Multiple-kernel learning for genomic data mining and prediction

Christopher M. Wilson[1], Kaiqiao Li[2], Xiaoqing Yu[1], Pei-Fen Kuan[2] and Xuefeng Wang[1*]

Abstract

Background: Advances in medical technology have allowed for customized prognosis, diagnosis, and treatment regimens that utilize multiple heterogeneous data sources. Multiple kernel learning (MKL) is well suited for the integration of multiple high throughput data sources. MKL remains to be under-utilized by genomic researchers partly due to the lack of unified guidelines for its use, and benchmark genomic datasets.

Results: We provide three implementations of MKL in R. These methods are applied to simulated data to illustrate that MKL can select appropriate models. We also apply MKL to combine clinical information with miRNA gene expression data of ovarian cancer study into a single analysis. Lastly, we show that MKL can identify gene sets that are known to play a role in the prognostic prediction of 15 cancer types using gene expression data from The Cancer Genome Atlas, as well as, identify new gene sets for the future research.

Conclusion: Multiple kernel learning coupled with modern optimization techniques provides a promising learning tool for building predictive models based on multi-source genomic data. MKL also provides an automated scheme for kernel prioritization and parameter tuning. The methods used in the paper are implemented as an R package called RMKL package, which is freely available for download through CRAN at https://CRAN.R-project.org/package=RMKL.

Keywords: Classification, Multiple kernel learning, Genomics, Data integration, Machine learning, Kernel methods

Background
Motivation

Data integration is an emerging topic of interest in cancer research. Making decisions based upon metabolomic, genomic, etc. data sources can lead to better prognosis or diagnosis than using clinical data alone. Though data sources may have different background noise levels, formats, and biological interpretations, a framework for integrating data of similar and heterogeneous types has been proposed [1]. Classification or prediction based on data from a single high throughput source may require machine learning techniques since the number of genes or metabolites will inevitably be larger than the number of samples. Both supervised and unsupervised machine learning methods have been successfully utilized for classification [2–4], regression [5, 6], and identification of

latent batch effects [7, 8]. In this paper, we focus on supervised classification of dichotomized survival outcome for various cancer types, specifically discussing support vector machines and multiple kernel learning.

Support vector machines

Support vector machines (SVMs) were originally proposed to find a hyperplane such that two classes of data are on different sides of the hyperplane and have the maximal distance between the two classes. There have been several improvements presented for SVM such as adding additional constraints to the optimization problem that allow the problem to be feasible when the two classes are not perfectly separable. An additional term is added to the objective function that penalizes misclassified samples, this resulting formulation is known as soft-margin [9].

A second major improvement to SVM is applying the kernel trick to allow for a non-linear classification rule. Kernel functions are used to provide different similarity measures between samples. The correlation (dot product) matrix is used to find a linear classifier. Other common kernels are polynomial kernels and radial kernels

*Correspondence: xuefeng.wang@moffitt.org
[1]Department of Biostatistics and Bioinformatics at Moffitt Cancer Center, Tampa, FL, USA
Full list of author information is available at the end of the article

for continuous features [10]. Moreover, kernels have been proposed for nominal and ordinal data, hence we can construct kernels based on demographic characteristics (race, gender, height, age, etc.) [11]. Below are formulas for different similarities between two samples x and y:

- Linear:$K(x,y) = <x,y> = x^T y$,   (1a)

- Polynomial:$K(x,y) = (v<x,y> + offset)^a$,   (1b)

- Radial:$K(x,y) = \exp(-\sigma||x-y||^2/2)$,   (1c)

- Clinical nominal:$K(x,y) = \begin{cases} 1, & \text{if } x = y, \\ 0, & \text{if } x \neq y, \end{cases}$ (1d)

- Clinical ordinal:$K(x,y) = \dfrac{r - |x-y|}{r}$,   (1e)

where a is the degree and v is the coefficient of the highest order term of an a degree polynomial, σ controls smoothness of the decision boundary for a radial kernel, and r is the range of the ordinal levels. We use the parameterization found in the kernlab R package for the linear, polynomial and radial kernels [12].

Kernel methods are attractive because they do not make parametric assumptions to construct a model, for instance, these methods are not sensitive to outliers and are distribution-free [13]. Unfortunately, the solutions can be sensitive to the choice of parameter and there is no universal best set of parameters for a given data type. Typically, cross-validation is used to identify the parameter that provides the highest prediction accuracy. Ultimately, there may not be one single optimal kernel, but a combination of kernels may provide a better classifier than a single kernel. It can be shown that the sum, product, and convex combination of kernels yields another kernel [14]. This leads to an opportunity to construct a classifier using a convex combination of candidate kernels.

Multiple kernel learning

Multiple kernel learning (MKL) algorithms aim to find the best convex combination of a set of kernels to form the best classifier. Many algorithms have been presented in recent years and they form two classes. First, wrapper methods solve MKL by first solving a single SVM problem for a given set of kernel weights, and then they update the kernel weights. Since wrapper functions rely on solving SVM, they appeal to existing well-developed solvers, thus they can be relatively easy to implement. The second class of MKL algorithms utilize more sophisticated optimization methods to greatly reduce the number of SVM computations, allowing for them to solve the problem with a much larger number of kernels than wrapper methods. We will focus on two wrapper methods (SimpleMKL [15], Simple and Efficient MKL (SEMKL) [16]), as well as, and an example of a second class of MKL algorithms DALMKL [17].

Sparse MKL solutions do not typically outperform uniformly weighted kernels [18]. There is still great value in sparse kernel weights, specifically, the model can be easier to interpret with fewer non-zero kernel weights. Each MKL method can provide an ordering for the importance of a data type or features that may prompt investigators towards data sources that contain the most relevant information for classification. Ranking data sources can help researchers focus their studies on gene/metabolite sets or the data types that are most likely to lead to meaningful results.

Several studies have applied MKL to genomic data. An extensive comparison of regression techniques, including support vector regression (SVR) and Bayesian multitask MKL, has been conducted to predict drug sensitivity using six genomic, epigenomic, and proteomic profiling datasets for human breast cancer cell lines [19]. Bayesian multitask MKL involves the selection of priors and different selection of priors can lead to a dramatically different result. MKL was also implemented to predict survival at 2000 days from diagnosis, using the METABRIC dataset for breast cancer, and observed that predictive accuracy can be increased by grouping genes within a pathway into a single kernel [20]. These papers illustrate that MKL can be effectively applied to data that is from multiple sources and how it can be used to analysis high dimensional data, however, MKL remains an under-utilized tool for genomic data mining. This article aims to bridge these gaps by providing a unified survey of MKL methodology, highlighting its unique benefits in tackling challenges in large-scale omics data analysis, and establishing benchmarked models for further algorithm development.

This paper is organized as follows. "Implementation" section discusses practical issues when conducting MKL, and describes the features offered our package RMKL. "Results" section describes the results from one experiment which uses simulated data, and two experiments that use real data from The Cancer Genome Atlas (TCGA). Lastly, in "Conclusion" section, we make observations regarding our results and mention several areas for future work.

Implementation

SimpleMKL uses subgradient descent to find the direction has the most improvement. Then it uses lines search to find the optimal set of kernel weights. For each candidate set of kernel weights, SimpleMKL must solve an SVM problem iteratively along the vector of maximum improvement. SEMKL can decrease the computational burden dramatically by updating the set of kernel weights with an explicit formula derived using the Cauchy-Schwarz inequality as opposed to using line search.

DALMKL optimizes the dual augmented Lagrangian of a proximal formulation of the MKL problem. This formulation presents a unique set of problems such as the conjugate of a loss function must have no non-differentiable points in the interior of its domain and cannot have a finite gradient at the boundary of its domain. The inner function is differentiable and the gradient and Hessian only depend on the active kernels making gradient descent efficient. DALMKL is written in C++, and uses Newton descent to update the kernel weights. A flowchart describing how general wrapper methods and DALMKL are implemented can be found in Additional file 1: Figure S2.

There are many considerations that must be made before conducting SVM or any MKL algorithm. One of the most important is the prioritization of features and kernels. Even though SVM does not deal with each feature directly it can suffer from the curse of dimensionality. If there are a large number of features and a very small number of features can separate the data, then SVM will not necessarily find the best subspace that separates the data. Feature prioritization can improve the accuracy of SVM [10, 20]. Features can be prioritized by determining which features have the biggest effect size or smallest p-value from a *t*-test or more robust two group comparisons such as the Wilcoxon rank-sum test.

Kernel prioritization is important to alleviate many potential problems for MKL. For instance, if many kernels share a lot of redundant information then the efficiency of MKL can greatly diminish since many wrapper methods seek a sparse combination of kernels. Kernels that can both classify the data and yet provide different boundaries, similar to ensemble learning. A potential method for prioritizing kernels is to conduct SVM with each candidate kernel, then determine the kernels with the largest accuracy or eliminate the kernels with accuracy lower than the no information rate. Figure 1 summarizes the workflow we use and recommend for the implementation of MKL. There has been work using minimal redundancy maximal relevance criteria, and kernel alignment to remove kernels that share too similar [21].

We present an R package, RMKL, which can implement cross-validation for training SVM and support vector regression models, as well as MKL for both classification and regression problems. Our package is equipped with implementations of SimpleMKL, SEMKL, and DALMKL under two loss functions. We demonstrate each of these three implementations in simulated and real data to compare their performance. RMKL is freely available for download through CRAN at https://CRAN.R-project.org/package=RMKL. Next, we further discuss the features of RMKL.

There are several features in RMKL that aim to make the implementation of MKL easier. For instance, we provide a wrapper function to compute kernel matrices which

Fig. 1 Recommended workflow for an MKL experiment
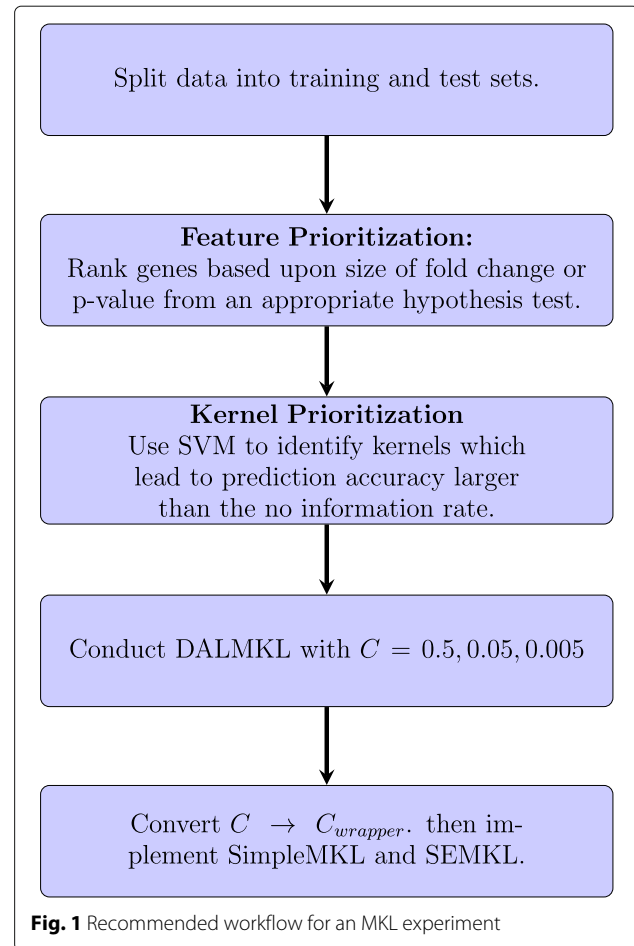
can provide kernels for training and test set. Another convenient function in RMKL is a wrapper function for conducting cross-validation for SVM. A challenge of MKL wrapper methods is that there are no guidelines for selecting the penalty parameter. Fortunately, there are recommended values for the penalty parameter (0.5, 0.05, and 0.005) for DALMKL. Unfortunately, direct comparisons between SimpleMKL, SEMKL, and DALMKL are not possible using the same cost parameter in all three implementations. We provide a function that uses the solution of DALMKL to estimate for a comparable cost parameter for SimpleMKL and SEMKL.

Results

Benchmark example

In addition to accuracy, an important characteristic of MKL is the learning of kernel weights. In this example, 9 datasets are generated with two groups and the amount overlap between the two groups varies. The two groups have 50 observations from a bivariate normal distribution where the mean of group 1 was fixed at (5,5), and the means of group 2 were {(-4,-4),(-3,-3), ..., (4,4)}. The covariance structure of the two groups were

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } \Sigma_2 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}.$$

If the two groups do not overlap, then we expect a radial kernel with a small scale parameter to have a larger weight than a radial kernel with a larger scale parameter (see kernel parameterization in the kernlab R package), leading to a smooth boundary. On the other hand, if there is a large amount of overlap between the groups, we expect lower accuracy and a less smooth classification rule. Thus a larger scale hyperparameter should be preferred.

We consider two radial kernels, denoted K_1 and K_2, with hyperparameters $\sigma_1 = 2$ and $\sigma_2 = 0.04$. In Fig. 2a, notice that as the amount of overlap between the two groups increases the weight for K_1 increases. This yields a classification rule that is less smooth and can accommodate for the overlapping groups. When there is little overlap between the groups, we see that K_2 is given much more weight than K_1, leading to a smooth classification rule for perfectly separable data All algorithms can classify perfectly when there is no overlap, but when the groups are completely overlapping, the prediction accuracy of each algorithm is approximately 0.5 (Fig. 2b).

TCGA ovarian
Bell et al. (2011) provide integrative analyses of The Cancer Genome Atlas (TCGA) ovarian cancer dataset [22]. Survivorship for ovarian cancer is difficult to predict from clinical information only, which is limited since most cancers are late stage. Information from high throughput data sources can be utilized to increase prediction accuracy. To illustrate MKL as a data integration tool, we perform MKL to find the best kernel for clinical and miRNA gene expression data separately, and then combine them into a single analysis. Our goal is to predict if a patient will live longer than three years after diagnosis and patients who were right-censored were not considered.

There are 283 samples in this dataset. We used 70% (198) as the train samples and 30% (85) as a test set. For all kernel and variable prioritization, only the training set was used, and then the final classification accuracy of MKL was computed for the final MKL model. Candidates for the clinical kernels were constructed using kernels for stage and age, and the average of these two as a kernel. To avoid the curse of dimensionality, we include the 65 top-ranked genes, based on p-value from testing for differences in mean expression for patients who survived more than 3 years and those who did not. We used these 65 genes to conduct SVM with 10 fold cross-validation for many several radial kernels ($\sigma = 10^{-10}, \ldots, 10^{10}$) to identify the range that leads to the highest predictive accuracy. Ultimately in our MKL analysis, we used a linear kernel, and 3 radial kernels with $\sigma = 10^{-4}, 10^{-3}, 10^{-2}$. Surprisingly, using miRNA data only has similar prediction accuracy as clinical information only, but using both data sources leads to a substantially higher accuracy than either of the individual data sources (Fig. 3).
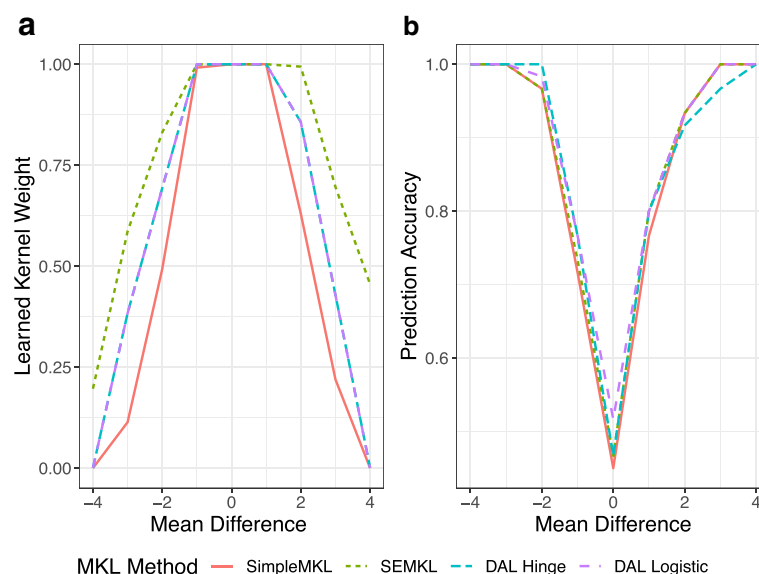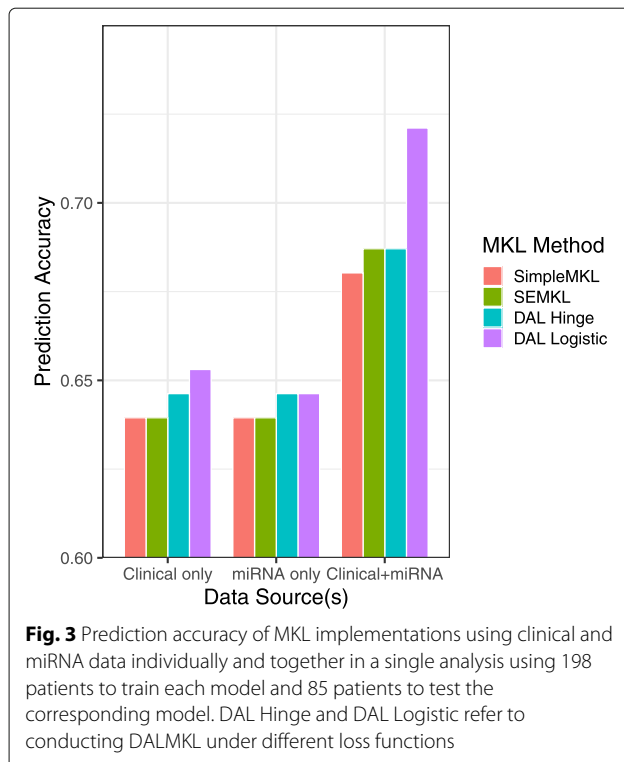

Fig. 2 Results from SEMKL, SimpleMKL, and DAMKL on 9 benchmark datasets, where two radial kernels K_1 and K_2 with $\sigma_1 = 2$ and $\sigma_2 = 0.05$ were used. **a** Displays the learned kernel weight of K_1 as the mean of each group changes. **b** Displays the predictive accuracy of each algorithm as the distance between each group changes. DAL Hinge and DAL Logistic refer to conducting DALMKL under different loss functions

Fig. 3 Prediction accuracy of MKL implementations using clinical and miRNA data individually and together in a single analysis using 198 patients to train each model and 85 patients to test the corresponding model. DAL Hinge and DAL Logistic refer to conducting DALMKL under different loss functions

Hoadley data

Hoadley et al. (2018) conducted integrative molecular analyses for all tumors in TCGA [23]. We studied 15 cancer types which were selected because they had a total of greater than 300 tumors and more than 20 events. Survival was dichotomized by a cutoff that was selected such that the proportion of patients that survived was 0.4–0.6. Patients who were right-censored were not included in the analysis. Gene expression data was used to find relationships between gene sets and our binary survival outcome. In this analysis, we focused on 50 gene sets that are included in the hallmark gene sets introduced by Liberzon et al. (2015) [24], which represent specific well-defined biological states or processes. The cancer types and survival cutoff are provided in Additional file 1: Table S1 and S2.

To identify gene sets that may aid in classification, SVM is employed with cross-validation to identify which kernel shape and hyperparameter is most suitable. Gene sets were not considered if the training accuracy was less than the no information rate (*NIR*). This occurred when SVM classified all patients into one class, typically the largest class. The remaining gene sets were introduced to MKL using their shape and hyperparameter that leads to the highest accuracy in SVM. Details for how many gene sets were included for MKL are in Additional file 1: Table S3. Gene sets that have significant importance can be areas for future study.

In Fig. 4 and Additional file 1: Figure S3, we see that kernel weights are similar in SEMKL, and DALMKL (both hinge and logistic loss), while SimpleMKL is quite different and is often times less sparse than other methods. Additional file 1: Table S3 displays the prediction accuracy for each method. DALMKL tends to be the most accurate. There are cases, such as ovarian cancer (OV), where SimpleMKL allocates weight more evenly across the gene sets and can achieve a significant increase in accuracy. On the other hand, when all methods only consider a small number of gene sets SimpleMKL performs the worst.

The pan-cancer pathway analysis revealed multiple gene sets that carry important prognostic values. Interestingly, many pathways such as KRAS signaling, inflammatory response and spermatogenesis had non-zero kernel-based importance scores across many cancer types. We hope the finding will spur additional research into the role of these pathways in cancer development and prognosis especially spermatogenesis, which is less studied compared with other pathways in cancer.

Conclusion

Integrating heterogeneous data sources into a single analysis allows patients to obtain more accurate prognoses or diagnoses. MKL can construct non-linear classification without any parametric assumptions for a single or multiple data types. Additionally, MKL may not suffer from overfitting because the final decision rule is based on a weighted average of SVM models. Kernel weights from MKL can have an appealing interpretation and help identify data sources that are most important in the classifier.

There are several considerations to be made regarding which MKL algorithm to use. If a small number of kernels are used, then each of the four methods seems to have similar performance. However, if a large number of kernels are used then DALMKL should be used. Regardless of the number of kernels, SimpleMKL and SEMKL have similar run times, however, DALMKL tends to run significantly faster. There are currently no recommendations for selection of cost parameter is SimpleMKL or SEMKL, while DALMKL provides recommendations and a formula to estimate a comparable cost for wrapper methods. DALMKL should be used first to get a range of cost values for SimpleMKL or SEMKL. A drawback of DALMKL is that the parameters for the optimization problem are more complicated and therefore not easy to interpret, while only a little bit of knowledge about SVM is needed to understand the parameters in SEMKL and SimpleMKL.

The real data analyses presented in this paper are biased, i.e. the censoring mechanism was completely ignored. Also, the selection of survival threshold was
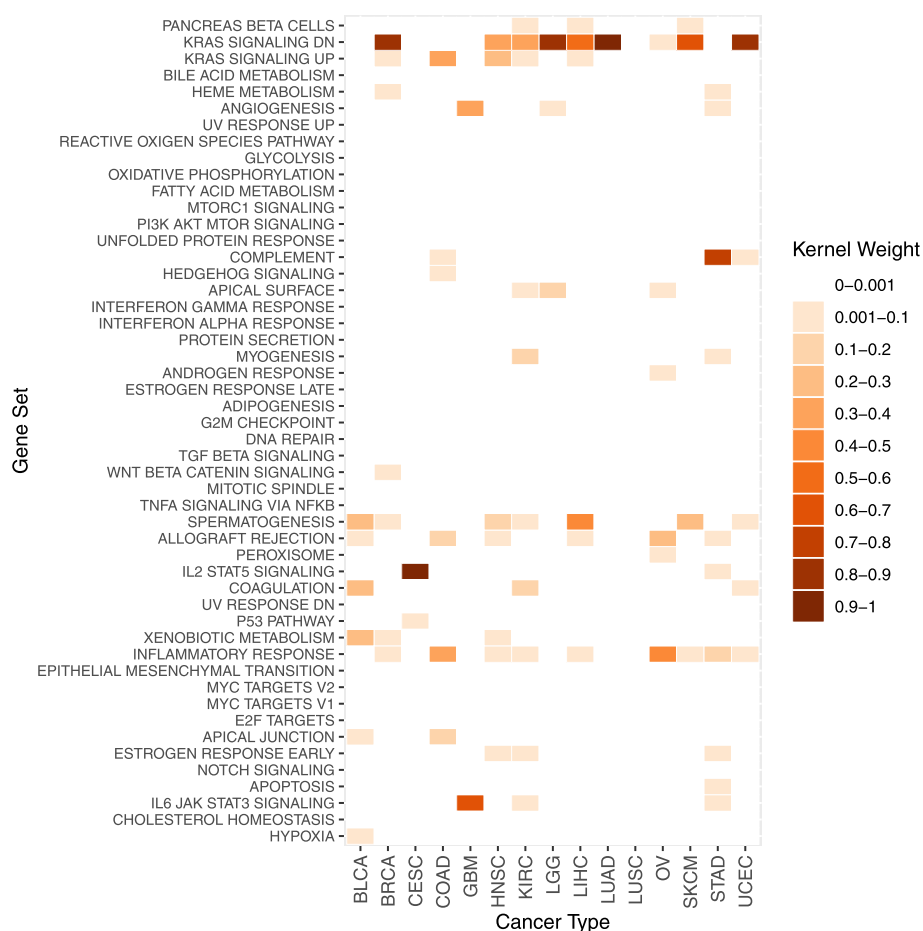
Fig. 4 Heatmap of gene set importance for each of the 15 cancer types considered. (DALMKL Logistic)

picked to provide an approximately even split of the binary outcomes potentially losing biological meaning. Extensions to MKL can be made to account for an imbalance of samples between the two groups, by modifying the objective function such that there is a different cost associated with misclassification for both classes. MKL presents opportunities to answer statistical questions. For instance, by considering different loss functions MKL can be extended to regression and survival analysis settings. The problem of missing data has been addressed for MKL [25] but there is still a lot of room for improvement. Utilizing kernel alignment is an additional step in kernel prioritization than can greatly increase the performance of MKL algorithms [21].

Additional file

Additional file 1: This file contains a brief summary of SVM and MKL, 3 tables, and 1 figure that better summarize the experiment with Hoadley data. (PDF 376 kb)

Abbreviations
DALMKL: Dual augmented lagrangian mkl; MKL: Multiple kernel learning; SEMKL: Simple and efficient MKL; SimpleMKL: Simple MKL; SVM: Support vector machine; TCGA: The cancer genome atlas

Availability and requirements
Project name: RMKL
Project home page: https://CRAN.R-project.org/package=RMKL
Operating system: Platform independent
Programming language: R
Other requirements:
License: GPL-3

Authors' contributions
XW and PK conceived the study. CMW, KL and XW designed the algorithm and implemented the software. CMW, XY and XW performed the analyses, interpreted the results and wrote the manuscript. All authors read and approved the final manuscript.

Wilson *et al. BMC Bioinformatics*   (2019) 20:426

Page 7 of 7

Author details

[1]Department of Biostatistics and Bioinformatics at Moffitt Cancer Center, Tampa, FL, USA. [2]Department of Applied Mathematics and Statistics at Stony Brook University, Stony Brook, NY, USA.

References

1. Hamid JS, Hu P, Roslin NM, Ling V, Greenwood CMT, Beyene J. Data integration in genetics and genomics: methods and challenges. Hum Genomics Proteomics HGP. 2009;2009:869093.
2. Austin E, Pan W, Shen X. Penalized regression and risk prediction in genome-wide association studies. Stat Anal Data Min ASA Data Sci J. 2013;6(4):315–28. https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11183.
3. Eetemadi A, Tagkopoulos I. Genetic Neural Networks: an artificial neural network architecture for capturing gene expression relationships. 2018. https://doi.org/10.1093/bioinformatics/bty945.
4. Kenidra B, Benmohammed M, Beghriche A, Benmounah Z. A Partitional Approach for Genomic-Data Clustering Combined with K-Means Algorithm. In: 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES). Los Alamitos: IEEE; 2016. p. 114–21.
5. Machuca C, Vettore MV, Krasuska M, Baker SR, Robinson PG. Using classification and regression tree modelling to investigate response shift patterns in dentine hypersensitivity. BMC Med Res Method. 2017;17:120.
6. Schrider DR, Kern AD. Supervised Machine Learning for Population Genetics: A New Paradigm. Trends Genet. 2018;34(4):301–12. http://www.sciencedirect.com/science/article/pii/S0168952517302251.
7. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet. 2015;16(6):321–32.
8. Reese SE, Archer KJ, Therneau TM, Atkinson EJ, Vachon CM, de Andrade M, et al. A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. Bioinformatics Oxf Engl. 2013;29(22):2877–83.
9. Cortes C, Vapnik V. Support-Vector Networks. Mach Learn. 1995;20(3): 273–97.
10. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Springer Series in Statistics. New York: Springer New York Inc.; 2001.
11. Daemen A, Moor BD. Development of a kernel function for clinical data. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Los Alamitos: IEEE; 2009. p. 5913–7.
12. Karatzoglou A, Smola A, Hornik K, Zeileis A. kernlab – An S4 Package for Kernel Methods in R. J Stat Softw. 2004;11(9):1–20. http://www.jstatsoft.org/v11/i09/.
13. Wang T, Zhao D, Tian S. An overview of kernel alignment and its applications. Artif Intell Rev. 2;43:179–92. https://doi.org/10.1007/s10462-012-9369-4.
14. Hofmann T, Schölkopf B, Smola AJ. Kernel methods in machine learning. Ann Stat. 2008;36(3):1171–220. https://doi.org/10.1214/009053607000000677.
15. Rakotomamonjy A, Bach F, Canu S, Grandvalet Y. SimpleMKL. J Mach Learn Res. 2008;9:2491–521.
16. Xu Z, Jin R, Yang H, King I, Lyu MR. Simple and Efficient Multiple Kernel Learning by Group Lasso. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10. USA: Omnipress; 2010. p. 1175–82. http://dl.acm.org/citation.cfm?id=3104322.3104471.
17. Suzuki T, Tomioka R. SpicyMKL: a fast algorithm for Multiple Kernel Learning with thousands of kernels. Mach Learn. 2011;85(1):77–108.
18. Kloft M, Brefeld U, Laskov P, Müller KR, Zien A, Sonnenburg S. Efficient and Accurate Lp-Norm Multiple Kernel Learning. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A, editors. Advances in Neural Information Processing Systems 22. New York: Curran Associates, Inc.; 2009. p. 997–1005. http://papers.nips.cc/paper/3675-efficient-and-accurate-lp-norm-multiple-kernel-learning.pdf.
19. Costello JC, Heiser LM, Georgii E, Gönen M, Menden MP, Wang NJ, et al. A community effort to assess and improve drug sensitivity prediction algorithms. Nat Biotechnol. 2014;32(12):1202–12.
20. Seoane J, Day INM, Gaunt TR, Campbell C. A pathway-based data integration framework for prediction of disease progression. 2014;30(6): 838–45.
21. Wu TT, Chen YF, Hastie T, Sobel E, Lange K. Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics Oxf Engl. 2009;25(6):714–21.
22. Network TCGAR, Bell D, Berchuck A, Birrer M, Chien J, Cramer DW, et al. Integrated genomic analyses of ovarian carcinoma. Nature. 2011;474:609 EP –. https://doi.org/10.1038/nature10166.
23. Hoadley KA, Yau C, Hinoue T, Wolf DM, Lazar AJ, Drill E, et al. Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer. Cell. 2018;173(2):291–304.e6. https://www.ncbi.nlm.nih.gov/pubmed/29625048.
24. Liberzon A, Birger C, Thorvaldsdóttir H, Ghandi M, Mesirov JP, Tamayo P. The Molecular Signatures Database Hallmark Gene Set Collection. Cell Syst. 2015;1(6):417–25.
25. Kobayashi V, Aluja T, Muñoz LAB. Handling missing values in kernel methods with application to microbiology data. Neurocomputing. 2013;141:110–16.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.