The Graph Grammar Library

a generic framework for chemical graph rewrite systems

Martin Mann[1], Heinz Ekker[2], and Christoph Flamm[2]
[1] Bioinformatics, Institute for Computer Science, Albert-Ludwigs-Universität Freiburg
[2] Institute for Theoretical Chemistry, University of Vienna

UNI FREIBURG

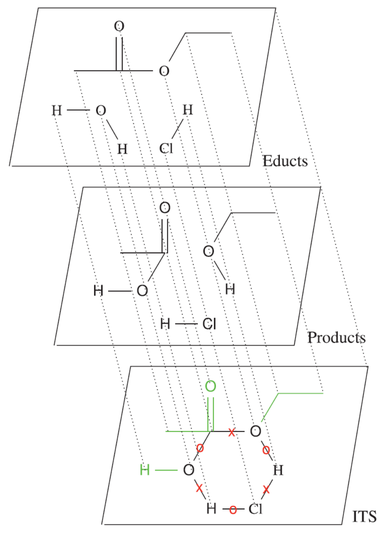
The Graph Grammar Library (GGL)

Graph rewrite systems are powerful tools to model and study complex problems in various fields of research. Their successful application to chemical reaction modeling on a molecular level was shown but no appropriate and simple system is available at the moment. The Graph Grammar Library (GGL) [1], fills this gap and provides feature-rich functionality especially for chemical transformation.

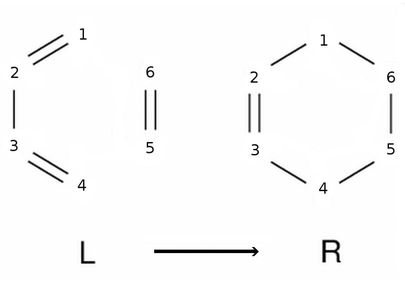
The GGL implements a simple generic Double Push Out approach for general graph rewrite systems on labeled undirected graphs. The object oriented C++ framework focuses on a high level of modularity as well as high performance, using state-of-the-art algorithms and data structures, and comes with extensive end user and API documentation. Central modules (e.g. graph matching, match handling, graph storage) are combined via simple interfaces, which enables an easy combining to tackle the problem at hand.

Preprocessing

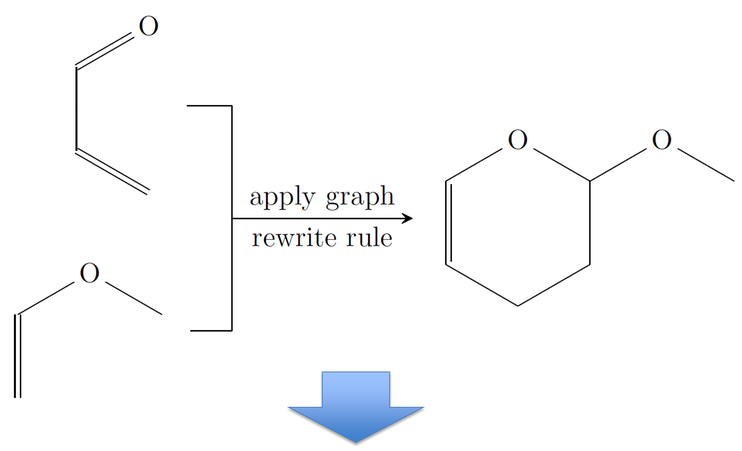
In order to apply graph grammar rules, a set of inial molecules has to be provided. Molecules are usually encoded in SMILES string notation that has to be parsed into molecular graphs. Since the SMILES notation ommits hydrogen information, a hydrogen filling step is applied to derive the full molecule graph encoding. A special problem for the latter is the identification and handling of aromatic rings that otherwise would enable different graph representations of the same molecule. We apply a novel machine learning based approach for aromaticity perception. Finally, all input molecules have to pass a couple of sanity checks to ensure their correctness and compatibility with the chemical graph rewrite system..



Graph Grammar Rule



Rule Application



Atom Mapping

Chemical Graph Grammar Rules can be designed either manually or derived based on atom mappings of known reactions. The latter requires the identification of the imaginary transition state (ITS), which basically encodes the molecular graph changes during a reaction.
We already provides rewrite rules for all enzymes listed in the KEGG LIGAND database.

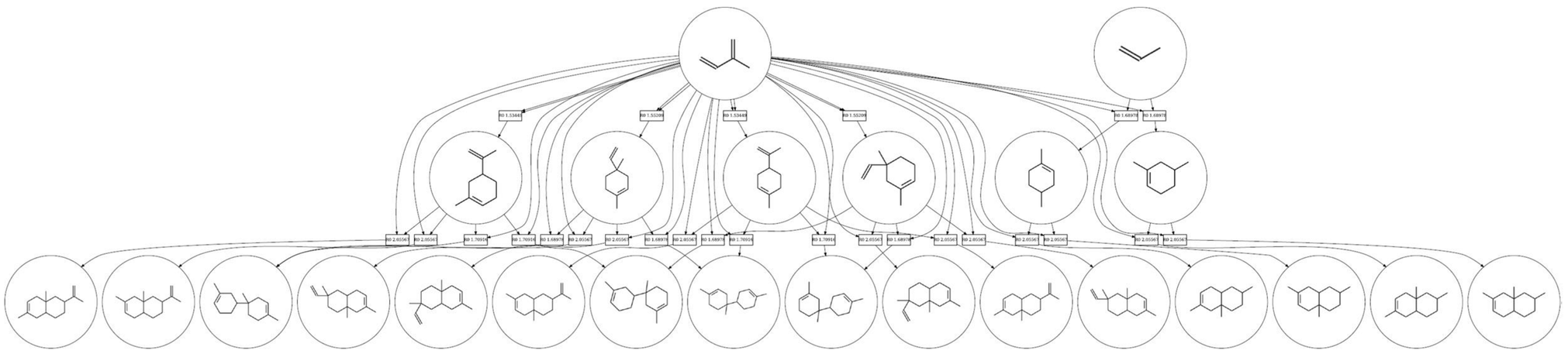
Graph Grammar Rule

We encode Graph Grammar Rules using a simple and easy to use graph encoding in the Graph Modeling Language (GML) format. It covers the encoding of the left side pattern L to be matched and its conversion into the result pattern R. In addition, further constraints can be provided to make the rule application more specific. This covers e.g. constraints on labels, adjacent edges or nodes, or the non-existence of edges.

Postprocessing

Molecules produced by chemical graph rewrites are not necessarily feasible molecular structures. We therefore apply a couple of filters on each produced molecule. Beforehand, an aromaticity perception has to be done to correct the representation of created or broken rings within the molecules.
Finally, we enable the estimation of the reaction rate using energy estimates in combination with the Arrhenius law.

Reaction Network Expansion



The large GGL chemistry module enables extensive and detailed studies of chemical systems. Here, molecules are represented as vertex and edge labeled undirected graphs while chemical reactions are described by according graph grammar rules, see Fig.~\ref{fig:example}. Such a graph grammar is a generating system for the explicit construction of an entire chemical space, i.e. all molecules reachable from the initial molecules by iterative reaction applications. An extensive system of wildcards, degree and adjacency constraints, and negative application conditions (NAC), such as the non-existence of edges, makes it easy to formulate very specific graph transformation rules by modulating their context dependent matching behaviour. Rules are encoded using the Graph Modelling Language (GML) easily understood and used by non-expert users. The molecule graphs produced by the graph grammar encoded chemical reactions have to pass extensive sanity checks and e.g. arromaticity correction to ensure the production of proper molecules only.

Besides the efficient handling of chemical transformation the GGL offers advanced cheminformatics algorithms. Among them are methods for the estimation of reaction rates or the free energies of molecules, the generation of canonical SMILES (a popular line notation for molecules) or chemical ring or aromaticity perception. Furthermore the entire functionality of the popular chemical toolbox Open Babel can be harnessed from within the GGL via the implementation of a bi-directional interface for the exchange of chemical graphs. All these features are used within the GGL-based toyChem tool, depcited here, which is part of the library. It enables the expansion and visualization of reaction networks given some initial molecules and a set of chemical reaction rewrite rules. The graph grammar based simulation of chemical reactions offered by the GGL is a powerful tool for extensive cheminformatics studies on a molecular level.

[1] Mann, M., Ekker, H., Flamm, C.: The graph grammar library – a generic framework for chemical graph rewrite systems. arXiv. http://arxiv.org/abs/1304.1356 (2013)