Quantitative Biology Center (QBiC)

Dr. Sven Nahnsen

Project supervisors:
Sven Fillinger,
Alexander Peltzer (alexander.peltzer@qbic.uni-tuebingen.de)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Summer Semester 2018

Data Management in Quantitative Biology

(BIOINF4399C)

Project 1: File-format Validation Tools

Hand-in date: 06.07.2018

Presentation date: 12.07.2018

Guidelines:

- Please upload your solutions (code and pdf) in a zip file to Ilias.
- Your code should be well-documented. If needed, provide an explanation on how to run it.
- Start early, so eventual problems can be fixed in time.

General Information

Ensuring data compatibility is amongst the most frequent tasks in biological data management. Although this can be achieved conceptually by applying checksums to ensure data consistency between computing resources, the fundamental validity of a biological data file format cannot be checked using, e.g., CRC32 or MD5 checksums. Unfortunately, this poses severe consequences for the automated data analysis in highly-automated processing facilities, as malformed or non-standard data can enter a facility in various ways and causes production pipelines to pause or (in the worst case) fail.

For this project, we are looking for a team to perform a literature or online survey of methods suitable to validate various file formats commonly found and used in computational biology (e.g. SAM/BAM, VCF, FastA, . . .). In case there is no example data, we will ensure that the team has access to sample files. After that, we intend to have the team implement an application to validate various biological file formats.

Task 1 *Implement a command line (CLI) app for file-format validation*

(a) Discuss the available methods that are already present

(b) Implement a Python-based application (with CLI, e.g. with `picocli`) to validate various formats

(c) Provide testcases for each supported file format (Unit tests)

(d) Set up a continous integration service (Travis CI) to automatically test your work

Task 2 *Extending the application and maintaining it*

(a) Extend the application by adding several file formats commonly used

(b) Create a PyPi `https://pypi.org/` project out of your CLI application

(c) Create a Bioconda `http://bioconda.github.io/` package for your application

(d) Create a project report.