Classification for Bioinformatics Data

Manu Madhavan

Lecture 16

Recap

- ANN and Bioinformatics

Outline

- Supervised Algorithms: Classification
- Features
- Evaluation
- Applications

Outline

- Supervised Algorithms: Classification
- Features
- Evaluation
- Applications

Sequence Encoding

- Label Encoding

```
dna_sequence_string =
“ATATATCCCGGGAATTTTCGTAGTTAGGCTGATTTTATTGGCGCGAAAATTTTTT”
dna_np_array = PyDNA.dna_sequence_np_array(dna_sequence_string)
dna_label_encoder = PyDNA.dna_label_encoder(dna_np_array)
print(“DNA sequence string:\n{}”.format(dna_sequence_string))
print(“DNA NumPy array:\n{}”.format(dna_np_array))
print(“Custom Label Encoding:\n{}”.format(dna_label_encoder))
```

Results:

```
DNA sequence string:
ATATATCCCGGGAATTTTCGTAGTTAGGCTGATTTTATTGGCGCGAAAATTTTTT
DNA NumPy array:
['a' 't' 'a' 't' 'a' 't' 'c' 'c' 'c' 'g' 'g' 'g' 'a' 'a' 't' 't' 't'
't' 'c' 'g' 't' 'a' 'g' 't' 't' 'a' 'g' 'g' 'c' 't' 'g' 'a' 't' 't'
't' 't' 'a' 't' 't' 'g' 'g' 'c' 'g' 'c' 'g' 'a' 'a' 'a' 'a' 't' 't'
't' 't' 't' 't']
Custom Label Encoding:
[0.25 1. 0.25 1. 0.25 1. 0.5 0.5 0.5 0.75 0.75 0.75 0.25 0.25 1. 1.
1. 1. 0.5 0.75 1. 0.25 0.75 1. 1. 0.25 0.75 0.75 0.5 1. 0.75 0.25 1.
1. 1. 1. 0.25 1. 1. 0.75 0.75 0.5 0.75 0.5 0.75 0.25 0.25 0.25 0.25
1. 1. 1. 1. 1. 1. ]
```

Sequence Encoding

- One-hot Encoding

```
DNA sequence string:
ATATATCCCGGGAATTTTCGTAGTTAGGCTGATTTTATTGGCGCGAAAATTTTTT
DNA NumPy array:
['a' 't' 'a' 't' 'a' 't' 'c' 'c' 'c' 'g' 'g' 'g' 'a' 'a' 't' 't' 't'
't' 'c' 'g' 't' 'a' 'g' 't' 't' 'a' 'g' 'g' 'c' 't' 'g' 'a' 't' 't'
't' 't' 'a' 't' 't' 'g' 'g' 'c' 'g' 'c' 'g' 'a' 'a' 'a' 'a' 't' 't'
't' 't' 't' 't']
DNA One-Hot Encoding with Scikit-Learn framework:
[[1. 0. 0. 0.] [0. 0. 0. 1.] [1. 0. 0. 0.] [0. 0. 0. 1.] [1. 0. 0.
0.] [0. 0. 0. 1.] [0. 1. 0. 0.] [0. 1. 0. 0.] [0. 1. 0. 0.] [0. 0. 1.
0.] [0. 0. 1. 0.] [0. 0. 1. 0.] [1. 0. 0. 0.] [1. 0. 0. 0.] [0. 0. 0.
1.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 1. 0. 0.] [0. 0. 1.
0.] [0. 0. 0. 1.] [1. 0. 0. 0.] [0. 0. 1. 0.] [0. 0. 0. 1.] [0. 0. 0.
1.] [1. 0. 0. 0.] [0. 0. 1. 0.] [0. 0. 1. 0.] [0. 1. 0. 0.] [0. 0. 0.
1.] [0. 0. 1. 0.] [1. 0. 0. 0.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 0.
1.] [0. 0. 0. 1.] [1. 0. 0. 0.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 1.
0.] [0. 0. 1. 0.] [0. 1. 0. 0.] [0. 0. 1. 0.] [0. 1. 0. 0.] [0. 0. 1.
0.] [1. 0. 0. 0.] [1. 0. 0. 0.] [1. 0. 0. 0.] [1. 0. 0. 0.] [0. 0. 0.
1.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 0. 1.] [0. 0. 0.
1.]]
```

Sequence Features

- Each sequence in the dataset was considered as a text document composed of A, C, U, and G alphabets
- k-mers (sub-strings of length k) can be considered as words in the document
- Each sequence is represented as bag-of-k-mers
- Weight of each k-mer is calculated using tf-idf method

TF-IDF computation

- $tf_{i,j} = \frac{count(k_i, s_j)}{max\{count(k, s_j) : \forall k \in s_j\}}$
- $idf_i = log \frac{N}{|\{s_j \in S : k_i \in s_j\}|}$
- $w_{ij} = tf_{ij} \times idf_i$

Illustration

$$
\begin{bmatrix} S \end{bmatrix} = \begin{matrix} s_1 \\ s_2 \\ \vdots \\ \\ \vdots \\ s_N \end{matrix} \begin{bmatrix} CGCCCGCAAUUCCCCCACGAGCCCUGGGGAGACCCAGCGCUAACCAGGGGUG \\ AGGAGCCGGGAGAGGCCCUCCUGGAGGUGGGCACAGCCCAGGCAAACAUCAG \\ \cdots\cdots\cdots \\ \vdots\vdots\vdots \\ UCCCCACUCUACCUGGAGCUGCUAGAAGGAGACG \\ ACCAGCAGAAAGACCACGAGCAUGGACAUGAUGGUCGCGCUCACUCCG \\ GUGCAGUGAGUGUCUGGGUGAUGGAUCCUAGAACCUGUU \end{bmatrix}
$$

k-mer extraction

$$
k-mer\ representation = \begin{matrix} s_1 \\ s_2 \\ \vdots \\ \\ \vdots \\ s_N \end{matrix} \begin{bmatrix} CGCC & GCCC & CCCG & CCGC & CGCA & GCAA & \cdots & \cdots & \cdots & GGGU \\ AGGA & GGAG & GAGC & AGCC & GCCG & CCGG & \cdots & \cdots & \cdots & UCAG \\ \cdots & \cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ UCCC & CCCC & CCCA & CCAC & CACU & ACUC & \cdots & \cdots & \cdots & GACG \\ ACCA & CCAG & CAGC & AGCA & GCAG & CAGA & \cdots & \cdots & \cdots & UCCG \\ GUGC & UGCA & GCAG & CAGU & AGUG & GUGA & \cdots & \cdots & \cdots & UGUU \end{bmatrix}
$$

$tf-idf$ computation

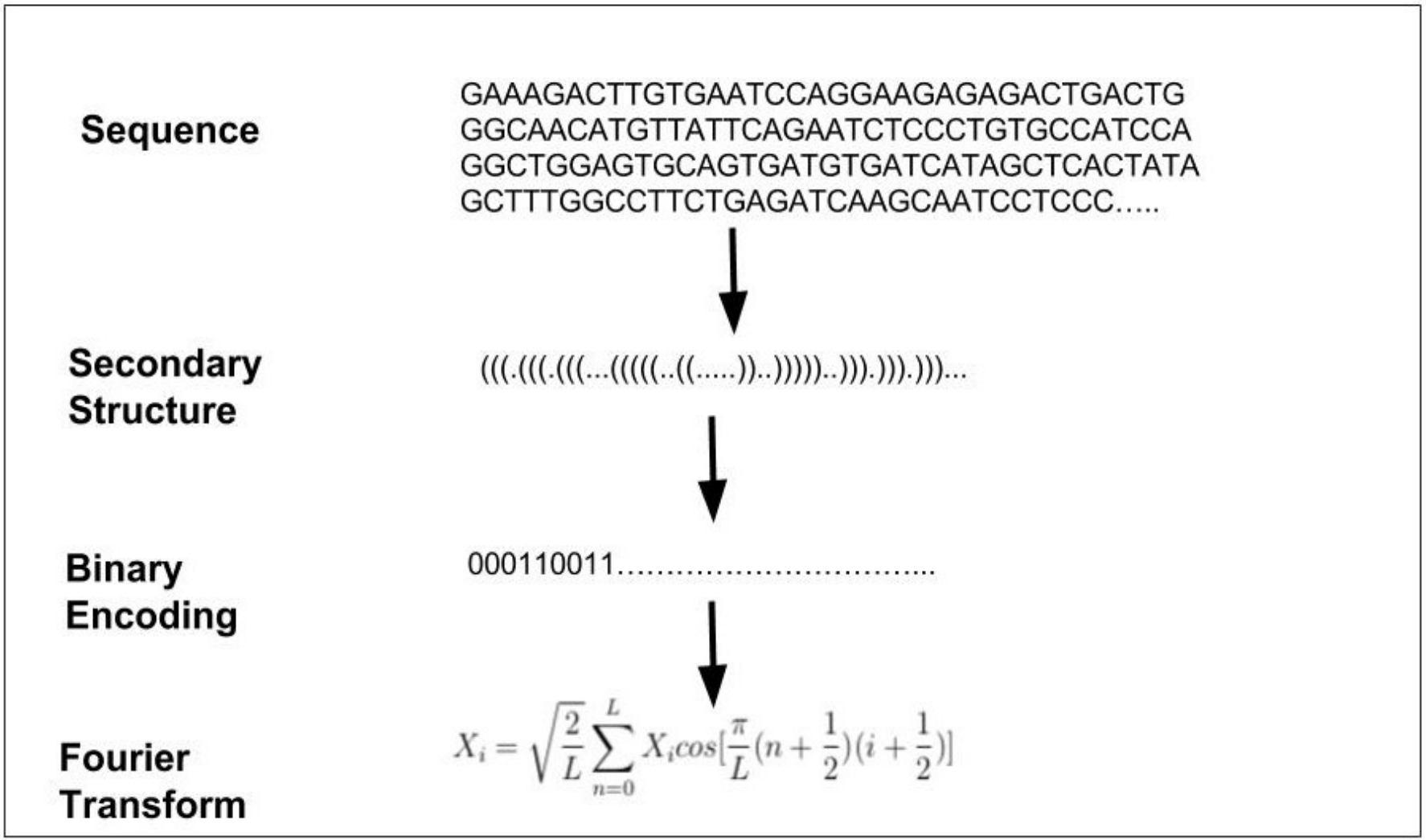
$$
sequence-term\ matrix \begin{matrix} \\ s_1 \\ s_2 \\ \vdots \\ \\ \vdots \\ \end{matrix} \begin{matrix} k_1 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} k_2 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} k_3 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} k_r \\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & \cdots & w_{1,r} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & \cdots & w_{2,r} \\ \cdots & \cdots & \cdots & \cdots & \cdots\cdots & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{N-2,1} & w_{N-2,2} & w_{2,3} & \cdots & \cdots & w_{N-2,r} \end{bmatrix}
$$

Structure Features

- Secondary structure is obtained from the primary sequence of RNA based on folds and pairs
- Represented by dot-bracket notation
- Replace dots and brackets with 0's and 1's
- Apply Fourier transformation on binary string

$$X_i = \sqrt{\frac{2}{L}} \sum_{n=0}^{L} X_i cos[\frac{\pi}{L}(n + \frac{1}{2})(i + \frac{1}{2})]$$

Illustration



GC Content

- GC Content is the percentage nitrogenous bases that are either guanine(G) or cytosine(C) from a possible four different bases (G, C, Adenine(A), and Uracil(U)).
- $GC\_Content = \frac{C(G)+C(C)}{C(A)+C(C)+C(G)+C(U)}$
- Recent studies revealed that GC content of lncRNA is low compared with that of coding RNAs

Molecular Weight

- Mass of a molecule, measured by summing the atomic weights of each element multiplied by the count of atoms of that element in the molecular formula
- It is observed that lncRNA has a high molecular weight compared with mRNAs
- The molecular weight compounds also help to control the folding and looping of the lncRNA sequence

Other features

- Sequence length, ORF length, etc
- Secondary structure related
- Interaction with other molecule
- Expression and co-expression details

General steps

- Data collection (from various sources)
- Pre-processing
- Feature selection
- Dimensionality Reduction (optional)
- Classification
- Evaluation
- Result analysis (for Biological significance)

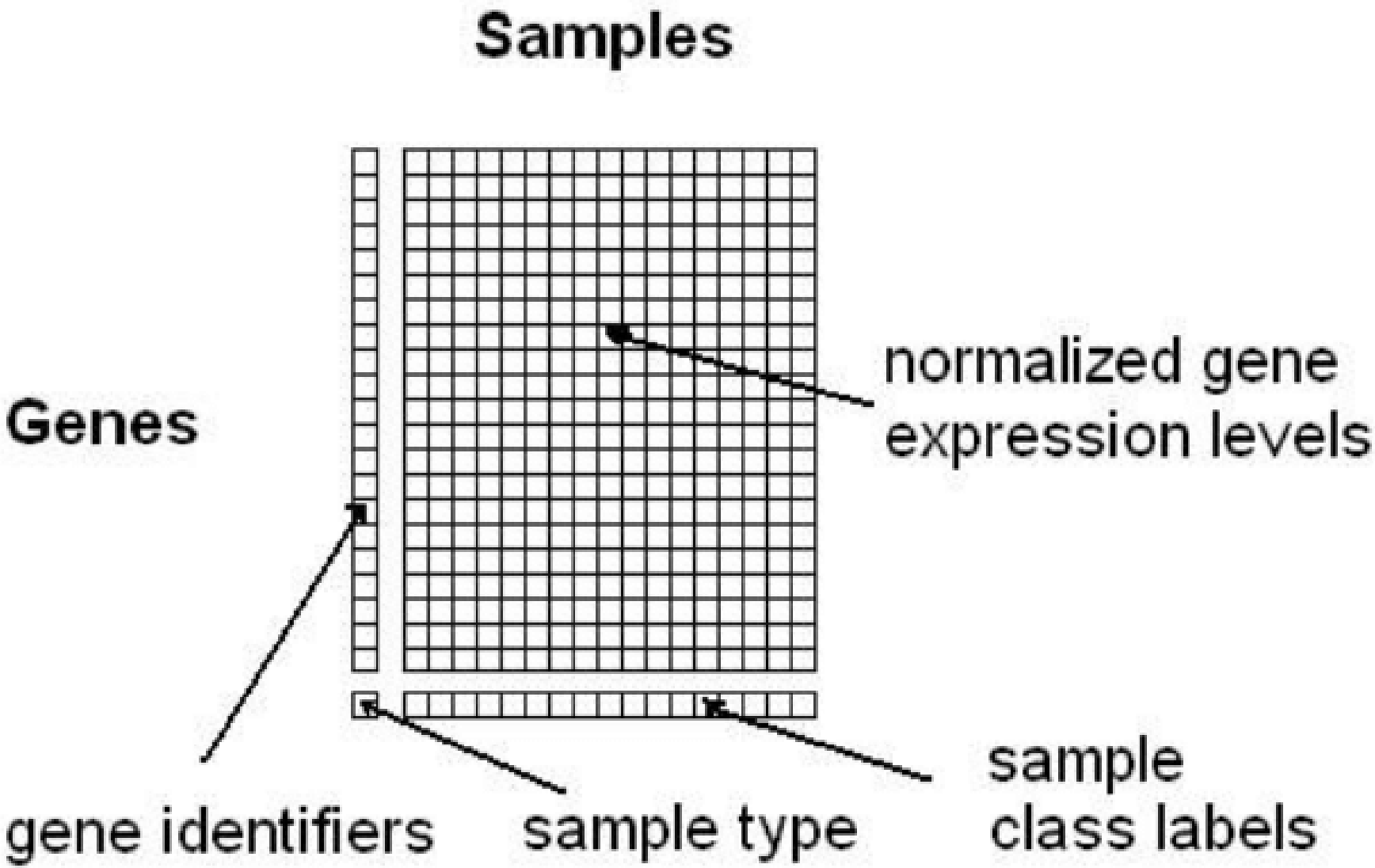
Classification Algorithm

- KNN
- Naive Bayes
- Random Forest
- XGBoost
- SVM
- Neural Networks

Evaluation Metrics

- Precision-Recall
- ROC-AUC
- AUPR
- Accuracy
- Statistical tests and Cross validations

Gene Expression Analysis



Other Applications

- Gene Function Prediction
- Gene-protein interaction prediction
- Protein-protein interaction
- Coding and non-coding gene classification
- Gene-disease association prediction

Next

- Clustering in Bioinformatics