Documentation: Motif Finder

# Introduction

Motif Finder is a project designed to find commonalities among scientific workflow templates and executions. The project explores a workflow repository and analyzes its contents applying different graph mining techniques. As a result, a set of the most common workflow fragments is obtained.

The final results are linked to the original contents of the repository according to the Workflow Fragment Description Ontology: <http://purl.org/net/wf-fd>. In addition, some of the fragments are exposed as internal macro and composite workflow motifs, as defined in <http://purl.org/net/wf-motifs#InternalMacro> and <http://purl.org/net/wf-motifs#CompositeWorkflow> respectively.

# Overview

Motif Finder exposes an extensible framework to:

1. **Query different scientific workflow specifications published in a repository**. (Currently supporting OPMW/OPM workflows).
2. **Reformat these specifications to produce ready for execution scripts for the different graph mining techniques**. After retrieving the dependencies between the different steps of the workflow, we expose the resultant graphs according to the format required by the different graph mining algorithms. (Currently supporting the SUBDUE algorithm, WIN OS).
3. **Interpret the results to find the different fragments found by each technique**.
4. **Produce the links between the fragments obtained and the different workflow specifications**. By stating this relationship, we are able to determine where the different fragments were found in the repository, and how often they appear.
5. **Expose the final results in RDF**, so we can query them via SPARQL queries.
6. **Validate the produced results**. In order to check that the results have been produced correctly, we

# Workflow

Explain that rather than an executable file, what is exposed here is a framework with all the functionalities, and offer them consistently is a work in progress.

Figure 1: Main abstract workflow for retrieving the most common workflow fragments, given a repository URI and a particular graph mining algorithm.Missing explanation of the edges

# Functionality

Here we have to talk about the input formats and the output formats as well. And the usage as well

What is the main architecture of the system?

What are the structures being used, and how do they relate to each other?

What is the main Diagram of UML classes modeling everything?

I have to explain here also the different functionalities of what the project does: how to process a template or trace form a repository (and which classes to extend), how to change a domain, how to save the files in a format, how produce the script to actually run SUBDUE or any other graph, how to create statistics, how to validate the fragments found, how to generate the fragment catalog in wffd, etc.

How to switch from different algorithms and how to attack different sources (which is in fact easy).