MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **90** (2023) 75–102
ISSN: 0340–6253
doi: 10.46793/match.90-1.075G

Comparison of Atom Maps

Marcos E. González Laffitte[1,2], Nora Beier[1], Nico Domschke[1], Peter F. Stadler[1–7,*]

[1]Bioinformatics Group, Department of Computer Science & Interdisciplinary Center for Bioinformatics & Leipzig University, D-04107 Leipzig, Germany

[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Leipzig University, D-04103 Leipzig, Germany

[3]German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig & Leipzig Research Center for Civilization Diseases, Leipzig University, D-04103 Leipzig, Germany

[4]Max Planck Institute for Mathematics in the Sciences, D-04109 Leipzig, Germany

[5]Department of Theoretical Chemistry of the University of Vienna, A-1090 Vienna, Austria

[6]Facultad de Ciencias, Universidad National de Colombia, Sede Bogotá, Colombia [7]Santa Fe Institute, Santa Fe NM 87501, USA

{marcos,nora,dnico,studla}@bioinf.uni-leipzig.de

(Received 27 January, 2023)

Abstract

The computation of reliable, chemically correct atom maps from educt/product pairs has turned out to be a difficult problem in cheminformatics because the chemically correct solution is not necessarily an optimal solution for combinatorial formulations such as maximum common subgraph problems. As a consequence, competing models have been devised and compared in extensive benchmarking studies. Due to isomorphisms among products and educts it is not immediately obvious, however, when two atom maps for

*Corresponding author.

a given educt/product pairs are the same. We formalize here the equivalence of atom maps and show that equivalence of atom maps is in turn equivalent to the isomorphism of labeled auxiliary graphs. In particular, we demonstrate that Fujita's Imaginary Transition State can be used for this purpose. Numerical experiments show that practical feasibility. Generalizations to the equivalence of subgraph matches, double pushout graph transformation rules, and mechanisms of multi-step reactions are discussed briefly.

1 Introduction

Chemical reactions, by definition, constitute the rearrangement of chemical bonds while preserving the atoms involved. In practice, chemical reactions are typically represented as transformations of a multiset of reactant molecules into a corresponding multiset of product molecules [7, 48]. The mechanism of a reactions, i.e., the bonds broken and newly formed, and, equivalently, the correspondence of the atoms in reactants and products, is not apparent from such data. In many practical applications, for instance the analysis of isotope labeling experiments [24], the inference of reaction rules [2], and in metabolic engineering [27, 47], however, it is key to track atoms across a reaction. To this end, structural formulas of reactants and products, respectively, are viewed as graphs G and H with vertices labeled by atom types and edges labeled by bond types. The atom map of the reaction transforming G to H then is a bijection $\alpha : V(G) \to V(H)$ that preserves atom types. By specifying the corresponding atoms in reactants and product, α implies the bonds that are broken and formed, see Fig. 1.

Determining the atom map of a chemical reaction given only the structural formulas of the reactants G and products H is a very challenging problem. Purely combinatorial methods phrase the task as Maximum Common Subgraph (MCS) or Maximum Common Edge Subgraph subgraph (MCES) problems [11, 13]. However, there is no guarantee that the optimal solutions correctly represent the mechanism of the reaction in question. Chemical realism is added for instance in MetaCyc by assigning weights to bonds that encode their propensity to break, resulting in a weighted MCES problem [27]. A constraint programming approach [32] was proposed to enforce constraints such as cyclic transition states. Most

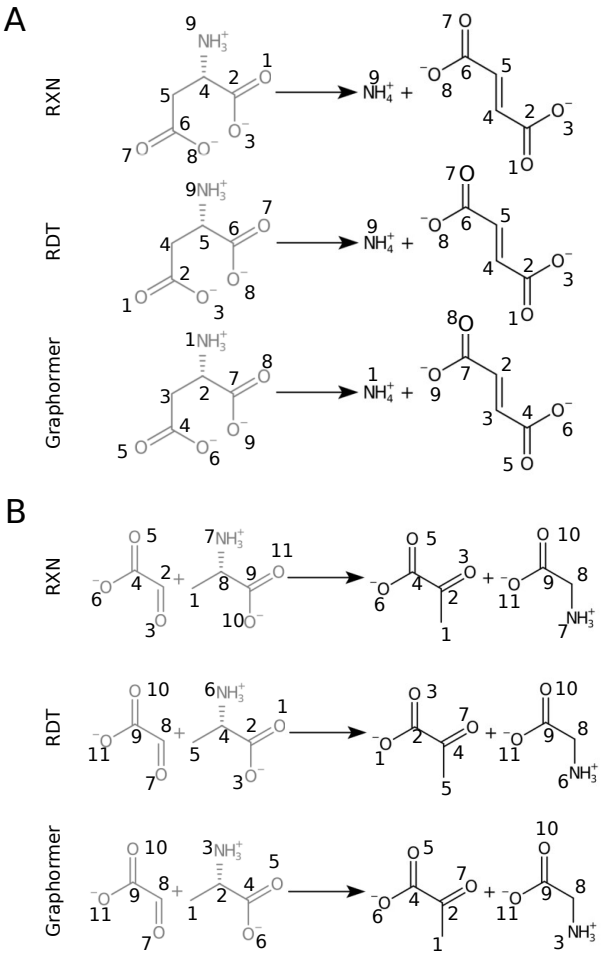

Figure 1. (A) Equivalent but different atom maps for the reaction of the L-aspartate to fumarate + NH_4^+. (B) Non-equivalent atom maps for the reaction of L-alanine + glyoxylate to pyruvate + glycine. Atom maps were computed with `RDT`, `RXNMapper` [46], and `GraphormerMapper`.

recently, several machine-learning tools have become available as an alternative approach to predicting atom maps [25, 29, 46].

The benchmarking of different atom mapping approaches requires a fair comparison of the predicted atom maps with each other and with

a (usually manually curated) ground truth set [28, 39, 41]. The various atom mapping tools convert their input into some internal representation, establish the bijection and return the map α in a mapper-specific format, for instance as annotated reactions SMILES. As a consequence, given a reactant graph G and a product graph H, an atom mapping tool does not usually return an atom map $\alpha : V(G) \to V(H)$ but rather an atom map $\beta : V(G') \to V(H')$ where $G' \simeq G$ and $H' \simeq H$, i.e., G' and H' are isomorphic but not necessarily identical to the input graphs G and H [39]. It is not trivial, therefore, to determine whether or not $\beta : V(G') \to V(H')$ describes the same atom map as $\alpha : V(G) \to V(H)$, see Figure 1.

In a recent benchmarking study [28], for instance, an atom map is *"considered to be correct"* if the Condensed Graphs of the Reactions (CGRs) [22] of the test and reference mapping *"coincide totally"*, where CGRs were compared using the `CGRtools` library [37]. The description suggests that the authors of [28] evaluated isomorphism of CGRs as a means of testing the equivalence of atom maps. To the best of our knowledge, no further justification for this method has been published. Here, we provide a rigorous proof for the correctness of the procedure.

Essentially the same question arises also in graph transformation models of chemical reactions [4, 7, 45], since the application of a rule requires that a pattern G is found as in target graph H, see Figure 2. Formally, one is interested in a map $\mu : V(G) \to V(H)$ such that for each edge xy of G, the image $\mu(x)\mu(y)$ is an edge in H and both vertex and edge labels are preserved.

This contribution is organized as follows. We formalize the equivalence of maps in Section 2, describe a general construction of an auxiliary graph, and prove our main result. Thm. 1 shows that equivalence of maps can be decided by checking isomorphism of the auxiliary graphs. In the following section we focus on the comparison of atom maps and show that for bijective maps a smaller auxiliary graph, which in essence is the Imaginary Transition State [14] or the Condensed Graph of a Reaction [22] is sufficient. We then briefly consider generalizations and open problems.

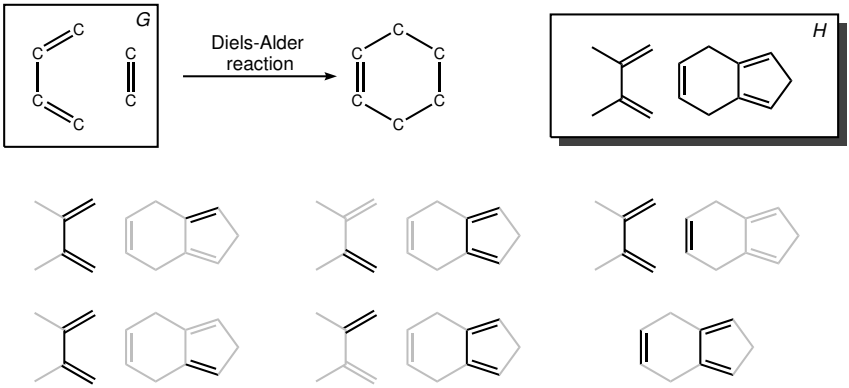

Figure 2. Equivalent and non-equivalent matches of a subgraph, shown on the example of a Diels-Alder reaction. In the reaction a dienophile reacts with a diene to form new carbon-carbon bonds. Due to the symmetries of the reactants (i.e., the connected components of H) some of the matches μ of the precondition, i.e., the graph, G are equivalent and thus generate the same reaction product. We indicate the location of the subgraph isomorphic to G in H by thick, black lines. Note that any such match corresponds to four matching morphisms μ, depending on the orientations in which the two connected components of G are matched. Since any diene contains two dienophile moieties, alternative mappings can arise. The entries in the first two columns each represent equivalent matches.

2 Equivalence of maps

2.1 Notation and basic definitions

We consider here simple, loop-free graphs, with both vertex and edge labels. We write $V(G)$, $E(G)$, $a_G : V(G) \to L_V$ and $b_G : E(G) \to L_E$ for the vertex set, edge set, vertex labeling function, and edge labeling functions, respectively. The edge between the (distinct) vertices $x, y \in V(G)$ will be denoted by $xy \in E(G)$. Since the graphs encode the educts and products of chemical reactions, which usually consist of more than one molecule, they will not be connected in general.

We will use the notation $f \circ g$ for the composite map $f \circ g : x \mapsto f(g(x))$. Two labeled graphs G and G' are isomorphic, in symbols $G \simeq G'$, if there

is a bijective map (called isomorphism) $\varphi : V(G) \to V(G')$ that preserves edges, non-edges, vertex labels, and edge labels. That is (i) $xy \in E(G)$ if and only if $\varphi(x)\varphi(y) \in V(H)$, (ii) $a_H \circ \varphi = a_G$, and (iii) $b_H(\varphi(x)\varphi(y)) = b_G(xy)$ for all $xy \in E(G)$. We write $\mathrm{ISO}(G, G')$ for the set of all such graph isomorphisms. We note that the isomorphisms are obtained as composition of automorphisms of G or G' and an arbitrary isomorphism $\tilde{\varphi} : V(G) \to V(G')$. Writing $\mathrm{Aut}(G)$ for the set of automorphisms of G, we have $\mathrm{ISO}(G, G') = \{\tilde{\varphi} \circ \vartheta | \vartheta \in \mathrm{Aut}(G)\} = \{\vartheta' \circ \tilde{\varphi} | \vartheta' \in \mathrm{Aut}(G')\}$.

We may also consider a more general set of bijective maps $\mathrm{GI}(G, G')$ that relate two graphs that are considered equivalent in a sense that is more general than isomorphism. For example, we may define $\mathrm{GI}(G, G')$ to consider graph isomorphisms that ignore edge labels (e.g. bond types) or that identify certain aspects of vertex labels (e.g. by ignoring charges). In [41], for example, bond-types were ignored to accommodate resonance structure with delocalized electrons. Since $\phi \in \mathrm{GI}(G, G')$ is bijective, there is a uniquely defined map $\phi^{-1} : V(G') \to V(G)$ such that $\varphi^{-1} \circ \varphi = \iota_G$, where $\iota_G : V(G) \to V(G)$, $x \mapsto \iota(x) = x$ is the identity on G. We will assume that these sets of structure-preserving maps have the following properties:

(i) If $\phi \in \mathrm{GI}(G, G')$ and $\psi \in \mathrm{GI}(G', G'')$ then $\psi \circ \phi \in \mathrm{GI}(G, G'')$

(ii) If $\phi \in \mathrm{GI}(G, G')$ then $\phi^{-1} \in \mathrm{GI}(G', G)$

Clearly, (graph) isomorphisms $\mathrm{ISO}(G, G')$ satisfy these requirements. We therefore call the maps in $\mathrm{GI}(G, G')$ the *generalized isomorphisms* from G to G'.

One easily checks that $\phi^{-1} \circ \phi = \iota_G \in \mathrm{GI}(G, G)$, and $\mathrm{GI}(G, G)$ is closed under composition of maps and forming inverse maps. Thus $(\mathrm{GI}(G, G), \circ)$ is a group acting on $V(G)$. Its orbits determine an associated equivalence relation on $V(G)$. In applications to molecular graphs, this relation identifies *chemically equivalent* atoms.

Conceptually, two maps $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are equivalent if there is a "renumbering" of G and H that makes α and β the "same". More formally, the renumbering on G and H corresponds to

two generalized isomorphisms $\varphi : V(G) \to V(G')$ and $\psi : V(H) \to V(H')$ such that, for all $x \in V(G)$, we have $\psi(\alpha(x)) = \beta(\varphi(x))$.

Definition 1. Two maps $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are *equivalent*, in symbols $\alpha \equiv \beta$ if there is $\varphi \in \mathrm{GI}(G, G')$ and $\psi \in \mathrm{GI}(H, H')$ such that $\psi \circ \alpha = \beta \circ \varphi$.

Note as well that $\alpha \equiv \beta$ implies $\mathrm{GI}(G, G') \neq \emptyset$ and $\mathrm{GI}(H, H') \neq \emptyset$.

Lemma 1. *Equivalence of maps, $\equiv$, is an equivalence relation.*

Proof. The relation $\equiv$ is reflexive since by assumption $\iota_G \in \mathrm{GI}(G, G)$ and $\iota_H \in \mathrm{GI}(H, H)$ and and $\alpha = \iota_H \circ \alpha = \alpha \circ \iota_G = \alpha$. From $\alpha \equiv \beta$, i.e., $\psi \circ \alpha = \beta \circ \varphi$ we obtain $\alpha \circ \varphi^{-1} = \psi^{-1} \circ \beta$. Clearly $\varphi^{-1} : V(G') \to V(G)$ and $\psi^{-1} : V(H') \to V(H)$ are graph isomorphisms, and thus $\beta \equiv \alpha$. Finally, consider $\alpha \equiv \beta$ and $\beta \equiv \gamma$ with $\gamma : V(G'') \to V(H'')$, $G' \simeq G''$, and $H' \simeq H''$. By definition, there are isomorphisms $\zeta : V(G') \to V(G'')$ and $\xi : V(H') \to V(H'')$ such that $\xi \circ \beta = \gamma \circ \zeta$. Using $\psi \circ \alpha \circ \varphi^{-1} = \beta$ we obtain $(\xi \circ \psi) \circ \alpha = \gamma \circ (\zeta \circ \varphi)$, where $\xi \circ \psi : V(H) \to V(H'')$ and $\zeta \circ \varphi : V(G) \to V(G'')$ are generalized isomorphisms. Thus $\alpha \equiv \gamma$. ■

If α and β are bijective, we can rewrite $\psi \circ \alpha = \beta \circ \varphi$ as $\psi = \beta \circ \varphi \circ \alpha^{-1}$. Def. 1 then can be rephrased in the following form:

Lemma 2. *Let $\alpha : V(G) \to V(H)$ and $\beta : V(G) \to V(H)$ be bijective maps. Then $\alpha \equiv \beta$ if and only if there is $\varphi \in \mathrm{GI}(G, G')$ such that $\beta \circ \varphi \circ \alpha^{-1} \in \mathrm{GI}(H, H')$.*

Thus $\alpha \equiv \beta$ can be checked by enumerating all generalized isomorphisms $\varphi \in \mathrm{GI}(G, G')$ and then checking whether $\beta \circ \varphi \circ \alpha^{-1} \in \mathrm{GI}(H, H')$ for at least one $\varphi \in \mathrm{GI}(G, G')$. This, however, appears unpleasantly complicated. Furthermore, the idea is applicable only to bijective maps.

2.2 Reduction to graph isomorphism

In order to address the computational problem of determining the equivalence of two maps, we translate the question into a graph-theoretical problem. More precisely, we encode $\alpha : V(G) \to V(H)$ in a simple, suitable labeled graph $\Gamma(G, H, \alpha)$, see Fig. 3, as follows:

Definition 2. Let (G, H, α) be a triple comprising two vertex and edge labeled graphs G and H linked by an arbitrary map $\alpha : V(G) \to V(H)$. Then the auxiliary graph $\Gamma(G, H, \alpha)$ has vertex set $V(G) \cup V(H)$, edge set $E(G) \cup E(H) \cup E(\alpha)$ where $E(\alpha) \coloneqq \{x\alpha(x) | x \in V(G)\}$, vertex labels $(a_G(x), 1)$ for $x \in V(G)$ and $(a_H(x), 2)$ for $x \in V(H)$ and edge labels $b(e) = b_G(e)$ if $e \in E(G)$, $b(e) = b_H(e)$ if $e \in E(H)$, and $b(e) = *$ with $*$ distinct from the vertex labels in G and H, if $e \in E(\alpha)$.

To see that this graph indeed unambiguously encode the map $\alpha : V(G) \to V(H)$, consider the class $\mathfrak{G}$ of labeled graphs with following properties:

(i) $V = V_1 \cup V_2$, where V_1 and V_2 are distinguished by labels of the form $a(x) = (a'(x), 1)$ if $x \in V_1$ and $a(x) = (a'(x), 2)$ if $x \in V_1$.

(ii) Every $x \in V_1$ is adjacent to exactly one neighbor $y_x \in V_2$.

(iii) $b(xy) = *$ for all edges $xy \in E$ with $x \in V_1$ and $y \in V_2$.

Lemma 3. *Let $\Gamma \in \mathfrak{G}$. Then there are unique labeled graphs G and H and a unique map $\alpha : V(G) \to V(H)$ such that $\Gamma = \Gamma(G, H, \alpha)$.*

Proof. Starting from $\Gamma \in \mathfrak{G}$, we construct G, H, and α explicitly. Condition (i) implies that the vertex labels completely determine the bipartition $V(\Gamma) = V_1 \cup V_2$ such that $a_\Gamma(x) = (a'(x), 1)$ for $x \in V_1$ and $a_\Gamma(x) = (a'(x), 2)$ for $x \in V_2$. We obtain the induced subgraphs $G = \Gamma[V_1]$ and $H = \Gamma[V_2]$ with vertex labels $a_G = a'(x)$ for all $x \in V_1$ and $a_H = a'(x)$ for all $x \in V_2$ and edge labels $b_G(e) = b_\Gamma(e)$ for $e \in E(G)$ and $b_H(e) = b_\Gamma(e)$ for $e \in E(H)$. Property (ii) stipulates that for every $x \in V_1 = V(G)$ there is $y_x \in V_2 = V(H)$. Thus $\alpha(x) \coloneqq y_x$ for all $x \in V_1$ defines a map $\alpha : V(G) \to V(H)$. This defines the desired triple (G, H, α). One easily checks that $\Gamma(G, H, \alpha)$ is the graph with $V_1 = V(G)$, $V_2 = V(G)$, vertex labels $a(x) = (a'(x), 1)$ for $x \in V_1$ and $a(x) = (a'(x), 2)$ for $x \in V_2$, edge set $E = E(G) \cup E(H) \cup \{xy | x \in V(G), y \in V(H)\}$, and edge labels $b(e) = b_\Gamma(e)$ for $e \in E(G) \cup E(H)$ and $b(xy) = b_\Gamma(xy) = *$ for $x \in V_1$ and $y \in V_1$. Thus $\Gamma(G, H, \alpha) = \Gamma$. ∎

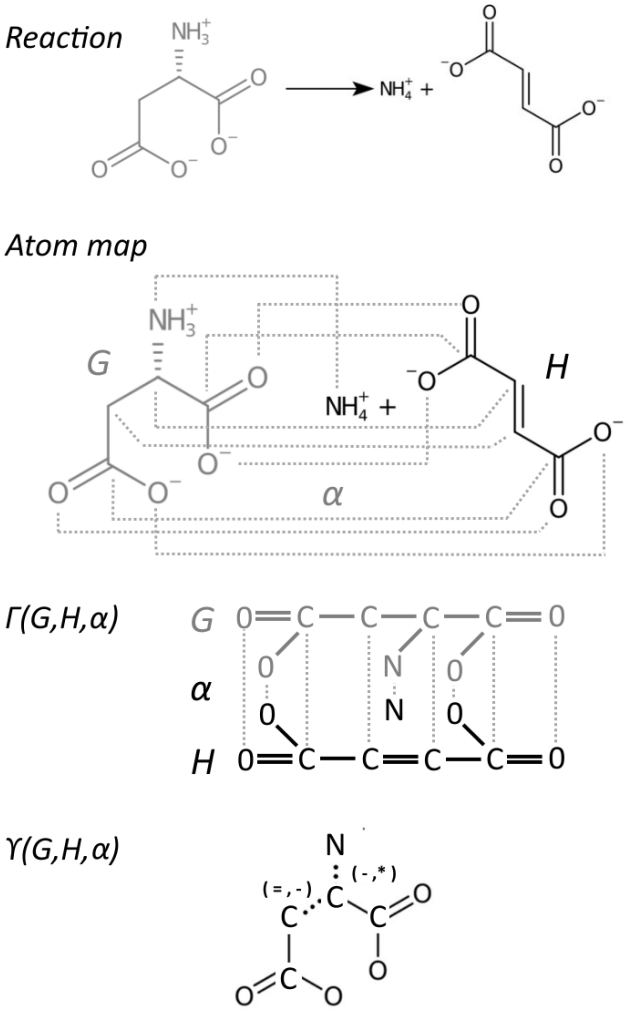

Figure 3. Atom maps and their equivalent graphs. Top: reaction of the L-aspartate to fumarate and $NH_4{}^+$, as in Fig. 1. Below, the chemically correct atom map α is shown, using dotted lines to connect corresponding atom. For simplicity of the presentation, the ammonium hydrogens are also suppresses in the auxiliary graphs below. $\Gamma(G, H, \alpha)$ is shown with vertices and edges of G in gray, vertices and edges of H in black, and the matching $E(\alpha)$ as dotted lines. In $\Upsilon(G, H, \alpha)$, vertices and edges with the same label in both G and H are drawn as full lines, while "reaction edges" are shown as dotted lines annotated by the corresponding label pairs: $(=, -)$ denotes a change from a double bond to a single bond and $(-, *)$ denotes the breaking of the single bond.

The relationship between (G, H, α) and $\Gamma(G, H, \alpha)$ preserves the identity of the vertices is i.e., we are not considering isomorphisms in this step. Thus $\Gamma(G, H, \alpha)$ is a faithful representation of the graphs G and H and the map connecting their vertex sets. Moreover, Γ contains G and H as induced subgraphs determined by the vertex sets with second coordinate of the vertex label being '1' and '2', respectively (shown in gray and black in Fig. 3). Furthermore, $E(\alpha)$ is a matching in $\Gamma(G, H, \alpha)$ whenever α is injective. Clearly, the matching $E(\alpha)$ is perfect if and only if α is bijective.

Theorem 1. *Suppose $G \simeq G'$ and $H \simeq H'$. Then two maps $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are equivalent if and only if their labeled auxiliary graphs are isomorphic, i.e., if and only if $\Gamma(G, H, \alpha) \simeq \Gamma(G', H', \beta)$.*

Proof. First suppose $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are equivalent, i.e., there are isomorphisms $\varphi \in \mathrm{ISO}(G, G')$ and $\psi \in \mathrm{ISO}(H, H')$ such that $\psi \circ \alpha = \beta \circ \varphi$. We claim that the union $\zeta := \varphi \cup \psi$ defined by $\zeta(x) = \varphi(x)$ if $x \in V(G)$ and $\zeta(x) = \psi(x)$ if $x \in V(H)$ is an isomorphism $\zeta \in \mathrm{ISO}(\Gamma)$. Consider two vertices $p, q \in V(\Gamma)$ and their images $p' := \zeta(p)$ and $q' := \zeta(q)$. If $p, q \in V(G)$, then $p' = \varphi(p)$ and $q' = \varphi(q)$. Since $\varphi \in \mathrm{ISO}(G, G')$, we have $pq \in E(\Gamma)$ if and only if $p'q' \in E(\Gamma')$ and edge and vertex labels are preserved. An analogous result holds for $p, q \in V(H)$. Now let $p \in V(G)$ and $q \in V(H)$. Then $pq \in E(\Gamma)$ if and only if $q = \alpha(p)$. Consider the images of $\zeta(p)$ and $\zeta(q)$ in Γ'. We have by construction $p' := \zeta(p) = \varphi(p)$ and $q' := \zeta(q) = \psi(q)$. Suppose $q = \alpha(p)$. Then $q' = \psi(\alpha(p)) = \beta(\varphi(p)) = \beta(p')$, and thus $p'q' \in E(\Gamma')$. Note that there is no other edge $p'q''$ with $p' \in V(G')$ and $q'' \in V(H')$. Thus for given p, we have $\zeta(p)\zeta(q) \in E(\Gamma')$ if and only if $pq \in E(\Gamma)$. We have already see that ζ preserved all vertex labels and the edge lables on $E(G)$ and $E(H)$; hence it remains to demonstrate that ζ also preserves the remaining edge labels: Since $p' \in V(G')$ and $q' \in V(H')$, we have $b_{\Gamma'}(p'q') = * = b_\Gamma(pq)$. In summary, $\zeta : V(\Gamma) \to V(\Gamma')$ is a labeled graph isomorphism.

For the converse, suppose $\zeta : V(\Gamma) \to V(\Gamma')$ is an isomorphism, i.e., $\zeta \in \mathrm{ISO}(\Gamma, \Gamma')$. First note that, by construction of the vertex labels, $V(G)$ and $V(G')$ are identifiable by labels of the form $(\cdot, 1)$, while $V(H)$ and $V(H')$

have vertex labels of the form $(\cdot, 2)$. Since ζ by assumption preserves vertex labels, it maps $V(G)$ to $V(G')$ and $V(H)$ to $V(H')$. Thus the restrictions ζ_G and ζ_H defined by $\zeta_G(x) = \zeta(x)$ for $x \in V(G)$ and $\zeta_H(x) = \zeta(x)$ for $x \in V(H)$ are isomorphisms of labeled graphs: $\zeta_G \in \mathrm{ISO}(G, G')$ and $\zeta_H \in \mathrm{ISO}(H, H')$. Now consider an edge $e \in E(\Gamma)$ that connects $V(G)$ and $V(H)$. By definition of the auxiliary graphs, $b_\Gamma(e) = *$, i.e., $e \in E(\alpha)$. By assumption we have $e = xy$ where x is a vertex with $a(x) = (\cdot, 1)$ and thus $x \in V(G)$ and y is a vertex with $a(y) = (\cdot, 2)$ and thus $y \in V(G)$. Therefore we have $\alpha(x) = y$. Since ζ is an isomorphism, $x'y' \coloneqq \zeta(x)\zeta(y) \in E(\Gamma')$, again with label $*$ and thus $x'y' \in E(\beta)$. Since $\zeta(x) = \zeta_G(x) \in V(G')$ and $\zeta(y) = \zeta_H(y) \in V(H')$, we conclude that $y' = \beta(x')$. This can be rewritten as $\beta(\zeta(x)) = \zeta(y)$ and $\beta(\zeta_G(x)) = \zeta_H(y)$. Inserting $y = \alpha(x)$ finally yields $\beta(\zeta_G(x)) = \zeta_H(\alpha(x))$. By construction of Γ, there is an edge xy with $y \in V(H)$ for every $x \in V(G)$. Thus $\zeta_H \circ \alpha = \beta \circ \zeta_G$. Finally, $\zeta_G \in \mathrm{ISO}(G, G')$ and $\zeta_H \in \mathrm{ISO}(H, H')$ implies $\alpha \equiv \beta$. ■

Thm. 1 has useful computational implications. First we note that given G, H, and α, the auxiliary graph $\Gamma(G, H, \alpha)$ can be constructed in linear time, since it only requires the insertion of mapping edges $E(\alpha)$ and a relabeling of the vertices of $V(G)$ and $V(H)$ to ensure the distinction of pre-image and image. The graph isomorphism problem, which is the name sake for the class of GI-complete computational problems [33], is computationally equivalent to the problem of computing generators of the automorphism group of a graph [30]. This can be solved in quasi-polynomial time for general graphs [6]. The graphs appearing in chemistry as models of molecules, moreover, have bounded degrees. This remains true for the auxiliary graph $\Gamma(G, H, \alpha)$, whose degree is bounded by the maximal degrees of G and H plus 1. In this restricted case, graph isomorphism is solvable in polynomial time [31]. Furthermore, highly efficient practical implementations of graph isomorphism tests have become available [5,34], see [18] for a recent review. Thm. 1 therefore reduces the map equivalence problem to a labeled graph isomorphism problem that can be solved efficiently for most practical applications in chemistry.

3 Equivalent atom maps and ITSs

3.1 Equivalence of bijective maps

The fact that α and β are bijective makes it possible to simplify $\Gamma(G, H, \alpha)$ further. To this end we first associate the information that $e \in E(\Gamma)$ is contained in $E(G)$ or $E(H)$ explicitly with the edge by replacing the edge label $b(xy)$ by $(b(xy), i)$ if both $a(x) = (a'(x), i)$ and $a(y) = (a'(y), i)$ for $i = 1, 2$, i.e., $xy \in E(G)$ and $xy \in E(H)$, respectively. Denote the auxiliary graph with these augmented edge labels by $\hat{\Gamma}(G, H, \alpha)$. Clearly $\zeta \in \mathrm{ISO}(\Gamma, \Gamma')$ if and only if $\zeta \in \mathrm{ISO}(\hat{\Gamma}, \hat{\Gamma}')$. To see this, it suffices to note that $\zeta \in \mathrm{ISO}(\Gamma, \Gamma')$ satisfies $\zeta(x) \in V(G')$ for $x \in V(G)$ and $\zeta(x) \in V(H')$ for $x \in V(H)$ and thus $b_\Gamma(xy) = b_\Gamma(\zeta(x)\zeta(y))$ implies $b_{\hat{\Gamma}}(xy) = b_{\hat{\Gamma}}(\zeta(x)\zeta(y))$, i.e., the augmented labeling does not break any isomorphisms.

As a second step we construct another graph $\hat{\Upsilon}(G, H, \alpha)$ from $\hat{\Gamma}(G, H, \alpha)$ by contracting the matching $E(\alpha)$, see Fig. 3. That is, we set $V(\hat{\Upsilon}) = V(G)$ and identify the vertices x and $\alpha(x)$ and the edges xy and $\alpha(x)\alpha(y)$. To keep track of the information associated with the contracted vertices, we associate with x the vertex label $a_{\hat{\Upsilon}}(x) \coloneqq (a_{\hat{\Gamma}}(x), a_{\hat{\Gamma}}(\alpha(x)))$ and with each edge in $xy \in E(\hat{\Upsilon})$ the following labels:

(i) $b_{\hat{\Upsilon}}(xy) \coloneqq (b_{\hat{\Gamma}}(xy), b_{\hat{\Gamma}}(\alpha(x)\alpha(y)))$ if $xy \in E(\Gamma)$ and $\alpha(x)\alpha(y) \in E(\Gamma)$.

(ii) $b_{\hat{\Upsilon}}(xy) \coloneqq (b_{\hat{\Gamma}}(xy), *)$ if $xy \in E(\Gamma)$ and $\alpha(x)\alpha(y) \notin E(\Gamma)$.

(iii) $b_{\hat{\Upsilon}}(xy) \coloneqq (*, b_{\hat{\Gamma}}(\alpha(x)\alpha(y)))$ if $xy \notin E(\Gamma)$ and $\alpha(x)\alpha(y) \in E(\Gamma)$.

Note these three cases are mutually exclusive and cover all edges in $\hat{\Upsilon}$.

Given such a graph, we can recover $\hat{\Gamma}$ unambiguously by splitting each vertex and assigning the vertex label $(a(x), 1)$ to one copy and $(a(\alpha(x)), 2)$ to the other. Then edges with labels $b_{\hat{\Gamma}}(xy)$ are inserted between vertices x and y with vertex labels $(a_G(x), 1)$ and $(a_G(y), 1)$ and edges $b_{\hat{\Gamma}}(\alpha(x)\alpha(y))$ are inserted between vertices $\alpha(x)$ and $\alpha(y)$ with vertex labels $(a_H(\alpha(x)), 2)$ and $(a_H(\alpha(y)), 2)$. No edges are inserted for labels of the form $(*, \cdot)$ between the first type of vertices. Correspondingly edges with labels of the

form $(\cdot, *)$ are ignored with regard to the second type of vertices. Clearly, this construction recovers $\hat{\Gamma}$ from $\hat{\Upsilon}$.

Lemma 4. *Suppose $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are bijective. Then $\hat{\Gamma}(G, H, \alpha) \simeq \hat{\Gamma}(G', H', \beta)$ if and only if $\hat{\Upsilon}(G, H, \alpha) \simeq \hat{\Upsilon}(G', H', \beta)$.*

Proof. We verify that every isomorphism in $\mathrm{ISO}(\hat{\Gamma}(G, H, \alpha), \hat{\Gamma}(G', H', \beta))$ can be translated into an isomorphism in $\mathrm{ISO}(\hat{\Upsilon}(G, H, \alpha), \hat{\Upsilon}(G', H', \beta))$ and *vice versa*.

Claim 1. If $\zeta \in \mathrm{ISO}(\hat{\Gamma}(G, H, \alpha), \hat{\Gamma}(G', H', \beta))$, then the restriction ζ_G of ζ to $V(G)$ satisfies $\zeta_G \in \mathrm{ISO}(\hat{\Upsilon}(G, H, \alpha), \hat{\Upsilon}(G', H', \beta))$.

Proof of the Claim. The vertex labels in $\hat{\Gamma} \coloneqq \hat{\Gamma}(G, H, \alpha)$ and $\hat{\Gamma}' \coloneqq \hat{\Gamma}(G', H', \alpha)$ unambiguously identify $V(G)$, $V(H)$, $V(G')$ and $V(H')$, respectively and ensure that the restriction ζ_G of ζ to $V(G)$ is a map $\zeta_G : V(G) \to V(G')$. By Thm. 1 furthermore, α and β are equivalent since ζ is an isomorphism, thus we have $\zeta(\alpha(x)) = \beta(\zeta(x))$ and hence $\zeta_G(\alpha(x)) = \beta(\zeta_G(x))$ for all $x \in V(G)$ since we identified x and $\alpha(x)$.

First we show that ζ_G preserves vertex labels. Using $\zeta_G(x) = \zeta(x)$, $a_{G'}(\zeta(x)) = a_G(x)$, and $a_{H'}(\zeta(\alpha(x))) = a_{H'}(\beta(\zeta(x))) = a_H(\alpha(x))$ together with the definition of the vertex labels in $\hat{\Upsilon}' \coloneqq \hat{\Upsilon}(G', H', \beta)$ yield $a_{\hat{\Upsilon}'}(\zeta_G(x)) = ((a_{G'}(\zeta_G(x)), 1), (a_{H'}(\beta(\zeta_G(x))), 2)) =$
$((a_G(x), 1), (a_H(\alpha(x)), 2)) = a_{\hat{\Upsilon}}(x)$ for all $x \in V(G)$.

Suppose $x, y \in V(G)$, $x \neq y$ and $xy \notin E(\hat{\Upsilon})$. Then by construction $xy \notin E(\hat{\Gamma})$ and $\alpha(x)\alpha(y) \notin E(\hat{\Gamma})$ and thus $\zeta(x)\zeta(y) \notin E(\hat{\Gamma}')$ and $\alpha(\zeta(x))\alpha(\zeta(y)) = \zeta(\beta(x))\zeta(\beta(y) \notin E(\hat{\Gamma}')$, which, by definition of $\hat{\Upsilon}'$ implies $\zeta_G(x)\zeta_G(y) \notin E(\hat{\Upsilon}')$. On the other hand, if $xy \in E(\hat{\Upsilon})$ then $xy \in E(\hat{\Gamma})$ or $\alpha(x)\alpha(y) \in E(\hat{\Gamma})$ and hence $\zeta(x)\zeta(y) \in E(\hat{\Gamma}')$ or $\alpha(\zeta(x))\alpha(\zeta(y)) = \zeta(\beta(x))\zeta(\beta(y)) \in E(\hat{\Gamma}')$, which in turn implies $\zeta_G(x)\zeta_G(y) \in E(\hat{\Upsilon}')$. For the edge labels we observe $b_{\hat{\Gamma}'}(\zeta_G(x)\zeta_G(y)) = b_{\hat{\Gamma}}(xy)$ whenever $xy \in E(\hat{\Gamma}')$ and $b_{\hat{\Gamma}'}(\beta(\zeta_G(x))\beta(\zeta_G(y))) = b_{\hat{\Gamma}'}(\zeta_G(\alpha(x))\zeta_G(\alpha(y))) = b_{\hat{\Gamma}}(\alpha(x)\alpha(y))$ whenever $\alpha(x)\alpha(y) \in E(\hat{\Gamma})$. Since $xy \in E(\hat{\Gamma})$ iff $\zeta_G(x)\zeta_G(y) \in E(\hat{\Gamma}')$ and $\alpha(x)\alpha(y) \in E(\hat{\Gamma})$ iff $\beta(\zeta_G(x))\beta(\zeta_G(y)) \in E(\hat{\Gamma}')$, we see that ζ_G also preserves the edge labels of the form $(b_{\hat{\Gamma}}(xy), *)$ and $(*, b_{\hat{\Gamma}}(\alpha(x)\alpha(y)))$. We conclude that $\zeta_G \in \mathrm{ISO}(\hat{\Upsilon}, \hat{\Upsilon}')$. ◁

Claim 2. Let $\xi \in \mathrm{ISO}(\hat{\Upsilon}(G,H,\alpha),\hat{\Upsilon}(G',H',\beta))$ and let $\theta : V(\hat{\Gamma}) \to V(\hat{\Gamma}')$ be the map defined by $\theta(x) = \xi(x)$ for $x \in V(G)$ and $\theta(y) \coloneqq \beta(\xi(\alpha^{-1}(y)))$ for $y \in V(H)$. Then $\theta \in \mathrm{ISO}(\hat{\Gamma}(G,H,\alpha),\hat{\Gamma}(G',H',\beta))$.

Proof of the Claim. First we note that $a_{\hat{\Upsilon}}(x) = ((a_G(x),1),(a_H(\alpha(x)),2))$ and thus $a_{\hat{\Gamma}}(x) = a_G(x)$ for $x \in V(G)$ and $a_{\hat{\Gamma}}(\alpha(x)) = a_H(\alpha(x))$ for $x \in V(G)$. For $x \in V(G)$ we have $a_{\hat{\Gamma}'}(\theta(x)) = a_{\hat{\Gamma}'}(\xi(x)) = a_{\hat{\Gamma}}(x)$. Since for every $z \in V(H)$ there is $x \in V(G)$ with $z = \alpha(x)$. Thus for $z \in V(H)$ there is a unique $x = \alpha^{-1}(z)$ with vertex label $((a_G(x),1),(a_H(z),2))$. It is mapped to $\xi(x) = \xi(\alpha^{-1}(z))$ with label $((a_{G'}(\xi(x)),1),(a_{H'}(\beta(\xi(x)),2))$. Since ξ is an isomorphism, it preserves vertex labels and thus $a_{\hat{\Gamma}}(z) = a_{\hat{\Gamma}'}(\beta(\xi(x))) = a_{\hat{\Gamma}'}(\beta(\xi(\alpha^{-1}(z)))) = a_{\hat{\Gamma}'}(\theta(z))$.

Now consider two distinct vertices $x,y \in V(\hat{\Gamma})$. If $x \in V(G)$ and $y \in V(H)$ then $xy \in E(\hat{\Gamma})$ if and only if $y = \alpha(x)$. In this case we have $\theta(x)\theta(y) = \xi(x)\beta(\xi(\alpha^{-1}(y))) = \xi(x)\beta(\xi(x))$, and thus $\theta(x)\theta(y) \in E(\hat{\Gamma}')$. For given $x \in V(G)$, this is the only edge $xu \in E(\hat{\Gamma}')$ with $u \in V(H)$. Next suppose $x,y \in V(G)$. Then $xy \in E(\hat{\Gamma})$ if $xy \in E(\hat{\Upsilon})$ and the edge label has first coordinate $b_G(xy)$, i.e., $b_{\hat{\Gamma}}(xy) = b_G(xy)$. Then $\theta(x)\theta(y) = \xi(x)\xi(y) \in E(\hat{\Upsilon}')$ with first coordinate of the label $b_{\hat{\Gamma}'}(\xi(x)\xi(y)) = b_{G'}(\xi(x)\xi(y)) = b_G(xy)$. Hence we have $b_{\hat{\Gamma}'}(\theta(x)\theta(y)) = b_{\hat{\Gamma}}(xy)$. Now consider $\bar{x} = \alpha(x)$ and $\bar{y} = \alpha(y)$. Then $\bar{x}\bar{y} \in E(\hat{\Gamma})$ if $xy \in E(\hat{\Upsilon})$ and the second coordinate of the edge label is $b_H(\alpha(x)\alpha(y))$. Then $\beta(\xi(x))\beta(\xi(y)) \in E(\hat{\Upsilon}')$ with second coordinate of the edge label $b_{\hat{\Gamma}'}(\beta(\xi(x))\beta(\xi(y))) = b_H(\alpha(x)\alpha(y))$ since ξ is an isomorphism. Since α is a bijection, there are unique $\bar{x},\bar{y} \in V(H)$ such that $x = \alpha^{-1}(\bar{x})$ and $y = \alpha^{-1}(y)$. Thus we have $b_{\hat{\Gamma}'}(\theta(\bar{x})\theta(\bar{y})) = b_{\hat{\Gamma}'}(\beta(\xi(\alpha^{-1}(\bar{x})))\beta(\xi(\alpha^{-1}(\bar{y})))) = b_H(\bar{x}\bar{y})$ for all $\bar{x}\bar{y} \in E(H)$. It follows that $\theta \in \mathrm{ISO}(\hat{\Gamma},\hat{\Gamma}')$. ◁

By Claim 1, $\hat{\Gamma}(G,H,\alpha) \simeq \hat{\Gamma}(G',H',\beta)$ yields $\Upsilon(G,H,\alpha) \simeq \Upsilon(G',H',\beta)$. Conversely, $\hat{\Upsilon}(G,H,\alpha) \simeq \hat{\Upsilon}(G',H',\beta)$ implies $\hat{\Gamma}(G,H,\alpha) \simeq \hat{\Gamma}(G',H',\beta)$ because of Claim 2, and thus the assertion of the Lemma. ■

We summarize the discussion of this section as follows:

Theorem 2. *Suppose $G \simeq G'$ and $H \simeq H'$ and $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are bijective. Then $\alpha \equiv \beta$ if and only if the labeled auxiliary graphs $\hat{\Upsilon}(G,H,\alpha)$ and $\hat{\Upsilon}(G',H',\beta)$ are isomorphic.*

Proof. By Thm. 1 we have $\alpha \equiv \beta$ iff $\Gamma(G,H,\alpha) \simeq \Gamma(G',H',\beta)$, which in turn is equivalent with $\hat{\Gamma}(G,H,\alpha) \simeq \hat{\Gamma}(G',H',\beta)$, and by Lemma 4, this condition in turn holds if and only if $\hat{\Upsilon}(G,H,\alpha) \simeq \hat{\Upsilon}(G',H',\beta)$. ■

3.2 Equivalence of atom maps

From a practical point of view, the most important application of map equivalence is the comparison of atom maps.

Definition 3. An atom map $\alpha : V(G) \to V(H)$ is a bijective map that preserves vertex labels, i.e., $a_H(\alpha(x)) = a_G(x)$ for all $x \in V(G)$.

Two atom maps α and β are "chemically the same", if the graphs G and H can be renumbered in such a way that α and β coincide, i.e., if $\alpha \equiv \beta$ for a pair of isomorphic reactant and product graphs $G \simeq G'$ and $H \simeq H'$, respectively.

Now suppose $\alpha : V(G) \to V(H)$ is not only bijective but also preserves vertex labels, α is an atom map. Then we have
$a_{\hat{\Upsilon}}(x) = ((a_G(x),1),(a_H(\alpha(x)),2)) = ((a_G(x),1),(a_G(x),2))$,
i.e., the vertex label is already completely determined by $a_G(x)$. We may therefore simplify the vertex labels and obtain Υ from $\hat{\Upsilon}$ by setting $a_\Upsilon(x) = \tilde{a}$ whenever $a_{\hat{\Upsilon}}(x) = ((\tilde{a},1),(\tilde{a},2))$. It is obvious that the relabeling does not affect isomorphisms.

Tracing back the stepwise construction of $\Upsilon(G,H,\alpha)$ and noting that $b_{\hat{\Gamma}}(xy) = b_G(xy)$ for $xy \in E(G)$ and $b_{\hat{\Gamma}}(xy) = b_H(\alpha(x)\alpha(y))$ for $\alpha(x)\alpha(y) \in E(H)$ yields the following observation:

Lemma 5. *Let $\alpha : V(G) \to V(H)$ be an an atom map. Then $\Upsilon(G,H,\alpha)$ is the graph with vertex set $V(G)$, vertex labels $a_\Upsilon(x) = a_G(x)$ for all $x \in V(G)$, an edges $xy \in E(\Upsilon)$ if and only if $xy \in G$ or $\alpha(x)\alpha(y) \in E(H)$, and edge labels*

$$b(xy) = \begin{cases} (b_G(xy), b_H(\alpha(x)\alpha(y))) & \text{if } xy \in E(G) \text{ and } \alpha(x)\alpha(y) \in E(H) \\ (b_G(xy), *) & \text{if } xy \in E(G) \text{ and } \alpha(x)\alpha(y) \notin E(H) \\ (*, b_H(\alpha(x)\alpha(y))) & \text{if } xy \notin E(G) \text{ and } \alpha(x)\alpha(y) \in E(H) \end{cases}$$

The graph specified in Lemma 5 is (a version of) the *Imaginary Transition State* (ITS) and the *Condensed Graph of a Reaction* (CGR) [22]. The ITS was introduced by Shinsaku Fujita already in 1986 as "an extended kind of chemical structure" That encodes reactants, products, and atom mappings within a single, undirected, connected graph [14]. The CGR was proposed a decade ago [22] as a condensed representation of chemical reaction with machine learning applications in mind. Both graphs are essentially the same, apart from details of the convention used to annotate the changing bonds. An essentially equivalent formulation in terms of the adjacency matrices of G and H was proposed already in 1973 [12]. We refer to $\Upsilon(G, H, \alpha)$ as ITS in the context of chemical reactions.

In chemical terms, the ITS $\Upsilon(G, H, \alpha)$ provides a complete description of a chemical reaction. From Thm. 2 and Lemma 5 we obtain

Corollary 1. *Let $G \simeq G'$, $H \simeq H'$ and suppose $\alpha : V(G) \to V(H)$ and $\beta : V(G') \to V(H')$ are atom maps. Then $\alpha \equiv \beta$ if and only if the ITSs $\Upsilon(G, H, \alpha)$ and $\Upsilon(G', H', \beta)$ are isomorphic.*

This result provides a rigorous justification for the computational approach in [28] to compare atom maps by testing for isomorphisms of their ITSs (or CGRs). Nevertheless we should add that the authors of [28] and [36] introduced and made use of a manually curated set of reactions for their benchmarking purposes, which they refer to as *Golden* data set and which consists of both stoichiometrically balanced and unbalanced reactions. In our contribution we also made use of this data set for our own computational analysis, but we only employed the former type of reactions, since it should be noted that the use of the latter type is in conflict with the formal requirement now established by Cor. 1, regarding α and β as bijective maps in order to properly use the ITSs (or CGRs) for the comparison of atom maps. This is of relevance for the next section.

Recall that all *molecular graphs*, i.e., graphs representing molecules, have bounded degree. This is trivially also true for the educt and product graph of every chemical reaction. As an immediate consequence, the ITS of any chemical reaction also has bounded degree. Since graph isomorphism can be tested in polynomial time for bounded degree [31] we have:

Corollary 2. *The equivalence of two atom maps for any given chemical reactions can be decided in polynomial time.*

4 Computational analysis

The mathematical results developed in the last sections, specifically Lemma 2, Thm. 1 and Cor. 1, can be readily converted into algorithms. With them we built the software `EEquAAM` (**E**valuation of the **Equi**valence of **A**tom-to-**A**tom **M**aps), a computational toolkit for the automatic comparison of atom maps implemented in Python. In order to assess the practical utility of the auxiliary graphs $\Gamma(G, H, \alpha)$ and $\Upsilon(G, H, \alpha)$, respectively, we make both a direct comparison of atom maps (called **ISO**-$\equiv$ in the following), and isomorphism tests on both types of auxiliary graphs (referred to as **AUX**-Γ and **ITS**-Υ below). To benchmark the three methods, we use both known chemical reactions and a set of artificial reactions. Since atom maps are bijective by definition, we consider only maps that represent stoichiometrically balanced reactions as input, i.e., G and H have the same number of vertices for each label. In the remainder of this section we describe the implementation and benchmarking of `EEquAAM`. The software and accompanying material is available at `github` [16].

4.1 Methodology and implementation

The three approaches are implemented in the following manner:

ISO-$\equiv$ Using Lemma 2, one can test for $\alpha \equiv \beta$ by first generating all isomorphisms $\mathrm{ISO}(G, G')$ and $\mathrm{ISO}(H, H')$, computing $\psi = \beta \circ \varphi \circ \alpha^{-1}$ and checking whether $\psi \in \mathrm{ISO}(H, H')$. We note that one can generate $\mathrm{ISO}(G, G')$ by means of a single fixed isomorphism $\theta \in \mathrm{ISO}(G, G')$ and all the automorphisms $\eta \in \mathrm{Aut}(G)$ as $\varphi = \theta \circ \eta$. Most atom mapping tools, however, operate by assigning unique integer labels to the vertices of G and H such that the atom maps α and β are later expressed as the identity map over these integers. Tools, and thus atom maps, differ only in the numbering of the vertices. Thus $\mathrm{ISO}(G, G')$ and $\mathrm{ISO}(H, H')$ are sets of permutations living on the same numbering of vertices. Since α and β

are the identity we have $\beta \circ \varphi \circ \alpha^{-1} = \varphi$ and the condition in Lemma 2 simplifies: the two atom maps are equivalent if there is a renumbering φ that is simultaneously an isomorphism for $G \simeq G'$ and $H \simeq H'$, i.e.,

$$\alpha \equiv \beta \quad \Longleftrightarrow \quad \mathrm{ISO}(G, G') \cap \mathrm{ISO}(H, H') \neq \emptyset\,,$$

AUX-Γ The graphs $\Gamma(G, H, \alpha)$ and $\Gamma(G', H', \beta)$ are constructed as specified in Def. 2. We then check directly for the existence of a label-preserving isomorphism from $\Gamma(G, H, \alpha)$ to $\Gamma(G', H', \beta)$, which by Thm. 1 is equivalent to $\alpha \equiv \beta$.

ITS-Υ The ITS graphs $\Upsilon(G, H, \alpha)$ and $\Upsilon(G', H', \beta)$ and constructed stepwise starting from the edge-less graph with double-labeled vertices. Then the edge lists of G and H (or G' and H', respectively) are traversed consecutively, with labels set as described in Lemma 5.

These three methods where implemented by making use of various python packages. We used `Pysmiles` [26] to process the input reaction SMILES. The `NetworkX` [19] library was used to construct and manipulate graphs. Enumeration of the isomorphism $\mathrm{ISO}(G, G')$ and $\mathrm{ISO}(H, H')$ as well as isomorphism tests for $\Gamma(G, H, \alpha) \simeq \Gamma(G', H', \beta)$ and $\Upsilon(G, H, \alpha) \simeq \Upsilon(G', H', \beta)$ were performed with the implementation of the VF2 algorithm [10] available in `NetworkX`. This implementation in particular allows enforcing the conservation of both vertex and edge labels. Moreover, we used `Numpy` [21] to process running time statistics and `Matplotlib` [23] for the visualization of the results. The Python programs are available as part of the `EEquAAM` suite.

We used three representative atom-mapping tools to produce test data: (1) the Reaction Decoder Tool (RDT) [42–44] implementing four different variations of the maximal common (vertex) subgraph methodology (MCS). (2) `RXNmapper` (RXN) [46], described by its authors as a *chemically agnostic attention-guided reaction mapper*, built over a Transformer Neural Network architecture and operating on reaction SMILES. Lastly (3) `GraphormerMapper` [36] is also based on a Transformer Neural Network architecture. The `EEquAAM` suite also provides a wrapper for these tools to interact with the comparison methods mentioned above.

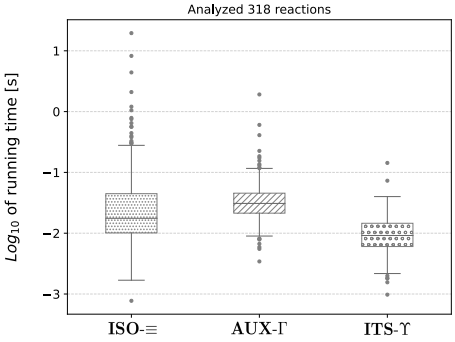

Figure 4. Distributions of the running times of the three methods for the comparison of atom maps for the 318 suitable reactions taken from the *Golden* data set.

To assess the performance of the different methods implemented in `EEquAAM` we first used a subset of the *Golden* data set, which was originally collected with the aim of benchmarking atom mapping tools [28,36]. The full set consists of 1851 annotated reaction SMILES for which manually curated atom maps are provided. Since the approaches taken here require bijective maps, we restricted ourselves to the subset of 318 stoichiometrically balanced reactions. Moreover, the reaction mappers produced complete atom maps for these 318 instances, providing us with 3 additional maps to be compared against each other and against the supplied manually curated one. Figure 4 summarizes the running times of the three methods implemented in `EEquAAM`. On this data set, **ISO**-≡, **AUX**-Γ and **ITS**-ϒ have comparable running times, although the smaller **ITS**-ϒ graphs as expected provide a small performance advantage.

A closer inspection of the test set shows, however, that the vast majority of the 318 instances has very small automorphism groups. In more than 60% of the instances $|\,\mathrm{Aut}(G)| \leq 4$ or $|\,\mathrm{Aut}(H)| \leq 4$, while automorphism groups with a size larger than 100 appear in only 6 reactions. In order to assess the effect of large automorphism groups we considered artificial reactions with highly symmetric molecules. In such cases the efforts necessary to enumerate $\mathrm{ISO}(G, G')$ and $\mathrm{ISO}(H, H')$ dominate the computational effort.

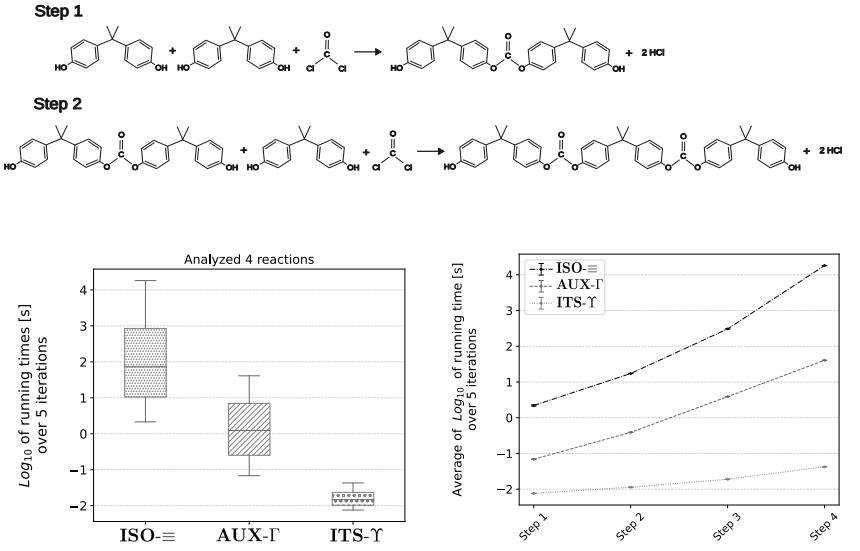

Figure 5. Polycondensation of bisphenol A (BPA) and phosgene. The first two iterations of the condensation reaction are shown on the top. Below, the distributions of the running times of the three methods over the four steps of the polycondensation reaction (left) and the average of running times as function of the steps in the reaction and thus the number symmetries in the molecules are shown.

The data clearly shows the practical advantage of using a single isomorphism test with the smaller ITS graphs over the alternatives. As a real-world example we use the first four reaction steps of the polycondensation reaction between bisphenol A (BPA) and phosgene [38], shown in Fig. 5. The size of the respective automorphism groups can be verified computationally. For *Step 1* of this reaction we have $|\operatorname{Aut}(G)| = 1024$ and $|\operatorname{Aut}(H)| = 256$, while for *Step 2* we have $|\operatorname{Aut}(G)| = 4096$ and $|\operatorname{Aut}(H)| = 2048$. Moreover, the number of symmetries produced by *Step 3* and *Step 4* of this reaction far surpasses the most symmetric examples in the *Golden* set, with $|\operatorname{Aut}(G)| = 32,768$ and $|\operatorname{Aut}(H)| = 16,384$ for the former, and $|\operatorname{Aut}(G)| = 262,144$ and $|\operatorname{Aut}(H)| = 131,072$ for the latter. A fifth step of this reaction yields millions of symmetries and thus the enumeration of automorphisms becomes prohibitive, as similar with

other larger polymers. It is worth noting that in this case the atom maps produced by the three tools not only are pairwise inequivalent most of the time, but rather they all are never equivalent. In this example we also find a large benefit for the ITS graphs over the larger auxiliary graphs (0.5s versus 40s). Although the graph isomorphism problem for bounded degree graphs can be solved in polynomial time [31] in theory, the VF2 approach does not provide such a guarantee.

5 Generalizations and open problems

In Section 2, we proved Thm. 1 for isomorphisms of labeled graphs. An analogous result still holds for generalized isomorphisms provided the following conditions are satisfied:

(i) The restriction of generalized isomorphisms $\zeta \in \mathrm{GI}(\Gamma(G,H,\alpha),\Gamma(G',H',\beta))$ to $V(G)$ and $V(H)$ induce generalized isomorphisms $\zeta_G : V(G) \to V(G')$ and $\zeta_H : V(H) \to V(H')$.

(ii) The generalized isomorphisms ζ allow vertex labels that distinguish two copies of a graph.

In this case, it suffices to ensure that the vertex labels on the auxiliary graph Γ are chosen such that, for all $\xi \in \mathrm{GI}(\Gamma,\Gamma)$, $\{a_\Gamma(\xi(x))|x \in V(G)\} \cap \{a_{\Gamma(y)}|y \in V(H)\} = \emptyset$, i.e., that the augmented labels for reactant and product side are disjoint. For convenience we also assume that label for the "mapping edges" $e \in E(\alpha)$, i.e., $b(e) = *$, is chosen such that $b_\Gamma(xy) = *$ implies $b_\Gamma(\xi(x)\xi(y)) = *$. Then it is easy to check that one can replace $\mathrm{ISO}(G,G')$, $\mathrm{ISO}(H,H')$, and $\mathrm{ISO}(\Gamma,\Gamma')$ by $\mathrm{GI}(G,G')$, $\mathrm{GI}(H,H')$, and $\mathrm{GI}(\Gamma,\Gamma')$. The practical use of this generalization is that, in particular, one can relax conditions on matching of labels. For instance, in [41] bond labels were ignored during the determination of chemically equivalent atoms. In the simplest case, equivalence classes of labels are defined. One can then relabel the graphs with fixed representatives and work with isomorphisms on the relabeled objects.

Another useful generalization considers partial maps, i.e., bijections between induced subgraphs G_s and H_s of G and H, respectively. Partial

maps of this type appear e.g. in various types of "graph alignments" [8]. The construction of the ITS is naturally extended by vertex label $(a_G(x), *)$ and $(*, a_H(y))$ for $x \in V(G) \setminus V(G_s)$ and $y \in V(H) \setminus V(H_s)$, respectively.

The approach is not limited to labeled graphs. It is also possible to consider set systems that have faithful graph representations. For instance undirected hypergraphs, as well as the directed hypergraphs with multiplicities that represent reactions networks [35] can be represented by their König graphs [49], in which both vertices and hyperedges become nodes (distinguished by distinct labels), while edges indicate the incidence of vertices and hyperedges. We suspect that very general set systems admit such faithful representations as labeled graphs. For instance, hierarchical clustering systems are equivalent to rooted trees. Another example are partially ordered sets and their Hasse diagrams.

Double Push-Out (DPO) graph grammars are particularly suitable as models of chemistry because the rules are reversible by construction and provide direct access to corresponding atom maps [2]. In this framework, each rule is a *span* $L \leftarrow K \rightarrow R$, where the "context" K is the subgraph common to the pattern L that needs to be present in the educts and in its replacement R that appears in the products. The application of the rule $L \leftarrow K \rightarrow R$ to a reactant graph G corresponds to the commutative diagram

$$\begin{array}{ccccc} L & \xleftarrow{\lambda} & K & \xrightarrow{\rho} & R \\ \downarrow{\scriptstyle \mu} & & \downarrow & & \downarrow \\ G & \longleftarrow & Q & \longrightarrow & H \end{array} \tag{1}$$

where $G \leftarrow Q \rightarrow H$ denotes the rewriting of G into H, with Q denoting the common subgraph of both G and H that is left unaffected by the transformation.

The notion of map equivalence $\equiv$ naturally generalizes to such diagrams containing multiple maps. Two diagrams are equivalent if there are isomorphisms between corresponding objects that commute with the maps between them. The construction of the auxiliary graph Γ thus generalizes immediately to diagrams, provided care is taken that the individual objects are assigned disjoint sets of vertex labels and thus remain iden-

tifiable. For example one can use the auxiliary graphs $\Gamma(G \leftarrow Q \rightarrow H)$ and $\Gamma(G' \leftarrow Q' \rightarrow H')$ to check whether two DPO graphs transformations are the equivalent. The same principle applies to the concatenations of reactions or rules [2]. Thus a joint graph representation of a sequence of atom maps can be used to check for the equivalence of entire pathways of reactions. We note in passing that, as an immediate consequence, isomorphy of the "overlap graphs" introduced in [1] is a necessary but not a sufficient condition for the equivalence of multi-step reactions.

The use of $\mathrm{GI}(G, G')$ in parts of this contribution instead of $\mathrm{ISO}(G, G')$ was motivated by notions of "chemical equivalence" that are not necessarily expressed by equivalence classes of edge and vertex labels. A good example are resonance structures [15]. For instance o-xylol (1,2-dimethylbenzene) may be represented with a single or a double bond between the two methyl groups. These two structures are chemical equivalent but distinct as labeled graphs. The generation of resonance structures was explored in [17], who give an algorithm that uses local graph transformations. We are not aware, however, of an algorithm that reliable recognizes the equivalence of resonance structures, although at least part of the issues are addressed by the PubChem chemical structure standardization protocol [20]. Similar issues may arises when stereochemical information is encoded as edge labels (wedge-and-dash notation [9]), or as a circular order of the neighbors at each vertex [3, 40].

***Acknowledgment*:** This work was supported in part by the Novo Nordisk Foundation (grant NNF21OC0066551 "MATOMICS") and the German Research Foundation (DFG) in the Priority Program SPP2363 (grant no STA 850/58-1 497135079). The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

References

[1] J. L. Andersen, R. Fagerberg, C. Flamm, W. Fontana, J. Kolčak, C. V. F. P. Laurent, D. Merkle, N. Nøjgaard, Representing catalytic mechanisms with rule composition, *J. Chem. Inf. Model.* **62** (2022) 5513–5524.

[2] J. L. Andersen, C. Flamm, D. Merkle, P. F. Stadler, Inferring chemical reaction patterns using graph grammar rule composition, *J. Syst. Chem.* **4** (2013) #4.

[3] J. L. Andersen, C. Flamm, D. Merkle, P. F. Stadler, Chemical graph transformation with stereo-information, in: J. de Lara, D. Plump (Eds.), *10th International Conference on Graph Transformation (ICGT 2017)*, Springer, Heidelberg, 2017, pp. 54–69.

[4] J. L. Andersen, C. Flamm, D. Merkle, P. F. Stadler, Rule composition in graph transformation models of chemical reactions, *MATCH Commun. Math. Comput. Chem.* **80** (2018) 661–704.

[5] J. L. Andersen, D. Merkle, A generic framework for engineering graph canonization algorithms, *ACM J. Exp. Alg.* **25** (2020) #1.2.

[6] L. Babai, Groups, graphs, algorithms: The graph isomorphism problem, in: B. Sirakov, P. Ney de Souza, M. V. Viana (Eds.), *Proceedings of the International Congress of Mathematicians (ICM 2018)*, World Scientific, Singapore, 2019, pp. 3303–3320.

[7] G. Benkö, C. Flamm, P. F. Stadler, A graph-based toy model of chemistry, *J. Chem. Inf. Comput. Sci.* **43** (2003) 1085–1093.

[8] J. Berg, M. Lässig, Local graph alignment and motif search in biological networks, *Proc. Nat. Acad. Sci. USA* **101** (2004) 14689–14694.

[9] J. Brecher, Graphical representation of stereochemical configuration (IUPAC Recommendations 2006), *Pure Appl. Chem.* **78** (2006) 1897–1970.

[10] L. Cordella, P. Foggia, C. Sansone, M. Vento, A (sub)graph isomorphism algorithm for matching large graphs, *IEEE Trans. Patt. Anal. Machine Intell.* **26** (2004) 1367–1372.

[11] E. Duesbury, J. Holliday, P. Willett, Comparison of maximum common subgraph isomorphism algorithms for the alignment of 2D chemical structures, *ChemMedChem* **13** (2018) 588–598.

[12] J. Dugundji, I. Ugi, An algebraic model of constitutional chemistry as a basis for chemical computer programs, *Topics Curr. Chem.* **39** (1973) 19–64.

[13] H. C. Ehrlich, M. Rarey, Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review, *WIREs* **1** (2011) 68–79.

[14] S. Fujita, Description of organic reactions based on imaginary transition structures. 1. introduction of new concepts, *J. Chem. Inf. Comput. Sci.* **26** (1986) 205–212.

[15] E. D. Glendening, C. R. Landis, F. Weinhold, Resonance theory reboot, *J. Am. Chem. Soc.* **141** (2019) 4156–4166.

[16] M. E. González Laffitte, N. Beier, N. Domschke, P. F. Stadler, EEquAAM: Github repository for the evaluation of the equivalence of atom-to-atom maps, `https://github.com/MarcosLaffitte/EEquAAM`, created on January 17th, 2023.

[17] A. Grinberg Dana, M. Liu, W. H. Green, Automated chemical resonance generation and structure filtration for kinetic modeling, *Int. J. Chem. Kinetics* **51** (2019) 760–776.

[18] M. Grohe, P. Schweitzer, Exploring the theoretical and practical aspects of the graph isomorphism problem, *Comm. ACM* **63** (2022) 128–134.

[19] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference*, Pasadena, 2008, pp. 11–15.

[20] V. D. Hähnke, S. Kim, E. E. Bolton, PubChem chemical structure standardization, *J Cheminform.* **10** (2018) #36.

[21] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, *Nature* **585** (2020) 357–362.

[22] F. Hoonakker, N. Lachiche, A. Varnek, A. Wagner, A representation to apply usual data mining techniques to chemical reactions – illustration on the rate constant of SN_2 reactions in water, *Int. J. Artif. Intell. Tools* **20** (2011) 253–270.

[23] J. D. Hunter, Matplotlib: A 2d graphics environment, *Comput. Sci. Engin.* **9** (2007) 90–95.

[24] C. Jang, L. Chen, J. D. Rabinowitz, Metabolomics and isotope tracing, *Cell* **173** (2018) 822–837.

[25] W. Jaworski, S. Szymkuć, B. Mikulak-Klucznik, K. Piecuch, T. Klucznik, M. Kaźmierowski, J. Rydzewski, G. Anna, B. A. Grzybowski, Automatic mapping of atoms across both simple and complex chemical reactions, *Nat. Comm.* **10** (2019) #1434.

[26] P. C. Kroon, Pysmiles, `https://github.com/pckroon/pysmiles`.

[27] M. Latendresse, M. Krummenacker, P. D. Karp, Optimal metabolic route search based on atom mappings, *Bioinformatics* **30** (2014) 2043–2050.

[28] A. Lin, N. Dyubankova, T. I. Madzhidov, R. I. Nugmanov, J. Verhoeven, T. R. Gimadiev, V. A. Afonina, Z. Ibragimova, A. Rakhimbekova, P. Sidorov, A. Gedich, S. Rail, R. Mukhametgaleev, J. Wegner, H. Ceulemans, A. Varnek, Atom-to-atom mapping: A benchmarking study of popular mapping algorithms and consensus strategies, *Mol. Inf.* **41** (2021) #2100138.

[29] E. E. Litsa, M. I. Peña, M. Moll, G. Giannakopoulos, G. N. Bennett, L. E. Kavraki, Machine learning guided atom mapping of metabolic reactions, *J. Chem. Inf. Model.* **59** (2019) 1121–1135.

[30] E. Luks, Permutation groups and polynomial-time computation, in: L. A. Finkelstein, W. M. Kantor (Eds.), *Groups and Computation*, AMS, Providence, 1993, pp. 139–175.

[31] E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comp. Syst. Sci.* **25** (1982) 42–65.

[32] M. Mann, F. Nahar, N. Schnorr, R. Backofen, P. F. Stadler, C. Flamm, Atom mapping with constraint programming, *Alg. Mol. Biol.* **9** (2014) #23.

[33] R. Mathon, A note on the graph isomorphism counting problem, *Inf. Proc. Lett.* **8** (1979) 131–136.

[34] B. D. McKay, A. Piperno, Practical graph isomorphism II, *J. Symb. Comp.* **60** (2014) 94–112.

[35] S. Müller, C. Flamm, P. F. Stadler, What makes a reaction network "chemical"? *J. Cheminform.* **14** (2022) #63.

[36] R. Nugmanov, N. Dyubankova, A. Gedich, J. K. Wegner, Bidirectional graphormer for reactivity understanding: Neural network trained to reaction atom-to-atom mapping task, *J. Chem. Inf. Model.* **62** (2022) 3307–3315.

[37] R. I. Nugmanov, R. N. Mukhametgaleev, T. Akhmetshin, T. R. Gimadiev, V. A. Afonina, T. I. Madzhidov, A. Varnek, CGRtools: Python library for molecule, reaction, and condensed graph of reaction processing, *J. Chem. Inf. Model.* **59** (2019) 2516–2521.

[38] M. Okamoto, A polycarbonate-made optical article and method of preparation therefor, Google Patents, 1989, eP0305214A2.

[39] N. Osório, P. Vilaça, M. Rocha, A critical evaluation of automatic atom mapping algorithms and tools, in: F. Fdez-Riverola, M. S. Mohamad, M. Rocha, J. F. De Paz, T. Pinto (Eds.), *11th International Conference on Practical Applications of Computational Biology & Bioinformatics*, Springer, Basel, 2017, pp. 257–264.

[40] A. E. Petrarca, M. F. L. Lynch, J. E. Rush, A method for generating unique computer structural representations of stereoisomers, *J. Chem. Doc.* **7** (1967) 154–165.

[41] G. A. Preciat Gonzalez, L. R. P. El Assal, A. Noronha, I. Thiele, H. S. Haraldsdóttir, R. M. T. Fleming, Comparative evaluation of atom mapping algorithms for balanced metabolic reactions: application to recon 3D, *J Cheminform.* **9** (2017) #3.

[42] S. A. Rahman, M. Bashton, G. L. Holliday, R. Schrader, J. M. Thornton, Small Molecule Subgraph Detector (SMSD) toolkit, *J. Cheminform.* **1** (2009) #12.

[43] S. A. Rahman, S. M. Cuesta, N. Furnham, G. L. Holliday, J. M. Thornton, EC-BLAST: a tool to automatically search and compare enzyme reactions, *Nature Methods* **11** (2014) 171–174.

[44] S. A. Rahman, G. Torrance, L. Baldacci, S. Martínez Cuesta, F. Fenninger, N. Gopal, S. Choudhary, J. W. May, G. L. Holliday, C. Steinbeck, J. M. Thornton, Reaction Decoder Tool (RDT): extracting features from chemical reactions, *Bioinformatics* **32** (2016) 2065–2066.

[45] F. Rossello, G. Valiente, Chemical graphs, chemical reaction graphs, and chemical graph transformation, *El. Notes Theor. Comp. Sci.* **127** (2005) 157–166.

[46] P. Schwaller, B. Hoover, J. L. Reymond, H. Strobelt, T. Laino, Extraction of organic chemistry grammar from unsupervised learning of chemical reactions, *Sci. Adv.* **7** (2021) #eabe4166.

[47] C. Starke, A. Wegner, MetAMDB: Metabolic atom mapping database, *Metabolites* **12** (2022) #122.

[48] A. V. Zeigarnik, On hypercycles and hypercircuits in hypergraphs, in: P. Hansen, P. W. Fowler, M. Zheng (Eds.), *Discrete Mathematical Chemistry*, AMS, Providence, 2000, pp. 377–383.

[49] A. A. Zykov, Hypergraphs, *Usp. Mat. Nauk* **29** (1974) 89–154.