



Multiscale Complex Genomics



**Project Acronym:** MuG

**Project title:** Multi-Scale Complex Genomics (MuG)

**Call:** H2020-EINFRA-2015-1

**Topic:** EINFRA-9-2015

**Project Number:** 676556

**Project Coordinator:** Institute for Research in Biomedicine (IRB Barcelona)

**Project start date:** 1/11/2015

**Duration:** 36 months

## Milestone 14: Publication of public data APIs

**Lead beneficiary:** European Molecular Biology Laboratory

**Dissemination level:** PUBLIC

Due date: 15/04/2017

Actual submission date: 15/04/2017

## Document History

Version	Contributor(s)	Partner	Date	Comments
0.1	Mark McDowall	EMBL-EBI	24/03/2017	First draft
0.2	Andrew Yates	EMBL-EBI	04/03/2017	Second draft. Clarifications added to all sections
1.0	Andrew Yates	EMBL-EBI	13/04/2017	Release Candidate

## Table of Contents

<b>1 EXECUTIVE SUMMARY</b>	<b>3</b>
<b>2 INTRODUCTION</b>	<b>3</b>
<b>3 DOCUMENTING APIs</b>	<b>3</b>
<b>4 CONCLUSION</b>	<b>4</b>

# 1 EXECUTIVE SUMMARY

For all software provided by the MuG consortium there will be freely accessible documentation provided so that the functions of the software are clear. The documentation has 2 audiences that have to be considered, the user and the developer, each has their own requirements. Based on MS13, software created by the consortium is made available through the MuG organisation on GitHub (<https://github.com/Multiscale-Genomics>). We have used ReadTheDocs as a publically accessible location for hosting each repository's documentation within the MuG GitHub organisation.

## 2 INTRODUCTION

With the creation of APIs, tools and pipelines it is important that these are fully documented so that others are able to install, use and develop the software. Documentation should be clear, easy to search and easy to view. Not only does good documentation allow the uses to get started with the APIs, tools and pipelines quicker, it also provides the user with level of confidence in the software that they are about to use. Clear and concise documentation reduces the need for extensive developer support when using a product. Good documentation is essential for the long-term sustainability of MuG software.

## 3 DOCUMENTING APIs

As defined in MS13, the minimum standard is to include a README file for every code repository with documentation about the repository, installation and simple usage instructions. We have chosen to implement these documents in Markdown, a lightweight markup language, to take advantage of GitHub's automatic rendering tools. README files are displayed on each GitHub repository's home page. Where this format is limiting the use of ReadTheDocs is then advisable as it provides the ability to extract the documentation from the code for each function within pipelines, tools and APIs that have been written for the project. ReadTheDocs websites are indexed by popular search engines including Google meaning that developers using MuG tools can use standard search methods to find help improving the usability of our software.

Access for the documentation data management API used by the Virtual Research Environment (VRE) is available at (<http://mg-dm-api.readthedocs.io/>). This describes the access protocols and the ways that tools are able to access the files within the VRE. Pipelines that use the DM-API include the mg-process-fastq and mg-process-files sets of pipelines, with links to all of the documentation from the repositories available at <http://multiscale-genomics.readthedocs.io>.

The DM-API will also back the REST APIs that are in development including mg-rest-dm (<https://github.com/Multiscale-Genomics/mg-rest-dm>) which directly uses the DM-API to list files for a given user.

## 4 CONCLUSION

The use of GitHub with README files and ReadTheDocs provide a complete suit of tools that allow for documentation and easy dissemination of APIs, pipelines, tools and documentation. These are standard methodologies used extensively in software development. Therefore this format of information will be expected by incoming developers.