# 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, Core Unit Bioinformatics, Charitéplatz 1, 10117 Berlin
[2] Charité – Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin
[3] Max-Delbrück-Center for Molecular Medicine, Robert-Rössle-Straße 10, 13125 Berlin

Corresponding Author:
Dieter Beule[1,3]
Charitéplatz 1, 10117 Berlin
Email address: dieter.beule@bihealth.de

## 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. We present an easy to implement, efficient, flexible, and extendable open source solution to address this challenge.

## 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 laboratory information system 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.

## Results

Table 1 compares popular software packages that range from full-blown LIMS systems to tools focused on the management of Illumina flow cells. 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 A1.1 and A1.2). Cloud-based solutions also raise data privacy concerns for human data.

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 but outstandingly well 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[4] and Picard tools[5] 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 is very popular with both wet lab and bioinformatics staff.

A detailed tutorial on how to use the Digestiflow components (and scripts to create example data sets for the processing with Digestiflow) is available in the online manual that is also available in its current form as Supplement Material.

## Methods

An overview of the architecture of the components of Digestiflow can be found in Fig. 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 Online Methods and user documentation.

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. It also supports users in validating sample sheet barcode information and comparing these user-defined barcodes with the actual barcode sequence extracted from the raw base calls. 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[6]

based workflow for demultiplexing which calls Illumina's bcl2fastq with appropriate parameters after writing the necessary sample sheet files. Optionally, Picard 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[7] is performed and aggregated with MultiQC[8]. 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 online methods and user documentation. 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[9] 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[10] packages for easy deployment and usage.

## 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. Illumina Inc. bcl2fastq Conversion Software. Available at: https://support.illumina.com/sequencing/sequencing_software/bcl2fastq-conversion-software.html. (Accessed: 22nd March 2019)
5. Broad Institute. Picard Tools - By Broad Institute. Available at: http://broadinstitute.github.io/picard/. (Accessed: 22nd March 2019)
6. Koster, J. & Rahmann, S. Snakemake--a scalable bioinformatics workflow engine. *Bioinformatics* **28**, 2520–2522 (2012).
7. 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)
8. 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).
9. Adam Wiggins. The Twelve-Factor App. Available at: https://12factor.net/. (Accessed: 22nd March 2019)
10. Dale, R. *et al.* Bioconda: A sustainable and comprehensive software distribution for the life sciences. *bioRxiv* (2017). doi:10.1101/207092

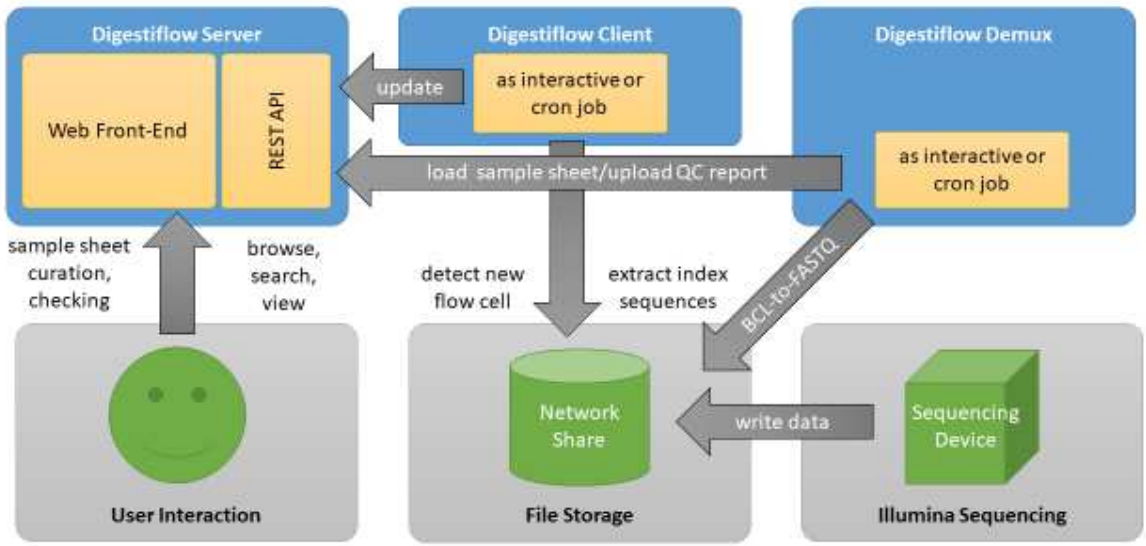
# Tables

**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. 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 sharing of data. 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.

| 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** | ✓ | – | – | – | – |

## Figures

**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.



## Supplemental Material

- Digestiflow Server Documentation

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27717v1 | CC BY 4.0 Open Access | rec: 9 May 2019, publ: 9 May 2019