DigestiFlow – Reproducible Demultiplexing for the Single Cell Era

Manuel Holtgrewe[1,2,*], Mikko Nieminen[1,3], Clemens Messerschmidt[1,2], Dieter Beule[1,3,*]

[1] Berlin Institute of Health (BIH), Core Unit Bioinformatics, Berlin, 10178 Germany

[2] Charité – Universitätsmedizin Berlin, Berlin, 10117 Germany

[3] Max Delbrück Center for Molecuar Medicine, 13125 Berlin, Germany

[*] Corresponding Authors

Emails: {manuel.holtgrewe,mikko.nieminen,clemens.messerschmidt,dieter.beule}@bihealth.de

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27717v2 | CC BY 4.0 Open Access | rec: 29 May 2019, publ: 29 May 2019

1 Abstract

An ever-increasing number of NGS library preparation protocols used in biomedical research requires complex barcoding schemes. In combination with the economic urge to use deep multiplexing on high volume sequencing devices this has turned the once mundane task of demultiplexing into a complex and error prone analysis step. At the same time, many organizations do not have sophisticated computer-aided tools setup for assisting with this task.

In this manuscript, we describe the package *Digestiflow* that focuses on the curation and management of Illumina flow cell raw data, in particular featuring excellent support for complex flow cell layouts. Namely, it allows for automated incorporation of flow cell runs, management of sample sheets, and the demultiplexing of Illumina base calls data. Digestiflow is an easy to implement, flexible, and extendable open source solution to address current and upcoming challenges in demultiplexing.

The software is available under a permissive open source license via https://github.com/bihealth/digestiflow-server, contributions are welcome.

Issue Section: Technical Note

Keywords: Demultiplexing, Sequencing, Illumina BCL, Flow Cell, Data Management

2 Introduction

Conversion from the base call (BCL) to read sequence (FASTQ) format is the first step after sequencing flow cells with Illumina instruments. In order to perform this processing step, one has to curate a sample sheet connecting the sample IDs and library names with single or multiple barcode sequences used and the lane information. This sample sheet is needed as input for subsequent demultiplexing and quality control steps. In our research organization we have already encountered flow cells with about 600 different libraries and expect further increases with emerging single cell applications and sequencing devices. Reliably handling such complex information is becoming increasingly challenging. Resolution of any assignment, information transfer, or simple typing errors is very difficult and time consuming because they are hard to detect as well as difficult and labor intensive to fix in an unambiguous way. Any undetected or insufficiently corrected error will ultimately reduce the power and conclusions of the downstream analysis or even spoil them completely. Thus, proper, reliable, and traceable performance of complex demultiplexing is an essential step for reproducible single cell research.

Many research organizations still keep their sample sheet information in spread sheets, which has many generic tracking and software specific[1,2] problems. This leads to hard to accept error rates and problem handling workloads. A possible alternative is the introduction of elaborate laboratory information systems (LIMS) that handle and track all relevant sample sheet information. While this approach may be feasible and also advisable for large scale sequencing service providers that offer a certain and slowly evolving set of sequencing protocols, it is usually not appropriate and achievable in a fast-paced research context where sequencing labs aim to provide a flexible, up to date, and thus quickly evolving protocol range. This is due to the time, effort, and cost required for the implementation and maintenance of a LIMS that tracks all relevant information. Furthermore, none of the systems we are aware of offers the demultiplexing flexibility modern single cells research protocols require or provides all the sample sheet, adapter, and BCL consistency checks that are

necessary to reduce error rates. To the best of our knowledge, no current tool supports complex situations such as different single-cell library preparation protocols on a single flow cell, e.g., with different lengths of (molecular) barcodes. The support of mixed flow cells enables economic and flexible usage of large volume sequencers for single cell applications while circumventing the problems with manual sample sheet curation.

3 Results

compares popular software packages that range from full-blown LIMS systems to tools focused on the management of Illumina flow cells. While license costs are usually low when compared to the total cost of a sequencing lab, free and open source licenses lower the entry barriers for research labs and allow for continuity compared to discontinued commercial software. Self-hosting resolves any doubts regarding storing data outside of the organisation. LDAP authentication facilitates the integration into existing authorization infrastructure. An API is required for the setup of automation solutions. Sample tracking ranges between only providing sample/project IDs with an API and fully-fledged LIMS systems. Most systems offer basic demultiplexing capabilities for homogenous flow cells (e.g., using the same barcoding scheme and barcode length for all libraries). Flexible demultiplexing allows for combining arbitrary barcoding schemes. Systems providing sheet checks proactively detect problems within sample sheets. Finally, BCL checks allow for the comparison of barcodes from the sample sheet and the barcodes from the base calls.

Digestiflow is the only one featuring dedicated support for the management and demultiplexing of complex flow cell layouts. There are few integrated tools for the validation of sample sheets and matching of such sheets with the actual base call data. Not all are freely available to all users or avoid vendor lock-in and thus fully qualify for FAIR data management requirements3 (in particular requirements A1.1 and A1.2). Cloud-based solutions also raise data privacy concerns for human data. Other solutions solutions like openBIS ELN-LIMS4 require a complex setup of dependencies (e.g., openBIS) and do not work in a stand-alone fashion.

Table 1 compares popular software packages that range from full-blown LIMS systems to tools focused on the management of Illumina flow cells. While license costs are usually low when compared to the total cost of a sequencing lab, free and open source licenses lower the entry

barriers for research labs and allow for continuity compared to discontinued commercial software. Self-hosting resolves any doubts regarding storing data outside of the organisation. LDAP authentication facilitates the integration into existing authorization infrastructure. An API is required for the setup of automation solutions. Sample tracking ranges between only providing sample/project IDs with an API and fully-fledged LIMS systems. Most systems offer basic demultiplexing capabilities for homogenous flow cells (e.g., using the same barcoding scheme and barcode length for all libraries). Flexible demultiplexing allows for combining arbitrary barcoding schemes. Systems providing sheet checks proactively detect problems within sample sheets. Finally, BCL checks allow for the comparison of barcodes from the sample sheet and the barcodes from the base calls.

Digestiflow is the only one featuring dedicated support for the management and demultiplexing of complex flow cell layouts. There are few integrated tools for the validation of sample sheets and matching of such sheets with the actual base call data. Not all are freely available to all users or avoid vendor lock-in and thus fully qualify for FAIR data management requirements[3] (in particular requirements A1.1 and A1.2). Cloud-based solutions also raise data privacy concerns for human data. Other solutions solutions like openBIS ELN-LIMS[4] require a complex setup of dependencies (e.g., openBIS) and do not work in a stand-alone fashion.

Table 1 Comparison of commercial and free software for the management of Illumina flow cells information popular in the sequencing community based on important properties and features.

Metric	Digestiflow	BaseSpace Clarity LIMS	OpenBIS LIMS-ELN	MendeLIMS	MISO
License	MIT	commercial	free for non-commercial	free for non-commercial	GPL
Hosting	self-hosted	Illumina Cloud	self-hosted	self-hosted	self-hosted

LDAP Auth	✓	✓	✓	✓	✓
(REST) API	✓	✓	✓	–	✓
Sample Tracking	minimal: ID+API	advanced functionality	basic	basic	basic
Basic Demux	✓	✓	✓	✓	–
Flexible Demux	✓	–	–	–	–
Sheet Checks	✓	–	–	–	–
BCL Checks	✓	–	–	–	–

compares popular software packages that range from full-blown LIMS systems to tools focused on the management of Illumina flow cells. While license costs are usually low when compared to the total cost of a sequencing lab, free and open source licenses lower the entry barriers for research labs and allow for continuity compared to discontinued commercial software. Self-hosting resolves any doubts regarding storing data outside of the organisation. LDAP authentication facilitates the integration into existing authorization infrastructure. An API is required for the setup of automation solutions. Sample tracking ranges between only providing sample/project IDs with an API and fully-fledged LIMS systems. Most systems offer basic demultiplexing capabilities for homogenous flow cells (e.g., using the same barcoding scheme and barcode length for all libraries). Flexible demultiplexing allows for combining arbitrary barcoding schemes. Systems providing sheet checks proactively detect problems within sample sheets. Finally, BCL checks allow for the comparison of barcodes from the sample sheet and the barcodes from the base calls.

Digestiflow is the only one featuring dedicated support for the management and demultiplexing of complex flow cell layouts. There are few integrated tools for the validation of sample sheets and matching of such sheets with the actual base call data. Not all are freely available to all users or

avoid vendor lock-in and thus fully qualify for FAIR data management requirements3 (in particular requirements A1.1 and A1.2). Cloud-based solutions also raise data privacy concerns for human data. Other solutions solutions like openBIS ELN-LIMS4 require a complex setup of dependencies (e.g., openBIS) and do not work in a stand-alone fashion.

Table 1 also illustrates that there is an important gap to be filled by a true open source solution but also for supporting flexible demultiplexing of complex flow cell layouts. We designed and created the Digestiflow Suite (short: *Digestiflow*) for management, curation, sanity checking, and quality control of Illumina flow cells. We focused on supporting sequencing labs solely in the processes from Illumina BCL files to FASTQ files and the subsequent quality control thereof. The specific focus was chosen because generic tracking of samples information is a highly complex topic and requires integration with existing infrastructures, e.g., data management systems in different research labs or clinical information systems for which no canonical installation and interfaces exists. In our opinion, Digestiflow is positioned in a sweet spot in terms of comprehensive functionality, relative ease-of-use, and high degree of automation. Because the system is developed as open source and exposes its functionality in open interfaces, comprehensive solutions can be reached by integrating the Digestiflow components with existing infrastructure. For example, the Digestiflow REST API can be used by other services for implementing continuous sample tracking. In the opinion of the authors, such integration is best done by the embedding components and the staff operating the system rather than by the system itself. Thus, instead of providing an either too restrictive or overly complex framework, Digestiflow offers primitives and functionality that can be easily used for building solutions optimized for the particular installation's use case.

Most notably, Digestiflow helps discovering and resolving of demultiplexing fallacies. These include duplicate barcode sequences, barcodes specified in the sample sheet but missing in the sequencing data, and vice versa, and common contaminations such as PhiX sequence. It handles different demultiplexing tools such as Illumina bcl2fastq[5] and Picard tools[6] transparently, tracks used

parameters, provides predefined adapter data sets for popular kits (such as Illumina TruSeq RNA-Seq and Agilent SureSelect). It further supports complex indexing schemes such as mixing libraries with different molecular barcode lengths and schemes such as from the 10X Genomics platform, the Takara platform, or Agilent SureSelect XT. This allows to implement automated demultiplexing of single-cell libraries with divergent designs but also of libraries from complex low-input protocols. We have deployed Digestiflow to three sequencing units in our organization and partner institutes and the solution was welcomed with both wet lab and bioinformatics staff.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27717v2 | CC BY 4.0 Open Access | rec: 29 May 2019, publ: 29 May 2019

4 Methods

An overview of the architecture of the components of Digestiflow can be found in Figure 1. The Digestiflow components are shown with blue background while surrounding system components and users are shown with a light gray one. A full list of features at the time of publication is available in the supplemental material.

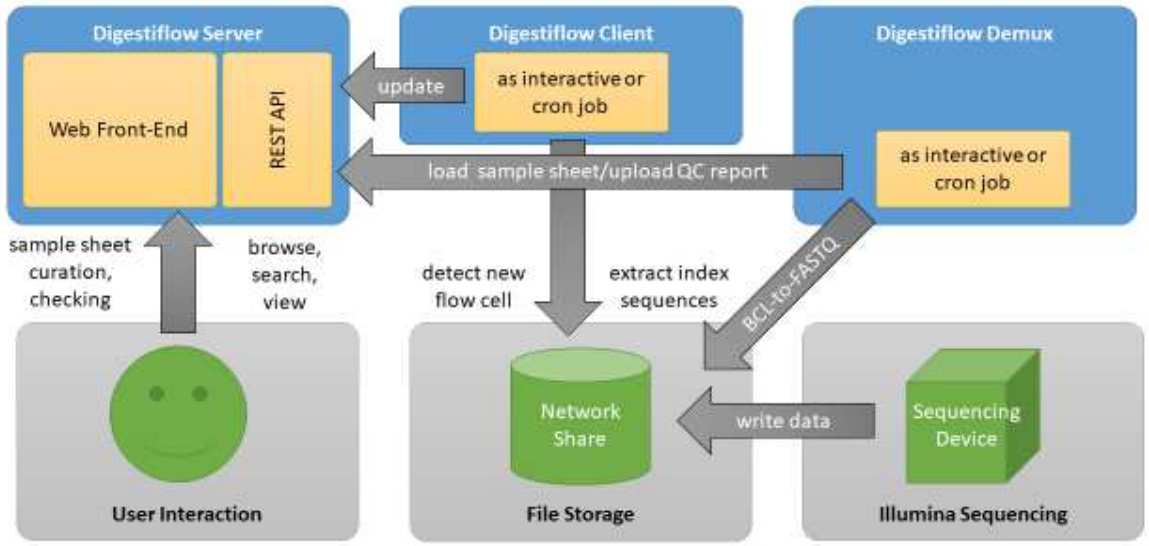

Figure 1 Architectural overview. Sequencing instruments write data to a specified file system storage. A periodically running Digestiflow Client detects new flow cells and registers them with the Digestiflow Server. Once sequencing is complete and sample sheet information has been approved by the operator, Digestiflow Demux performs the conversion to FASTQ files and creates all QC reports. Users can browse and view but also manage and curate flow cells and their sample sheets through Digestiflow Server.

4.1 System Architecture

The largest component is Digestiflow Server which is based on Python and Django. It provides the data model for flow cells, libraries, and barcodes as well as connected sequencing devices. Users can access and modify this information through an easy-to-use web front-end which also includes

authentication, authorization, and role management. A REST API is provided for automation and integration purposes. The web front-end allows for creating and editing of sample sheets as well leaving notes and comments on flow cell objects and assists users in sanity-checking the sample sheets as described above. Once sequencing of a flow cell is finished, its sample sheet has been approved, and marked as ready by an operator user the base calls can are converted to read sequences and full quality control reports are generated. The Digestiflow Client is a command line application developed in the Rust programming language and screens the file system for newly created BCL output directories. Metadata written out by the instrument is automatically extracted and the new base call directories are registered via the REST API of the Digestiflow Server.

The Digestiflow Demux component uses information from the Web REST API and information extracted from the base call directory on the file system. It is implemented as a Snakemake[7] based workflow for demultiplexing which calls Illumina's bcl2fastq with appropriate parameters after writing the necessary sample sheet files. Optionally, Picard[6] can be used for demultiplexing which allows for the automated processing of complex indexing schemes such as the ones commonly used in single-cell sequencing. After successful completion, automated quality control using FastQC[8] is performed and aggregated with MultiQC[9]. The results (or report of failure) is reported back to the Digestiflow Server REST API. The program call with all parameters and output are also made available in log files for the purpose of both keeping an audit trail and helping resolution in case of problems. The whole process can also be controlled by other third-party software components through the use of comprehensive APIs.

Further details on user and programmatic interfaces as well as on installation, validation, operation and documentation can be found in the supplemental material. All software is available under the permissive MIT open source license from our GitHub repositories. Digestiflow Server is developed as a "twelve factor" web application[10] and thus, designed for easy deployment in virtual machines,

containers, and platform-as-a-service environments. The other components Digestiflow Client and Digestiflow Demux are available as Conda/Bioconda[11] packages for easy deployment and usage.

4.2 Solving common demultiplexing fallacies with Digestiflow

This section enumerates common problems with demultiplexing data and how Digestiflow addresses them:

1) ***Barcodes from the sample sheet are not visible in the actual base calls.*** Digestiflow attempts to find every barcode from the sample sheet in the actual base calls. If this fails, a warning message is displayed in the web interface that can be used for investigation of possible problems. The user can then either correct the sample sheet or acknowledge the warning and thus suppress it.
2) ***Barcodes from the base calls do not exist in the sample sheet.*** Digestiflow attempts to look up every barcode from the base calls above 0.1% abundance in the sample sheet. Similar to (1), a warning is displayed if this fails that can either be corrected by fixing the sample sheet or acknowledging/suppressing the warning. Further, adapters existing in the base calls but missing in the sample sheet are looked up in the database of known adapter sequences. This allows to easily detect sample swaps, in particular if other sequences from the same library as detected in the base calls were used in the sample sheets.
3) ***Common contaminations are visible in the case calls and lead to false positives when resolving (2).*** Digestiflow Web ships with a list of common variants (e.g., poly-N or PhiX sequence) and provides a special highlighting for such sequence. These sequences are not flagged as error but only as “common contamination”. Further, if a sequence containing an “N” character occurs with above 0.1% abundance, it is ignored.

By employing such rules, Digestiflow is able to detect the majority of common problems with libraries in demultiplexing. As both sample sheet and base call index information can generally be

made available before the sequencing run finishes, problems can be detected while the sequencer is still running instead of following issues in the downstream analysis.

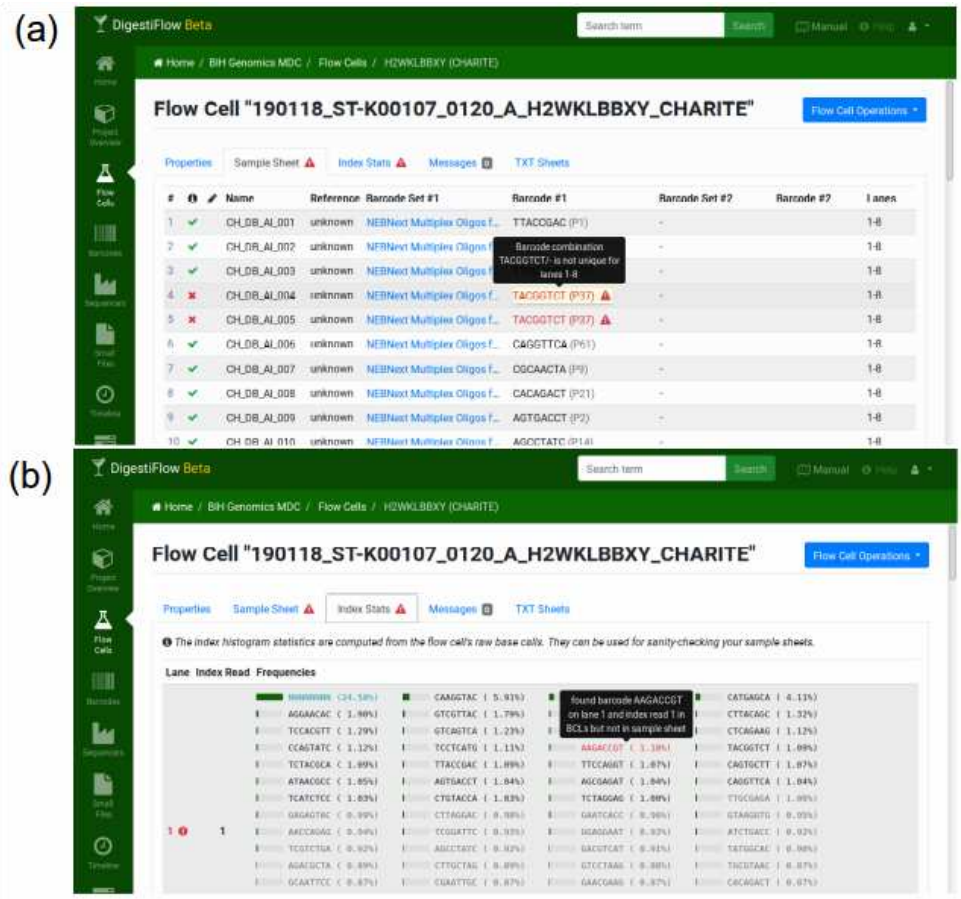

Figure 2 When adding the sample sheet (not shown), the operator made a small mistake. The adapter P37 is given twice for the same lane in the sample sheet while the adapter sequence "AAGACCGT" occurs in the raw base calls but not in the sample sheet. This information can then be used for debugging sample sheet information. This is highlighted in the sample sheet (a) and the display of the adapters read from the raw base call data (b).

4.3 Demultiplexing Strategies

The community standard tools for demultiplexing Illumina base call data are the vendor (Illumina) tool bcl2fastq and Picard Tools (short: Picard). While bcl2fastq is more efficient than Picard, it is

tightly coupled to how Illumina indexing works. At the time of writing, these *bcl2fastq restrictions* means up to two template reads ("T"; with bases from the sequenced fragment) and up to two indexing/barcode reads "B" in the order "template read, optional barcode read, optional second barcode read, optional second template read". A fixed number of bases/cycles can be skipped, but bcl2fastq does not support ligation of barcodes to template read sequence and also does not allow using molecular barcodes in places of index barcodes (as, e.g,. used by Agilent SureSelectXT Low Input protocols). Picard allows arbitrarily complex indexing schemes (as long as index lengths are fixed per library) but requires longer running times. It is thus desirable to use bcl2fastq where possible and fall back to Picard when necessary.

Digestiflow allows to split each template and index read into multiple ones but does not support output reads spanning more than one physical read from the sequencing program. For example, when sequencing one 100bp template read with one 10 bp index reads, 10M90T8B2S is valid (10bp molecular barcode, 90bp template read, 8bp index read, 2bp skipped) but 99T9I2S (10bp template read, 9 bp index read, 2bp skipped) is not as the index reads spans the physical template and index reads. Demultiplexing schemes where no output reads span the physical ones are called *valid*.

On the user interface side, users can enter a custom demultiplexing scheme for the flow cell, overriding the information automatically derived from the raw BCL information.

4.3.1 Reducing Single-Cell RNA-seq Data to the "Common Case"

As long as the output demultiplexing scheme is valid, there are no more than two of each template and indexing reads, and the order does not violate the bcl2fastq restrictions, bcl2fastq can be used for demultiplexing. Digestiflow recognizes this case and generates the appropriate sample sheet and call to the bcl2fastq executable. This optimizes running time over using Picard.

4.3.2 Supporting Exotic Indexing Schemes with Picard

For all other valid demultiplexing schemes, the sample sheet for and the comparatively more complex calls to Picard are generated by Digestiflow. This allows to implement automated “zero touch” demultiplexing of single-cell libraries from various platforms but of libraries from complex low-input protocols.

4.3.3 Mixed Flow Cells

The description above leaves one point open: how to handle flow cells on which more than one indexing scheme is used? Examples are HiSeq flowcells that have libraries from different single-cell library preparation platforms on different or even the same lane?

For this case, Digestiflow Web allows the user to optionally enter a demultiplexing scheme for each library. Digestiflow Demux will then gather all entered demultiplexing schemes for all libraries on the flowcell and perform one call to bcl2fastq or Picard for each demultiplexing scheme. The Snakemake workflow in Digestiflow Demux then combines the resulting FASTQ files into the output directories and regardless of the demultiplexing schemes given, will run the QC and QC aggregation steps on the resulting files.

4.4 Software Component Integration

Digestiflow supports the integration with existing authentication infrastructure using the LDAP protocol. Up to two LDAP repositories can be specified which allows for the integration with two different Active Directories domains, for example.

Digestiflow Web provides its data through an easy-to-use REST API. Thus, it is easy to develop new clients akin to Digestiflow CLI or Digestiflow Demux independent from the programming language used.

5 Discussion

As the first computational step in any high-throughput sequencing project, correct and efficient demultiplexing of raw base calls is of high importance. Still, prior methods for the management of sample sheets and the demultiplexing of Illumina base calls have considerable drawbacks of either being limited in functionality, being part of large and monoloithicith large other systems, and/or not available for self-hosting.

Digestiflow provides a modular, yet tightly integrated toolbox for the core tasks of managing Illumina flow cells and demultiplexing data. Moreover, focusing on only flow cells allows to provide an intuitive user interface covering all regularly needed functionality for curating flow cells and operating demultiplexing. Using a REST API, it is possible to easily extend the system and integrate it into existing infrastructure.

Digestiflow is developed as open source software on GitHub and welcomes patches and contributions by any party. To foster software quality, we added comprehensive software tests and automated them, e.g., with continuous integration solutions.

Overall, we hope that publishing our efforts in a well-documented, well-tested, and open source manner helps other members of the community in their raw flow cell management tasks. We have rolled out Digestiflow to three sequencing units in our organizations to replace “Excel files on network shares,” or worse “Excel files as email attachments,” solutions with great success. We are confident that other organizations will also find the middle-way between primitive means and a fully-fledge LIMS system easy both to adopt and helping them to improve their workflows.

6 Availability of source code and requirements

All software is available under a permissive open source license from our GitHub repositories. Digestiflow Web is developed as a "twelve factor" web application and thus designed for easy deployment in virtual machines, containers, and platform-as-a-service environments. The other components Digestiflow CLI and Digestiflow Demux are available as Conda/Bioconda[12] packages for easy deployment and usage.

Project name:	Digestiflow Suite
Project home page:	https://www.cubi.bihealth.org/projects/digestiflow
Operating system:	Linux
Programming Languages:	Python 3 and the Rust programming language
Other Requirements:	Dependencies are given by the respective Python/Rust build systems
License:	MIT License

7 Availability of supporting data

The Digestiflow Server manual contains a detailed tutorial with example data and scripts to create further example data.

8 Declarations

8.1 Supplementary Files

The *Digestiflow Server Manual* is attached as Supplemental Material in the version at the time of publication. An up-to-date version is available online and linked-to from the project and GitHub pages. This documented contains extensive documentation of the system and a tutorial about its practical setup and usage.

8.2 List of Abbreviations

API	Application Programming Interface
BCL	Base Call (file format)
BIH	Berlin Institute of Health
CLI	Command Line
FAIR	Findable Accessible Interoperable Reusable (data)
FASTQ	FASTA with Quality, a file format
ID	Identity
LIMS	Laboratory Information Management System
LDAP	Light
MDC	Max Delbrück Center for Molecular Medicine
MIT	Massachusetts Institute of Technology (also name of a software license)
NGS	Next-Generation Sequencing
QC	quality control
REST	Representational State Transfer

8.3 Competing Interest

The authors have no competing interests to declare.

8.4 Funding

All authors acknowledge support from the Berlin Institute of Health as staff.

8.5 Author's Contribution

MH, CM, and MN contributed to the implementation of the software. MC and DB edited the manuscript and CM and MN contributed to the manuscript's text.

8.6 Acknowledgements

The authors are grateful for the members of the sequencing facilities at Charite, MDC, and BIH and the the members of the Core Unit Bioinformatics of BIH for their feedback on Digestiflow.

9 References

1. Zeeberg, B. R. *et al.* Mistaken Identifiers: Gene name errors can be introduced inadvertently when using Excel in bioinformatics. *BMC Bioinformatics* **5**, 80 (2004).

2. Ziemann, M., Eren, Y. & El-Osta, A. Gene name errors are widespread in the scientific literature. *Genome Biol.* **17**, 177 (2016).

3. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data 2016 3* (2016).

4. Barillari, C. *et al.* openBIS ELN-LIMS: an open-source database for academic laboratories. *Bioinformatics* **32**, 638–640 (2016).

5. Illumina Inc. bcl2fastq Conversion Software. Available at: https://support.illumina.com/sequencing/sequencing_software/bcl2fastq-conversion-software.html. (Accessed: 22nd March 2019)

6. Broad Institute. Picard Tools - By Broad Institute. Available at: http://broadinstitute.github.io/picard/. (Accessed: 22nd March 2019)

7. Koster, J. & Rahmann, S. Snakemake--a scalable bioinformatics workflow engine. *Bioinformatics* **28**, 2520–2522 (2012).

8. Simon Andrews. FastQC A Quality Control tool for High Throughput Sequence Data. Available at: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/. (Accessed: 22nd March 2019)

9. Ewels, P., Magnusson, M., Lundin, S. & Käller, M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32**, 3047–3048 (2016).

10. Adam Wiggins. The Twelve-Factor App. Available at: https://12factor.net/. (Accessed: 22nd March 2019)

11. Chicco, D. *et al.* Bioconda: A sustainable and comprehensive software distribution for the life sciences. 1–12 (2017). doi:10.1101/207092

12. Grüning, B. *et al.* Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat. Methods* **15**, 475–476 (2018).