Cichonska, Anna; Pahikkala, Tapio; Szedmak, Sandor; Julkunen, Heli; Airola, Antti; Heinonen, Markus; Aittokallio, Tero; Rousu, Juho

Learning with multiple pairwise kernels for drug bioactivity prediction

Published in:
Bioinformatics

DOI:
10.1093/bioinformatics/bty277

Published: 01/07/2018

Document Version
Publisher's PDF, also known as Version of record

Bioinformatics, 34, 2018, i509–i518
doi: 10.1093/bioinformatics/bty277
ISMB 2018

OXFORD

Learning with multiple pairwise kernels for drug bioactivity prediction

Anna Cichonska[1,2,*], Tapio Pahikkala[3], Sandor Szedmak[1], Heli Julkunen[1], Antti Airola[3], Markus Heinonen[1], Tero Aittokallio[1,2,4] and Juho Rousu[1]

[1]Department of Computer Science, Helsinki Institute for Information Technology HIIT, Aalto University, Espoo, Finland, [2]Institute for Molecular Medicine Finland FIMM, University of Helsinki, Helsinki, Finland, [3]Department of Information Technology and [4]Department of Mathematics and Statistics, University of Turku, Turku, Finland

*To whom correspondence should be addressed.

Abstract

Motivation: Many inference problems in bioinformatics, including drug bioactivity prediction, can be formulated as pairwise learning problems, in which one is interested in making predictions for pairs of objects, e.g. drugs and their targets. Kernel-based approaches have emerged as powerful tools for solving problems of that kind, and especially multiple kernel learning (MKL) offers promising benefits as it enables integrating various types of complex biomedical information sources in the form of kernels, along with learning their importance for the prediction task. However, the immense size of pairwise kernel spaces remains a major bottleneck, making the existing MKL algorithms computationally infeasible even for small number of input pairs.

Results: We introduce *pairwiseMKL*, the first method for time- and memory-efficient learning with multiple pairwise kernels. *pairwiseMKL* first determines the mixture weights of the input pairwise kernels, and then learns the pairwise prediction function. Both steps are performed efficiently without explicit computation of the massive pairwise matrices, therefore making the method applicable to solving large pairwise learning problems. We demonstrate the performance of *pairwiseMKL* in two related tasks of quantitative drug bioactivity prediction using up to 167 995 bioactivity measurements and 3120 pairwise kernels: (i) prediction of anticancer efficacy of drug compounds across a large panel of cancer cell lines; and (ii) prediction of target profiles of anticancer compounds across their kinome-wide target spaces. We show that *pairwiseMKL* provides accurate predictions using sparse solutions in terms of selected kernels, and therefore it automatically identifies also data sources relevant for the prediction problem.

Availability and implementation: Code is available at https://github.com/aalto-ics-kepaco.

Contact: anna.cichonska@aalto.fi

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In the recent years, several high-throughput anticancer drug screening efforts have been conducted (Barretina *et al.*, 2012; Smirnov *et al.*, 2018; Yang *et al.*, 2012), providing bioactivity measurements that allow for the identification of compounds that show increased efficacy in specific human cancer types or individual cell lines, therefore guiding both the precision medicine efforts as well as drug repurposing applications. However, chemical compounds execute their action through modulating typically multiple molecules, with proteins being the most common molecular targets, and ultimately both the efficacy and toxicity of the treatment are a consequence of those complex molecular interactions. Hence, elucidating drug's mode of action (MoA), including both on- and off-targets, is critical for the development of effective and safe therapies.

The increased availability of drug bioactivity data for cell lines (Smirnov *et al.*, 2018) and protein targets (Merget *et al.*, 2017), together with the comprehensive characteristics of drug compounds, proteins and cell lines, has enabled construction of supervised machine learning models, which offer cost-effective means for fast, systematic and large-scale pre-screening of chemical compounds and their potential targets for further experimental verification, with the aim of accelerating and de-risking the drug discovery process

(Ali *et al.*, 2017; Azuaje, 2017; Cheng *et al.*, 2012; Cichonska *et al.*, 2015). Under the general framework of drug bioactivity prediction, two related machine learning tasks are identified: (i) prediction of anticancer drug responses and (ii) prediction of drug–protein interactions, both of which can be tackled through similar machine learning techniques. In particular, kernel-based approaches have emerged as powerful tools in computational drug discovery (Cichonska *et al.*, 2017; Marcou *et al.*, 2016; Pahikkala *et al.*, 2015).

Both the drug response in cancer cell line prediction and drug–protein interaction prediction are representative examples of pairwise learning problems, where the goal is to build predictive model for pairs of objects. Classical kernel-based methods for pairwise learning rely merely on a single pairwise kernel. However, such approaches are unlikely to be optimal in applications where a growing variety of biological and molecular data sources are available, including chemical and protein structures, pharmacophore patterns, gene expression signatures, methylation profiles as well as genomic variants found in cell lines. In fact, the advantage of integrating different data types for the multi-level analysis has been highlighted in the recent studies (Ebrahim *et al.*, 2016; Elefsinioti *et al.*, 2016). Multiple kernel learning (MKL) methods, which search for an optimal combination of several kernels, hence enabling the use of different information sources simultaneously and learning their importance for the prediction task, have therefore received significant attention in bioinformatics (Brouard *et al.*, 2016; Kludas *et al.*, 2016; Shen *et al.*, 2014), especially in drug bioactivity inference (Ammad-ud-din *et al.*, 2016; Costello *et al.*, 2014; Nascimento *et al.*, 2016).

However, the existing MKL methods do not scale up to the massive size of pairwise kernels, in terms of both processing and memory requirements, making the kernel weights optimization and model training computationally infeasible even for small numbers of input pairs, such as drugs and cell lines or drugs and protein targets. The recently introduced *KronRLS-MKL* algorithm for pairwise learning of drug–protein interactions interleaves the optimization of the pairwise prediction function parameters with the kernel weights optimization (Nascimento *et al.*, 2016). However, it finds two sets of kernel weights, separately for drug kernels and protein kernels instead of pairwise kernels, and therefore it does not fully exploit the information contained in the pairwise space.

Here, we propose *pairwiseMKL*, to our knowledge, the first method for time- and memory-efficient learning with multiple pairwise kernels, implementing both efficient pairwise kernel weights optimization and pairwise model training. In the first phase, the algorithm determines a convex combination of input pairwise kernels by maximizing the centered alignment (i.e. matrix similarity measure) between the final combined kernel and the *ideal* kernel derived from the label values (response kernel); in the second phase, the pairwise prediction function is learned. Both steps are performed without explicit construction of the massive pairwise matrices (Fig. 1). We demonstrate the performance of *pairwiseMKL* in two important subtasks of quantitative drug bioactivity prediction. In case of drug response in cancer cell line prediction subtask, we used the bioactivity data from 15 376 drug–cell line pairs from the Genomics of Drug Sensitivity in Cancer (GDSC) project (Yang *et al.*, 2012). We encoded similarities between the drug compounds and cell lines using kernels constructed based on various types of molecular fingerprints, gene expression profiles, methylation patterns, copy number data and genetic variants, resulting in 120 pairwise kernels (10 drug kernels $\times$ 12 cell line kernels). In the larger subtask of drug–protein binding affinity prediction, we used the recently published bioactivities from 167 995 drug–protein pairs (Merget *et al.*, 2017) and constructed 3120 pairwise kernels (10 drug kernels $\times$ 312 protein kernels) based on molecular fingerprints, protein sequences and gene ontology annotations. We show that *pairwiseMKL* is very well-suited for solving large pairwise learning problems, it outperforms *KronRLS-MKL* in terms of both memory requirements and predictive power, and, unlike *KronRLS-MKL*, it (i) allows for missing values in the label matrix and (ii) finds a sparse combination of input pairwise kernels, thus enabling automatic identification of data sources most relevant for the prediction task. Moreover, since *pairwiseMKL* scales up to large number of pairwise kernels, tuning of the kernel hyperparameters can be easily incorporated into the kernel weights optimization process.

In summary, this article makes the following contributions.

- We implement a highly efficient centered kernel alignment procedure to avoid explicit computation of multiple huge pairwise matrices in the selection of mixture weights of input pairwise kernels. To achieve this, we propose a novel Kronecker decomposition of the centering operator for the pairwise kernel.
- We introduce a Gaussian response kernel which is more suitable for the kernel alignment in a regression setting than a standard linear response kernel.
- We introduce a method for training a regularized least-squares model with multiple pairwise kernels by exploiting the structure of the weighted sum of Kronecker products. We therefore avoid explicit construction of any massive pairwise matrices also in the second stage of learning pairwise prediction function.
- We show how to effectively utilize the whole exome sequencing data to calculate informative real-valued genetic mutation profile feature vectors for cancer cell lines, instead of binary mutation status vectors commonly used in drug response prediction models.
- *pairwiseMKL* provides a general approach to MKL in pairwise spaces, and therefore it is widely applicable also outside the drug bioactivity inference problems. Our implementation is freely available.

2 Materials and methods

This section is organized as follows. First, Section 2.1 explains a general approach to two-stage multiple pairwise kernel regression which forms the basis for our *pairwiseMKL* method described in Section 2.2. We demonstrate the performance of *pairwiseMKL* in the two tasks of (i) anticancer drug potential prediction and (ii) drug–protein binding affinity prediction, but we selected the former as an example to explain the methodology. Finally, Section 2.3 introduces the data and kernels we used in our experiments.

2.1 Multiple pairwise kernel regression

In supervised pairwise learning of anticancer drug potential, training data appears in the form $(\mathbf{x_d}, \mathbf{x_c}, y)$, where $(\mathbf{x_d}, \mathbf{x_c})$ denotes a feature representation of a pair of input objects, drug $\mathbf{x_d} \in \mathcal{X}_\mathcal{D}$ and cancer cell line $\mathbf{x_c} \in \mathcal{X}_\mathcal{C}$ (e.g. molecular fingerprint vector and gene expression profile, respectively), and $y \in \mathbb{R}$ is its associated response value (also called the label), i.e. a measurement of sensitivity of cell line $\mathbf{x_c}$ to drug $\mathbf{x_d}$. Given $N \leq n_d \times n_c$ training instances, they can be represented as matrices $\mathbf{X_d} \in \mathbb{R}^{n_d \times t_d}$, $\mathbf{X_c} \in \mathbb{R}^{n_c \times t_c}$ and label vector $\mathbf{y} \in \mathbb{R}^N$, where n_d denotes the number of drugs, n_c the number of cell lines, t_d and t_c the number of drug and cell line features, respectively.

The aim is to find a pairwise prediction function f that models the relationship between $(\mathbf{X_d}, \mathbf{X_c})$ and $\mathbf{y}$; f can later be used to predict sensitivity measurements for drug–cell line pairs outside the training space. The assumption is that structurally similar drugs show similar effects in cell lines having common genomic backgrounds. We apply kernels to

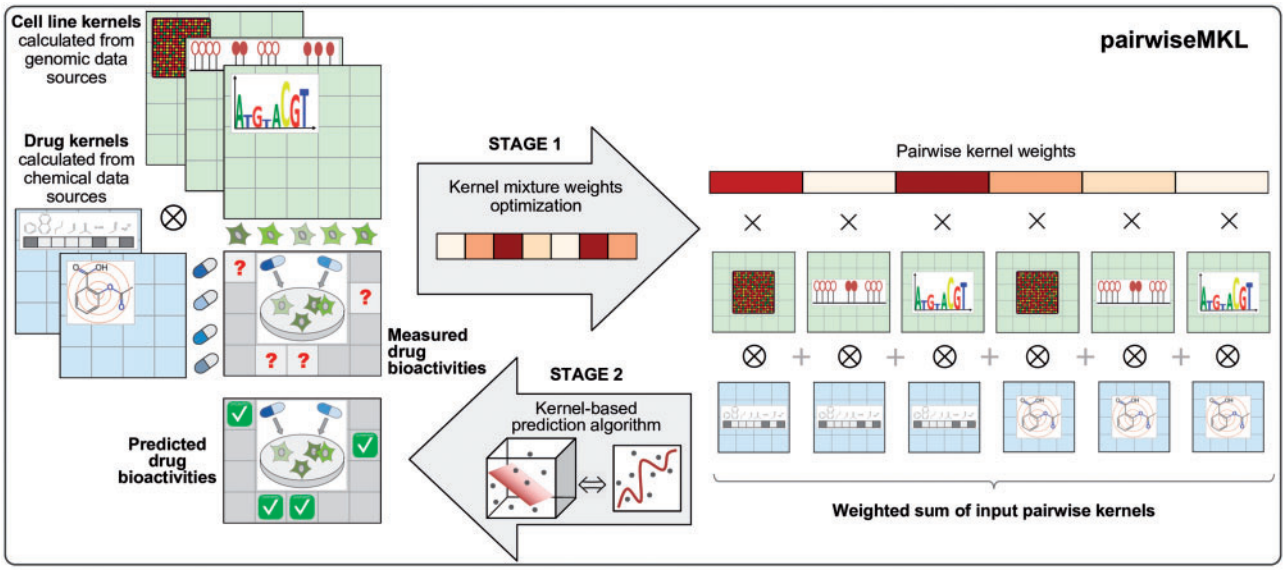


Fig. 1. Schematic figure showing an overview of *pairwiseMKL* method for learning with multiple pairwise kernels, using the drug response in cancer cell line prediction as an example. First, two drug kernels and three cell line kernels are calculated from available chemical and genomic data sources, respectively. The resulting matrices associate all drugs and all cell lines, and therefore a kernel can be considered as a similarity measure. Since we are interested in learning bioactivities of pairs of input objects, here drug–cell line pairs, pairwise kernels relating all drug–cell line pairs are needed, and they are calculated as Kronecker products ($\otimes$) of drug kernels and cell line kernels (2 drug kernels $\times$ 3 cell line kernels = 6 pairwise kernels). In the first learning stage, pairwise kernel mixture weights are determined (Section 2.2.1), and then a weighted combination of pairwise kernels is used for anticancer drug response prediction with a regularized least-squares pairwise regression model (Section 2.2.2). Importantly, *pairwiseMKL* performs those two steps efficiently by avoiding explicit construction of any massive pairwise matrices, and therefore it is very well-suited for solving large pairwise learning problems

encode the similarities between input objects, such as drugs or cell lines. Kernels offer the advantage of increasing the power of classical linear learning algorithms by providing a computationally efficient approach for projecting input objects into a new feature space with very high or even infinite number of dimensions. A linear model in this implicit feature space corresponds to a non-linear model in the original space (Shawe-Taylor and Cristianini, 2004). Formally, a kernel is a positive semidefinite (PSD) function that for all $\mathbf{x_d}, \mathbf{x'_d} \in \mathcal{X}_\mathcal{D}$ satisfies $k_d(\mathbf{x_d}, \mathbf{x'_d}) = \langle \boldsymbol{\phi}(\mathbf{x_d}), \boldsymbol{\phi}(\mathbf{x'_d}) \rangle$, where $\boldsymbol{\phi}$ denotes a mapping from the input space $\mathcal{X}_\mathcal{D}$ to a high-dimensional inner product feature space $\mathscr{H}_\mathcal{D}$, i.e. $\boldsymbol{\phi} : \mathbf{x_d} \in \mathcal{X}_\mathcal{D} \rightarrow \boldsymbol{\phi}(\mathbf{x_d}) \in \mathscr{H}_\mathcal{D}$ (the same holds for cell line kernel k_c). It is, however, possible to avoid explicit computation of the mapping $\boldsymbol{\phi}$ and define the kernel directly in terms of the original input features, such as gene expression profiles, by replacing the inner product $\langle \cdot, \cdot \rangle$ with an appropriately chosen kernel function (so-called kernel trick), e.g. the Gaussian kernel (Shawe-Taylor and Cristianini, 2004).

Kernels can be easily employed for pairwise learning by constructing a pairwise kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ relating all drug–cell line pairs. Specifically, $\mathbf{K}$ is calculated as a Kronecker product of drug kernel $\mathbf{K_d} \in \mathbb{R}^{n_d \times n_d}$ (computed from, e.g. drug fingerprints) and cell line kernel $\mathbf{K_c} \in \mathbb{R}^{n_c \times n_c}$ (computed from, e.g. gene expression), forming a block matrix with all possible products of entries of $\mathbf{K_d}$ and $\mathbf{K_c}$:

$$\mathbf{K} = \mathbf{K_d} \otimes \mathbf{K_c} = \begin{pmatrix} k_d(\mathbf{x}_{\mathbf{d}1}, \mathbf{x}_{\mathbf{d}1})\mathbf{K_c} & k_d(\mathbf{x}_{\mathbf{d}1}, \mathbf{x}_{\mathbf{d}2})\mathbf{K_c} & \cdots & k_d(\mathbf{x}_{\mathbf{d}1}, \mathbf{x}_{\mathbf{d}n_d})\mathbf{K_c} \\ k_d(\mathbf{x}_{\mathbf{d}2}, \mathbf{x}_{\mathbf{d}1})\mathbf{K_c} & k_d(\mathbf{x}_{\mathbf{d}2}, \mathbf{x}_{\mathbf{d}2})\mathbf{K_c} & \cdots & k_d(\mathbf{x}_{\mathbf{d}2}, \mathbf{x}_{\mathbf{d}n_d})\mathbf{K_c} \\ \vdots & \vdots & \ddots & \vdots \\ k_d(\mathbf{x}_{\mathbf{d}n_d}, \mathbf{x}_{\mathbf{d}1})\mathbf{K_c} & k_d(\mathbf{x}_{\mathbf{d}n_d}, \mathbf{x}_{\mathbf{d}2})\mathbf{K_c} & \cdots & k_d(\mathbf{x}_{\mathbf{d}n_d}, \mathbf{x}_{\mathbf{d}n_d})\mathbf{K_c} \end{pmatrix}. \quad (1)$$

Then, the prediction function for a test pair $(\mathbf{x_d}, \mathbf{x_c})$ is expressed as

$$f(\mathbf{x_d}, \mathbf{x_c}) = \sum_{l=1}^{N} \alpha_l k((\mathbf{x}_{\mathbf{d}l}, \mathbf{x}_{\mathbf{c}l}), (\mathbf{x_d}, \mathbf{x_c})) = \boldsymbol{\alpha}^T \mathbf{k}, \quad (2)$$

where $\mathbf{k}$ is a column vector with kernel values between each training drug–cell line pair $(\mathbf{x}_{\mathbf{d}l}, \mathbf{x}_{\mathbf{c}l})$ and test pair $(\mathbf{x_d}, \mathbf{x_c})$ for which the prediction is made, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ denotes a vector of model parameters to be obtained by the learning algorithm through minimizing a certain objective function. In kernel ridge regression (KRR, Saunders *et al.*, 1998), the objective function is defined in terms of total squared loss along with L2-norm regularizer, and the solution for $\boldsymbol{\alpha}$ is found by solving the following system of linear equations:

$$(\mathbf{K} + \lambda \mathbf{I})\boldsymbol{\alpha} = \mathbf{y}, \quad (3)$$

where λ indicates a regularization hyperparameter controlling the balance between training error and model complexity ($\lambda > 0$), and $\mathbf{I}$ is the $N \times N$ identity matrix.

Due to the wide availability of different chemical and genomic data sources, both drugs and cell lines can be represented with multiple kernel matrices $\mathbf{K}_\mathbf{d}^{(1)}, \ldots, \mathbf{K}_\mathbf{d}^{(p_d)}$ and $\mathbf{K}_\mathbf{c}^{(1)}, \ldots, \mathbf{K}_\mathbf{c}^{(p_c)}$, therefore forming $P = p_d \times p_c$ pairwise kernels $\mathbf{K}^{(1)}, \ldots, \mathbf{K}^{(P)}$ (Kronecker products of all pairs of drug kernels and cell line kernels). The goal of two-stage multiple pairwise KRR is to first find the combination of P pairwise kernels

$$\mathbf{K}_{\boldsymbol{\mu}} = \sum_{i=1}^{P} \mu_i \mathbf{K}^{(i)}, \quad (4)$$

and then use $\mathbf{K}_{\boldsymbol{\mu}}$ instead of $\mathbf{K}$ in Equation (3) to learn the pairwise prediction function.

2.1.1 Centered kernel alignment

The observation that a similarity between the centered input kernel $\mathbf{K}^{(i)}$ and a linear kernel derived from the labels $\mathbf{K}_y = \mathbf{y}\mathbf{y}^T$ (response kernel) correlates with the performance of $\mathbf{K}^{(i)}$ in a given prediction task, has inspired a design of the *centered kernel alignment*-based MKL approach (Cortes *et al.*, 2012). Both $\mathbf{K}^{(i)}$ and $\mathbf{K}_y$ measure similarities between drug–cell line pairs. However, $\mathbf{K}_y$ can be considered as a ground-truth as it is calculated from the bioactivities which we aim to predict, and hence the ideal input kernel would capture the same information about the similarities of drug–cell line pairs as the response kernel $\mathbf{K}_y$. The idea is to first learn a linear mixture of centered input kernels that is maximally aligned to the response kernel, and then use the learned mixture kernel as the input kernel for learning a prediction function.

Centering a kernel $\mathbf{K}$ corresponds to centering its associated feature mapping $\boldsymbol{\phi}$, and it is performed by $\widehat{\mathbf{K}} = \mathbf{CKC}$, where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is an idempotent ($\mathbf{C} = \mathbf{CC}$) centering operator of the form $\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right]$, $\mathbf{I}$ indicates $N \times N$ identity matrix, and $\mathbf{1}$ is a vector of N components, all equal to 1 (Cortes *et al.*, 2012). Centered kernel alignment measures the similarity between two kernels $\widehat{\mathbf{K}}$ and $\widehat{\mathbf{K}}'$:

$$A\left(\widehat{\mathbf{K}}, \widehat{\mathbf{K}}'\right) = \frac{\langle \widehat{\mathbf{K}}, \widehat{\mathbf{K}}' \rangle_F}{\sqrt{\langle \widehat{\mathbf{K}}, \widehat{\mathbf{K}} \rangle_F \langle \widehat{\mathbf{K}}', \widehat{\mathbf{K}}' \rangle_F}} = \frac{\langle \widehat{\mathbf{K}}, \widehat{\mathbf{K}}' \rangle_F}{||\widehat{\mathbf{K}}||_F ||\widehat{\mathbf{K}}'||_F}. \tag{5}$$

Above, $\langle \cdot, \cdot \rangle_F$ denotes a Frobenius inner product, $|| \cdot ||_F$ is a Frobenius norm and A can be viewed as the cosine of the angle, correlation, defined between two matrices. Kernel mixture weights $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P)$ are determined by maximizing the centered alignment between the final combined kernel $\mathbf{K}_{\boldsymbol{\mu}}$ and the response kernel $\mathbf{K}_y$ (Cortes *et al.*, 2012):

$$\max_{\boldsymbol{\mu}} A\left(\widehat{\mathbf{K}}_{\boldsymbol{\mu}}, \widehat{\mathbf{K}}_y\right) = \max_{\boldsymbol{\mu}} \frac{\langle \widehat{\mathbf{K}}_{\boldsymbol{\mu}}, \mathbf{K}_y \rangle_F}{||\widehat{\mathbf{K}}_{\mu}||_F}, \quad \text{subject to : } ||\boldsymbol{\mu}||_2 = 1, \ \boldsymbol{\mu} \geq 0. \tag{6}$$

In (6), $||\widehat{\mathbf{K}}_y||_F$ is omitted because it does not depend on $\boldsymbol{\mu}$, and $\langle \widehat{\mathbf{K}}_{\boldsymbol{\mu}}, \widehat{\mathbf{K}}_y \rangle_F = \langle \widehat{\mathbf{K}}_{\boldsymbol{\mu}}, \mathbf{K}_y \rangle_F$ through the properties of centering. The optimization problem (6) can be solved via:

$$\min_{\mathbf{v} \geq 0} \mathbf{v}^T \mathbf{M} \mathbf{v} - 2\mathbf{v}^T \mathbf{a}, \tag{7}$$

with the vector $\mathbf{a}$ and the symmetric matrix $\mathbf{M}$ defined by:

$$(\mathbf{a})_i = \langle \widehat{\mathbf{K}}^{(i)}, \mathbf{K}_y \rangle_F, \qquad i = 1, \ldots, P, \tag{8}$$

$$(\mathbf{M})_{ij} = \langle \widehat{\mathbf{K}}^{(i)}, \widehat{\mathbf{K}}^{(j)} \rangle_F, \qquad i, j = 1, \ldots, P. \tag{9}$$

Optimal kernel weights are given by $\boldsymbol{\mu}^* = \mathbf{v}^*/||\mathbf{v}^*||$, where $\mathbf{v}^*$ is the solution to (7). Then, the combined kernel $\mathbf{K}_{\boldsymbol{\mu}}$ is calculated with Equation (4) and used to train a kernel-based prediction algorithm (Cortes *et al.*, 2012).

Such centered kernel alignment-based strategy has proved to have a good predictive performance (Brouard *et al.*, 2016; Cortes *et al.*, 2012; Kludas *et al.*, 2016; Shen *et al.*, 2014), but it is not applicable to most of the pairwise learning problems because the size of pairwise kernel matrices, $\mathbf{K}^{(1)}, \ldots, \mathbf{K}^{(P)}$, grows very quickly with the number of drugs and cell lines (Supplementary Fig. S1) making the mixture weights optimization procedure computationally intractable even for small number of inputs (Table 1).

For instance, given 10 kernels for 100 drugs and 10 kernels for 100 cell lines ($P = 10 \times 10 = 100$, $N = 100 \times 100 = 10\,000$ assuming that bioactivities of all combinations of drugs and cell lines are known), the computation of the matrix $\mathbf{M}$ requires $\frac{(P+1)P}{2} = 5\,050$ evaluations of Frobenius products between pairwise kernels composed of 100 million entries each, and additional 100 evaluations to calculate vector $\mathbf{a}$ (the number of evaluations increases when applying a cross validation). Given 200 drugs and 200 cell lines, the size of a single pairwise kernel grows to 1.6 billion entries taking roughly 12 GB memory. For comparison, in case of a more standard learning problem, such as drug response prediction in a single cancer cell line using drug features only, there would be 200 drugs as inputs instead of drug–cell line pairs, and the resulting kernel matrix would be composed of 40 000 elements taking 0.32 MB memory.

Table 1. Memory and time needed for a naïve MKL approach explicitly computing pairwise kernels (Section 2.1) and *pairwiseMKL* (Section 2.2), depending on the number of drugs and cell lines used in the drug bioactivity prediction experiment

Number of drugs	Number of cell lines	Memory (GB)		Time (h)	
		Naïve approach	*pairwiseMKL*	Naïve approach	*pairwiseMKL*
50	50	9.810	0.001	2.976	0.003
60	60	20.290	0.001	7.797	0.005
70	70	37.750	0.043	17.678	0.057
80	80	64.000	0.044	37.691	0.069
90	90	103.180	0.046	77.408	0.087
100	100	156.890	0.048	145.312	0.106
110	110	229.670	0.050	>168.000[a]	0.118
120	120	>256.000[b]	0.053	≫168.000	0.123

Note: A single round of 10-fold CV was run using different-sized subsets of the data on anticancer drug responses (described in Section 2.3.1) with 10 drug kernels and 12 cell line kernels. Regularization hyperparameter λ was set to 0.1 in both methods.

[a]Program did not complete within 7 days (168 h).

[b]Program did not run given 256 GB of memory.

2.2 *pairwiseMKL*

2.2.1 Stage 1: optimization of pairwise kernel weights

In this work, we devise an efficient procedure for optimizing kernel weights in pairwise learning setting. Specifically, we exploit the known identity

$$\langle \mathbf{K}_d \otimes \mathbf{K}_c, \mathbf{K}'_d \otimes \mathbf{K}'_c \rangle = \langle \mathbf{K}_d, \mathbf{K}'_d \rangle \langle \mathbf{K}_c, \mathbf{K}'_c \rangle \tag{10}$$

to avoid explicit computation of the massive Kronecker product matrices in Equations (8) and (9). The main difficulty comes from the centering of the pairwise kernel; in particular, the fact that one cannot obtain a centered pairwise kernel $\widehat{\mathbf{K}}$ simply by computing the Kronecker product of centered drug kernel $\widehat{\mathbf{K}}_d$ and cell line kernel $\widehat{\mathbf{K}}_c$, i.e. $\widehat{\mathbf{K}} \neq \widehat{\mathbf{K}}_d \otimes \widehat{\mathbf{K}}_c$.

In order to address this limitation, we introduce here a new, highly efficient Kronecker decomposition of the centering operator for the pairwise kernel:

$$\mathbf{C} = \sum_{q=1}^{2} \mathbf{Q}_d^{(q)} \otimes \mathbf{Q}_c^{(q)}, \tag{11}$$

where $\mathbf{Q}_d{}^{(q)} \in \mathbb{R}^{n_d \times n_d}$ and $\mathbf{Q}_c{}^{(q)} \in \mathbb{R}^{n_c \times n_c}$ are the factors of $\mathbf{C}$. Exploiting the structure of $\mathbf{C}$ allows us to compute the factors efficiently by solving the singular value problem for a matrix of size 2×2 only, regardless of how large N is (the detailed procedure is provided in Supplementary Material).

Decomposition (11) allows us to greatly simplify the calculation of the matrix $\mathbf{M}$ and vector $\mathbf{a}$ needed in the kernel mixture weights optimization by (7):

$$\begin{aligned}(\mathbf{M})_{ij} &= \langle \widehat{\mathbf{K}}^{(i)}, \widehat{\mathbf{K}}^{(j)} \rangle_F = \mathrm{tr}\left(\mathbf{C}\mathbf{K}^{(i)}\mathbf{C}\mathbf{K}^{(j)}\mathbf{C}\right) \\ &= \sum_{q=1}^{2}\sum_{r=1}^{2} \mathrm{tr}\left(\mathbf{Q}_\mathbf{d}^{(q)}\mathbf{K}_\mathbf{d}^{(i)}\mathbf{Q}_\mathbf{d}^{(r)}\mathbf{K}_\mathbf{d}^{(j)}\right)\mathrm{tr}\left(\mathbf{Q}_\mathbf{c}^{(q)}\mathbf{K}_\mathbf{c}^{(i)}\mathbf{Q}_\mathbf{c}^{(r)}\mathbf{K}_\mathbf{c}^{(j)}\right),\end{aligned} \tag{12}$$

with $\mathrm{tr}(\cdot)$ denoting a trace of a matrix (a full derivation is given in Supplementary Material). Hence, the inner product in the massive pairwise space ($N \times N$) is reduced to a sum of inner products in the original much smaller spaces of drugs ($n_d \times n_d$) and cell lines ($n_c \times n_c$). The computation of the elements of $\mathbf{a}$ is simplified to the inner product between two vectors by first exploiting the block structure of the Kronecker product matrix through the identity $(\mathbf{A} \otimes \mathbf{B})\mathrm{vec}(\mathbf{D}) = \mathrm{vec}\left(\mathbf{B}\mathbf{D}\mathbf{A}^T\right)$:

$$\begin{aligned}\langle \mathbf{K}^{(i)}, \mathbf{K}_\mathbf{y} \rangle_F &= \left\langle \left(\mathbf{K}_\mathbf{d}^{(i)} \otimes \mathbf{K}_\mathbf{c}^{(i)}\right),\ \mathbf{y}\mathbf{y}^T \right\rangle_F \\ &= \left\langle \mathbf{y},\ \left(\mathbf{K}_\mathbf{d}^{(i)} \otimes \mathbf{K}_\mathbf{c}^{(i)}\right)\mathbf{y} \right\rangle \\ &= \left\langle \mathbf{y},\ \mathrm{vec}\ \left(\mathbf{K}_\mathbf{c}^{(i)}\mathbf{Y}\mathbf{K}_\mathbf{d}^{(i)}\right) \right\rangle,\end{aligned} \tag{13}$$

and then accounting for the centering:

$$\begin{aligned}(\mathbf{a})_i &= \langle \widehat{\mathbf{K}}^{(i)}, \mathbf{K}_\mathbf{y} \rangle_F = \langle \mathbf{y},\ \mathbf{h} \rangle, \\ \mathbf{h} &= \sum_{q=1}^{2}\sum_{r=1}^{2} \mathrm{vec}\left(\left(\mathbf{Q}_\mathbf{c}^{(q)}\mathbf{K}_\mathbf{c}^{(i)}\mathbf{Q}_\mathbf{c}^{(r)}\right)\mathbf{Y}\left(\mathbf{Q}_\mathbf{d}^{(q)}\mathbf{K}_\mathbf{d}^{(i)}\mathbf{Q}_\mathbf{d}^{(r)}\right)\right),\end{aligned} \tag{14}$$

where $\mathbf{Y} \in \mathbb{R}^{n_c \times n_d}$ is the label matrix (if $N < n_c \times n_d$, missing values in $\mathbf{Y}$ are imputed with column (drug) averages to calculate $\mathbf{a}$), and $\mathrm{vec}(\cdot)$ is the vectorization operator which arranges the columns of a matrix into a vector, $\mathrm{vec}(\mathbf{Y}) = \mathbf{y}$.

Gaussian response kernel. The standard linear response kernel $\mathbf{y}\mathbf{y}^T$ used in Equations (13) and (14) is well-suited for measuring similarities between labels in classification tasks, where $y \in \{-1, +1\}$, but not regression, where $y \in \mathbb{R}$. *pairwiseMKL* therefore employs a Gaussian response kernel, a gold standard for measuring similarities between real numbers (Shawe-Taylor and Cristianini, 2004). In particular, we first represent each label value $y_i, i = 1, \ldots, N$, with a feature vector of length S which is a histogram corresponding to a probability density function of all the labels $\mathbf{y}$, centered at y_i, and stored as row vector in the matrix $\Psi \in \mathbb{R}^{N \times S}$. Then, the Gaussian response kernel compares the feature vectors of all pairs of labels by calculating a sum of S inner products:

$$\mathbf{K}_\mathbf{y} = \sum_{s=1}^{S} \boldsymbol{\psi}^{(s)}\boldsymbol{\psi}^{(s)T}, \tag{15}$$

where $\boldsymbol{\psi}^{(s)} \in \mathbb{R}^N$ is a column vector of Ψ.

By replacing the linear response kernel $\mathbf{y}\mathbf{y}^T$ in Equations (13) and (14) with the Gaussian response kernel defined in Equation (15), vector $\mathbf{a}$ in regression setting is calculated as a sum of S inner products between two vectors:

$$\begin{aligned}(\mathbf{a})_i &= \langle \widehat{\mathbf{K}}^{(i)}, \mathbf{K}_\mathbf{y} \rangle_F = \sum_{s=1}^{S} \langle \boldsymbol{\psi}^{(s)},\ \mathbf{w} \rangle, \\ \mathbf{w} &= \sum_{q=1}^{2}\sum_{r=1}^{2} \mathrm{vec}\left(\left(\mathbf{Q}_\mathbf{c}^{(q)}\mathbf{K}_\mathbf{c}^{(i)}\mathbf{Q}_\mathbf{c}^{(r)}\right)\mathbf{Z}\left(\mathbf{Q}_\mathbf{d}^{(q)}\mathbf{K}_\mathbf{d}^{(i)}\mathbf{Q}_\mathbf{d}^{(r)}\right)\right),\end{aligned} \tag{16}$$

where $\mathbf{Z} \in \mathbb{R}^{n_c \times n_d}$, $\mathrm{vec}(\mathbf{Z}) = \boldsymbol{\psi}^{(s)}$ [$\mathbf{Z}$ is analogous to $\mathbf{Y}$ in Equations (13) and (14)]. We used $S = 100$ in our experiments.

Taken together, *pairwiseMKL* determines pairwise kernel mixture weights $\boldsymbol{\mu}$ efficiently through (7) with the matrix $\mathbf{M}$ and vector $\mathbf{a}$ constructed by (12) and (16), respectively, without explicit calculation of massive pairwise matrices.

2.2.2 Stage 2: pairwise model training

Given pairwise kernel weights $\boldsymbol{\mu}$, Equation (3) of pairwise KRR has the following form:

$$\left(\mu_1\mathbf{K}_\mathbf{d}^{(1)} \otimes \mathbf{K}_\mathbf{c}^{(1)} + , \ldots, + \mu_P\mathbf{K}_\mathbf{d}^{(P)} \otimes \mathbf{K}_\mathbf{c}^{(P)} + \lambda\mathbf{I}\right)\boldsymbol{\alpha} = \mathbf{y}. \tag{17}$$

Since the bioactivities of all combinations of drugs and cell lines might not be known, meaning that there might be missing values in the label matrix $\mathbf{Y} \in \mathbb{R}^{n_c \times n_d}$, $\mathrm{vec}(\mathbf{Y}) = \mathbf{y}$, we further get

$$\begin{aligned}\mathbf{U}\boldsymbol{\alpha} &= \mathbf{y}, \\ \mathbf{U} &= \mathbf{B}\left(\mu_1\mathbf{K}_\mathbf{d}^{(1)} \otimes \mathbf{K}_\mathbf{c}^{(1)} + , \ldots, + \mu_P\mathbf{K}_\mathbf{d}^{(P)} \otimes \mathbf{K}_\mathbf{c}^{(P)} + \lambda\mathbf{I}\right)\mathbf{B}^T,\end{aligned} \tag{18}$$

where $\mathbf{B}$ is an indexing matrix denoting the correspondences between the rows and columns of the kernel matrix and the elements of the vector $\boldsymbol{\alpha}$: $\mathbf{B}_{i\ell} = 1$ denotes that the coefficient α_i corresponds to the ℓth row/column in the kernel matrix. Training the model, i.e. finding the parameters $\boldsymbol{\alpha}$ of the pairwise prediction function, is equivalent to solving the above system of linear equations (18). We solve the system with the conjugate gradient (CG) approach that iteratively improves the result by carrying out matrix-vector products between $\mathbf{U}$ and $\boldsymbol{\alpha}$, which in general requires a number of iterations proportional to the number of data. However, in practise one usually obtains as good or even better predictive performance with only a few iterations. Restricting the number of iterations acts as an additional regularization mechanism known in the literature as early stopping (Engl *et al*, 1996).

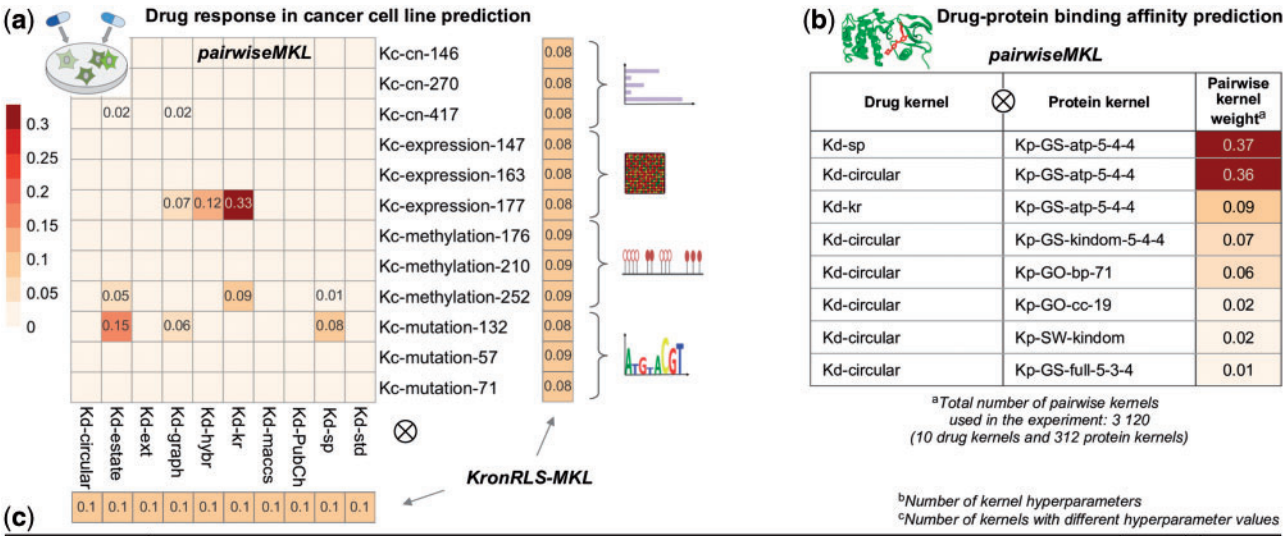
We further accelerate the matrix-vector product $\mathbf{U}\boldsymbol{\alpha}$ by taking advantage of the structural properties of the matrix $\mathbf{U}$. In (Airola and Pahikkala, 2017), we introduced the generalized vec-trick algorithm that carries out matrix–vector multiplications between a principal submatrix of a Kronecker product of type $\mathbf{B}\left(\mathbf{K}_\mathbf{d}^{(1)} \otimes \mathbf{K}_\mathbf{c}^{(1)}\right)\mathbf{B}^T$ and a vector $\boldsymbol{\alpha}$ in $O(Nn_d + Nn_c)$ time, without explicit calculation of pairwise kernel matrices. Here, we extend the algorithm to work with sums of multiple pairwise kernels, i.e. to solve the system of equations (18). In particular, the matrix $\mathbf{U}$ is a sum of P submatrices of type $\mathbf{B}\left(\mathbf{K}_\mathbf{d}^{(1)} \otimes \mathbf{K}_\mathbf{c}^{(1)}\right)\mathbf{B}^T$, and hence each iteration of CG is carried out in $O(PNn_d + PNn_c)$ time (see Supplementary Material for pseudocode and more details).

In summary, our *pairwiseMKL* avoids explicit computation of any pairwise matrices in both stages of finding pairwise kernel weights and pairwise model training, which makes the method suitable for solving problems in large pairwise spaces, such as in case of drug bioactivity prediction (Table 1).

2.3 Dataset

2.3.1 Drug bioactivity data

Drug responses in cancer cell lines. In order to test our framework, we used anticancer drug response data from GDSC project initiated by Wellcome Trust Sanger Institute (release June 2014, Yang *et al*., 2012). Our dataset consists of 124 drugs and 124 human cancer cell lines, for which complete $124 \times 124 = 15\ 376$ drug sensitivity measurements are available in the form of $\ln(\mathrm{IC}_{50})$ values in nanomolars (Ammad-ud-din *et al*., 2016).



(c)

Abbreviation		Feature description and kernel type			#Hyper[b]	#Kernels[c]
Drug kernels	$K_{d\text{-circular}}$	Extended connectivity 1 024-bit fingerprint with a maximum diameter set to 6 (ECFP6).	Binary	Fingerprint-based Tanimoto kernel	0	1
	$K_{d\text{-estate}}$	79-bit fingerprint corresponding to Estate substructures described by Hall and Kier (1995).			0	1
	$K_{d\text{-ext}}$	Path-based, hashed 1 024-bit fingerprint taking into account ring systems.			0	1
	$K_{d\text{-graph}}$	Path-based, hashed 1 024-bit fingerprint considering connectivity.			0	1
	$K_{d\text{-hybr}}$	Path-based, hashed 1 024-bit fingerprint considering hybridization states.			0	1
	$K_{d\text{-kr}}$	4 860-bit fingerprint defined by Klekota and Roth (2008).			0	1
	$K_{d\text{-maccs}}$	166-bit fingerprint based on MACCS structural keys developed by MDL Information Systems.			0	1
	$K_{d\text{-PubCh}}$	881-bit fingerprint defined by PubChem.			0	1
	$K_{d\text{-sp}}$	1 024-bit fingerprint based on the shortest paths between atoms taking into account ring systems and charges.			0	1
	$K_{d\text{-std}}$	Path-based, hashed 1 024-bit fingerprint.			0	1
Cell line kernels	$K_{c\text{-cn}}$	Copy number variation measurements of 43 255 genes.	Real-valued	Gaussian kernel	1	3
	$K_{c\text{-expression}}$	Basal gene expression measurements of 13 321 genes.			1	3
	$K_{c\text{-methylation}}$	Methylation levels of 482 892 CpG sites.			1	3
	$K_{c\text{-mutation}}$	Real-valued profiles of 12 366 somatic mutations.			1	3
Protein kernels	$K_{p\text{-SW-full}}$	Full amino acid sequences.	String	Smith-Waterman kernel	0	1
	$K_{p\text{-SW-kindom}}$	Amino acid sub-sequences of kinase domains.			0	1
	$K_{p\text{-SW-atp}}$	Amino acid sub-sequences of ATP binding pockets.			0	1
	$K_{p\text{-GS-full}}$	Full amino acid sequences + amino acid properties.	String + real-valued	Generic string kernel	3	100
	$K_{p\text{-GS-kindom}}$	Amino acid sub-sequences of kinase domains + amino acid properties.			3	100
	$K_{p\text{-GS-atp}}$	Amino acid sub-sequences of ATP binding pockets + amino acid properties.			3	100
	$K_{p\text{-GO-mf}}$	Gene ontology (GO) profile composed of 415 GO terms from *molecular function* domain.	Real-valued	Gaussian kernel	1	3
	$K_{p\text{-GO-bp}}$	GO profile composed of 3 926 GO terms from *biological process* domain.			1	3
	$K_{p\text{-GO-cc}}$	GO profile composed of 352 GO terms from *cellular component* domain.			1	3

Fig. 2. Pairwise kernel mixture weights obtained with *pairwiseMKL* and *KronRLS-MKL* (average across 10 outer CV folds) in the task of (**a**) drug response in cancer cell line prediction and (**b**) drug–protein binding affinity prediction (note: *KronRLS-MKL* did not execute with 1 TB memory); only the weights different from 0 are shown. *KronRLS-MKL* finds separate weights for drug kernels and cell line (protein) kernels instead of pairwise kernels. Numbers at the end of kernel names indicate the kernel hyperparameter values, in particular (i) kernel width hyperparameter in case of Gaussian kernels (e.g. *Kc-cn-146* with $\sigma_c = 146$), and (ii) maximum sub-string length L, σ_1 controlling for the shifting contribution term and σ_2 controlling for the amino acid similarity term in case of GS kernels (e.g. *Kp-GS-atp-5-4-4* with $L = 5, \sigma_1 = \sigma_2 = 4$, see Section 2.3.2 for details). (**c**) Summary of drug, cell line and protein kernels used in this work for the two prediction problems.

Drug–protein binding affinities. In the second task of drug–protein binding affinity prediction, we used a comprehensive kinome-wide drug–target interaction map generated by Merget *et al.* (2017) from publicly available data sources, and further updated with the bioactivities from version 22 of ChEMBL database by Sorgenfrei *et al.* (2017). Since the original interaction map is extremely sparse, we selected drugs with at least 1% of measured bioactivity values across the kinase panel, and also kinases with kinase domain and ATP binding pocket amino acid sub-sequences available in PROSITE (Sigrist *et al.*, 2013), resulting in 2967 drugs, 226 protein kinases, and 167 995 binding affinities between them in the form of $-\log_{10}(\mathrm{IC}_{50})$ values in molars.

We computed drug kernels, cell line kernels and protein kernels as described in the following sections and summarized in Figure 2c.

2.3.2 Kernels

Drug kernels. For drug compounds, we computed Tanimoto kernels using 10 different molecular fingerprints (Fig. 2c), i.e. binary vectors representing the presence or absence of different substructures in the molecule, obtained with rcdk R package (Guha, 2007): $k_d(\mathbf{x_d}, \mathbf{x'_d}) = H_{x_d,x'_d} / \left(H_{x_d} + H_{x'_d} - H_{x_d,x'_d}\right)$, where H_{x_d} is the number of 1-bits in the drug's fingerprint $\mathbf{x_d}$, and H_{x_d,x'_d} indicates the number of 1-bits common to fingerprints of two drug molecules $\mathbf{x_d}$ and $\mathbf{x'_d}$ under comparison. The above Tanimoto similarity measure is a valid PSD kernel function (Gower, 1971).

Cell line kernels. For cell lines, we calculated Gaussian kernels $k_c(\mathbf{x_c}, \mathbf{x'_c}) = = \exp\left(-||\mathbf{x_c} - \mathbf{x'_c}||^2 / 2\sigma_c^2\right)$, where $\mathbf{x_c}$ and $\mathbf{x'_c}$ denote feature representation of two cell lines in the form of (i) gene expression signature, (ii) methylation pattern, (iii) copy number variation or

(iv) somatic mutation profile (details given in Figure 2c and Supplementary Material); σ_c indicates a kernel width hyperparameter. We derived real-valued mutation profile feature vectors, instead of employing commonly used binary mutation indicators. In particular, each element x_{c_i}, $i = 1, \ldots, M$, corresponds to one of M mutations. If a cell line represented by $\mathbf{x_c}$ has a negative ith mutation status, then $x_{c_i} = 0$; otherwise, x_{c_i} indicates a negative logarithm of the proportion of all cell lines with positive mutation status. This way, x_{c_i} is high for a mutation specific to a cell line represented by $\mathbf{x_c}$, giving more importance to such genetic variant.

Protein kernels. For proteins, we computed Gaussian kernels based on real-valued gene ontology (GO) annotation profiles, as well as Smith–Waterman (SW) kernels and generic string (GS) kernels based on three types of amino acid sequences: (i) full kinase sequences, (ii) kinase domain sub-sequences and (iii) ATP binding pocket sub-sequences (Fig. 2c).

Gaussian GO-based kernels were calculated separately for molecular function, biological process and cellular component domains as $k_p\left(\mathbf{x_p}, \mathbf{x'_p}\right) = \exp\left(-||\mathbf{x_p} - \mathbf{x'_p}||^2/2\sigma_p^2\right)$, where $\mathbf{x_p}$ and $\mathbf{x'_p}$ denote GO profiles of two protein kinases. Each element of the GO profile feature vector, x_{p_i}, $i = 1, \ldots, G$, corresponds to one of G GO terms from a given domain. If a kinase represented by $\mathbf{x_p}$ is not annotated with term i, then $x_{p_i} = 0$; otherwise, x_{p_i} indicates a negative logarithm of the proportion of all proteins annotated with term i.

SW kernel measures similarity between amino acid sequences $\mathbf{x_p}$ and $\mathbf{x'_p}$ using normalized SW alignment score SW (Smith and Waterman, 1981): $k_p(\mathbf{x_p}, \mathbf{x'_p}) = \mathrm{SW}(\mathbf{x_p}, \mathbf{x'_p}) / \sqrt{\mathrm{SW}\left(\mathbf{x_p}, \mathbf{x_p}\right)\mathrm{SW}\left(\mathbf{x'_p}, \mathbf{x'_p}\right)}$. Although SW kernel is commonly used in drug–protein interaction prediction, it is not a valid PSD kernel function, and hence we examined all obtained matrices. The matrix corresponding to ATP binding pocket sub-sequences was not PSD, and therefore, we shrunk its off-diagonal entries until we reached the PSD property (126 shrinkage iterations with the shrinkage factor of 0.999 were needed for the matrix to become PSD). There are other ways of finding the nearest PSD matrix, e.g. by setting negative eigenvalues to 0, but we selected shrinkage since it smoothly modifies the whole spectrum of eigenvalues.

Finally, GS kernel compares each sub-string of $\mathbf{x_p}$ of size $l \leq L$ with each sub-string of $\mathbf{x'_p}$ having the same length: $k_p\left(\mathbf{x_p}, \mathbf{x'_p}\right) = \sum_{l=1}^{L} \sum_{i=0}^{|\mathbf{x_p}|-l} \sum_{j=0}^{|\mathbf{x'_p}|-l} \exp\left(-\frac{(i-j)^2}{2\sigma_1^2}\right) \exp\left(-\frac{||\boldsymbol{\xi}^l - \boldsymbol{\xi}'^l||^2}{2\sigma_2^2}\right)$, where vector $\boldsymbol{\xi}^l$ contains properties of l amino acids included in the sub-string under comparison (Giguère *et al.*, 2013). Each comparison results in a score that depends on the shifting contribution term (difference in the position of two sub-strings in $\mathbf{x_p}$ and $\mathbf{x'_p}$) controlled by σ_1, and the similarity of amino acids included in two sub-strings, controlled by σ_2. We used BLOSUM 50 matrix as amino acid descriptors in the GS kernel and in the SW sequence alignments.

We computed each Gaussian kernel with three different values of kernel width hyperparameter, determined by calculating pairwise distances between all data points, and then selecting 0.1, 0.5 and 0.9 quantiles. In case of each GS kernel, we selected the potential values for its three hyperparameters $L = \{5, 10, 15, 20\}$, $\sigma_1 = \{0.1, 1, 2, 3, 4\}$ and $\sigma_2 = \{0.1, 1, 2, 3, 4\}$ by ensuring that the resulting kernel matrices have a spectrum of different histograms of kernel values.

3 Results

To demonstrate the efficacy of *pairwiseMKL* for learning with multiple pairwise kernels, we tested the method in two related regression tasks of (i) prediction of anticancer efficacy of drug compounds and (ii) prediction of target profiles of anticancer drug compounds. In particular, we carried out a nested 10-fold cross validation (CV; 10 outer folds, 3 inner folds) using 15 376 drug responses in cancer cell lines and 167 995 drug–protein binding affinities, as well as chemical and genomic information sources in the form of kernels. We constructed a total of 120 pairwise drug–cell line kernels from 10 drug kernels and 12 cell line kernels, and 3120 pairwise drug–protein kernels from 10 drug kernels and 312 protein kernels (Fig. 2c).

We compared the performance of *pairwiseMKL* against the recently introduced algorithm for pairwise learning with multiple kernels *KronRLS-MKL* (Nascimento *et al.*, 2016). Both are regularized least-squares models, but *pairwiseMKL* first determines the kernel weights, and then optimizes the model parameters, whereas *KronRLS-MKL* interleaves the optimization of the model parameters with the optimization of kernel weights. Although *KronRLS-MKL* was originally used for classification only, it is a regression algorithm in its core, and with few modifications to the implementation (see Supplementary Material), we applied it here to quantitative drug bioactivity prediction. We used the same CV folds for both methods to ensure their fair comparison. We also conducted elastic net regression with standard feature vectors instead of kernels (see Supplementary Material).

Unlike *pairwiseMKL*, *KronRLS-MKL* assumes that bioactivities of all combinations of drugs and cell lines (proteins) are known, i.e. it does not allow for missing values in the label matrix storing drug–cell line (drug–protein) bioactivities. Therefore, in the experiments with *KronRLS-MKL*, we mean-imputed the originally missing bioactivities, as well as bioactivities corresponding to drug–cell line (drug–protein) pairs in the test folds. We assessed the predictive power of the methods with root mean squared error (RMSE), Pearson correlation and F1 score between original and predicted bioactivity values.

We tuned the regularization hyperparameter λ of *pairwiseMKL* and regularization hyperparameters λ and σ of *KronRLS-MKL* within the nested CV from the set $\{10^{-5}, 10^{-4}, \ldots, 10^{0}\}$. Instead of tuning the kernel hyperparameters in this standard way, we constructed several kernels with different carefully selected hyperparameter values (see Section 2.3.2 for details).

3.1 Drug response in cancer cell line prediction

In the task of anticancer drug response prediction with 120 pairwise kernels, *pairwiseMKL* provided accurate predictions, especially for those drug–cell line pairs with more training data points (Fig. 3a). It outperformed *KronRLS-MKL* in terms of predictive power, running time and memory usage (Table 2 and Supplementary Fig. S2). In particular, *pairwiseMKL* was almost 6 times faster and used 68-times less memory. Even though both *pairwiseMKL* and *KronRLS-MKL* achieved high Pearson correlation of 0.858 and 0.849, respectively, the accuracy of predictions from *KronRLS-MKL* decreased gradually when going away from the mean value to the tails of the response distribution (Supplementary Fig. S2), as indicated by 13% increase in RMSE and 40% decrease in F1 score when comparing to *pairwiseMKL* (Table 2). These extreme responses are often the most important cases to predict in practice, as they correspond to sensitivity or resistance of cancer cells to a particular drug treatment.

Importantly, *pairwiseMKL* returned a sparse combination of only 11 out of 120 input pairwise kernels, whereas all kernel weights from *KronRLS-MKL* are nearly uniformly distributed (Fig. 2a). The final model generated by *pairwiseMKL* is much

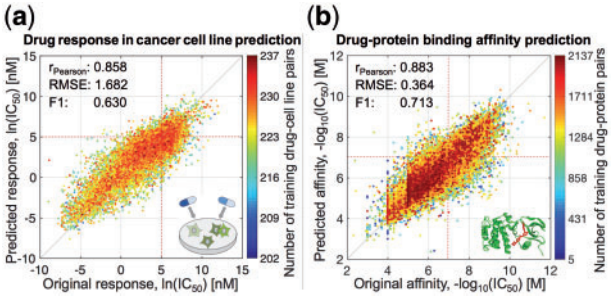


Fig. 3. Prediction performance of *pairwiseMKL* in the tasks of (**a**) drug response in cancer cell line prediction and (**b**) drug–protein binding affinity prediction. Scatter plots between original and predicted bioactivity values across (a) 15 376 drug–cell line pairs and (b) 167 995 drug–protein pairs. Performance measures were averaged over 10 outer CV folds. F1 score was calculated using the threshold of (a) $\ln(IC_{50}) = 5$ nM, (b) $-\log_{10}(IC_{50}) = 7$ M, both corresponding to low drug concentration of roughly 100 nM, i.e. relatively stringent potency threshold (red dotted lines). Color coding indicates the number of training data points, i.e. drug–cell line (respectively drug–protein) pairs including the same drug or cell line (drug or protein) as the test data point.

Table 2. Prediction performance, memory usage and running time of *pairwiseMKL* and *KronRLS-MKL* methods in the task of drug response in cancer cell line prediction.

Anticancer drug response prediction	RMSE	$r_{Pearson}$	F1 score	Memory (GB)	Time (h)
pairwiseMKL	1.682	0.858	0.630	0.057	1.45
KronRLS-MKL	1.899	0.849	0.378	3.890	8.42

Performance measures were averaged over 10 outer CV folds. F1 score was calculated using the threshold of $\ln(IC_{50}) = 5$ nM.

simpler due to fewer active kernels, and can therefore inform about the predictive power of different chemical and genomic data sources. Figure 2a shows that the pairwise kernel calculated using gene expression in cancer cell lines and molecular fingerprint defined by Klekota and Roth (2008) carries the greatest weight in the model. Klekota-Roth fingerprint is the longest among the considered ones, consisting of 4860 bits representing different substructures in the chemical compound, and gene expression profiles have previously been reported as most predictive of anticancer drug efficacy (Costello *et al.*, 2014). Other fingerprints identified by *pairwiseMKL* as relevant for drug response inference include information on Estate substructures (Hall and Kier, 1995), connectivity and shortest paths between compound's atoms. Among the cell line kernels, the ones calculated using genetic mutation and methylation patterns, in addition to gene expression, paired with the above-mentioned fingerprint-based drug kernels, were selected for the construction of the optimal pairwise kernel. Those information sources can therefore be considered as complementing each other. The copy number variation data did not prove effective in the prediction of anticancer drug responses.

3.2 Drug–protein binding affinity prediction

In the larger experiment of prediction of target profiles of almost 3000 anticancer drug compounds, *pairwiseMKL* achieved again high predictive performance (Pearson correlation of 0.883, RMSE of 0.364, F1 score of 0.713; Fig. 3b). Notably, our method constructed the final model using only 8 out of 3120 pairwise kernels (Fig. 2b). *KronRLS-MKL* did not execute given 1 TB memory, whereas *pairwiseMKL* required just a fraction of that memory (2.21 GB).

pairwiseMKL assigned the highest weights to two pairwise kernels build upon amino acid sub-sequences of ATP binding pockets, together with either shortest path fingerprints or extended connectivity fingerprints. A relatively high weight was given also to the pairwise kernel constructed from Klekota-Roth fingerprints and sub-sequences of kinase domains. In fact, kinase domain sequences include short sequences of ATP binding pockets, and capture also their neighboring context. In all of the above selected pairwise kernels, protein sequences were compared using GS kernel. Our results therefore suggest that ATP binding pockets are more informative than full amino acid sequences, and that GS kernel is more powerful in capturing similarities between amino acid sequences than a commonly used protein kernel based on SW amino acid sequence alignments, at least for the prediction of drug interactions with protein kinases investigated. Finally, gene ontology profiles of proteins, in particular those from biological process and cellular component domains, provided also a modest contribution to the optimal pairwise kernel used for drug–protein binding affinity prediction (Fig. 2b).

3.3 Kernel hyperparameters tuning

Our results demonstrate that *pairwiseMKL* provides also a useful tool for tuning the kernel hyperparameters. In particular, we constructed each kernel with different hyperparameter values from a carefully chosen range (see Section 2.3.2 for details), and the algorithm then selected the optimal hyperparameters by assigning non-zero mixture weights to corresponding kernels (Fig. 2). Notably, *pairwiseMKL* always picked a single value for the Gaussian kernel width hyperparameter, σ_c in case of cell line kernels and σ_p in case of protein kernels. This is well-represented in Figure 2a where, for each cell line data source, the weights are different from zero only in one of the three rows of the heatmap. Furthermore, *pairwiseMKL* selected also only a single out of 100 combinations of three values of the hyperparameters (L,σ_1,σ_2) for the GS kernel (Fig. 2b).

4 Discussion

The enormous size of the chemical universe, estimated to consist of up to 10^{24} molecules displaying good pharmacological properties (Reymond and Awale, 2012), makes the experimental bioactivity profiling of the full drug-like compound space infeasible in practice, and therefore calls for efficient *in silico* approaches that could aid various stages of drug development process and identification of optimal therapeutic strategies (Azuaje, 2017; Cheng *et al.*, 2012; Cichonska *et al.*, 2015). Especially kernel-based methods have proved good performance in many applications, including inference of drug responses in cancer cell lines (Costello *et al.*, 2014) and elucidation of drug MoA through drug–protein binding affinity predictions (Cichonska *et al.*, 2017). Pairwise learning is a natural approach for solving such problems involving pairs of objects, and the benefits from integrating multiple chemical and genomic information sources into clinically actionable prediction models are well-reported in the recent literature (Cheng and Zhao, 2014; Costello *et al.*, 2014; Ebrahim *et al.*, 2016).

To tackle the computational limitations of the current MKL approaches, we introduced here *pairwiseMKL*, a new framework for time- and memory-efficient learning with multiple pairwise kernels. *pairwiseMKL* is well-suited for massive pairwise spaces, owing to our novel, highly efficient formulation of Kronecker decomposition of the centering operator for the pairwise kernel, and a fast

method for training a regularized least-squares model with a weighted combination of multiple kernels. To illustrate our approach, we applied *pairwiseMKL* to two important problems in computational drug discovery: (i) the inference of anticancer potential of drug compounds and (ii) the inference of drug–protein interactions using up to 167 995 bioactivities and 3120 kernels.

pairwiseMKL integrates heterogeneous data types into a single model which is a sparse combination of input kernels, thus allowing to characterize the predictive power of different information sources and data representations by analyzing learned kernel mixture weights. For instance, our results demonstrate that among the genomic views, gene expression, followed by genetic mutation and methylation patterns contributed mostly to the final pairwise kernel adopted for anticancer drug response prediction (Fig. 2a). Although methylation plays an essential role in the regulation of gene expression, typically repressing the gene transcription, the association between these two processes remains incompletely understood (Wagner *et al.*, 2014). Therefore, it can be hypothesized that these genomic and epigenetic information levels are indeed complementing each other in the task of drug response modeling.

In case of prediction of target profiles of anticancer drugs, we observed the highest contribution to the final pairwise model from Tanimoto drug kernels, coupled with GS protein kernels applied to ATP binding pockets (Fig. 2b). This could be explained by the fact that majority of anticancer drugs, including those considered in this work, are kinase inhibitors designed to bind to ATP binding pockets of protein kinases, and therefore constructing kernels from short sequences of these pockets is more meaningful in the context of drug-kinase binding affinity prediction compared to using full protein sequences. Moreover, GS kernel is more advanced than the commonly used SW kernel as it compares protein sequences including properties of amino acids. GS kernel also enables to match short sub-sequences of two proteins even if their positions in the input sequences differ notably. In both prediction problems, *pairwiseMKL* was able to tune kernel hyperparameters by selecting a single kernel out of several kernels with different hyperparameter values (Fig. 2). It has been noted by Cortes *et al.* (2012) in the context of *ALIGNF* algorithm that sparse kernel weight vector is a consequence of the constraint $\boldsymbol{\mu} \geq 0$ in the kernel alignment maximization (Equation 6). This has been observed empirically in other works as well (e.g. Brouard *et al.*, 2016; Shen *et al.*, 2014). In particular, it appears that *pairwiseMKL* and *ALIGNF*, given a set of closely related kernels, such as those calculated using the same data source and kernel function but different hyperparameters, tend to select a representative kernel for the group to the optimized kernel mixture.

We compared the performance of *pairwiseMKL* to recently introduced method for pairwise learning with multiple kernels *KronRLS-MKL*. *pairwiseMKL* outperformed *KronRLS-MKL* in terms of predictive power, running time and memory usage (Table 2 and Supplementary Fig. S2). Unlike *pairwiseMKL*, *KronRLS-MKL* does not consider optimizing pairwise kernel weights, i.e. it finds separate weights for drug kernels and cell line (protein) kernels (Fig. 2a), and therefore it does not fully exploit the information contained in the pairwise space. The reduced predictive performance of *KronRLS-MKL* can be also attributed to the fact that it does not allow for any missing values in the label matrix storing bioactivities between drugs and cell lines (proteins), which need to be imputed as a pre-processing step and included in the model training. *KronRLS-MKL* has two regularization hyperparameters that need to be tuned, hence lengthening the training time. Furthermore, determining the parameters of the pairwise prediction function involves a computation of large matrices, which requires significant amount of memory that grows quickly with the number of drugs and cell lines (proteins). Finally, *KronRLS-MKL* applies L2 regularization on the kernel weights, thus not enforcing sparse kernel selection. In fact, *KronRLS-MKL* returned a nearly uniform combination of input kernels, not allowing for the interpretation of the predictive power of different data sources.

We tested *pairwiseMKL* using CV on the level of drug–cell line (drug–protein) pairs which corresponds to evaluating the performance of the method in the task of filling experimental gaps in bioactivity profiling studies. However, *pairwiseMKL* could also be applied, e.g. to the inference of anticancer potential of a new candidate drug compound or prediction of sensitivity of a new cell line to a set of drugs. We plan to tackle these important problems in the future work.

In this work, we put an emphasis on the regression task, since drug bioactivity measurements have a real-valued nature, but we also implemented an analogous method for solving the classification problems with support vector machine. Other potential applications of our efficient Kronecker decomposition of the centering operator for the pairwise kernel include methods which involve kernel centering in the pairwise space, such as pairwise kernel PCA. Finally, even though we focused here on the problems of anticancer drug response prediction and drug–target interaction prediction, *pairwiseMKL* has wide applications outside this field, such as in the inference of protein–protein interactions, binding affinities between proteins and peptides or mRNAs and miRNAs.

Acknowledgements

We acknowledge the computational resources provided by the Aalto Science-IT project and CSC - IT Center for Science, Finland.

Funding

This work was supported by the Academy of Finland [289903 to A.A.; 295496 and 313268 to J.R.; 299915 to M.H.; 311273 and 313266 to T.P.; 295504, 310507 and 313267 to T.A.].

Conflict of Interest: none declared.

References

Airola,A. and Pahikkala,T. (2017) Fast Kronecker product kernel methods via generalized vec trick. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–4. https://ieeexplore.ieee.org/abstract/document/7999226/

Ali,M. *et al.* (2017) Global proteomics profiling improves drug sensitivity prediction: results from a multi-omics, pan-cancer modeling approach. *Bioinformatics*, **1**, 10.

Ammad-Ud-Din,M. *et al.* (2016) Drug response prediction by inferring pathway-response associations with kernelized Bayesian matrix factorization. *Bioinformatics*, **32**, i455–i463.

Azuaje,F. (2017) Computational models for predicting drug responses in cancer research. *Brief, Bioinform.*, **18**, 820–829.

Barretina,J. *et al.* (2012) The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483**, 603–607.

Brouard,C. *et al.* (2016) Fast metabolite identification with input output kernel regression. *Bioinformatics*, **32**, i28–i36.

Cheng,F. *et al.* (2012) Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.*, **8**, e1002503.

Cheng,F. and Zhao,Z. (2014) Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *J. Am. Med. Inform. Assoc.*, **21**, e278–e286.

Cichonska,A. *et al*. (2015) Identification of drug candidates and repurposing opportunities through compound-target interaction networks. *Exp. Opin. Drug Discov*., **10**, 1333–1345.

Cichonska,A. *et al*. (2017) Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. *PLoS Comput. Biol*., **13**, e1005678.

Cortes,C. *et al*. (2012) Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res*., **13**, 795–828.

Costello,J.C. *et al*. (2014) A community effort to assess and improve drug sensitivity prediction algorithms. *Nat. Biotechnol*., **32**, 1202–1212.

Ebrahim,A. *et al*. (2016) Multi-omic data integration enables discovery of hidden biological regularities. *Nat. Commun*., **7**, 13091.

Elefsinioti,A. *et al*. (2016) Key factors for successful data integration in biomarker research. *Nature Rev Drug Discov*., **15**, 369–370.

Engl,H.W. *et al*. (1996) *Regularization of Inverse Problems. Vol. 375*, Netherlands: Springer Science & Business Media.

Giguère,S. *et al*. (2013) Learning a peptide-protein binding affinity predictor with kernel ridge regression. *BMC Bioinformatics*, **14**, 82.

Gower,J.C. (1971) A general coefficient of similarity and some of its properties. *Biometrics*, **27**, 857–871.

Guha,R. (2007) Chemical informatics functionality in R. *J. Stat. Soft*., **18**, **1**, 16.

Hall,L.H. and Kier,L.B. (1995) Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. *J. Chem. Inf. Comput. Sci*., **35**, 1039–1045.

Klekota,J. and Roth,F.P. (2008) Chemical substructures that enrich for biological activity. *Bioinformatics*, **24**, 2518–2525.

Kludas,J. *et al*. (2016) Machine learning of protein interactions in fungal secretory pathways. *PLoS One*, **11**, e0159302.

Marcou,G. *et al*. (2016) Kernel target alignment parameter: a new modelability measure for regression tasks. *J. Chem. Inf. Model*., **56**, 6–11.

Merget,B. *et al*. (2017) Profiling prediction of kinase inhibitors: toward the virtual assay. *J. Med. Chem*., **60**, 474–485.

Nascimento,A.C. *et al*. (2016) A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinformatics*, **17**, 46.

Pahikkala,T. *et al*. (2015) Toward more realistic drug-target interaction predictions. *Brief. Bioinformatics*, **16**, 325–337.

Reymond,J.L. and Awale,M. (2012) Exploring chemical space for drug discovery using the chemical universe database. *ACS Chem. Neurosci*., **3**, 649–657.

Saunders,C. *et al*. (1998) Ridge regression learning algorithm in dual variables. *In*: *Proceedings of the 15th International Conference on Machine Learning*, pp. 515–521.

Shawe-Taylor,J. and Cristianini,N. (2004) *Kernel Methods for Pattern Analysis*. New York: Cambridge University Press.

Shen,H. *et al*. (2014) Metabolite identification through multiple kernel learning on fragmentation trees. *Bioinformatics*, **30**, i157–i164.

Sigrist,C.J. *et al*. (2013) New and continuing developments at PROSITE. *Nucleic Acids Res*., **41**, D344–D347.

Smirnov,P. *et al*. (2018) PharmacoDB: an integrative database for mining in vitro anticancer drug screening studies. *Nucleic Acids Res*., **46**, D994–D1002.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

Sorgenfrei,F.A. *et al*. (2017) Kinomewide profiling prediction of small molecules. *ChemMedChem*, **12**, 1–6.

Wagner,J.R. *et al*. (2014) The relationship between DNA methylation, genetic and expression inter-individual variation in untransformed human fibroblasts. *Genome Biol*., **15**, R37.

Yang,W. *et al*. (2012) Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res*., **41**, D955–D961.