Identifying Fungi from Metagenomic Sequence Data

Charlie Curnin
Department of Computer Science
Stanford University
Stanford, CA 94305
ccurnin@stanford.edu

Nicholas Midler
Department of Biomedical Informatics
Stanford University
Stanford, CA 94305
midler@stanford.edu

Abstract

The study of the human mycobiome remains limited by the lack of tools capable of identifying fungal species from metagenomic sequence data. We develop naive Bayes classifiers and neural networks for this task, and assess their performance given several featurization strategies and hyperparameter settings. We find that simple approaches work remarkably well. Given input preprocessed with protein-prediction and sequence-annotation tools, a naive Bayes classifier with binary features tracking species-level sequence annotations attains an area under the precision-recall curve (AUPRC) of 0.982. Caveats remain, especially given the small size of our dataset (n=729), but these results at minimum demonstrate the suitability of machine-learning approaches (even simple ones) for this task when combined with effective preprocessing.

1 Introduction

The human mycobiome refers to the estimated one billion fungal species living symbiotically in the human body [Auchtung et al., 2018]. While it's believed that many of these fungal species impact or can serve as biomarkers for diseases, the mycobiome remains understudied. A primary obstacle to further study of the mycobiome is the difficulty in identifying different fungal species in the body.

Here, we present a machine learning-based tool for detecting fungal reads from raw sequencing data. Our approach includes two-stage preprocessing through existing tools to extract information useful for classification. From raw sequencing data, we obtain gene prediction results through MetaEuk [Levy Karin et al., 2020]. From these data, we find the top hits for related proteins through HMMER [Potter et al., 2018]. These results, which we consider featurizing in several different ways, are the input to naive Bayes models and neural networks that predict whether a given sequence belongs to a fungal species.

We used data from fungal and non-fungal genomes from the NCBI Genomes database [Leinonen et al., 2011] to develop and evaluate models. We took inspiration for this project from VIBRANT, a phage-detection program that uses protein annotations as input features.

2 Related work

The problem of identifying a class of organisms from sequencing data is not new to biology. The task of identifying a particular species (whether bacterial, human, viral, fungal) often contains idiosyncratic challenges based on the biology.

VIBRANT predicts viral sequences by annotating genes against databases of viral and non-viral proteins [Kieft et al., 2020]. Protein annotations are assigned using hidden Markov models that leverage knowledge of site-specific evolutionary patterns to assign likely proteins to sequences.

Preprint. Under review.

VIBRANT scans the protein descriptions, assigning high scores to annotations with keywords denoting a viral protein (such as capsid, spike, capsid, sheath). These scores are used as the input features for a neural network model.

HumanMycobiomeScan (HMS) predicts the presence of fungal reads through sequence homology [Soverini et al., 2019]. The HMS algorithm aligns the input reads against a reference database of fungal genomes using bowtie2 [Langmead and Salzberg, 2012]. HMS uses FASTQ files as inputs, allowing it to discard lower quality reads.

The disadvantage of sequence-matching algorithms is their inability to identify novel species of fungi. This became clear when we tested HMS using its default reference database with fungal and non-fungal reads obtained from NCBI SRA [Leinonen et al., 2011] (accession numbers SRR9634362, SRR14683046). The model had a precision of 0.988, but an extremely low recall of 0.00164. This is likely due to the relatively small size (1.2 GB) of the default database.

3 Dataset and features

3.1 Dataset

We collected sequencing data from NCBI Genomes [O'Leary et al., 2016] to develop and evaluate our models. The dataset comprises 729 examples — 508 fungal sequences and 221 non-fungal ones. The dataset covers 30 fungal samples, spanning a wide berth of the fungal phylogenetic tree, and 10 non-fungal eukaryotic samples. We limited our non-fungal examples to eukaryotes, which allowed us to standardize our gene prediction by only using MetaEuk. Prokaryotic organisms have different transcriptional motifs, and MetaEuk is only designed for eukaryotic gene prediction. Each genome was downsampled into five contigs with lengths varying from 5-15K base pairs. This follows a similar preprocessing routine used by [Kieft et al., 2020] to simulate the short contig lengths found in metagenomic data.

3.2 Preprocessing

Our approach relies on preprocessing through bioinformatics software to extract important information from sequence data. It includes two steps: gene prediction and sequence annotation.

We used MetaEuk to obtain gene-prediction results from FASTA-format sequence data. (Where necessary, FASTQ files were converted to FASTA format with a small script.) MetaEuk scans a given FASTA file against a provided reference database to search for matches. We used Pfam [Mistry et al., 2020], which currently contains 47 million sequences, as a reference database.

Within the protein-prediction results for a given sequence, we seek to identify proteins known to correspond to fungi and other organisms. We ran HMMER [Potter et al., 2018], which uses hidden Markov models to align sequence data, to compare our sequences to Ensembl Genomes [Howe et al., 2020], which includes data from fungi, plants, metazoa, bacteria and protists. For a given example, the response from HMMER includes a list of *hits*, matches between the input amino-acid sequence and the database. (We used the web API, maintaining the default setting of returning at most 10 hits.) Each hit comes with information including the associated species and its phylum.

3.3 Features

We considered several different methods of converting HMMER results into features suitable for machine-learning models. And as an extremely simple baseline for testing the efficacy of different featurizations, we considered one strategy (denoted `Empty` in Table 1) that provides no information — creating a single feature with a value of zero for each example.

3.3.1 Species-based features

The species associated with HMMER hits is likely to be informative for classifying a sequence as fungal or not. We evaluated several methods for featurizing this information. One approach is binary features that indicate, for each species seen in any training example, whether that species occurs in a given example (denoted `BySpeciesCheck` in 1). Another is integer-valued values that, for each species seen in any training example, count how many times that species occurs in a given

example (`BySpeciesCount`). A third option is features that represent each hit in an example with a numerical index, associated with each species seen in any training example (`BySpeciesIndex`). (For the `BySpeciesCheck` and `BySpeciesCount` featurization strategies, we also consider variants that normalize feature values across each sample.)

3.3.2 Phylum-based features

While the species associated with each hit is likely informative about whether a sequence is fungal or not, phylum-level information could suffice, too. In fact, phylum-level features may offer advantages, given that there are fewer phyla than species, meaning a model may have fewer parameters to learn. We implemented for phyla all the same featurization strategies described above for species (denoted `ByPhylumCheck` through `ByPhylumIndex` in Table 1).

3.3.3 Species- and phylum-based features

Lastly, we sought to test whether composing corresponding species- and phylum-based featurizations could lead to optimal results, combining the advantages of each approach (denoted `ByPhylumAndSpeciesCheck` through `ByPhylumAndSpeciesIndex` in Table 1).

4 Methods

We developed two kinds of models for this task: Naive Bayes classifiers and neural networks. Both work to classify examples as fungi or not based on input features, but estimate in different ways the probabilities of those outcomes.

4.1 Naive Bayes

Naive Bayes classifiers estimate the probability of a label y for an example $x = \langle x_1, ..., x_d \rangle$ by learning parameters that define the prior probabilities of labels, and the probabilities of feature values given each label. Their "naivete" comes from the built-in assumption that feature values are conditionally independent given a label. Symbolically, borrowing an equation from CS 229 lecture notes [Ng], a naive Bayes classifier computes the probability of the label $y = 1$ given data x as

$$\begin{aligned} p(y = 1|x) &= \frac{p(x|y = 1)p(y = 1)}{p(x)} \\ &= \frac{\big(\prod_{j=1}^{d} p(x_j|y = 1)\big)p(y = 1)}{\prod_{j=1}^{d} p(x_j|y = 1)\big)p(y = 0) + \prod_{j=1}^{d} p(x_j|y = 1)\big)p(y = 0)} \end{aligned}$$

In some naive Bayes applications, features are binary-valued, so for each feature x_j and label y_i, we only need to learn one parameter, $\phi_{j|y=y_i} = p(x_j = 1|y = y_1) = 1 - p(x_j = 0|y = y_1)$. In other applications, features can take on more than two values, and we model features given label as multinomial distributions [Ng], requiring learning more parameters.

4.2 Neural networks

Neural networks estimate the probabilities of labels by composing *neurons* that perform computations on the input data. A single neuron typically applies a nonlinear function to the dot product of some learned parameters θ and its input x: We can think of it symbolically like, $h_\theta(x) = \sigma(\theta^T x)$, for nonlinear $\sigma : \mathbb{R} \to \mathbb{R}$. These neurons are arranged in layers, with neurons in the lowest layer taking the features as their input, and neurons in higher layers taking the output from lower layers as their input. In classification problems, the model's final layer contains one neuron for each class, and we consider the model to predict the class corresponding to the neuron with the highest value.

We learn a model's parameters through backpropagation, an algorithm that computes the gradient of a loss function $J(\theta)$ with respect to the parameters in each layer, making use of the network's graph-like structure to "propagate back" gradients through the network and avoid redundant calculations. Then, given the gradient $\nabla_{\theta_\gamma} J$ for some parameter θ_γ, we update θ_γ through gradient descent using a hyperparameter learning rate α.

5 Results and discussion

We evaluated various types of models, strategies for featurization, and settings of hyperparameters to find the classifier with the best performance. (See Table 1.) We developed and assessed models with 10-fold cross-validation. That is, we split the dataset into 10 non-overlapping portions, $X_1, ..., X_{10}$. Then, for $i \in \{1, ..., 10\}$, we trained a model on the union of all portions except X_i, and evaluated the model on X_i. We then averaged evaluation metrics across the 10 folds. As our primary evaluation metric, we used area under the precision-recall curve (AUPRC). This follows guidance from CS229 course materials about evaluation with imbalanced classes. [course staff]

5.1 Results

Featurization has considerable impact on performance as measured by AUPRC. Among neural networks, AUPRC varied from a low of 0.694 to a high of 0.968 based on featurization strategy. In general, the more granular species-based featurizations are more effective than phylum-based ones. And adding phylum-based features onto species-based ones generally does not raise performance as measured by AURPC beyond what species-based features achieve alone.

Model choice also affects performance, but less drastically, with naive Bayes models and neural nets generally attaining similar AUPRCs.

Across all models, the highest performance came from a naive Bayes model with binary features tracking the presence or absence of a given species in an example's hits. (See Figure 5.1 for precision-recall curves.) Noticeably, this strategy is simpler than several other featurizations.

Figure 1: Precision-recall curve by fold for optimal classifier

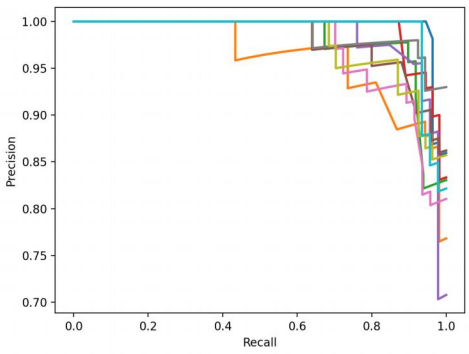

Table 1: AUPRC by featurization strategy, model type

Featurization	Model type	
	Naive Bayes	Neural network
Empty	0.848	0.848
BySpeciesCheck	0.978	0.968
BySpeciesCheckNorm	**0.982**	0.968
BySpeciesCount	0.980	0.974
BySpeciesCountNorm	0.980	0.971
BySpeciesIndex	0.804	0.694
ByPhylumCheck	0.876	0.802
ByPhylumCheckNorm	0.958	0.950
ByPhylumCount	0.950	0.928
ByPhylumCountNorm	0.961	0.901
ByPhylumIndex	0.685	0.692
ByPhylumAndSpeciesCheck	0.979	0.973
ByPhylumAndSpeciesCheckNorm	0.980	0.981
ByPhylumAndSpeciesCount	0.978	0.975
ByPhylumAndSpeciesCountNorm	0.980	0.974
ByPhylumAndSpeciesIndex	0.839	0.704

5.2 Neural network hyperparameters

We developed neural networks whose single hidden layer was 10-dimensional, and trained them with a learning rate of 0.001. We set those hyperparameters based on a grid search over eight hidden-layer dimensionalities ($\{1, 3, 5, 10, 25, 50, 75, 100\}$) and five learning rates ($\{0.01, 0.001, 0.0001, 0.00001, 0.000001\}$). The setting $(10, 0.001)$ led to the highest AUPRC, as measured through 10-fold cross-validation using `BySpeciesCheckNorm` featurization.

Table 2: AUPRC by hyperparameter

Learning rate	Hidden layer dimensionality							
	1	3	5	10	25	50	75	100
0.01	0.948	0.967	0.968	0.965	0.964	0.966	0.965	0.965
0.001	0.862	0.959	0.926	**0.976**	0.972	0.973	0.972	0.971
0.0001	0.848	0.873	0.916	0.891	0.938	0.965	0.975	0.973
0.00001	0.655	0.732	0.702	0.704	0.767	0.755	0.795	0.777
0.000001	0.733	0.698	0.701	0.678	0.670	0.691	0.702	0.760

5.3 Discussion

Simple models excelled in our experiments here. The leading classifier is a naive Bayes model with binary features tracking the presence or absence of a given species in an example's hits. Counting the number of hits per species, instead of recording presence/absence, doesn't improve performance as measured by AURPC on our dataset. Using a single-hidden-layer neural network instead of a naive Bayes classifier doesn't improve performance.

The success of this lightweight model speaks to the effectiveness of the preprocessing through MetaEuk [Levy Karin et al., 2020] and HMMER [Potter et al., 2018]. The gene-prediction and sequence-annotation results these tools provide are ripe for use with machine-learning models for fungi identification.

Several caveats remain. We split our limited dataset into training and evaluation sets, but not a test set for final evaluation after all hyperparameter tuning. Additionally, our dataset was limited in the number of examples it contained and the length of those sequences. These provide apt directions for future work.

5.4 Implementation details

We developed neural networks in PyTorch [Paszke et al., 2019] (using cross-entropy loss and Adam [Kingma and Ba, 2017] as an optimizer), and naive Bayes classifiers with `scikit-learn` [Pedregosa et al., 2011], which we also used to compute AUPRC.

6 Conclusion

We developed a naive Bayes classifier that achieves robust performance in identifying fungal sequence data, given preprocessing through gene-prediction and sequence-annotation bioinformatic software. We show that a lightweight naive Bayes approach performs better than some alternative featurization strategies, or models like a neural network. Limitations in our work come from limitations in the dataset we assembled, but the success of our models indicates significant potential for the application of machine-learning methods to the task of fungal sequence identification.

7 Acknowledgments

We formatted this report with the NeurIPS 2020 LaTeX template and style file. We're very grateful to our mentor Ian Tullis for helpful feedback and encouragement.

8 Contributions

Charlie handled preprocessing with HMMER, and developing and evaluating naive Bayes classifiers and neural networks. Nicholas created the databases, handled preprocessing with MetaEuk, and evaluated the performance of HMSan2.0.

References

Thomas A. Auchtung, Tatiana Y. Fofanova, Christopher J. Stewart, Andrea K. Nash, Matthew C. Wong, Jonathan R. Gesell, Jennifer M. Auchtung, Nadim J. Ajami, and Joseph F. Petrosino. Investigating colonization of the healthy adult gastrointestinal tract by fungi. *mSphere*, 3(2): e00092–18, Mar 2018. ISSN 2379-5042. doi: 10.1128/mSphere.00092-18. URL https://pubmed.ncbi.nlm.nih.gov/29600282. 29600282[pmid].

CS 229 course staff. URL http://cs229.stanford.edu/notes2021spring/notes2021spring/friday_lecture4.pdf.

Kevin L Howe, Premanand Achuthan, James Allen, Jamie Allen, Jorge Alvarez-Jarreta, M Ridwan Amode, Irina M Armean, Andrey G Azov, Ruth Bennett, Jyothish Bhai, Konstantinos Billis, Sanjay Boddu, Mehrnaz Charkhchi, Carla Cummins, Luca Da Rin Fioretto, Claire Davidson, Kamalkumar Dodiya, Bilal El Houdaigui, Reham Fatima, Astrid Gall, Carlos Garcia Giron, Tiago Grego, Cristina Guijarro-Clarke, Leanne Haggerty, Anmol Hemrom, Thibaut Hourlier, Osagie G Izuogu, Thomas Juettemann, Vinay Kaikala, Mike Kay, Ilias Lavidas, Tuan Le, Diana Lemos, Jose Gonzalez Martinez, José Carlos Marugán, Thomas Maurel, Aoife C McMahon, Shamika Mohanan, Benjamin Moore, Matthieu Muffato, Denye N Oheh, Dimitrios Paraschas, Anne Parker, Andrew Parton, Irina Prosovetskaia, Manoj P Sakthivel, Ahamed I Abdul Salam, Bianca M Schmitt, Helen Schuilenburg, Dan Sheppard, Emily Steed, Michal Szpak, Marek Szuba, Kieron Taylor, Anja Thormann, Glen Threadgold, Brandon Walts, Andrea Winterbottom, Marc Chakiachvili, Ameya Chaubal, Nishadi De Silva, Bethany Flint, Adam Frankish, Sarah E Hunt, Garth R IIsley, Nick Langridge, Jane E Loveland, Fergal J Martin, Jonathan M Mudge, Joanella Morales, Emily Perry, Magali Ruffier, John Tate, David Thybert, Stephen J Trevanion, Fiona Cunningham, Andrew D Yates, Daniel R Zerbino, and Paul Flicek. Ensembl 2021. *Nucleic Acids Research*, 49(D1):D884–D891, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa942. URL https://doi.org/10.1093/nar/gkaa942.

Kristopher Kieft, Zhichao Zhou, and Karthik Anantharaman. Vibrant: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. *Microbiome*, 8(1):90–90, Jun 2020. ISSN 2049-2618. doi: 10.1186/s40168-020-00867-0. URL https://pubmed.ncbi.nlm.nih.gov/32522236. 32522236[pmid].

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, Mar 2012. ISSN 1548-7105. doi: 10.1038/nmeth.1923. URL https://pubmed.ncbi.nlm.nih.gov/22388286. 22388286[pmid].

Rasko Leinonen, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. The sequence read archive. *Nucleic acids research*, 39 (Database issue):D19–D21, Jan 2011. ISSN 1362-4962. doi: 10.1093/nar/gkq1019. URL https://pubmed.ncbi.nlm.nih.gov/21062823. 21062823[pmid].

Eli Levy Karin, Milot Mirdita, and Johannes Söding. Metaeuk—sensitive, high-throughput gene discovery, and annotation for large-scale eukaryotic metagenomics. *Microbiome*, 8(1):48, Apr 2020. ISSN 2049-2618. doi: 10.1186/s40168-020-00808-x. URL https://doi.org/10.1186/s40168-020-00808-x.

Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik L L Sonnhammer, Silvio C E Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, Robert D Finn, and Alex Bateman. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, 10 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa913. URL https://doi.org/10.1093/nar/gkaa913.

Andrew Ng. URL hhttp://cs229.stanford.edu/notes2020spring/cs229-notes2.pdf.

Nuala A. O'Leary, Mathew W. Wright, J. Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, Alexander Astashyn, Azat Badretdin, Yiming Bao, Olga Blinkova, Vyacheslav Brover, Vyacheslav Chetvernin, Jinna Choi, Eric Cox, Olga Ermolaeva, Catherine M. Farrell, Tamara Goldfarb, Tripti Gupta, Daniel Haft, Eneida Hatcher, Wratko Hlavina, Vinita S. Joardar, Vamsi K. Kodali, Wenjun Li, Donna Maglott, Patrick Masterson, Kelly M. McGarvey, Michael R. Murphy, Kathleen O'Neill, Shashikant Pujar, Sanjida H. Rangwala, Daniel Rausch, Lillian D. Riddick, Conrad Schoch, Andrei Shkeda, Susan S. Storz, Hanzhen Sun, Francoise Thibaud-Nissen, Igor Tolstoy, Raymond E. Tully, Anjana R. Vatsan, Craig Wallin, David Webb, Wendy Wu, Melissa J. Landrum, Avi Kimchi, Tatiana Tatusova, Michael DiCuccio, Paul Kitts, Terence D. Murphy, and Kim D. Pruitt. Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic acids research*, 44(D1):D733–D745, Jan 2016. ISSN 1362-4962. doi: 10.1093/nar/gkv1189. URL https://pubmed.ncbi.nlm.nih.gov/26553804. 26553804[pmid].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

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*, 12:2825–2830, 2011.

Simon C Potter, Aurélien Luciani, Sean R Eddy, Youngmi Park, Rodrigo Lopez, and Robert D Finn. HMMER web server: 2018 update. *Nucleic Acids Research*, 46(W1):W200–W204, 06 2018. ISSN 0305-1048. doi: 10.1093/nar/gky448. URL https://doi.org/10.1093/nar/gky448.

Matteo Soverini, Silvia Turroni, Elena Biagi, Patrizia Brigidi, Marco Candela, and Simone Rampelli. Humanmycobiomescan: a new bioinformatics tool for the characterization of the fungal fraction in metagenomic samples. *BMC genomics*, 20(1):496–496, Jun 2019. ISSN 1471-2164. doi: 10.1186/s12864-019-5883-y. URL https://pubmed.ncbi.nlm.nih.gov/31202277. 31202277[pmid].