Technical University Munich

TUM Data Innovation Lab

How to Handle Data from Living Cells

Anne Christopher
anne.christopher@tum.de

Magdalena Eberl
magdalena.eberl@tum.de

Sebastian Zett
sebastian.zett@tum.de

Scientific Lead:	Dr. Joachim Wiest (cellasys GmbH)
Co-Mentor:	Dr. Stephan Haug (Dept. of Mathematics)
Project Lead:	Dr. Ricardo Acevedo Cabra (Dept. of Mathematics)
Supervisor:	Prof. Dr. Massimo Fornasier (Dept. of Mathematics)

July 27, 2019

Abstract

cellasys works at the interface of electronic engineering and life sciences and delivers system solutions for Microphysiometry. The systems from cellasys are capable to monitor different parameters directly from living cells. These parameters include extracellular acidification rate (measured with pH sensors), cellular respiration (oxygen) and morphology (impedance). The information about cellular metabolic activity contained by measured sensor data dynamics is superimposed by manifold sources of error. A very careful consideration of data processing is necessary to eliminate these errors as much as possible and to facilitate the data for a correct interpretation about the cellular activities. The aim of this project is to deal with this raw data and prepare it for interpretation regarding the underlying cell activity. Protocols can then be used to determine for example the eye irritation potency of new chemicals and replace animal experiments. In short, we focus on **"How to handle data from living cells?"**. We aim to develop efficient and accurate algorithms to choose only valid data for interpretation. We also refine the data by filtering out noisy signals. The goal to be achieved is to make the data suitable for an interpretation about the cellular activities.

Contents

1 Introduction

This documentation summarizes the approaches and results of the TUM Data Innovation Lab's project *How to Handle Data from Living Cells*, conducted in summer semester 2019 in collaboration with cellasys GmbH.

1.1 cellasys and Microphysiometry

cellasys was founded in 2007 as a spin-off from Technical University Munich, providing high-quality approaches and systems for analysis of living cells. cellasys works at the interface of electronic engineering and life sciences and delivers system solutions for microphysiometry. Microphysiometry is a novel way of detecting the extracellular acidification rate (EAR) of cultured cells grown directly on the surface of a sensor chip. The measurement of EAR with pH sensors and oxygen uptake rates (OUR) with dissolved oxygen sensors is a well-accepted research tool for cell biology, using both electrochemical and optochemical sensor technology. cellasys employs the Intelligent Mobile Lab for in-vitro diagnostic (IMOLA-IVD) technology which involves multisensor devices for the purpose of cell monitoring.

cellasys performs standard microphysiometric experiments structured in discrete measurement intervals to generate raw sensor data. The main goal of this project is to devise techniques to pre-process this data in a careful manner to prepare it for interpretation about the underlying cellular activities.

1.2 Problem Definition

When conducting a microphysiometric experiment, cellasys collects data which can be distinguished as configuration data and measurement data. Configuration data gathers details regarding the experiment setup like configuration details of the pump, which injects the different media into the cell culture at regular intervals. In certain intervals, the system outputs the current measurement values of its sensors, which are stored with the respective time stamp. This is what we refer to as measurement data.

While the configuration data is required to track the set-up of the experiment, the measurement data may be used to infer conclusions about the metabolism of the cells. These deductions obviously require confidence in the validity of the underlying data. Since a microphysiometric system is a complex arrangement of purpose-built electronic

components, some unwanted behavior might disturb the experiment. These may refer to situations like, when a sensor may be defected during the experiment, when air bubbles interfere with a sensor, or when the door of the incubator containing the cell cultures was opened during the experiment causing a shift in the temperature which can affect the behavior of the cells.

As there is a wide range of factors that could disturb the experiment, it is of great importance to assess the quality of the measured data before using it for scientific purposes. In this project a trustworthy approach to estimate the validity and improve the quality of the measurement data through noise elimination was developed.

1.3 General Experimental Setup

A basic experiment involves microphysiometric cell assays and a fluidic system. The system transports cell culture media in an alternating pump and rest phases through "micro-reaction volumes" (MRVs), where the specimens are trapped and cultured on a Bio-Chip. Changes of the extracellular environment due to the activity of cellular core metabolism, i.e. increasing pH values and dissolved oxygen concentrations, will occur during the rest phase while the pump phase are necessary to re-establish the known conditions of the fresh medium.

The setup includes six modules (each being referred to as a different IMOLA).

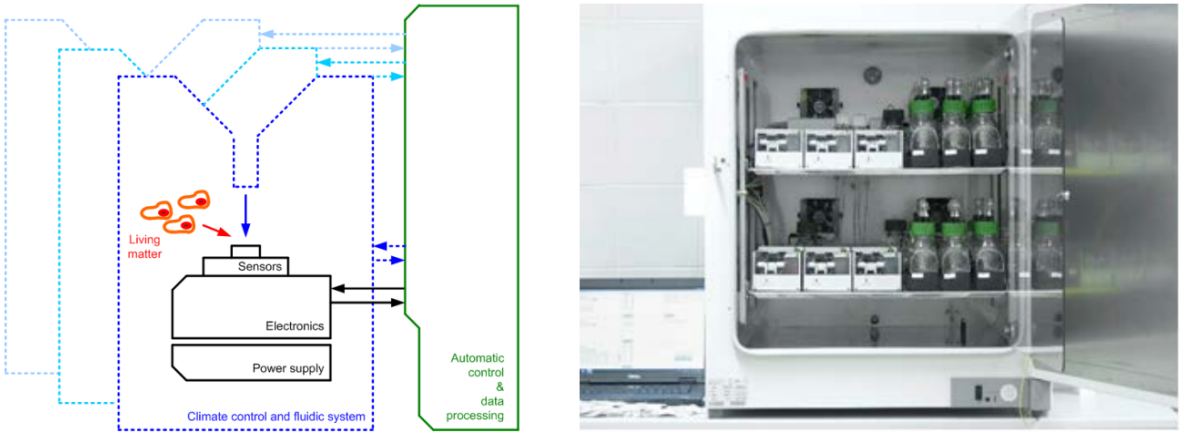


Figure 1.1: Setup of one IMOLA (left) and the incubator containing all six IMOLAs (right)

Some of the IMOLAs may serve as control modules to check the functionality of the experimental system: As positive control a fluid is applied to the cell cultures that will safely and quickly destroy any viability, whereas another module will not be treated with a toxic substance to assure that the cells can proliferate in the system (negative control). A third module will be run with no living cells to check if there are some interactions

with the micro-sensor system. The remaining three modules will be treated with the testing material.

For a common experiment, in the beginning the cell lines are seeded into the wells of the test plate and incubated for a certain time (e.g. 6h) under standard conditions for cellular attachment. To enable the detection of extracellular acidification the medium is replaced by a running medium. In the next period, the exchange of the medium in the MRVs takes place each ten minutes to apply fresh media and/or drugs. These ten minutes are considered as one interval, where the first five minutes correspond to the rest phase and the next five minutes correspond to the pump phase. After these standard conditions (running medium without additional agents), the test compound is added to the treated groups. In the end of the experiment, a toxic fluid (Sodium Lauryl Sulfate (SLS)) is added to all six IMOLAs to destroy the cell line.

2 Data Pre-Processing

2.1 Data Collection

The data obtained for analysis is obtained as a .exp file from the software DALiA client 3.0 from cellasys. This .exp file contains all the relevant data regarding the experiment as IMOLA measurements and pump configurations. However, the information from different sensors is collected separately and combined only in the output experiment file. For the data analysis it is required to join this available information from all the different data sensors into one single data frame. The required information from the .exp file is as follows:

- **Experiment Details**: Start time stamp, Stop time stamp, Experiment ID
- **IMOLA Measurement**: Sensor values for pH, temperature, oxygen and impedance (real and imaginary value) of the Bio-Chip under analysis
- **IMOLA Configuration**: IMOLA Power and LED ON/OFF details
- **Pump Configuration**: Pump speed
- **ISM Configuration**: Valves 1-4 ON/OFF details
- **Comments**: Bubble detected True/False information from two bubble detectors

All these separate data frames need to be combined to one single data frame with one row corresponding to one observation in one IMOLA at a certain time point.

2.1.1 Bubble

The steps defined above are carried out in a module named Bubble. The *Bubble.R* script firstly initializes separate data frames for each of the different data snippets from the .exp file. It also creates a column 'Time[s]' for each of the data frames, computed as the time elapsed from the start time stamp. Afterwards it assigns the most recent pump configuration, ISM configuration and further information to each IMOLA measurement by comparing the 'Time[s]' column of the different data frames. Finally it splits the entire data frame again into 6 data frames corresponding to the 6 different IMOLAs, specified by the 'ImolaNr' variable. These results are then written into 6 .dat (text format) output files and can be used to analyze the data in DALiA.

2.2 Data Description

As mentioned before, one experiment setup usually contains six modules. The modules measure metabolic and morphological parameters of the living cells, more specific pH value (2 sensors), temperature, dissolved oxygen, and impedance (2 sensors, each having a real and imaginary measurement value) as summarizes in table 2.1.

Parameter	unit
pH1	[mV]
pH2	[mV]
Temp.	[mV]
Oxygen	[mV]
$IDES1_{Re}$	[Ohm]
$IDES1_{Im}$	[Ohm]
$IDES2_{Re}$	[Ohm]
$IDES2_{Im}$	[Ohm]

Table 2.1: Sensor measurements

With this setup, continuous and real-time measurements for each five seconds can be provided. For a 24h experiment with six modules, we obtain 17,280 measuring points for each IMOLA, and 103,680 in total. The most important measure to describe cell metabolism is the change in pH, which can be used to calculate the EAR. This concept is based on periodic cycles with five minutes of rest and five minutes of pumping of the involved fluid into the system. In theory, an increasing pH value is expected during the rest phase. The rate of pH change corresponds to the EAR and can be measured as the slope of the pH curve. The impedance sensors' values can also provide information regarding the proliferation of cells. A temperature sensor is required to ensure the quality of the experiment, i.e to check if the cell culture was maintained at body temperature throughout the experiment.

2.3 Data Preparation

Before using the data in different analysis algorithms, two things need to be taken care of. Firstly, we need to check whether the data we obtained comes from real cell culture. There can be cases in which the measurement system was used for other purposes as checking the functionality of the sensors and hence did not use original cell culture for the procedure. We only use data for further analysis if it comes from a treated cell culture. Hence, a check for cell culture is implemented in module Open Circuit as described in section 2.3.1.

The second step of data preparation is a common data pre-processing procedure for data analysis. Often a comparison of the values of different sensors and/or IMOLAs

provides interesting insight. The sensors however may exhibit different scales, inhibiting the usefulness for comparison. Therefore, we prepare the data for further analysis procedures by normalizing the data as described in section 2.3.2.

2.3.1 Open Circuit

Certain test conditions exist to confirm that the system's electronical components work properly and the sensors record valid readings. This check is done by replacing the original cell culture and sensors with a test chip, called dummy Bio-Chip. Under certain conditions readings are also taken with the system having an open circuit (i.e no voltage difference across points in the circuit).

The module Open Circuit checks whether the recorded data falls into either of these categories by comparing the range of the pH, temperature, oxygen and impedance readings from the data against predefined ranges for a dummy Bio-Chip and open circuit systems. For each IMOLA and measure, the percentage of values falling into the two ranges is recorded in two separate tables. Finally, in each table a flag is given to each IMOLA, specifying whether this IMOLA has a dummy Bio-Chip inside or is open circuit, respectively. This flag becomes true if and only if all recorded values fall into the respective range. The module can be run directly on an experiment file. The underlying logic is also used as an initial step in the module Validation, since any data that shows dummy or open circuit behaviour is not valid for further analysis.

2.3.2 Pre-processing

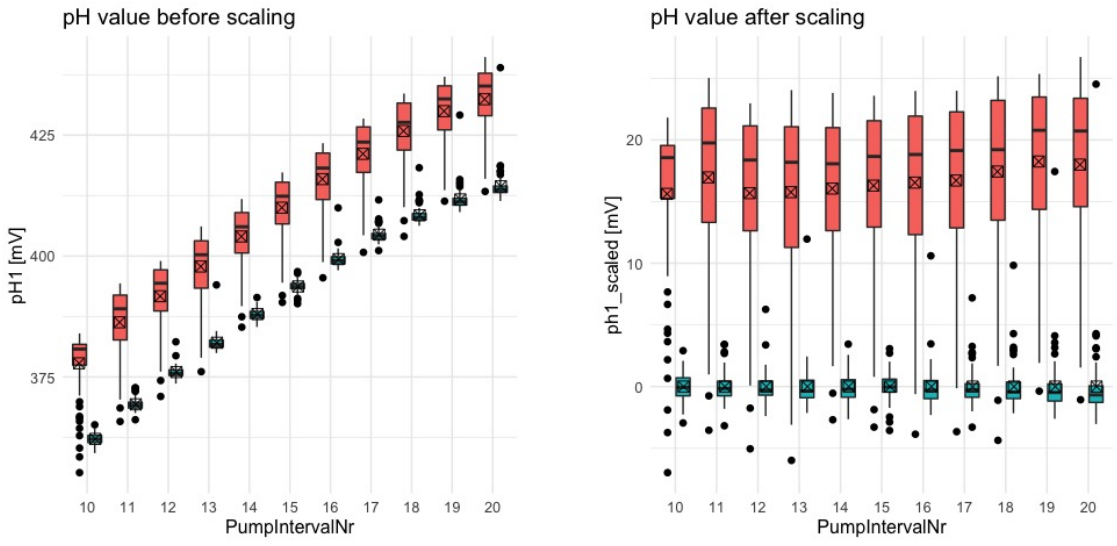


Figure 2.1: pH scaling to solve the sensor drift problem

The raw data obtained cannot be used directly for making conclusions about the experiment or for certain analysis procedures. Firstly, the whole data received might not be relevant for analysis. The data corresponding to the valid experiment interval (i.e, 24hrs) is cut off from the original data.

Secondly, the different sensors record values in mV and furthermore are not calibrated. For easier interpretation we re-scale the temperatures to 37 °C ten hours after the start of the experiment, since the incubator containing the IMOLAs is regulated to maintain human-body temperature.

Finally, the pH sensors very often record different measurement ranges and a drifting signal, i.e. the absolute measurement values increase or decrease constantly during the time which cannot be attributed to real changes. A typical example for this undesired behavior is depicted in Figure 2.1 (The plots show the development of measurements from pH sensor 1 for an exemplary series of intervals. Red boxplots correspond to the rest phase and blue boxplots to the pump phase). The second figure shows that re-scaling helps to solve the sensor drift problem The pH value of the fresh culture medium which is pumped into the system after every rest phase is known and constant. To make the pH values comparable across intervals, we normalize them by subtracting the mean and dividing by the standard deviation of the corresponding.

3 Data Analysis

As described earlier, one goal is to develop a module that provides a statement about the data quality. The pre-processed data needs to be analyzed to understand how the cell culture reacts to the medium being added to it during the different phases of the experiment. Numerous approaches were used to achieve this. These include examining the trend of the data with respect to time (which can further define the points during which the cell culture was treated differently by various media), analyzing the pattern that the data follows, extracting the periodic behavior from the data and filtering the data to removing noise. This chapter describes these approaches in detail.

3.1 State of the Art Approaches and Algorithms

3.1.1 Linear Regression and Trend Estimation

Linear regression models a relationship between a dependent variable and one or more independent variables. Piece-wise linear regression is a form of regression that allows multiple linear models to be fitted to the data. One of the major applications of Linear Regression is to estimate the trend shown by the data. When a series of measurements of a process is treated as a time series, trend estimation can be used to make and justify statements about tendencies in the data, by relating the measurements to the times at which they occurred. This model can then be used to describe the trend of the observed data across time.

3.1.2 Functional Data Analysis

Functional data analysis (FDA) is a branch of statistics that analyses data providing information about curves, surfaces or anything else varying over a continuum by interpreting the data as being generated by a function. Broadly interpreted, FDA deals with the analysis and theory of data that are in the form of functions.

Functional Data Generation

When we have a finite set of measurements $y_1, y_2, y_3 ... y_n$ the first task is to convert these into functions of x with values $x(t)$ computable for any desired argument t. If these observations are error-less, then the process is simple *interpolation*, but if they have some observational error which needs to be removed, then the conversion from finite

data points to functions involve *smoothing*. The functions are build basically in two steps:

- Defining a set of functional building blocks called *Basis Functions*.
- Setting up a vector, a matrix or an array of coefficients to define the functions as a linear combination of these basis functions.

The functions that are to be approximated fall into two categories: *periodic* or *non-periodic*. For periodic functions, a *Fourier basis functions* is used, whereas for non periodic functions *Spline basis functions* is used.

A set of functional blocks $\phi_k, k = 1, 2, ..., K$ called *basis functions* are used to define the function which we wish to estimate. Mathematically, the function $x(t)$ is a linear combination of these basis functions ϕ_k.

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \mathbf{c}'\phi(t)$$

This expression is called the *basis function expansion*. The parameters $c_1, c_2, ...c_k$ are the coefficients of the expansion. We use spline basis functions, which are basically piecewise polynomials. Splines can be constructed by dividing the interval of observations into sub intervals with boundaries at points called *break points*. Over any sub-interval the spline function is a polynomial of fixed degree or order, but the nature of the polynomial changes from one sub interval to another. More detailed information regarding the functional data analysis procedure can be found in (5).

Clustering of Functional Data

Clustering is an unsupervised learning process that aims to group a set of data objects into clusters or groups, such that data objects within the same cluster are more similar than to that across clusters based on some metric. Functional data clustering treats this data objects as functions or curves. i.e, Functional data clustering aims at clustering data that has similar functional behaviour (or curves). In the k-means clustering method, the basic idea hinges on cluster centers, the means for the clusters. The cluster centers are established through algorithms aiming to partition the observations into k clusters such that the within-cluster sum of squared distances, centering around the means, is minimized.

3.1.3 Discrete Fourier Transformation

The idea of Fourier Transformation is to convert a signal from its original domain to a frequency domain and vice versa. The Fourier transform (FT) decomposes a function of time (a signal) into its constituent frequencies. The Fourier transform is itself a complex valued function of frequency whose "magnitude" (modulus) represents the amount of frequency that is present in the original signal (function) and whose "argument" are

the phase of the basic sinusoid in that frequency. The Discrete Fourier Transformation convert a sequence of samples x_t, with $t = 0, ..., N-1$ into a sequence of complex numbers:

$$X_k = \sum_{t=0}^{N-1} x_t \exp\left(-i\frac{2\pi}{N}tk \right).$$

N is the number of samples, x_t the current sample and $\frac{2\pi k}{N}$ the circular frequencies. The modulus of the X_k is the amplitude of that signal with frequency k, and the argument of X_k is the phase of the sinus curve. The Inverse Discrete Fourier Transformation is the opponent from above, its converts a signal from the frequency domain back into the time domain:

$$x_t = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp\left(i\frac{2\pi}{N}tk \right).$$

The Inverse Fourier transformation is often used for applying filters, e.g. setting unwanted frequencies to zero. This is helpful for a further analysis of a specific frequency or frequency band or reducing the noise of the signals. For these purposes the frequencies corresponding to the noise are cleared out and afterwards the Inverse Fourier transform is applied to the signal. Taking the real part of the filtered values will result in more or less noise-free data, when filtering out the right frequencies. More about Fourier transformation and Inverse Fourier transformation can be found in (8).

3.2 Linear Regression for Slope Estimation

In order to assess the changes in the cellular activity of the cell cultures, the development of the parameters pH, temperature and oxygen over time is of huge interest. Each IMOLA has several time intervals, where one interval corresponds to one pump off phase (rest) followed by a pump on phase. We are interested in examining the trend shown by the data from the cell culture in each of these intervals. Experts in microphysiometry expect this data to show some patterns as described later in section 3.3. Thus, it is a fundamental approach (as mentioned in section 3.1.1) to fit piece-wise linear regression models and then examine the linear trends followed by the data across the different intervals. Based on the output from the Bubble script (section 2.1.1), the module Gradient therefore estimates a linear regression of each of these parameters against the time, for each rest cycle separately. The slope of each of these regressions is then produced as the output data frame. The estimation of the slope is also useful in terms of estimating the EAR as mentioned previously.

3.3 Validation of the Data

Once we get the normalized pre-processed data (which is also checked with the open circuit logic if the experiment had cell culture), the first step is to identify valid intervals

that can be used to make decisions about the cell reactions in the entire experiment. This is important since there could data which is disturbed by noise from the sensors, or data that shows unreliable sensor values because of surrounding disturbances like air bubbles produced by the pump.

As mentioned earlier, for each IMOLA we have several intervals, one interval corresponds to one pump off phase (rest) followed by a pump on phase. A 24 hour experiment with an interval size of 10 minutes results in 144 intervals. The goal is classifying the intervals into valid and invalid ones. When the number of invalid intervals is more than half of all intervals of one IMOLA, the whole IMOLA data is declared invalid.

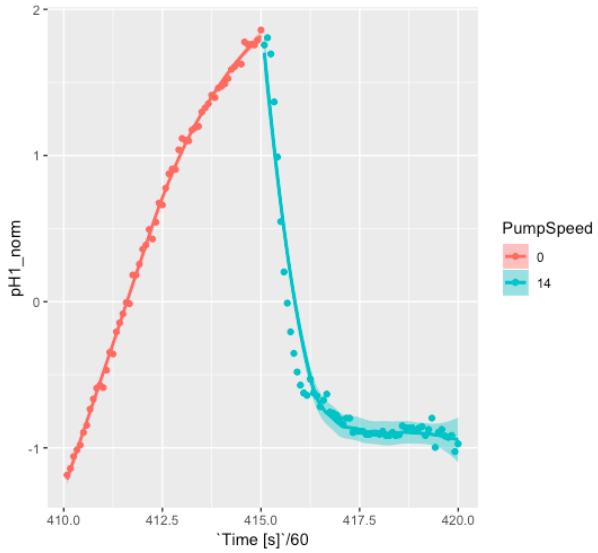
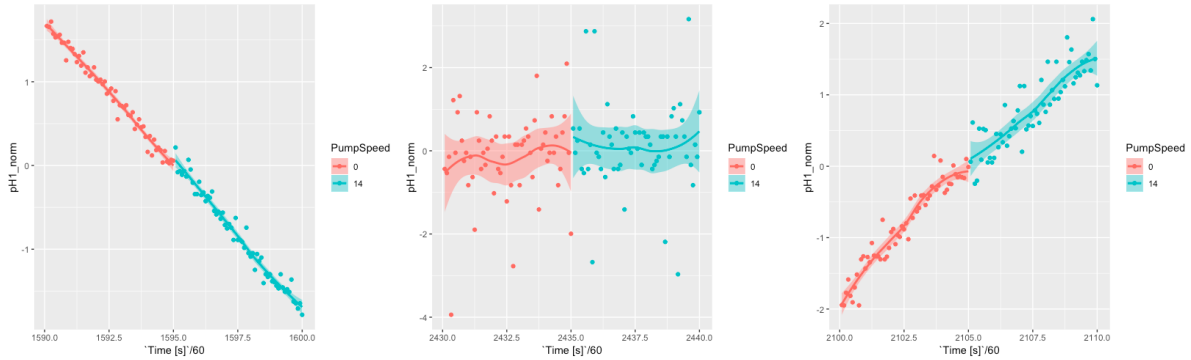


Figure 3.1: Valid pH curve

A valid pH curve looks more like a shark fin curve as shown in Figure 3.1. The red region corresponds to the rest phase where the pH values of the cells are expected to increase. The blue region corresponds to the pump on phase where a medium is pumped into the cell culture and the pH values are expected to first decrease and then stay constant. However, from the data we could also see many other pH curves which were invalid and could not be used for the further analysis procedure. Some examples of these curves are shown in Figure 3.2.

To ensure data quality and ensure that the experiment setup (e.g. sensors, bubble detectors) work correctly, we created two different ways to validate the data. The first approach based on manual criteria which we define based on a set of criteria that was build after having a close look on the data. The other is base on the ideas of functional data analysis.



(a) Decreasing pH **(b)** Constant pH **(c)** Increasing pH

Figure 3.2: Invalid pH curves

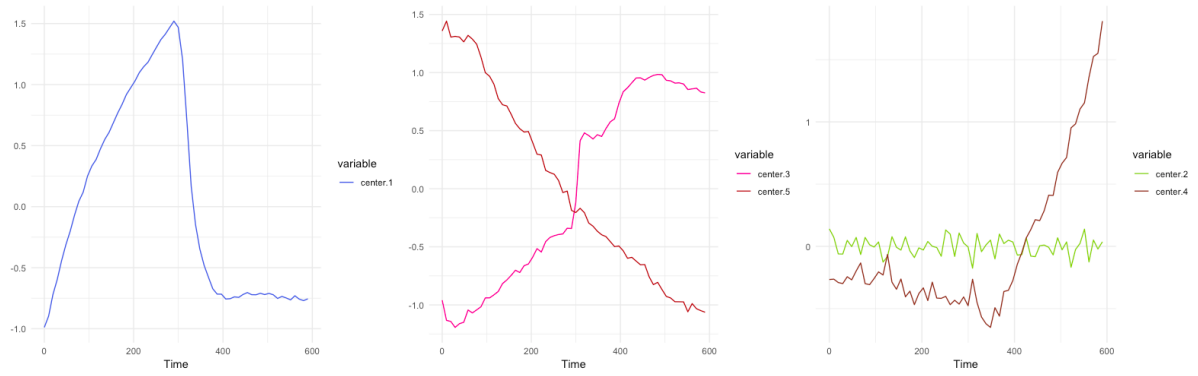
3.3.1 Criterion-based Validation

For the criteria based validation we consider the normalized pH values and based on whether an interval satisfies a set of pre-determined criterion, it is classified as being valid or not.

This set of pre-determined criterion was developed manually after looking into the data and observing the different invalid patterns that the data exhibits. From Figure 3.2 we see the mostly observed invalid pH curves. The pH curves are not expected to show a strictly increasing/constant/decreasing behavior throughout an interval. This led to development of the 3 criterion based on which we decide if an interval is valid or not. These are given below:

- **Negative pH slope** during rest phase of an interval: The pH during the rest phase of an interval is always expected to increase. Hence eliminating all the intervals with a decreasing pH during the rest phase removes a set of invalid intervals. Curves similar to those in Figure 3.2a get eliminated by this criteria.
- **High Mean Squared Error (MSE)** during an interval: The pH during an interval is not expected to remain constant. The constant pH curve might be a result of data with high variance or a sign for dead cells. This can be captured by a criteria which checks for a high mean squared error. Curves similar to those in Figure 3.2b get eliminated by this criteria.
- **Mean pH of pump phase higher than 75th percentile of rest phase** during an interval: The pH during an interval is always expected to be lesser than the corresponding rest phase. However we see pH curves like those in 3.2c where this is not true. To eliminate such intervals we require that the mean pH of the pump phase should be lesser than the 75th percentile of the rest phase.

Once we classified each interval as valid or invalid, the next task is to decide whether an IMOLA corresponds to valid cell culture data or not. This decision is then made de-



(a) **(b)** **(c)**

Figure 3.3: Different cluster patterns identified by the k-means clustering algorithm

Figure 3.3a shows a valid cluster corresponding to valid pH curves like Figure 3.1. 3.3b corresponds to clusters that summarizes the increasing/decreasing invalid pH curves like those in Figure 3.2..
3.3c identifies the constant pH curves and those with unexpected higher pump phase pH values than the corresponding rest phase pH values

pending on whether at least half of the intervals corresponding to the respective IMOLA is valid or not. If more than half of the intervals corresponding to an IMOLA is invalid, then the data from the IMOLA is not used for further analysis procedure.

3.3.2 Validation based on Clustering of the Functional Data

From section 3.3.1 it is clear that the validation of an interval relies on the shape of the pH curve during this interval. This led to the implementation of the clustering techniques described in section 3.1.2. For this approach we employ the R-package fda and fda.usc. For performing the k-means clustering the function kmeans.fd was used.

The approach of functional data analysis is very much beneficial to this project in terms of identifying the pH sensor data as functions of time. The pH curves are expected to show a particular pattern with respect to the time over which the pumping of different materials into the cell culture takes place.

The pH data for each interval was modelled as functional data using spline basis functions. The functional data is then used by a clustering algorithm to group together the similar pH curves. After trying iterations over different number of clusters, this parameter was decided to be 5, since this number of clusters could easily capture the different patterns observed in the data. An overview of the results from the clustering algorithm is shown in Figure 3.3. According to these generated clusters, every pH curve which belongs to cluster 1 (Figure 3.3a) can be labelled as valid and all the others as invalid.

Once the cluster patterns identify all the different patterns exhibited by the data, for each new dataset, it is only necessary to find for each interval its closest cluster center. When a interval has pH curve which is closest to the first cluster (Figure 3.3a)(in terms of the metric L2 distance between the functional data) than to any other cluster, it can be then labelled as valid and all others as invalid.

3.4 Fourier Transformation and Filtering

As described in section 3.1.3, Fourier transformation converts a signal from its original domain (time in seconds) to a frequency domain and vice versa. The Fourier transformation of the pH values could produce interesting results which showed the periodic behaviour of the data. The frequency spectra were later used to filter the data as described in this section. Fig 3.4 shows the frequency domain of the pH values for one IMOLA.

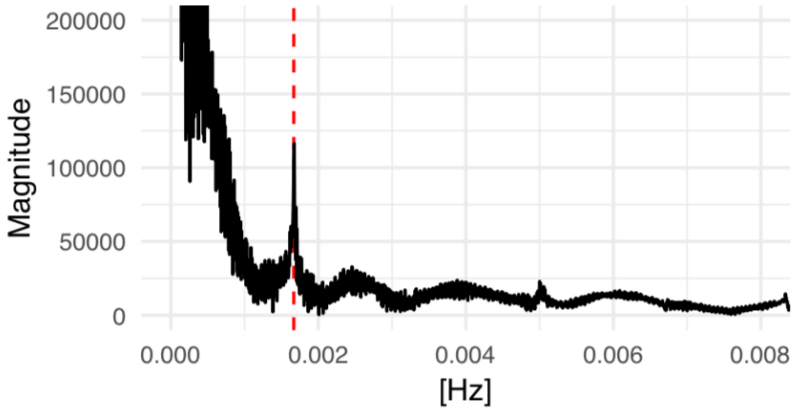
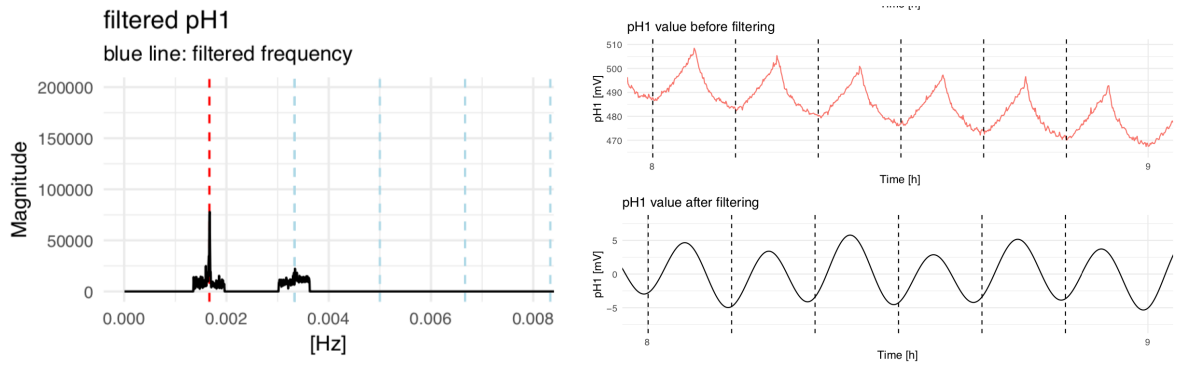


Figure 3.4: Frequency domain for pH for one IMOLA

The red line corresponds to the frequency $\frac{1}{600} Hz = 1.667$ *mHz. This shows that there is a periodic behaviour in the signal, which repeats itself after 600 sec (corresponding to the interval length of 600s).*

Filtering

Since we know that all important microphysiometric information must be stored in the frequency domain of the pump frequency (see peaks in figure 3.4), we can apply several filters to reduce the noise of the signal. One approach is to filter around the frequencies of the pump frequency. In Figure 3.5a, an exemplary bandwidth $1.36-1.96$ mHz and two harmonics were used. Two harmonics with same bandwidth means that all frequencies except $[1.367, 1.997]$ mHz and $[1.367 + 1.667, 1.997 + 1.667]$ mHz $= [3.033667, 3.663667]$ mHz will be deleted.



(a) Frequency domain plot of Filter for the pH values

(b) pH curves before (red) and after (black) filtering

Figure 3.5: Used filter and the filtered pH curves

Applying this filter to the frequency domain and calculating the inverse transformation led to results shown in Figure 3.5b. The filtered curve is smoother than the original, since for the filtered curve only a small part of the frequency domain is used. The low frequencies ($\leq 1.6667 - \frac{\text{bandwidth}}{2}$ mHz) have a high magnitude, which corresponds to sine curves with high amplitude (large values). After deleting those frequencies we see that the values of the filtered pH data ($[-4, 4]$) are much smaller compared to the raw pH values ($[470, 510]$). Furthermore, the drift of the pH values is eliminated (see 3.5b).

4 Results

This section describes the results achieved throughout the duration of this project and their evaluation. The main tasks involved improving the efficiency of existing algorithms, implementing new modules and increasing the accuracy of the algorithms against the evaluations.

4.1 Innovations and Improvements of Algorithms

At the initial stage of the project, cellasys had numerous R scripts which did different data pre-processing tasks. The performance of these scripts could be improved drastically. These basic pre-processing scripts like Module Bubble had a runtime of around 30 minutes to prepare the data from a single experiment. The initial steps of this project involved recreating these scripts to achieve a faster execution of the bubble script. The new version is almost 10 times faster and finishes execution in around 3 minutes.

Module	Runtime (approx.)
Bubble (old)	1800 seconds
Bubble (new)	180 seconds
Bubble_for_Cluster	60 seconds
Bubble_for_Online	6 seconds

Table 4.1: Runtime comparisons of newly developed Bubble modules against the old Module Bubble (for 24h experiment's data)

The new Bubble module already uses some parallelization of the code, but is limited to using a maximum of two cores, since it should be able to run on standard computers. If larger input files need to be assessed, the execution consumes a significant amount of time. Therefore we implemented an alternative Bubble module, called *Bubble_for_cluster.R*, which uses up to twelve cores. This bubble script implemented on the LRZ cluster is 30 times faster than the existing bubble module and hence finishes execution in around 1 minute. The input and output formats of this module are the same as the ones of the original Bubble script.

Sometimes not all of the information provided by the Bubble module is needed. Therefore, a downgraded version of the original script, called *Bubble_for_Online.R*, was developed. This module performs the same tasks of the original bubble script, but collects

only information regarding the IMOLA measurements (ignoring data about bubble detection's, pump configuration etc). This decreases the runtime of the module heavily, and may be used to display the measurements in parallel with the reception of the sensor readings. The performance comparison of these modules is given in Table 4.1.

4.2 Module-based Implementations

During the course of the project we decided to recreate the existing R modules from cellasys corresponding to the workflow of data processing and analysis steps. This makes it easier to maintain and developed existing and new code. This section describes the modules that were created, their functionality with respect to the previously defined data preparation and analysis chapters and the nature of the results they produce.

Based on an input experiment file (*.exp* format) containing the raw data from the cells (provided by the DALiA client), the initial module Bubble creates structured data which can then be used as input for further modules which perform more advanced validation and analysis tasks. Figure 4.1 depicts the data processing flow with all modules. In the following short descriptions of the modules' functionality are provided.

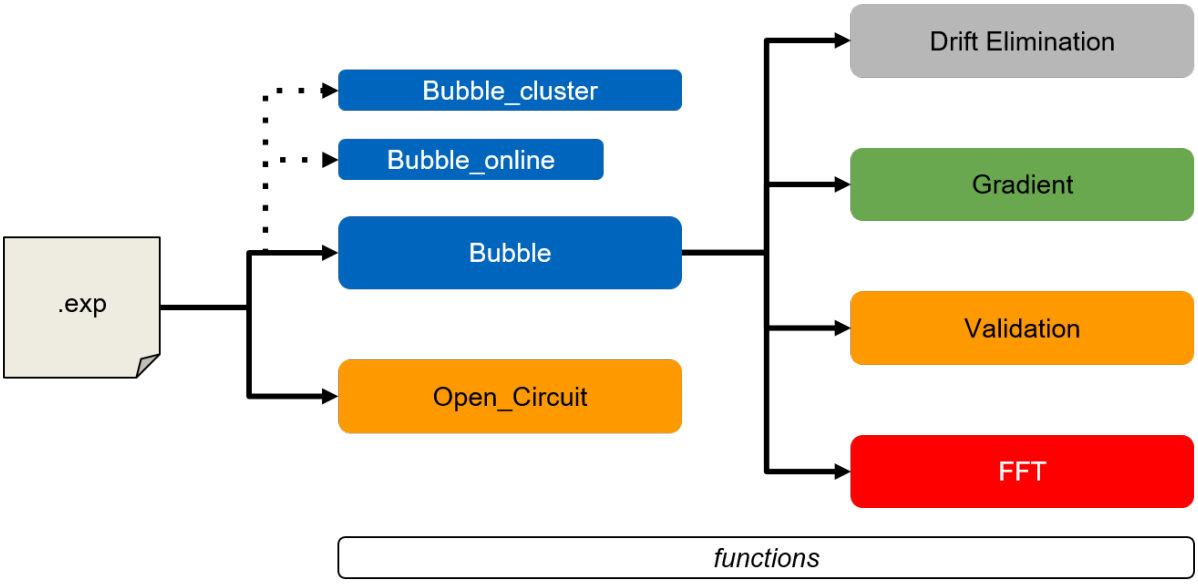


Figure 4.1: Process flow of the modules

1. **Auxiliary Script Functions**
   Some of the analysis tasks need to be performed in more than one script. Therefore we created an R script *functions* which makes these code snippets available for all modules.

2. **Module Bubble**
   This is the initial module that takes in the raw input data from the microphysiometric system and prepares the data. According to the requirement specifications, three different versions of this script were created as described in section 4.1.

3. **Module Open Circuit** This module checks the ranges of the sensor values to identify if the microphysiometric system had a cell culture inside during the experiment as described in Section 2.3.1. The output of this module describes if the data from each IMOLA shows ranges corresponding to an open circuit experiment, a dummy Bio-Chip or a living cell culture.

4. **Module Drift Elimination** This module prepares the data to facilitate the comparison of the data values which are originally in different scales. The output file is an extension of the original input .dat files with additional information regarding the drift-eliminated data values.

5. **Module Gradient** This module fits a linear regression model on the sensor values with respect to time and determines the slope of this regression line to facilitate further analysis procedures as described in section 3.2. The output of this module is an extension of its input file with additional columns for the determined slopes.

6. **Module Validation** This module validates each interval of an experiment to qualify it for potential conclusions. The validation procedure is described in section 3.3. The output of this module are .dat files that state whether the corresponding IMOLAs are valid or invalid.

7. **Module FFT** This module finds the Fourier transformation of the pH data and applies filters to eliminate noise as described in section 3.1.3. The output of this module are .dat files containing the filtered pH data.

4.3 Validation Results

The validation as described in section 3.3 involves deciding if each interval and finally if each IMOLA is valid or not. The two techniques employed were the criterion-based validation and the functional data analysis based clustering approach. The achieved results were compared against the results provided by experts. This evaluation for two different experiments is shown in figure 4.2 and figure 4.3.

In both figures, part (a) has Column Y which corresponds to the number of valid intervals and column N that corresponds to the number of invalid intervals and in part (b), Column 1 through 5 corresponds to the number of intervals belonging to the corresponding clusters, where Cluster 1 is considered valid and all others invalid.

ImolaNr	Y	N	valid/Imola [%]
1	100	15	86.956522
2	107	36	74.825175
3	130	13	90.909091
4	110	33	76.923077
5	6	137	4.195804
6	27	116	18.881119

(a) Criterion-based validation (Y = valid and N = invalid)

ImolaNr	1	2	3	4	5	valid/Imola [%]
1	99	3	5	1	7	86.086957
2	107	1	29	5	1	74.825175
3	129	2	7	3	2	90.209790
4	111	0	29	2	1	77.622378
5	5	27	58	7	46	3.496503
6	16	73	21	12	21	11.188811

(b) Clustering-based validation (Cluster 1 = valid, Cluster 2-5 = invalid)

Figure 4.2: Experiment 1 evaluation

Expected Results: IMOLA 5 and 6 are invalid, all others are valid.
Obtained Results: IMOLA 5 and 6 are invalid, all others are valid considering a 50% benchmark for the number of valid intervals.

ImolaNr	Y	N	valid/Imola [%]
1	62	75	45.255474
2	137	6	95.804196
3	135	8	94.405594
4	35	108	24.475524
5	137	6	95.804196
6	5	126	3.816794

(a) Criterion-based validation (Y = valid and N = invalid)

ImolaNr	1	2	3	4	5	valid/Imola [%]
1	42	31	20	7	37	30.656934
2	132	3	5	2	1	92.307692
3	123	14	1	1	4	86.013986
4	11	124	0	5	3	7.692308
5	134	2	3	1	3	93.706294
6	0	15	3	4	109	0.000000

(b) Clustering-based validation (Cluster 1 = valid, Cluster 2-5 = invalid)

Figure 4.3: Experiment 2 evaluation

Expected Results: IMOLA 1, 4 and 6 are invalid, all others are valid.
Obtained Results: IMOLA 1, 4 and 6 are invalid, all others are valid considering a 50% benchmark on the number of valid intervals.

4.4 Noise Elimination

The noise elimination of signals was achieved through the Fast Fourier transformation as described in section 3.1.3. This helped to smoothen the pH curves used for analysis, for example to analyze the toxicity of a test material. The curves before and after filtering can be seen in Figure 4.6

In order to find the best filter for the experiment, we select data from tje IMOLA with the most valid intervals, based on our clustering validation. The corresponding frequency domain plot is show in Figure 4.4.

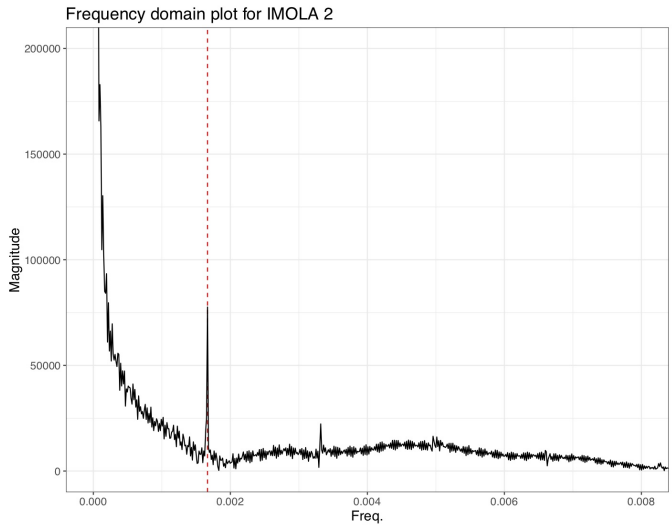


Figure 4.4: Frequency domain plot for IMOLA 2 (Experiment 2)

We varied the number of harmonics from 1 to 10 and the bandwidth from 0.1 mHz to 0.83 mHz with a stepsize of 0.02 mHz. Our starting point is always the frequency at $\frac{1}{600}$ Hz $= 1.667$ mHz which corresponds to the pump frequency and hence the highest peak. When using one harmonic and a bandwidth of 0.3 mHz, we will delete all frequencies except $1.667 \text{ mHz} \pm 0.3 \text{ mHz} = [1.367, 1.997]$ mHz. For two harmonics and the same bandwidth we will delete all frequencies except $[1.367, 1.997]$ mHz and $[1.367 + 1.667, 1.997 + 1.667]$ mHz $= [3.033667, 3.663667]$ mHz, this kind of filter is shown in section 3.4.

Varying over the 42 different bandwidths and 10 different number of harmonics led to 420 different filters. To evaluate the filters we decided to calculate the slope for each interval after the filtering and compare it to the original slope. We want to guarantee that the shape of the curve does not change heavily (since we only want to reduce the noise in the data). A drastic change of the slope implies that we filtered out frequencies that actually correspond to the information from cellular vitality, which is undesired of course.

To get comparable slopes we normalize the slopes per filter (i.e, subtract the mean and divide by the standard deviation), which we also did with the raw data. With the normalized slopes we can calculate the difference between the raw and the filtered slope. To get a measure per filter, we calculated the mean and median difference of each filter. The results are shown in Figure 4.5

From 4.5 we can conclude that using **one harmonic and bandwidth 0.15 mHz** (corresponding to number 3 in the legend) led to the best results. Applying the inverse Fourier transformation led to the results shown in Figure 4.6, which depicts an exemplary

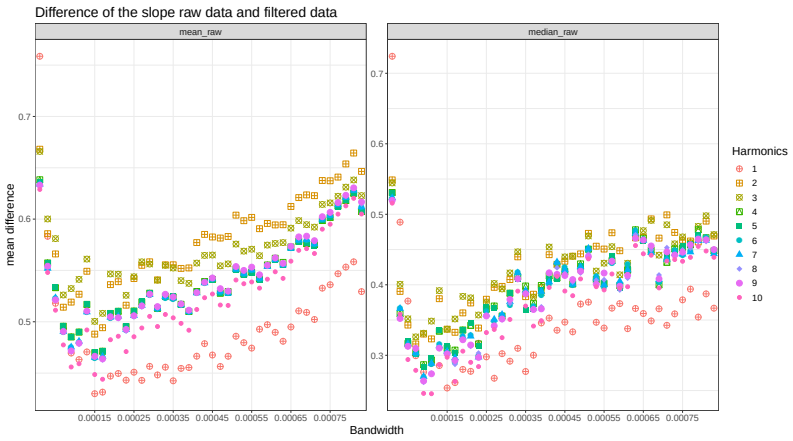


Figure 4.5: Different filter combinations

Left: mean difference for each number of harmonics and bandwidth combination. Right: median difference for each number of harmonics and bandwidth combination. The different colours corresponds to the different number of harmonics.

plot of the raw pH values and filtered pH values for a period of two hours.

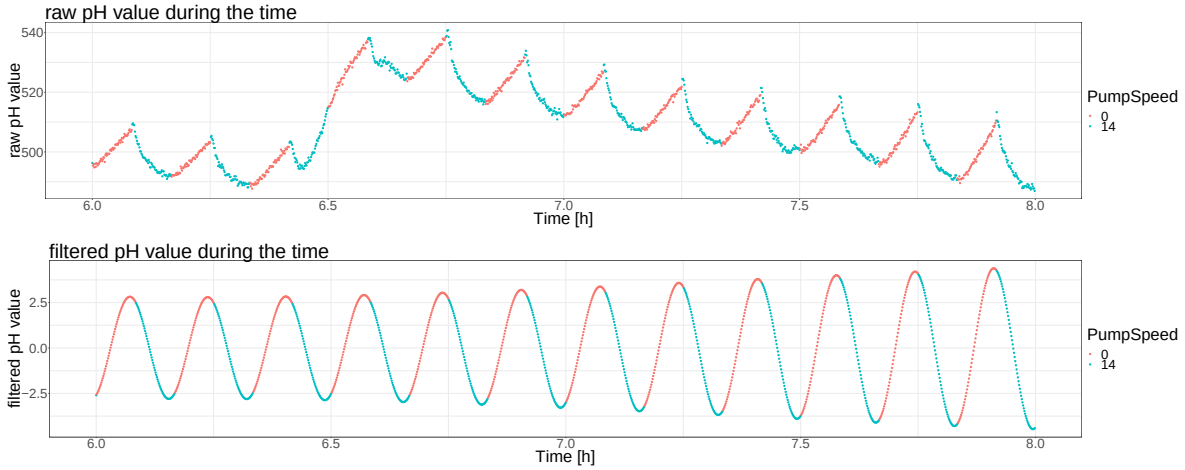


Figure 4.6: Raw data vs. filtered data for a period of 2h

5 Conclusion

The system from cellasys measures different parameters from a living cell culture like extracellular acidification rate (measured with pH sensors), cellular respiration (oxygen) and morphology (impedance). The information about cellular metabolic activity contained by measured sensor data dynamics is superimposed by manifold sources of error. A very careful consideration of data processing is necessary to eliminate these errors as much as possible and to facilitate the data for a correct interpretation about the cellular activities. During this project, we could device techniques to organize and prepare this raw data for evaluation, improve the performance of existing algorithms by a large scale and use different validation techniques to validate this data for further interpretation. We also refine the data by filtering the data from noisy signals. We achieved the goal of making the raw data suitable for correct interpretations about cellular activities for applications like assessing toxicity.

Bibliography

[1] Brischwein, M., Wiest, J. *Microphysiometry.* Bioanalytical Reviews, Springer, 2018, doi:10.1007/11663_2018_2

[2] Wiest, J., Namias, A., Pfister, C., Wolf, P., Demmel, F., Brischwein, M. *Data processing in cellular microphysiometry.* IEEE Transactions on Biomedical Engineering, 2016, 63/11, 2368-2375, doi:10.1109/TBME.2016.2533868.

[3] Eggert, S., Alexander, F.A., Wiest, J. *An automated microphysiological assay for toxicity evaluation.* 37th Annual International Conference of the IEEE EMBS, Milano, Italy, 2015, 2175-2178, doi:10.1109/EMBC.2015.7318821.

[7] Wiest, J. *Fourier Analysis in Microphysiometry.* Advances in Medicine and Biology 136, Nova Science Publisher, Inc., 2019, ISBN: 978-1-53613-722-3.

[5] J.O Ramsay,Giles Hooker, Spencer Graves *Functional Data Analysis with R and MATLAB.* Springer, 2009, ISBN: 978-0-38798-184-0

[6] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller, *Functional Data Analysis.* Annual Review of Statistics and Its Application, 2016, 3/1, 257-295, doi:10.1146/annurev-statistics-041715-033624.

[7] OriginLab. *Fast Fourier Transforms*
https://www.originlab.com/doc/Origin-Help/FFT, Accessed: 2019-07-10

[8] João Neto. *Fourier Transform: A R Tutorial*
http://www.di.fc.ul.pt/ jpn/r/fourier/fourier.html, Accessed: 2019-07-01