NLP and genomics: the vaccines of the future

Andrea Palladino, *Sr. Data Scientist in GSK*

Disclaimer

- Conflict of interests: *AP* is an employee of the GSK group of companies.
- No GSK data are reported in the presentation.
- This work was sponsored by GlaxoSmithKline Biologicals SA which was involved in all stages of the study conduct and analysis
- *AP* is involved in the PrIMAVeRa, that is part of the AMR accelerator. The AMR Accelerator has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under Grant Agreements No 853967 I 853989 I 853979 I 853932 I 853800 I 853903 I 853976 I 101007873 I 101034420

My background

- Master degree in Physics (2014)
- PhD in Physics (2017)
- Working at DESY (Berlin, Germany) since 2017 to 2020 as Researcher in Astroparticle Physics
- Working as Data Scientist at Apheris AI since 2020 to 2022
- Working as Sr. Data Scientist in GSK since June 2022

GSK

Natural Language Processing (NLP) applied to genomic sequences

GSK

Genome

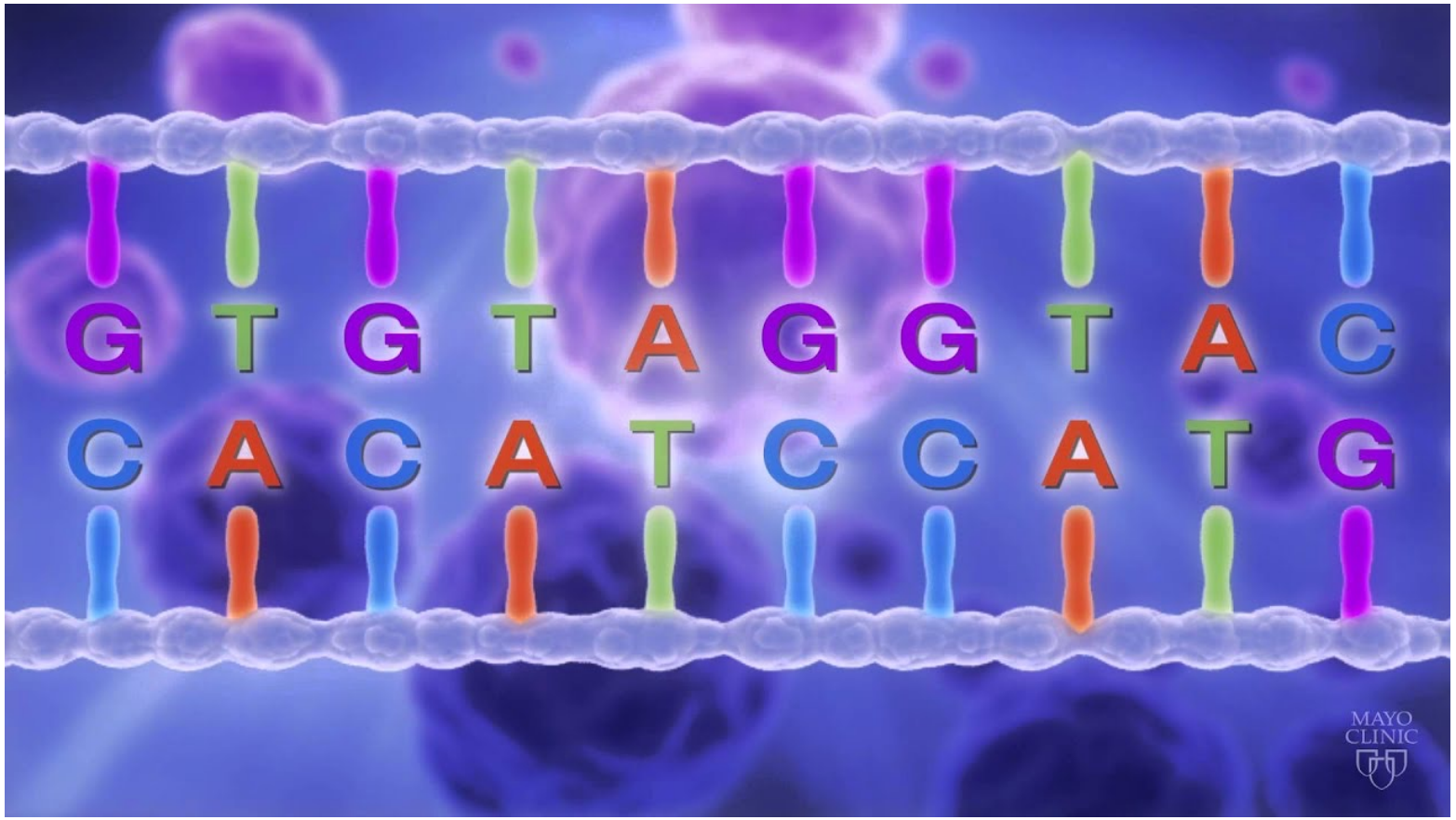
A genome is the complete set of genetic information in an organism.

It consists of nucleotide sequences of DNA.

A DNA sequence is made up of nucleotides that represent the primary structure of a DNA, with the ability to convey information.

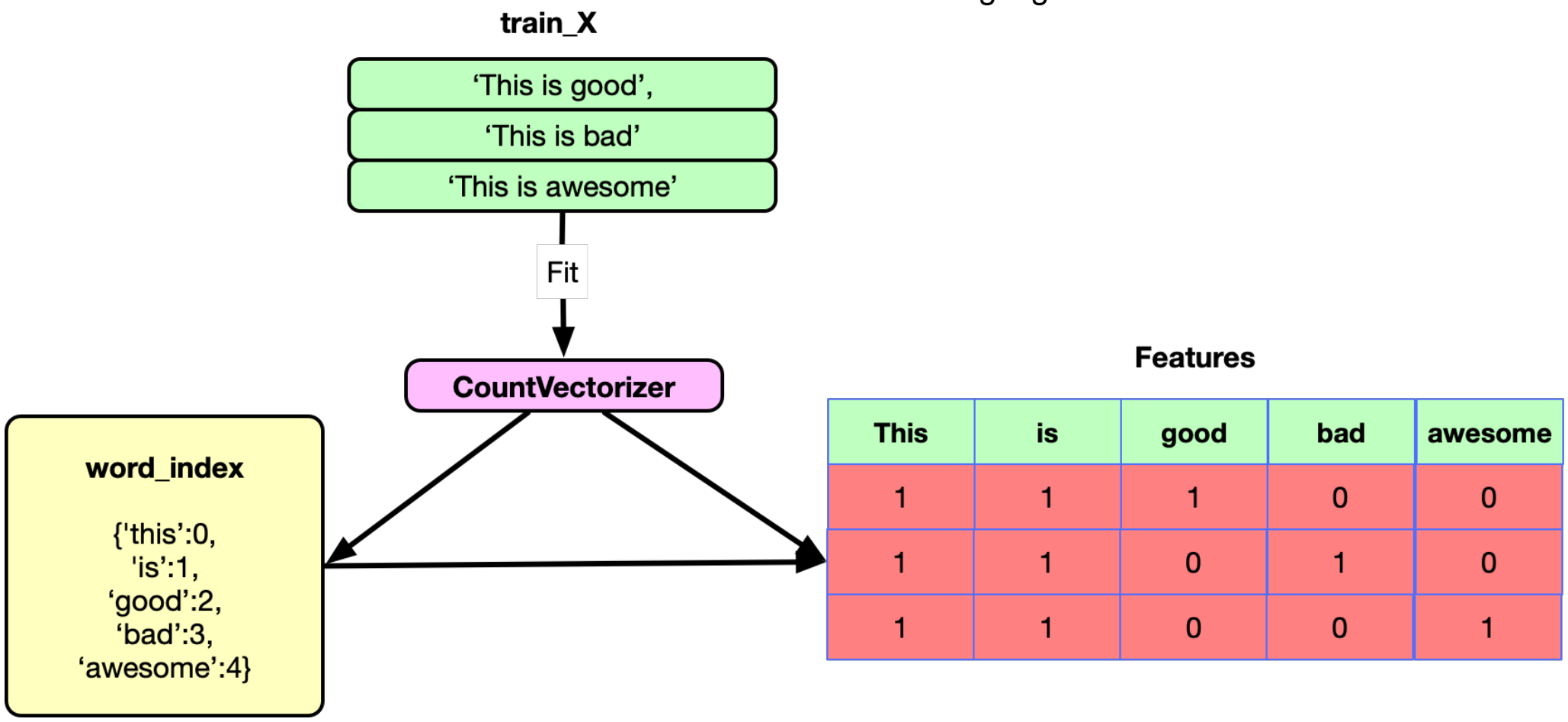
Possible letters in a DNA sequence are A, C, G and T and represent the four nucleotide bases:

- **A**denine
- **C**ytosine
- **G**uanine
- **T**hymina



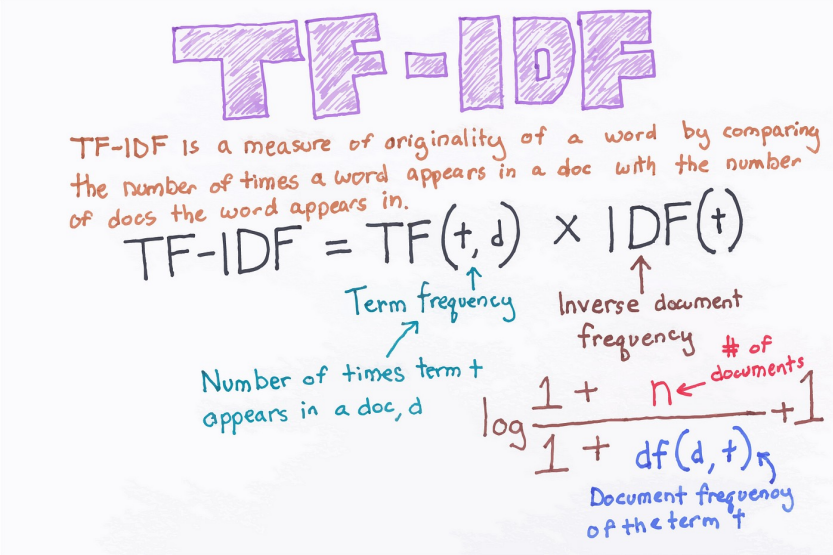
Analysis of text

- A text can be transformed into a numeric matrix
- The numeric matrix is then used as input of the machine learning algorithm



Word-cloud

On the right, a possible graphical representation of the matrix previously obtained, where the size of the word is directly proportional to its importance.



Importance ≠ Frequency

ML applied to genomes (1)

Natural language processing (NLP)

Example (not real DNA sequences)

As seen in the previous slides, we convert sequences into a matrix

```
array(['CGCCGTACGGGACGAACATTCTTGTGAGTACCCAATGTGGAATACGGTCACGACCGGGTCGGCGACGGTCTAGGCAGCGGTGACTACATAAGTCCGAGATCCATTCTTGCCTAGATTCAGATAGTAGCACCACCGAGTCATT
GTAACGAAGATTACTACTGAATTGAATGCAGGGACATGTACTACGGCCTGATTGTCGA',
       'ATGTATCGGAGTGATTCGCCTCCGATTTAGCCCATGGCATTTGCGATCGTCCTCTCACCGCAAGGGTAATGGGTAACGACACCGCGGCGGCAGGGGGGGAGTCGTGTTCGAATGCCGTACGCCTCGCAGGCAACCCGGAGCA
CCACTAGGTAGCCACGGTAACTGCACGCCAGCTCATTGGTTCCATGAGCGAAAGCGCA',
       'AGAAGGTTCATGAGTCTTTACCTTGCGCCGGCGAGAAAAGGTCCATAGGGTAGTAGGACGGGTCCGCTTACGGTGAGCACTTAAACGTTCCAACCGCTCGACAGAGGAGCTAACTACGACTACAACGACAATTGAGCGTTTT
GGGACTGTCTCCCGTGCCACGAGGTGCAGCACGGGGTTTGGATATCTATAGGCAAGAA',
       'CTCCATAAGCCTCGGTGTTGCATGCAAAGAACTCTACTATGTGGGGGACTATTAAGGTGTCGTTTACTCCTGCGACGGTGGTTTAAGGTGTAAAACTCCCTAAAGTTAACGCTCTCGCATCGGCAATCGTCCACGTACTATT
ATCCGCTACACGACATCAAAGGCACAGAGGACCCATGATAGTTCCGCAATTTGGAAGT',
       'AACTATGACCATTGGCAATCAAGATGATGAAAATTCGGGAAAGGTGCTTCTCACACTCGGCGACCGTACTGGTCGGAAGGAAAGCGCTCTTCTACTAAGATTCGATGGCATAAGCCGACCGGGCGTCCCAATGGGTAAGAGG
CCACGTCTAGAATAGACCACAATTGTCTGAACAGACCATGCATGGTCGCTCCCCGTGA',
       'CACGCTAGTTAATATAAAGATCCCTTTATCGAGGAGGTGCGTCACCGCTAAGATAGGAAGTCTCCTCGACCCTCTAGAACACGTCGAAAAAAAAATGGTAGGTTGAGATTCTACCGAGAATCAGACTGCACACGAGTGGGGTC
TACAACTAAAAAAAGACCGGTTACGAACTGGCTCCCCCAAGTGGTCTAGCACGATGAC',
       'GAGGGTTGCGGACGGGGCTGAGCAAGCGCTCTGCGCATGAAGCGCCAAGGCAATGAATACGGTCCCGTTTTGGTCTCCCAGGATGGTAGTCGACAGTTATTGCGAACCACTCCAGCAGGCCCGAACAGCGCCCCCCCTTGGC
GTGTGCCAGCAGATCCTTCATGACAATGGTATTTTTGCGCAACCCATTAGTGACCACG',
       'CGACAGACATTCCCTTTCAGCACAGGTCAGGATGTCCACCTCCGTCAGGTCCCGTTGAGCGCAAGATTCTGCAACTCTTGGGGGTAGACTGCATACGCGATCTACGGTTCACGATAGAAGCCCCTTCCTATCCTCCCTCACA
TGGCGAAGTCTACAAACACGGTGTCGGCGGAAGCTCCTCCTGACATAGGGATGTCCGA',
       'ATATGATGAGTTTATGTTTAATGACTTCGTGCGGTACCTCCGCATGCAGGTTTCGACCGCTAGCTTGTAACGGTTTTCCTAATGGGAGTCAGAAACAGAGTAGAGACGGTGTTTGGCTCCGAGCACTCTAGGACCCAACGAA
CACACATCTAATCCTTCTGCACGAAGTCTTAATGTTGTTGTGCAATTAACGTCTGCCC',
       'ATCTTCCGCATCCTCCTGGCTAGGCCTCAGCGATAGAGAACCTAGCACTGTGCACTTTAACTGAATATGAGGTTTGCACAGATCTTTGAGGTAAGCGTTGGAAGGAGTCTCACATGCCAGGCAGGGGGGATCTACTACTTCC
AACGATTTCAAAATACGAGTTGAGCATAGGCTGGAGACCGTATGCTGGTCGGGGTTCT',
       'CCCTACGCGCTTGGCTTACTTTACACTATTAGACTATGTCAGCAGGATGTTACGCACAAATAGTTCTCATACGTAAGTGCATTATTCTTTTAACGGCCCGCAGCACGCTCTGCAGTTGTCGATTATGACCGCTGAACGCTGT
AAATTGATGCAGTTAAATTTTGATCCCTATTTCGACATACAAAGCCGCTAAGTGCACT',
       'CCCTAAGGGCGATAATGGCTATGAAGCCAGAGTAATACTTCGCTCCTCTCAGCCCGCGTCGAACCTCCCGCCTTGCTGCTTGCTCCCCGATTGGGGTCGAGACGTTAAATTGAGTAGTGGCGTGTGATAGCCCCATTAGGAT
GCGTAAAATTCTCCTTATGGTGTGCAATAGGTGGTGACTGTTACTCTCGATGGACTTC',
       'CCCTCCTTACTAGCCCCCAGAACGGCAGAACAGCTGAACGCCGCATGTCGGGGGTGCGGCCTAACAGCATGATCTCCTCCTTGATACAATTCGAGGGAGGCTATGCCGAGGAGTTTGGTGGCGGCAGACCCCGGATCGATGA
ATATCGGCAAGGAATAGGAGAGTCCACGCAGTTTGTTGTCTGGTGACTGTGTGCGAGC',
       'TGCTGAATTGCAAAGTCAGTGCAGCTTCGGCATAACGTCATCCATCTTTGCAGAGTGGGACTGTTGCTATTTAACGGCACCTAACATGAAAAAGACGGTACTCCTATTCGATGGAGCGGCCAAGACGGAACATGTACCCCTA
TCCCGGTCGAAAAGCCAGGACTTCACTGACCACGTTAAACTGGGCGCCTACGAGCATA',
       'ACACTAACCTCTTCGCCATGCGAGAATTATGTCTGTGTTAGAACCAGAACATTCATTAGTATGTGGCAGTACGGTGGCTTGTCGTAGCATTGTGGAAAGACCCCCACGGCCCGTTATAACACACTAAGGCCCTGCCGTAATC
TTCGGGTAGTTGTTAATCTTGGTTTGCCCCATGGAGTTAGCCTGGCTCCAGTGCACCG',
       'CTAACATTCCTTTGGTTGATCCTGCCTCATCAATCGTTCGAGGTGGCAGGTCACGTGCAAAGAAACCGTTCAAATCTAGTCTTGTCCACGTTGATCTCGACGGTGTGGTGTCATCTATACGTTAGGACGTGGCAGGATACCG
ATAAGATACTGCTGACTTTGTTATGGTGATGAGCTCCGGCGGCATGAAATGTATTGAC',
       'GGAGCTGACCCTGAGCTCCCTACCGTCAATTCTGACGGTGATCGTCAACCGCGCAACTCTGTTACCAGCACACCCTTTCGGTGGTGCAGAACGTGACCGCTGGCAGCAATCTCTGATGTGATTTTTCGGCTATCGGCATTAC
CCCTGCGGATGTCACAATGAGAAACTCCATTTGTTCGCTGTAGACTAGGCGGTTCAAC',
       'GCGTCACAGATCCGCTCCATATCGCCTGTCACTTTCCGTGGAAGATTGGCGCCAGCTCTGTCGAGATGGATCGTTGTGCCGCTCCAGCAGAAGGCTGGGTTGAGCAGCAGCCGGGGCCACCTAGGCCGGGCCACCGGACTAT
CGAAGGCCTGCGTGTCCTGGTCGTAAGTAAGAGTGCCCTTCTGTGGCAGGGATTGGGG',
       'CAACCTTGGGCGGGACCTATTGGCAGTTATCTTCAGCCAAGTTCGCAGAATGACAGAAGACGGCATAAAGAGGTCTGAACCGGGTTTGTTACATCCTACTCGGCGCTATAAGCTACAACGTATTAAGATCTCTCAATCTTGG
```

ML applied to genomes (2)

→ we convert the words into a dictionary

- In this simple example, we define a dictionary having words of variable length between 3 and 4

- For each position we have 4 different possibilities (A, C, G, T), therefore:

  4^3+4^4=320 words in total

- We calculate the frequency of each word

	aaa	aaaa	aaac	aaag	aaat	aac	aaca	aacc	aacg	aact	...	ttg	ttga	ttgc	ttgg	ttgt	ttt	ttta	tttc	tttg	tttt
0	0	0	0	0	0	2	1	0	1	0	...	5	1	1	0	3	0	0	0	0	0
1	1	0	0	1	0	3	0	1	1	1	...	2	0	1	1	0	2	1	0	1	0
2	3	1	1	1	0	4	0	1	2	1	...	4	1	1	2	0	4	1	0	2	1
3	5	1	1	3	0	3	0	0	1	2	...	2	0	1	1	0	3	2	0	1	0
4	4	1	0	2	1	2	1	0	0	1	...	2	0	0	1	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
745	4	2	0	2	0	3	2	0	0	1	...	2	1	0	1	0	2	1	0	1	0
746	1	0	0	0	1	1	0	1	0	0	...	3	2	0	1	0	2	0	0	1	1
747	0	0	0	0	0	1	0	0	0	1	...	3	0	1	1	1	1	0	0	1	0
748	1	0	0	0	1	3	0	2	0	1	...	2	1	0	0	1	5	1	2	1	1
749	4	1	0	2	1	4	1	1	2	0	...	7	2	1	0	4	3	0	0	3	0

750 rows × 320 columns

Train a classificator

- Example problem (toy problem). Let's create a simple sequence dataset where the output:
  - If the word 'TATA' is present in the sequence, the label is 1
  - If it is not present, the associated label is 0
- We train a classifier, to which we provide the sequences as input (X) and the labels as output (y)
- We use a random forest or XGBoost (or any nonlinear algorithm) to build a predictive model
- In this simple and trivial example the algorithm achieves 100% performance in test

Train a random forest classifier

[9]:
```
from sklearn.ensemble import RandomForestClassifier
```

[17]:
```
rf = RandomForestClassifier(n_estimators = 100,
                            max_depth = 10)

rf.fit(X_train_transf, y_train)

y_pred = rf.predict(X_test_transf)
```

[18]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       118
           1       1.00      1.00      1.00       132

    accuracy                           1.00       250
   macro avg       1.00      1.00      1.00       250
weighted avg       1.00      1.00      1.00       250
```

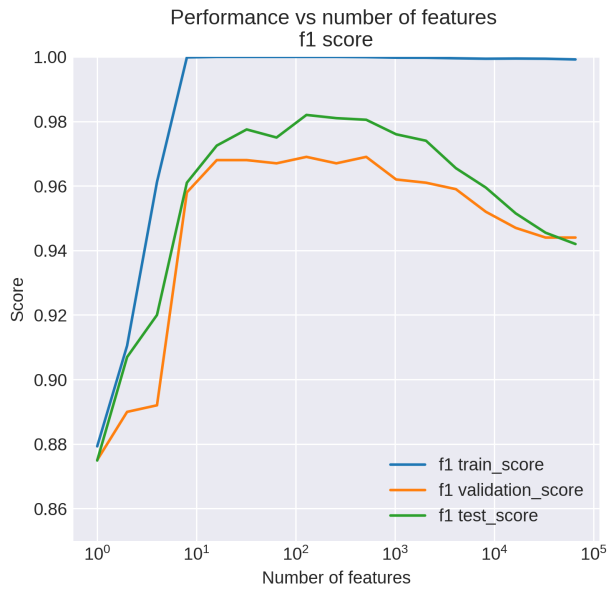
The algorithm learns this rule autonomously

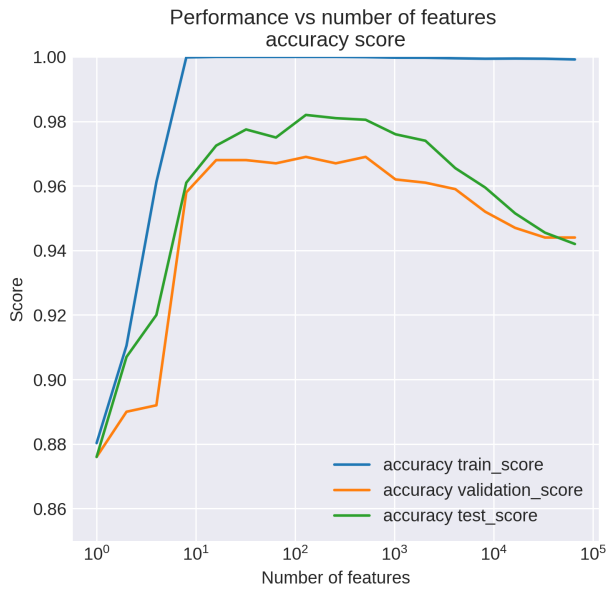
A real case

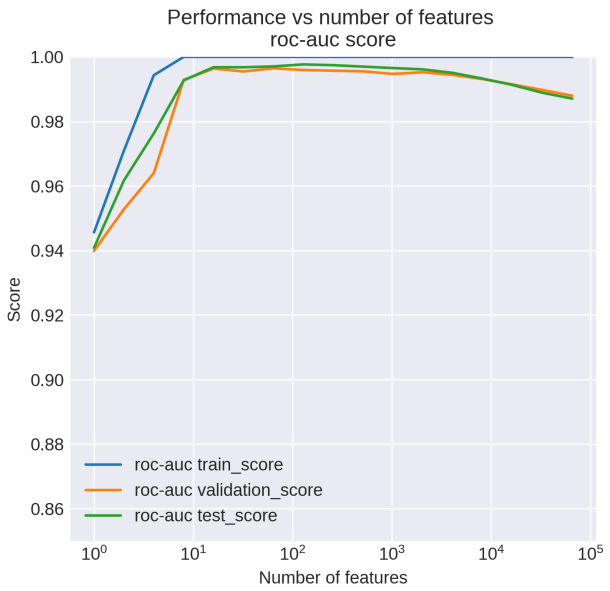
We applied NLP to approximately 20,000 real genomes (2 million nucleotides for each genome):

- Some of them are covered by a certain vaccine, others are not covered by the vaccine
- the algorithm was trained with a dictionary composed of words having between 3 and 8 letters (83760 in total)
- very high performance in validation and tests, as we can see from the attached plots

An algorithm of this type can be used to evaluate the efficacy of a vaccine X after a mutation of the virus







Kaggle competition

Gene family	Number	Class label
G protein coupled receptors	531	0
Tyrosine kinase	534	1
Tyrosine phosphatase	349	2
Synthetase	672	3
Synthase	711	4
Ion channel	240	5
Transcription factor	1343	6

	sequence	class
0	ATGCCACAGCTAGATACATCCACCTGATTTATTATAATCTTTTCAA...	4
1	ATGAACGAAAATCTATTCGCTTCTTTCGCTGCCCCCTCAATAATAG...	4
2	ATGGAAACACCCTTCTACGGCGATGAGGCGCTGAGCGGCCTGGGCG...	6
3	ATGTGCACTAAAATGGAACAGCCCTTCTACCACGACGACTCATACG...	6
4	ATGAGCCGGCAGCTAAACAGAAGCCAGAACTGCTCCTTCAGTGACG...	0

If you are interested in applying machine learning to DNA sequences, I leave you the link for a Kaggle competition:

https://www.kaggle.com/code/nageshsingh/demystify-dna-sequencing-with-machine-learning

```
from sklearn.feature_extraction.text import TfidfVectorizer

tf = TfidfVectorizer(ngram_range=(3, 10), analyzer='char', max_features=10000)

X_transf = tf.fit_transform(X)
X_transf = X_transf.toarray()
```

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()

rf.fit(X_train, y_train)
rf.score(X_test, y_test)
```

Deep learning

Deep learning applied to genomes

Gunasekaran et al., 2021

NLP is not the only possibility to study DNA sequences.

Deep learning also works very well when:

- the sequences are short (a few hundred nucleotides)
- it is important to preserve the order of the sequence

If these two conditions exist, we can use Bidirectional Recurrent Neural Networks (BRNNs)

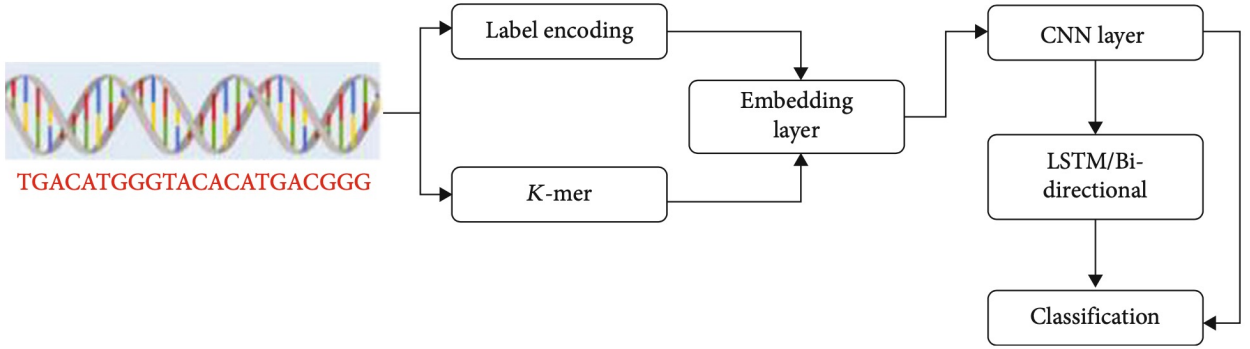


FIGURE 6: Workflow of the proposed model for the classification of DNA sequence.

TABLE 2: Complete architecture specification of proposed CNN model.

Layer (type)	Output shape	Param #
Embedding	(None, 1000, 8)	128
Conv1D_1	(None, 1000, 128)	3200
MaxPooling1D_1	(None, 500, 128)	0
Conv1D_2	(None, 500, 64)	24640
MaxPooling_2	(None, 250, 64)	0
Flatten	(None, 16)	0
Dense_1	(None, 128)	2176
Dense_2	(None, 64)	8256
Dense_3	(None, 6)	390

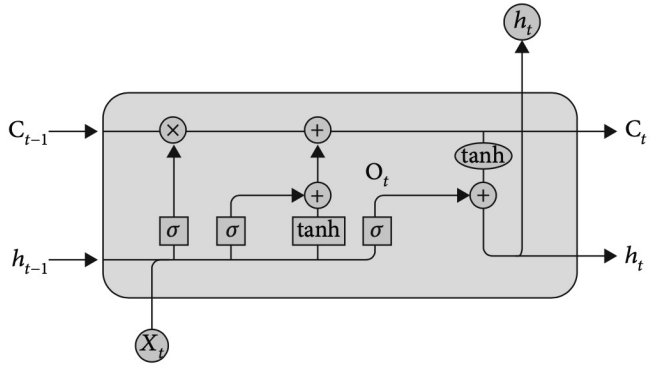


FIGURE 7: Architecture of the LSTM model.

Example of one-hot encoding

An example taken from Tensorflow Blog.

Tensorflow is one of the leading libraries for machine learning, developed by the Google Brain Team and released in November 2015.

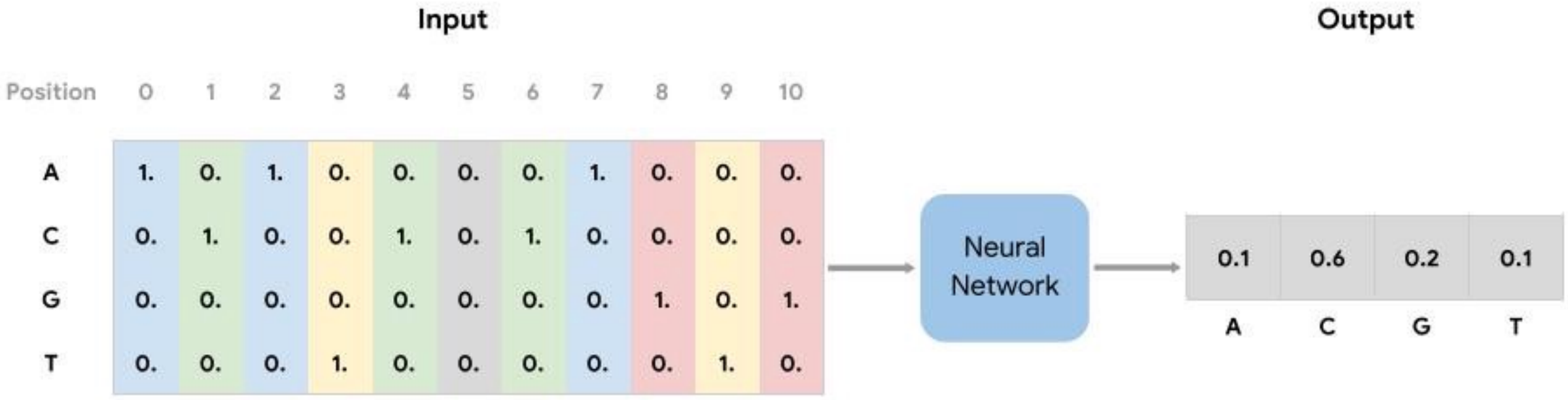


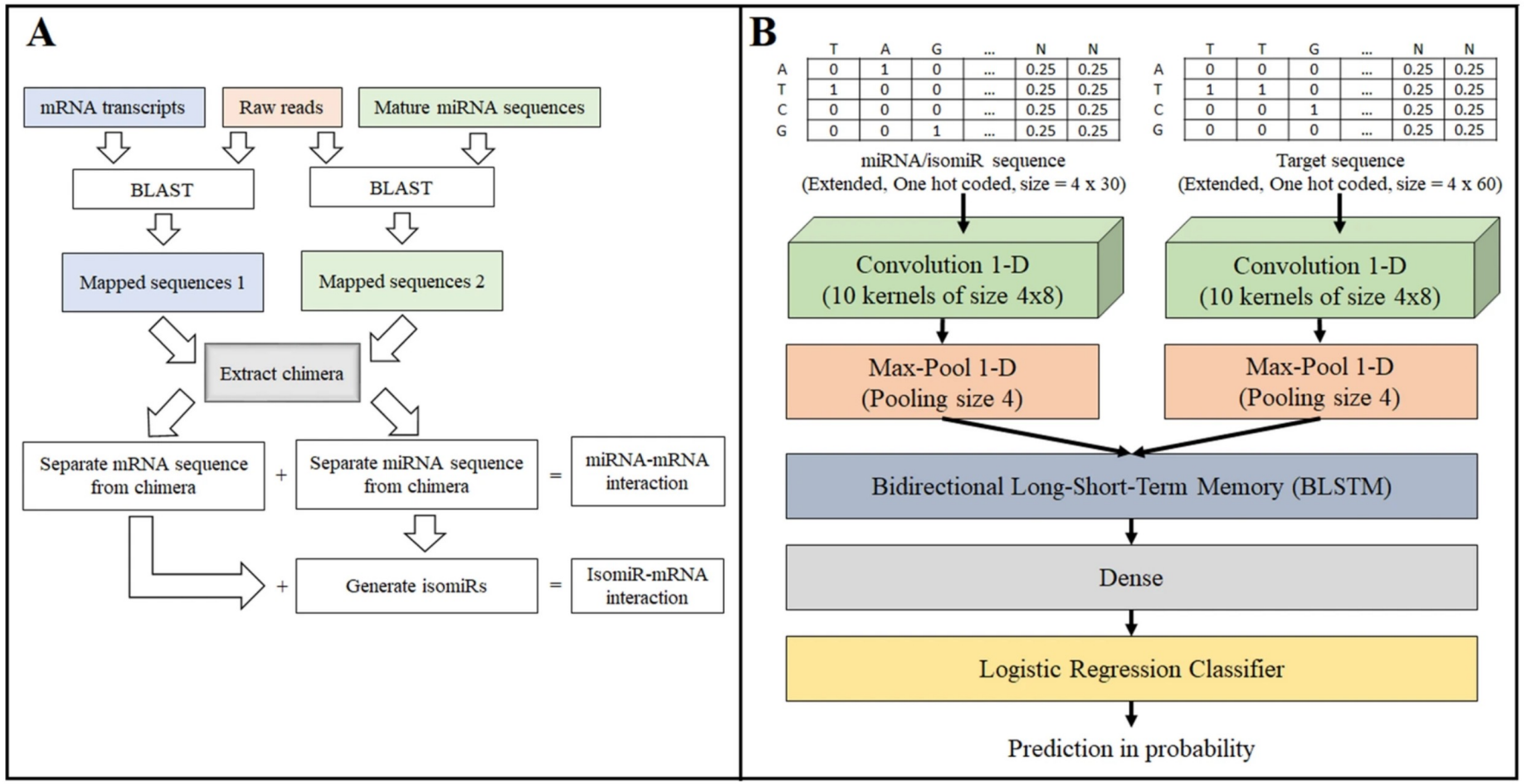
Figure 3: The one hot encoded input is shown above. The middle base, which we want the model to predict, has been zeroed out. The reference DNA sequence is ACATCCCAGTG.

The network outputs a distribution over the possible bases. The final prediction is the base with highest probability, which is C in this case.

Interaction between micro-RNA and target genes

Figure 1

From: A deep learning method for miRNA/isomiR target detection Scientific Reports, Nature, 2022



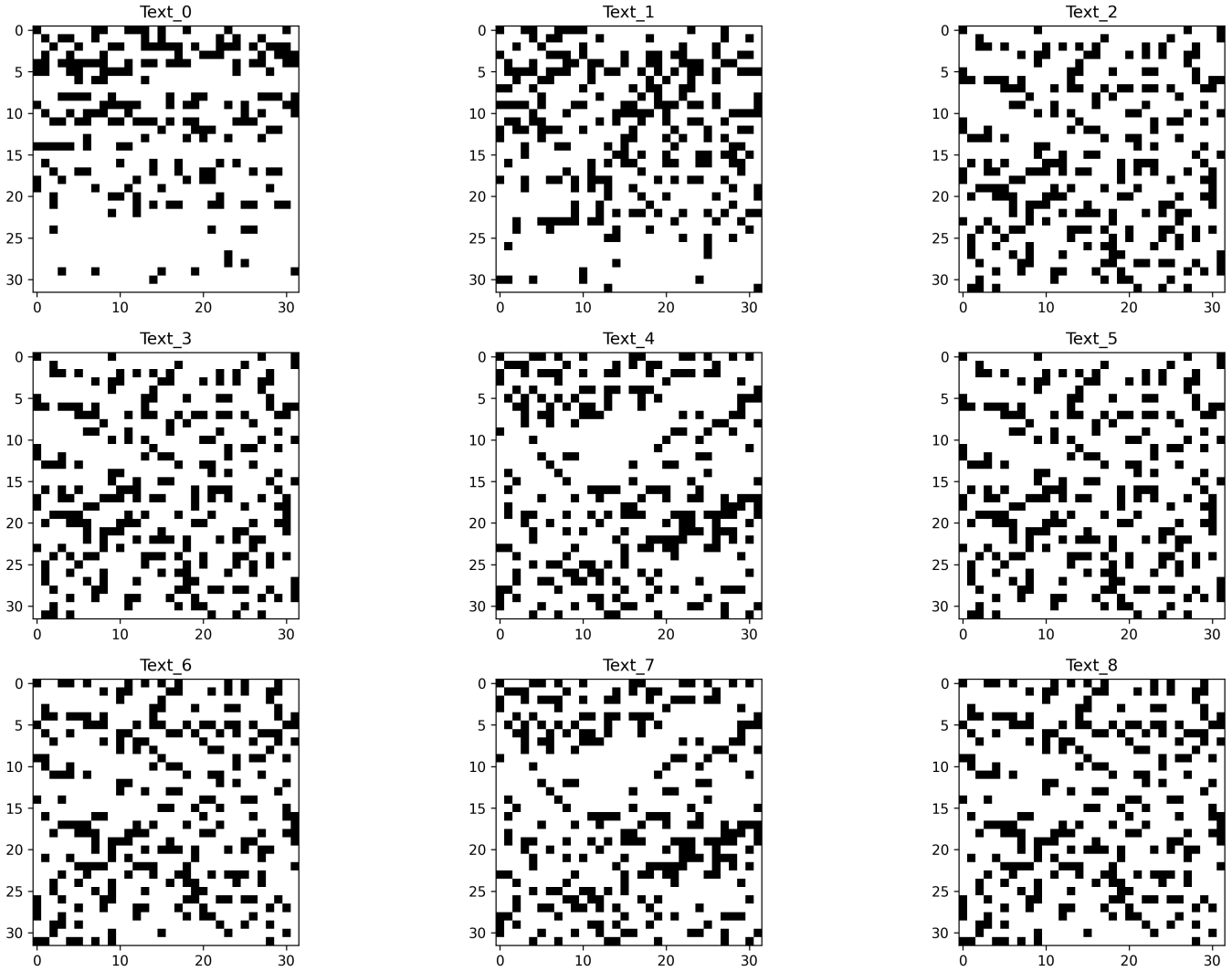
(**A**) The pipeline to obtain miRNA/isomiR–mRNA interactions. (**B**) The DMISO model structure.

Deep learning can also be used to study interactions between:

- a sequence of micro-RNA (a few dozen nucleotides)
- a target (order of hundreds of nucleotides)

In the example on the left, the sequences are converted into images and computer vision techniques are used

Sequences converted into images



On the left panel, there are 9 sequences (few hundreds of nucleotides) converted into images, using the one-hot encoding technique explained in the previous slides

Applications

Derive information from sequences

Figura da Mykytyn et al., Science 2022

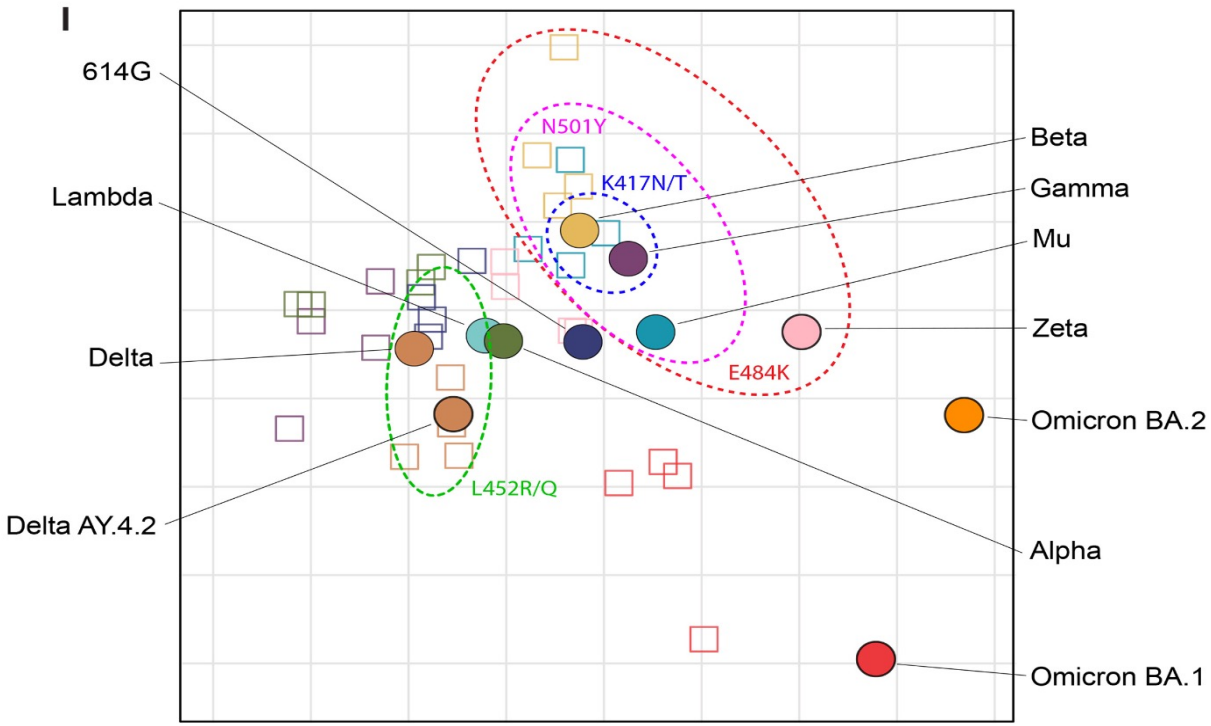
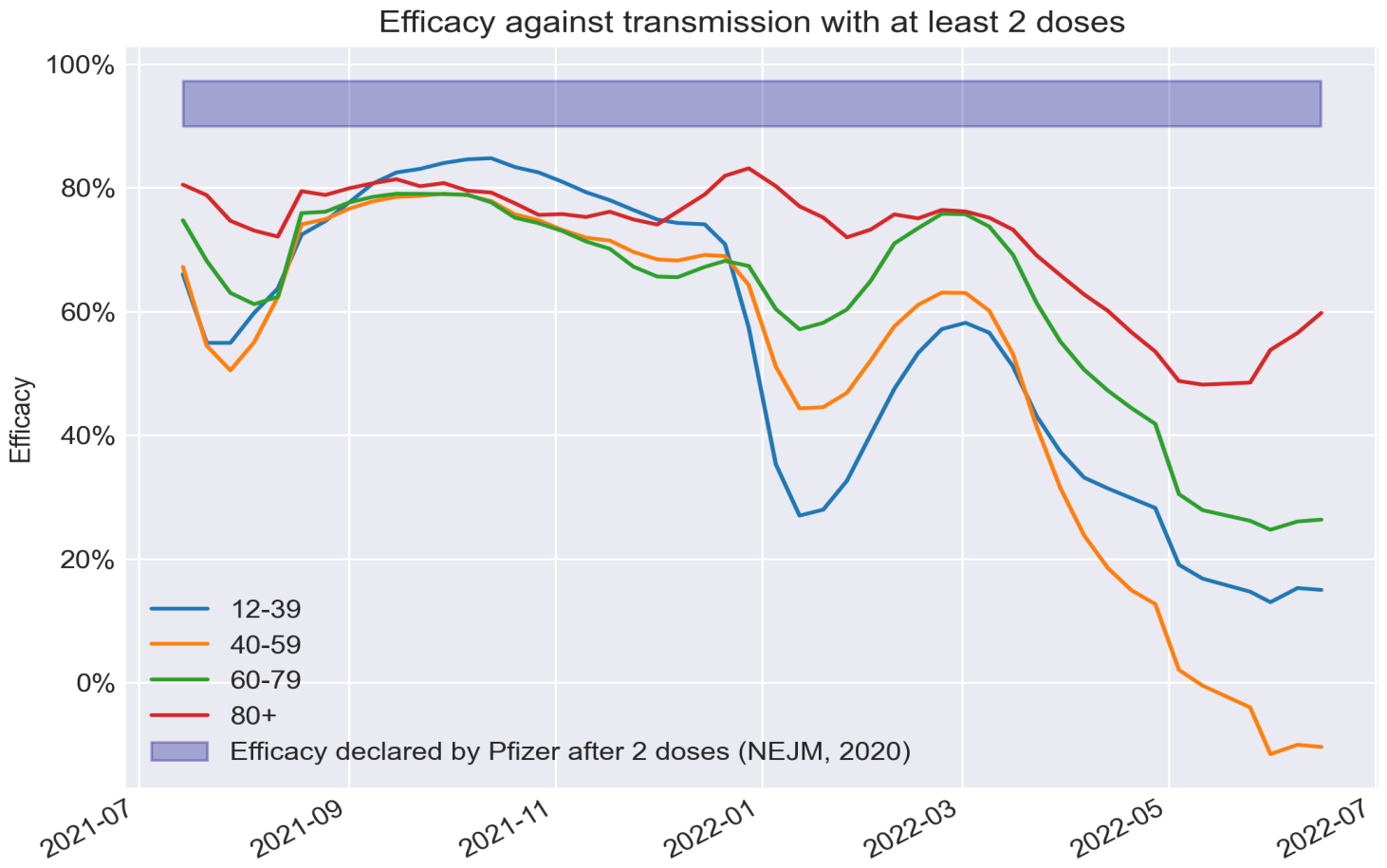


Fig. 4. Antigenic cartography using authentic SARS-CoV-2.

(**A-H**) Neutralizing titers of hamsters infected with either (**A**) 614G, (**B**) Alpha, (**C**) Beta, (**D**) Gamma, (**E**) Zeta, (**F**) Delta, (**G**) Mu or (**H**) Omicron BA.1 viruses. (**I**) Multidimensional scaling was used to create an antigenic map utilizing PRNT50 titers generated from authentic SARS-CoV-2 on Calu-3 cells. See legend to Fig. 3 for details. Subdivided by dotted ellipses are variants possessing overlapping substitutions as indicated. Geometric mean is displayed above each graph. PRNT50: plaque reduction neutralization titers resulting in 50% plaque reduction. Dotted lines indicate limits of detection. Error bars indicate SEM.

Derive information from “real world data”



The effectiveness of the Covid vaccine (in protecting against infection) decreases over time.

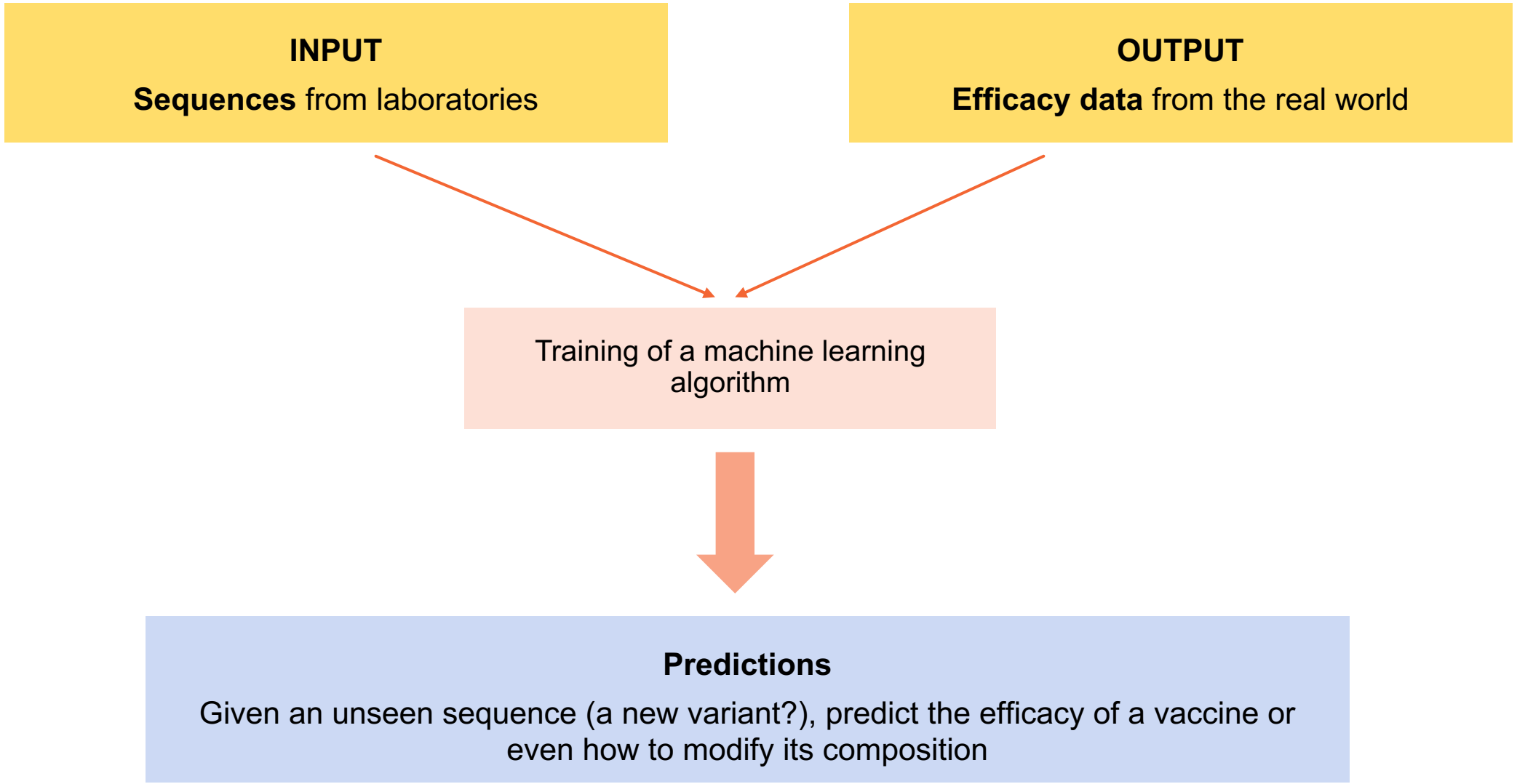
Possible explanations:

- mutations of the virus
- limited duration of effectiveness

The «bumps» in the curves are linked to the administration of the booster doses

The curves of efficacy are obtained by using the public repository https://github.com/apalladi/covid_vaccini_monitoraggio

Combining molecular biology with real world data



Machine Learning applied to “Human Data”

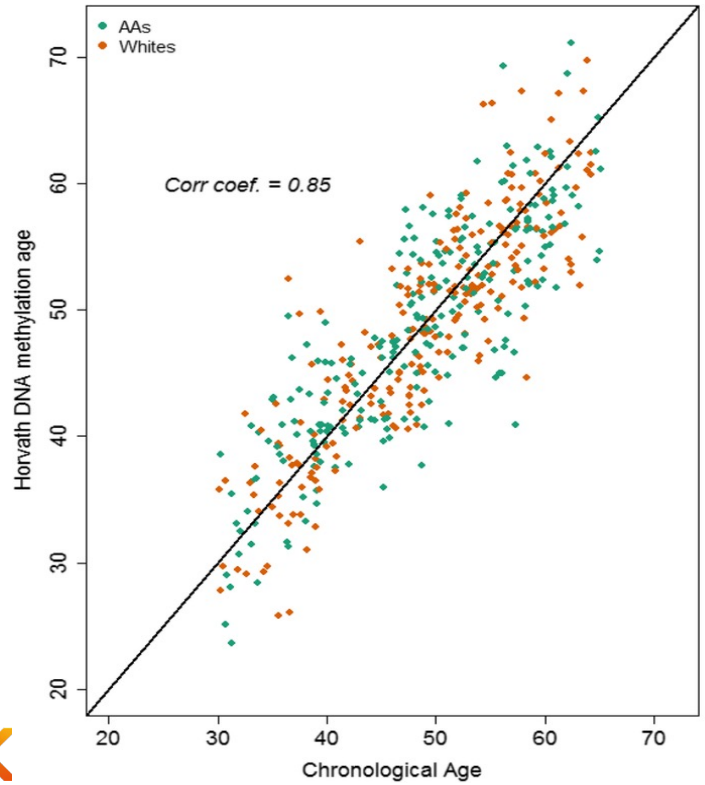
GSK

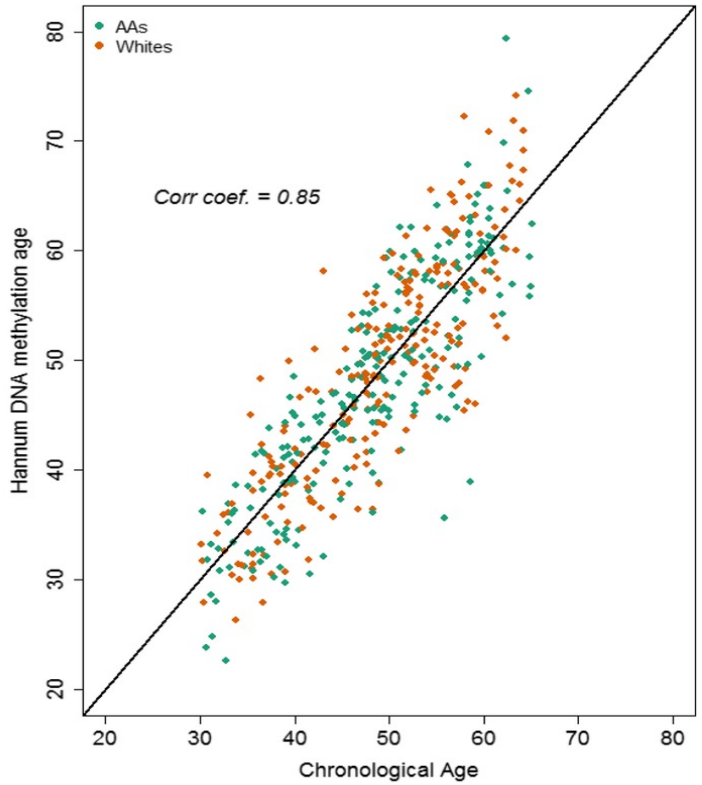
Definition of “Human Data”

With the term “Human Data” we refer to data relating to biomarkers capable of providing information about an individual.

Example: levels of specific proteins in the blood

Use: prediction of the chronological and biological age of a subject





Biological age is usually defined as the difference between observed age and predicted age.

In other words, the subjects above the line have a biological age greater than their chronological age.

The subjects under the line have a biological age lower than their chronological age

Figure from Tajuddin et al., BMC 2019

“Inflammaging” is a hallmark of aging and influences response to vaccines

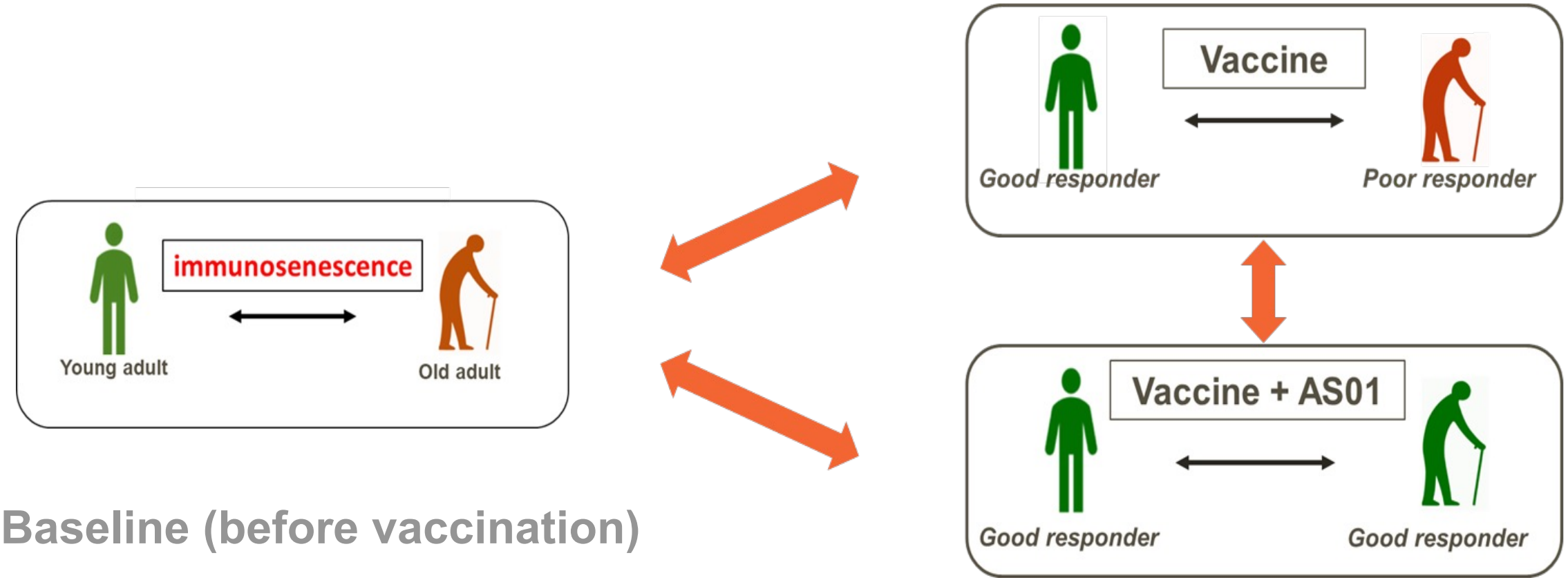
Biological age reflects “inflammaging” more accurately than chronological age [1,2]
The increase in basal inflammation in old age negatively affects the response to vaccination [3,4,5]

Low bio-age

High bio-age

1. Sayed, N., et al. Nat. Aging. (2021). https://doi.org/10.1038/s43587-021-00082-y
2. Furman, D., et al. Nat. Med. (2017). https://doi.org/10.1038/nm.4267
3. Pereira, B., et al Front. Immunol. (2020). https://doi.org/10.3389/fimmu.2020.583019
4. Vukmanovic-Stejic, M., et al. J. Allergy Clin. Immunol. (2018). https://doi.org/10.1016/j.jaci.2017.10.032
5. Mannick, J.B., et al. Sci. Transl. Med (2014). https://doi.org/10.1126/scitranslmed.3009892

Understanding the role of biological age in response to vaccination



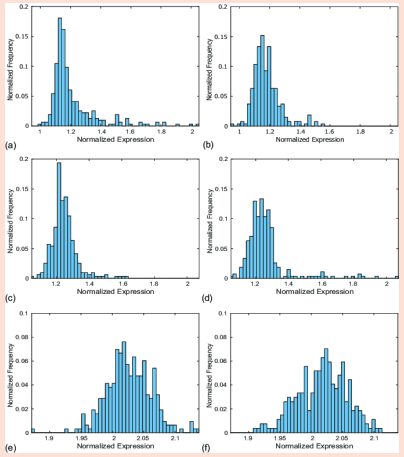
Baseline (before vaccination)

After vaccination

Machine learning with “human data”

Input:

- proteomics data (hundreds of biomarkers)
- transcriptomics data (thousands of biomarkers)
- DNA methylation data (hundreds of thousands of biomarkers)

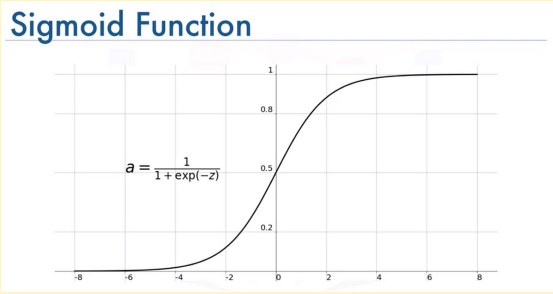


Mostly used algorithms:

- Random forest
- XGBoost
- SVM
- Logistic Regression
- Lasso/Ridge regression

Output:

- Prediction of chronological age
- Biological age prediction
- Classification of the immune response
- Frailty index classification

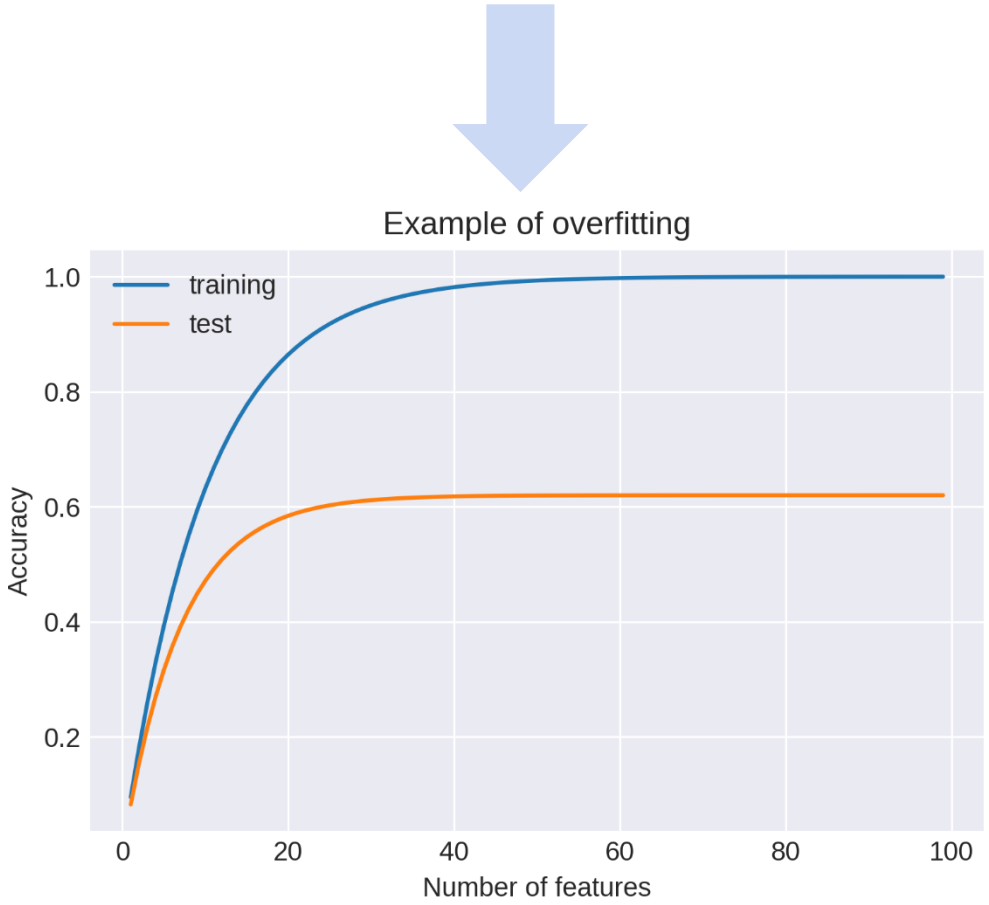


GSK

Typical situation with “human data”

Lots of biomarkers (thousand), few samples (hundreds)

Implication of di n_samples << n_biomarkers



How can we mitigate overfitting?

- **Feature selection**
- Using **simple models**

Complex models, such as Random Forest, XGBoost or neural networks are usually not adequate to tackle these types of problems.

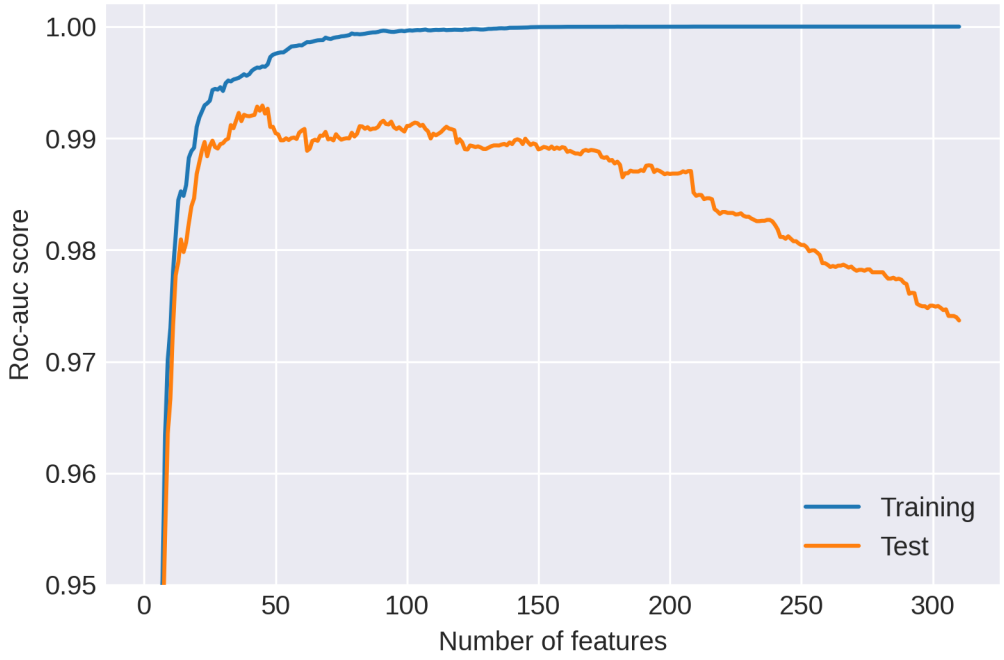
Example of typical human data problem

Problem: we have approximately 100 subjects and 311 biomarkers (proteins) for each subject were measured in the blood

Objective: build a model to classify patients based on age (over 45 and under 45) and evaluate which are the most important biomarkers

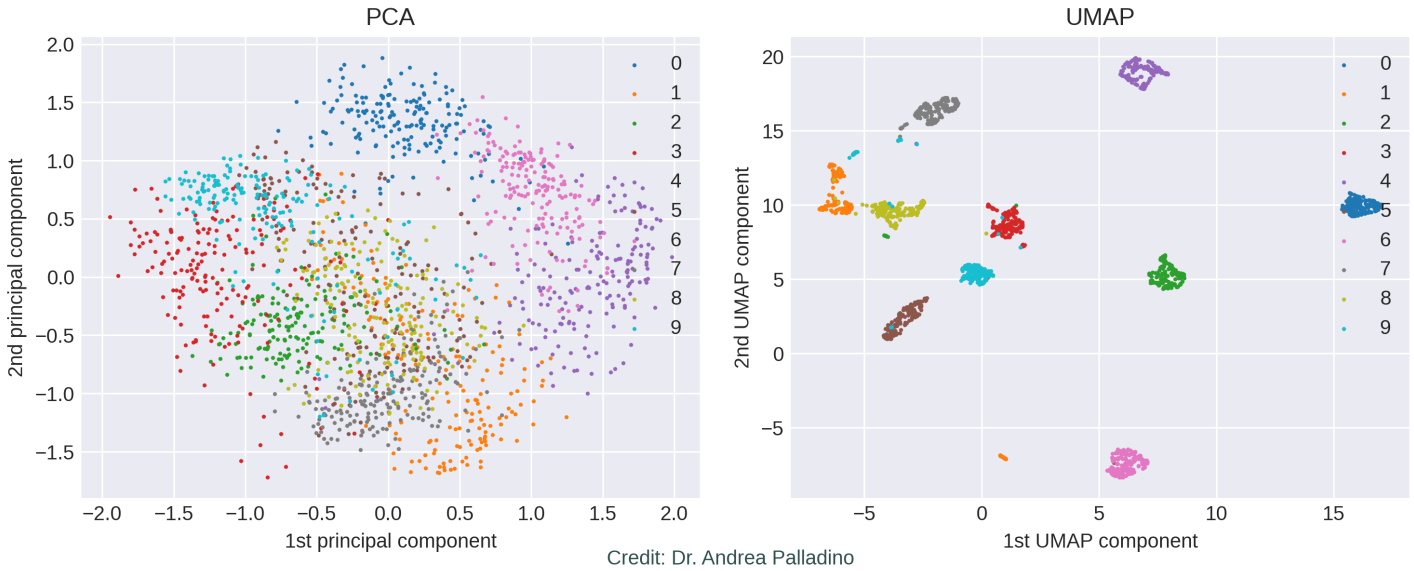
Solution:

- Recursive Feature Elimination (RFE) to produce a biomarker ranking
- Multivariate model (logistic regression) starting from the most important biomarker and adding one at a time
- 5 fold cross validation
- The best average test score is obtained with 45 biomarkers, with roc-auc score above 0.99

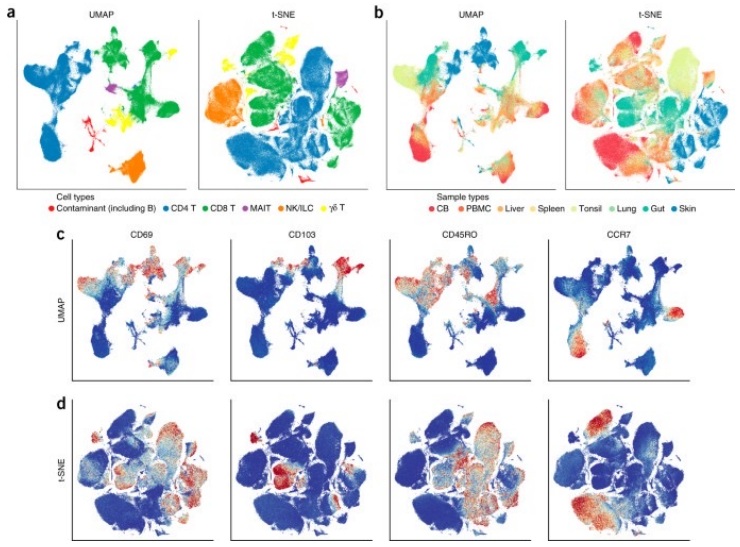


Example of dimensionality reduction

- **PCA**: linear combination of features, to maximize the variance
- **UMAP**: non linear combination of features, to preserve the distances



Credit: Dr. Andrea Palladino



Dimensionality reduction for visualizing single-cell data using UMAP, Nature Biotechnology 2019

Epidemiological models

GSK

Machine learning and pandemic

Machine learning has proven to be very powerful in various fields.

But during the Covid19 pandemic, analytical models outperformed AI/ML algorithms

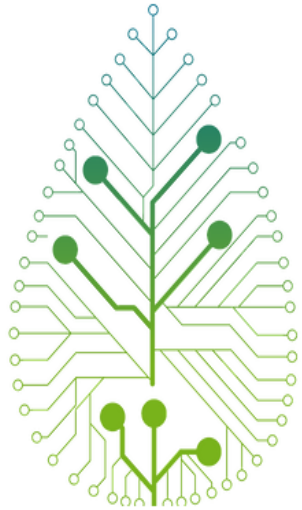
'PrIMAVeRa' means 'Predicting the Impact of Monoclonal Antibodies & Vaccines on Antimicrobial Resistance'.

It is a European project funded by the Innovative Medicines Initiative 2 (IMI2) with the goal of developing an open-source, web-based platform combining mathematical models with a comprehensive epidemiological repository (i.e., with data referring to health and economic outcomes). This platform will aim to enable policymakers to reach data-driven decisions regarding the prioritisation of specific vaccines and monoclonal antibodies (mAbs), informing the strategic allocation of limited resources.

A project sustainability plan will also be designed by the project partners with the goal of providing long-term access to the project results, including the models, once the project has finished.

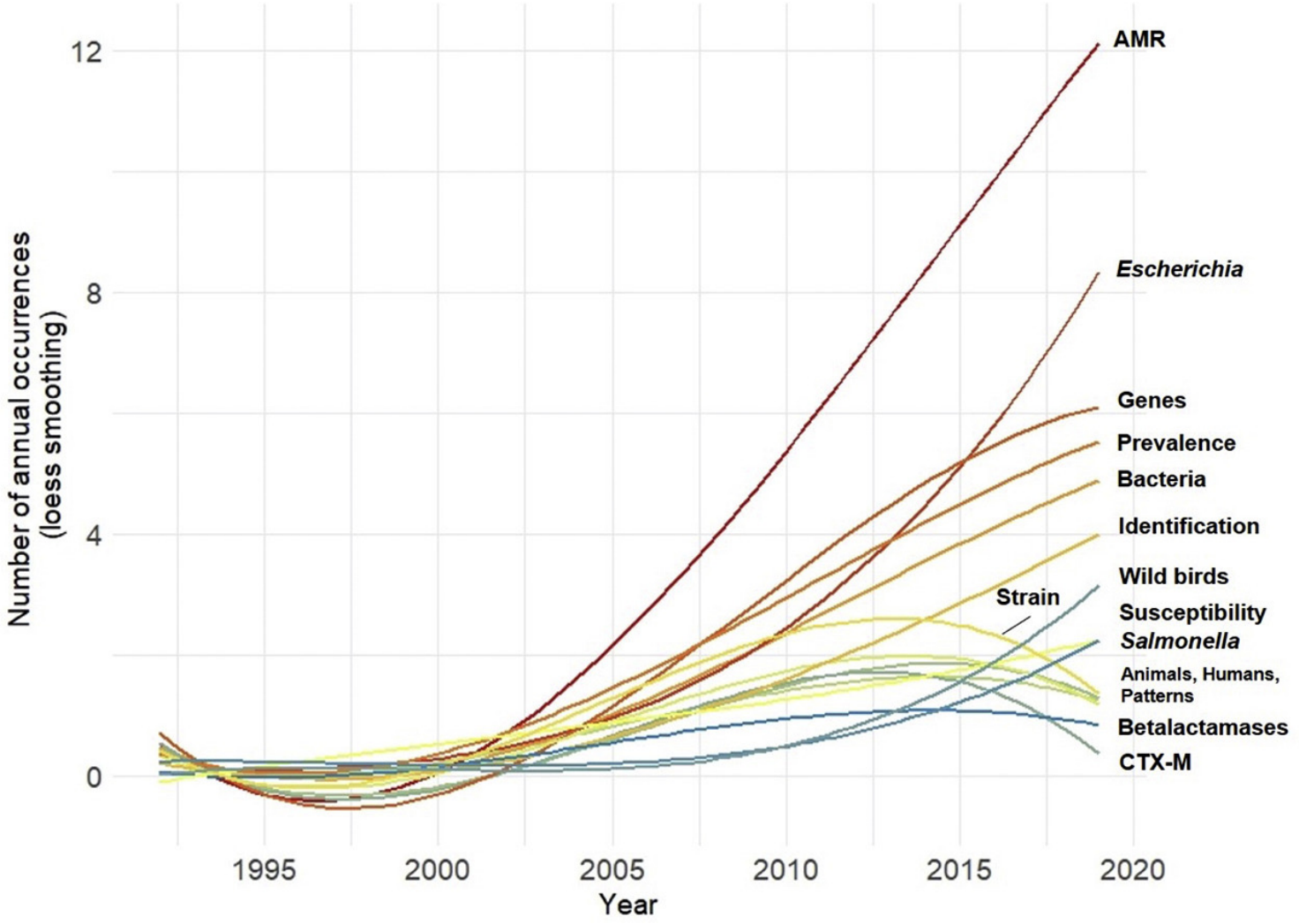
PRIMAVERA is part of the AMR Accelerator Programme.

IMI AMR Accelerator
Tackling antibiotic resistance together



GSK

Resistant pathogens



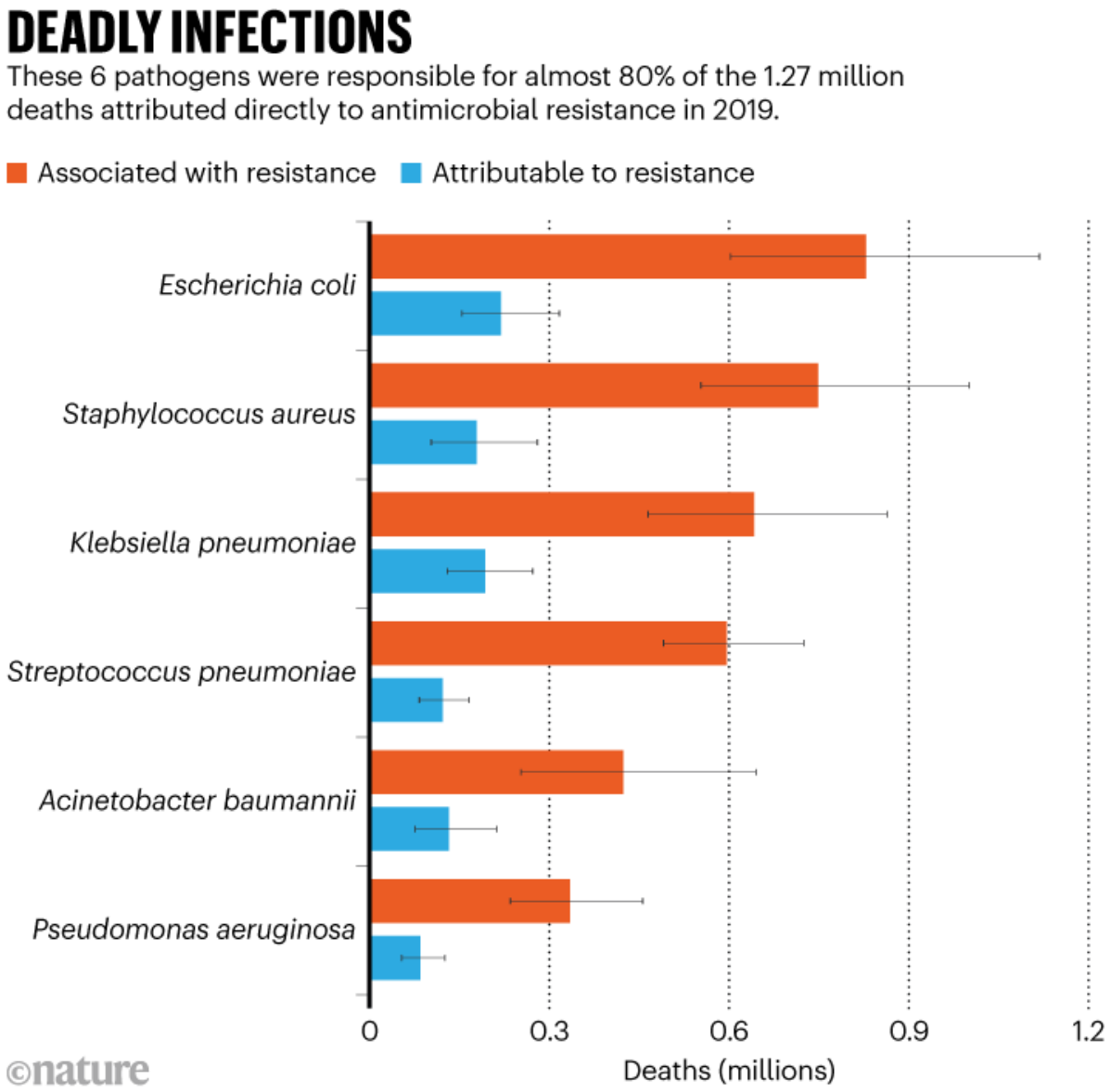
Torres et al., Elsevier, 2020

Number of deaths associated with antibiotic resistance (AMR)

1.27 million deaths in 2022 were directly caused by antibiotic resistance, meaning that antibiotic resistance caused more deaths than:

- HIV/AIDS (864,000 deaths
- malaria (643,000 deaths)

Nature 2022



Compartmental model

A “compartmental model” is a type of mathematical model that

simulates

how individuals belonging to different “compartments”

interact

GSK

Transitions between compartments

Generally, the equation of a state is expressed as:

$$\frac{dD_{pi}}{dt} = F_{ap} - F_{dp}$$

where:

- D_{pi} denotes the state of the population of the compartment *I*
- F_{ap} expresses the flow of individuals/day **entering** the state *I*
- F_{dp} expresses the flow of individuals/day **leaving** the state *I*

GSK

The simplest compartmental model: SIR model

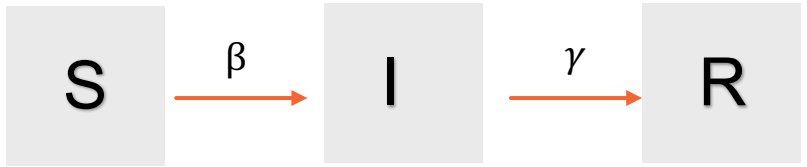
A system of Ordinary Differential Equations (ODE)

The SIR model consists of 3 components:

- **S** for the number of Susceptibles
- **I** for the number of Infected
- **R** for the number of Removed (Recovered + Deaths)

$$\begin{cases} \frac{dS}{dt} = -\beta I \frac{S}{N} \\ \frac{dI}{dt} = \beta I \frac{S}{N} - \gamma I \\ \frac{dR}{dt} = \gamma I \end{cases}$$

The SIR model has two free parameters: β, γ



$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0$$

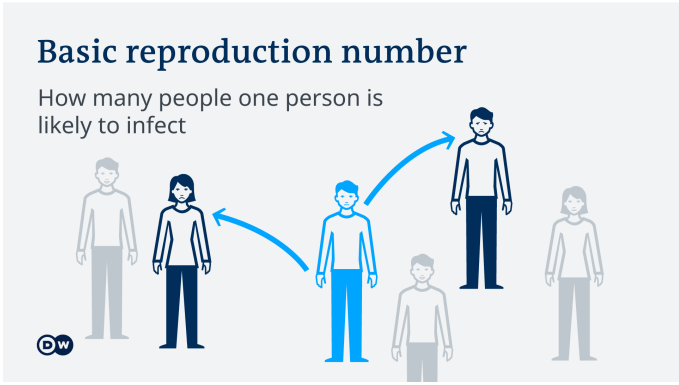
The total population is constant

Parameters of SIR model

- β is related to the rate of contacts among persons
- γ is the recovery rate

The **basic reproduction number (R0)** is defined as follows:

$$R_0 = \frac{\beta}{\gamma}$$

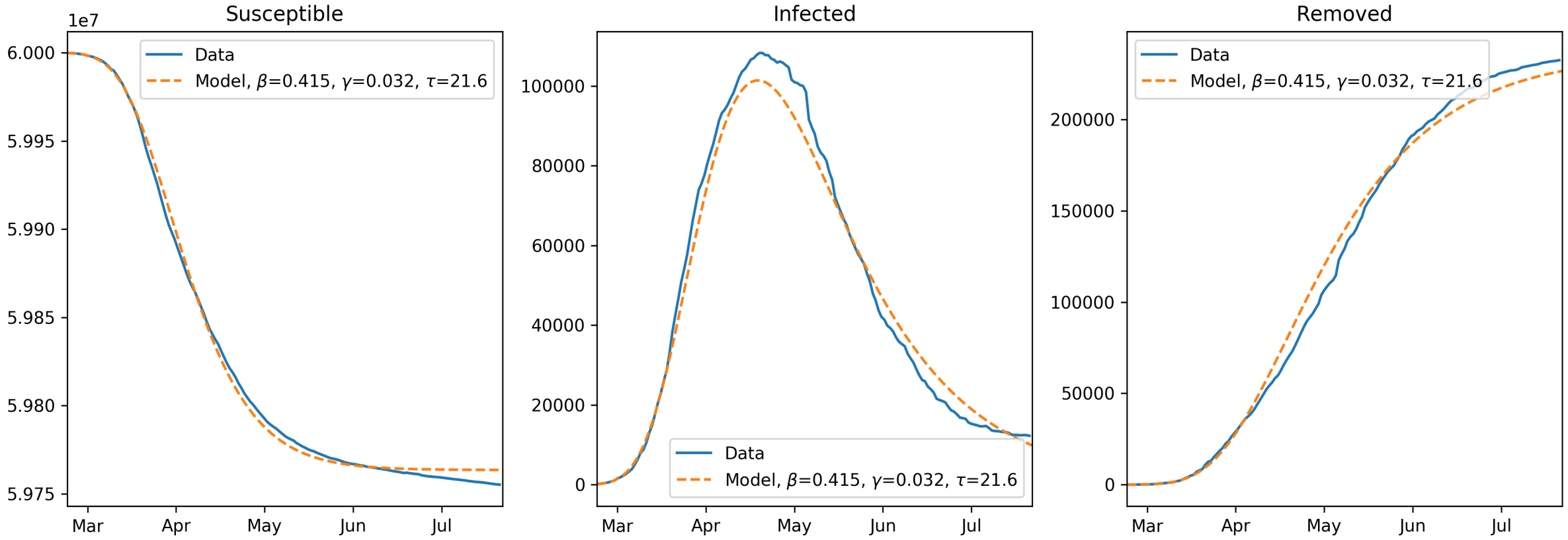


The evolution over time of the previous parameter, commonly called Rt, is given by:

$$R_t(t) = \frac{\beta}{\gamma}\frac{S(t)}{N}$$

Time-dependent SIR model

Description of the first wave of Covid in Italy (March – July 2020)



Palladino et al., arXiv https://arxiv.org/abs/2005.08724

The time-dependent SIR model offers a good description of the observed data, with an average error of 6% over the three curves

An example of complex compartmental model applied to AMR real case

Kaufhold et al., CID 2019:68

- S = susceptible
- V = vaccinated
- I = infected
- R = removed
- C = chronic disease

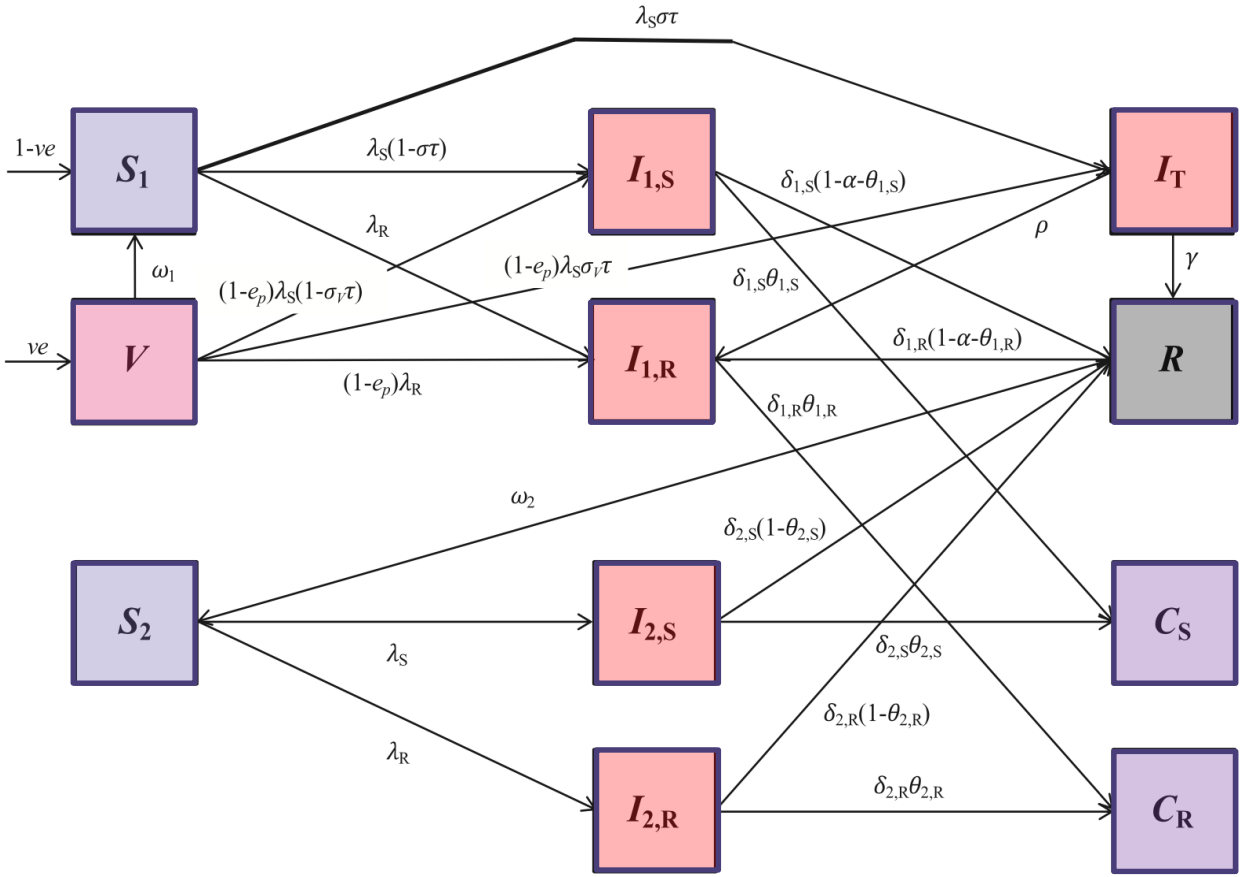


Figure 1. Compartmental structure of transmission dynamic model. The states and parameters are defined in the "Model description" section, while the differential equations for the model are in the Supplementary Material. Natural mortality from each compartment (at rate μ) was excluded from the diagram for visual clarity.

GSK

Conclusion

Conclusion

In this talk we discussed the flexibility and power of machine learning, which also makes its use very useful in the medical and pharmaceutical fields for:

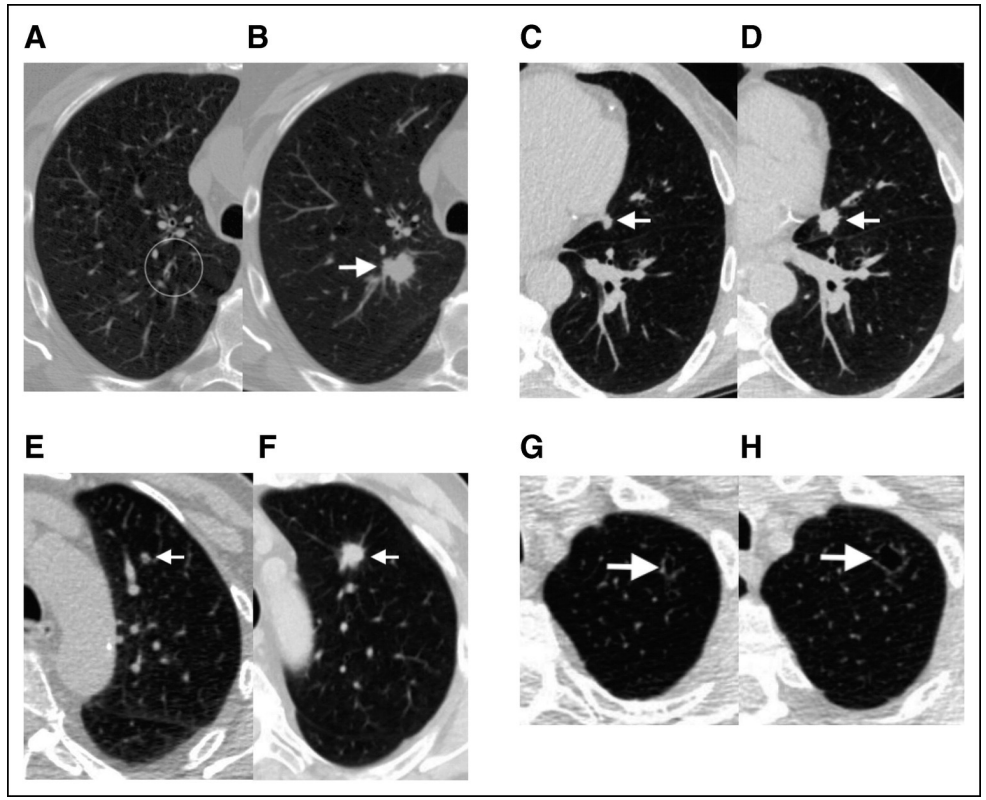
- extraction of information from DNA sequences (NLP)
- image recognition in the medical field (e.g. Sybil, prediction of the development of lung cancer)
- prediction of response to vaccines using biomarkers (e.g. protein levels in the blood)
- dimensionality reduction and clustering to group biomarkers

We should not forget that analytical models also exist and in some fields (such as epidemiology) they still work well and have not been replaced by machine learning.

Backup slides

Computer vision: Sybil

Sybil: A Validated Deep Learning Model to Predict Future Lung Cancer Risk From a Single Low-Dose Chest Computed Tomography, Journal of Clinical Oncology, April 2023



Cos'è: Sybil è un algoritmo di computer vision allenato su immagini provenienti da Low-dose computed tomography (LDCT).

Scopo: Predire da una LDCT la probabilità con cui un paziente svilupperà cancro ai polmoni a distanza di 1 e 6 anni

Utilizzo: Sybil richiede in input una LDCT, mentre non richiede altri dati clinici o annotazioni da parte dei radiologi.

Validazione: Sybil è stato validato utilizzando 3 diversi dataset:

- 6,282 LDCTs da NLST (National Lung Screening Trial)
- 8,821 LDCTs da Massachusetts General Hospital (MGH)
- 12,280 LDCTs da Chang Gung Memorial Hospital

Performance:

- 0.92 roc-auc score nella predizione ad 1 anno
- 0.79 roc-auc score nella predizione a 6 anni