Protein Function Prediction on the PATRIC Dataset

Neil Getty
Argonne National Laboratory
Illinois Institute of Technology
Email: ngetty@anl.gov

Fangfang Xia
Argonne National Laboratory
University of Chicago
Email: fangfang@anl.gov

***Abstract*—The ever-falling cost of genetic sequencing and the resulting abundant data has enabled genomics tasks to be done automatically and accurately. Gene annotation is one such task concerned with what specific genes do. Genes code for proteins which serve many general purposes such as catalysis, transport and regulation. These functions are determined by the physical structure of the proteins, which is directly coded by genes. In this paper, we describe the design and experimentation of machine learning models trained for the task of protein function prediction. We focus on one such algorithm, LightGBM, for its high accuracy and efficiency.**

I. Introduction

A. Motivation

Thanks to faster sequencing techniques there has been an explosion of genomics data. The goal of this work was simple, a faster, simpler and modern genome annotation approach while maintaining or outperforming prediction accuracy of traditional methods. Secondly was the integration with this problem with RAN, a remote memory pool.

B. Background

1) Protein Function: Amino acid sequences determine a proteins structure. Intuitively, proteins with very similar sequences will often code for similar protein structures and ultimately function. Due to gaps in sequencing as well as insertions, deletions and mutations between organism's genes, these underlying similarities may not be so straightforward. Traditional methods such as BLAST/RAST sequence alignment and profile scanning may be slow and require several steps. These methods may be incapable of learning very small differences between similar sequences that may map to similar but separate functions.

2) Data: Patric [1] is a bacterial genome database of over 100K genomes and a suite of tools including assembly and annotation.

Dataset	Classes	Examples	Avg Seq Length
Small	100	14862	1237
Core	1000	488626	1066

The above table reports the characteristics of the datasets used. The coreseed dataset is considered cleaner, with true labels the result of experimentation and not automatically generated.

II. Related Work

There are numerous other works on gene annotation, most focusing on the multi-label tasks concerning go-term or fun-cat annotation taxonomies. Such tasks relate genes with hierarchical term associations, i.e. terms have relationships and depth corresponding to levels of specificity.

Rifaioglu et. al. [2] developed a hierarchical aware deep-learning system trained on the Uniprot/Swiss-prot dataset with manual/experimental evidence. The authors propagates go term association via positive relationships to associate more terms per protein. They divide terms by level in the go directed-acyclic-graph and by number of protein associations to create 269 models for each ontology (cellular, molecular, biological). Sequences are featurized by using SPMAP, which clusters subsequences based on common associations.

III. Design

A. Traditional and Baseline Methods

The multiple sequence alighment program, MAFFT [3] may be used in conjunction with the HMMER [4] tool, which generates hidden markov model based profiles given aligned sequences of a particular class. The workflow is thus:

Algorithm 1 Align and Profile

1: Convert all sequences to fasta representations
2: Create an alignment of all sequences for each functional class separately using MAFFT
3: Create a HMM profile for each alignment using hmmbuild
4: Conduct inference on each test fasta sequence using hmmscan

B. Kmer Featurization

Current featurization techniques use subsequence clustering techniques or presence of unique subsequences. Our technique aggregates kmer counts in parallel for all sequences. This results in many more data points than a unique subsequence technique or SPMap (clustering). The resulting dataset is potentially muchlarger and more sparse yet holds more fine-grained information. Any subsequence featurization method loses locale information, though this information is more present as kmer size increases.

C. LightGBM

LightGBM (Gradient Boosted Machine) [5] is an efficient algorithm that uses an ensemble of gradient boosted weak learner decision trees to make predictions. The framework was developed by Microsoft as a supposed successor to

XGBoost [6], another gradient favorite of online machine learning competitions such as those found at Kaggle. The developers experimentally purport faster training speed, higher efficiency, lower memory usage, and on-par or better accuracy. The algorithm may be parallelized or trained using a GPU implicitly. Parallelization experiments evidence a near-linear speed-up.

The algorithm works by training many decision trees (learners) on subsets of features and/or examples. These learner's loss is minimized using the differential computed gradient, allowing convergence of many learners at once, reducing potential for a sub-optimal solution as a single learner may be more prone to come by.

Decision trees grow by leaf, not level, allowing faster and more efficent tree growth. The algorithm may bin continuous features as well as combine mutually exclusive features to save memory. To reduce communication overhead, all workers are provided with a full copy of the dataset, trading higher memory costs for computational performance. An additional improvement is the use of reduce scatter to merge only different features among workers, further reducing communication costs.

IV. Results

A. Function Prediction

In this section we report results using the described methods.

TABLE I: Sequence Alignment

Dataset	Train Acc	Val Acc	Top 5 Val Acc
Coreseed (1000 class)	0.967	0.963	0.979
Small (100 class)	0.991	0.992	0.994

Results obtained using sequence alignment and HMM profiles

TABLE II: LGBM Model

Dataset	Train Acc	Val Acc	Top 5 Val Acc
Coreseed (1000 class)	0.999	0.964	0.987
Small (100 class)	0.999	0.975	0.991

Results obtained training LGBM model with amino acid 1,2,3,4mers and nucleotide 1,3mers.

As shown above, the LightGBM model matches or outperforms the results of the other methods for both training and validation accuracy on the coreseed dataset. The fact that the sequence alignment method outperforms on the smaller dataset is expected, as some of the "true" labels for that set were obtained not through experiment but were automatically generated.

Table III and IV show the scope of the problem in terms of computation time and memory (in terms of known feature dimensionality. Accuracy results using LGBM are reported for the different combinations of features. The model is able to accept and learn from more data well, and is seemingly resilient to what might be noise.

TABLE III: Combinatorial Feature Testing

# of Features	Train Acc	Val Acc	Top 5 Val Acc	Time (sec)
20	0.997	0.832	0.947	34
400	0.997	0.921	0.969	50
8000	0.992	0.953	0.975	133
160000	0.968	0.948	0.975	168
420	1	0.935	0.978	46
8420	0.999	0.962	0.987	159
168420	0.999	0.975	0.991	426

Amino acid kmer featurization testing for small protein sequence and function dataset. Rows show datasets constructed using different combinations of kmer sizes. Growth is exponential and yet the accuracy consistently improves with more data.

V. Conclusion

In this paper we tested the performance of state of the art and traditional methods on Patric bacterial gene annotation data. Specifically, we compared the results of the Light Gradient Boosting Machine framework with a traditional, oft in production sequence alignment and hidden-markov-model profiling approach. Data was featurized using kmers, k-length contiquous subsequence counts with no explicit feature decomposition, clustering or other techniques that would result in a potentially meaningful loss of information.

The accuracy of the ensemble learning framework proved to be on-par or to outperform the traditional method, with a much more streamlined development, tuning and testing. Such an approach would allow much easier retraining given new data, as well as quicker inference.

A. Future Work

1) Gene Annotation: There are several possible avenues for expanding the protein function annotation work. One simple enhancement would be to include downstream and upstream genetic data when training and testing the model, i.e. the genes contiguous to the target gene. While genes may code for proteins independently, they did not evolve independent of the organism and complete genetic structure.

Furthermore, gene annotation is not an isolated task but is one part understanding genes and organisms. Other tasks such as sequence assembly and gene prediction are closely related. Multi-task models may be designed via ensemble methods, shared-layers or transfer learning. To this end it may prove useful to improve on end-end deep learning implementations.

While tree-based methods proved useful for learning simply from subsequence counts, such featurization methods ignores locality information. and does so at a cost of exponentially increasing feature dimensionality. Combining ensemble methods trained on aggregate counts with deep-learning models trained on raw sequences with feedback, memory and attention may prove useful for both generalizable prediction accuracy as well as any multi-modal or multi-task efforts.

Another means of expansion is the integration of protein data labeled with go-terms. Go-terms are hierarchical and

connected functional descriptors. Utilizing additional datasets such as the UniprotKB/Swiss-prot may potentially offer a clearer picture of the protein function latent space, allowing predictive models to generalize better, potentially even to genes and proteins not seen before. Such a leap may prove vital for quickly understanding newly discovered or mutated organisms, or may even aide in genetic engineering or biohacking efforts.

ACKNOWLEDGMENT

This research has been funded in part and used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

REFERENCES

[1] A. Wattam, J. Davis, R. Assaf, S. Boisvert, T. Brettin, C. Bun, N. Conrad, E. Dietrich, T. Disz, J. Gabbard, S. Gerdes, C. Henry, R. Kenyon, D. Machi, C. Mao, E. Nordberg, G. Olsen, D. Murphy-Olson, R. Olson, R. Overbeek, B. Parrello, G. Pusch, M. Shukla, V. Vonstein, A. Warren, F. Xia, H. Yoo, and R. Stevens, "Patric, the bacterial bioinformatics database and analysis resource." *Nucleic Acids Res*, vol. 42, 2016. [Online]. Available: https://www.patricbrc.org/

[2] A. S. Rifaioglu, T. Doan, M. J. Martin, R. Cetin-Atalay, and M. V. Atalay, "Multi-task deep neural networks in automated protein function prediction," 2017.

[3] M. Katoh and M. Kuma, "Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform." *Nucleic Acids Res*, vol. 30, pp. 3059–3066, 2002.

[4] S. Eddy and T. Wheeler. Biological sequence analysis using profile hidden markov models. [Online]. Available: http://hmmer.org/

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye1, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[6] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[7] P. S. Foundation. Python language reference, version 2.7. [Online]. Available: https://docs.python.org

[8] G. Valentini, "Hierarchical ensemble methods for protein function prediction," *ISRN Bioinform*, vol. 2014, p. 901419, 2014.

[9] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Guldener, G. Mannhaupt, M. Munsterkotter, and H. W. Mewes, "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes," *Nucleic Acids Res.*, vol. 32, pp. 5539–5545, 2004.

[10] M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, and R. White, "The Gene Ontology (GO) database and informatics resource," *Nucleic Acids Res.*, vol. 32, pp. 258–261, 2004.

[11] I. Friedberg, "Automated protein function predictionthe genomic challenge," *Briefings in Bioinformatics*, vol. 7, no. 3, pp. 225–242, 2006.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] Y. Jiang, T. R. Oron, W. T. Clark, A. R. Bankapur, and e. a. Daniel D'Andrea, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy," 2016.

[14] T. Seemann, "Prokka: rapid prokaryotic genome annotation," *Bioinformatics*, vol. 30, p. 20682069, 2014.

[15] R. Overbeek, T. Begley, R. M. Butler, J. V. Choudhuri, H. Y. Chuang, M. Cohoon, V. de Crecy-Lagard, N. Diaz, T. Disz, R. Edwards, M. Fonstein, E. D. Frank, S. Gerdes, E. M. Glass, A. Goesmann, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, L. Krause, M. Kubal, N. Larsen, B. Linke, A. C. McHardy, F. Meyer, H. Neuweger, G. Olsen, R. Olson, A. Osterman, V. Portnoy, G. D. Pusch, D. A. Rodionov, C. Ruckert, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein, "The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes," *Nucleic Acids Res*, vol. 33, no. 17, pp. 5691–5702, 2005.

[16] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, and et. al., "A large-scale evaluation of computational protein function prediction," *Nat. Methods*, vol. 10, no. 3, pp. 221–227, 2013.

[17] D. Lee, O. Redfern, and C. Orengo, "Pedicting protein function from sequence and structure," *Nat. Rev. Mol. Cell Biol.*, vol. 8, no. 12, pp. 995–1005, 2007.