Classifying the toxicity of pesticides to honey bees via support vector machines with random walk graph kernels

Ping Yang[1], E. Adrian Henle[1], Xiaoli Z. Fern[2], and Cory M. Simon[1*]

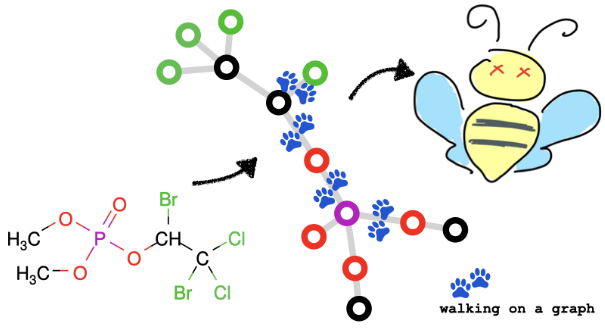
[1]School of Chemical, Biological, and Environmental Engineering. Oregon State University. Corvallis, OR. USA.
[2]School of Electrical Engineering and Computer Science. Oregon State University. Corvallis, OR. USA.
[*]Cory.Simon@oregonstate.edu

March 8, 2022

Abstract

Pesticides benefit agriculture by increasing crop yield, quality, and security. However, pesticides may inadvertently harm bees, which are agriculturally and ecologically vital as pollinators. The development of new pesticides—driven by pest resistance to and demands to reduce negative environmental impacts of incumbent pesticides—necessitates assessments of pesticide toxicity to bees.

We leverage a data set of 382 molecules labeled from honey bee toxicity experiments to train a classifier that predicts the toxicity of a new pesticide molecule to honey bees. Traditionally, the first step of a molecular machine learning task is to explicitly convert molecules into feature vector representations for input to the classifier. Instead, we (i) adopt the fixed-length random walk graph kernel to express the similarity between any two molecular graphs and (ii) use the kernel trick to train a support vector machine (SVM) to classify the bee toxicity of pesticides represented as molecular graphs. We assess the performance of the graph-kernel-SVM classifier under different walk lengths used to describe the molecular graphs. The optimal classifier, with walk length 5, achieves an (mean over 100 runs) accuracy, precision, and recall of 0.83, 0.71, and 0.72 on a test data set.



1 Introduction

1.1 Pesticide toxicity to bees

Pesticides (incl. insecticides, fungicides, and herbicides) are used in agriculture as an economic means to control weeds, pests, and pathogens. Thereby, pesticides increase expected crop yield and quality and contribute to food security [1–4]. However, widespread pesticide use has negative externalities on both aquatic and terrestrial ecosystems and human health [5–9]. For example, pesticides can harm agriculturally beneficial species not deliberately targeted, such as earthworms and bees [10, 11].

Though still under debate, extensive pesticide use in agriculture may play a role [11–16] in the widespread decline [17–20] of bee populations (see Ref. [21] for a synopsis) via both lethal and sublethal toxicity [11, 21]. Harms to bee populations are especially concerning because bees are vital for agricultural production [22]: (1, primary value) bees serve as pollen vectors[1] for many plants that produce fruits, vegetables, nuts, oil, and stimulants for human consumption [24, 25]; (2, secondary value [26]) honey bees produce honey and beeswax. In addition, bees are ecologically valuable pollen vectors for plants in natural habitats [27].

Because insect [28], weed [29], and fungi [30] populations can develop resistance to an insecticide, herbicide, and fungicide, respectively, new pesticides must be continually discovered and deployed [1]. In addition, new pesticide development is driven by the aim to reduce negative environmental impacts of incumbent ones [31]. Virtual screenings can accelerate the discovery of new pesticides operating under a known mechanism. For example, suppose a protein in an insect is a known target for insecticides. Then, computational protein-ligand docking [32] can score candidate compounds for insecticide activity, informing experimental campaigns. [33–39] However, newly proposed pesticides must also be assessed for toxicity to honey bees [40] (see US EPA website [41]).

A computational model that accurately predicts the toxicity of pesticides to bees is valuable [42] (i) as an *in silico* toxicity filter in virtual and experimental screenings of compounds for pesticide activity; (ii) in emergency situations where an immediate assessment of toxicity risk is needed; and (iii) to focus greater scrutiny and more thorough toxicity assessments on existing and new pesticides predicted to be toxic. Generally, training machine learning models to predict the toxicity of compounds to biological organisms is an active area of research [43, 44]. And, indeed, open data from bee toxicity experiments [45–56] have been leveraged to train machine learning models to computationally predict the toxicity of pesticides to bees [57–63].

1.2 Representing molecules for supervised machine learning tasks

A flurry of research activity is devoted to the data-driven prediction of the properties of molecules via supervised machine learning [64]. An essential starting point is to design a machine-readable

[1] Bees visit the flowers of plants (angiosperms) to collect pollen or nectar as a food source. In the process, bees [inadvertently] transfer pollen from the anther of one flower to the stigma of another flower, a necessary step in the production of seeds and fruit for many plants. [23]

representation of the molecule for input to the machine learning model [65–67].

A vertex- and edge-labeled graph (vertices = atoms, edges = bonds, vertex label = element, edge label = bond order) is a fundamental representation of the concept of a small molecule[2]. For many classes of molecules, the mapping of the concept of a molecule to a molecular graph is one-to-one. Though, molecular graph representations break down for certain classes of molecules [65] and are invariant to 3D structure and stereoisomerism [68].

Classical machine learning algorithms operate on inputs that lie in a Euclidean vector space. Consequently, much research is devoted to the design of fixed-size, information-rich feature vectors ("fingerprints") that encode salient features of the molecule [65, 69]. Many fingerprinting methods, e.g. Morgan circular fingerprints [70], extract topological features directly from the molecular graph [71] to produce a "bag of fragments" representation [72]. Other hand-crafted molecular feature vectors include chemical, electronic, and structural/shape (3D) properties of the molecule as well [65, 72].

Two advanced supervised machine learning approaches circumvent explicit hand-crafting of vector representations of molecular graphs:

1. graph representation learning [73], such as message passing neural networks (MPNNs) [74, 75] that *learn* task-specific vector representations of molecular graphs for prediction tasks in an end-to-end manner
2. graph kernels [76–81], which (loosely speaking) measure the similarity between any two input graphs, allowing for the use of kernel methods [82], such as support vector machines [83], kernel regression/classification [82], and Gaussian processes [84], for prediction tasks.

I.e., MPNNs and kernel methods take the molecular graph representation directly as an input, bypassing manual feature vector engineering as the first step of a molecular machine learning task.

While MPNNs are powerful models, they require large amounts of labeled training data. Kernel methods with graph kernels, in comparison, are more appropriate when training data are limited, as they (i) are easier to train, possess fewer hyper-parameters, and are less susceptible to overfitting and (ii) can perform on par with MPNNs for molecular prediction tasks [85].

1.3 Our contribution: building a bee toxicity classifier of pesticides via random walk graph kernels

Our objective is to train and evaluate a graph-kernel-SVM classifier to predict the toxicity of pesticide molecules to bees. Enabling a machine learning approach, the BeeToxAI project [57] compiled labeled data from bee toxicity experiments—composed of (molecule, bee toxicity) pairs. We adopt the fixed-length random walk kernel to describe the similarity between any two molecular graphs representing two pesticides. As we explain, this graph kernel describes a molecule by the

[2] If we wished to communicate a small molecule to an intelligent, extraterrestrial life form that has just arrived on earth and does not know our language, we would likely sketch a vertex- and edge-labeled, undirected graph.

distribution of sequences of vertex- and edge-labels along all walks of a given length on the molecular graph. Our classifier achieves an accuracy, precision, and recall of 0.83, 0.71, and 0.72.

2 Problem setup: classifying toxicity of a pesticide to honey bees

The pesticide toxicity classification task. We wish to construct a classifier $f : \mathcal{G} \rightarrow \{-1, 1\}$ that maps any molecular graph G (see Sec. 3.1) representing a pesticide molecule to a predicted binary label $\hat{y} = f(G)$, where $\hat{y} = 1$ is toxic to honey bees (*Apis mellifera*) and $\hat{y} = -1$ is nontoxic. The classifier f is valuable as a cheap-to-evaluate "surrogate model" of an expensive bee toxicity experiment. See Fig. 1a.

The labeled bee toxicity data set. We have labeled data $\{(G_n, y_n)\}_{n=1}^{N}$ composed of N examples of molecular graphs $G_n \in \mathcal{G}$ representing pesticide and pesticide-like molecules and their experimentally-determined labels $y_n \in \{-1, 1\}$ (1: toxic, -1: nontoxic). Particularly, we took the acute contact toxicity data set, compiled by the BeeToxAI project [57], containing $N = 382$ molecules—113 toxic and 269 nontoxic (see Fig. 1b).

The machine learning approach: support vector machine based on a random walk graph kernel. We wish to leverage the labeled bee toxicity data set to train a support vector machine (see Sec. 3.6) as the classifier $f(G)$. A support vector machine (SVM) is a versatile machine learning algorithm that aims to find the maximum-margin separator (a hyperplane) between the positive and negative training examples in a mapped feature space. The mapped feature space does not need to be explicitly constructed. Instead, kernel functions can be used to perform the needed operations (dot product) in the mapped feature space. In this work, we use the fixed-length random walk graph kernel (see Sec. 3.4) as our kernel function, which [implicitly] converts each molecular graph into a feature vector representation for the SVM via a fixed-length random walk feature map (see Sec. 3.3).

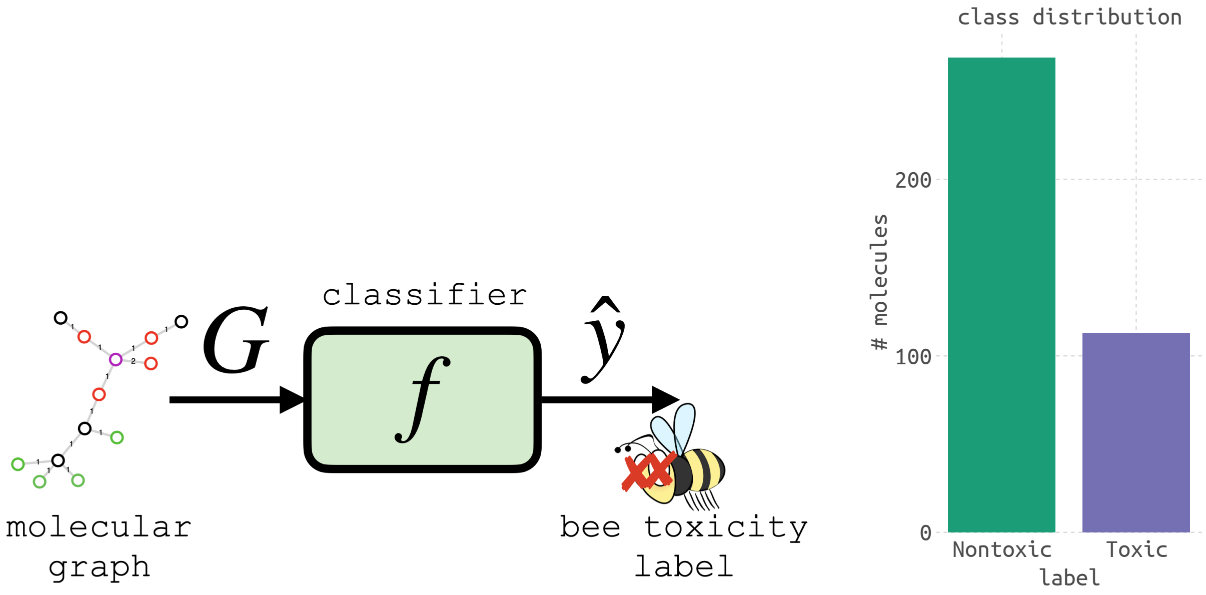
3 Methods

3.1 The vertex- and edge-labeled graph representation of a molecule

A fundamental representation of a molecule is as a vertex- and edge-labeled, undirected graph $G = (\mathcal{V}, \mathcal{E}, \ell_v, \ell_e)$, with:

- $\mathcal{V} = \{v_1, ..., v_N\}$ the set of vertices representing the N atoms (excluding hydrogen atoms[3])
- $\mathcal{E}$ the set of edges representing chemical bonds; $e = \{v_i, v_j\} \in \mathcal{E}$ iff the atoms represented by vertices $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ are bonded.

[3] We exclude H atoms in the molecular graph to avoid redundancy. E.g., an H bonded to a C can be inferred from the order of the bond between C and another atom, encoded in the edge labels.



(a) The prediction task.

(b) Class distribution.

Figure 1: Problem setup. (a) Our objective is to train a classifier f that maps a molecular graph G to a binary prediction $\hat{y}$ of the bee toxicity of the pesticide molecule it represents. (b) The BeeToxAI [57] bee toxicity data set we leverage to train a machine learning model as the classifier f contains 382 molecules with the shown label distribution.

- $\ell_v : \mathcal{V} \rightarrow \{\text{C}, \text{N}, \text{O}, \text{S}, \text{P}, \text{F}, \text{Cl}, \text{I}, \text{Br}, \text{Si}, \text{As}\}$ the vertex-labeling function that provides the chemical element of each vertex (atom).
- $\ell_e : \mathcal{E} \rightarrow \{1, 2, 3, a\}$ ("a" for aromatic) the edge-labeling function that provides the bond order of each edge (bond).

For example, Fig. 2c visualizes the molecular graph representation of the molecule in Fig. 2a. This *molecular graph* representation of a molecule describes its topology and is invariant to translations and rotations of the molecule and to bond stretching, bending, and rotation.

Let $\mathcal{G}$ be the set of possible molecular graphs, so $G \in \mathcal{G}$.

3.2 Walks on a molecular graph and label sequences along them

Random walk feature maps and graph kernels describe molecular graphs by the set of walks on them and the label sequences along these walks.

A walk. A *walk* w of length L on a molecular graph G is a sequence of vertices such that consecutive vertices are joined by an edge:

$$w = (v_1, ..., v_{L+1}) \text{ s.t. } \{v_i, v_{i+1}\} \in \mathcal{E} \text{ for } i \in \{1, ..., L\} \tag{1}$$

The length L refers to the number of edges (not necessarily unique) traversed along the walk. E.g., a length-3 walk on a molecular graph is highlighted in Fig. 2c.

Let $\mathcal{W}_L(G)$ be the set of all possible walks of length L on a graph G.

The label sequence of a walk. The *label sequence* $s = \ell_w(w)$ of a walk $w = (v_1, ..., v_{L+1})$ gives the progression of vertex and edge labels along the walk:

$$\ell_w(w) = [\ell_v(v_1), \ell_e(\{v_1, v_2\}), ..., \ell_e(\{v_L, v_{L+1}\}), \ell_v(v_{L+1})] =: s. \tag{2}$$

Let $\mathcal{S}_L = \{s_1, ..., s_{S_L}\}$ be the set of all possible label sequences among length-L walks on the space of molecular graphs, $\mathcal{G}$—so $|\mathcal{S}_L| = S_L$.

3.3 The fixed-length random walk graph-to-vector feature map

Both classical and kernel machine learning methods explicitly and implicitly, respectively, rely on a feature map for prediction tasks on molecular graphs. Generally, a feature map maps graphs into a vector space.

The fixed-length random walk feature map maps a molecular graph to a vector whose entries give the number of walks of a given length on the graph with each of the different possible vertex- and edge-label sequences. The idea is to describe a molecular graph by the distribution of label sequences of fixed-length random walks on it [86–88]—pertaining to equipoise random walks on a molecular graph [89].

The fixed-length-L feature map $\phi^{(L)} : \mathcal{G} \rightarrow \mathbb{R}^{S_L}$ constructs a vector representation of a graph $G \in \mathcal{G}$ whose element i is a count of length-L walks on G with label sequence s_i:

$$\phi^{(L)}(G) :=[\phi_1^L(G), ..., \phi_{S_L}^{(L)}(G)] \tag{3}$$

$$\text{where } \phi_i^{(L)}(G) := |\{w \in \mathcal{W}_L(G) : \ell_w(w) = s_i\}|. \tag{4}$$

As a length $L = 0$ walk constitutes an atom, the feature map $\phi^{(0)}(G)$ lists counts of atom types in the molecule. As a length $L = 1$ walk constitutes two (ordered) atoms joined by a bond, the feature map $\phi^{(L)}(G)$ lists counts of each particular (ordered) pairing of atoms joined by a particular bond type in the molecule.

For large L, explicit construction of $\phi^{(L)}(G)$ for all G in (a large subset of) $\mathcal{G}$ for a machine learning task may be infeasible because of the size of the set of possible label sequences $\mathcal{S}_L$ present in length-L walks on graphs[4]. [87] Thankfully, (1) kernel methods of machine learning can be cast to rely only on inner products $\phi^{(L)}(G) \cdot \phi^{(L)}(G')$ of pairs of vector representations of molecular graphs and (2) the fixed-length random walk kernel $k^{(L)}(G, G') = \phi^{(L)}(G) \cdot \phi^{(L)}(G')$ allows us to circumvent explicit construction of $\phi^{(L)}(G)$.

[4] Given V possible vertex labels and E possible edge labels, theoretically $|\mathcal{S}_L| = V^{L+1}E^L$ label sequences are possible, but of course chemistry imposes many constraints.

3.4 The fixed-length random walk graph kernels

The fixed length-L random walk graph kernel [86, 90] (L-RWGK) $k^{(L)} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ is a (symmetric, positive semidefinite) function such that evaluating $k(G, G')$ is implicitly equivalent to (i) mapping the two input graphs G and G' into the vector space $\mathbb{R}^{\mathcal{S}_L}$ via the fixed length L feature map $\phi^{(L)}$ then (ii) taking the inner product of these two vectors:

$$k^{(L)}(G, G') = \phi^{(L)}(G) \cdot \phi^{(L)}(G'). \tag{5}$$

Seen from eqn. 4, term i of $k^{(L)}(G, G')$ is the number of pairs of length-L walks—one in graph G, the other in graph G'—with label sequence $s_i \in \mathcal{S}_L$. So, $k^{(L)}(G, G')$ sums counts of pairs of length-L walks on the two graphs G, G' sharing a label sequence:

$$k^{(L)}(G, G') = \sum_{s \in \mathcal{S}_L} |\{w \in \mathcal{W}_L(G) : \ell_w(w) = s\}||\{w' \in \mathcal{W}_L(G') : \ell_w(w') = s\}|. \tag{6}$$

As the term associated with a label sequence s is nonzero iff *both* graphs G and G' possess a length-L walk with label sequence s, this sum may be restricted to be over the subset of label sequences in common between length-L walks on the two graphs, $\ell_w(\mathcal{W}_L(G)) \cap \ell_w(\mathcal{W}_L(G'))$.

Intuitively:

- The 0-RWGK $k^{(0)}(G, G')$ sums counts of pairs of atoms of a particular atom type between the two graphs G, G' and is equal to the number of nodes in the direct product graph $G_\times = G \times G'$.
- The 1-RWGK $k^{(1)}(G, G')$ sums counts of pairs of two particular (ordered) atoms joined by a particular bond between the two graphs G, G' and is equal to twice the number of edges in the direct product graph $G_\times = G \times G'$.

The optimal length L in the L-RWGK for a particular molecular prediction task may be found via a cross-validation procedure.

How do we evaluate the L-RWGK without explicitly performing the dot product in eqn. 5? Next, we presents the direct product graph to facilitate counting pairs of label sequences in common between walks on two graphs.

3.5 The direct product graph to compute RWGKs

Given two input graphs $G, G' \in \mathcal{G}$, we construct a new graph, the direct product graph $G_\times = G \times G' = (\mathcal{V}_\times, \mathcal{E}_\times, \ell_{v,\times}, \ell_{e,\times})$ for evaluating the L-RWGK $k^L(G, G')$ between G, G'. The direct product graph $G_\times$ is constructed such that there is a one-to-one mapping between (i) walks in $G_\times$ and (ii) pairs of walks—one on G and one on G'—with the same label sequence.

Definition of the direct product graph. Each vertex of the direct product graph $G_\times = G \times G'$ is an ordered pair of vertices—the first in G, the second in G'. The vertices of the direct product graph are constituted by the subset of pairs of vertices between G and G' with the same vertex label:

$$\mathcal{V}_\times := \{(v, v') \in \mathcal{V} \times \mathcal{V}' \mid \ell_v(v) = \ell'_v(v')\}. \tag{7}$$

An edge joins a pair of two vertices of the direct product graph $G_\times = G \times G'$ iff (i) the two involved vertices of G are joined by an edge in $\mathcal{E}$ and (ii) the two involved vertices of G' are joined by an edge in $\mathcal{E}'$ and (iii) these two edges in $\mathcal{E}$ and $\mathcal{E}'$ have the same label:

$$\begin{aligned}\mathcal{E}_\times := \{\{(u, u'), (v, v')\} \mid\ & (u, u') \in \mathcal{V}_\times \ \wedge \ (v, v') \in \mathcal{V}_\times \wedge \\ & \{u, v\} \in \mathcal{E} \ \wedge \ \{u', v'\} \in \mathcal{E}' \wedge \\ & \ell_e(\{u, v\}) = \ell'_e(\{u', v'\})\}. \end{aligned} \tag{8}$$

We equip the direct product graph $G_\times = G \times G'$ with vertex- and edge-labeling functions that give the (same) label of the involved vertices and edges in G and G':

$$\ell_{v,\times}((v, v')) = \ell_v(v) = \ell'_v(v') \tag{9}$$

$$\ell_{e,\times}(\{(u, u'), (v, v')\}) = \ell_e(\{u, v\}) = \ell'_e(\{u', v'\}). \tag{10}$$

For example, Fig. 2e visualizes the direct product graph of the two molecular graphs in Figs. 2c and 2d.

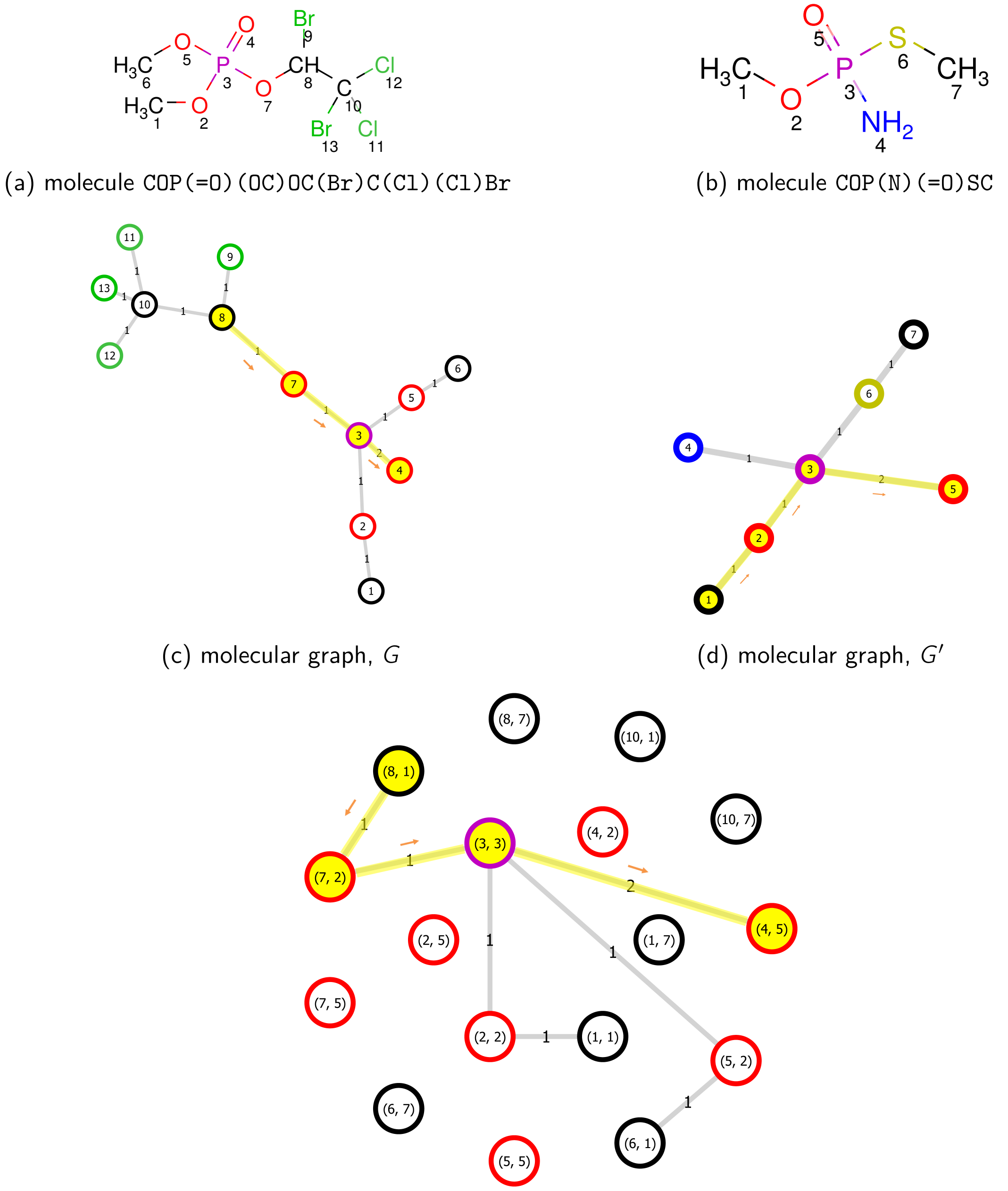
Utility of the direct product graph for evaluating the L-RWGK. By construction, any given length-L walk $w_\times$ on the direct product graph $G_\times = G \times G'$ with label sequence $\ell_{w,\times}(w_\times)$ pertains to a unique pair of walks $\{w, w'\}$, with $w \in \mathcal{W}_L(G)$, $w' \in \mathcal{W}_L(G')$, possessing the label sequence $\ell_w(w) = \ell'_w(w') = \ell_{w,\times}(w_\times)$, and vice versa (giving a bijection). Fig. 2 illustrates (see highlighted walks). Therefore, all three of the following quantities are equivalent:

- the number of length-L walks on the direct product graph $G_\times = G \times G'$
- the number of pairs of length-L walks on G and G' with the same label sequence
- via eqn. 6, the value of the L-RWGK $k^{(L)}(G, G')$.

The key to counting length-L walks on $G_x = G \times G'$—and thus to evaluating $k^{(L)}(G, G')$—lies in its $|\mathcal{V}_x| \times |\mathcal{V}_x|$ adjacency matrix A_x whose entry (i, j) is one if vertices $v_{x,i}, v_{x,j} \in \mathcal{V}_x$ are joined by an edge and zero otherwise. The number of walks of length L from vertex $v_{x,i}$ to vertex $v_{x,j}$ is given by element (i, j) of A_x^L. Therefore:

$$k^{(L)}(G, G') = \sum_{i=1}^{|\mathcal{V}_\times|} \sum_{j=1}^{|\mathcal{V}_\times|} [A_\times^L]_{i,j}. \tag{11}$$

Summary of evaluating the L-RWGK. Computing the L-RWGK $k^{(L)}(G, G')$, therefore, involves (i) constructing the direct product graph $G_\times = G \times G'$, (ii) building the adjacency matrix $A_\times$ of $G_\times$, (iii) computing the L-th power of A_x, $A_\times^L$, then (iv) summing its entries.



(a) molecule COP(=O)(OC)OC(Br)C(Cl)(Cl)Br

(b) molecule COP(N)(=O)SC

(c) molecular graph, G

(d) molecular graph, G'

(e) direct product graph, $G_\times = G \times G'$

Figure 2: Illustrating the (e) direct product graph $G_\times = G \times G'$ of two molecular graphs (c) G and (d) G' representing molecules in (a) and (b) (SMILES strings shown). Vertex labels in $G, G', G_\times$ indicated by color. Note the one-to-one correspondence between (i) a walk on $G_\times$ and (ii) two walks on G and G' with the same label sequence. We highlight one such correspondence in yellow.

3.6 Support vector machines (SVMs) as classifiers

A support vector machine (SVM) [82, 83, 91, 92] is a kernel method for binary classification. We have:

- a labeled training data set $\{(G_n, y_n)\}_{n=1}^{N}$, with $G_n \in \mathcal{G}$ and $y_n \in \{-1, 1\}$.
- the fixed-length feature map $\phi^{(L)} : \mathcal{G} \to \mathbb{R}^{S_L}$ and its associated graph kernel $k^{(L)} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$.

An SVM classifier draws a decision boundary in the feature space $\mathbb{R}^{S_L}$ with a hyperplane:

$$\hat{y} = f(G) = \text{sign}(w \cdot \phi^{(L)}(G) + b), \tag{12}$$

where $w \in \mathbb{R}^{S_L}$ and $b \in \mathbb{R}$ are learnable parameters. Assuming the two classes in the training data are separable by eqn. 12, the idea behind training an SVM is to seek the hyperplane that:

- separates the classes. I.e., $y_n(w \cdot \phi^{(L)}(G_n) + b) \geq 0$ for $n \in \{1, ..., N\}$.
- gives the maximal margin. I.e., maximizes the distance of the closest data points—the "support vectors"—to the hyperplane. Mathematically, if we define a support vector $\phi(G_i)$ as satisfying $y_i(w \cdot \phi(G_i) + b) = 1$, we wish for $1/||w||$ to be large.

In practice, the data are not separable, and we must have a "soft" margin that allows for misclassifications through slack variables.

The primal optimization problem associated with training a soft margin SVM is to find the parameters w, b that give the maximal (soft) margin while penalizing constraint violations:

$$\min_{w,b} \left(\frac{1}{2}||w||^2 + C \sum_{n=1}^{N} \xi_n \right) \tag{13}$$

$$\text{s.t. } \xi_n \geq 0 \text{ for } n \in \{1, ..., N\} \tag{14}$$

$$y_n(w \cdot \phi(G_n) + b) \geq 1 - \xi_n \text{ for } n \in \{1, ..., N\}. \tag{15}$$

The slack variable ξ_n associated with data vector $\phi(G_n)$ allows, if it is nonzero, violation of the constraint $y_n(w \cdot \phi(G_n) + b) \geq 1$ that it lies on the correct side of the decision boundary and outside of or on the margin. If $\xi_n > 1$, data point n is misclassified; if $0 < \xi_n < 1$, it is classified correctly but lies inside the margin. The hyperparameter $C \geq 0$ trades a large margin for constraint violations.

In practice, SVM is typically trained computationally by solving the Lagrangian dual of the optimization problem, a quadratic programming problem in N variables $\{\alpha_1, ..., \alpha_N\}$, where α_n is the Lagrange multiplier for the constraint on $\phi(G_n)$:

$$\max_{\alpha} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi^{(L)}(G_i) \cdot \phi^{(L)}(G_j) \right) \tag{16}$$

$$\text{s.t. } 0 \leq \alpha_n \leq C \text{ for } n \in \{1, ..., N\} \tag{17}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0, \tag{18}$$

where the optimal solution to the dual problem α^* and the optimal solution to the primal problem w^* satisfy:

$$w^* = \sum_{n=1}^{N} \alpha_n^* y_n \phi^{(L)}(G_n). \tag{19}$$

Importantly, the objective of the dual problem in eqn. 16 depends only on $\phi^{(L)}(G_i) \cdot \phi^{(L)}(G_j)$ of the training data, which we can replace with the graph kernel $k^{(L)}(G_i, G_j)$ (see eqn. 5) via the kernel trick! The kernel trick bypasses the explicit mapping of the graphs G and G' into the vector space $\mathbb{R}^{S_L}$ to compute $\phi^{(L)}(G) \cdot \phi^{(L)}(G')$; instead, we evaluate the kernel function between them, $k^{(L)}(G, G')$, via eqn. 11. Using eqn. 19, we can also rewrite the decision rule in eqn. 12 for a new graph G in terms of the kernel between it and the graphs in the training data set:

$$f(G) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n^* y_n k^{(L)}(G_n, G) + b^*\right). \tag{20}$$

An important characteristic of the solution to the dual problem is that many α_n^*'s could be/likely are zero. If $\alpha_n = 0$, data point (G_n, y_n) does not contribute to the decision rule in eqn.20. The subset of data vectors $\phi^{(L)}(G_n)$ for which $\alpha_n > 0$ constitute the support vectors. The decision rule in eqn. 20 depends only on how similar the new graph G is to the graphs constituting the support vectors.

Finally, we address the question of how to determine b^* in eqn. 20. One can show that (margin) support vectors such that $0 < \alpha_n < C$ lie on the margin and thus satisfy $y_i(w^* \cdot \phi(G_i) + b^*) = 1$. Together with eqn. 19, this gives an equation for b^* involving only the (margin) support vectors.

For more details on the SVM, consult Refs. [82, 91].

In practice, we store the inner products between all pairs of molecular graphs in a $N \times N$ Gram matrix $K^{(L)}$, whose element (i, j) gives the L-RWGK $k^{(L)}(G_i, G_j)$ between molecular graphs G_i and G_j.

Note the SVM will perform better if the feature vectors $\{\phi^{(L)}(G_1), ..., \phi^{(L)}(G_N)\}$ are first centered [93]. Again to avoid explicit construction of them, the double-centering trick [82] allows us to obtain the inner products of the centered feature vectors from the inner products of the uncentered feature vectors in the Gram matrix $K^{(L)}$. Particularly, the centered Gram matrix $\tilde{K}^{(L)} = CK^{(L)}C$ with centering matrix $C = I - \frac{1}{N}oo^\intercal$ (I the identity matrix, o a vector of ones). See Appendix B in Ref. [94].

3.7 Classification performance metrics

Performance metrics of a classifier $\hat{y} = f(G)$ include, measured over a labeled test data set:

- accuracy: fraction of examples classified correctly
- precision: among the examples classified as toxic ($\hat{y}_n = 1$), what fraction are truly toxic ($y_n = 1$)?

- recall: among the examples that are truly toxic ($y_n = 1$), what fraction are correctly predicted as toxic ($\hat{y}_n = 1$)?

4 Results

We now share the results of using a fixed length-L random walk graph kernel (L-RWGK) support vector machine (SVM) to classify the toxicity of pesticide molecules to honey bees. All data and Julia code for reproducibility is available at `github.com/SimonEnsemble/graph-kernel-SVM-for-toxicity-of-pesticides-to-bees`.

4.1 Machine learning procedures

Data preparation. We first converted the SMILES strings representing the pesticide molecules from the BeeToxAI project [57] into molecular graphs. Together with the outcome of the bee toxicity experiment, this gives a labeled data set $\{(G_n, y_n)\}_{n=1}^{N}$ described in Sec. 2.

Computing the Gram matrix. For $L \in \{0, ..., 12\}$, we compute the $N \times N$ Gram matrix $K^{(L)}$ whose element (i, j) gives the L-RWGK $k^{(L)}(G_i, G_j)$ (see Sec. 3.4) between molecular graphs G_i and G_j. We wrote our own code to construct the direct product graph $G_\times = G_i \times G_j$ and compute the L-RWGK $k^{(L)}(G_i, G_j)$ (see Sec. 3.5).

A train-test run. A train-test run of a L-RWGK-SVM (see Sec. 3.6), with L specified, comprises the following procedure. We randomly shuffle then split the examples into a 80%/20% train/test split via a stratified scheme to preserve the distribution of class labels in Fig. 1b. Using the training set only, we use K-fold ($K = 3$, again, stratified by class labels) cross-validation to determine the optimal C parameter of the SVM classifier. We choose C_{opt} as the one providing the L-RWGK-SVM with the maximal (mean over K folds) validation score (product of precision and recall)—a grid search over 15 values of $\log_{10} C$ between -5 and 1, equally spaced. Finally, we train a deployment SVM with $C = C_{opt}$ on all training data and evaluate its performance (precision, recall, and accuracy—see Sec. 3.7) on the test set.

N.b., for each SVM trained, we center the Gram matrix $K^{(L)}$ pertaining only to the training graphs via the double-centering trick [82]. We adopt a similar centering trick [94] for the Gram matrix giving the similarity of the test graphs with the training graphs when we feed it as input to the SVM for a prediction on the test set.

We used the SVC implementation and Gram matrix centerer in scikit-learn [95] with our precomputed Gram matrix. We scaled the C parameter in eqn. 13 seen by the slack variables pertaining to each class to balance penalization of constraint violations for each class.

Overall procedure. For $L \in \{0, ..., 12\}$, we conducted 100 (stochastic, owing to the random test/train and K-folds split) train/test runs, where we evaluated the performance of a C-hyperparameter-optimized, trained L-RWGK-SVM classifier on the hold-out test set for each run.

Run times. The majority of the computational run time for generating our results was in computing the 382×382 Gram matrix $K^{(L)}$. Using four cores, the run time ranged from less than five minutes ($L = 0, 1$) to ∼20-25 minutes for $L \geq 7$ (see SI).

4.2 Cross-validation results

Fig. 3a shows the validation scores (product of precision and recall) from the cross-validation procedure for each (i) C SVM hyperparameter and (ii) walk length L in the L-RWGK. Overall, the optimal C parameter tended to decrease with the walk length, consistent viewing the inverse of C as a regularization parameter that should increase when the representation of the examples are more complex.

4.3 Classification performance on the test set

Fig. 3b shows the classification performance, judged by mean accuracy, precision, and recall (see Sec. 3.7) on the hold-out test set of pesticide molecules, as a function of the walk length L of the L-RWGK used for the SVM. The optimal classifier, selected according to the maximal mean product of recall and precision in the cross-validation procedure, used the 5-RWGK and achieved a mean accuracy, precision, and recall of 0.83, 0.71, and 0.72 on the test data set. We conclude the optimal length L of the walk for expressing the similarity of pesticide molecules via the L-RWKGK is length $L = 5$.

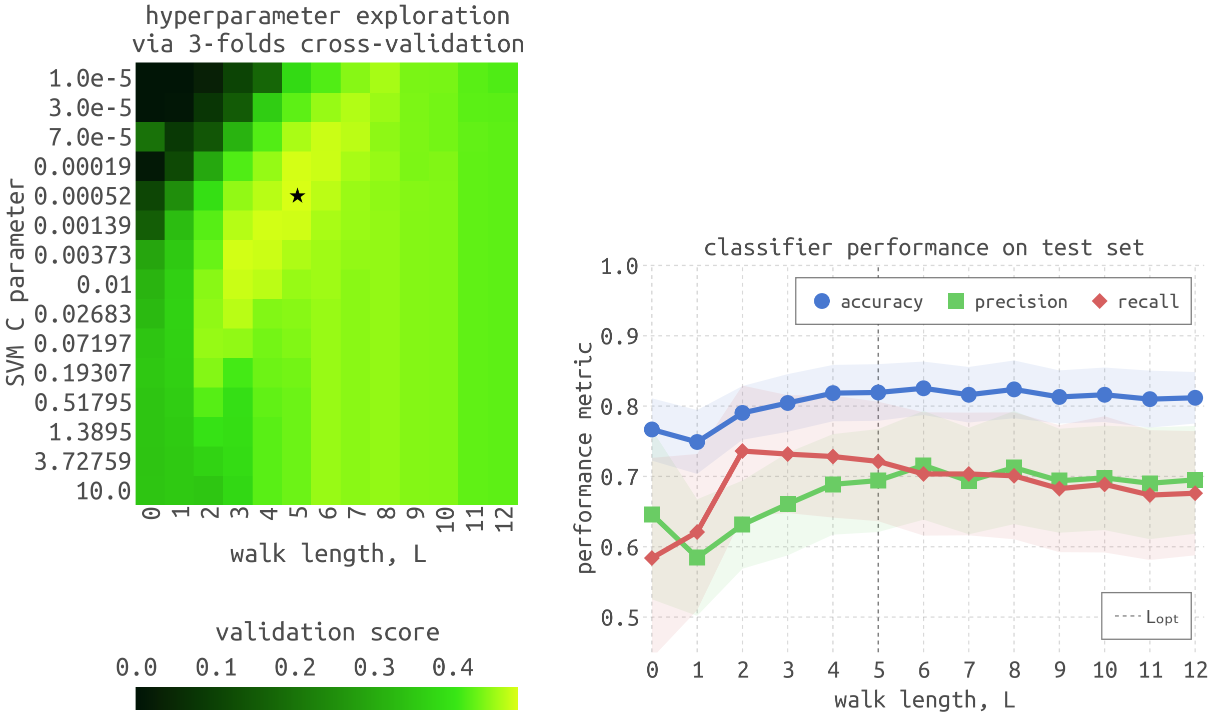
5 Discussion

We provided an exposition on the L-RWGK-SVM for molecular prediction tasks. The key idea behind the L-RWGK is to express the similarity of any two molecular graphs by the count of pairs of length L walks—one on each graph—with the same sequence of vertex- and edge-labels. We then leveraged the BeeToxAI [57] data set to train a L-RWGK-SVM classifier that predicts the toxicity of pesticide molecules to honey bees.

Graph kernels have been used before, with SVMs, Gaussian processes, and kernel regression, to predict the properties of molecules and materials in the chemical sciences [77,96], such as to classify proteins [97], score protein-protein interactions [98], predict methane uptake in nanoporous materials [99], predict atomization energy of molecules [100, 101], and predict thermodynamic properties of pure substances [102].

We mention that a Gaussian process model [84] using the L-RWGK would enable uncertainty quantification in the prediction.

Disadvantages of the L-RWGK include (i) its compute- and memory-intensity to evaluate, hence poor scalability to large molecules and large data sets [77] and (ii) tottering. Expanding on (ii): by definition, the vertices in a walk (see eqn. 1) may not be distinct (then, it would be a *path*). Thus, long walks that totter back and forth between the same few nodes—e.g., at the extreme: $w = (u, v, u, v, ..., u, v)$—are accounted for in the L-RWGK. These walks do not contribute



(a)

(b)

Figure 3: Average results over 100 (stochastic) runs of test/train splits. (a) Validation scores (product of precision and recall) during the 3-fold cross-validation procedure to tune (i) the SVM C parameter and (ii) the length L of the walks in the L-RWGK. The $\star$ indicates the optimal SVM classifier. (b) Toxicity classification performance of the deployment SVM (with the optimal C parameter) as a function of the length L of the walks in the L-RWGK. Shaded bands show standard deviation.

extra information about the similarity of two graphs—e.g., for our extreme example, no more information beyond the length-2 walk (u, v). Tottering thus could lead to a "dilution" of the similarity metric expressed by the random walk kernel. [90] Modification of the random walk kernel can prevent tottering walks [103] from contributing to the similarity metric.

The L-RWGK can be generalized further by defining a kernel between two walks w and w' as a product of the kernel between the edges and vertices along the walk [77, 87, 97]. A (non-Dirac) kernel between vertices could account for similarity of chemical elements.

In addition to the fixed-length-L random walk kernel we employed, the (i) max-length-L random walk kernel and (ii) geometric random walk kernel count pairs of length-ℓ walks with a shared label sequence (i) for $\ell \in \{0, ..., L\}$ and (ii) $\ell \in \{0, ..., \}$ [87, 90].

In addition to random walk kernels, other graph kernels can be used to express the similarity of molecular graphs [77]: shortest-path [104], graphlet [105], tree- and cyclic-pattern [106, 107], and optimal assignment kernels [108].

Acknowledgements

Thanks to Jana Doppa and Aryan Deshwal for stimulating Cory's interest in graph kernels. We acknowledge support from the National Science Foundation, award #1920945.

References

[1] Fernando P Carvalho. Agriculture, pesticides, food security and food safety. *Environmental Science & Policy*, 9(7-8):685–692, 2006.

[2] E-C Oerke. Crop losses to pests. *The Journal of Agricultural Science*, 144(1):31–43, 2006.

[3] József Popp, Károly Pető, and János Nagy. Pesticide productivity and food security. a review. *Agronomy for Sustainable Development*, 33(1):243–255, 2013.

[4] Jerry Cooper and Hans Dobson. The benefits of pesticides to mankind and the environment. *Crop Protection*, 26(9):1337–1348, 2007.

[5] Polyxeni Nicolopoulou-Stamati, Sotirios Maipas, Chrysanthi Kotampasi, Panagiotis Stamatis, and Luc Hens. Chemical pesticides and human health: the urgent need for a new concept in agriculture. *Frontiers in Public Health*, 4:148, 2016.

[6] Clevo Wilson and Clem Tisdell. Why farmers continue to use pesticides despite environmental, health and sustainability costs. *Ecological Economics*, 39(3):449–462, 2001.

[7] Isra Mahmood, Sameen Ruqia Imadi, Kanwal Shazadi, Alvina Gul, and Khalid Rehman Hakeem. Effects of pesticides on environment. In *Plant, Soil and Microbes*, pages 253–269. Springer, 2016.

[8] David Tilman, Joseph Fargione, Brian Wolff, Carla D'antonio, Andrew Dobson, Robert Howarth, David Schindler, William H Schlesinger, Daniel Simberloff, and Deborah Swackhamer. Forecasting agriculturally driven global environmental change. *science*, 292(5515):281–284, 2001.

[9] Sebastian Stehle and Ralf Schulz. Agricultural insecticides threaten surface waters at the global scale. *Proceedings of the National Academy of Sciences*, 112(18):5750–5755, 2015.

[10] Johnson Stanley, Gnanadhas Preetha, and Stanley. *Pesticide toxicity to non-target organisms*. Springer, 2016.

[11] Lennard W Pisa, Vanessa Amaral-Rogers, Luc P Belzunces, Jean-Marc Bonmatin, Craig A Downs, Dave Goulson, David P Kreutzweiser, Christian Krupke, Matthias Liess, Melanie McField, et al. Effects of neonicotinoids and fipronil on non-target invertebrates. *Environmental Science and Pollution Research*, 22(1):68–102, 2015.

[12] Benjamin P Oldroyd. What's killing american honey bees? *PLoS Biology*, 5(6):e168, 2007.

[13] Ben A Woodcock, JM Bullock, RF Shore, MS Heard, MG Pereira, J Redhead, L Ridding, H Dean, D Sleep, P Henrys, et al. Country-specific effects of neonicotinoid pesticides on honey bees and wild bees. *Science*, 356(6345):1393–1395, 2017.

[14] Adam J Vanbergen and the Insect Pollinators Initiative. Threats to an ecosystem service: pressures on pollinators. *Frontiers in Ecology and the Environment*, 11(5):251–259, 2013.

[15] Richard J Gill, Oscar Ramos-Rodriguez, and Nigel E Raine. Combined pesticide exposure severely affects individual-and colony-level traits in bees. *Nature*, 491(7422):105–108, 2012.

[16] Dave Goulson. An overview of the environmental risks posed by neonicotinoid insecticides. *Journal of Applied Ecology*, 50(4):977–987, 2013.

[17] Dennis VanEngelsdorp, Jerry Hayes Jr, Robyn M Underwood, and Jeffery Pettis. A survey of honey bee colony losses in the us, fall 2007 to spring 2008. *PloS One*, 3(12):e4071, 2008.

[18] Insu Koh, Eric V Lonsdorf, Neal M Williams, Claire Brittain, Rufus Isaacs, Jason Gibbs, and Taylor H Ricketts. Modeling the status, trends, and impacts of wild bee abundance in the united states. *Proceedings of the National Academy of Sciences*, 113(1):140–145, 2016.

[19] Sydney A Cameron, Jeffrey D Lozier, James P Strange, Jonathan B Koch, Nils Cordes, Leellen F Solter, and Terry L Griswold. Patterns of widespread decline in north american bumble bees. *Proceedings of the National Academy of Sciences*, 108(2):662–667, 2011.

[20] Marla Spivak, Eric Mader, Mace Vaughan, and Ned H Euliss Jr. The plight of the bees, 2011.

[21] Dave Goulson, Elizabeth Nicholls, Cristina Botías, and Ellen L Rotheray. Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. *Science*, 347(6229):1255957, 2015.

[22] Nicola Gallai, Jean-Michel Salles, Josef Settele, and Bernard E Vaissière. Economic valuation of the vulnerability of world agriculture confronted with pollinator decline. *Ecological Economics*, 68(3):810–821, 2009.

[23] P. H. Raven, G. B. Johnson, J. B. Losos, K. A. Mason, and S. R. Singer. *Biology*. McGraw-Hill higher education, 2008.

[24] Rachael Winfree, Neal M Williams, Hannah Gaines, John S Ascher, and Claire Kremen. Wild bee pollinators provide the majority of crop visitation across land-use gradients in new jersey and pennsylvania, usa. *Journal of Applied Ecology*, 45(3):793–802, 2008.

[25] Alexandra-Maria Klein, Bernard E Vaissiere, James H Cane, Ingolf Steffan-Dewenter, Saul A Cunningham, Claire Kremen, and Teja Tscharntke. Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*, 274(1608):303–313, 2007.

[26] MD Levin. Value of bee pollination to us agriculture. *American Entomologist*, 29(4):50–51, 1983.

[27] Keng-Lou James Hung, Jennifer M Kingston, Matthias Albrecht, David A Holway, and Joshua R Kohn. The worldwide importance of honey bees as pollinators in natural habitats. *Proceedings of the Royal Society B: Biological Sciences*, 285(1870):20172140, 2018.

[28] Thomas C Sparks and Ralf Nauen. Irac: Mode of action classification and insecticide resistance management. *Pesticide Biochemistry and Physiology*, 121:122–128, 2015.

[29] Micheal DK Owen and Ian A Zelaya. Herbicide-resistant crops and weed resistance to herbicides. *Pest Management Science: formerly Pesticide Science*, 61(3):301–311, 2005.

[30] John A. Lucas, Nichola J. Hawkins, and Bart A. Fraaije. Chapter two - the evolution of fungicide resistance. In SIMA SARIASLANI and GEOFFREY MICHAEL GADD, editors, *Advances in Applied Microbiology*, volume 90 of *Advances in Applied Microbiology*, pages 29–92. Academic Press, 2015.

[31] Noriharu Umetsu and Yuichi Shirai. Development of novel pesticides in the 21st century. *Journal of Pesticide Science*, 45(2):54–74, 2020.

[32] Sergio Filipe Sousa, Pedro Alexandrino Fernandes, and Maria Joao Ramos. Protein–ligand docking: current status and future challenges. *Proteins: Structure, Function, and Bioinformatics*, 65(1):15–26, 2006.

[33] Fangli Gang, Xiaoting Li, Chaofu Yang, Lijuan Han, Hao Qian, Shaopeng Wei, Wenjun Wu, and Jiwen Zhang. Synthesis and insecticidal activity evaluation of virtually screened phenylsulfonamides. *Journal of Agricultural and Food Chemistry*, 68(42):11665–11671, 2020.

[34] Toshiyuki Harada, Yoshiaki Nakagawa, Takehiko Ogura, Yutaka Yamada, Takehiro Ohe, and Hisashi Miyagawa. Virtual screening for ligands of the insect molting hormone receptor. *Journal of Chemical Information and Modeling*, 51(2):296–305, 2011.

[35] Xueping Hu, Bin Yin, Kaat Cappelle, Luc Swevers, Guy Smagghe, Xinling Yang, and Li Zhang. Identification of novel agonists and antagonists of the ecdysone receptor by virtual screening. *Journal of Molecular Graphics and Modelling*, 81:77–85, 2018.

[36] Ting-Ting Yao, Shao-Wei Fang, Zhong-Shan Li, Dou-Xin Xiao, Jing-Li Cheng, Hua-Zhou Ying, Yong-Jun Du, Jin-Hao Zhao, and Xiao-Wu Dong. Discovery of novel succinate dehydrogenase inhibitors by the integration of in silico library design and pharmacophore mapping. *Journal of Agricultural and Food Chemistry*, 65(15):3204–3211, 2017.

[37] Shinri Horoiwa, Taiyo Yokoi, Satoru Masumoto, Saki Minami, Chiharu Ishizuka, Hidetoshi Kishikawa, Shunsuke Ozaki, Shigeki Kitsuda, Yoshiaki Nakagawa, and Hisashi Miyagawa. Structure-based virtual screening for insect ecdysone receptor ligands using mm/pbsa. *Bioorganic & Medicinal Chemistry*, 27(6):1065–1075, 2019.

[38] Galen J Correy, Daniel Zaidman, Alon Harmelin, Silvia Carvalho, Peter D Mabbitt, Viviane Calaora, Peter J James, Andrew C Kotze, Colin J Jackson, and Nir London. Overcoming insecticide resistance through computational inhibitor design. *Proceedings of the National Academy of Sciences*, 116(42):21012–21021, 2019.

[39] Kerem Teralı. An evaluation of neonicotinoids' potential to inhibit human cholinesterases: Protein–ligand docking and interaction profiling studies. *Journal of Molecular Graphics and Modelling*, 84:54–63, 2018.

[40] Axel Decourtye, Mickaël Henry, and Nicolas Desneux. Overhaul pesticide testing on bees. *Nature*, 497(7448):188–188, 2013.

[41] US EPA. Pollinator risk assessment guidance. `https://www.epa.gov/pollinator-protection/pollinator-risk-assessment-guidance`. Accessed: 2022-02-20.

[42] Glenn J. Myatt, Ernst Ahlberg, Yumi Akahori, David Allen, Alexander Amberg, Lennart T. Anger, Aynur Aptula, Scott Auerbach, Lisa Beilke, Phillip Bellion, Romualdo Benigni, Joel Bercu, Ewan D. Booth, Dave Bower, Alessandro Brigo, Natalie Burden, Zoryana Cammerer, Mark T.D. Cronin, Kevin P. Cross, Laura Custer, Magdalena Dettwiler, Krista Dobo, Kevin A. Ford, Marie C. Fortin, Samantha E. Gad-McDonald, Nichola Gellatly, Véronique Gervais, Kyle P. Glover, Susanne Glowienke, Jacky Van Gompel, Steve Gutsell, Barry Hardy, James S. Harvey, Jedd Hillegass, Masamitsu Honma, Jui-Hua Hsieh, Chia-Wen Hsu, Kathy Hughes, Candice Johnson, Robert Jolly, David Jones, Ray Kemper, Michelle O. Kenyon, Marlene T. Kim, Naomi L. Kruhlak, Sunil A. Kulkarni, Klaus Kümmerer, Penny Leavitt, Bernhard Majer, Scott Masten, Scott Miller, Janet Moser, Moiz Mumtaz, Wolfgang Muster, Louise Neilson, Tudor I. Oprea, Grace Patlewicz, Alexandre Paulino, Elena Lo Piparo, Mark Powley, Donald P. Quigley, M. Vijayaraj Reddy, Andrea-Nicole Richarz, Patricia Ruiz, Benoit Schilter, Rositsa Serafimova, Wendy Simpson, Lidiya Stavitskaya, Reinhard Stidl, Diana Suarez-Rodriguez, David T. Szabo, Andrew Teasdale, Alejandra Trejo-Martin, Jean-Pierre Valentin, Anna Vuorinen, Brian A. Wall, Pete Watts, Angela T. White, Joerg Wichard, Kristine L. Witt, Adam Woolley, David Woolley, Craig Zwickl, and Catrin Hasselgren. In silico toxicology protocols. *Regulatory Toxicology and Pharmacology*, 96:1–17, 2018.

[43] Arwa B Raies and Vladimir B Bajic. In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 6(2):147–172, 2016.

[44] Flor A Quintero, Suhani J Patel, Felipe Munoz, and M Sam Mannan. Review of existing qsar/qspr models developed for properties used in hazardous chemicals classification system. *Industrial & Engineering Chemistry Research*, 51(49):16101–16115, 2012.

[45] Takao Iwasa, Naoki Motoyama, John T Ambrose, and R Michael Roe. Mechanism for the differential toxicity of neonicotinoid insecticides in the honey bee, apis mellifera. *Crop Protection*, 23(5):371–378, 2004.

[46] Daniela Laurino, Marco Porporato, Augusto Patetta, Aulo Manino, et al. Toxicity of neonicotinoid insecticides to honey bees: laboratory tests. *Bulletin of Insectology*, 64(1):107–13, 2011.

[47] Francisco Sanchez-Bayo and Koichi Goka. Pesticide residues and bees–a risk assessment. *PLOS One*, 9(4):e94482, 2014.

[48] EL Atkins and D Kellum. Comparative morphogenic and toxicity studies on the effect of pesticides on honeybee brood. *Journal of Apicultural Research*, 25(4):242–255, 1986.

[49] Helen M Thompson. Assessing the exposure and toxicity of pesticides to bumblebees (bombus sp.). *Apidologie*, 32(4):305–321, 2001.

[50] Yee-Tung Hu, Tsai-Chin Wu, En-Cheng Yang, Pei-Chi Wu, Po-Tse Lin, and Yueh-Lung Wu. Regulation of genes related to immune signaling and detoxification in apis mellifera by an inhibitor of histone deacetylation. *Scientific Reports*, 7(1):1–14, 2017.

[51] Krystyna Pohorecka, Teresa Szczęsna, Monika Witek, Artur Miszczak, and Piotr Sikorski. The exposure of honey bees to pesticide residues in the hive environment with regard to winter colony losses. *Journal of Apicultural Science*, 61(1):105–125, 2017.

[52] Christopher A Mullin, Maryann Frazier, James L Frazier, Sara Ashcraft, Roger Simonds, Dennis VanEngelsdorp, and Jeffery S Pettis. High levels of miticides and agrochemicals in north american apiaries: implications for honey bee health. *PLOS one*, 5(3):e9754, 2010.

[53] A Decourtye, James Devillers, Evelyne Genecque, K Le Menach, H Budzinski, S Cluzeau, and MH Pham-Delegue. Comparative sublethal toxicity of nine pesticides on olfactory learning performances of the honeybee apis mellifera. *Archives of environmental contamination and toxicology*, 48(2):242–250, 2005.

[54] Thaís S Bovi, Rodrigo Zaluski, and Ricardo O Orsi. Toxicity and motor changes in africanized honey bees (apis mellifera l.) exposed to fipronil and imidacloprid. *Anais da Academia Brasileira de Ciências*, 90:239–245, 2018.

[55] Mohamed El Badawy, Hoda M Nasr, and Entsar I Rabea. Toxicity and biochemical changes in the honey bee apis mellifera exposed to four insecticides under laboratory conditions. *Apidologie*, 46(2):177–193, 2015.

[56] Nadejda Tsvetkov, Olivier Samson-Robert, Keshna Sood, HS Patel, DA Malena, PH Gajiwala, Philip Maciukiewicz, Valerie Fournier, and Amro Zayed. Chronic exposure to neonicotinoids reduces honey bee health near corn crops. *Science*, 356(6345):1395–1397, 2017.

[57] José T. Moreira-Filho, Rodolpho C. Braga, Jade Milhomem Lemos, Vinicius M. Alves, Joyce V.V.B. Borba, Wesley S. Costa, Nicole Kleinstreuer, Eugene N. Muratov, Carolina Horta Andrade, and Bruno J. Neves. Beetoxai: An artificial intelligence-based web app to assess acute toxicity of chemicals to honey bees. *Artificial Intelligence in the Life Sciences*, 1:100013, 2021.

[58] Fan Wang, Jing-Fang Yang, Meng-Yao Wang, Chen-Yang Jia, Xing-Xing Shi, Ge-Fei Hao, and Guang-Fu Yang. Graph attention convolutional neural network model for chemical poisoning of honey bees prediction. *Science Bulletin*, 65(14):1184–1191, 2020.

[59] Edoardo Carnesecchi, Andrey A. Toropov, Alla P. Toropova, Nynke Kramer, Claus Svendsen, Jean Lou Dorne, and Emilio Benfenati. Predicting acute contact toxicity of organic binary mixtures in honey bees (a. mellifera) through innovative qsar models. *Science of The Total Environment*, 704:135302, 2020.

[60] Mabrouk Hamadache, Othmane Benkortbi, Salah Hanini, and Abdeltif Amrane. QSAR modeling in ecotoxicological risk assessment: application to the prediction of acute contact toxicity of pesticides on bees (apis mellifera l.). *Environmental Science and Pollution Research*, 25(1):896–907, October 2017.

[61] F. Como, E. Carnesecchi, S. Volani, J.L. Dorne, J. Richardson, A. Bassan, M. Pavan, and E. Benfenati. Predicting acute contact toxicity of pesticides in honeybees (apis mellifera) through a k-nearest neighbor model. *Chemosphere*, 166:438–444, 2017.

[62] Xuan Xu, Piaopiao Zhao, Zhiyuan Wang, Xiaoxiao Zhang, Zengrui Wu, Weihua Li, Yun Tang, and Guixia Liu. In silico prediction of chemical acute contact toxicity on honey bees via machine learning methods. *Toxicology in Vitro*, 72:105089, 2021.

[63] Xiao Li, Yuan Zhang, Hongna Chen, Huanhuan Li, and Yong Zhao. Insights into the molecular basis of the acute contact toxicity of diverse organic chemicals in the honey bee. *Journal of Chemical Information and Modeling*, 57(12):2948–2957, 2017.

[64] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[65] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):1–22, 2020.

[66] Daniel S Wigh, Jonathan M Goodman, and Alexei A Lapkin. A review of molecular representation in the age of machine learning. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, page e1603, 2022.

[67] Lagnajit Pattanaik and Connor W Coley. Molecular representation: going long on fingerprints. *Chem*, 6(6):1204–1207, 2020.

[68] Lagnajit Pattanaik, Octavian-Eugen Ganea, Ian Coley, Klavs F Jensen, William H Green, and Connor W Coley. Message passing networks for molecules with tetrahedral chirality. *arXiv preprint arXiv:2012.00094*, 2020.

[69] Tu Le, V Chandana Epa, Frank R Burden, and David A Winkler. Quantitative structure–property relationship modeling of diverse materials properties. *Chemical Reviews*, 112(5):2889–2919, 2012.

[70] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[71] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[72] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-aided Molecular Design*, 30(8):595–608, 2016.

[73] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[74] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[75] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[76] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

[77] Matthias Rupp and Gisbert Schneider. Graph kernels for molecular similarity. *Molecular Informatics*, 29(4):266–273, 2010.

[78] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph kernels: State-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*, 2020.

[79] Jan Ramon and Thomas Gärtner. Expressivity versus efficiency of graph kernels. In *Proceedings of the first international workshop on mining graphs, trees and sequences*, pages 65–74, 2003.

[80] Liva Ralaivola, Sanjay J. Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005. Neural Networks and Kernel Methods for Structured Domains.

[81] Giannis Nikolentzos, Giannis Siglidis, and Michalis Vazirgiannis. Graph kernels: A survey. *Journal of Artificial Intelligence Research*, 72:943–1027, 2021.

[82] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[83] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[84] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[85] Yan Xiang, Yu-Hang Tang, Guang Lin, and Huai Sun. A comparative study of marginalized graph kernel and message-passing neural network. *Journal of Chemical Information and Modeling*, 61(11):5414–5424, 2021.

[86] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.

[87] Nils M Kriege, Marion Neumann, Christopher Morris, Kristian Kersting, and Petra Mutzel. A unifying view of explicit and implicit feature maps of graph kernels. *Data Mining and Knowledge Discovery*, 33(6):1505–1547, 2019.

[88] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 321–328, 2003.

[89] Douglas J Klein, José Luis Palacios, Milan Randić, and Nenad Trinajstić. Random walks and chemical graph theory. *Journal of Chemical Information and Computer Sciences*, 44(5):1521–1525, 2004.

[90] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

[91] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[92] Asa Ben-Hur and Jason Weston. A users guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

[93] Marina Meilă. Data centering in feature space. In *International Workshop on Artificial Intelligence and Statistics*, pages 209–216. PMLR, 2003.

[94] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[96] MP Preeja and KP Soman. Walk-based graph kernel for drug discovery: A review. *nternational Journal of Computer Applications*, 94:1–7, 2014.

[97] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[98] Cunliang Geng, Yong Jung, Nicolas Renaud, Vasant Honavar, Alexandre M J J Bonvin, and Li C Xue. iScore: a novel graph kernel-based function for scoring proteinprotein docking models. *Bioinformatics*, 36(1):112–121, 06 2019.

[99] Hiroshi Ohno and Yusuke Mukae. Machine learning approach for prediction and search: application to methane storage in a metal–organic framework. *The Journal of Physical Chemistry C*, 120(42):23963–23968, 2016.

[100] Yu-Hang Tang and Wibe A de Jong. Prediction of atomization energy using graph kernel and active learning. *The Journal of Chemical Physics*, 150(4):044107, 2019.

[101] Grégoire Ferré, Terry Haut, and Kipton Barros. Learning molecular energies using localized graph kernels. *The Journal of Chemical Physics*, 146(11):114107, 2017.

[102] Yan Xiang, Yu-Hang Tang, Hongyi Liu, Guang Lin, and Huai Sun. Predicting single-substance phase diagrams: A kernel approach on graph representations of molecules. *The Journal of Physical Chemistry A*, 125(20):4488–4497, 2021.

[103] Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 70, 2004.

[104] Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[105] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR, 2009.

[106] Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167, 2004.

[107] Pierre Mahé and Jean-Philippe Vert. Graph kernels based on tree patterns for molecules. *Machine learning*, 75(1):3–35, 2009.

[108] Holger Fröhlich, Jörg K Wegner, Florian Sieker, and Andreas Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd international conference on Machine learning*, pages 225–232, 2005.