Explora: A new language to define powerful structural queries

Th Hanser, E Rosser, S Werner, P Görny

Lhasa Limited, 22-23 Blenheim Terrace, Woodhouse Lane, Leeds LS2 9HD, United Kingdom
thierry.hanser@lhasalimited.org

Abstract
Exploring chemical structure spaces is an increasingly important activity in modern Bio/Chem-informatics applications. Structural queries in chemical databases, fragment statistics, patenting and SAR modeling are only a few examples of where structural spaces and queries are key elements. At Lhasa Limited we use structural queries to define 2D toxicophores (Derek®) and biotransformation sites (Meteor®); they capture the expert’s SAR knowledge in these fields of structural queries. From an abstract view point, structural queries describe a domain in the chemical space that is of interest in a given context. In most contexts, the accuracy of these queries is critical and their fine grained expression is challenging.

Current approaches use structural information refined with generic atoms or bonds or more sophisticated descriptors like SLN®, SMARTS® strings and more recently MQL® queries. In some cases, these approaches are not flexible enough and the resulting queries are often not intuitive. In order to overcome these limitations and create accurate and easy to read chemical spaces, we have developed Explora, a powerful structural query language. This work is an extension of the development of the concept of L-Patterns (logical Markush structures).

Defining accurate chemical scopes

Describing a chemical scope or a structural query is like creating a painting, the language used to define this scope is like the brush used by the painter. The finer the brush the more accurate will be the painting. If the scientist (the painter) does not have the right query language (the right brush) he won’t be able to accurately define the intended chemical scope.

Explora is an expression language that is used to define constraints on individual atoms, bonds or the structure as a whole. We wanted to keep the benefit of a 2D visual representation of the structure together with the power of an expression based attribute definition. The user defines a query structure using a structure editor. Each atom, bond or the structure as a whole can then be further enriched with an Explora script to express a set of constraints. Explora scripts using descriptors, functions and operators allow the user to build sophisticated constraints thus contouring accurately the intended chemical scope of the query. The example below presents a simple use case of Explora.

An intuitive scripting language

Ideally we would like Explora scripts to be close to plain English statements. We therefore defined a context-free grammar that supports functions, operators, lists, ranges, etc. and introduces the concept of descriptor (function with zero arguments). The grammar has been designed to provide high legibility and minimum interpretation effort. Since Explora is meant to be extended to a full chemical scripting language in the future, its grammar has also been carefully shaped to anticipate features like variables, value assignments, flow control, encapsulation, etc. The examples in the following tables demonstrate the flexibility and the legibility of Explora.

User friendly

To facilitate the input of Explora based structural queries we have developed a structure editor and a fully featured Explora editor. The user can easily attach Explora scripts to an atom or a bond using the Explora tool in the structure editor. The Explora editor supports syntax colouring, error highlighting, multiline comments, etc. It also provides integrated help and the possibility to apply in real-time the Explora script on a test structure.

Fast and Easy to integrate

The Explora grammar has been designed using ANTLR® and we developed the corresponding compiler in JAVA. Explora scripts can be directly integrated as instances of JAVA objects ready to be invoked in a query (at a later stage). They are therefore very fast and easy to integrate in any software running on the JAVA platform. A typical Explora expression is executed in a few micro seconds.

The compiled Explora JAVA objects provide an API that can be easily called inside search algorithms (e.g. Ulman subgraph isomorphism) or for general screening purposes. Our reference implementation of Explora is based on CERES the JAVA based chemical engine used at Lhasa.

Since Explora expressions are text based scripts, they can be easily merged in chemical information file formats that support string based properties. CERES uses an XML based structure file format that supports Explora expressions in a natural and elegant way. Alternatively, Explora expressions can also be integrated in the MDL SD file format as SD tags.

Extensible

The current version of Explora provides a collection of ready-to-use operators, functions and descriptors designed to cover our needs in SAR modeling. The language can be easily extended with new functionalities written in Java thanks to a simple and dynamic language definition API. The developer creates a JAVA class that implements the extension API and implements the desired functionality in the eval() method. An instance of this class can then be registered at run time as a new feature of the language.

Current development of Explora includes new physicochemical descriptors, additional ring types (e.g. bridge head) and atom hybridization types.

Conclusion and perspectives

Explora’s key idea is to enrich structural queries with constraints that can be expressed in a domain specific language. We designed Explora to capture these constraints in the form of scripts that are easy to interpret and fast to execute. Explora’s extensible library of functions combined with logical operators provides a new structural query paradigm. The core language can be easily and dynamically extended with new functionality written in JAVA. In the future Explora will evolve into a fully featured chemical language supporting variables, flow control statements, parall processing, etc. We hope that Explora will become a powerful, natural, intuitive and attractive way to build chemical information workflows without requiring programming skills.