AI for Genomics: from CNNs and LSTMs to Transformers

Nikolay Oskolkov, Group Leader (PI) at LIOS, Riga, Latvia

Physalia course, 09.09.2025

Session 3b: CNN applications for metagenomics data and source tracking problem

@NikolayOskolkov

@oskolkov.bsky.social

Personal homepage:
https://nikolay-oskolkov.com

Image generated at www.canva.com

Session content

Topics we'll cover in this session:

1) Introduction to metagenomics source tracking problem

2) Shotgun metagenomics is a potential Big Data

3) Introduction to Human Microbiome Project (HMP) data

4) Introduction to microbial metagenomics data analysis

5) Predicting microbial composition with CNN neural networks

Deep Learning for DNA sequencing

PERSPECTIVE
https://doi.org/10.1038/s41588-018-0295-5
nature genetics

A primer on deep learning in genomics

James Zou[1,2,3]*, Mikael Huss[4,5], Abubakar Abid[3], Pejman Mohammadi[6,7], Ali Torkamani[6,7] and Amalio Telenti[6,7]*

Deep learning methods are a class of machine learning techniques capable of identifying highly complex patterns in large datasets. Here, we provide a perspective and primer on deep learning applications for genome analysis. We discuss successful applications in the fields of regulatory genomics, variant calling and pathogenicity scores. We include general guidance for how to effectively use deep learning methods as well as a practical guide to tools and resources. This primer is accompanied by an interactive online tutorial.

Deep learning has made impressive recent advances in applications ranging from computer vision to natural-language processing. This primer discusses the main categories of deep learning methods and provides suggestions for how to effectively use deep learning in genomics. The primer is intended for bioinformaticians who are interested in applying deep learning approaches, and for genomicists and general biomedical researchers who seek a high-level understanding of this rapidly evolving field. Computer scientists may also use the primer as an introduction to the exciting applications of deep learning in genomics. However, we do not provide a survey of deep learning in the biomedical field, which has been broadly covered in recent reviews[1–5]. This paper is accompanied by an interactive tutorial that we have created for interested readers to build a convolutional neural network to discover DNA-binding motifs (see URLs).

Deep learning as a class of machine learning methods

Machine learning techniques have been extensively used in genomics research[3,6]. Machine learning tasks fall within two major categories: supervised and unsupervised. In supervised learning, the goal is predicting the label (classification) or response (regression) of each data point by using a provided set of labeled training examples. In unsupervised learning, such as clustering and principal component analysis, the goal is learning inherent patterns within the data themselves.

The ultimate goal in many machine learning tasks is to optimize model performance not on the available data (training performance) but instead on independent datasets (generalization performance). With this goal, data are randomly split into at least three subsets: training, validation and test sets. The training set is used for learning the model parameters (detailed discussion on parameter optimization in ref. [1]), the validation set is used to select the best model, and the test set is kept aside to estimate the generalization performance (Fig. 1). Machine learning must reach an appropriate balance between model flexibility and the amount of training data. An overly simple model will underfit and fail to let the data 'speak'. An overly flexible model will overfit to spurious patterns in the training data and will not generalize.

Large neural networks, a main form of deep learning, are a class of machine learning algorithms that can make predictions and perform dimensionality reduction. The key difference between deep learning and standard machine learning methods used in genomics—e.g., support vector machine and logistic regression—is that deep learning models have a higher capacity and are much more flexible. Typical deep learning models have millions of trainable parameters. However, this flexibility is a double-edged sword. With appropriately curated training data, deep learning can automatically learn features and patterns with less expert handcrafting. It also requires greater care to train on and to interpret the underlying biology. Box 1 summarizes the main messages of this primer on how to effectively use deep learning in genomics.

Setting up deep learning

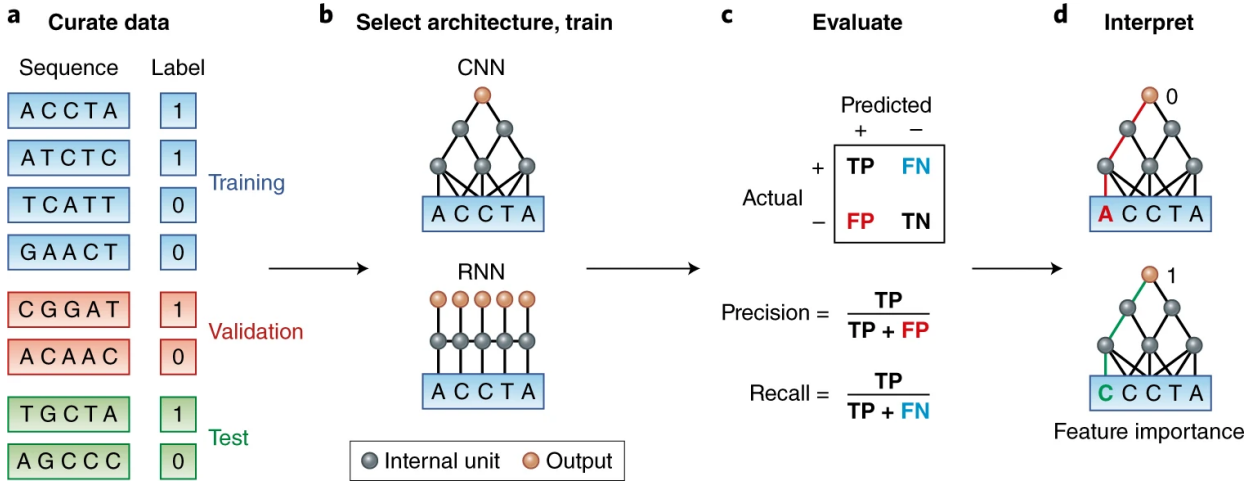
Deep learning is an umbrella term that refers to the recent advances in neural networks and the corresponding training platforms (e.g., TensorFlow and PyTorch). The starting point of a neural network is an artificial neuron, which takes as input a vector of real values and computes the weighted average of these values followed by a nonlinear transformation, which can be a simple threshold[7]. The weights are the parameters of the model that are learned during training. The power of neural networks stems from individual neurons being highly modular and composable, despite their simplicity[8]. The output of one neuron can be directly fed as input into other neurons. By composing neurons together, a neural network is created.

The input into a neural network is typically a matrix of real values. In genomics, the input might be a DNA sequence, in which the nucleotides A, C, T and G are encoded as [1,0,0,0], [0,1,0,0], [0,0,1,0] and [0,0,0,1]. Neurons that directly read in the data input are called the first, or input, layer. Layer two consists of neurons that read in the outputs of layer one, and so on for deeper layers, which are also referred to as hidden layers. The output of the neural network is the prediction of interest, e.g., whether the input DNA is an enhancer. Box 2 describes key terms and concepts in deep learning.

There are three common families of architectures for connecting neurons into a network: feed-forward, convolutional and recurrent. Feed-forward is the simplest architecture[7]. Every neuron of layer i is connected only to neurons of layer $i + 1$, and all the connection edges can have different weights. Feed-forward architecture is suitable for generic prediction problems when there are no special relations among the input data features.

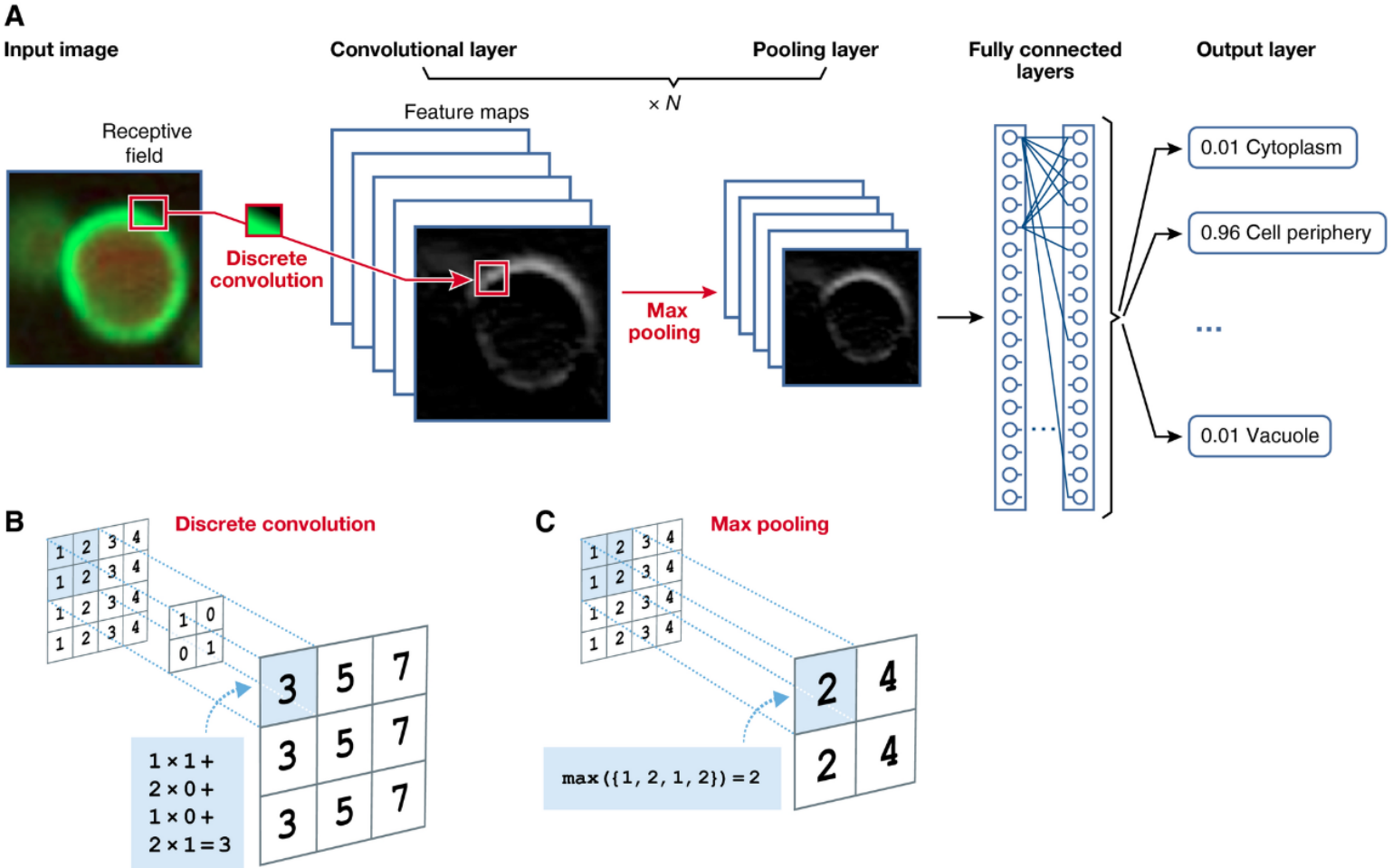
In a convolutional neural network (CNN), a neuron is scanned across the input matrix, and at each position of the input, the CNN

[1]Department of Biomedical Data Science, Stanford University, Palo Alto, CA, USA. [2]Chan-Zuckerberg Biohub, San Francisco, CA, USA. [3]Department of Electrical Engineering, Stanford University, Palo Alto, CA, USA. [4]Peltarion, Stockholm, Sweden. [5]Department of Learning, Informatics, Management and Ethics, Karolinska Institutet, Stockholm, Sweden. [6]Scripps Research Translational Institute, La Jolla, CA, USA. [7]Department of Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA, USA. *e-mail: jamesz@stanford.edu; atelenti@scripps.edu



One-dimensional convolutional neural networks (CNNs) and Recurrent Neural Networks (example: LSTMs) are suitable for working with sequential data. Therefore, can be used for encoding DNA nucleotide sequences.

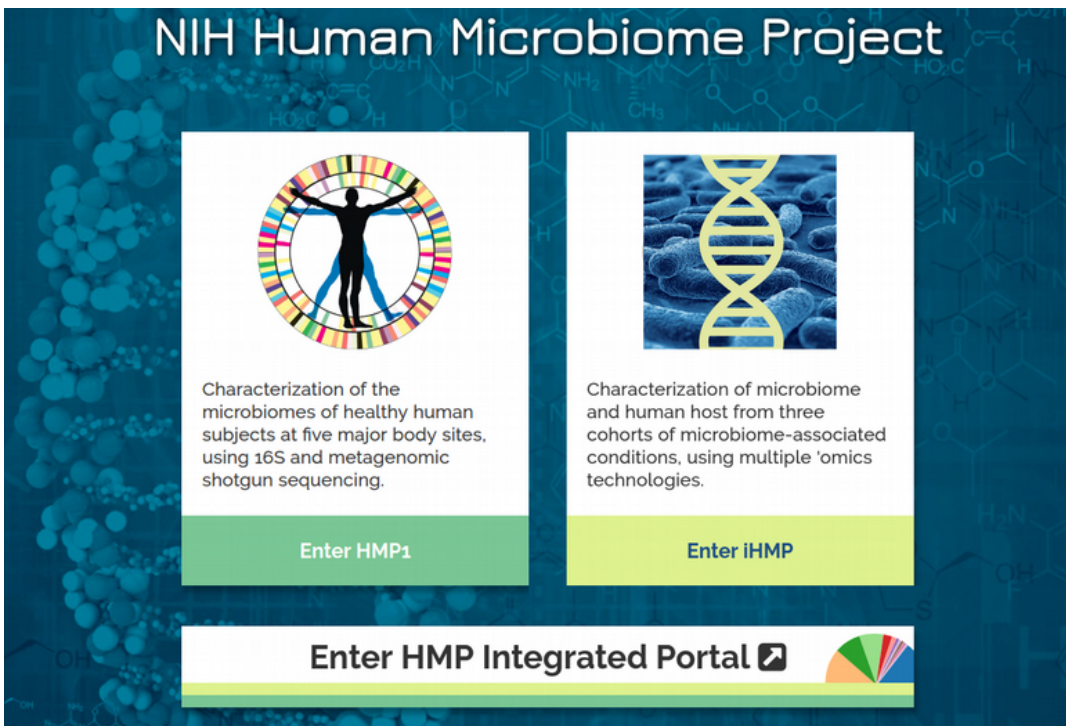
Convolutional Neural Networks (CNNs)

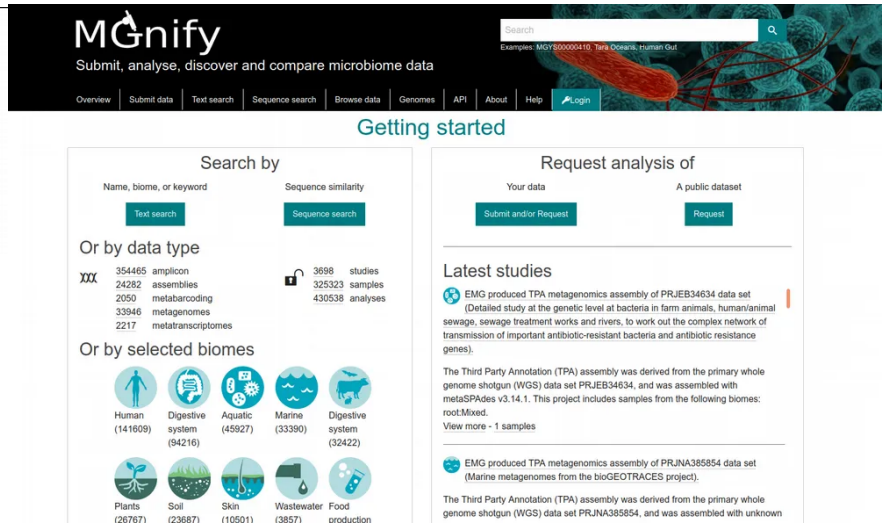


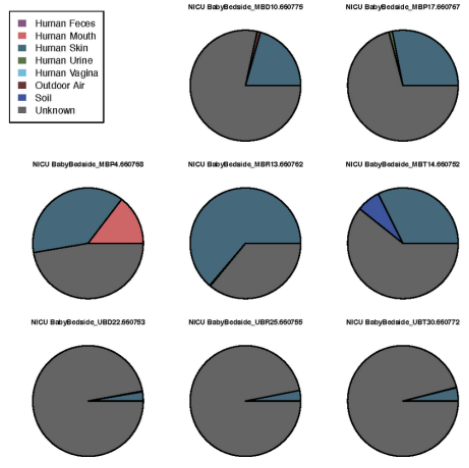
Angermueller et al., Molecular Systems Biology 2016

Deep Learning for Metagenomics

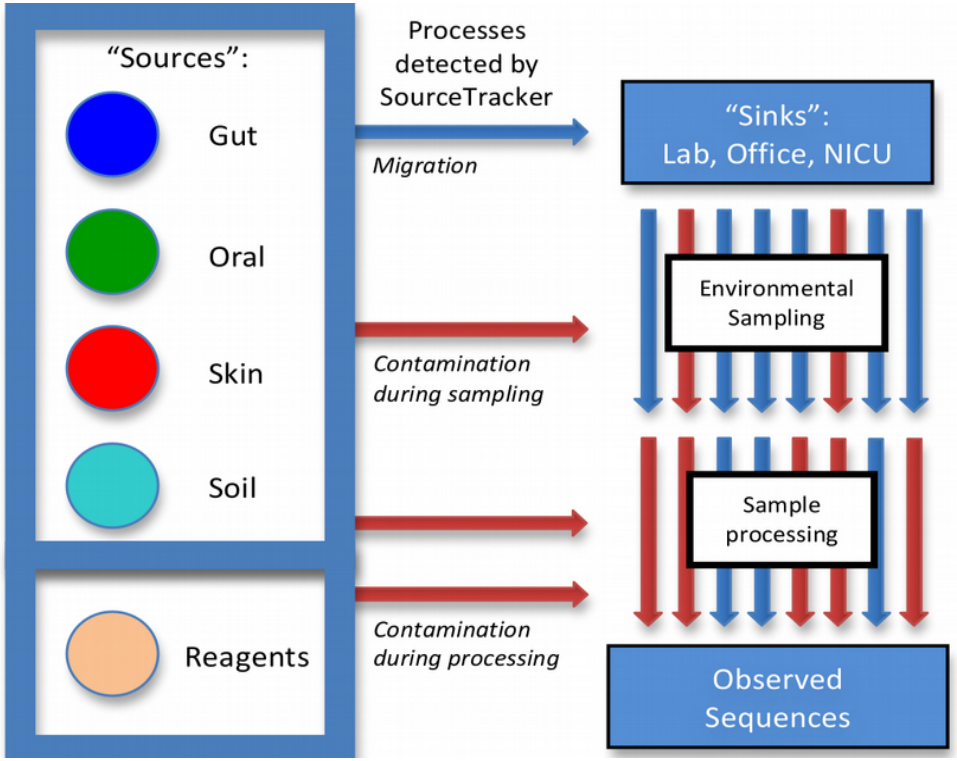
Microbial metagenomics Big Data



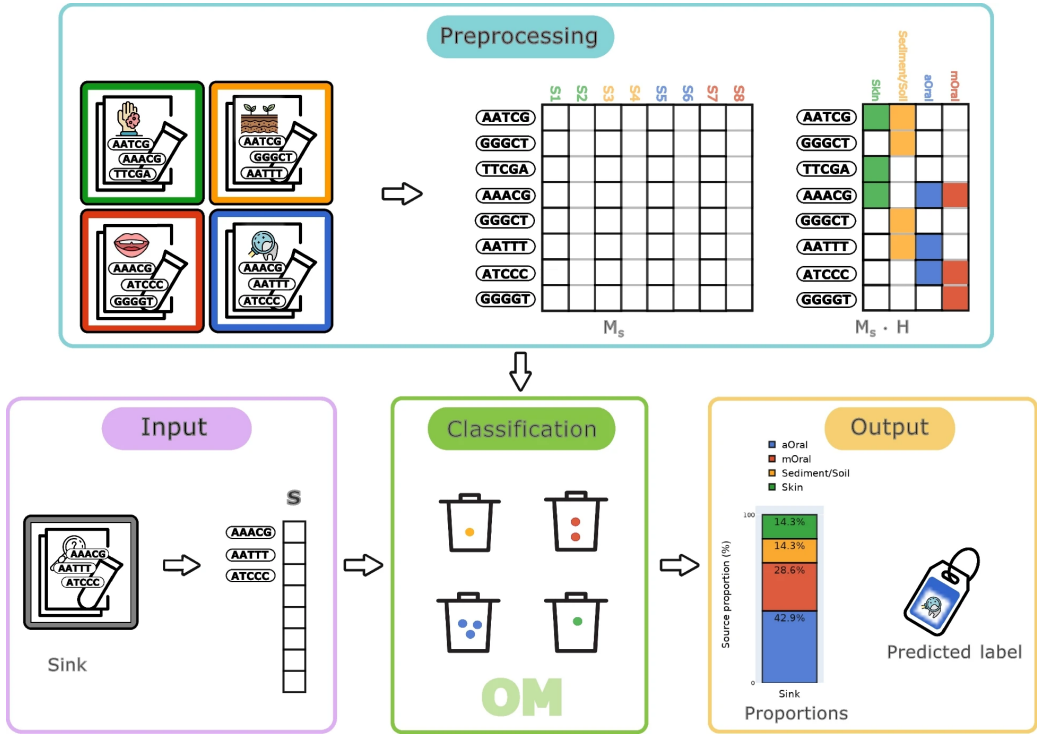




Metagenomics source prediction

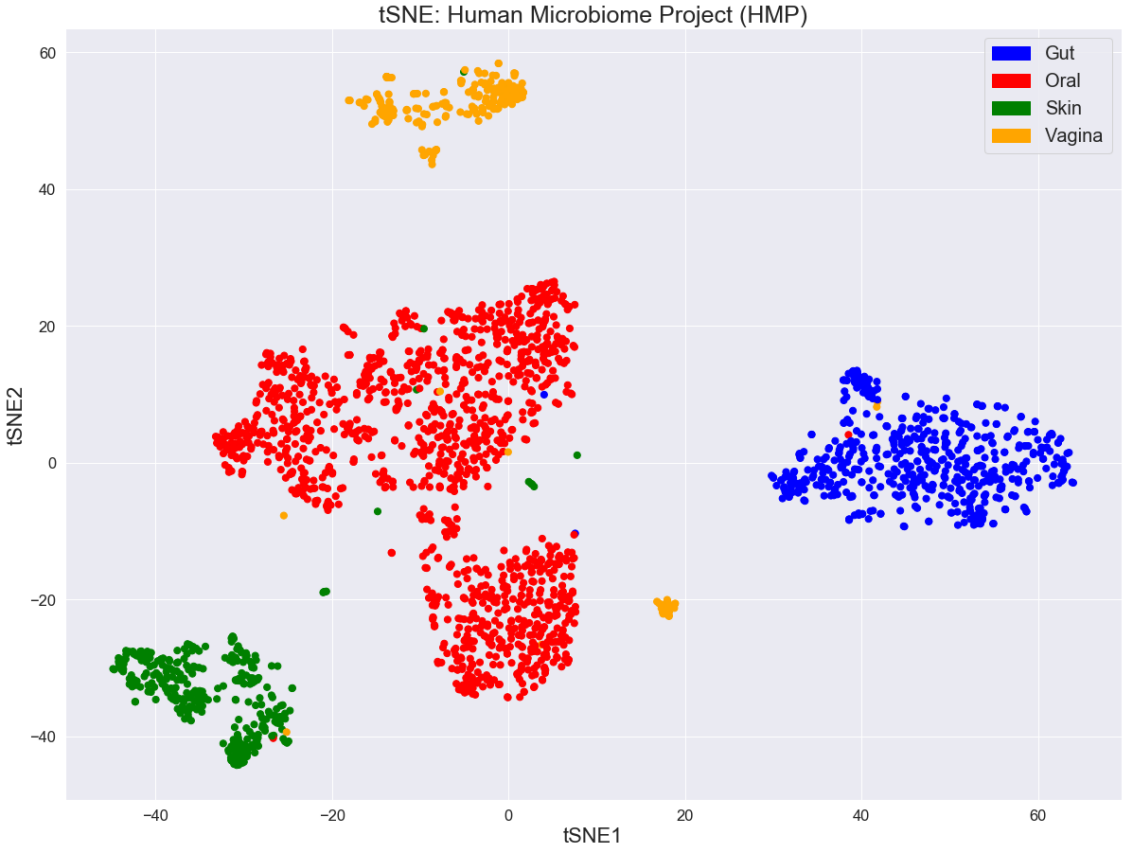


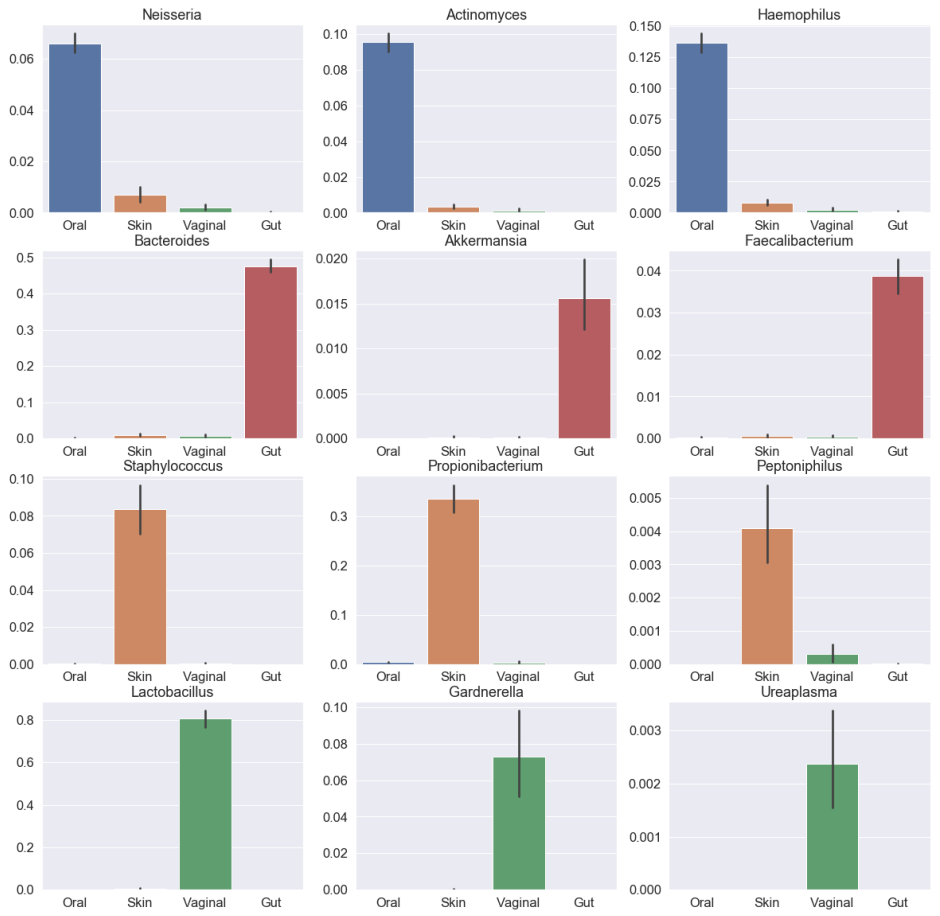
SourceTracker



deCOM

HMP visualization of sources and drivers

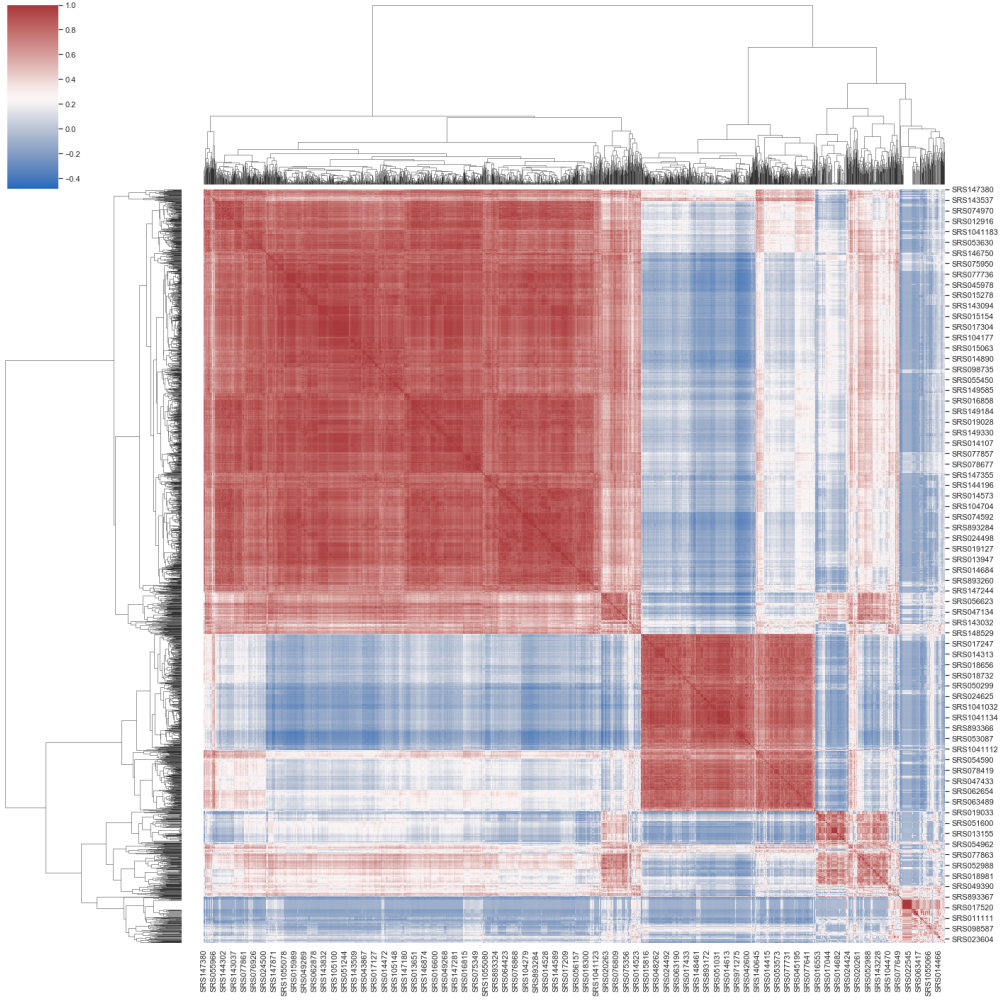




Can we use the environmental annotation for each sequence to train Deep Learning?

Distinct microbial communities in HMP

Clustering HMP samples by Spearman correlation distance



	Oral	Gut	Skin	Vagina
0	Neisseria	Blautia	Staphylococcus	Mobiluncus
1	Veillonella	Faecalibacterium	Peptoniphilus	Sphingopyxis
2	Actinomyces	Bacteroides	Citrobacter	Ureaplasma
3	Haemophilus	Dorea	Enhydrobacter	Caulobacter
4	Rothia	Akkermansia	Finegoldia	Gardnerella
5	Leptotrichia	Clostridium	Propionibacterium	Chlamydia
6	Cardiobacterium	Ruminococcus	Acinetobacter	Asticcacaulis
7	Capnocytophaga	Subdoligranulum	Massilia	Mycobacterium
8	Oribacterium	Oxalobacter	Hymenobacter	Herbaspirillum
9	Alloprevotella	Oscillibacter	Corynebacterium	Lactobacillus
10	Gemella	Eubacterium	Bacillus	Achromobacter
11	Fusobacterium	Bilophila	Micrococcus	Atopobium

Environment-specific genera

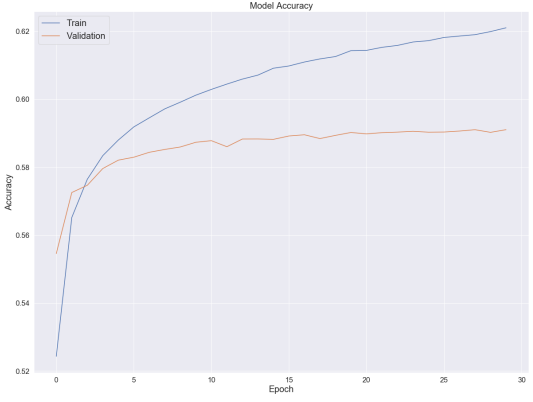
Distinct microbial communities in HMP

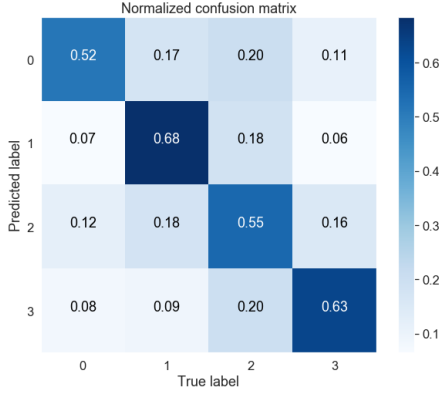
Labeling and one-hot-encoding each sequence

	SEQ	LAB
0	GCACGACATTTACAGCTTACCGCCACGTTTTATCCGCAATTTGAGTGCGATGGTCAGGCAGGACGATGTTTTATAGTGGA	0
1	CAGCCTGAAAAATTCACATACCTTGCAAGCAGGAAAACGCATTCGGATTAAAGCAAATAACCTTGATAATGCAGTACAAG	0
2	GCCAAATACCCGTCCCCTTACCATTCCCGAAATATCCGTTCAAACTTGGAGCAGCGTTACGGCAAAGAAAACATCACCTC	0
3	GGCGGGGGGATGCCACGCCGGCAGCGCCGTTGTCGGCGGGCCATTTGCCGTTATTCAGGTAATACCCGGTAACGGCTGATT	0
4	AATCCCAACGTACCGGCTGATATTCTTCGCGCTTGGGGTGTTTTGGTTCAGGTCTATCAAGTAAGCCAGGCCACCCGCGT	0
...	...	...
12971523	CCATGGTTGCCCCGCCCCCGCCAACGCCAATCACTCAAGTCCAACAGGGCGGTGACGGTGGCGCCGGGGGCACCGGATCC	3
12971524	GAGCGGAGCACATGACACGATACGACTCGCTGTTGCAGGCCTTGGGCAACACGCCGCTGGTTGGCCTGCAGCGATTGTCG	3
12971525	TCGACGGGGGTGGCGTTCGCGTTCGGGCCGGGCGTCACTGTCGAAGGCATGCTGTTCGACATCATCCGACGGTGACCGCC	3
12971526	GCGAGGTCCGGAGCGGGGCGGAAATTTCATTGAACAGCCGTAGAAGTTCAGCCAGGACCGGAACGGATCCAGCGGCAAGC	3
12971527	AACTGGGCACGGCGCTTGACGCACCGGTGCTGATGCATCCCGGCGACGCCGTGCTGTGGCGAATGACTCACCCGGACAAA	3

12971528 rows × 2 columns

Training a CNN to recognize microbial community





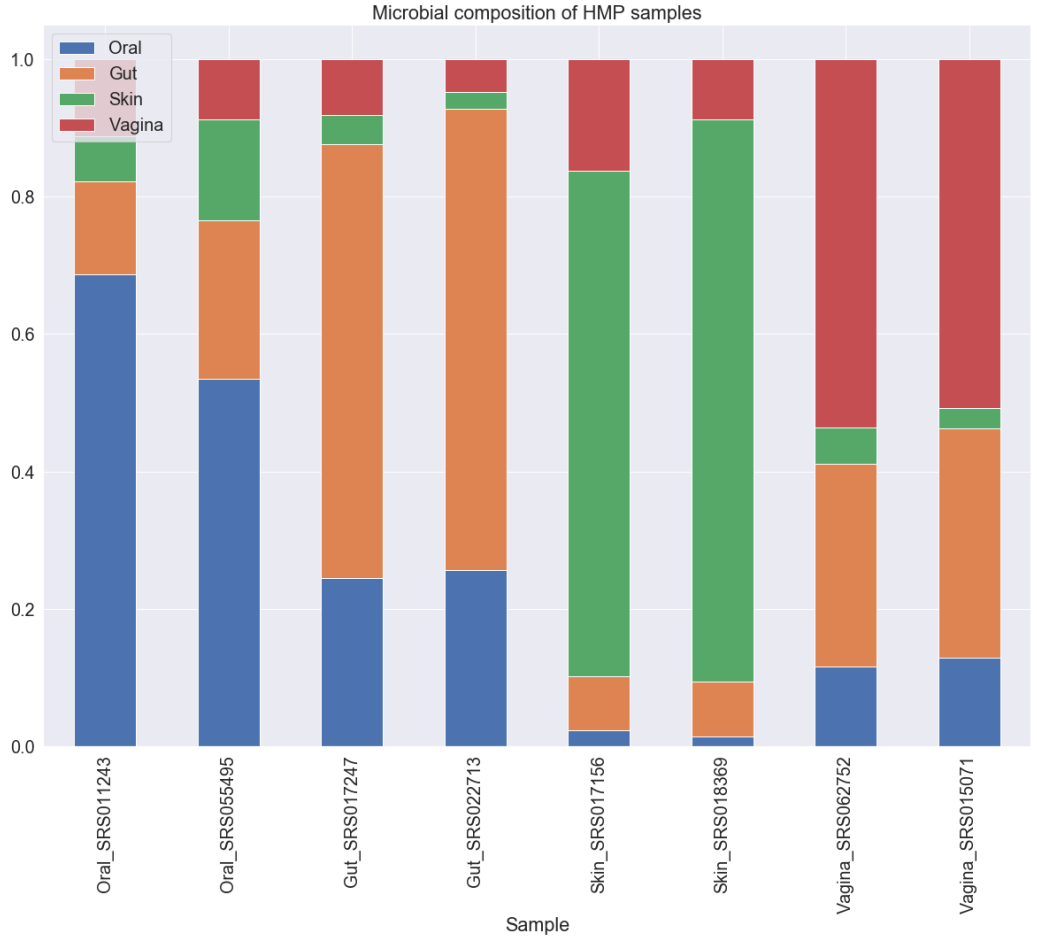
Training Convolutional Neural Network

A CNN learns to recognize patterns that are generally invariant across space, by trying to match the input sequence to a number of learnable "filters" of a fixed size. In our dataset, the filters will be motifs within the DNA sequences. The CNN may then learn to combine these filters to recognize a larger structure (e.g. the presence or absence of an ancient site on a sequence). We will start with defining Convolutional Neural Network (CNN) model and summarize the fitting parameters of the model.

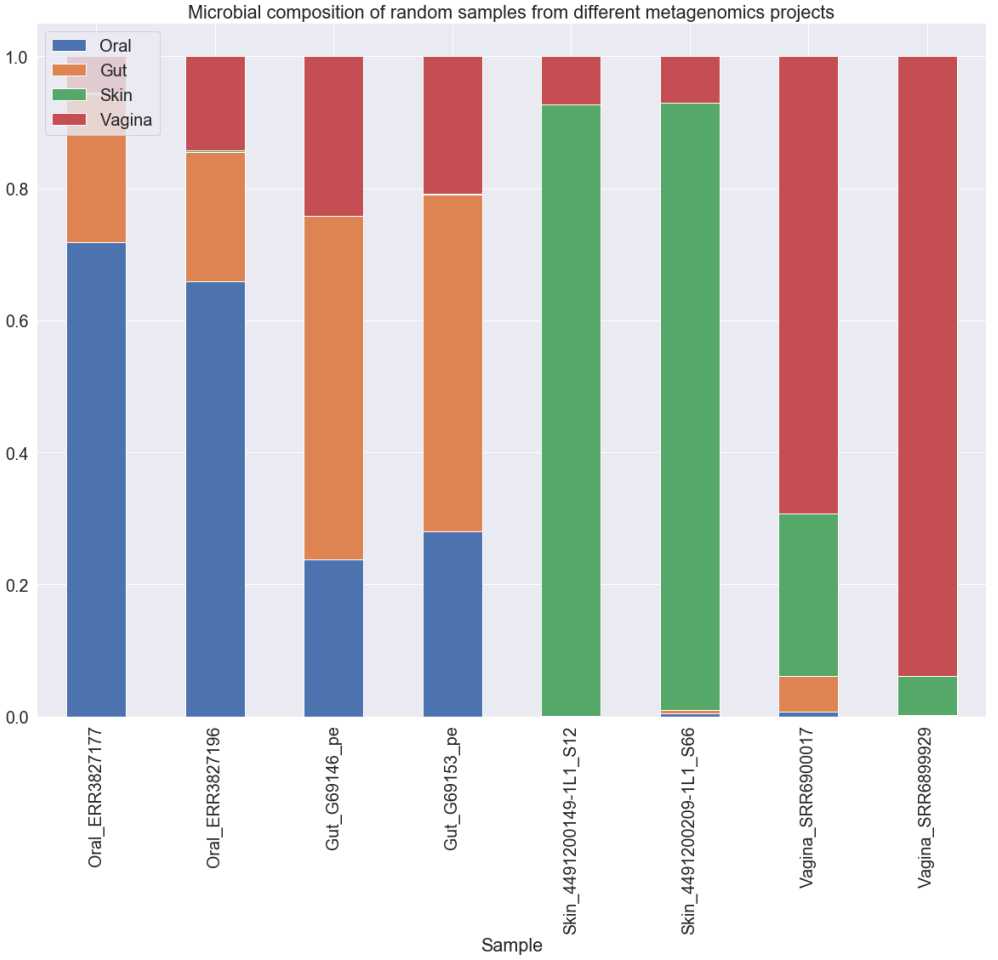
```
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.layers import Conv1D, Dense, MaxPooling1D, Flatten, Dropout
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Conv1D(filters = 512, kernel_size = 20, input_shape = (train_features.shape[1], 4),
                 padding = 'same', activation = 'relu'))
#model.add(MaxPooling1D(pool_size = 2))
#model.add(Dropout(0.3))
#model.add(Conv1D(filters = 512, kernel_size = 10, padding = 'same', activation = 'relu'))
model.add(MaxPooling1D(pool_size = 2))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.1))
model.add(Dense(4, activation = 'softmax'))

epochs = 30
lrate = 0.01
decay = lrate / epochs
sgd = SGD(lr = lrate, momentum = 0.9, decay = decay, nesterov = False)
model.compile(loss = 'categorical_crossentropy', optimizer = sgd, metrics = ['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 80, 512)           41472
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 40, 512)           0
_________________________________________________________________
dropout (Dropout)            (None, 40, 512)           0
_________________________________________________________________
flatten (Flatten)            (None, 20480)             0
_________________________________________________________________
dense (Dense)                (None, 256)               5243136
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 1028
=================================================================
Total params: 5,285,636
Trainable params: 5,285,636
Non-trainable params: 0
_________________________________________________________________
```

Model validation on independent data



Validation on different HMP samples



Validation on non-HMP samples (MGnify)

Session summary

Take home messages of the session:

1) One-hot-encoding is essential to train CNN on DNA

2) Deep Learning powerful for functional element detection

3) Metagenomics is a potential Big Data

4) Deep Learning successful for microbial source prediction

5) DNA is a text, which is suitable for NLP applications

National Bioinformatics Infrastructure Sweden (NBIS)

Knut och Alice Wallenbergs Stiftelse

Vetenskapsrådet

Lunds Universitet