Biocaml: The OCaml Bioinformatics Library

Ashish Agarwal, ashish.agarwal@nyu.edu, New York University

Sebastien Mondet, sebastien.mondet@nyu.edu, New York University

Philippe Veber, philippe.veber@univ-lyon1.fr, Centre National de la Recherche Scientifique

Christophe Troestler, Christophe.Troestler@umons.ac.be, Université de Mons

Francois Berenger, berenger@riken.jp, Riken

Abstract

Biology is an increasingly computational discipline due to rapid advances in experimental techniques, especially DNA sequencing, that are generating data at unprecedented rates. The computational techniques needed range from the complex (.e.g algorithms, distributed computing) to the simple (e.g. scripting, parsing), and there are hundreds of thousands of Biologists now involved in computing. We propose that OCaml can serve virtually the full spectrum of computational tasks needed by Biologists, improving both programmer productivity and computational efficiency. To support this end, we have developed Biocaml.

Biocaml aims to be a standard library for the Biology domain. We provide features that are needed in a broad range of applications and avoid including overly specialized methods. The current feature set can be split into 3 broad categories: stream parsing/printing of many data formats, data structures for genomics, and access to public data repositories. We will demonstrate how some complex calculations can be performed quite easily with the current API, and describe our efforts to make a uniform API with comprehensive documentation. Finally, there is a BioX library for X equal to any programming language. The most widely used is BioPerl, and we will compare Biocaml with these alternatives.

Biocaml and other OCaml libraries have now been successfully used in multiple high-profile Biology projects (e.g. modENCODE, ENCODE, NYU's Genomics Core Facility, and others). Some time will be spent discussing the social aspect of bringing a novel language to the Biology community. We will attempt to elucidate strategies that are successful and those that are not. In particular, it will be argued that discussions regarding programming language choices need to be more scientific.