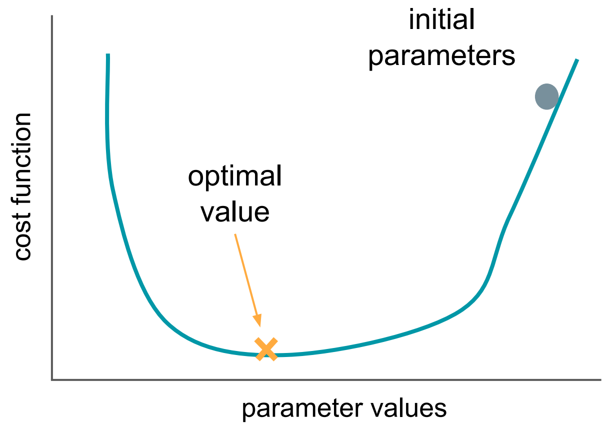
for binary classification, this equality holds:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} \overset{\downarrow}{=} \frac{TP + TN}{\underbrace{\boxed{TP}}_{\text{true positives}} + \underbrace{\boxed{TN}}_{\text{true negatives}} + \underbrace{\boxed{FP}}_{\text{false positives}} + \underbrace{\boxed{FN}}_{\text{false negatives}}}$$

Other metrics: balanced accuracy, precision, recall, sensitivity, specificity, ROC curve, AUC, cross-entropy

Training the machine learning model

- ❏ We want to minimize the cost function
- ❏ For instance, we can use optimization algorithm called gradient descent:

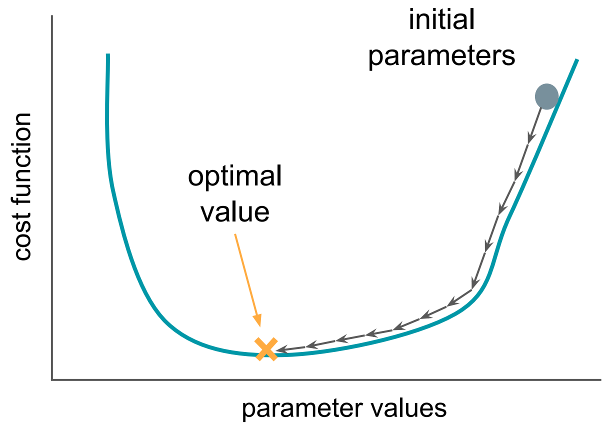


Repeat until optimal solution / max number of iterations:

1. Find derivative of the cost function w.r.t. each of the parameters of the model
2. Update each parameter incrementally using the cost function as a starting point for the update computation

Training the machine learning model

- We want to minimize the cost function
- For instance, we can use optimization algorithm called gradient descent

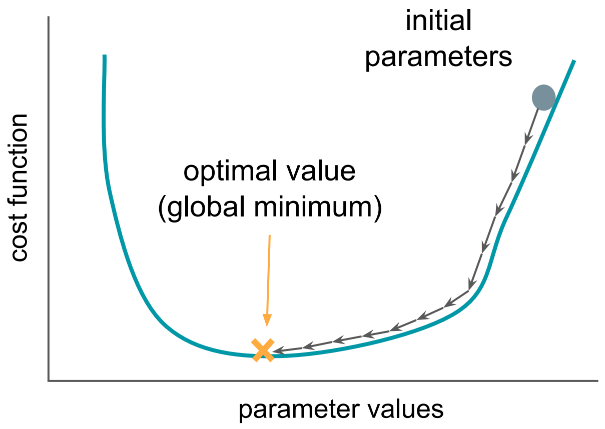


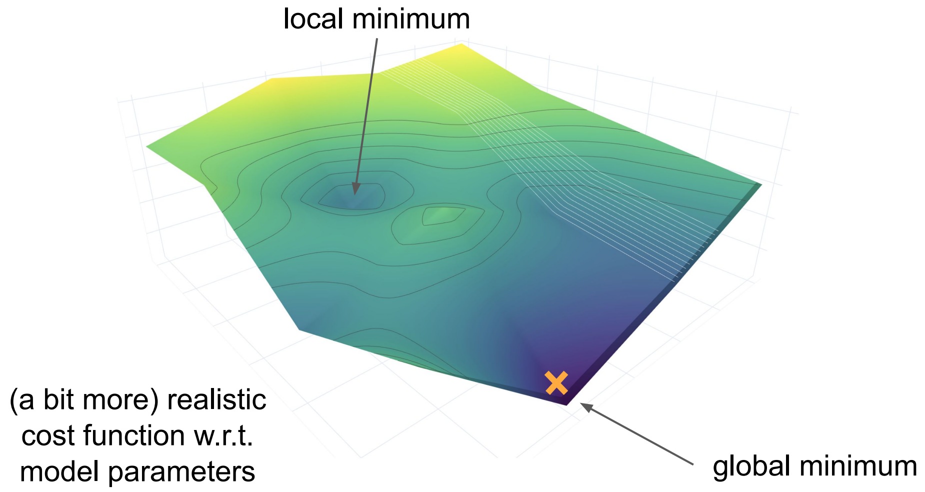
Repeat until optimal solution / max number of iterations:

1. Find derivative of the cost function w.r.t. each of the parameters of the model
2. Update each parameter incrementally using the cost function as a starting point for the update computation

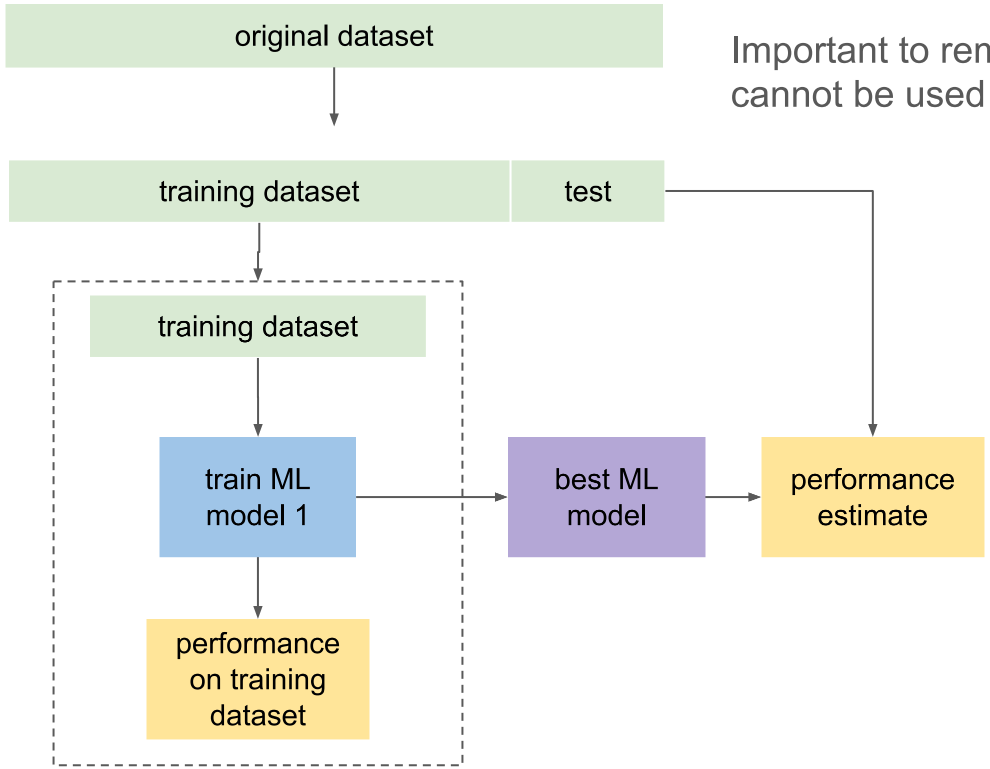
Training the machine learning model

- We want to minimize the cost function
- For instance, we can use optimization algorithm called gradient descent





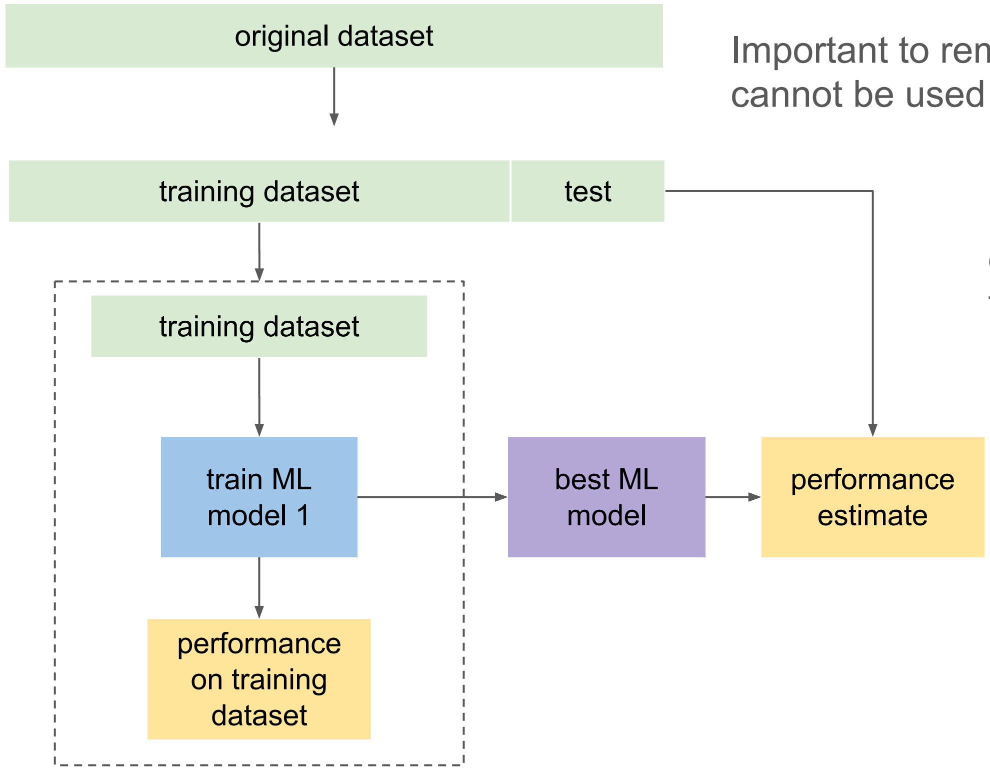
Machine learning workflow



One way to set up a machine learning workflow

Important to remember: data used to assess the performance cannot be used during training

Machine learning workflow



One way to set up a machine learning workflow

Important to remember: data used to assess the performance cannot be used during training

Performance on the test data (not seen during training) will typically be worse than performance on validation

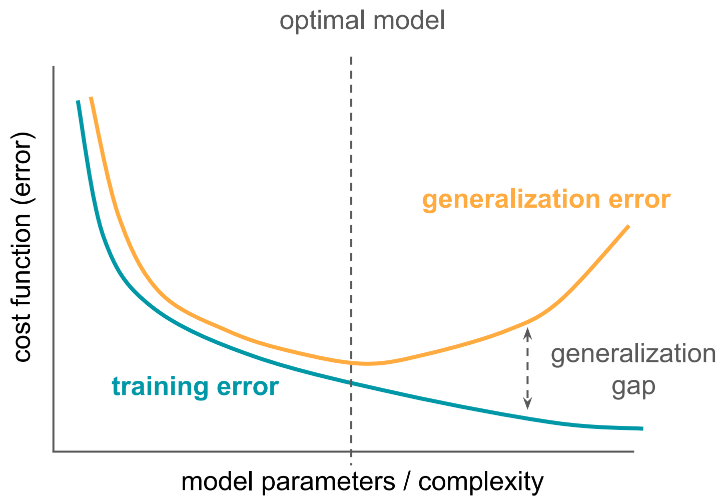
And we will come back to this later...

Generalization in ML

- ❏ Generalization is the ability of an ML model to perform well on previously unseen data
- ❏ We use error on the test set as an estimate of generalization error
- ❏ Generalization error is the expected error on new data

We want a model which will have:

- ❏ Small error on the training set
- ❏ Small gap between training set error and test set error



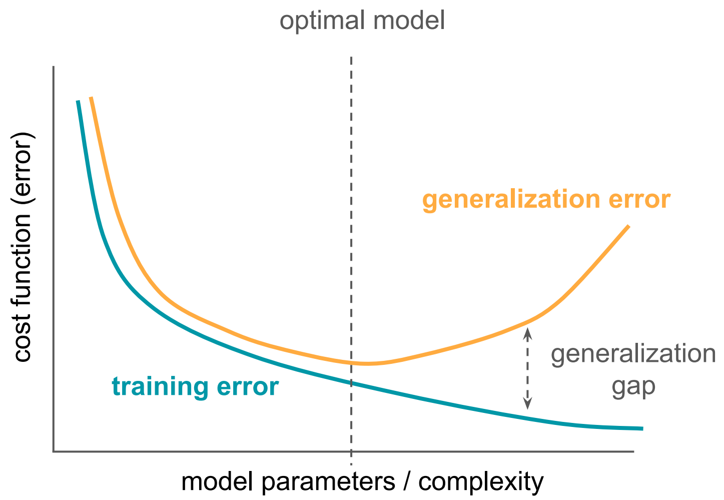
Generalization in ML

Remember that we can talk about generalization like this only if the i.i.d. assumption at least approximately holds.

- ❏ Generalization is the ability of an ML model to perform well on previously unseen data
- ❏ We use error on the test set as an estimate of generalization error
- ❏ Generalization error is the expected error on new data

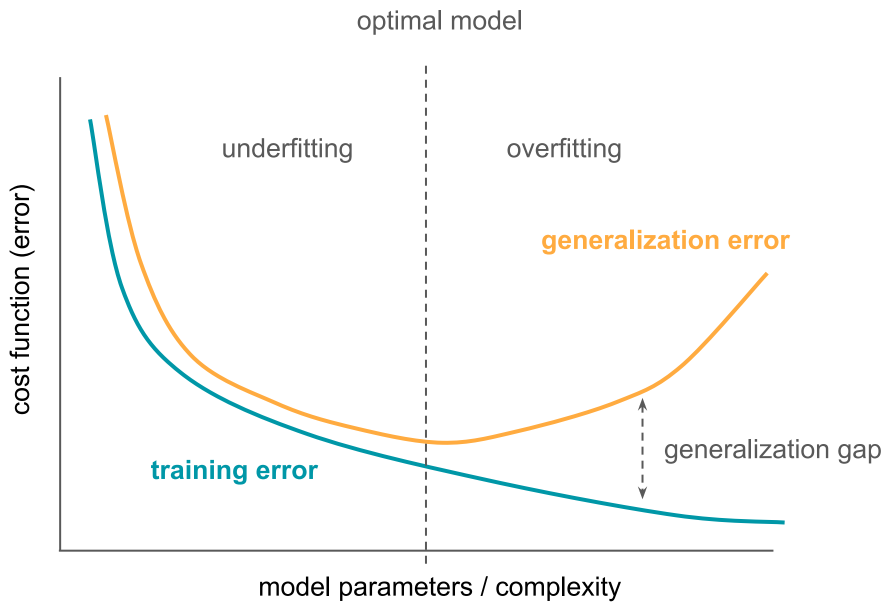
We want a model which will have:

- ❏ Small error on the training set
- ❏ Small gap between training set error and test set error



Overfitting and underfitting

- **Underfitting**: the model was not able to learn from the training data - it had high training error
- **Overfitting**: the generalization gap is too large because the model fit the training data too well but failed to extract patterns which would enable good performance on the new (test) data



References

Goodfellow IJ, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016. https://mitpress.mit.edu/books/deep-learning

Mitchell T. *Machine Learning*. McGraw Hill; 1997. http://www.cs.cmu.edu/~tom/mlbook.html

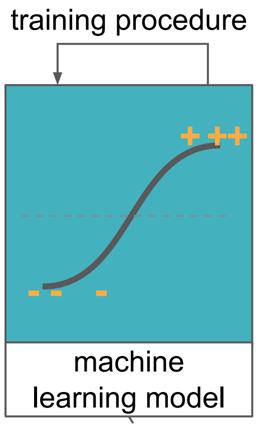
Part 3: ML Models and Algorithms

Machine learning in computational biology - outline

- Introduction to machine learning:
  - What is machine learning, types of problems, assumptions, workflow, generalization
- **Machine learning models and algorithms:**
  - **Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)**
- Data representation:
  - Considerations and examples, one-hot encoding, feature engineering, representation learning
- Model comparison and uncertainty:
  - Model assessment, model selection, uncertainty, cross-validation
- Transparency and reproducibility

ML models

- We mentioned logistic regression before - a simple model for binary classification



Task: estimate function f so that f(X) = Y

Training procedure:

1. Start with some function f with some parameters
   for example, logistic regression:

$$g(\omega x + b) = \left(1 + e^{-(\omega x + b)}\right)^{-1}$$

$$f(x) = \begin{cases} 1, & g(\omega x + b) \geq 0.5 \\ 0, & g(\omega x + b) < 0.5 \end{cases}$$

Some terminology regarding ML models and algorithms

- **Learning algorithm**: a function that, given a set of examples and their labels, constructs a model, e.g., logistic regression
- **Model**: a function which was fit to the data using the learning algorithm, e.g., logistic regression with specific coefficients

Dietterich 1998

Usually model and learning algorithm are used interchangeably but they mean slightly different things

Capacity of the model

- A model's capacity is its ability to fit a wide variety of functions, for instance:

  linear regression:

  $$\hat{y} = b + \omega x$$

  polynomial regression:

  $$\hat{y} = b + \omega_1 x + \omega_2 x^2$$

Capacity of the model

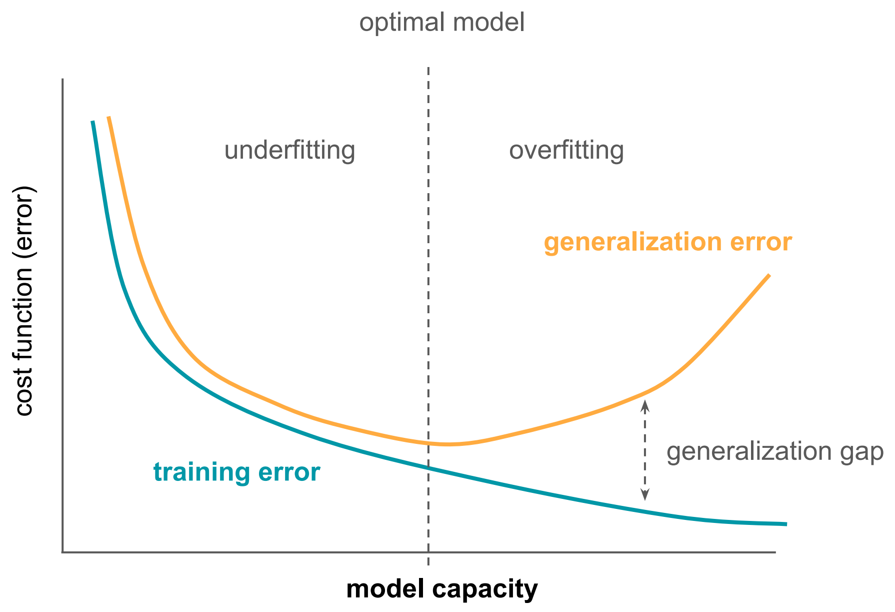
- A model's capacity is its ability to fit a wide variety of functions, for instance:

  linear regression:

$$\hat{y} = b + \omega x$$

  polynomial regression:

$$\hat{y} = b + \omega_1 x + \omega_2 x^2$$

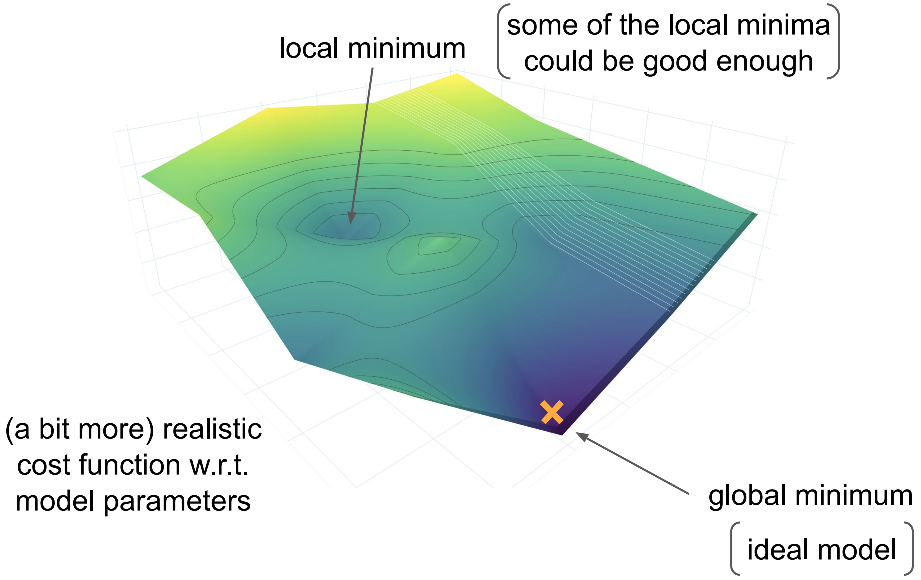


Searching through a hypothesis space using optimization algorithms

Fitting the parameters of the model is an optimization problem (there are different optimization algorithms, but one example is **gradient descent**)

We can get stuck in local minima

If the performance is good enough, we can accept that “suboptimal” model


local minimum
some of the local minima could be good enough
(a bit more) realistic cost function w.r.t. model parameters
global minimum
ideal model

Regularization restricts what models can be learned

Regularization – restricting the values of parameters of the model that can be learned (e.g., based on our domain knowledge) so that the model performs better on new data

Typical forms of regularization (also called penalty):

L1 (lasso):

$$regularization\,(parameters) \;=\; \sum_i |\,parameter_i\,|$$

L2 (ridge):

$$regularization\,(parameters) \;=\; \sum_i parameter_i^{\,2}$$

Logistic Regression

- An algorithm for binary classification:

function computing log-odds for the positive class

$$\boxed{g}(\boxed{\omega}\boxed{x} + \boxed{b}) = \left(1 + e^{-\boxed{(\omega x + b)}}\right)^{-1}$$

model parameters (coefficients)

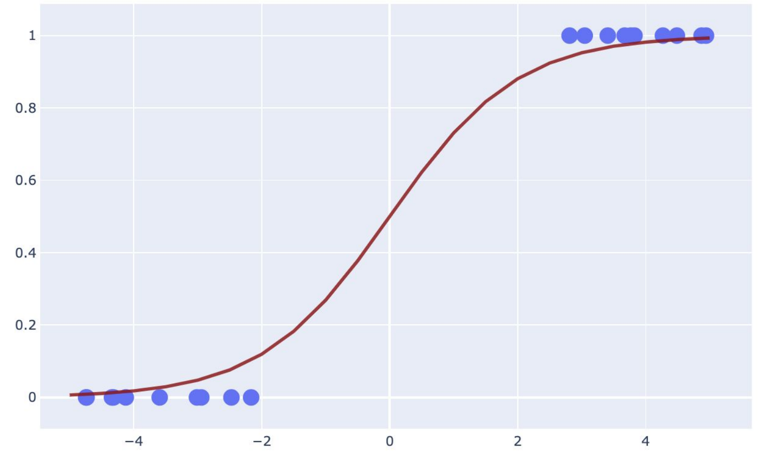
model parameter (bias or intercept)

linear combination of features

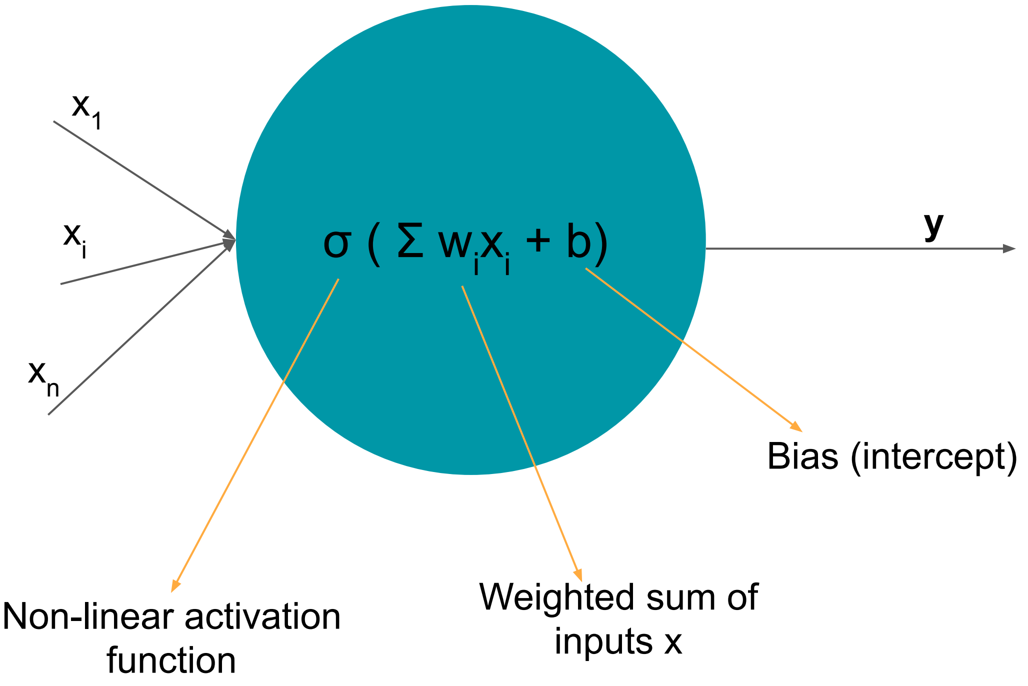
data (design matrix or feature vector)

function making the class prediction based on the threshold from log-odds value

$$\boxed{f(x)} = \begin{cases} 1, & g(\omega x + b) \geq 0.5 \\ 0, & g(\omega x + b) < 0.5 \end{cases}$$

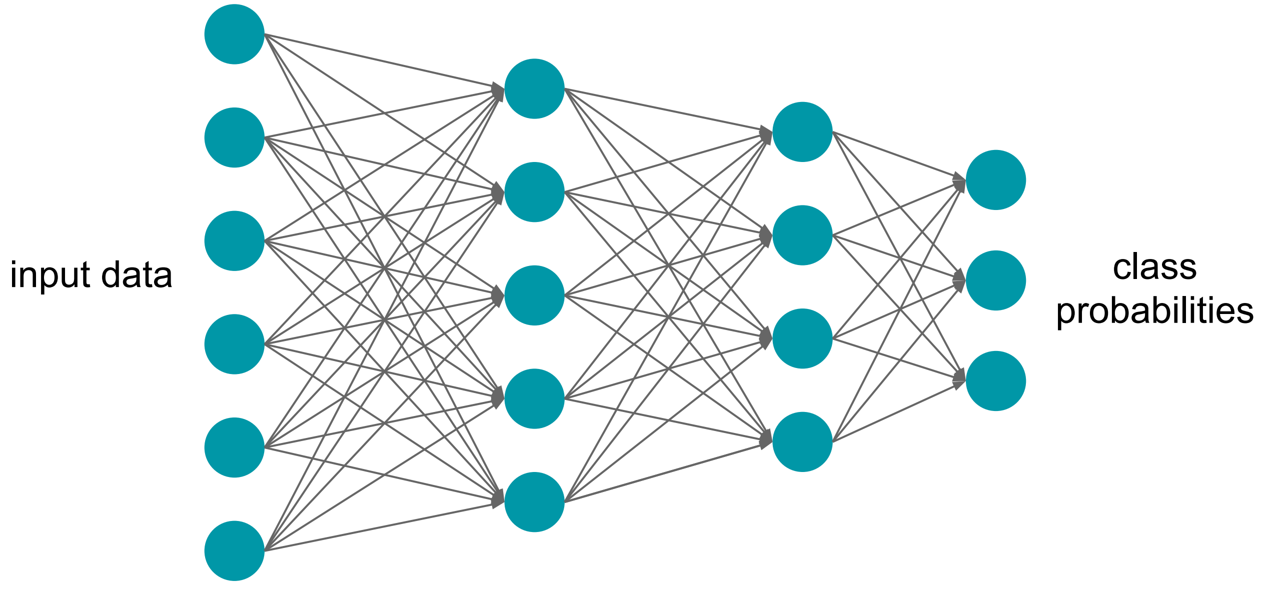


Single nodes in the neural network do something similar


x_1
x_i
x_n
$\sigma\ (\ \Sigma\ w_i x_i + b)$
y
Bias (intercept)
Non-linear activation function
Weighted sum of inputs x

Neural Networks

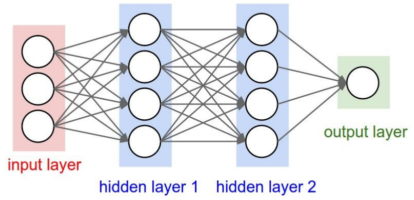
- Nodes in neural networks are organized into layers
- Number of nodes in the layer and number of layers are hyperparameters (not optimized during training, but instead set manually)
- Hierarchical structure makes them very powerful



Fully connected neural network

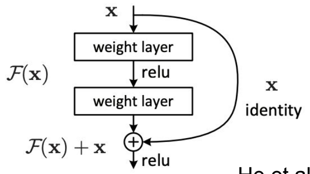
Types of neural networks

- ❑ **Fully connected networks**:
  can approximate almost any function



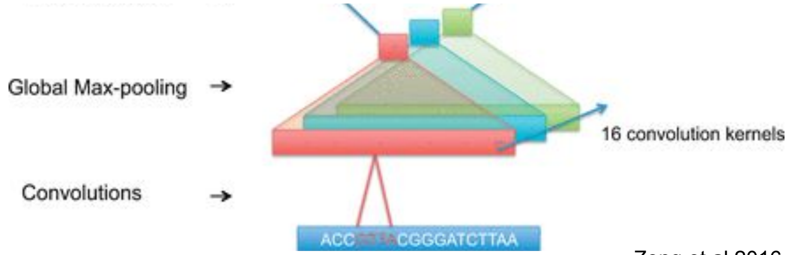
https://cs231n.github.io/neural-networks-1/

- ❑ **Residual networks**:
  can learn both simple (e.g. identity) and more complex functions


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
$\mathbf{x}$
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

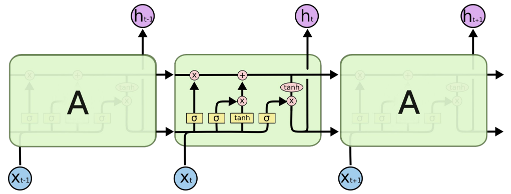
He et al 2015

- ❑ **Convolutional neural networks**:
  detect position-invariant local patterns



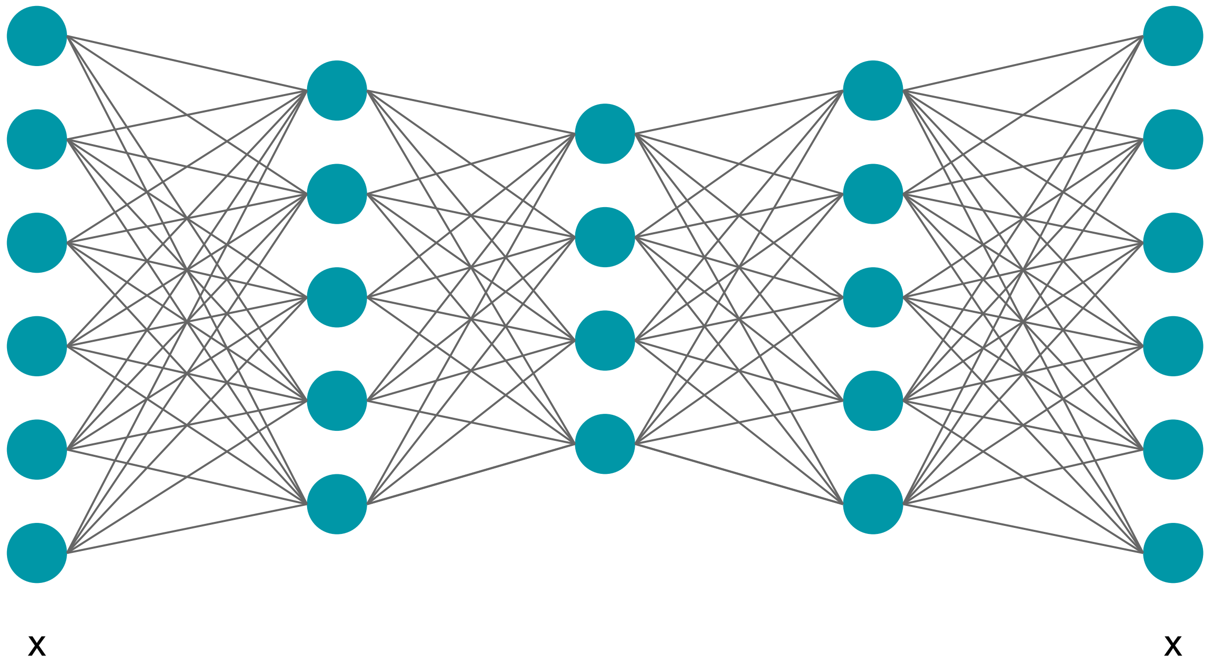
Zeng et al 2016

- ❑ **Recurrent neural networks**:
  can be Turing-complete, often used for long(er)-term dependencies in e.g., sequence data



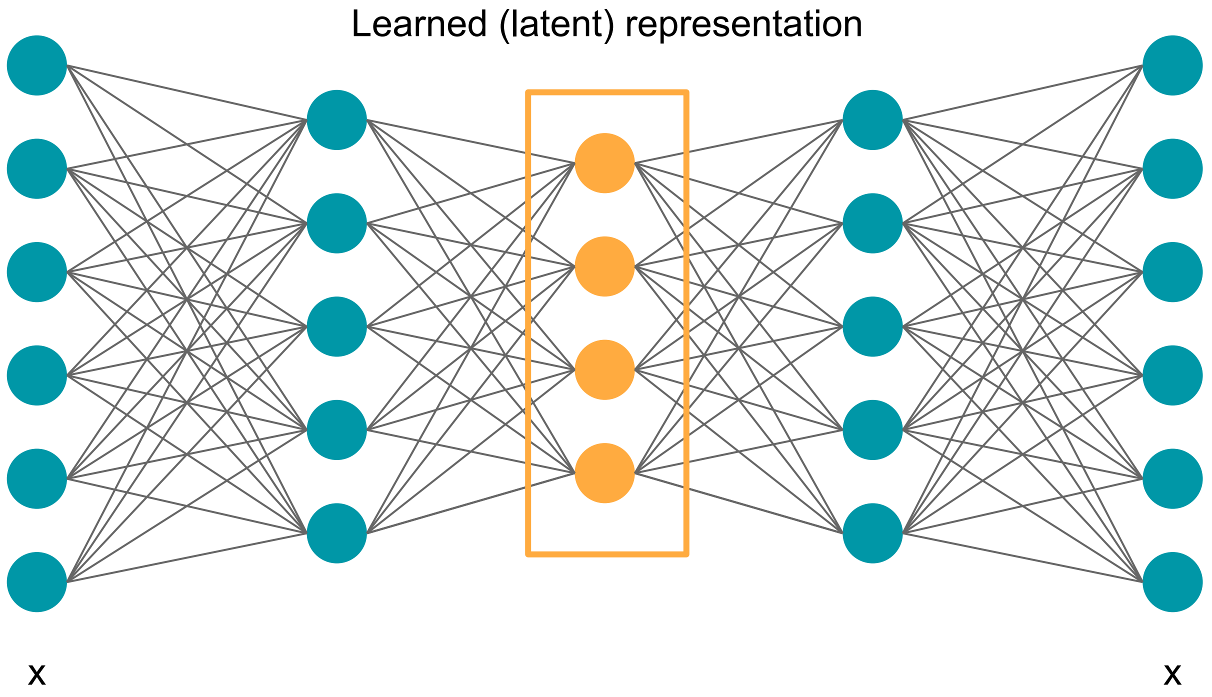
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Unsupervised algorithm - autoencoder example



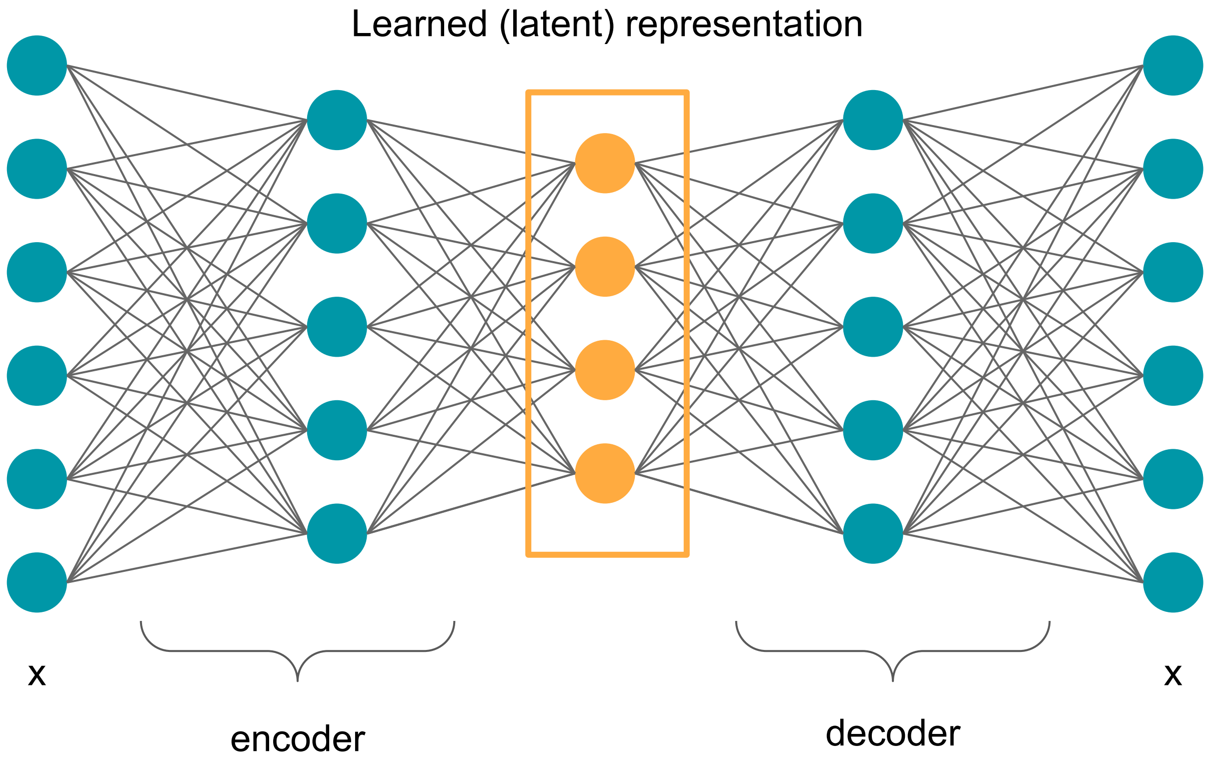
- Autoencoder is a neural network trained to attempt to copy its input to its output

Unsupervised algorithm - autoencoder example



- Autoencoder is a neural network trained to attempt to copy its input to its output while passing through a latent representation

Unsupervised algorithm - autoencoder example



- ❑ Autoencoder is a neural network trained to attempt to copy its input to its output while passing through a latent representation
- ❑ Learned representation can have useful properties: reduced dimensionality, easy to visualize, but there are other tasks as well

References

Dietterich TG. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput*. 1998;10(7):1895–1923. doi:10.1162/089976698300017197

Goodfellow IJ, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016. https://mitpress.mit.edu/books/deep-learning (especially chapter 5 - Machine Learning Basics)

Killoran, Nathan, Leo J. Lee, Andrew Delong, David Duvenaud, and Brendan J. Frey. ‘Generating and Designing DNA with Deep Generative Models’. *ArXiv:1712.06148 [Cs, q-Bio, Stat]*, 17 December 2017. http://arxiv.org/abs/1712.06148.

Zeng, Haoyang, Matthew D. Edwards, Ge Liu, and David K. Gifford. ‘Convolutional Neural Network Architectures for Predicting DNA–Protein Binding’. *Bioinformatics* 32, no. 12 (15 June 2016): i121–27. https://doi.org/10.1093/bioinformatics/btw255.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ‘Deep Residual Learning for Image Recognition’. *ArXiv:1512.03385 [Cs]*, 10 December 2015. http://arxiv.org/abs/1512.03385.

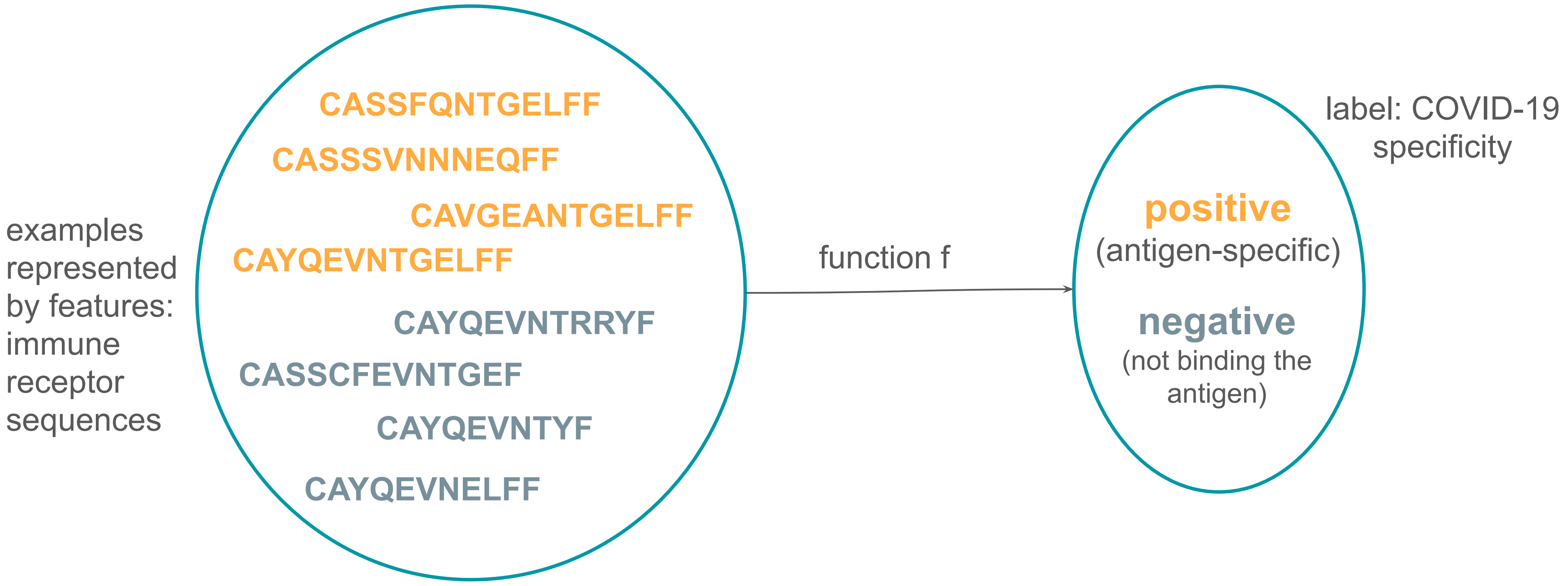
UiO : Universitetet i Oslo

Part 4: Data Representation

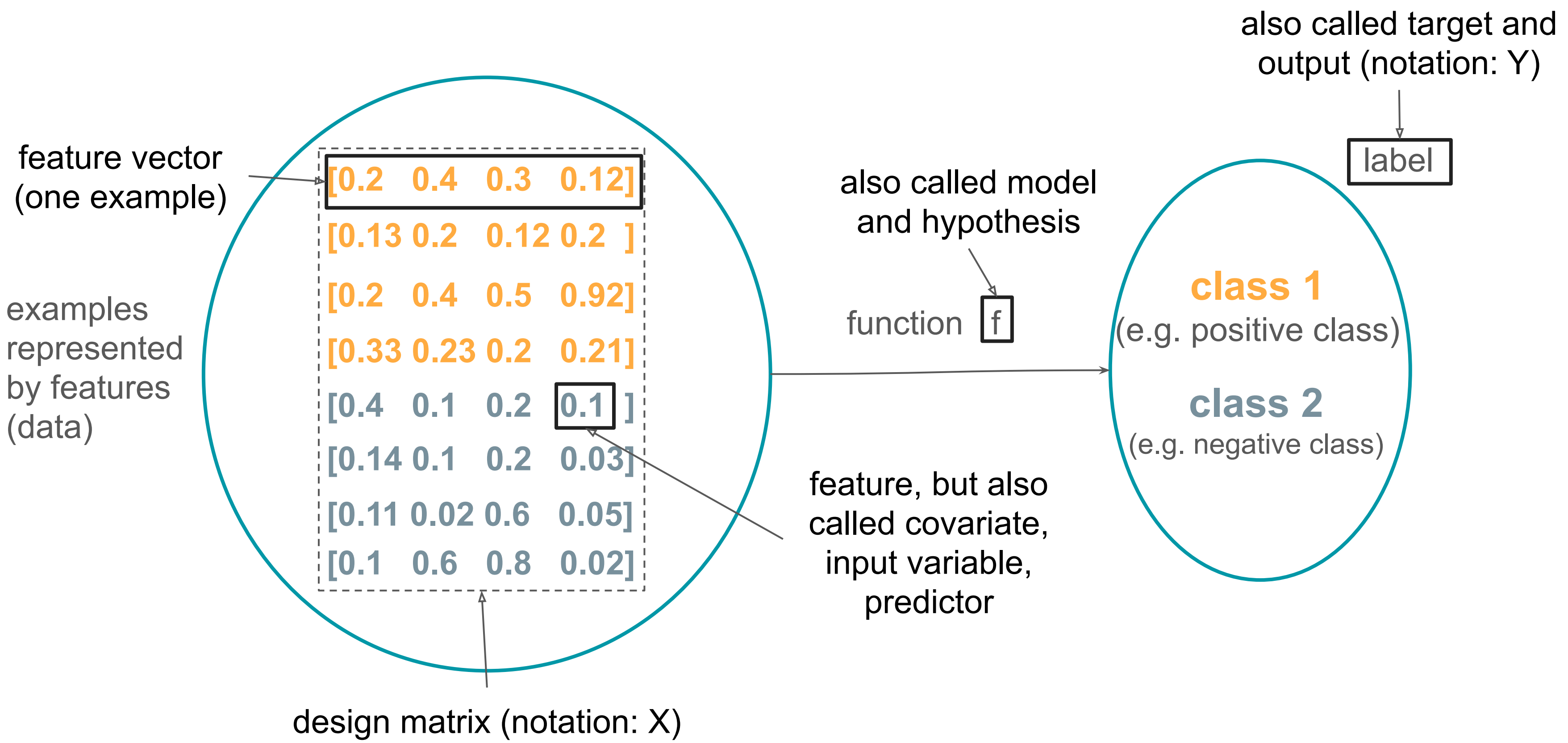
Machine learning in computational biology - outline

- Introduction to machine learning:
  - What is machine learning, types of problems, assumptions, workflow, generalization
- Machine learning models and algorithms:
  - Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)
- **Data representation:**
  - **Considerations and examples, one-hot encoding, feature engineering, representation learning**
- Model comparison and uncertainty:
  - Model assessment, model selection, uncertainty, cross-validation
- Transparency and reproducibility

We are given a set of sequences… but algorithms only understand numbers!



We can represent sequences by their physicochemical properties, for instance


also called target and output (notation: Y)
label
feature vector (one example)
[0.2 0.4 0.3 0.12]
[0.13 0.2 0.12 0.2]
[0.2 0.4 0.5 0.92]
[0.33 0.23 0.2 0.21]
[0.4 0.1 0.2 0.1]
[0.14 0.1 0.2 0.03]
[0.11 0.02 0.6 0.05]
[0.1 0.6 0.8 0.02]
examples represented by features (data)
also called model and hypothesis
function f
class 1
(e.g. positive class)
class 2
(e.g. negative class)
feature, but also called covariate, input variable, predictor
design matrix (notation: X)

Some examples of data representation (encoding)

- ❑ One-hot encoding
- ❑ K-mer frequencies

Data representation heavily depends on data, so in different domains, there will be different representations:

When classifying images with classical approaches: number of edges, objects

When predicting the length of the trip: number of traffic lights, time of day

When predicting if an email is a spam or not: certain words, presence/absence of personal name

Some examples of data representation (encoding)

- One-hot encoding
- K-mer frequencies

We have to be careful how we choose features - we must not introduce information that should not be there!

Data representation heavily depends on data, so in different domains, there will be different representations:

When classifying images with classical approaches: number of edges, objects

When predicting the length of the trip: number of traffic lights, time of day

When predicting if an email is a spam or not: certain words, presence/absence of personal name

One-hot encoding

- A common way to represent categorical data where only one value can be chosen: rows represent the possible values
- Also called *dummy variables* in statistics

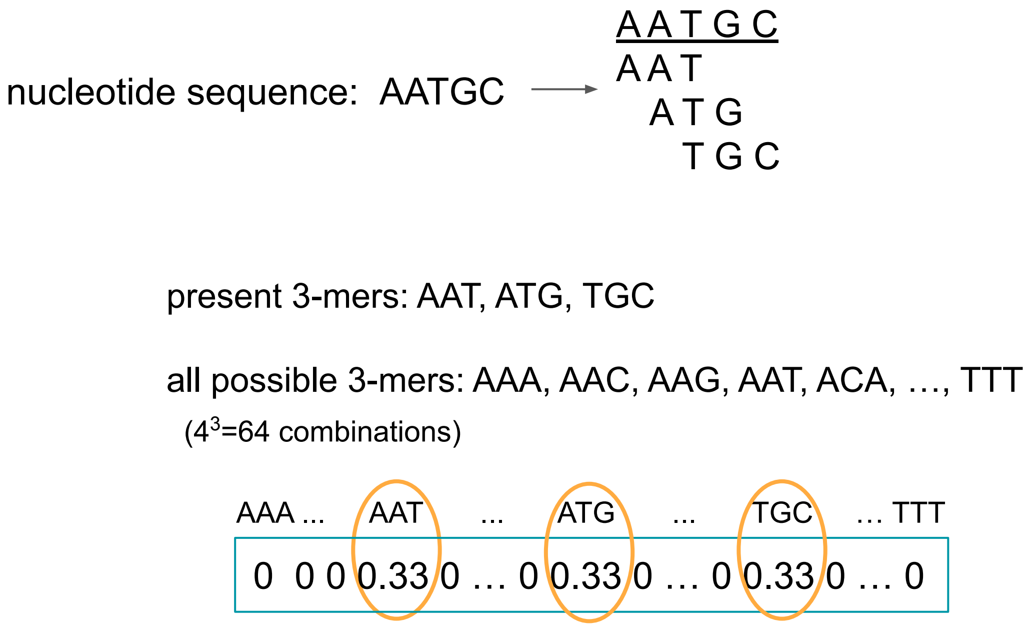
nucleotide sequence: AATGC

$$
\begin{array}{l|ccccc|}
 & A & A & T & G & C \\
\hline
\text{is it A} & 1 & 1 & 0 & 0 & 0 \\
\text{is it C} & 0 & 0 & 0 & 0 & 1 \\
\text{is it G} & 0 & 0 & 0 & 1 & 0 \\
\text{is it T} & 0 & 0 & 1 & 0 & 0 \\
\hline
\end{array}
$$

one-hot encoding

K-mer frequency

- Often used for sequence representation
- k-mers are (optionally overlapping) subsequences of length k


nucleotide sequence: AATGC → A A T G C

```
A A T G C
A A T
  A T G
    T G C
```

present 3-mers: AAT, ATG, TGC

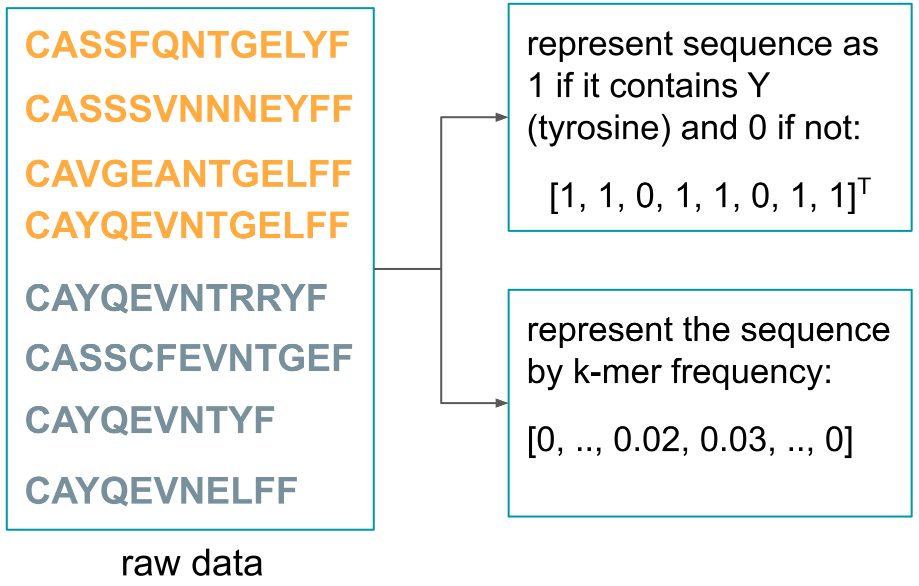
all possible 3-mers: AAA, AAC, AAG, AAT, ACA, …, TTT
(4^3=64 combinations)

AAA ...			AAT	...		ATG	...		TGC	... TTT
0	0	0	0.33	0 … 0		0.33	0 … 0		0.33	0 … 0

k-mer frequency encoding (k=3)

ML algorithm performance heavily depends on data representation

- Data representation refers to choosing and constructing features
- We don’t always know it advance which features are the best for the problem: we have to know the domain:



CASSFQNTGELYF
CASSSVNNNEYFF
CAVGEANTGELFF
CAYQEVNTGELFF
CAYQEVNTRRYF
CASSCFEVNTGEF
CAYQEVNTYF
CAYQEVNELFF

raw data

represent sequence as 1 if it contains Y (tyrosine) and 0 if not:

$[1, 1, 0, 1, 1, 0, 1, 1]^T$

represent the sequence by k-mer frequency:

[0, .., 0.02, 0.03, .., 0]

data representation: option 1

Which one is better?

data representation: option 2

Feature engineering & feature selection

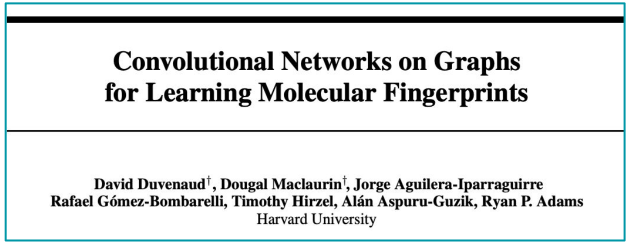
- Feature engineering: together with domain experts, ML researchers would discuss and derive features which they believe could be useful for the model

  Example: for biological sequences, there are a few popular alternatives like k-mer frequencies and physicochemical properties

- This way a lot of features could be constructed and the best ones would be selected as a part of fitting the model (feature selection)

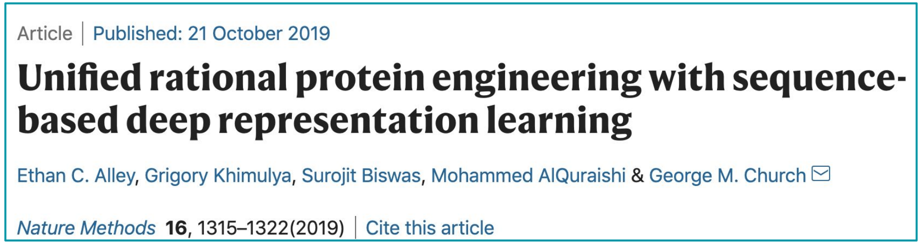
Representation learning

- Most often in context of neural networks: the many layers of the network learn a hierarchical, alternative representation of the (raw) data that was provided as input



Convolutional Networks on Graphs for Learning Molecular Fingerprints

David Duvenaud†, Dougal Maclaurin†, Jorge Aguilera-Iparraguirre
Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams
Harvard University



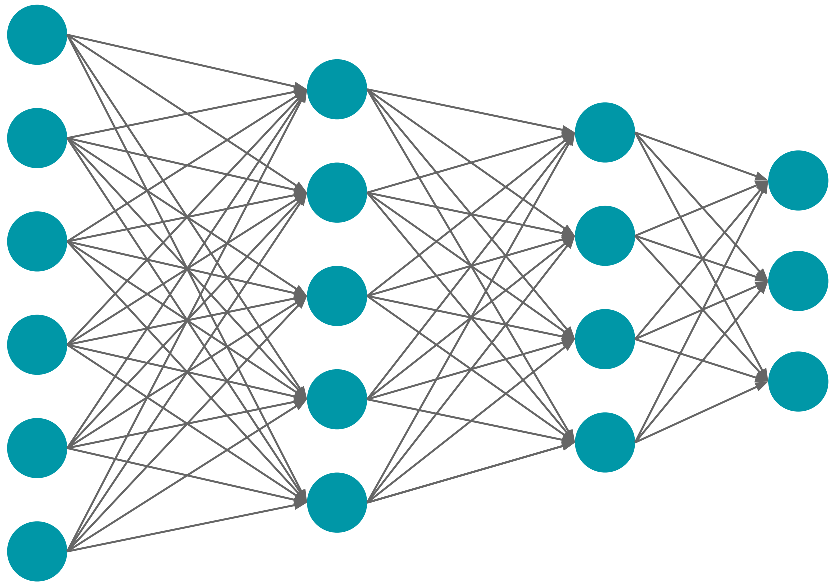
Article | Published: 21 October 2019

Unified rational protein engineering with sequence-based deep representation learning

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi & George M. Church

Nature Methods **16**, 1315–1322(2019) | Cite this article

Representation learning - hidden layers in neural networks can be seen as different representations

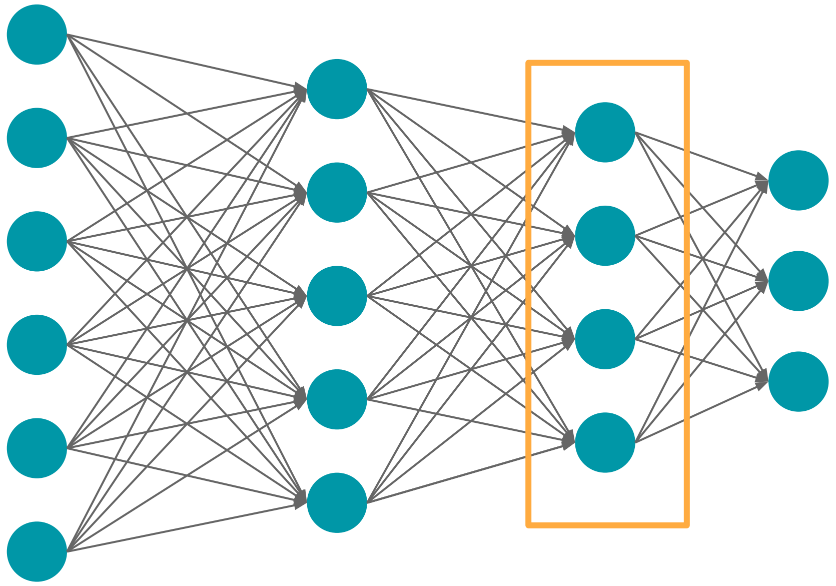


Deep neural network

“A good representation is the one that makes the learning task easier.”

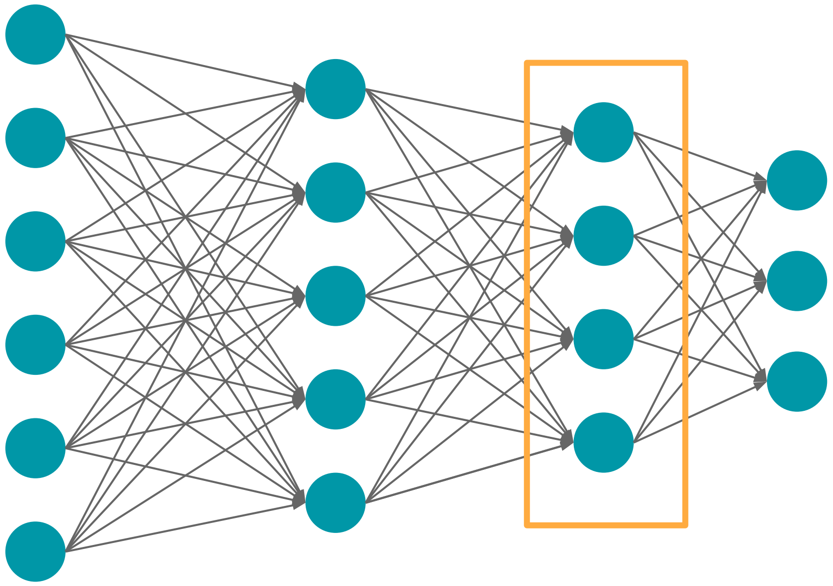
Goodfellow et al. 2016

Representation learning - hidden layers in neural networks can be seen as different representations

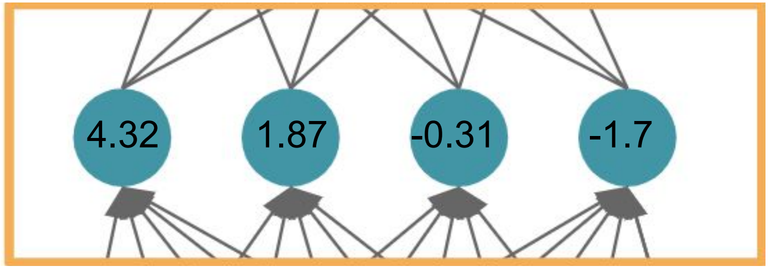


Deep neural network

Representation learning - hidden layers in neural networks can be seen as different representations

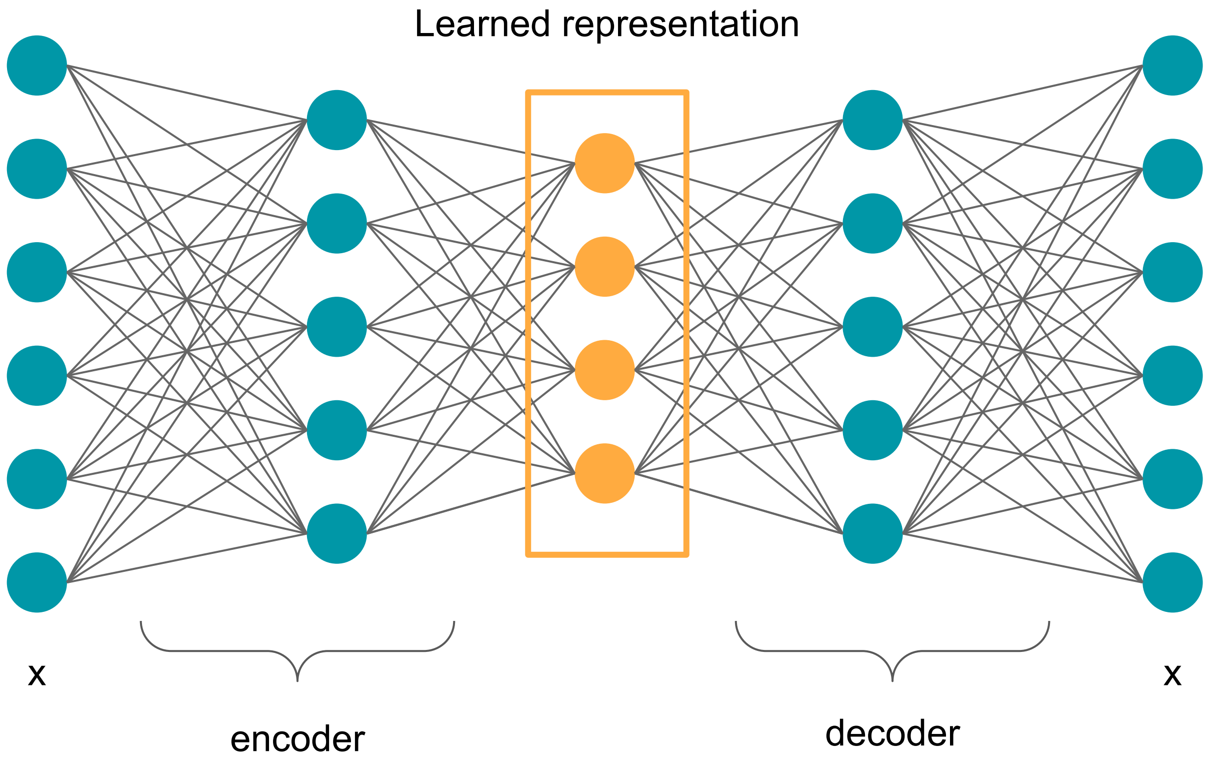


Deep neural network



New data representation

Representation learning with autoencoders



- ❑ Also used for dimensionality reduction and visualization
- ❑ Different types of autoencoders: denoising, sparse

Part 5: ML Model Comparison and Uncertainty

Machine learning in computational biology - outline

- Introduction to machine learning:
  - What is machine learning, types of problems, assumptions, workflow, generalization
- Machine learning models and algorithms:
  - Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)
- Data representation:
  - Considerations and examples, one-hot encoding, feature engineering, representation learning
- **Model comparison and uncertainty:**
  - **Model assessment, model selection, uncertainty, cross-validation**
- Transparency and reproducibility

Model selection and model assessment

In a typical workflow, we perform two tasks:

Model selection: estimating the performance of different models in order to choose the best one

Model assessment: having chosen the final model, estimating its prediction error (generalization error) on new data

Definitions from: Hastie et al. 2009

Model selection and model assessment

In a typical workflow, we perform two tasks:

Model selection: estimating the performance of different models in order to choose the best one

different learning algorithms or same learning algorithms, but different hyperparameters (e.g., different number of layers in a neural network)

Model assessment: having chosen the final model, estimating its prediction error (generalization error) on new data

Definitions from: Hastie et al. 2009

Evaluation of an ML algorithm

- To do model selection (and model assessment) we need to know how to do model evaluation
- A suitable cost function has to be chosen

Evaluation of a ML algorithm

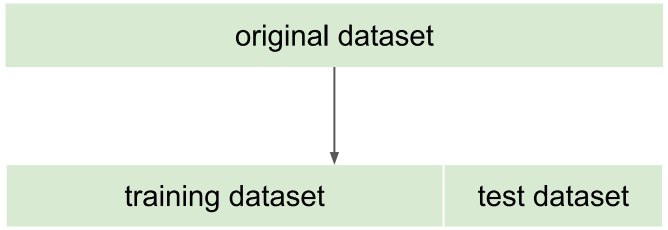
- ❏ Two ML algorithms are different if and only if their percentage of correct classifications would be different on average when trained on a training set of a given fixed size and tested on all data points in the population

Dietterich 1998

Of course we cannot test on the entire population, but we should be aware that the difference we see in one particular result might not be the true difference between algorithms

Evaluation of an ML algorithm: random holdout test set

- The simplest scenario: split the dataset to train/test dataset

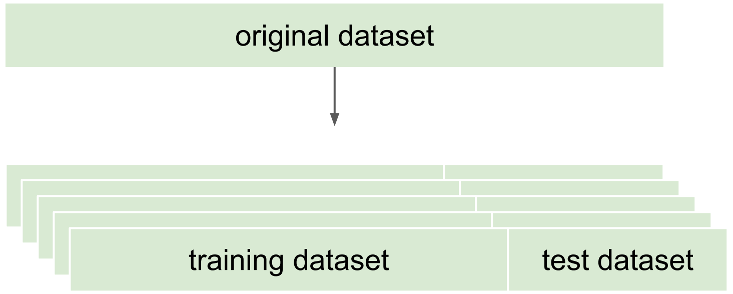


Test dataset is typically 25-30% of the original dataset, but in different applications, these numbers vary from 10 to 50%

- Estimate performance of the algorithm on the test set

Evaluation of a ML algorithm: random holdout test sets

- A bit more robust scenario: split the dataset randomly into multiple train/test datasets

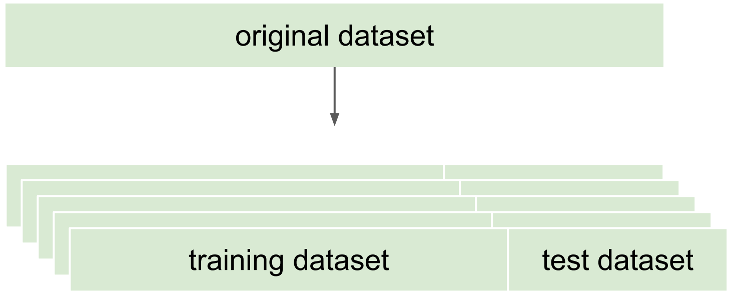


Test dataset is typically 25-30% of the original dataset.

- Estimate performance of the algorithm as the average of the performances on test sets

Evaluation of a ML algorithm: random holdout test sets

- A bit more robust scenario: split the dataset randomly into multiple train/test datasets

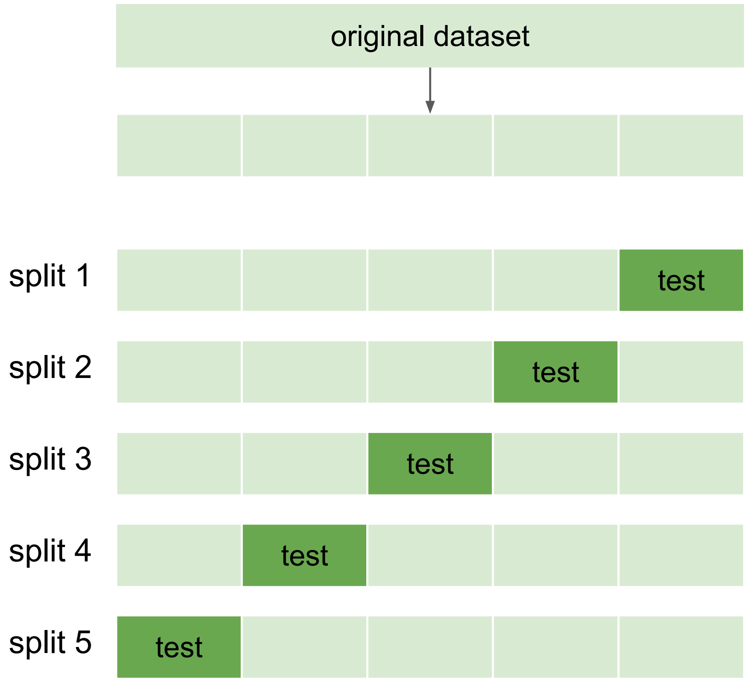


Test dataset is typically 25-30% of the original dataset.

- Estimate performance of the algorithm as the average of the performances on test sets
- The same examples could be in multiple test sets: biased estimate of the performance!

K-fold cross validation

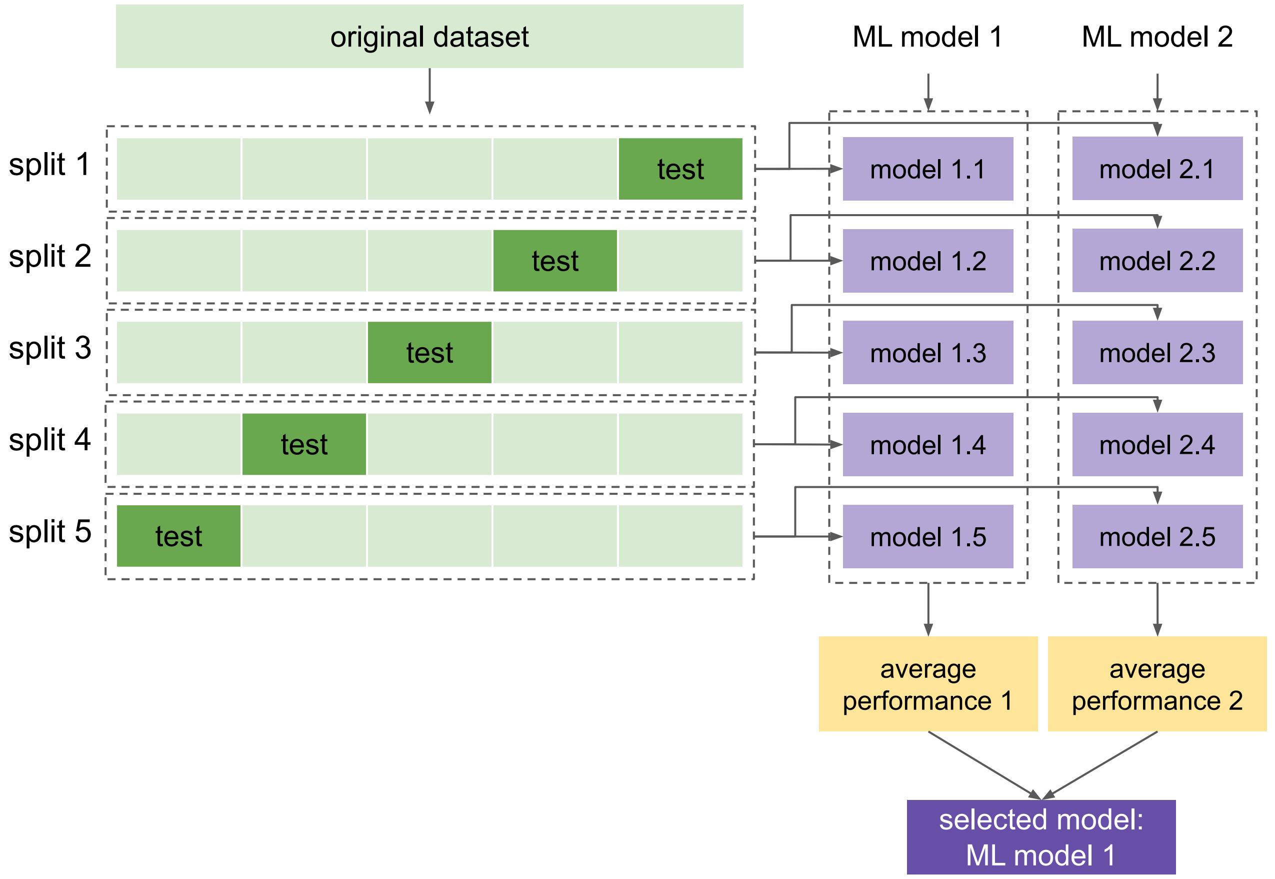
- K-fold cross validation (CV) refers to splitting the data into k training and test set pairs so that each example is once a part of the test set



- The model is trained and evaluated on each train/test set pair
- The estimated performance is the obtained as average over all test set performances
- Typical values of k: 5, 10
- Leave-one-out CV

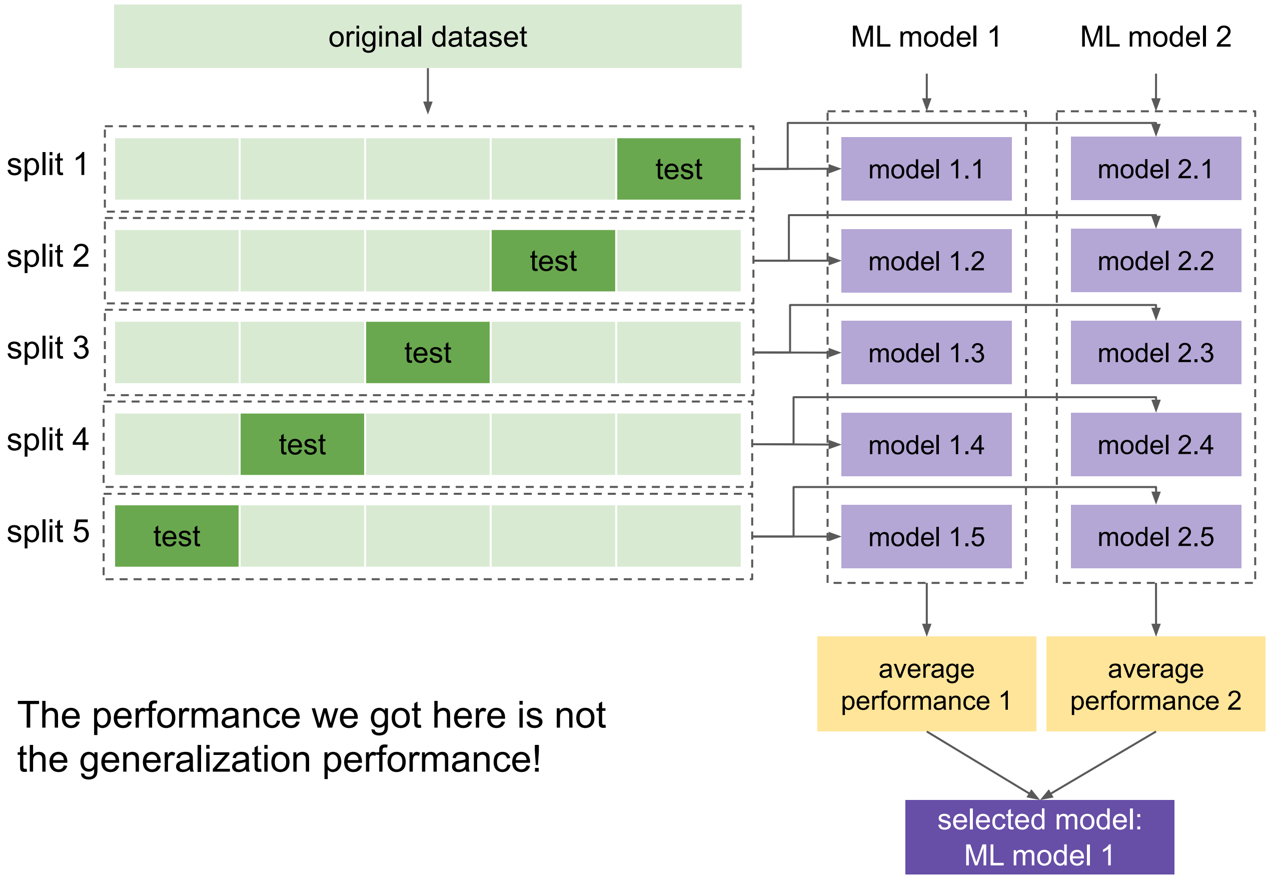
How to do model selection

1. Split the original dataset to train/test dataset per one of the previous strategies (e.g., 5-fold CV)
2. For each model compute the performance as the average of performances on test sets (as described in the previous slide)
3. Select the model with best average performance



How to do model selection

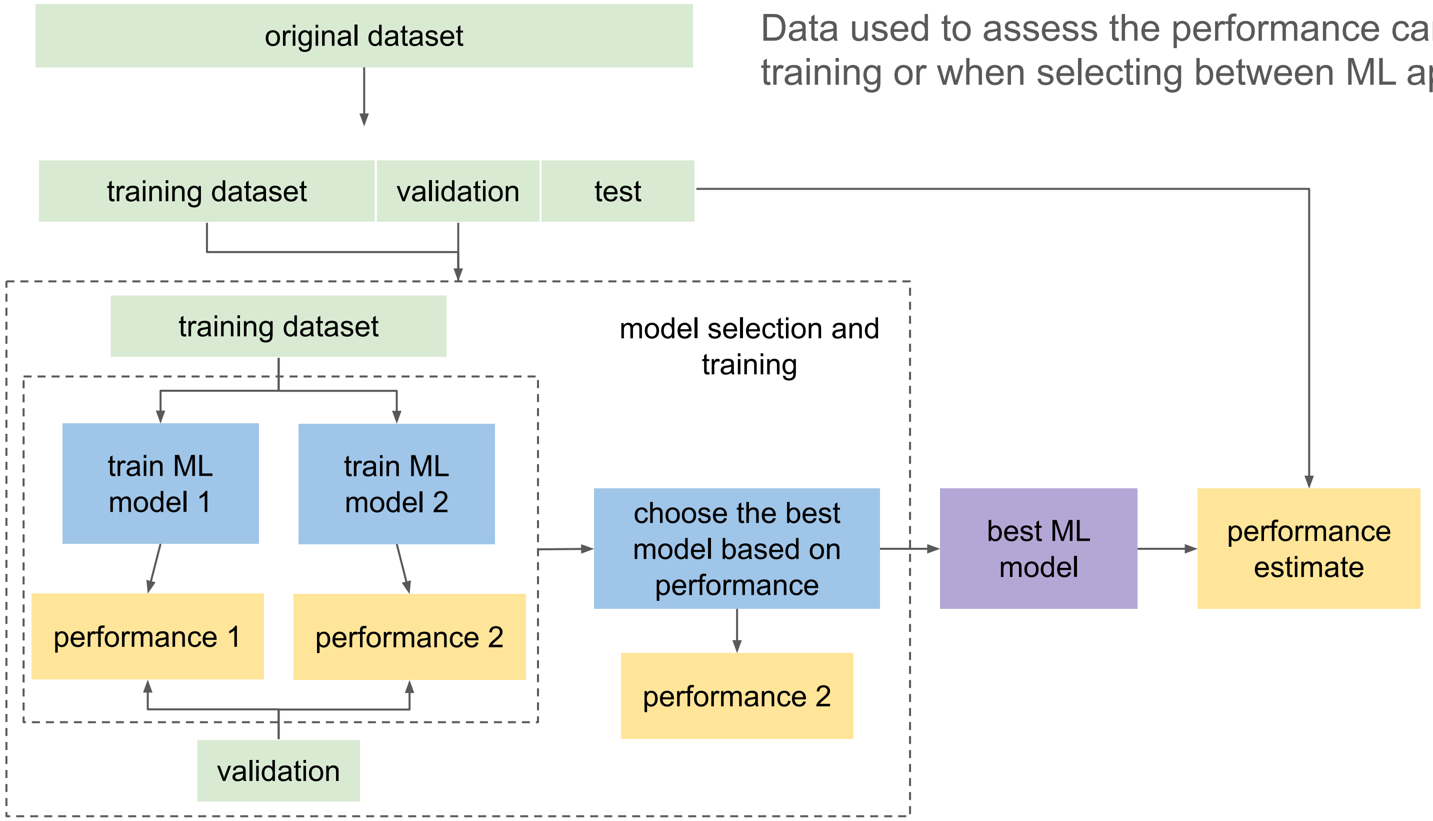
1. Split the original dataset to train/test dataset per one of the previous strategies (e.g., 5-fold CV)
2. For each model compute the performance as the average of performances on test sets (as described in the previous slide)
3. Select the model with best average performance



The performance we got here is not the generalization performance!

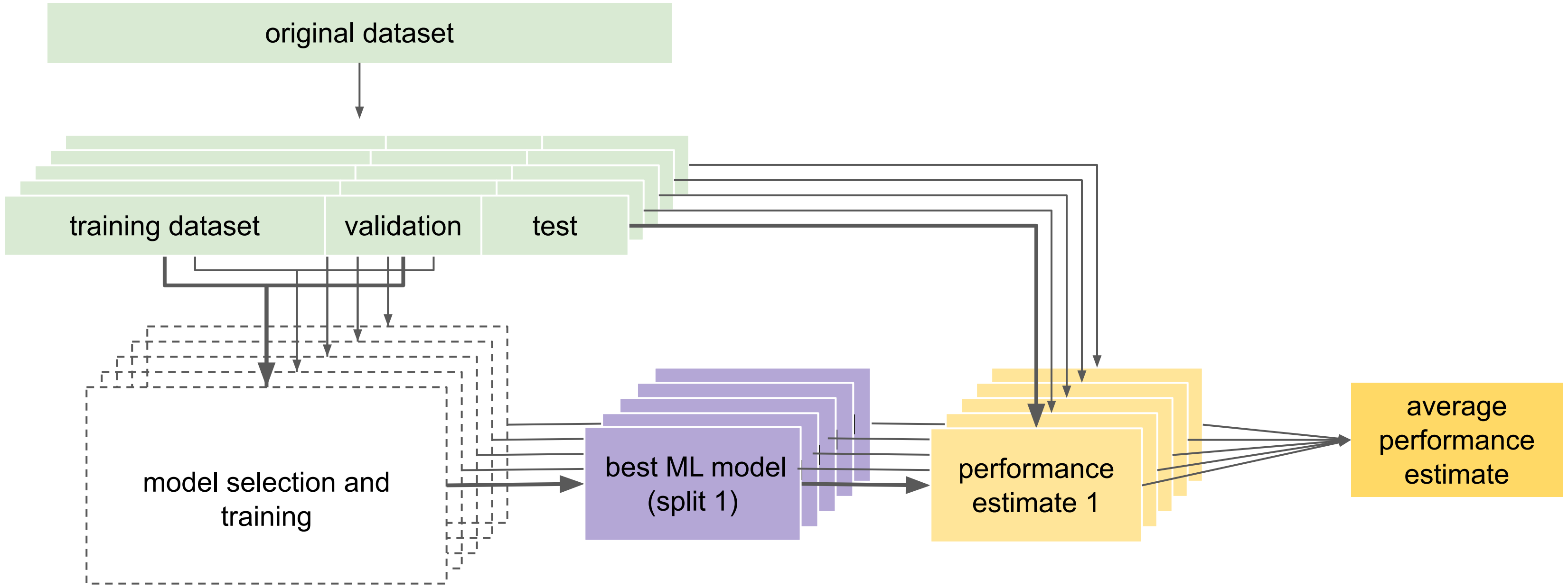
Model assessment uses a separate test set not used during model selection

Data used to assess the performance cannot be used during training or when selecting between ML approaches



Model assessment with multiple test sets: a more robust estimate

- We can employ all the same techniques (e.g., CV) to split the data, do model selection as described before and use separate test sets to estimate generalization performance



Recommendations for ML in biology

DOME: recommendations for supervised machine learning validation in biology

Ian Walsh, Dmytro Fishman, Dario Garcia-Gasulla, Tiina Titma, Gianluca Pollastri, ELIXIR Machine Learning Focus Group, Jennifer Harrow, Fotis E. Psomopoulos & Silvio C. E. Tosatto

Nature Methods **18**, 1122–1127 (2021) | Cite this article

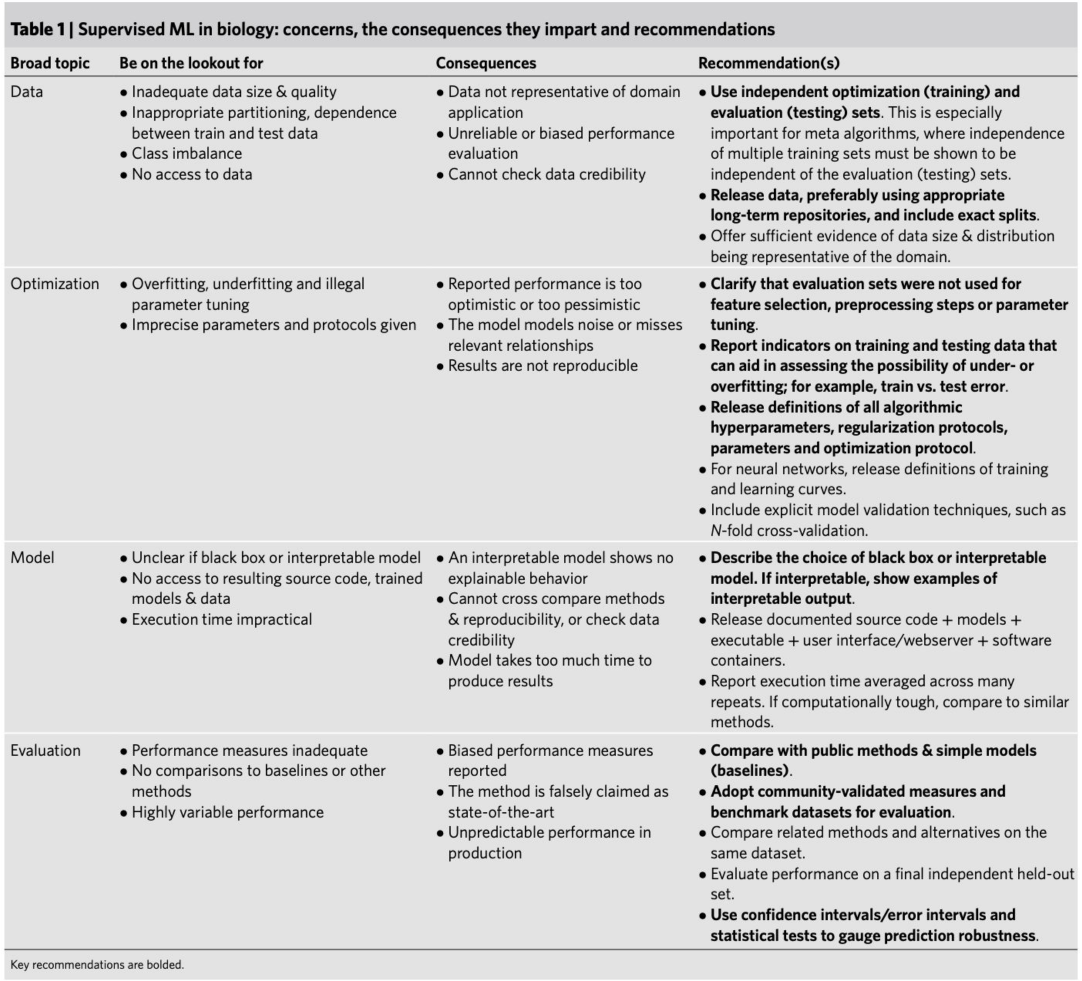


Table 1 | Supervised ML in biology: concerns, the consequences they impart and recommendations

Broad topic	Be on the lookout for	Consequences	Recommendation(s)
Data	• Inadequate data size & quality • Inappropriate partitioning, dependence between train and test data • Class imbalance • No access to data	• Data not representative of domain application • Unreliable or biased performance evaluation • Cannot check data credibility	• **Use independent optimization (training) and evaluation (testing) sets**. This is especially important for meta algorithms, where independence of multiple training sets must be shown to be independent of the evaluation (testing) sets. • **Release data, preferably using appropriate long-term repositories, and include exact splits**. • Offer sufficient evidence of data size & distribution being representative of the domain.
Optimization	• Overfitting, underfitting and illegal parameter tuning • Imprecise parameters and protocols given	• Reported performance is too optimistic or too pessimistic • The model models noise or misses relevant relationships • Results are not reproducible	• **Clarify that evaluation sets were not used for feature selection, preprocessing steps or parameter tuning**. • **Report indicators on training and testing data that can aid in assessing the possibility of under- or overfitting; for example, train vs. test error**. • **Release definitions of all algorithmic hyperparameters, regularization protocols, parameters and optimization protocol**. • For neural networks, release definitions of training and learning curves. • Include explicit model validation techniques, such as *N*-fold cross-validation.
Model	• Unclear if black box or interpretable model • No access to resulting source code, trained models & data • Execution time impractical	• An interpretable model shows no explainable behavior • Cannot cross compare methods & reproducibility, or check data credibility • Model takes too much time to produce results	• **Describe the choice of black box or interpretable model. If interpretable, show examples of interpretable output**. • Release documented source code + models + executable + user interface/webserver + software containers. • Report execution time averaged across many repeats. If computationally tough, compare to similar methods.
Evaluation	• Performance measures inadequate • No comparisons to baselines or other methods • Highly variable performance	• Biased performance measures reported • The method is falsely claimed as state-of-the-art • Unpredictable performance in production	• **Compare with public methods & simple models (baselines)**. • **Adopt community-validated measures and benchmark datasets for evaluation**. • Compare related methods and alternatives on the same dataset. • Evaluate performance on a final independent held-out set. • **Use confidence intervals/error intervals and statistical tests to gauge prediction robustness**.

Key recommendations are bolded.

References

Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. Springer-Verlag; 2009. doi:10.1007/978-0-387-84858-7

Dietterich TG. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput*. 1998;10(7):1895–1923. doi:10.1162/089976698300017197

Goodfellow IJ, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016. https://mitpress.mit.edu/books/deep-learning

Walsh, I., Fishman, D., Garcia-Gasulla, D. *et al.* DOME: recommendations for supervised machine learning validation in biology. *Nat Methods* 18, 1122–1127 (2021). https://doi.org/10.1038/s41592-021-01205-4

UiO : Universitetet i Oslo

Part 6: Transparency and Reproducibility

Machine learning in computational biology - outline

- Introduction to machine learning:
  - What is machine learning, types of problems, assumptions, workflow, generalization
- Machine learning models and algorithms:
  - Discriminative vs generative models, supervised models (logistic and linear regression, kNN, neural networks), unsupervised models (dimensionality reduction, clustering)
- Data representation:
  - Considerations and examples, one-hot encoding, feature engineering, representation learning
- Model comparison and uncertainty:
  - Model assessment, model selection, uncertainty, cross-validation
- **Transparency and reproducibility**

Transparency & interpretability are crucial for ML

- "**Transparency** refers to how easily a stakeholder can examine the model and understand, or explain, how the model operates when combining the inputs to produce output, regardless how accurate the model is".

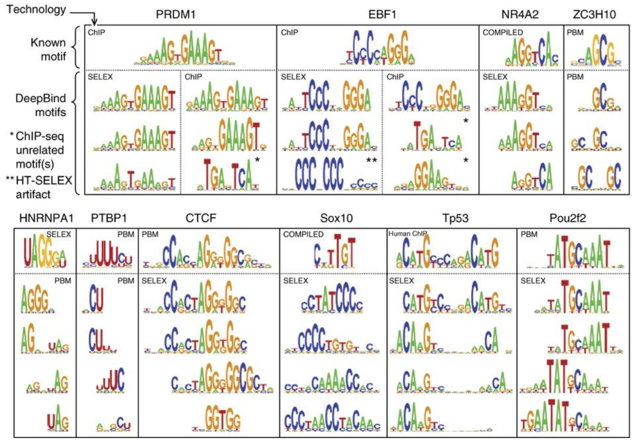
Wainberg et al. 2018

- Transparency would enable experts to check the model prediction and use it or check it more
- **Interpretability**: "understanding the patterns in the data may be just as important as fitting the data"

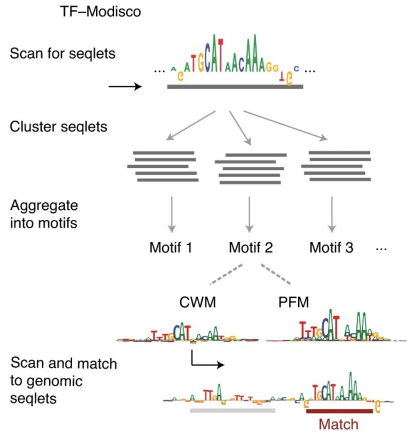
Ching et al. 2018

Interpretability of neural networks

- Some techniques:
  - Visualizing filters in (the first layers of) CNNs
  - Clustering high contribution regions to derive motifs
  - Backpropagating w.r.t. input data (saliency maps)
  - Visualizing class activation maps

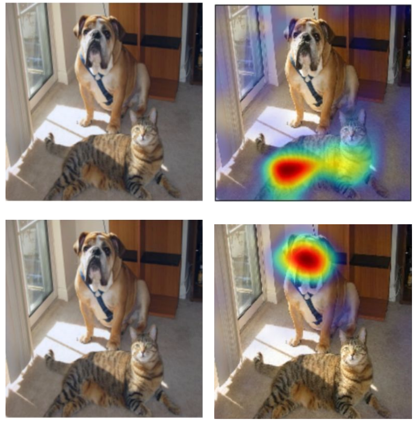


Alipanahi et al. 2015



Avsec et al. 2021

Simonyan et al. 2014



Selvaraju et al. 2017

Reproducibility in ML

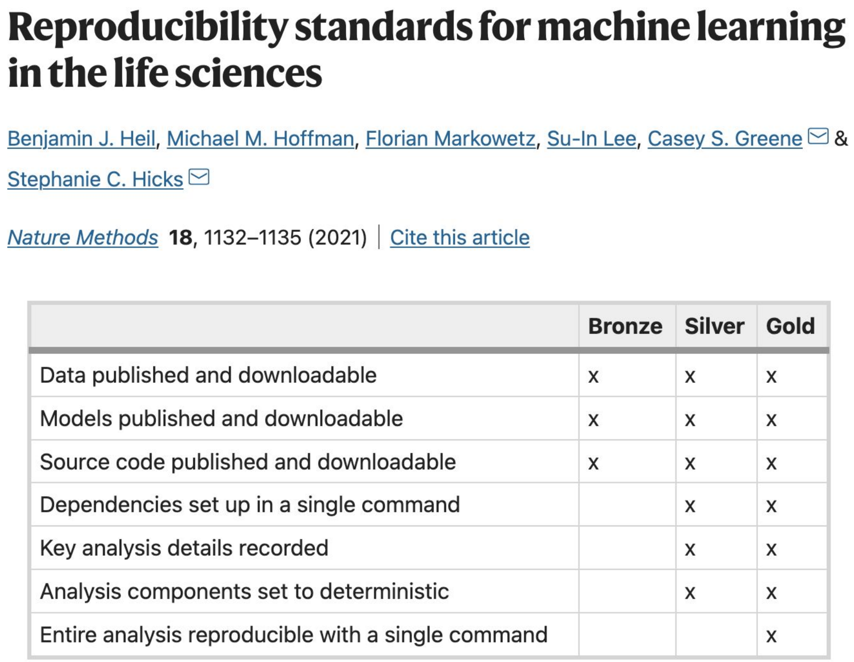
Reproducibility: “the provision of enough detail about study procedures and data so the same procedures could be exactly repeated”

Goodman et al. 2016

Achieving reproducibility:

- ❑ Keep track how each result is produced
- ❑ Record all intermediate results [in standardized formats]
- ❑ Store raw data behind all plots
- ❑ Provide public access to scripts, runs and results
- ❑ Archive the exact versions of all external programs used
- ❑ Use version control

Sandve et al. 2013



Reproducibility standards for machine learning in the life sciences

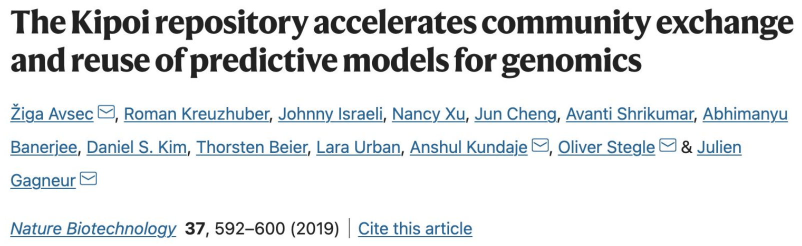
Benjamin J. Heil, Michael M. Hoffman, Florian Markowetz, Su-In Lee, Casey S. Greene & Stephanie C. Hicks

Nature Methods **18**, 1132–1135 (2021) | Cite this article

	Bronze	Silver	Gold
Data published and downloadable	x	x	x
Models published and downloadable	x	x	x
Source code published and downloadable	x	x	x
Dependencies set up in a single command		x	x
Key analysis details recorded		x	x
Analysis components set to deterministic		x	x
Entire analysis reproducible with a single command			x

Reproducibility in ML

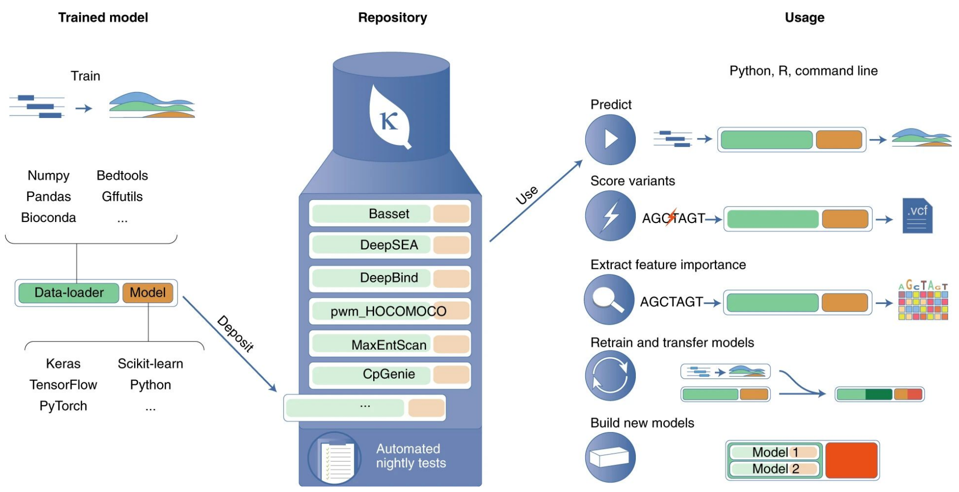
- The code and models should be publicly available (GitHub, Zenodo, or other services)
- Possible to inspect, validate, (re)use, and improve models



The Kipoi repository accelerates community exchange and reuse of predictive models for genomics

Žiga Avsec, Roman Kreuzhuber, Johnny Israeli, Nancy Xu, Jun Cheng, Avanti Shrikumar, Abhimanyu Banerjee, Daniel S. Kim, Thorsten Beier, Lara Urban, Anshul Kundaje, Oliver Stegle & Julien Gagneur

Nature Biotechnology **37**, 592–600 (2019) | Cite this article



References

Lipton ZC. The Mythos of Model Interpretability. *arXiv:160603490 [cs, stat]*. Published online March 6, 2017. http://arxiv.org/abs/1606.03490

Alipanahi, B., Delong, A., Weirauch, M. *et al.* Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* 33, 831–838 (2015). https://doi.org/10.1038/nbt.3300

Avsec, Ž., Weilert, M., Shrikumar, A. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat Genet* 53, 354–366 (2021). https://doi.org/10.1038/s41588-021-00782-6

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. 'Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps'. *ArXiv:1312.6034 [Cs]*, 19 April 2014. http://arxiv.org/abs/1312.6034.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 'Grad-CAM:Visual Explanations from Deep Networks via Gradient-Based Localization'. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–26, 2017. https://doi.org/10.1109/ICCV.2017.74.

Goodman SN, Fanelli D, Ioannidis JPA. What does research reproducibility mean? *Science Translational Medicine*. 2016;8(341):341ps12-341ps12. doi:10.1126/scitranslmed.aaf5027

Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten Simple Rules for Reproducible Computational Research. *PLOS Computational Biology*. 2013;9(10):e1003285. doi:10.1371/journal.pcbi.1003285

Heil, B.J., Hoffman, M.M., Markowetz, F. *et al.* Reproducibility standards for machine learning in the life sciences. *Nat Methods* 18, 1132–1135 (2021). https://doi.org/10.1038/s41592-021-01256-7

Avsec, Ž., Kreuzhuber, R., Israeli, J. *et al.* The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nat Biotechnol* 37, 592–600 (2019). https://doi.org/10.1038/s41587-019-0140-0