metabolites

Article

Improved Small Molecule Identification through Learning Combinations of Kernel Regression Models

Céline Brouard [1,*], Antoine Bassé [2], Florence d'Alché-Buc [2] and Juho Rousu [3,†]

[1] Unité de Mathématiques et Informatique Appliquées de Toulouse, UR 875, INRA, 31326 Castanet Tolosan, France
[2] LTCI, Télécom Paris, Institut Polytechnique de Paris, 75634 Paris, France
[3] Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, 00076 Espoo, Finland
* Correspondence: celine.brouard@inra.fr
† Part of this work was commenced during J.R.'s research visit to Telecom ParisTech.

Received: 29 June 2019; Accepted: 31 July 2019; Published: date

Abstract: In small molecule identification from tandem mass (MS/MS) spectra, input–output kernel regression (IOKR) currently provides the state-of-the-art combination of fast training and prediction and high identification rates. The IOKR approach can be simply understood as predicting a fingerprint vector from the MS/MS spectrum of the unknown molecule, and solving a pre-image problem to find the molecule with the most similar fingerprint. In this paper, we bring forward the following improvements to the IOKR framework: firstly, we formulate the IOKRreverse model that can be understood as mapping molecular structures into the MS/MS feature space and solving a pre-image problem to find the molecule whose predicted spectrum is the closest to the input MS/MS spectrum. Secondly, we introduce an approach to combine several IOKR and IOKRreverse models computed from different input and output kernels, called IOKRfusion. The method is based on minimizing structured Hinge loss of the combined model using a mini-batch stochastic subgradient optimization. Our experiments show a consistent improvement of top-k accuracy both in positive and negative ionization mode data.

Keywords: metabolite identification; machine learning; structured prediction; kernel methods

1. Introduction

In recent years, the massively increased amounts of publicly available reference tandem mass (MS/MS) spectra in databases such as GNPS [1] and MassBank [2] have caused a revolution in small molecule identification. In particular, the use of modern machine learning approaches has become feasible [3], and led to the generation of a host of machine learning approaches and identification tools such as FingerID [4,5], CFM-ID [6,7], CSI:FingerID [8], CSI:IOKR [9], magnitude-preserving IOKR [10] ChemDistiller [11], SIMPLE [12], ADAPTIVE [13] and SIRIUS [14]. The identification rates have witnessed a step-change upward, and consequently the use of the tools in practical work-flows has massively increased (see, e.g., [15]).

The majority of the machine learning methods rely on the same conceptual scheme [3] introduced with FingerID [4]: predicting molecular fingerprints from MS/MS data and finding the most similar fingerprint from the molecular structure database. This approach has been very successful, for example, CSI:FingerID [8] and CSI:IOKR [9] have been top performers in the most recent CASMI contests (2016: [16] and 2017: [17]). The alternative conceptual approach for small molecule identification, sometimes called in silico fragmentation [3], calls for predicting MS/MS spectra for a set of candidate molecular structures and choosing the most similar predicted MS/MS spectrum to the observed

MS/MS spectrum. This approach is used, e.g., in the non-machine learning based MetFrag [18,19] as well as CFM-ID [6,7], which is the most notable machine learning tool relying on the in silico fragmentation approach.

CSI:FingerID uses an array of Support Vector Machines with multiple kernel learning [20] to individually predict each bit of the molecular fingerprint vector, whereas CSI:IOKR predicts the molecular structures through a single structured output prediction [21] algorithm, called Input–Output Kernel Regression (IOKR) [22], where both inputs and outputs are kernelized for the best performance. Due to this approach, CSI:IOKR is extremely fast to train and is on par with CSI:FingerID in accuracy. Both CSI:FingerID and CSI:IOKR make use of multiple data sources, fused using the multiple kernel learning (MKL) algorithm ALIGNF [23] that sets importance weights to the input kernels prior learning the fingerprint prediction models. Interestingly, CSI:FingerID [8] benefits from the MKL technique more than CSI:IOKR [9] that provides equally good or better results using uniform weights for the inputs, a technique referred to as *uniform* MKL or Unimkl. This corresponds to summing up or averaging the input kernels.

In this paper, we bring forward two methodological contributions. Firstly, we extend the IOKR [9] approach by formulating essentially an in silico fragmentation problem which we call IOKRreverse. From a set of candidate molecular structures, we implicitly (through a kernel function) predict a representation of an MS/MS spectrum for each candidate, and solve a pre-image problem to output the molecular structure whose predicted MS/MS is the closest to the observed one. All this computation is done through kernel matrices of the inputs (MS/MS spectra) and outputs (molecular structures).

Secondly, we introduce an approach called IOKRfusion to combine multiple IOKR and IOKRreverse models, which arise from the use of different input and output kernels on the training data. The models are combined by minimizing the structured Hinge loss [24], which is frequently used in structured output learning, and corresponds to a convex (thus efficiently computable) upper bound for maximizing top-1 accuracy over a candidate set (an NP-hard task). We bring forward a mini-batch subgradient algorithm for the optimization. This way of aggregating multiple data sources is sometimes called *late fusion*, since the model learning happens before the aggregation, as compared to the multiple kernel learning using ALIGNF [23], which happens before model learning, making it an *early fusion* approach.

The structure of the paper is as follows. In Section 2, we review the IOKR model and present the IOKRreverse model as well as the late fusion approach for minimizing the structured Hinge loss of the combined model. Section 3 presents our experiments with the models and Section 4 presents the discussion.

2. Materials and Methods

In the following, we note $\mathcal{X}$ the set of tandem mass spectra and $\mathcal{Y}$ a set containing 2D molecular structures. We consider a set of ℓ training examples $\{x_i, y_i\}_{i=1}^{\ell} \subseteq \mathcal{X} \times \mathcal{Y}$.

2.1. Input–Output Kernel Regression

Input Output Kernel Regression (IOKR) [22] is a machine learning framework that can be used for solving structured prediction problems. Structured output prediction involves the prediction of outputs corresponding to complex structured objects, for example graphs or trees, rather than scalar values as in regression and classification problems. Structure output prediction can also be used in the case where we search to predict multiple interdependent outputs. Structured data can generally be decomposed into several parts and structured prediction approaches make use of the dependencies existing between these parts.

The IOKR framework has been used in CSI:IOKR [9] for compound structure identification (CSI), where we search to predict the molecular structures of metabolites from their MS/MS spectra. In [9], the similarities between the molecular structures were encoded using an output kernel function $k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. This kernel is associated with a high-dimensional vector space $\mathcal{F}_y$, referred to as the output feature space, and a function $\psi : \mathcal{Y} \to \mathcal{F}_y$ that maps outputs (molecules) to the output feature space $\mathcal{F}_y$. The inner product in the feature space can be evaluated by computing the values of the output kernel:

$$\langle \psi(y), \psi(y') \rangle_{\mathcal{F}_y} = k_y(y, y'), \ \forall y, y' \in \mathcal{Y}.$$

IOKR solves the metabolites identification problem by first learning a function h from $\mathcal{X}$ to $\mathcal{F}_y$ that approximates the output feature map ψ. h is learned by solving the following regression problem:

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{\ell} \|h(x_i) - \psi(y_i)\|^2_{\mathcal{F}_y} + \lambda_h \|h\|^2_{\mathcal{H}}, \tag{1}$$

where $\lambda_h > 0$ is a regularization parameter. h is modeled as: $h(x) = W\phi(x)$, $\forall x \in \mathcal{X}$ where $\phi : \mathcal{X} \to \mathcal{F}_x$ is a feature map from $\mathcal{X}$ to an input feature space $\mathcal{F}_x$ associated with an input kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ measuring a similarity between MS/MS spectra. W is a linear operator from $\mathcal{F}_x$ to $\mathcal{F}_y$. When using this model, the solution to Problem (1) is given by:

$$h(x) = \sum_{i=1}^{\ell} \alpha_i(x)\psi(y_i), \ \text{ with } \boldsymbol{\alpha}(x) = (\lambda_h I_\ell + K_X)^{-1} k_X^x, \tag{2}$$

where K_X is the kernel matrix of k_x on the training set: $[K_X]_{i,j} = k_x(x_i, x_j) \, \forall i, j = 1, \ldots, \ell$ and $k_X^x = \left[k_x(x_1, x), \ldots k_x(x_\ell, x)\right]^T \in \mathbb{R}^\ell$ collects the kernel evaluations of the training inputs against x.

Given the prediction $h(x_i)$ for an MS/MS spectrum x_i, the prediction of the corresponding molecule y_i requires solving a pre-image problem. For this, we consider a subset $\mathcal{Y}_{x_i}$ of molecular structures from a large database such as PubChem [25], for example the set of molecules having the same molecular formula as x_i if it is known or having a similar mass to the one measured for x_i. We then search for the nearest molecule to the prediction $h(x_i)$ in the output feature space $\mathcal{F}_y$:

$$f(x_i) = \underset{y \in \mathcal{Y}_{x_i}}{\operatorname{argmin}} \|h(x_i) - \psi(y)\|^2_{\mathcal{F}_y}.$$

When replacing $h(x_i)$ by the solution given in (2), this can be rewritten as follows:

$$f(x_i) = \underset{y \in \mathcal{Y}_{x_i}}{\operatorname{argmin}} \, \boldsymbol{\alpha}(x_i)^T K_Y \boldsymbol{\alpha}(x_i) + k_y(y, y) - 2\boldsymbol{\alpha}(x_i)^T k_Y^y,$$

where K_Y and k_Y^y are defined similarly to K_X and k_X^x.

2.2. IOKRreverse: Mapping Kernel Representations of Molecules to Kernel Representation of MS/MS Spectra

In this subsection, we introduce a variant of IOKR called IOKRreverse inspired from recent works designed to remedy the problem of hubness in high dimensional nearest-neighbour approaches [26]. Hubness in k-nearest neighbours refers to the emergence of hubs, i.e., points that are among the k-nearest neighbours of a lot of data. The presence of hubs can have a bad impact on k-nearest neighbours accuracy. Recently, this phenomenon observed in high dimensional search spaces has also been identified to be an important source of error in Zero-Shot Learning [27]. In a nutshell, Zero-Shot Learning [28,29] is a realistic machine learning setting for multi-class classification especially meaningful when the number of classes is extremely large: it consists of learning a classifier able to predict classes not seen during the training phase. One of the most relevant approaches to zero-shot learning relies on a regression-based scheme very similar to IOKR. Labels are first mapped onto a Euclidean space. Then, a function is learned to solve the regression problem in the Euclidean space instead of solving a classification problem. Eventually, to make a prediction on a new example, a nearest neighbour search provides the label closest to the mapped example. Recent contributions [27] have shown that reversing the regression problem, meaning attempting to approximate the relationship between the outputs (in this case the mapped labels) and the inputs (for instance images) allows for mitigating the hubness problem and provides a significant accuracy improvement. Variance of the data on which the nearest neighbour search is performed is key to the hubness. As regression has a shrinkage effect on mapped data impacting their variance, direct regression and reverse regression do not have the same effect. The authors in [27] have demonstrated that reverse regression is expected to provide smaller variance and better performance when retrieving the output objects.

As the pre-image problem in IOKR boils down to search for the nearest neighbour, we propose to adopt a similar scheme in the context of IOKR. Instead of learning a function h that maps the input examples to the output feature space, we learn a function g that maps the output examples to the input feature space, in this case, molecular structures to MS/MS feature space (see Figure 1).

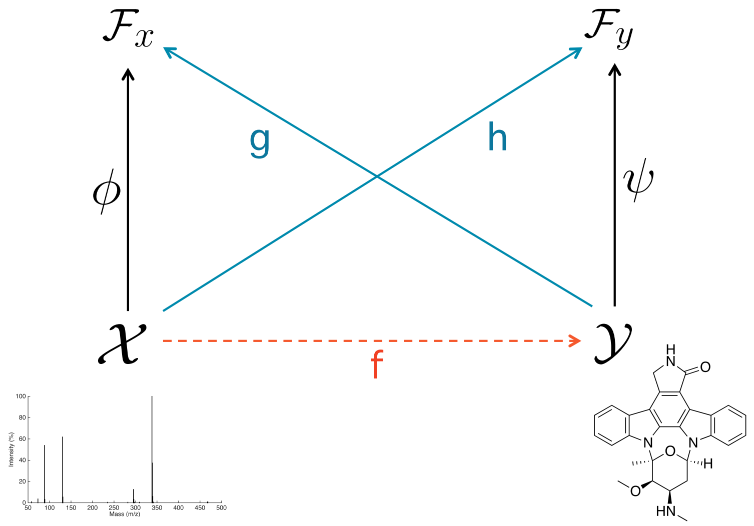


Figure 1. Schematic illustration of IOKR and IOKRreverse approaches. IOKR learns a function h to map MS/MS spectra to a molecular feature space $\mathcal{F}_y$, whereas IOKRreverse learns a function g to map the molecular structures to a MS/MS feature space $\mathcal{F}_x$.

To learn this function g, we solve the following optimization problem:

$$\min_{g \in \mathcal{G}} \sum_{i=1}^{\ell} \|g(y_i) - \phi(x_i)\|^2_{\mathcal{F}_x} + \lambda_g \|g\|^2_{\mathcal{G}},\ \lambda_g > 0. \tag{3}$$

This time the function g is modeled as $g(y) = V\psi(y)$, $\forall y \in \mathcal{Y}$, where V is a linear operator from $\mathcal{F}_y$ to $\mathcal{F}_x$. When replacing $g(y)$ by this expression, the solution of the IOKRreverse optimization problem in (3) is given by:

$$g(y) = \sum_{i=1}^{\ell} \beta_i(y)\phi(x_i), \quad \text{where } \boldsymbol{\beta}(y) = (\lambda_g I_\ell + K_Y)^{-1} k_Y^y.$$

In IOKRreverse, the pre-image problem consists of solving a nearest neighbor problem in the input feature space $\mathcal{F}_x$. Given the input feature vector of an MS/MS spectrum x_i, we use the function g learned in the previous step for predicting the input feature vectors for all the candidates in $\mathcal{Y}_{x_i}$ (the candidate set of x_i). We then search the closest candidate to $\phi(x_i)$ in the input feature space $\mathcal{F}_x$:

$$f(x_i) = \operatorname*{argmin}_{y \in \mathcal{Y}_{x_i}} \|g(y) - \phi(x_i)\|^2_{\mathcal{F}_x}.$$

Using the kernel trick in the input space, this can be rewritten under the following form:

$$f(x_i) = \operatorname*{argmin}_{y \in \mathcal{Y}_{x_i}} \boldsymbol{\beta}(y)^T K_X \boldsymbol{\beta}(y) + k_x(x_i, x_i) - 2\boldsymbol{\beta}(y)^T k_X^{x_i}.$$

2.3. Combining Multiple Models to Maximize Top-1 Accuracy

Combining multiple representations and data sources is a potent way of improving the predictive capabilities of machine learning models. This task can be implemented in several ways [30], in particular using early fusion, where data sources and representations are combined prior to learning the model, or late fusion, where the models learned using different representations are combined after model learning. Given multiple input kernels, a popular early fusion approach is to use multiple kernel learning [23] to find a liner combination of the input kernels so that the combined kernel would be similar to a given target kernel. The learned combined kernel is then used as the input kernel in the next phase. This approach has been previously used in both CSI:FingerID [14] and CSI:IOKR [9].

Here, we propose instead to combine the set of models learned by using individual input and output kernels after learning the models, that is, using late fusion. Several models are learned using the IOKR and IOKRreverse approaches for different pairs of input, output kernels. For each of these models, we can compute a compatibility score $s(x,y)$ for an input–output pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$, for example using the normalized cosine similarity:

$$\begin{aligned} s(x,y) &= \frac{\langle h(x), \psi(y) \rangle_{\mathcal{F}_y}}{\|h(x)\| \|\psi(y)\|} \text{ in the case of IOKR,} \\ s(x,y) &= \frac{\langle g(y), \phi(x) \rangle_{\mathcal{F}_x}}{\|g(y)\| \|\phi(x)\|} \text{ for IOKRreverse.} \end{aligned}$$

We then search to learn a linear combination of the score functions obtained with IOKR and IOKRreverse: $\sum_{k=1}^{K} w_k s_k(x,y)$. The goal is to learn a vector $\mathbf{w}$ such that the linear combination of the scores for the correct pair (x_i, y_i) is separated from the combined scores of all incorrect pairs (x_i, y) for $y \in \mathcal{Y}_{x_i}$. For this, we solve the optimization problem proposed in the structured SVM approach [31]:

$$\min_{\mathbf{w}} J(\mathbf{w}) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{\ell} \sum_{i=1}^{\ell} \ell(\mathbf{w}, x_i, y_i),\ \lambda \geq 0, \tag{4}$$

where

$$\ell(\mathbf{w}, x_i, y_i) = \max_{y \in \mathcal{Y}_{x_i}} \Big(\Delta(y_i, y) - \mathbf{w}^T (\mathbf{s}(x_i, y_i) - \mathbf{s}(x_i, y)) \Big)$$

is the structured Hinge loss and $\mathbf{s}(x,y) = (s_1(x,y), \ldots, s_K(x,y))^T$ is a vector of compatibility scores. $\Delta(y_i, y)$ is a measure of distance between the two output structures y_i and y. Here, we used the Hamming loss between molecular fingerprints: $\Delta(y_i, y) = \frac{1}{d} \sum_{j=1}^{d} \mathbb{1}_{fp(y_i) \neq fp(y)}$. We solve this optimization problem using a mini-batch subgradient descent (see Algorithm 1). This is an iterative optimization algorithm. At each step, a mini batch $\mathcal{B}$ of m training examples is selected at random and the weights are updated as follows:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - t \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{\mathbf{w}} J_i(\mathbf{w}),$$

where t is the step size and $J_i(\mathbf{w}) = \frac{\ell \lambda}{2} \|\mathbf{w}\|^2 + \ell(\mathbf{w}, x_i, y_i)$.

Algorithm 1: Mini-batch subgradient descent for the score aggregation.

```
Initialize $\mathbf{w}^{(0)}$;
for $k = 1$ to $K$ do
    Select mini batch $\mathcal{B}$ of size $m$ from the training set;
    For each pair $(x_i, y_i)$ in $\mathcal{B}$, find the candidate output $y$ with the highest loss:
        $\bar{y}_{x_i} = \underset{y \in \mathcal{Y}_{x_i}}{\text{argmax}} \left(\Delta(y_i, y) - \mathbf{w}^T \mathbf{s}(x_i, y_i) + \mathbf{w}^T \mathbf{s}(x_i, y)\right)$
    Update the weights:
        $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - t(\lambda \mathbf{w}^{(k-1)} + \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{s}(x_i, \bar{y}_{x_i}) - \mathbf{s}(x_i, y_i))$
end
Output $\mathbf{w}^K$
```

2.4. Kernels

In this subsection, we present the different input and output kernels used to measure the similarity between MS/MS spectra and molecular structures, respectively.

2.4.1. Input Kernels

We considered 15 different kernels on MS/MS spectra (listed in Table 1). Among these kernels, 14 were defined based on fragmentation trees [32,33] that model the fragmentation process of the metabolites as a tree. In this tree, each node corresponds to a peak in the MS/MS spectrum and is annotated by the predicted molecular formula for the corresponding fragment. The edges correspond to losses and give information about the fragmentation reactions existing between pairs of fragments. Different kernels on fragmentation trees can be defined by comparing the nodes, the edges or the paths for pairs of fragmentation trees. In addition, we used the recalibrated probability product kernel (PPKr) [4,8] that models the peaks in an MS/MS spectrum as a two-dimensional, normal distribution with σ_m and σ_i the standard deviations on the mass-to-charge ratio and the intensity. This kernel writes as:

$$k_x^{PPKr}(x, x') = \frac{1}{n_x, n_{x'}} \frac{1}{4\pi\sigma_m\sigma_i} \sum_{\ell,\ell'=1}^{n_x, n_{x'}} \exp\left(-\frac{(m(x_\ell) - m(x'_{\ell'}))^2}{4\sigma_m^2}\right) \exp\left(-\frac{(i(x_\ell) - i(x'_{\ell'}))^2}{4\sigma_i^2}\right),$$

where $m(x_\ell)$ denotes the mass-to-charge ratio of the ℓ-th peak of spectrum x and $i(x_\ell)$ its intensity. n_x indicates the number of peaks contained in x.

Table 1. Description of the input kernels (see [34] for further details).

	Name	Description
LI	Loss intensity	counts the number of common losses weighted by the intensity
RLB	Root loss binary	counts the number of common losses from the root to some node
RLI	Root loss intensity	weighted variant of RLB that uses the intensity of terminal nodes
JLB	Joined loss binary	counts the number of common joined losses
LPC	Loss pair counter	counts the number of two consecutive losses within the tree
MLIP	Maximum loss in path	counts the maximum frequencies of each molecular formula in any path
NB	Node binary	counts the number of nodes with the same molecular formula
NI	Node intensity	weighted variant of NB that uses the intensity of nodes

Table 1. *Cont.*

	Name	Description
NLI	Node loss interaction	counts common paths and weights them by comparing the molecular formula of their terminal fragments
SLL	Substructure in losses and leafs	counts for different molecular formula in how many paths they are conserved (part of all nodes) or cleaved off intact (part of a loss)
NSF	Node subformula	considers a set of molecular formula $\mathcal{M}$ and counts how often each of them occurs as subset of nodes in both trees
NSF3		takes the value of NSF to the power of three
GJLSF	Generalized joined loss subformula	counts how often each molecular formula from $\mathcal{M}$ occurs as subset of joined losses in both fragmentation graphs
RDBE	Ring double-bond equivalent	compares the distribution of ring double-bond equivalent values between two trees
PPKr	Recalibrated probability product kernel	computes the probability product kernel on preprocessed spectra

2.4.2. Output Kernels

We measured similarities between molecular structures by using kernels between molecular fingerprints. A molecular fingerprint represents the structure of a molecule as a binary vector, where each value indicates the presence or absence of a molecular property. A molecular property can encode the presence of a certain bond, substructure or atom configuration. As in [9], we used a linear and a Gaussian kernel on molecular fingerprints. In addition, we considered the Tanimoto kernel [35] that is commonly used for comparing molecular fingerprints:

$$k_y^{tan}(y,y') = \frac{|fp(y) \cap fp(y')|}{|fp(y) \cup fp(y')|} = \frac{fp(y)^T fp(y')}{\|fp(y)\|^2 + \|fp(y')\|^2 - fp(y)^T fp(y')}.$$

We also proposed a modified version of the Gaussian kernel, in which the distance is replaced by the distance between the feature vectors associated with the Tanimoto kernel:

$$\begin{aligned} k_y^{gauss-tan}(y,y') &= \exp\left(-\gamma \left\|\psi^{tan}(y) - \psi^{tan}(y')\right\|^2_{\mathcal{F}_y}\right) \\ &= \exp\left(-\gamma \left(k_y^{tan}(y,y) + k_y^{tan}(y',y') - 2k_y^{tan}(y,y')\right)\right). \end{aligned}$$

3. Results

We used two subsets of tandem mass spectra from GNPS (Global Natural Products Social molecular networking) [1] and MassBank [2] to evaluate the performance of our method. The first subset contains 6974 MS/MS spectra, corresponding to 6504 structures, measured with a positive ionization mode while the second subset contains 3578 MS/MS spectra, corresponding to 2376 structures, measured with a negative ionization mode. In the positive ionization mode, the spectra have the following adducts: $[M + H]^+$, $[M + K]^+$, $[M + Na]^+$, $[M - H2O + H]^+$, $[M]^+$ and $[M + H3N + H]^+$, while the following adducts are observed in the negative ionization mode: $[M - H]^-$, $[M]^-$, $[M + Cl]^-$, $[M - H2O - H]^-$, $[M + CH2O2 - H]^-$ and $[M + C2H4O2 - H]^-$. We consider separately the MS/MS spectra measured with negative and positive ionization modes as the mechanisms of fragmentation of positive and negative ions are different. We visualize this difference in the supplementary materials by comparing the spectra similarities in the different ionization modes.

The spectra correspond to LC-MS/MS data measured with Quadrupole-Time of Flight (Q-ToF), Orbitrap, Fourier Transform Ion Cyclotron Resonance (FTICR) and ion trap instruments. The MS/MS spectra measured with different collision energies have been merged together. We considered molecular fingerprints containing 7593 molecular properties computed using the Chemistry Development

Kit (CDK) [36]. These fingerprints contain molecular properties from FP2 (55 bits), FP3 (307 bits), MACCS (166 bits), Pubchem fingerprint (881 bits), Klekota–Roth (4860 bits) [37] and ECFP (Extended-connectivity Fingerprints) (1324 bits) [38].

3.1. Experimental Protocol

The performance of the models were evaluated using 5-fold cross-validation (CV). The MS/MS spectra corresponding to the same molecular structures were contained in the same fold. This avoids having the case where an MS/MS spectrum in the training set has the same molecular structure as a test example. In each round of the cross-validation, we used three folds for training the IOKR and IOKRreverse models, one fold to train the score aggregator and the last fold as test set. We evaluated the performance by computing the averaged top-k accuracy over the test examples. This corresponds to the percentage of test examples for which the true molecular structure was found among the k top ranked molecules.

The regularization parameters λ_h and λ_g were tuned on the training set of each fold among a grid. We selected the parameters that minimize the averaged mean squared error. Regarding the parameter λ used in the aggregation model, it was selected on the validation set using 4-CV such that it maximized the top-1 accuracy. Regarding the parameter γ of the Gaussian and Gaussian–Tanimoto output kernels, we took the value for which the entropy is maximal. All of the kernels have been centered and normalized.

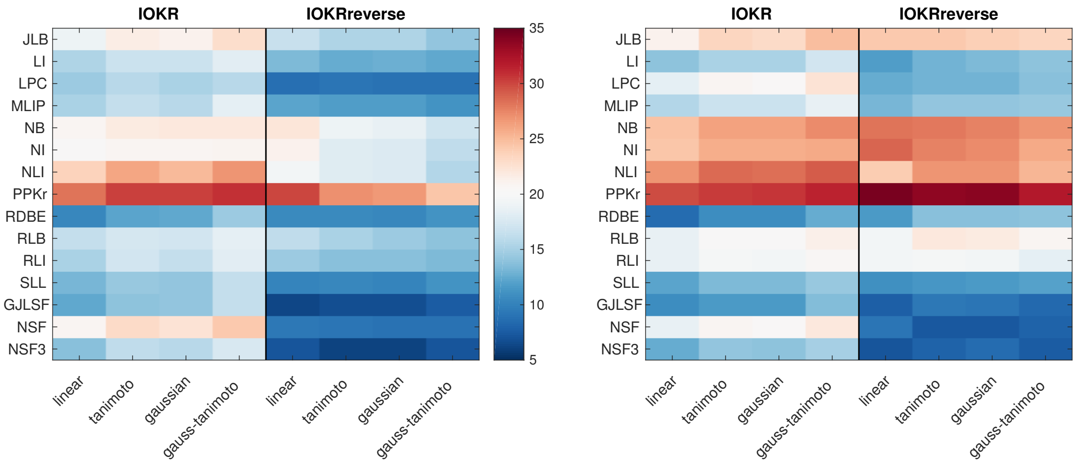
In the pre-image, we assumed the molecular formula of the test spectra to be known and we considered the molecules from Pubchem with the same molecular formula as candidates.

For the Mini-batch subgradient descent, we used 30 epochs, this means that we passed on the full training set 30 times. In each epoch, the training set was split randomly into small batches of fixed size. The mini-batch size was selected by cross-validation on the validation set. The learning rate was set to $t = \frac{1}{\lambda k}$ at iteration k.

In Section 3.4, we include the predictive performance obtained with the competing method CSI:FingerID [8]. These results were obtained using the CSI:FingerID 1.1 version with the modified Platt scoring, for which the best predictive performance have been observed in [8]. We applied this method on the same cross-validation folds, using four folds for training and the last fold for testing, and the parameter c was tuned on the training sets. We used as input kernel the combination of the 15 input kernels learned with the ALIGNF algorithm [23].

3.2. Results Obtained with IOKR and IOKRreverse Using Different Kernels

We first report on using IOKR and IOKRreverse as standalone models, and selecting a single input and output kernel at the time for the model. In Figure 2, we visualized the top-1 accuracy obtained with the IOKR and IOKRreverse approaches using different pairs of single input and single output kernel. The best input kernel is PPKr, consistently for both IOKR and IOKRreverse, for both ionization modes and different output kernels. In the case of negative ionization mode, the predictive performance obtained with IOKRreverse is worse than the ones obtained with IOKR for most of the kernels. With the PPKr kernel, the top-1 accuracy of IOKRreverse using a linear output kernel is equal to 29.85, slightly below the highest top-1 accuracy obtained with IOKR (30.74). In addition, the fragmentation tree based kernels do not work well as standalone input kernels in negative ionization mode, in particular for IOKRreverse. In the positive ionization mode, the results are different. IOKR still performs better for most of the kernels. However, IOKRreverse is better than IOKR for three out of the four best input kernels (NB, NI, NLI, PPKr). This improvement is especially important for the best performing input kernel PPKr. When using this input kernel, IOKRreverse increases the top-1 accuracy by three percentage points: 34.36 instead of 31.22 for IOKR.



(**a**) Negative ionization mode

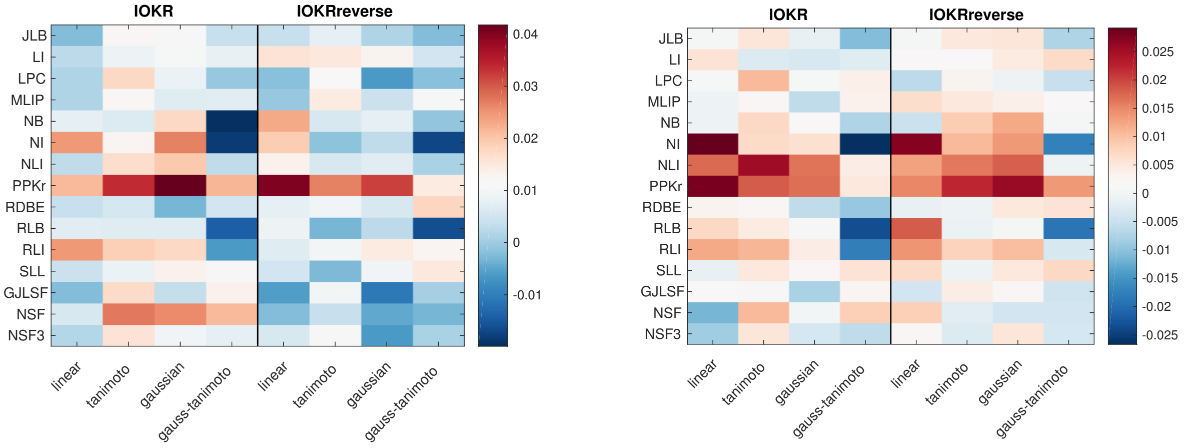
(**b**) Positive ionization mode

Figure 2. Heatmap of the top-1 accuracy obtained with IOKR and IOKRreverse for different input and output kernels in the negative ionization mode (**a**) and the positive ionization mode (**b**). The rows correspond to the different input kernels while the columns indicate the output kernels.

3.3. Weights Learned by the Aggregation Model

Next, we turn our attention to the proposed score aggregation method (IOKRfusion). We first visualize the weights learned for the individual models in the two combined models (we have separate negative and positive mode models). The weights are shown in Figure 3.

We first notice that all models with PPKr as the input kernel are highly weighted in the combined model, regardless of the output kernel or whether IOKR or IOKRreverse is used. This is true for both combined models (positive and negative modes). In addition, the high aggregation weights tend to appear for models that are good predictors in the standalone setting as well (c.f. Figure 2), indicating that the models have complementarity besides good individual performance.



(**a**) Negative ionization mode

(**b**) Positive ionization mode

Figure 3. Heatmap of the weights learned during the score aggregation in the negative ionization mode (**a**) and the positive ionization mode (**b**). The weights have been averaged over the five cross-validation folds.

3.4. Results for Combined Models

Next, we report on the predictive performance of different aggregated models, including the early fusion MKL approaches and the proposed IOKRfusion approach relying on late fusion. We compare the results of score aggregation to IOKR Unimkl and CSI:FingerID. IOKR Unimkl denotes using IOKR with a linear combination of the 15 kernels as the single input kernel. In this combination, the same weight, 1, is given to each input kernel. This approach has previously shown to be a competitive way to combine input kernels for IOKR models [9]. For IOKRfusion, we show separately the top-k accuracy of the model restricted to combining the 60 IOKR models (4*15) and the model restricted to combining the 60 IOKRreverse models, as well as the combination including both IOKR and IOKRreverse models. The results obtained are shown in Table 2.

We first note that IOKRfusion aggregating all scores (IOKR and IOKRreverse) gives the best results by a significant margin in both negative and positive mode. Interestingly, the two restricted models, IOKRfusion aggregating either IOKR scores or IOKRreverse scores, do not give a consistent improvement over the IOKR Unimkl variants. Among the IOKR Unimkl variants, there is no clear winner: positive and negative mode seem to favor different output kernels, but the differences are relatively small.

Table 2. Comparison of the top-k accuracy between CSI:FingerID, IOKR Unimkl and IOKRfusion in the negative and positive ionization modes. The highest top-k accuracies are shown in boldface.

Method	**Negative Mode**			**Positive Mode**		
	Top-1	**Top-5**	**Top-10**	**Top-1**	**Top-5**	**Top-10**
CSI:FingerID	31.9	60.2	69.9	36.0	67.5	76.5
IOKR Unimkl - Linear	30.1	58.8	68.6	34.9	66.9	76.0
IOKR Unimkl - Tanimoto	31.0	60.0	69.7	35.2	67.6	76.5
IOKR Unimkl - Gaussian	31.0	60.3	69.6	35.0	67.7	76.3
IOKR Unimkl - Gaussian Tanimoto	30.9	61.0	70.5	33.9	66.5	75.2
IOKRfusion - only IOKR scores	28.4	57.0	67.2	33.5	64.4	73.4
IOKRfusion - only IOKRreverse scores	30.1	60.4	71.4	37.6	69.2	77.9
IOKRfusion - all scores	**32.1**	**62.4**	**71.8**	**37.8**	**69.7**	**78.4**

In Figures 4 and 5, we visualize the top-k accuracies of IOKR Unimkl and CSI:FingerID with the combination of all scores (IOKRfusion) in the negative (Figure 4) and positive (Figure 5) ionization modes. The plots in these figures also represent the top-k accuracy difference compared to the top-k accuracy obtained with CSI:FingerID. We observe that IOKRfusion consistently obtains better results than the other approaches. In the positive ionization mode, the score aggregation model improves upon CSI:FingerID and all the IOKR Unimkl models by around two percentage points for top-1 to top-10 accuracy. In the negative ionization mode, we observe a similarly consistent increase of one percentage point compared to the other models.

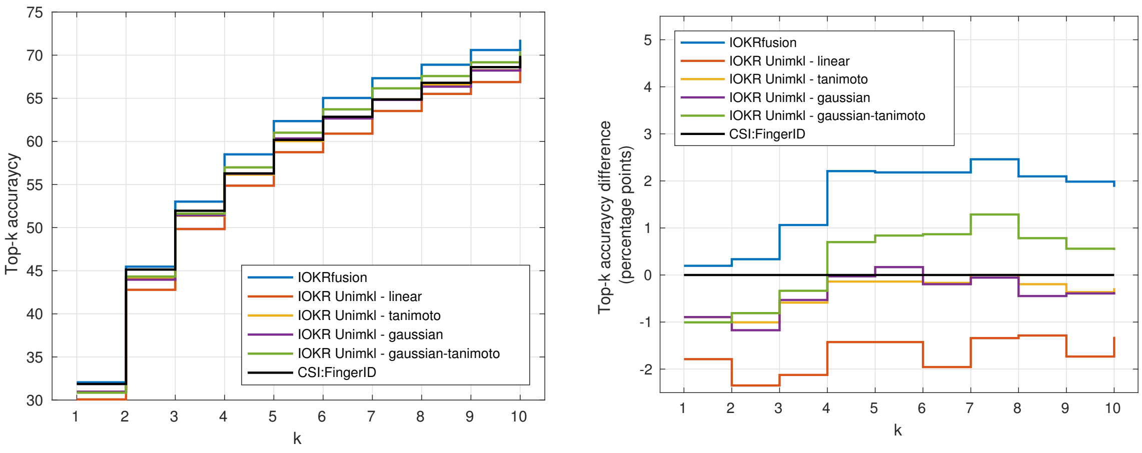


Figure 4. Plot of the top-k accuracy for IOKR Unimkl, CSI:FingerID and IOKRfusion in the negative ionization mode.

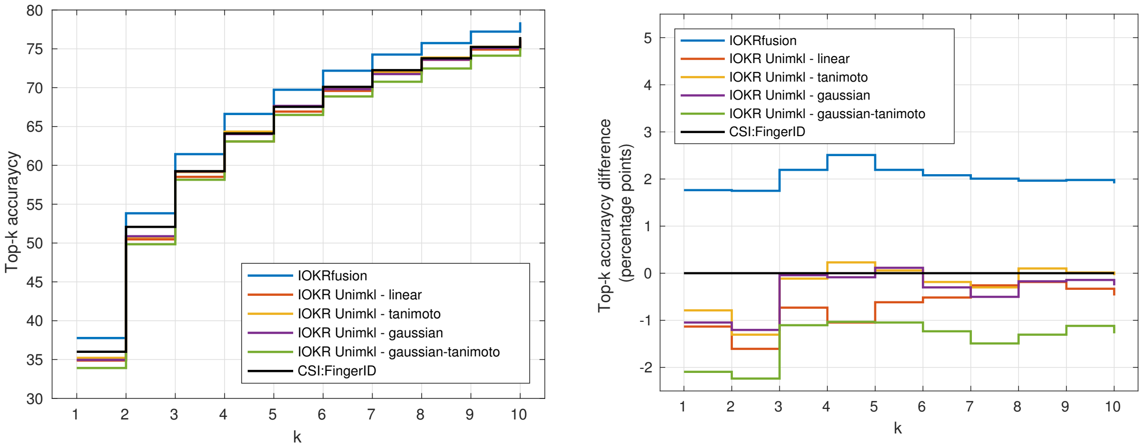


Figure 5. Plot of the top-k accuracy for IOKR Unimkl, CSI:FingerID and IOKRfusion in the positive ionization mode.

3.5. Running Times

We evaluated the running times of the different approaches on the negative dataset using 2859 spectra in the training set and 719 spectra in the test set (see Table 3). In this evaluation, we fixed the values of the different hyperparameters. In the score aggregation algorithm, we set the batch size to 15 and the number of epochs to 30. For the IOKR and IOKRreverse models that use a single kernel in input, we evaluated the running times for each of the 15 input kernels and averaged the training and test times. All of the models were trained on a single computer without any GPU acceleration or special infrastructures.

From the table, we can see that all single kernel IOKR models are trained in less than 10 s each, while computing the predictions (computing the pre-image) takes the majority of the time, 1–10 min depending on the output kernel. IOKRreverse models are equally fast to train, but the predictions are heavier to compute than for IOKR, the time consumed being in the interval of 28–35 min depending on the output kernel. The models based on multiple kernel learning (Unimkl) are only slightly more demanding to train, 4–12 s per model.

Computing the IOKRfusion model is comparatively very efficient after the component models have been trained and their predictions extracted. Learning a combination of 120 models: 15 (input kernels) × 4 (output kernels) × 2 (IOKR and IOKRreverse models), took slightly over three minutes, corresponding to around 1.5 s amortized time per model. Testing is much faster still, taking only 0.1 s in total, starting from the scores of the individual models.

The running times for CSI:FingerID are not included, but it has been shown in [9] that IOKR Unimkl is approximately 7000 times faster to train and is faster to test than CSI:FingerID.

Table 3. Running times for the training and the test steps.

Method	Training Time	Test Time
IOKR - linear	0.85 s	1 min 15 s
IOKR - tanimoto	3.9 s	7 min 40 s
IOKR - gaussian	7.2 s	8 min 38 s
IOKR - gaussian-tanimoto	7.6 s	8 min 44 s
IOKRreverse - linear	3.9 s	28 min 20 s
IOKRreverse - tanimoto	4.1 s	33 min 57 s
IOKRreverse - gaussian	7.4 s	34 min 49 s
IOKRreverse - gaussian-tanimoto	7.5 s	35 min 4 s
IOKR Unimkl - linear	4.3 s	1 min 10 s
IOKR Unimkl - tanimoto	8.7 s	7 min 52 s
IOKR Unimkl - gaussian	11.7 s	8 min 28 s
IOKR Unimkl - gaussian-tanimoto	11.9 s	8 min 42 s
IOKRfusion	3 min 3 s	0.1 s

4. Discussion

In this paper, we presented extensions to the IOKR framework that were shown to improve the identification rates of molecular structures from the MS/MS data. The first extension, IOKRreverse, changes the learning setting so that the regression problem is performed in the input (MS/MS) feature space rather than the output feature space. Using IOKRreverse as a standalone model in the positive ionization mode improved the IOKR results, but similar behaviour was not observed in the negative ionization mode.

The IOKRreverse model, which with a slight abuse of concepts, could be thought as a 'in silico fragmentation model' in that the model implicitly predicts a feature map of an MS/MS spectrum from the molecular structure. However, we must stress that there is no explicit fragmentation model present in IOKRreverse. Implicit fragmentation model could be seen if the input kernel is based on a fragmentation tree. Then, IOKRreverse can be interpreted as mapping molecular structures into a feature space where fragmentation trees are embedded, and the pre-image problem finds the molecular structure whose predicted fragmentation tree embedding is closest to the predicted one.

The proposed IOKRfusion approach, which combines several IOKR and IOKRreverse models trained with individual input and output kernels, obtained the best results in both negative and positive ionization mode, showing the potential of the approach. In particular, we note the late fusion approach, used by IOKRfusion, improves over the early fusion MKL approach Unimkl, which was previously found to be the best choice for CSI:IOKR [9]. In addition, the IOKRfusion approach turned out to outperform CSI:FingerID, which relies on an ALIGNF algorithm for MKL early fusion. The proposed IOKRfusion approach is also extremely fast to train and test. The dominant time cost is the extraction of the predictions of the individual models to be combined. In conclusion, IOKRfusion can be seen to maintain the computational efficiency of the IOKR framework, while improving the small molecule identification accuracy.

Supplementary Materials: The following are available online at http://www.mdpi.com/2218-1989/9/8/160/s1, Figure S1: Boxplots of the input kernel values for pairs of MS/MS spectra having the same molecular structure.

Author Contributions: Conceptualization: C.B., A.B., F.d.-B. and J.R.; methodology: C.B., F.d.-B. and J.R.; software: C.B.; validation: C.B.; formal analysis: C.B., F.d.-B., and J.R. investigation: C.B.; writing—original draft: C.B., J.R. and F.d.-B.; visualization: C.B.

Funding: The work of J.R. has been in part supported by Academy of Finland grants 310107 (MACOME) and 313268 (TensorBiomed).

Acknowledgments: We acknowledge the computational resources provided by the Aalto Science-IT project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, M.; Carver, J.J.; Phelan, V.V.; Sanchez, L.M.; Garg, N.; Peng, Y.; Nguyen, D.D.; Watrous, J.; Kapono, C.A.; Luzzatto-Knaan, T.; et al. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. *Nat. Biotechnol.* **2016**, *34*, 828–837. [CrossRef] [PubMed]
2. Horai, H.; Arita, M.; Kanaya, S.; Nihei, Y.; Ikeda, T.; Suwa, K.; Ojima, Y.; Tanaka, K.; Tanaka, S.; Aoshima, K.; et al. MassBank: A public repository for sharing mass spectral data for life sciences. *J. Mass Spectrom.* **2010**, *45*, 703–714. [CrossRef] [PubMed]
3. Nguyen, D.H.; Nguyen, C.H.; Mamitsuka, H. Recent advances and prospects of computational methods for metabolite identification: A review with emphasis on machine learning approaches. *Briefings Bioinform.* **2018**. [CrossRef] [PubMed]
4. Heinonen, M.; Shen, H.; Zamboni, N.; Rousu, J. Metabolite identification and molecular fingerprint prediction through machine learning. *Bioinformatics* **2012**, *28*, 2333–2341. [CrossRef] [PubMed]
5. Shen, H.; Zamboni, N.; Heinonen, M.; Rousu, J. Metabolite identification through machine learning—Tackling CASMI challenge using fingerID. *Metabolites* **2013**, *3*, 484–505. [CrossRef] [PubMed]

6. Djoumbou-Feunang, Y.; Pon, A.; Karu, N.; Zheng, J.; Li, C.; Arndt, D.; Gautam, M.; Allen, F.; Wishart, D.S. CFM-ID 3.0: Significantly Improved ESI-MS/MS Prediction and Compound Identification. *Metabolites* **2019**, *9*, 72. [CrossRef] [PubMed]
7. Allen, F.; Pon, A.; Wilson, M.; Greiner, R.; Wishart, D. CFM-ID: A web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. *Nucleic Acids Res.* **2014**, *42*, W94–W99. [CrossRef] [PubMed]
8. Dührkop, K.; Shen, H.; Meusel, M.; Rousu, J.; Böcker, S. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 12580–12585. [CrossRef] [PubMed]
9. Brouard, C.; Shen, H.; Dührkop, K.; d'Alché-Buc, F.; Böcker, S.; Rousu, J. Fast metabolite identification with Input Output Kernel Regression. *Bioinformatics* **2016**, *32*, i28–i36. [CrossRef]
10. Brouard, C.; Bach, E.; Böcker, S.; Rousu, J. Magnitude-preserving ranking for structured outputs. In Proceedings of the Asian Conference on Machine Learning, Seoul, Korea, 15–17 November 2017; pp. 407–422.
11. Laponogov, I.; Sadawi, N.; Galea, D.; Mirnezami, R.; Veselkov, K.A. ChemDistiller: an engine for metabolite annotation in mass spectrometry. *Bioinformatics* **2018**, *34*, 2096–2102. [CrossRef]
12. Nguyen, D.H.; Nguyen, C.H.; Mamitsuka, H. SIMPLE: Sparse Interaction Model over Peaks of moLEcules for fast, interpretable metabolite identification from tandem mass spectra. *Bioinformatics* **2018**, *34*, i323–i332. [CrossRef] [PubMed]
13. Nguyen, D.H.; Nguyen, C.H.; Mamitsuka, H. ADAPTIVE: leArning DAta-dePendenT, concIse molecular VEctors for fast, accurate metabolite identification from tandem mass spectra. *Bioinformatics* **2019**, *35*, i164–i172. [CrossRef]
14. Dührkop, K.; Fleischauer, M.; Ludwig, M.; Aksenov, A.A.; Melnik, A.V.; Meusel, M.; Dorrestein, P.C.; Rousu, J.; Böcker, S. SIRIUS 4: A rapid tool for turning tandem mass spectra into metabolite structure information. *Nat. Methods* **2019**, *16*, 299–302. [CrossRef] [PubMed]
15. CSI:FingerID Passed 10 Million Compound Queries. Available online: https://bio.informatik.uni-jena.de/2019/01/csifingerid-passed-10-million-compound-queries/ (accessed on 26 January 2019).
16. Schymanski, E.L.; Ruttkies, C.; Krauss, M.; Brouard, C.; Kind, T.; Dührkop, K.; Allen, F.; Vaniya, A.; Verdegem, D.; Böcker, S.; et al. Critical assessment of small molecule identification 2016: Automated methods. *J. Cheminform.* **2017**, *9*, 22. [CrossRef] [PubMed]
17. Webpage of CASMI 2017 contest. Available online: http://casmi-contest.org/2017/index.shtml (accessed on 31 July 2019).
18. Wolf, S.; Schmidt, S.; Müller-Hannemann, M.; Neumann, S. In silico fragmentation for computer assisted identification of metabolite mass spectra. *BMC Bioinform.* **2010**, *11*, 148. [CrossRef] [PubMed]
19. Ruttkies, C.; Schymanski, E.L.; Wolf, S.; Hollender, J.; Neumann, S. MetFrag relaunched: incorporating strategies beyond in silico fragmentation. *J. Cheminform.* **2016**, *8*, 3. [CrossRef] [PubMed]
20. Shen, H.; Dührkop, K.; Böcker, S.; Rousu, J. Metabolite identification through multiple kernel learning on fragmentation trees. *Bioinform.* **2014**, *30*, i157–i164. [CrossRef] [PubMed]
21. Bakir, G.H.; Hofmann, T.; Schölkopf, B.; Smola, A.J.; Taskar, B.; Vishwanathan, S.V.N. *Predicting Structured Data (Neural Information Processing)*; The MIT Press: Cambridge, MA, USA, 2007.
22. Brouard, C.; Szafranski, M.; d'Alché-Buc, F. Input Output Kernel Regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *J. Mach. Learn. Res.* **2016**, *17*, 1–48.
23. Cortes, C.; Mohri, M.; Rostamizadeh, A. Algorithms for Learning Kernels Based on Centered Alignment. *J. Mach. Learn. Res.* **2012**, *13*, 795–828.
24. Hazan, T.; Keshet, J.; McAllester, D.A. Direct loss minimization for structured prediction. In Proceeding of Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–11 December 2010; pp. 1594–1602.
25. Bolton, E.; Wang, Y.; Thiessen, P.; Bryant, S. Chapter 12—PubChem: Integrated platform of small molecules and biological activities. *Annu. Rep. Comput. Chem.* **2008**, *4*, 217–241.
26. Radovanovic, M.; Nanopoulos, A.; Ivanovic, M. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *J. Mach. Learn. Res.* **2010**, *11*, 2487–2531.
27. Shigeto, Y.; Suzuki, I.; Hara, K.; Shimbo, M.; Matsumoto, Y. Ridge regression, hubness, and zero-shot learning. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin, Germany, 2015; pp. 135–151.

28. Larochelle, H.; Erhan, D.; Bengio, Y. Zero-data Learning of New Tasks. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, IL, USA, 13–17 July 2008; pp. 646–651.
29. Xian, Y.; Schiele, B.; Akata, Z. Zero-Shot Learning—The Good, the Bad and the Ugly. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 3077–3086.
30. Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 423–443. [CrossRef]
31. Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.* **2005**, *6*, 1453–1484.
32. Böcker, S.; Rasche, F. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinfomatics* **2008**, *24*, i49–i55. [CrossRef]
33. Böcker, S.; Dührkop, K. Fragmentation trees reloaded. *J. Cheminform.* **2016**, *8*, 5. [CrossRef]
34. Dührkop, K. Computational Methods for Small Molecule Identification. Ph.D. Thesis, Friedrich-Schiller-Universität Jena, Jena, Germany, 2018.
35. Ralaivola, L.; Swamidass, S.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Netw.* **2005**, *18*, 1093–1110. [CrossRef]
36. Willighagen, E.L.; Mayfield, J.W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O.; et al. The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *J. Cheminform.* **2017**, *9*, 33. [CrossRef]
37. Klekota, J.; Roth, F. Chemical substructures that enrich for biological activity. *Bioinformatics* **2008**, *24*, 2518–2525. [CrossRef]
38. Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754. [CrossRef]