MultiQC: Summarize analysis results for multiple samples in a single report

Philip Ewels1,*, Måns Magnusson1, Max Käller2.

1 Department of Biochemistry and Biophysics, Science for Life Laboratory, Stockholm University, Stockholm 106 91, Sweden

2 Science for Life Laboratory, School of Biotechnology, Division of Gene Technology, Royal Institute of Technology, Stockholm, Sweden

* To whom correspondence should be addressed.

Abstract

Summary Fast and accurate quality control is essential for studies involving next-generation sequencing data. Numerous tools exist to quantify QC metrics, but assessing analysis results across an entire project can be time consuming and error prone. For example, it is not possible to interpret hundreds of FastQC reports; simply picking a handful may miss group patterns or outlier samples.

Results We present MultiQC, a tool to create a single report with interactive plots generated for multiple analyses across many samples. Reports enable global trends to be quickly visualized and outlier samples identified. MultiQC can plot data from many common bioinformatics tools and is built to allow easy extension and customization. It requires little or no configuration, is run on the command line and generates a single stand-alone HTML report.

Availability and implementation The package is available with an MIT license through GitHub and the Python Package Index. Example reports, documentation and downloads are available at: http://multiqc.info

Introduction

Advances in next-generation sequencing are leading to an avalanche of data. Whilst opening doors to new analysis types and experimental designs, this provides a challenge for bioinformaticians. A typical analysis can yield many log files and analysis metrics. This leads to the complex and time consuming task of finding and compiling statistics from numerous reports and log files for every sample.

MultiQC addresses this problem by searching an analysis directory for any recognized content and creating a single summary report. It is a general use tool, designed to parse log files from multiple programs and create plots with overlaid data. This provides a fast method to scan key statistics for a project quickly and easily. Shared plots allow accurate comparison between samples, which is not possible when switching between different reports. It produces a stand-alone HTML report with embedded interactive plots and tools.

MultiQC is suitable for integration with existing workflows with little to no configuration. Its routine use can improve data interpretation and save a great deal of time for bioinformaticians.

Running MultiQC

MultiQC is run on the command line - specified paths are searched recursively for any recognized files and a report is compiled. These are HTML files which can be opened in any modern web browser.

Plots are resizable and interactive, some with click and drag zooming. A toolbox allows sample renaming, highlighting and hiding. Plots can be exported in a range of publication-ready formats, including PNG, SVG and PDF. Parsed data is also saved as tab delimited text files for downstream use. This is especially useful when MultiQC is used with single cell data, which can have thousands of samples.

MultiQC is run in the same way for any analysis type. Each module will run only if it's log files are present, meaning that little to no configuration is required. At the time of writing, MultiQC comes with modules to parse logs from a range of tools including FastQC, FastQ Screen, Cutadapt, Bowtie 1/2, STAR, Tophat, Bismark, Picard, Preseq, Subread featureCounts and Qualimap.

Example use: Development project

- Comparing insert sizes
- Needs bioanalyzer module
- Duplicates / reads aligned

Example use: Sequencing facility

- QC of samples (FastQ Screen / FastQC)
- Spotting failed samples

Author Contributions

PE responsible for the idea, the package and the manuscript. MM helped with some code reorganization. MK directed the research group.

Acknowledgements

The authors would like to thank Guillermo Carrasco, Robin Andeer, Johannes Alneberg and Remi Ohlsson for useful discussion. Guillermo Carrasco was responsible for the first version of the Qualimap module.

References