

Scarph: an implementation of the dagger categorical model for graph data

Alexey Alekhin¹ and Eduardo Pareja-Tobes¹

¹*oh no sequences!* research group, [Era7](#) bioinformatics

June 11, 2015

Abstract

What is *Scarph*?
Baby, don't hurt me,
Don't hurt me no more!

Contents

[Introduction](#)

2

Introduction

At the moment the standard way of accessing a graph database is either by using Blueprints Java API or one of the specialized graph query languages, such as [Cypher](#) or [Gremlin](#). The common problem of all such approaches is that they don't take into account the graph schema on the language level. So using these query languages one can easily make a mistake and write a *syntactically correct* query, which doesn't make any sense from the point of view of the data model and therefore will *fail in runtime*.

As most of our work is concentrated on using Bio4j in the cloud distributed systems, it is really important to be able to formulate a query to the database in such way that will strictly conform to the data model, making the safety of evaluating this query known in advance.

While there are several frameworks to work with SQL databases from Scala *in a type-safe manner* (see [ScalaQuery](#), [Squeryl](#), [Sqltyped](#) and [Slick](#)), there is nothing similar for graph databases. Scala is our language of choice because of it's effective combination of powerful type system, functional programming paradigm and compatibility with Java. So we want to take advantage of the host language type system to be able to work with Bio4j in a way that is clearly and strictly determined by the graph schema. This the aim of the **Scarph** project which obviously target the second global objective of our project: to develop a new system to extract a particular type of data from the Bio4j graph database.

So, in short, **Scarph** is an API / DSL (domain specific language) for accessing graph DB and building queries in a type-safe manner.