SNP based literature and data retrieval

Werner Veldsman and Alan Christoffels

3523520@myuwc.ac.za; alan@sanbi.ac.za

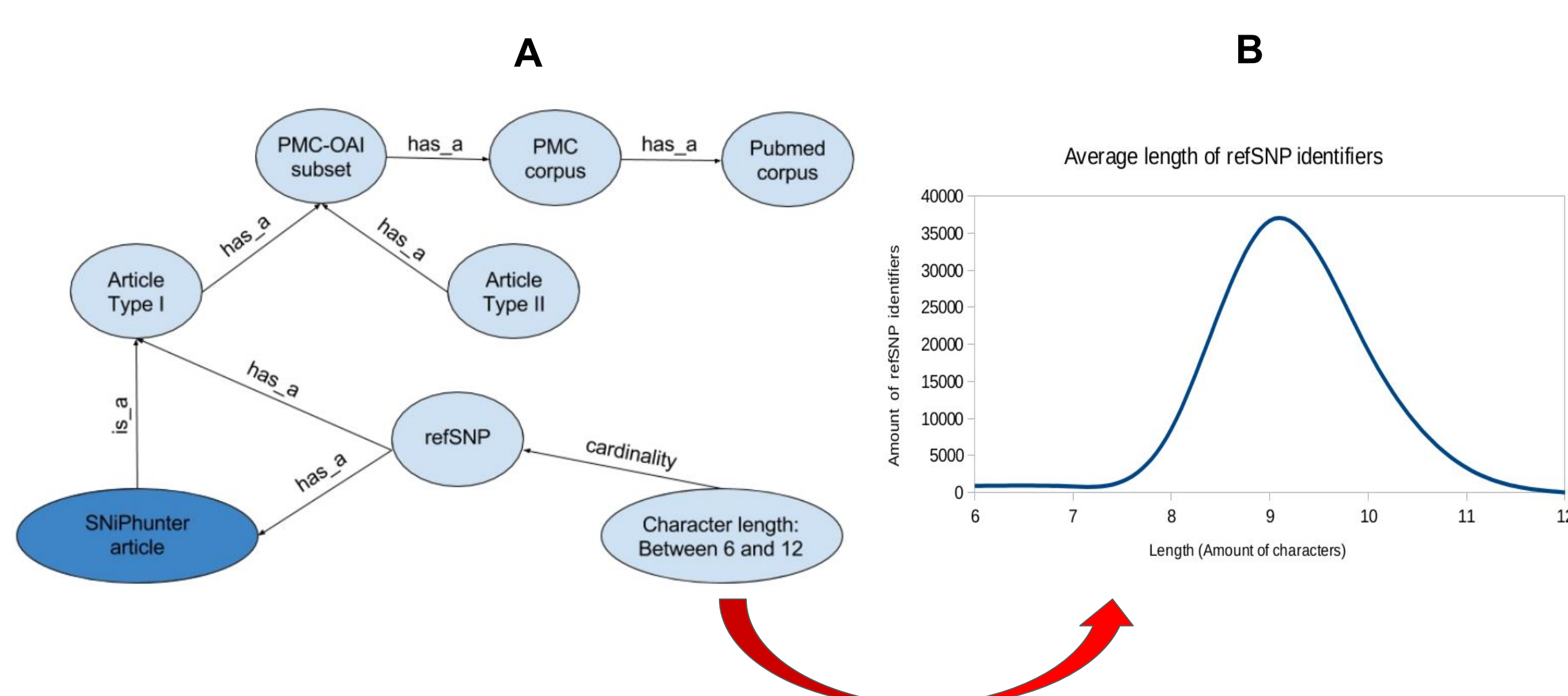Project website: http://sniphunter.sanbi.ac.za

Introduction

Biomedical literature retrieval is a time consuming process. Popular search engines such as Google Scholar, Pubmed and Scopus have useful features, however, ontological integration of search engines has not yet advanced to the state where optimum results are returned to a user interested in retrieving academic literature relevant to a reference SNP identifier (refSNP). For this reason, the results retrieved when using refSNPs as keywords, are not specific enough. The goal of this project was to design a proof-of-concept search engine that returns results to the user with an indication of the relevance of a refSNP keyword to articles within the Pubmed Central open access initiative. This project contributes to the resources provided by the Southern African Human Genome Programme.

Methods

Figure 1: SNiPhunter data was derived from Pubmed's PMC-OAI subset (**Figure 1A**). Potential refSNP identifiers could be constrained to a character length range of between six and twelve characters (**Figure 1B**). This avoided inclusion of non-refSNP terms due to ambiguities with unrelated naming conventions.
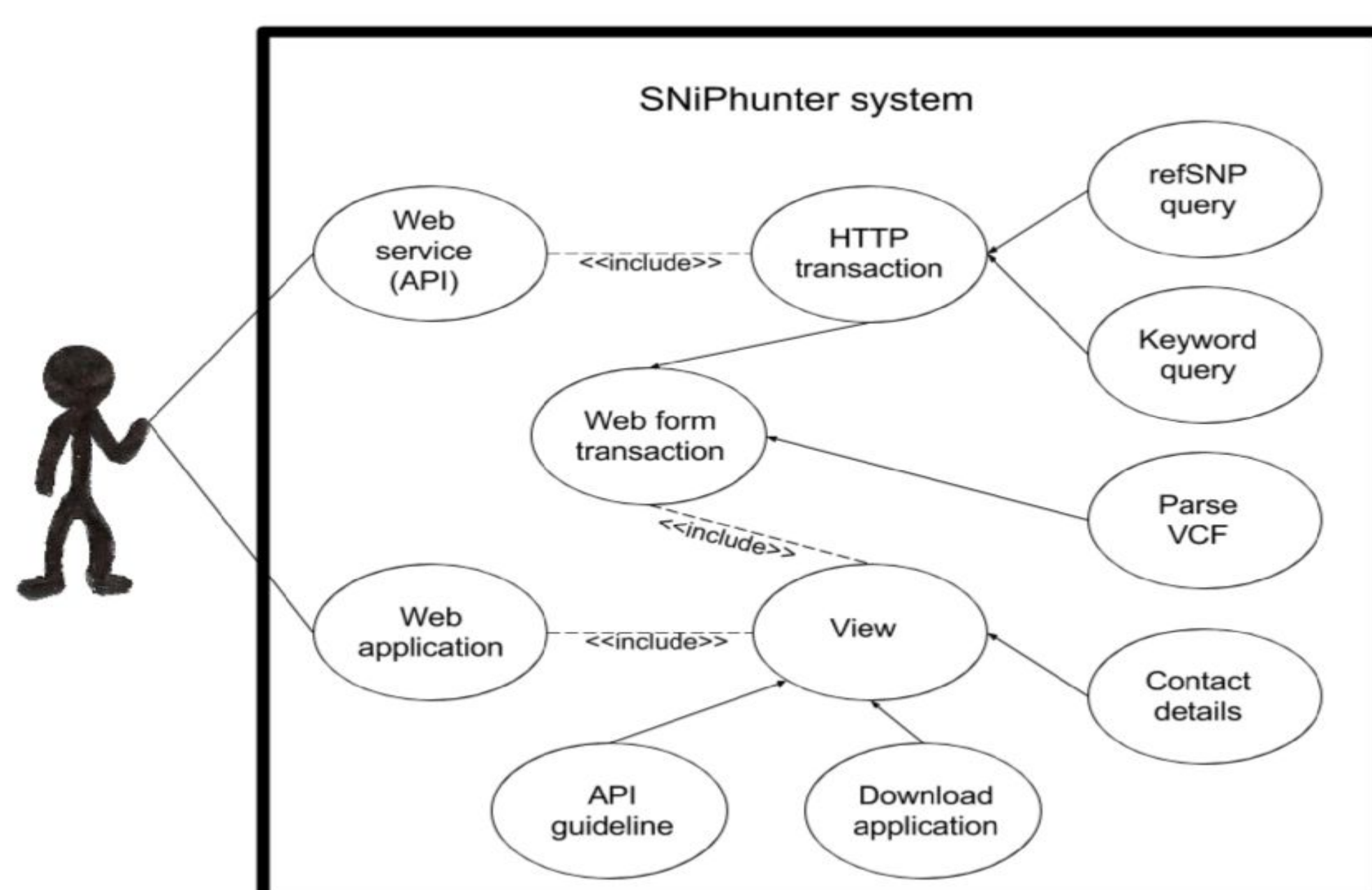
Figure 2: SNiPhunter search engine was designed with a dual interaction model that allows its database to be accessed by both humans (via a website) and machines (via an API).

Results

Example search using rs1000

Figure 3: (**A**) SNiPhunter home page, (**B**) search term entered prior to query launch and (**C**) result returned.

Take a photo of the quick response code (QR code) image to the right and submit the photo to a QR code decoder. You will be redirected to the SNiPhunter search engine.

Conclusion

A search engine, called **SNiPhunter**, was designed. It has an index of 69,463 unique refSNP identifiers and 8,743 author defined keywords in 20,650 biomedical articles. A variant call format (VCF) file parser and multiple query file upload service is packaged with the application. The underlying NoSQL database allows RESTful access, and serves results via a web application and through an application programming interface (API).

Acknowledgments