NLP-MeTaxa: A Natural Language Processing approach for Metagenomic Taxonomic Binning based on deep learning

Supplementary material

# 1 Building Corpus

We built the corpus that contains all nucleotides concatenation possibilities for a given metagenomic dataset.

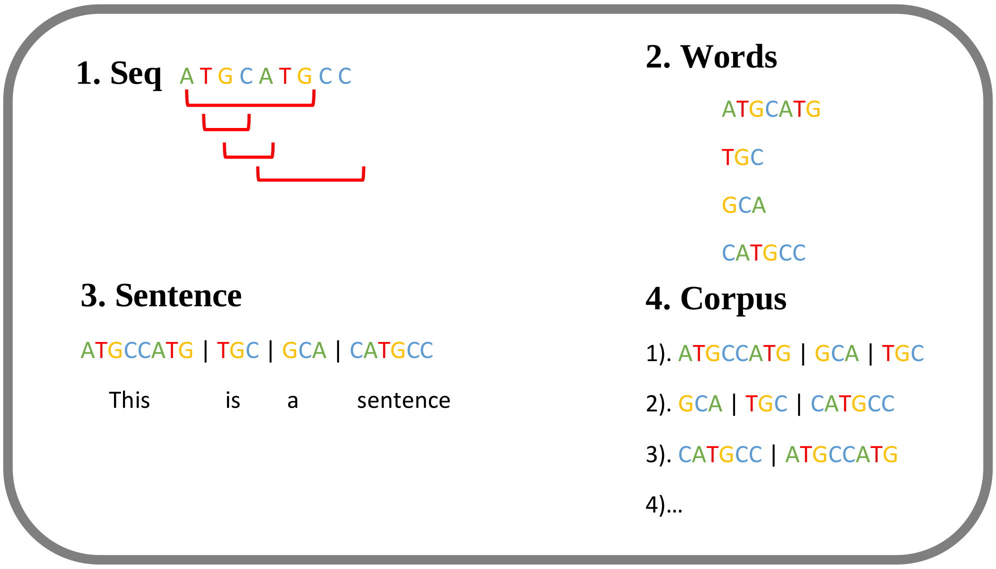

Figure S1: Building corpus

# 2 DataSets

The used datasets consist of three different simulated metagenomes. The first one contains a single sample dataset of low complexity community(20 circular elements and 40 genomes), the second dataset consists of differential abundance dataset with two samples of a medium complexity community(100 circular elements and 132 genomes) and the third metagenome is made up of two insert sizes and a time series dataset with five samples from a high complexity community (478 circular elements and 596 genomes).

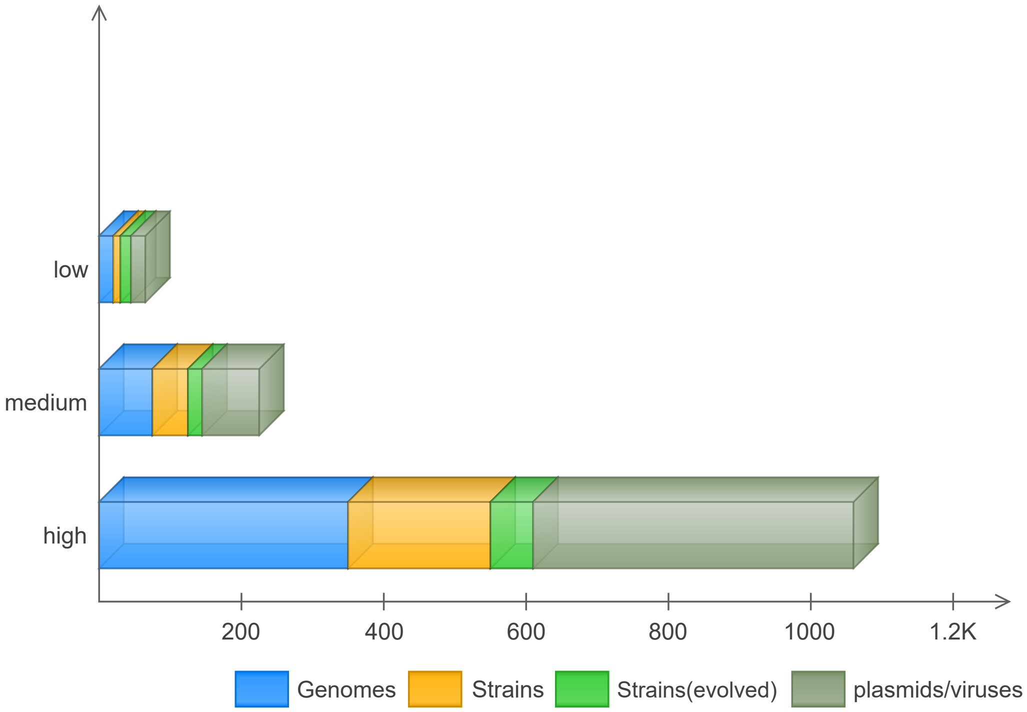

Figure S2: Number of genomes, plasmids, viruses and other circular elements in the datasets

# 3 Models parameter settings

## 3.1 NLP Model

To enhance the Word2vec training speed and quality, the following settings were used.

| Parameter | Value | Description |
|---|---|---|
| word-length-low | 1 | minimum word length |
| word-length-high | 8 | maximum word length |
| vec-dim | 100 | vector dimension size |
| epoch | 5 | training epoch number |
| context | 10 | set of adjacent words surrounding the targeted word |

Table S1: embedding model tuning parameters

## 3.2 CNN model

Many trials of model configuration were tested to build the CNN model. The best configuration was taken in the degree of complexity and the degree of performance

| Layer Type | Output | Number of Parameters |
|---|---|---|
| ***conv2d_1 (Conv2D)*** | (None, 10, 10, 32) | 832 |
| ***max_pooling2d_1 (MaxPooling2 (None, 5, 5, 32))*** | 0 | |
| ***conv2d_2 (Conv2D)*** | (None, 5, 5, 64) | 18496 |
| ***conv2d_3 (Conv2D)*** | (None, 5, 5, 128) | 73856 |
| ***max_pooling2d_2 (MaxPooling2 (None, 2, 2, 128))*** | 0 | |
| ***conv2d_4 (Conv2D)*** | (None, 1, 1, 256) | 131328 |
| ***max_pooling2d_3*** | (MaxPooling2 (None, 1, 1, 256) | 0 |
| ***flatten_1 (Flatten)*** | (None, 256) | 0 |
| ***dense_1 (Dense)*** | (None, 608) | 156256 |
| ***dropout_1 (Dropout)*** | (None, 608) | 0 |
| ***dense_2 (Dense)*** | (None, 608) | 370272 |
| ***dropout_2 (Dropout)*** | (None, 608) | 0 |
| ***dense_3 (Dense)*** | (None, 608) | 370272 |
| ***dropout_3 (Dropout)*** | (None, 608) | 0 |
| ***dense_4 (Dense)*** | (None, Numbers of Label) | 17052 |

Table S2: CNN parameters settings

# 4 Word length

Each metagenomic DNA fragment was split into a set of non-overlapping words of length L which varies from 1 to 8. Then, for each length, the vector representation was calculated. Finally, we measured NLP-MeTaxa accuracy.

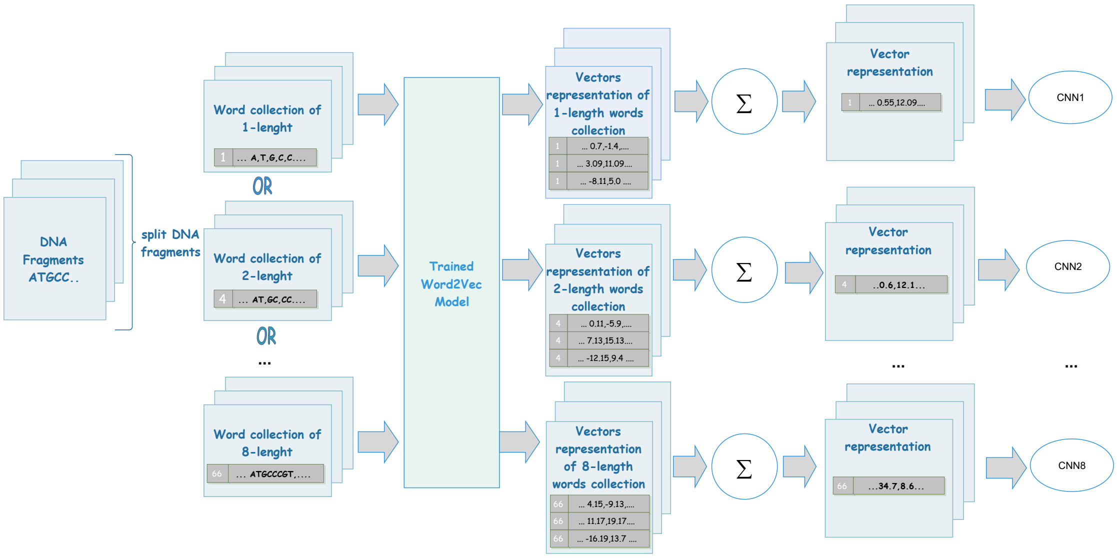

Figure S3: checking word length experiment

## 5 Overall metrics

NLP-MeTaxa performance was compared to other tools results obtained in the first CAMI challenge.

| Dataset | Tool | Precision % | Recall % |
|---|---|---|---|
| Low | **NLP-MeTaxa** | **89** | **87** |
| | taxator-tk | 76 | 8 |
| | MEGAN | 67 | 0.1 |
| | PhyloPythiaS+ | 74 | 22 |
| | Kraken | 86 | 2 |
| Medium | **NLP-MeTaxa** | 69 | **75** |
| | taxator-tk | 80 | 4 |
| | MEGAN | 71 | 0.5 |
| | PhyloPythiaS+ | 68 | 29 |
| | Kraken | **84** | 6 |
| High | **NLP-MeTaxa** | 66 | **75** |
| | taxator-tk | 71 | 1 |
| | MEGAN | 41 | 0.7 |
| | PhyloPythiaS+ | 67 | 30 |
| | Kraken | **73** | 10 |

Table S3: Overall precision and recall comparison across the three datasets

## 6 Data distribution in the three datasets

To investigate NLP-MeTaxa performance we needed to know data distribution across the three datasets.

| Dataset | Rank | Number |
|---|---|---|
| Low | species | 7417 |
|  | genus | 8293 |
|  | order | 93 |
|  | superkingdom | 20 |
|  | no rank | 3676 |
| Medium | species | 24764 |
|  | genus | 21282 |
|  | family | 621 |
|  | order | 67 |
|  | superkingdom | 29 |
|  | no rank | 16684 |
| High | species | 12350 |
|  | genus | 18692 |
|  | family | 1587 |
|  | no rank | 9409 |

Table S4: Labeled data distribution in training datasets