Parasighter: a k-mer based classifier for apicomplexan 18S rRNA sequences

Jason Moggridge

28/10/2020

Introduction

Apicomplexans are a large phylum of single-celled eukaryotes, all of which are obligate endoparasites of metazoa except in the rarest of cases (Saffo et al. 2010). The phylum is named for their apical complex, a unique organellar structure specialized for host-cell invasion. They often have complex life-cycles that can involve several hosts (Seeber and Steinfelder 2016). As the agents responsible for many common and serious diseases - including malaria, toxoplasmosis, babesiosis, cryptosporiosis, *etc.* - they represent a major burden to health and economy across the globe (Flegr et al. 2014; Kristmundsson and Freeman 2018). As such, it is of great interest to find and classify these organisms as a first step towards diagnosis and treatment.

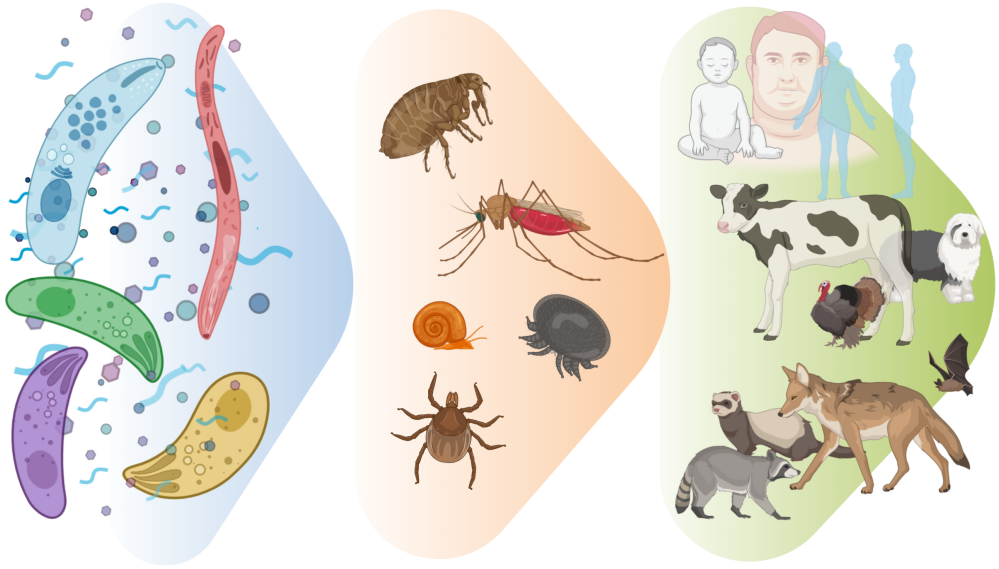

Figure 1. Apicomplexa are eukaryotic endoparasites of metazoa. Their life-cycles can be complex, involving transmission by an ectoparasitic vector such as ticks or mosquitoes. Health and economic impacts of this group are significant as they affect wildlife, livestock, and humans globally. (Figure made with BioRender)

Detection of parasites has traditionally been performed by tedious microscopic analyses (Renoux et al. 2017). Ribosomal RNA genes, particularly the eukaryotic 18S and prokaryotic 16S, have long been employed as phylogenetic markers (*ie.* barcodes) due to their high degree of conservation while still containing enough variability to be useful for taxonomic discrimination. Currently, DNA barcoding allows for the simultaneous identification of many organisms in an environmental sample by differences in a chosen marker gene. However, the utility of these methods is dependent upon reference sequences and accurate classifications to these. Often, methods for classifying large numbers of barcode sequences discriminate sequences based on their k-mer profiles in lieu of computationally costly-alignments. For example, the RDP classifier is a widely used tool that employs a naive Bayes classifier on k-mer profiles (Wang et al. 2007). These methods are often superior in speed and accuracy compared to BLAST, provided that the classifier is trained on an adequate reference database (Porter and Hajibabaei 2018).

In this work, I evaluated the performance of popular classification algorithms on the task of genus-level assignment of unknown Apicomplexan 18S sequences. I selected the best performing algorithm to create a species-level classifier for 18S sequences of the most common apicomplexan species in the NCBI database. Further, I trained another classifier to evaluate the discrimination of closely related species using a dataset of *Babesia* and *Thelieria* 18S sequences. All analyses were performed in the R statistical programming language.

Analysis

18S sequence data acquisition

I obtained 15,093 apicomplexan 18S ribosomal rRNA sequences from the NCBI 'nuccore' database using the `rentrez` R package on 2020-10-20. Retrieved sequences were 300-2500 bp long, nuclear genomic sequences only, and from with Genbank or Refseq (search expression: "*18S ribosomal rna[Title] AND "apicomplexa"[Organism] AND 300:2500[SLEN] AND biomol_genomic[PROP] NOT genome[TITL] AND (ddbj_embl_genbank[filter] OR refseq[filter])*".

Sequence quality control and filtering

Taxonomic labels were checked for consistency and edited where necessary. Records were discarded if they met any of the following criteria:

- ambiguous organism name (uncultured, sp., cf. labels)
- molecule type not explicitly labelled as nuclear genomic DNA
- contains any non-ACGT characters
- duplicated sequence already in database
- title contains 'internally transcribed spacer' or 'ITS'

Upon checking the distribution of sequence lengths, I found that they differ widely between taxa; hence, I decided not to filter by sequence length but to simply keep this in mind (fig. 2).

Sampling, feature generation and data-splitting

After filtering steps, ~5k sequences were in the remaining dataset comprising 350 different species from 50 genera. At the genus rank, the top seven genera account for the vast majority of the sequences. I created three subsets of sequences for classifier training and validation:

- *'genus-rank' dataset*: 1,575 18S sequences, evenly split between seven genera: *Babesia*, *Cryptosporidium*, *Theileria*, *Sarcocystis*, *Hepatozoon*, *Eimeria*, *Plasmodium*
- *species-rank dataset*: 1,180 18S sequences, sampled equally from the 20 most common apicomplexan species in the filtered data
- *Babesia/Theileria dataset*: 686 sequences sampled equally from 14 closely-related species of genera *Babesia* and *Theileria*

Proportions of (1-4)-mers were generated for each dataset using `Biostrings`; all datasets were split 50-50 into training and validation sets using the `caret` package prior to training.

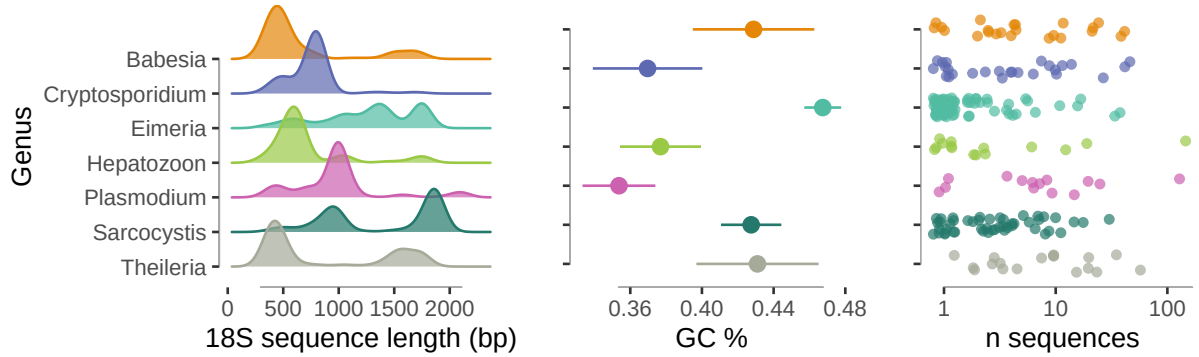

Figure 2. Charateristics of sequences from seven apicomlexan genera in the data used in comparing various classification algorithms (RF, NB, GBM, kNN). *left*: Sequence length varies by genera with some groups being bimodal; *middle*: GC-content for each genus (mean+/-sd); *right*: The dataset contains 208 species; to illustrate the distribution of sequences to species within each genera, each species is represented as a point and plotted the sequence count (eg. *Eimeria* and *Theileria* contain many species with low representation in the dataset; *Hepatozoon* and *Plasmodium* each contain one species that accounts for a large proportion of the sequences).

Training classifiers of apicomlexan 18S sequences at the genus-rank by various methods

I tried four different classification algorithms (random forest, stochastic gradient boosting, naive Bayes, and k-nearest neighbors) to ascertain which might be best-suited to the classification of apicomplexan 18S k-mer profiles. For each model, I used the `caret` package to perform a search through various model tuning parameters, selecting an optimal model based on estimated accuracy on the training data. Resampling was done by bootstrapping (5 repeats) to strengthen accuracy estimates. Computations were parallelized to speed up the training of classifiers, using the `parallel` package.

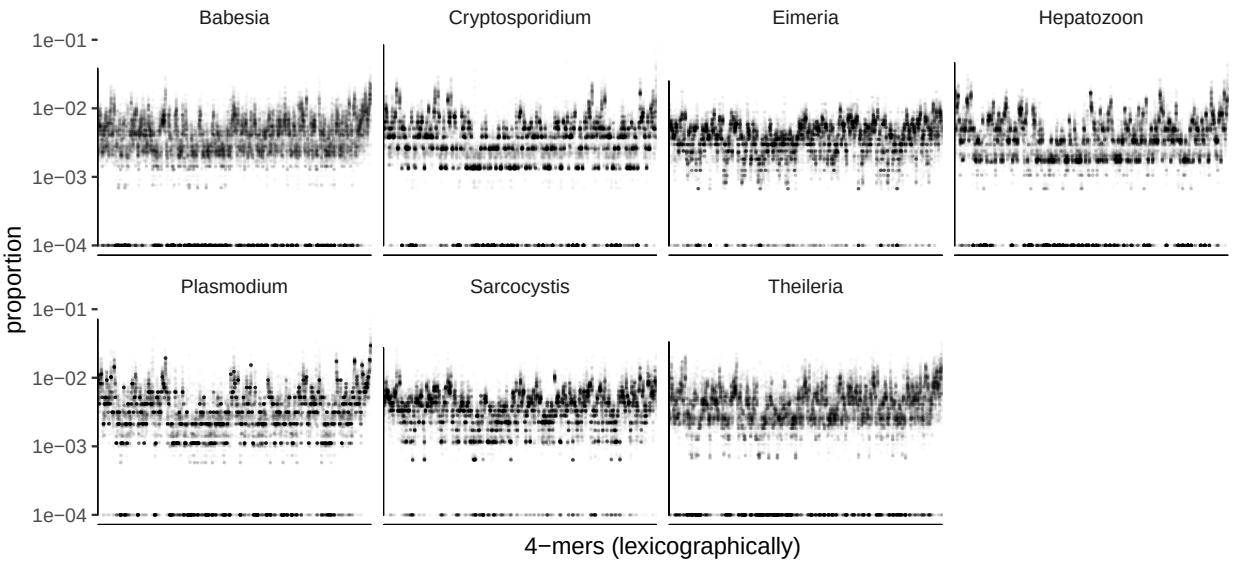

Figure 3. 4-mer profiles of apicomplexan genera from 18S sequences in the genus-rank dataset used to compare classification algorithms. Each point shows the proportion of a k-mer (along lexicographic y-axis) in a sequence; all sequences are plotted in separate panels by genus. Zero-proportion values are omitted. This gives a visual representation of the overall distribution of k-mer proportions with each group and a general sense of groups differences. The banding patterns might arise from the uneven distribution of species within genera (fig2:right).

Random Forest (RF): Random Forest is a powerful and robust classification/regression method based on creating a group of decision trees. Model tuning was performed by varying the `mtry` parameter, which represents the number of variables that can be used to split data on at a given node in a tree; the default value is the square-root of the number of predictors (18). Optimal accuracy (98.9%) was achieved with `mtry = 30`, though accuracy was very high across the range applied (fig.5:RF).

Stochastic gradient boosting (GBM): Similar to random forest, stochastic gradient boosting also employs a another tree ensemble-based approach, except with a large number of tiny decision stumps strengthened through 'boosting'. GBM is prone to over-training as the algorithm doesn't stop automatically.

Classifiers were trained in two tuning searches over a range of stump-depths (1,2, or 3 per node; GBM1), learning rates (0.01, 0.1; GBM2), and boosting iterations (50-150 GBM1; 100-500 GBM2). After performing the first tuning search (fig.5:GBM1), it appeared that the optimal model (depth=2, ntree=150) could still be improved with more boosting iterations. This was not the case in fact, though it was confirmed that the model would not be improved by decreasing the learning rate, given 100-500 boosting iterations (fig.5:GBM2). Bootstrapped accuracy estimates of the GBM1 and GBM2 models were 98 % (sd=0.6) and 97.5 % (sd=0.7).

Naive Bayes (NB): Naive Bayes uses a probabilistic approach to classification that makes strong assumptions of linear independence of predictor variables. For these models, I centered and scaled (mean=0, sd=1) each k-mer proportion variable in the train and test datasets (using the `caret` package once again).

For the initial naive Bayes model, the only tuning parameter varied was `use-kernel` and the best setting was 'false' (*ie.* using Gaussian distribution) (Fig.5:NB). Accuracy was 95.4% (95% CI: 93.7, 96.8) but this was worse than RF or GBM classifiers in training. Particularly, the classification of *Babesia* and *Thelieria* sequences was relatively weak.

K-Nearest Neighbors (kNN): kNN requires data to be normalized, so centering and scaling were performed prior to training. Ten models with varying k (5-23) all had high accuracy, though performance was best with fewer neighbours (fig.4:kNN). The best model (k=5) had accuracy of 98.4 % (sd: 0.63 %), rivaling the random forest classifier.

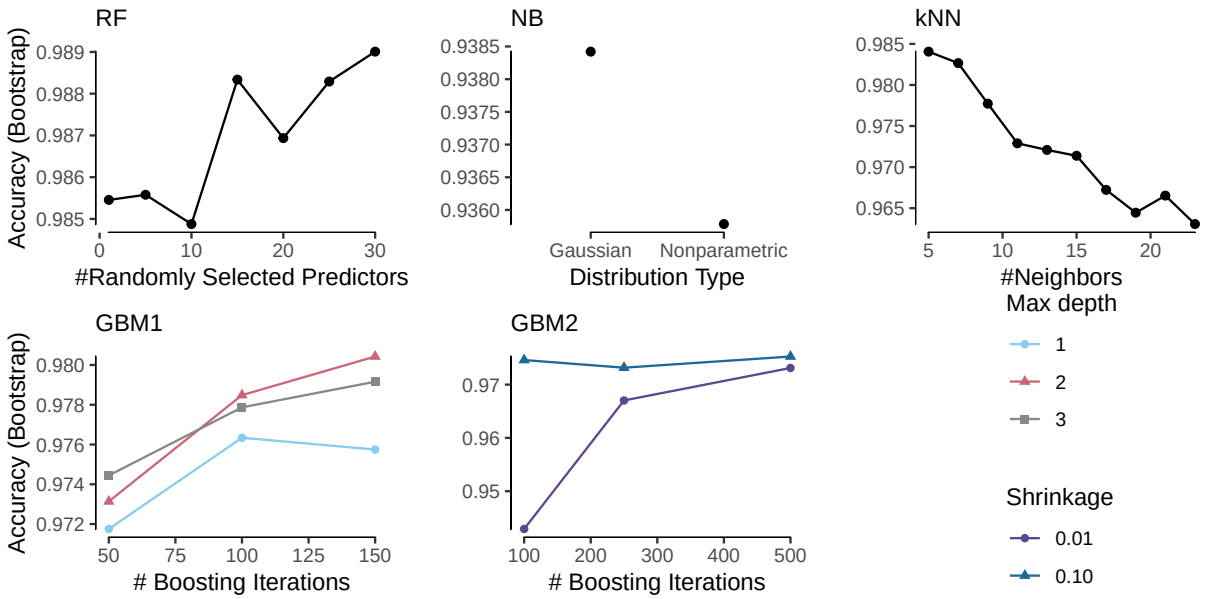

Figure 4. Tuning searches for each classification algorithm in training on the genus-rank dataset of apicomplexan 18S k-mer profiles. Each point represents an individual classifier explored during the tuning search by `caret::train`.

Testing the 'genus-rank' classifiers on unknown data

Five classifiers (GBM1, GBM2, kNN, NB, RF) were tested against the genus-rank test dataset for validation. The overall and class-specific performance measures of each classifier the against unseen data were evaluated (fig. 6). In terms of overall accuracy, the k-NN (99 %) and RF classifiers (99.1 %) performed slightly better than the GBM models (98.5 and 98.9%), but these differences were small relative to the error on these estimations (fig.6:left). Naive Bayes performed poorly in comparison, with a pitiful 96.6 % accuracy on the unseen data. In terms of class-specific performance, four algorithm (RF, GBM, kNN) show a similar performance profile, with lower performance on *Babesia* and *Theileria* sequences (fig.6:right). In contrast, the naive Bayes struggled on *Sarcocystis* and *Eimeria*, whereas the other classifiers performed well on these groups.

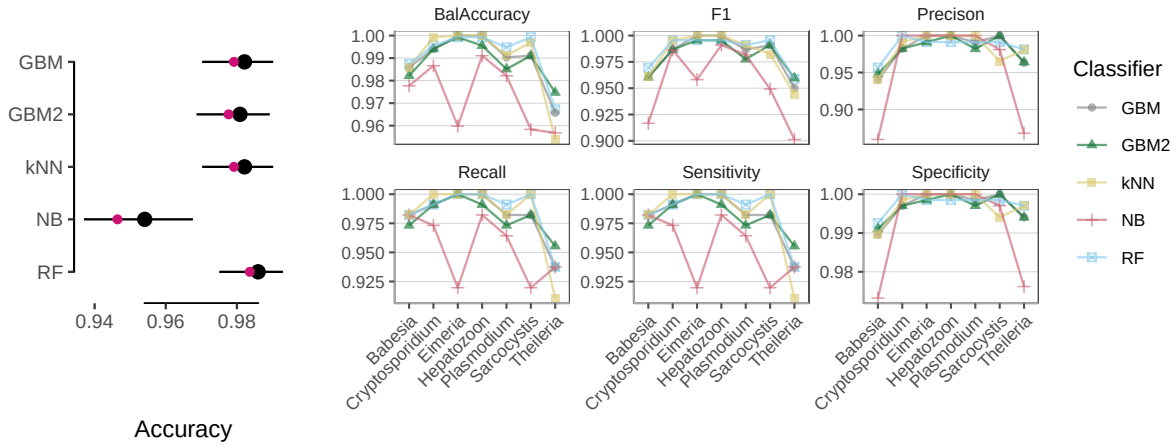

Figure 6. Performance of various genus-rank classifiers in validation with unseen 18S kmer-profiles from the test dataset. *left* Overall accuracy +/- sd is shown as a point-range in black, Kappa statistic is shown as a red point; *right* Genus-specific performance statistics for the five classifiers tested. (GBM: stochastic gradient boosting, kNN: k-nearest neighbors, NB: naive Bayes, RF: random forest)

Random forest classifier for species-level assignments

A random forest classifier was trained on the 'species-rank' training set of 600 18S rRNA k-mer profiles from 20 different apicomplexan species. Models were generated using range of `mtry` (predictors per node) values (1-30) and accuracy estimated were derived from by a 5 times 5-fold cross-validation re-sampling. The best random forest classifier (mtry=25) had a training accuracy of 93.9% (+/-2.3%); Kappa was 93.6% (+/- 2.4). The classifier was validated by testing on an unseen dataset of 580 kmer profiles evenly divided between species; the test accuracy was 93.4% (95% CI: 91.1, 95.3); Kappa 93.1%.

To test the limits of the random forest algorithm, I assembled a dataset of 686 18S rRNA k-mer profiles, including 8 species of the genus *Babesia* and 6 species of the closely-related genus *Theileria*, with sequences evenly-split between species. This set was split 1:1 into training and test datasets. Training was performed for a range of `mtry` values; re-sampling was done by a 5 times 10-fold cross-validation strategy. The best model (`mtry=30`) had an estimated accuracy of 94.1 % (Kappa=93.6%). When tested on the unseen portion of the data, the accuracy decreased only slightly to 93.5% (95% CI: 91.1, 95.3; Kappa : 0.931).

Discussion

In comparing algorithms used to create genus-level classifiers for apicomplexan 18S kmer profiles, it was noted that the tree-based methods random forest (RF) and stochastic gradient boosting (GBM) both performed exceptionally well. K-Nearest neighbours also acheived a high degree of accuracy and was second only to the random forest classifier. Notably, the naive Bayes classifier could not equal the performance of these algorithms within the tuning parameters that were search, and with the preprocessing that was performed (performance was similar on raw proportions)

The naive Bayes classification algorithm depends on predictors being independent, however k-mer proportions can be highly dependent (eg. 'xGGG', and 'GGGx' are likely to be correlated because the contain a common substring). As such, the naive Bayes classifier likely input data with more extensive preprocessing than was performed here, either involving elimination of features with high collinearity, and/or tansformation and normalization of probabilities. Tree-based methods are more robust when faced with non-normalized data and this was the case in this study also.

When tasked with species-level classification, the performance of the random forest classifier was only slightly worse in terms of accuracy than what was seen in genus-rank assignments. Even when faced with a dataset of 14 closely-related species from *Babesia* and *Theileria* genera (which were often among the misclassifions at the genus-rank), the classification accuracy remained surprisingly strong.

Acknowledgements

Most of all, I would like to thank Dr. Adamowicz and Jacqueline May for putting together a challenging and fun learning experience; I really enjoyed this portion of the course and putting together this paper.

This project would not even have been imaginable if it weren't for the wonderful R community and the wealth of amazing packages created and shared by so many brilliant people. In particular, the `caret` package by Max Kuhn did much of the computational heavy-lifting for this project. I also I benefited greatly from the (excellent `caret` package documentation)[https://topepo.github.io/caret/] and watching Max's webinar video.

References

Flegr, Jaroslav, Joseph Prandota, Michaela Sovičková, and Zafar H. Israili. 2014. "Toxoplasmosis – A Global Threat. Correlation of Latent Toxoplasmosis with Specific Disease Burden in a Set of 88 Countries." Edited by Delmiro Fernandez-Reyes. *PLoS ONE* 9 (3): e90203. https://doi.org/10.1371/journal.pone.0090203.

Kristmundsson, Árni, and Mark Andrew Freeman. 2018. "Harmless Sea Snail Parasite Causes Mass Mortalities in Numerous Commercial Scallop Populations in the Northern Hemisphere." *Scientific Reports* 8 (1): 7865. https://doi.org/10.1038/s41598-018-26158-1.

Porter, Teresita M., and Mehrdad Hajibabaei. 2018. "Automated High Throughput Animal CO1 Metabarcode Classification." *Scientific Reports* 8 (1): 4226. https://doi.org/10.1038/s41598-018-22505-4.

Renoux, Lance P., Maureen C. Dolan, Courtney A. Cook, Nico J. Smit, and Paul C. Sikkel. 2017. "Developing an Apicomplexan DNA Barcoding System to Detect Blood Parasites of Small Coral Reef Fishes." *Journal of Parasitology* 103 (4): 366–76. https://doi.org/10.1645/16-93.

Saffo, M. B., A. M. McCoy, C. Rieken, and C. H. Slamovits. 2010. "Nephromyces, a Beneficial Apicomplexan Symbiont in Marine Animals." *Proceedings of the National Academy of Sciences* 107 (37): 16190–5. https://doi.org/10.1073/pnas.1002335107.

Seeber, Frank, and Svenja Steinfelder. 2016. "Recent Advances in Understanding Apicomplexan Parasites." *F1000Research* 5 (June): 1369. https://doi.org/10.12688/f1000research.7924.1.

Wang, Qiong, George M. Garrity, James M. Tiedje, and James R. Cole. 2007. "Naïve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy." *Applied and Environmental Microbiology* 73 (16): 5261–7. https://doi.org/10.1128/AEM.00062-07.