PROGRAMMING3 BIG DATA

Supervised machine learning to predict protein function on InterPROscan datasets.

18TH OF SEPTEMBER

DAAN STEUR
Hanze university of applied sciences

Introduction

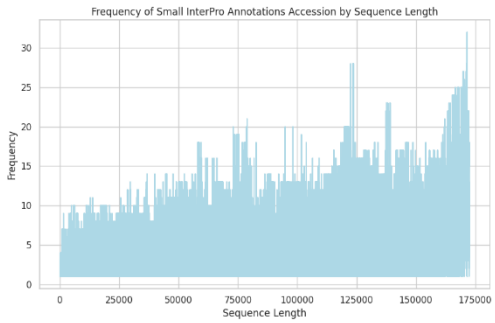
In the world of microbiology, exploration of new species and the functions of those species can be time consuming. Let say a new species is discovered and you want to know the specific protein functions inside that species. To get to that answer, the bacteria need to be grown, induced, cells need to be extracted, protein needs to be extracted and purified, that protein eventually needs to be sequenced and you in the end you align its DNA against tools like BLAST, to see if any useful results pop up. How wonderfully refreshing it would be to simply this by a large number of steps. You grow the bacteria, you sequence the bacteria, you run a model over that data and a list potential active proteins and the functions of those proteins gets printed onto a csv. The problem however is not the lack of data, the problem is the lack of accurate predictive models. In this report an attempt is made using supervised learning techniques to generate an accurate model to predict the protein function based on feature characteristics.

Materials & method

This project aims to create and train a machine learning model to predict InterPro annotations of small proteins.

Data Processing

this project PySpark to construct a structured DataFrame from the designated TSV file. The data preprocessing phase unfolds in several steps. To begin with that row marked with "-" for InterPro annotations, signifying the absence of annotations, are filtered out. A ratio is calculated, representing the proportion of the sequence analysed compared to the entire sequence length. This ratio serves to differentiate between small and large InterPro annotations, using a threshold of 0.9. Proteins featuring both small and large InterPro annotations are identified, ensuring that they have at least one of each. Distinct Data Frames are crafted to accommodate small and large InterPro annotations, with InterPro accessions aggregated and counted. For proteins with multiple annotations, the most significant InterPro annotation is retained. Extraneous columns are pruned for efficiency.



Machine Learning Data Preparation

The code orchestrates the creation of a machine learning (ML) DataFrame, a pivotal step in the training process. This ML DataFrame has both small and large InterPro annotations, with features meticulously assembled using Vector Assembler because of the large string components. Concurrently, the target variable is indexed using String Indexer for optimal readability. The data undergoes division into training and testing datasets, with a default ratio of 70% for training and 30% for testing. The user reserves the flexibility to modify this partition ratio as needed. Training data is stored as a csv file at “/students/2021-2022/master/DaanSteur_DSLS”.

Model Training and Evaluation

Subsequently, the training and evaluation of a Naive Bayes classifier, in previous papers it was shown to be an effective model for this task. Model training is executed employing the Naive Bayes algorithm, encompassing various smoothing hyperparameters. Cross-validation, a pivotal technique, is harnessed to select the optimal model, discerned primarily through accuracy metrics. An initial model, devoid of cross-validation, is trained to establish a baseline accuracy reference. The script diligently records model accuracies and the duration of the training process, storing these metrics in text files. The trained models are meticulously serialized and preserved in pickled format, stored at “/students/2021-2022/master/DaanSteur_DSLS”.

Results

In the original dataset, we started with a little over 4 million entries. After applying preprocessing steps, the dataset had a final size of 1.198.607 entries. This refined dataset includes 413.683 large InterPro annotations accession and 784.924 small InterPro annotations accession. For the machine learning phase, we split the dataset into training and testing sets using a 70/30 split, resulting in 121.064 rows in the training data and 51.524 rows in the testing data.

We evaluated model types of the naïve bayes machine learning models to predict InterPro annotations. we included different variants of Naïve Bayes, namely Naïve Bayes (Multinomial), Naïve Bayes (Gaussian), and Naïve Bayes (Bernoulli).

Model	Naïve bayes (multinomial)	Naïve bayes (gaussian)	Naïve bayes (Bernoulli)
Accuracy	0.68	0.61	Not suitable method
Accuracy parameter tuned	0.76	0.73	Not suitable method
Time (Hours)	4.24	5.36	Not suitable method

Please note that the Naïve Bayes (Bernoulli) model results are not available due to it not being found to be suitable type for multi class classification. These results demonstrate the effectiveness of the Naïve Bayes Multinomial model, in counterpart to gaussian model, in predicting InterPro annotations, even after hyperparameter tuning.

Conclusion

Our investigation focused on the efficacy of different Naïve Bayes models for predicting InterPro annotations. We explored various variants, including Naïve Bayes (Multinomial), Naïve Bayes (Gaussian), and Naïve Bayes (Bernoulli). The results revealed that the Naïve Bayes (Bernoulli) model was unsuitable for multi-class classification, leading to unavailable accuracy metrics. Importantly, our analysis demonstrated that the Naïve Bayes (Multinomial) model outperformed the Naïve Bayes (Gaussian) model in predicting small InterPro annotations. While the accuracy achieved with the Multinomial model was not as prominent as earlier mentioned models, it is still a suitability candidate for the task at hand.

Output directories

Interpro dataset: "/data/dataprocessing/interproscan/all_bacilli.tsv"

Training data: '/students/2021-2022/master/DaanSteur_DSLS/train_data.csv'

Testing data: '/students/2021-2022/master/DaanSteur_DSLS/test_data.csv'

Pyspark logbook: '/students/2021-2022/master/DaanSteur_DSLS/ spark_logs.txt'

Naive bayes (Multinomial) results: "/students/2021-2022/master/DaanSteur_DSLS/nb_multinomial_cv_results.txt"

Naive bayes (Multinomial) model: '/students/2021-2022/master/DaanSteur_DSLS/nb_model_multinomial.pkl'

Naive bayes (Multinomial) tuned model: '/students/2021-2022/master/DaanSteur_DSLS/cv_model_multinomial.pkl'

Naive bayes (Gaussian) results: '/students/2021-2022/master/DaanSteur_DSLS/nb_gaussian_cv_results.txt'

Naive bayes (Gaussian) model: '/students/2021-2022/master/DaanSteur_DSLS/nb_model_gaussian.pkl'

Naive bayes (Gaussian) tuned model: '/students/2021-2022/master/DaanSteur_DSLS/cv_model_gaussian.pkl'