On Efficient Meta-Filtering of Big Data

Keith Dillon[1] and Yu-Ping Wang[2]

***Abstract*— There is an explosion in the number of new approaches being developed for extracting information from biological data, creating a need for intelligent ways to utilize so many methods. We take a perspective based on viewing methods as filters which reject undesired data and which may have complementary and redundant performance. We consider approaches for efficiently combining such filters. We provide quantitative strategies for choosing which filters to use and the best order to apply them, based on viewing each filter as a coordinate transformation on performance metrics. The approach is demonstrated using a range of methods to filter sequence data for homology detection, where we show the advantages of more sophisticated strategies in terms of achieving competitive speed with significantly-improved rejection of undesired data.**

I. Introduction

Researchers across many fields are continually developing new methods that are often overlapping as well as complementary in their application, leading to a new challenge in determining which is the best method to use. For example [1] (and accompanying website [2]) lists over 100 methods in high-throughput sequencing, comparing them for a list of factors to aid in selection of the most appropriate method for a task. A related direction of research is the combined use of multiple methods to achieve better performance than any of the individual methods could attain. Improvements could potentially be achieved in terms of accuracy and extraction of independent information; one approach is the use of voting methods to combine predictions from different methods (e.g. [3]); another is to devise methods that merge other approaches somehow, for example to address the closely-related problem of combining multi-omic information [4], [5].

Gains made in computational tractability by utilizing combinations of methods have also proven critical to research progress. A wide variety of pre-processing techniques are employed out of necessity, generally on an ad hoc basis, in order to produce a problem of manageable size. Indeed, improvements in computational load is one of the key benefits advertised for the broad strategy of feature selection [6], [7]. For example in bioinformatics, BLAST filtering [8] is commonly used to eliminate regions of DNA with low probability of matching a target sequence for homology detection. The presumption is that with the pre-processing step, an equally-accurate final result is achieved using far less time and fewer resources. A recent approach which turns the process somewhat in the other direction, is taken by research on screening methods for LASSO regression [9], [10]. Here, the pre-processing algorithm is specifically created with the performance limitations of the final algorithm in mind. The goal is to similarly eliminate variables using an efficient univariate test, before applying the LASSO optimization algorithm on the reduced problem. The above-mentioned accuracy assumption is guaranteed by design for so-called "safe screening" methods [11] which only reject variables which are not in the final LASSO solution.

These strategies for combining methods can be themselves viewed as broader methods, or "meta-methods", which combine the advantages of multiple tools, and produce a result superior to that which any can individually achieve. This overall performance metric used to select such strategies depends on not just resulting accuracy, but also computational speed. Such a perspective is taken by [12], which considered a high-level formulation of sequence filters for homology-detection where the collection of potential filters is the set of candidate methods. Individual filters were rated based on metrics for accuracy, efficiency, and computation time, using empirical testing. In this paper we follow a similar approach. We will generally view different variable selection methods as filters and consider a process whereby the above metrics are estimated for a set of candidate filters. Then we will develop optimal strategies for selecting combinations of filters. We provide simulated results demonstrating improvements in performance that may be achieved by different strategies.

II. Methods

We presume we have an ensemble of methods (which we call "filters") that perform variable elimination, generally with different performances. The goal is to choose the best set of filters, which reject the most true negatives for some task, while taking the least time to do so. Filters are rated based on their empirical accuracy, efficiency, and time; t_i is the average processing time per variable of the i^{th} filter, e_i is the efficiency of the i^{th} filter, and N is the total number of variables in the data being filtered. These ratings must be estimated from representative data. Accuracy is the fraction of true positives retained by filtering, and in this paper we assume all filters have perfect accuracy, as we can simply reject those which are not highly-accurate. Hence we have the situation depicted generally in Fig. 1, where multiple overlapping options are available to filter variables. Efficiency is the fraction retained after filtering, which in the case where the number of true variables is very small, approximates the rate of false positives.

The authors wish to thank the NIH (R01 GM109068, R01 MH104680, R01 MH107354) and NSF (1539067) for their partial support.

The authors are with the Department of Biomedical Engineering, Tulane University, New Orleans, LA 70118, USA

[1] kdillon1@tulane.edu

[2] wyp@tulane.edu

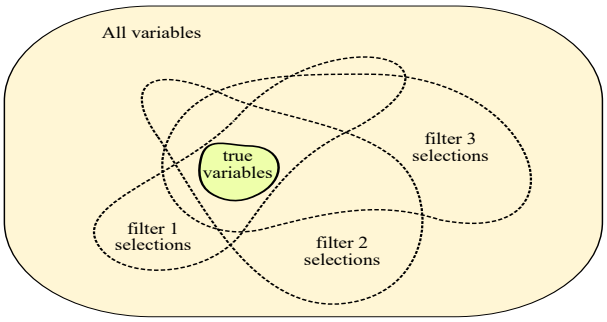

Fig. 1. Multiple filters, partially-overlapping and rejecting different subsets of true negatives; all are assumed to pass true positives.

A. Independent Filters

Before moving on to more sophisticated methods, we will start by considering the common assumption of statistically independent filters. If we presume the probability of rejection of a variable to be independent for different filters (except for true variables, which are retained by all), then the net efficiency of a series of filters is equal to the product of the individual efficiencies. So for example the efficiency resulting from applying filter 1 followed by 2 is $e_{1,2} = e_1 e_2$. Using this fact, we can write the total time taken by a series of filters $\{1, 2, 3, ...\}$ as a series of products where the time each filter takes is multiplied by the remaining number of variables at that point, yielding

$$\begin{aligned} T_{total} &= t_1 N + t_2 e_1 N + t_3 e_{1,2} N + t_4 e_{1,2,3} N + ... \\ &= t_1 N + t_2 e_1 N + t_3 e_1 e_2 N + t_4 e_1 e_2 e_3 N + ... \end{aligned} \quad (1)$$

We can equivalently view this as a series of affine transformations on coordinates for time and efficiency, as follows,

$$\frac{T_{total}}{N} = t_1 + e_1 \left(t_2 + e_2 \left(t_3 + e_3 \left(... \right) \right) \right) \quad (2)$$

To see how we can separate or reorder the filters, we first implement each of these transformations in homogeneous coordinates with the following transformation,

$$\mathbf{F}_i^{(ind)} = \begin{pmatrix} e_i & 0 & 0 \\ t_i & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

The product of multiple filters has the same form as the transformation for a single filter, with a composite efficiency that is the product of efficiencies from each component filter, and a composite time that depends on the order they are applied. If we compare the products $\mathbf{F}_i^{(ind)} \mathbf{F}_j^{(ind)}$ versus $\mathbf{F}_j^{(ind)} \mathbf{F}_i^{(ind)}$, we have the following,

$$\begin{pmatrix} e_i e_j & 0 & 0 \\ t_i + t_j e_i & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ versus } \begin{pmatrix} e_j e_i & 0 & 0 \\ t_j + t_i e_j & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

This is applied to a vector $(N, T, 1)^T$, containing N the net number of variables remaining, and T the net time spent thus far. The faster ordering is the choice with a smaller resulting time term. In this case the first option in Eq. (4) is faster if $t_i + t_j e_i$ is smaller, and the second is faster if $t_j + t_i e_j$ is smaller. The ranking between these two factors is equal to the ranking between factors of the form,

$$f_k = \frac{1 - e_k}{t_k} \quad (5)$$

This suggests a simple ordering strategy: apply the filters in order of decreasing f_k. It is straightforward to prove that this ordering strategy is optimal for any length composition by considering the optimal ordering of each pair and noting that reducing the time of any component transformation improves the net time.

B. Successive Safe Filters

Next we consider the extreme case where filters are not independent but pass overlapping subsets with decreasing size, generalizing the strategy of combining a safe-screening method with LASSO regression. Here we assume $e_S = \min \{e_i \mid i \in S\}$, the net efficiency of a composition equals the best efficiency of the members. Only an ordering in terms of decreasing efficiency is worth considering here, so we can describe the result with successive transformations of the form

$$\mathbf{F}_i^{(safe)} = \begin{pmatrix} 0 & 0 & N e_i \\ t_i & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (6)$$

Here we compare the product $\mathbf{F}_i^{(safe)} \mathbf{F}_j^{(safe)}$ to simply applying $\mathbf{F}_j^{(safe)}$, which yields the ranking of f_j versus $1/t_i$. Noting that $1/t_i \geq f_i$, we can again utilize a ranking based on decreasing f_k.

C. Disjoint Filters

We next consider the extreme case where different filters exclude completely different subgroups of variables. This is, in a sense, the best case scenario for multiple filters of a given set of efficiency and time parameters. Here the action of a given filter is a fixed reduction in the number of remaining variables. This can be modeled with the transformation,

$$\mathbf{F}_i^{(dis)} = \begin{pmatrix} 1 & 0 & -(1 - e_i) N \\ t_i & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (7)$$

The comparison between the orderings $\mathbf{F}_i^{(dis)} \mathbf{F}_j^{(dis)}$ and $\mathbf{F}_j^{(dis)} \mathbf{F}_i^{(dis)}$, yields the same ordering strategy as with the independent case, which can be shown to be optimal using similar arguments.

D. Multidimensional Filter Signatures

In order to maintain more structural information about filters and their relatedness, we extend the efficiency to a vector model $\mathbf{e}_i$, which has multiple elements for the efficiency over multiple dimensions or groups of variables, which we choose to best provide independent information. Ideally we would like as many dimensions as possible to capture structural information, but we may need to limit the length of $\mathbf{e}_i$ for practical reasons, and so partition our

variables into groups. The coordinate vector we now use is of the form $(\mathbf{n}^T, T, 1)^T$; $\mathbf{n}$ is a length-D vector containing the number of variables in each dimension, and T is a scalar containing the total time as before. We use a higher-order version of independent filters; transformations are now formulated as

$$\mathbf{F}_i^{(ind)} = \begin{pmatrix} \mathbf{E}_i & \mathbf{0} & \mathbf{0} \\ \mathbf{1}^T t_i & 1 & 0 \\ \mathbf{0}^T & 0 & 1 \end{pmatrix}, \tag{8}$$

where $\mathbf{E}_i$ is a $D \times D$ diagonal matrix with $\mathbf{e}_i$ on the diagonal, and $\mathbf{1}$ and $\mathbf{0}$ are vectors of zeros and ones, respectively, of length D.

Comparing pairwise products $\mathbf{F}_i^{(ind)}\mathbf{F}_j^{(ind)}$ versus $\mathbf{F}_j^{(ind)}\mathbf{F}_i^{(ind)}$ as before, we get a ranking between factors of the form

$$f_k = \frac{1 - \mathbf{e}_k^T \mathbf{n}}{t_k}. \tag{9}$$

So the efficiency vectors operate as a weighting on $\mathbf{n}$, the vector of net number of remaining variables in each dimension. Unfortunately we cannot employ arguments as used in the scalar cases to guarantee the optimal ordering for a longer chain of filters. However our scalar results suggests a promising greedy strategy, where we start with a $(\mathbf{n}^T, T, 1)^T$, and choose the best filter at each stage according to the best f_k via Eq. (9). In the next section we will compare this and simpler methods in numerical experiments.

III. Example: DNA sequence filtering

In this section we will compare the performance of different strategies for a set of randomly-chosen candidate filters. The filters to be used are DNA sequence filters designed according to the method of [12], for locating matches to a target sequence fragment. We generated a diverse range of filters using a sweep on input parameters for the method, and retained those which achieved a minimum of 95 percent accuracy on a random test sequence, resulting in 250 filters total. The efficiency versus time to perform filtering, and a histogram of the efficiencies is given in Fig. 2. The time is

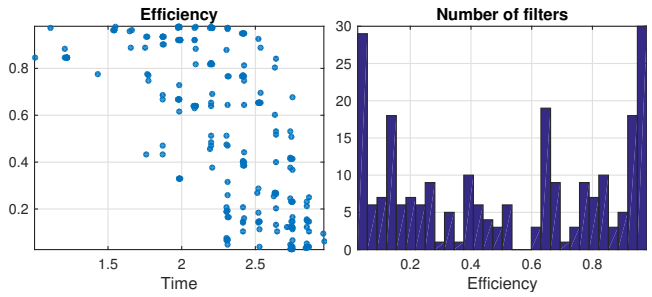

Fig. 2. Processing time versus efficiency, and histogram of efficiencies, for 250 candidate filters. Note that a lower efficiency is better, see [12]

normalized to that of the fastest filter. To capture information about filter correlation, we used a random sequence one million base pairs long, and computed the binary vector representing pass/reject decisions at each base pair as the $\mathbf{e}_i$ for the multidimensional models. In both the scalar and multidimensional versions of the algorithm, we used a stopping criterion of 0.001 efficiency, meaning we stopped applying new filters when the composite filter efficiency was within 0.001 of the efficiency possible achieved by applying all filters in the set of candidates.

Next we investigated the performance in terms of various properties of the candidate filter set to better understand the effect of correlations between filters. For each data point we performed 1000 realizations with a different subset of filters chosen randomly each time, employing three different ways of varying that subset choice. We also compared the performance using the scalar and multidimensional strategies to three simpler strategies. The first of these simpler strategies is to simply use the most efficient filter (which we labeled "No Screen"), and the second and third use that same efficient filter but with a pre-screening performed by the fastest ("T-Screen") or lowest f factor ("F-Screen") filters. Fig. 3 gives the average net time, efficiency, and composite filter length for ten of these 1000-realization experiments, where each experiment used 20 filters chosen randomly from a subset of the 250 that is restricted by a minimum efficiency (given as the x-axis). So towards the right of the plots, the sets of candidate filters become increasingly inefficient (recall a filter with an efficiency of 1.0 rejects nothing). We see that as the sets of candidates become more inefficient, both the scalar and multidimensional strategies take increasingly longer but are also increasingly more efficient than the simple strategies. Further the multidimensional strategy is roughly twice as fast as the scalar strategy, while achieving the same efficiency.

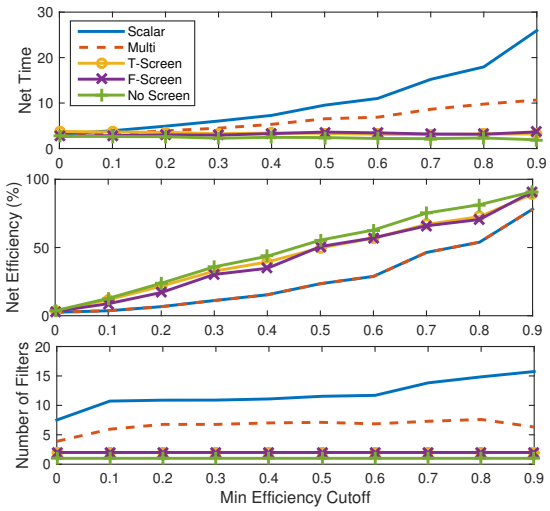

Fig. 3. Average performance of different strategies using 1000 realizations of random subsets of 20 filters, repeated using different sets of candidates restricted by different minimum-efficiency cutoffs.

Fig. 4 gives results for similar experiments, but with filters chosen from a pool with an increasingly-higher maximum efficiency cutoff. So to the left of the plot, only the slowest-but-most-efficient filters are allowed, while the pool of candidates includes a wider range of efficiency/speed trade-offs

towards the right of the plots. Here we see somewhat closer performance between all methods in terms of both time and efficiency, though the inferior efficiency of the simple methods becomes increasingly significant towards the right of the plots. Again the multidimensional method is faster than the scalar method, though the advantage is smaller.

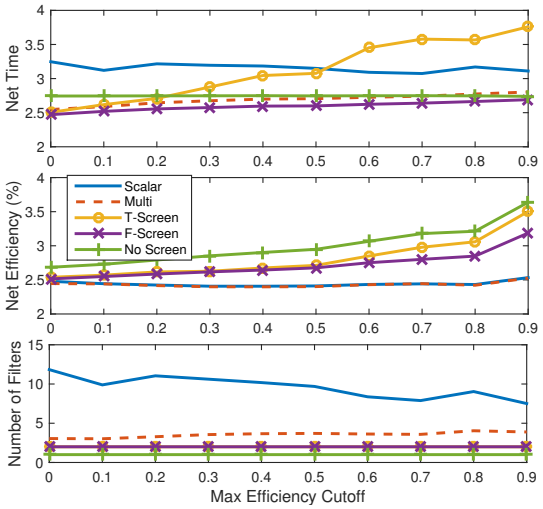

Fig. 4. Average performance of different strategies using 1000 realizations of random subsets of 20 filters, repeated using different sets of candidates restricted by different maximum-efficiency cutoffs.

Roughly similar results are seen in Fig. 5, where we varied only the size of candidate filter sets. Hence towards the right of the plots, there was a larger set of candidates for each realization. We see that this leads to an increasingly-superior relative performance for the multidimensional strategy, which on average, came to dominate all other strategies except the F-Screen in terms of speed (while remaining superior in terms of efficiency). Smaller candidate sets chosen randomly from the entire pool of 250 are more likely to be independent, hence we see the relatively-improved performance of the scalar strategy towards the left of the net time plots.

IV. Discussion

In this paper we developed a framework for forming meta-filters, based on composing multiple filters intelligently. Our goal is to support new avenues of development for computationally-intensive methods, which typically employ only single-stage screening. In the most direct sense, employing such "meta-strategies" provide the potential for significantly-improved performance. For example in our experiments, by (at-worst) roughly doubling the time spent in the screening stage (the net time taken as compared to the simple methods) we can improve the efficiency significantly. This means a subsequent stage utilizing an extremely complex and resource-intensive method becomes usable where it may not have been before. In a more indirect sense, these kinds of strategies may also provide for improvements in terms of reduced time in development as well as greater "future proofing", or increased lifespan, of methods. In the conventional research culture, each screening method

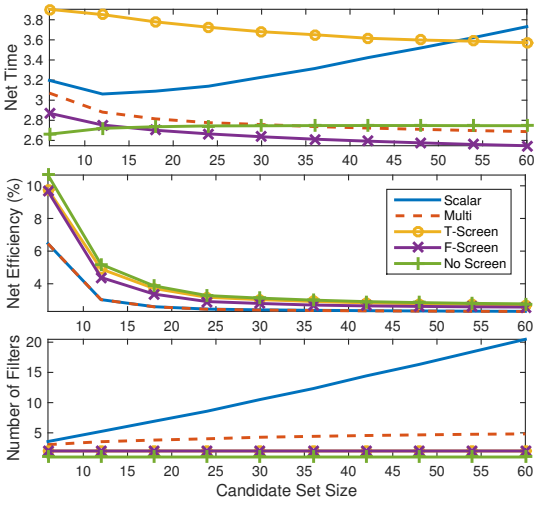

Fig. 5. Average performance of different strategies using 1000 realizations of random subsets of filters, repeated using different sizes for sets of candidate filters.

becomes obsolete and discarded as the subsequent, better method, is introduced. Given the pace of development, this may occur before the earlier method even reaches final publication.

References

[1] Nuno A. Fonseca, Johan Rung, Alvis Brazma, and John C. Marioni. Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–3177, December 2012.

[2] http://www.ebi.ac.uk/~nf/hts_mappers/.

[3] Honghui Yang, Jingyu Liu, Jing Sui, Godfrey Pearlson, and Vince D. Calhoun. A hybrid machine learning method for fusing fMRI and genetic data: combining both improves classification of schizophrenia. *Frontiers in Human Neuroscience*, 4:192, 2010.

[4] Hongbao Cao, Junbo Duan, Dongdong Lin, Yin Yao Shugart, Vince Calhoun, and Yu-Ping Wang. Sparse representation based biomarker selection for schizophrenia with integrated analysis of fMRI and SNPs. *NeuroImage*, 102 Pt 1:220–228, November 2014.

[5] Dongdong Lin, Hongbao Cao, Vince D. Calhoun, and Yu-Ping Wang. Sparse models for correlative and integrative analysis of imaging and genetic data. *Journal of Neuroscience Methods*, 237:69–78, November 2014.

[6] Isabelle Guyon and Andr Elisseeff. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[7] Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, 9(5):392–403, September 2008.

[8] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[9] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, November 2008.

[10] Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, March 2012.

[11] Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems. *arXiv:1009.4219 [cs, math]*, September 2010. arXiv: 1009.4219.

[12] Shaojie Zhang, Ilya Borovok, Yair Aharonowitz, Roded Sharan, and Vineet Bafna. A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements. *Bioinformatics*, 22(14):e557–e565, July 2006.