Aus dem Institut für Medizinische Informatik
der Hochschule Mannheim und
dem Institut für Medizintechnologie
der Universität Heidelberg und
der Hochschule Mannheim

New tools for functional and morphological data analysis and quality management in biological research

Inauguraldissertation
zur Erlangung des akademischen Grades
Doctor scientiarum humanarum (Dr. sc. hum.)
der
Medizinischen Fakultät Mannheim
der Ruprecht-Karls-Universität
zu
Heidelberg

vorgelegt von
Angelo Torelli

aus
Barquisimeto, Venezuela
2023

Dekan: Prof. Dr. med. Sergij Goerdt

Referent: Prof. Dr. Ivo Wolf

Contents

1 Introduction **1**
- 1.1 Aim of the study . . . 3
- 1.2 Structure of the Thesis . . . 4

2 Functional data analysis **5**
- 2.1 Background . . . 5
- 2.2 Related Work . . . 8
  - 2.2.1 GFR Marker . . . 9
  - 2.2.2 Compartment Model . . . 12
- 2.3 Material & Methods . . . 15
  - 2.3.1 Device, marker and animal preparation . . . 15
  - 2.3.2 Data . . . 17
  - 2.3.3 GFRmeasure . . . 18
  - 2.3.4 GFRmeasure+ . . . 20
- 2.4 Results . . . 29
  - 2.4.1 Evaluation of GFRmeasure+ automation . . . 29
  - 2.4.2 Evaluation of available data . . . 31
  - 2.4.3 Comparison with other solutions . . . 33
- 2.5 Discussion . . . 36

3 Morphological data analysis **39**
- 3.1 Background . . . 39
- 3.2 Related Work . . . 41
- 3.3 Material & Methods . . . 45
  - 3.3.1 Datasets . . . 45
  - 3.3.2 Overview of AutoCellSeg . . . 50
  - 3.3.3 Image Analysis Pipeline . . . 52
  - 3.3.4 Pipeline for Quality Management . . . 62
- 3.4 Results . . . 66
  - 3.4.1 Evaluation Methodology . . . 66
  - 3.4.2 Dark background benchmark . . . 68

3.4.3 Inverted background benchmark 71
3.4.4 Batch benchmark 75
3.4.5 Fluorescent cell segmentation 77
3.4.6 Single images benchmark 79
3.4.7 Control/ test analysis 80
3.4.8 CFU growth over time 83
3.5 Discussion 85

4 Summary **89**

Curriculum Vitae **103**

1

Introduction

Data analysis plays a crucial role in biological research, enabling scientists to extract meaningful insights from complex datasets. Understanding the dynamic behavior and structural characteristics of biological systems is essential for uncovering underlying mechanisms, identifying biomarkers, and developing new treatments.

In recent years, advancements in data acquisition technologies have revolutionized the way biological research is conducted. Moreover, the increasing availability of sophisticated measurement technologies, such as high-throughput sequencing, imaging techniques, and wearable sensors, has led to the generation of large-scale datasets capturing the dynamic behavior and structural characteristics of biological systems. This wealth of data presents both opportunities and challenges for researchers seeking to extract meaningful information and derive actionable insights.

Throughout the collaborative doctoral program titled ”Tissue analytics for stem cell-based diagnostics”, a significant volume of data was generated across multiple projects. The diverse projects undertaken as part of this doctoral program involved the collection and characterization of various types of biological data, including molecular profiles, temporal data, imaging data and clinical information. These datasets were crucial for investigating the potential of different types of cells in the respective projects.

Initially, manual analysis was conducted, involving meticulous examination and interpretation of the data by domain experts. This process allowed for detailed scrutiny of the information, identification of patterns, and extraction of meaningful insights. However, due to the large volume of data generated, manual analysis alone proved to be error-prone, time-consuming and resource-intensive.

To address these challenges, programmatically-driven analysis techniques were employed. This involved the development and application of computational algorithms, statistical models, and machine learning methods to process and analyze the vast datasets. By leveraging the power of automation and advanced data analysis tools, the research team was able to extract valuable information from the data more efficiently and effectively.

Furthermore, functional and morphological data analysis techniques are vital for understanding biological processes at different scales, from the molecular level to complex organisms. Functional data analysis encompasses a range of statistical and computational techniques designed to analyze data that vary over time. It involves the exploration, modeling, and interpretation of complex patterns embedded within dynamic biological processes. For instance, functional data analysis can be applied to study gene expression profiles over time, brain activity captured by functional magnetic resonance imaging, or like in the case of this thesis the filtration rate of glomeruli in the kidneys. By capturing the temporal dependencies within the data, functional data analysis provides valuable insights into the underlying mechanisms governing physiological processes, disease progression, and their responses to various stimuli or interventions.

Morphological data analysis, on the other hand, focuses on the study of shape, structure, and form within biological systems. It involves techniques for quantifying and characterizing various morphological features, ranging from cellular and tissue structures to whole organ shapes. Advances in imaging technologies, such as confocal microscopy, electron microscopy, and medical imaging modalities, have facilitated the acquisition of high-resolution morphological data. Through morphological data analysis, researchers can investigate cellular processes, embryonic development, disease pathology, and structural abnormalities, providing valuable insights into the functioning and organization of biological systems. The focus of morphological data analysis for this work is quantification of colony forming unit (CFU).

Both functional and morphological data analysis techniques enable researchers to identify biomarkers, detect abnormalities, classify samples, predict outcomes, and guide therapeutic interventions. Furthermore, it requires interdisciplinary collaborations, bringing together expertise from fields such as biology, statistics, computer science, and engineering.

The analysis of functional and morphological data poses various challenges. These include dealing with high dimensionality, handling noisy or missing data, accounting for dependencies within the data, ensuring data quality and reliability, and developing appropriate statistical models and computational algorithms. Overcoming these challenges requires the development of innovative methodologies, statistical frameworks, and computational tools tailored to the specific characteristics of the data and the biological questions being addressed.

1.1 Aim of the study

The aim of this thesis, is to contribute to the doctoral program by facilitating researchers to analyze data by introducing novel tools, techniques, and quality management approaches.

More specifically the aim is to address data analysis challenges associated with:

- Transcutaneous measurement of glomerular filtration rate, which allows for the animal to stay conscious during the non-intrusive procedure and avoids the time consuming serial blood and urine sample collection previously required. The experimenter is assisted by calculating the elimination kinetics curve after the data is collected.
- Counting and characterization of colony forming units (CFUs), in order to replace the manual counting with automatic petri dish extraction and detection of each individual

CFU as well as the possibility for correction, plotting and managing of the results. Quality management in form of tracking over time in respect to the growth of each individual CFU in order to quantify the quality of the experiment.

The aim is to advance the understanding of biological systems by automating the evaluation of large amounts of data to a greater extent, to make evaluation possible in the first place or at least to make it more reliable compared to manual evaluation and therefore extracting meaningful insights from complex datasets.

1.2 Structure of the Thesis

This thesis will be structured in two main chapters for each aim it tries to accomplish. These are then subdivided in background, related work, material & methods, results and discussion. Chapter 2 called ”Functional data analysis” deals with one-dimensional data over time, more specifically glomeruli filtration rate data. While chapter 3 named ”Morphological data analysis” targets the analysis of color images of petri dishes containing colony forming units. The thesis concludes with a summary of all projects and possible future developments.

2

Functional data analysis

This chapter will concentrate on the problem of kidney function measurements and part of it has already been published [1]. The chapter background illustrates the principles and the importance of the subject. Related works including methods and programs used to achieve similar goals are laid out in the next chapter. Afterwards follows the Material and Methods chapter, where the experiment setup, the data, the devices as well as the methodology and the resulting program called GFRmeasure are described in detail. A comparison between existing programs used by other facilities and GFRmeasure is presented and discussed in the last two chapters.

2.1 Background

Chronic kidney failure, most commonly known as chronic kidney disease (CKD), is an increasing worldwide public health issue [2]. Prevalence is estimated to be $>$10% worldwide [3]. Complications linked with early-stage CKD include increased all-cause and cardiovascular mortality, kidney-disease progression, acute kidney injury, cognitive decline, anemia, mineral and bone disorders and fractures [4]. This is problematic since symptoms of CKD are unspecific and remain often undetected until a late stage. Early kidney dysfunction diagnosis and treatment would increase the odds of preventing the need of dialysis and preserving the remaining healthy nephrons as well as reducing cardiovascular-related outcomes. Unfortunately, awareness of the disorder, however, remains low in many communities and among many physicians and patients [5].

Worldwide, diabetes mellitus and hypertension are the most common cause of chronic kidney disease [6], but in some regions, other causes, such as herbal and environmental toxins, are more common [7]. Since 2016, CKD has risen rapidly to be ranked 11th among the leading causes of death globally [6]. The economic impact of CKD is global and tangible and the prevention of CKD progression into later stages, and therefore delaying or avoiding the requirement for dialysis and transplants, can save patients and health systems a substantial fortune [8]. This would also mean a significantly higher level of quality of life for the patients.

Most patients with CKD have comorbidities that are susceptible to a methodical approach, which helps with prevention and early management [9]. Most common preventing measures for kidney damage for patient with comorbidities include monitoring and lowering blood pressure [10] and heathier lifestyle changes such as smoking remission [11], physical activity [12] and blood sugar control [13] predominantly in diabetic CKD patients. The strongest evidence of treatment benefit for patients with proteinuric diabetic renal disease with early CKD are the use of angiotensin-converting enzyme inhibitors [14] or angiotensin II-receptor blockers [15].

Early diagnosis requires a highly sensitive kidney function test that is accepted by both physicians and patients alike. There is no golden standard method available up to today allowing for a simple and precise measurement of renal functions. One of the reasons behind it is due the high complexity and flexibility of the kidneys allowing for extreme damage compensation as far as an entire missing kidney. Most of the used qualitative and quantitative kidney function assessments are based on estimated glomerular filtration rate (e-GFR).

The glomerular filtration rate (GFR) is the amount of fluid filtered from all the glomeruli into their Bowman's capsule per unit time [16]. The glomerulus is the smallest filtration unit of the kidney and the area surrounding it is called the Bowman's capsule as depicted in Fig. 2.1 from Mikael Häggström [1]. Together with the tubule it makes up the nephron, which is the functional unit of the kidney. There around a million nephron per kidney in a healthy human adult [17].

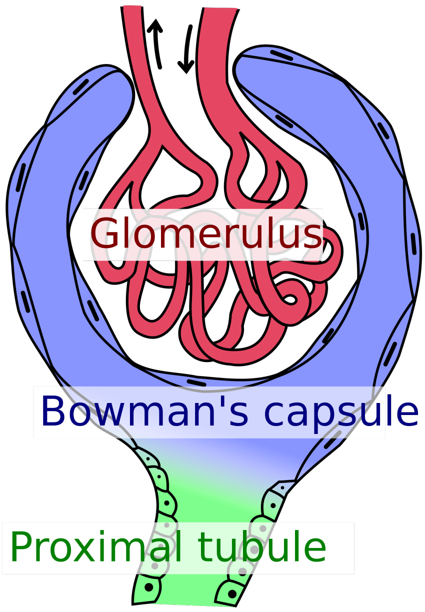


Figure 2.1: Glomerulus (red), Bowman's capsule (blue) and proximal tubule (green). "Bowman's capsule and glomerulus" by Mikael Häggström in the Public Domain, CC0.

This study aims to compare existing methods and approaches to be able to establish a tool that is able to give an automatic and more accurate result than previous attempts. Additionally the proposed methods integrated into the tool are able to help detecting potential mistakes that might have emerged during the experiment. Existing and new estimation formula are combined to adapt to the various absorption and excretion distributions. Furthermore, half clearance times in the various states of kidney function are determined and adapted to the according distribution.

---

[1] `https://en.wikipedia.org/wiki/Glomerulus_(kidney)#/media/File:Bowman's_capsule_and_glomerulus.svg`"Bowman's capsule and glomerulus" in the Public Domain, CC0.

2.2 Related Work

Age (years)	**Average e-GFR** (ml/min/1.73 m^2)
20–29	116
30–39	107
50–59	99
60–69	93
70+	85

Table 2.1: The average e-GFR in humans based on age according to the US National Kidney Foundation guidelines [18].

In 2002, the Kidney Disease Outcomes Quality Initiative (K/DOQI) of the US National Kidney Foundation defines CKD as the presence of functional or structural kidney damage and/or an adjusted glomerular filtration rate[2] lower than 60 ml/min/1.73m^2 for at least three months [19]. This definition was then revised and updated in the 2012 Clinical Practical Guideline for the Evaluation and Management of Chronic Kidney Disease by the Kidney Disease: Improving Global Outcomes (KDIGO) to include albuminuria in the classification, which was shown to have an important effect on outcomes [18]. These provided guidelines are widely used by healthcare professionals for the diagnosis and classification of CKD. In young adults, a typical GFR is approximately 120 ml/min/1.73m^2, and this value naturally declines as individuals age (see Table 2.1).

Category	**Deviation from supposed value**	**Description**
G1	$\geq$90%	Normal or healthy kidney function
G2	60% - 89%	Kidney damage with mild loss of kidney function
G3a	45% - 59%	Mild to moderate loss of kidney function
G3b	30% - 44%	Moderate to severe loss of kidney function
G4	15% - 39%	Severe loss of kidney function
G5	<15%	Kidney failure

Table 2.2: Chronic Kidney Disease (CKD) classification according to the US National Kidney Foundation guidelines [18].

Based on factors such as age, sex, and body size, each individual has a supposed e-GFR value. Deviations from this value can be subdivided into six stages of CKD. The first stage is considered normal or healthy while each subsequent stage indicates different levels of kidney dysfunction and associated risks ending with kidney failure. Table 2.2 shows the CKD classification from the US National Kidney Foundation.

---

[2] Units adjusted for body size with 1.73 m^2 being the estimated body surface area of an adult with a mass of 63 kg and a height of 1.7 m.

2.2.1 GFR Marker

These days most physicians use endogenous markers such as creatinine, urea, electrolytes, cystatin C or proteinuria found in either the plasma or urine. This is because these methods are easy and cheap to perform. However, they have limitations and their estimations are often restricted to a specific population type. The most common estimation formulas and their disadvantages are described briefly: The Cockcroft-Gault formula [20] tries to estimate creatinine clearance rate without the need of 24 hours urine samples but due to the disregard of muscle mass often overestimates [21]. The Modification of Diet in Renal Disease (MDRD) formula [22] also uses creatinine clearance and accounts for muscle mass by adjusting for body surface area but is less accurate in early CKD and underestimates e-GFR [23]. The Chronic Kidney Disease Epidemiology Collaboration (CKD-EPI) formula [24] outperforms the MDRD formula in accuracy and bias using creatinine clearance. The Schwartz formula measures e-GFR in children using creatinine clearance [25] and the Mayo Quadratic formula [26] tries to better measure e-GFR in patients with restricted kidney function by applying creatinine clearance but often fails to do so [27].

The most common GFR measure marker used in clinical practice is by far serum creatinine. Creatinine, first used in 1926 by Poul Brandt Rehberg to measure GFR, is a natural breakdown product of creatine phosphate from muscles making its concentration already steady in the blood but highly dependent of the muscle mass of the patient [28]. Additionally creatinine is not only filtered by the glomerulus but also actively secreted by the peritubular capillaries. These are the main reasons many of the formulas presented above over- or underestimates actual GFR and should be used with caution in patients with very low muscle mass such as cachectic patients and patients with cirrhosis. Other agents have also been tested such as the inhibitor of cysteine protease protein secreted by most cells in the body called cystatin C [29]. Alternatively to endogenous markers would be the use of exogenous tracers that have specific properties and are not found otherwise in the body.

To better understand the principles behind the most accurate kidney function assessments, it must be clear how substances are normally distributed in the body [17]. After a substance enters the systemic circulation, it penetrates the vascular wall into the interstitial fluid through perfusion and is then distributed to the body's tissues. Depending on the tissue mass, the permeability of cell membranes, the regional pH and the tissue vascularization the distribution of a substance into a tissue can occur at various rates. For tissues like muscle and fat where the perfusion is poorly, the distribution is very slow, this is especially the case when the tissue has a high affinity for the substance. After distribution equilibrium, when entry and exit rates are identical, the plasma concentration reflects the substance concentrations in tissue and in extracellular fluids. Furthermore, the substance is being metabolized and excreted and since excretion and metabolism occur simultaneously with distribution, the process is highly dynamic and complex. Volume of distribution, total body clearance and half-life are usually used to describe these processes.

By choosing the right exogenous marker, the process of distribution and excretion can be simplified drastically [30]. This would be the case if the chosen marker is non-toxic and water-soluble and would only undergo glomerular filtration. This happens when a tracer does not enter the intracellular space, nor binds to protein or fat, nor is metabolized or excreted by other organs, nor is absorbed or secreted by the peritubular capillaries. The closer the marker can be designed to have these properties the more accurate the concentration in urinary clearance would reflect the concentration in the plasma directly and therefore the easier would be to calculate the GFR. Since so far there is no perfect marker with all the desired properties, there will always be some trade-offs with precision.

There is an important relationship between volume of distribution, clearance and elimination half-life. The volume of distribution (VD) gives information of the distribution of a substance in a living body. A lower VD value represents a higher concentration of the substance in plasma, meaning that the substance can propagate to where it was intended to be. VD is calculated by dividing the concentration of the substance in plasma (mg/L) from the amount of the substance in the body (mg). Total body clearance (ClTOT) reflects the elimination of the substance from a body through the liver, lungs and kidney. Since most advanced markers are only eliminated by the kidney it can be simplified as the eliminating rate through urine (mg/hr) divided by the substance remaining in plasma (mg/L). Depending if a substance is being eliminated by zero-order or first-order elimination kinetics, the decay will be either linear or exponential, respectively. Only a small portion of substances such as Cisplatin, Fluoxetin, Omeprazol, Phenytoin, Salicylates and Ethanol fall under the zero-order (also known as the Michaelis-Menten) elimination kinetics, because for these substances there is a limit on how many metabolic enzymes are available. In this case the system is saturable. Half-life ($t_{1/2}$) determines the time needed for a reactant concentration to reach half of its initial concentration and is calculated differently depending on the order of the elimination kinetics. While $t_{1/2}$ for zero-order kinetic is dependent by the initial concentration C_0 and the elimination rate constant of a substance k: $t_{1/2} = \frac{C_0}{2k}$, it is not the case for the first-order kinetics. Here it is expressed as: $t_{1/2} = \frac{ln(2)}{k}$ making $t_{1/2}$ constant [31].

The administration of exogenous markers is usually executed either as a single bolus injection or a constant intravenous infusion [32,33]. Starting from the 1930s inulin was showing signs of an optimal exogenous marker [34]. Having to have a constant infusion, multiple urine samples over the 24 hours and not being water-soluble at room temperatures, made the use of inulin a complicated endeavor. Sinistrin solved the problem with the water solubility at room temperature, but because of the similar structure to inulin, both markers can only be detected in plasma and urine samples with the help of complicated, time-consuming enzymatic detection methods. At around the same time radioisotopes, such as 51Cr-EDTA, 99mTc-DTPA, 131I-Diatrizoat and 131I-Iothalamat, were used because of their easy tracking ability [21]. Not only were these radioisotopes radioactive but also difficult, expensive and time consuming to store, to handle, to recycle and to analyze. An alternative widely used in the 90s was an ionic tracer called Iohexol [21].

A different approach is to use the combination of fructosamine and a fluorophore, which allows to be detected and analyzed in plasma and urine with fluorescence spectroscopy. This type of marker is easily handled and costs only a fraction of what a radioisotopes costs. The groundwork for Fluorescein isothiocyanate (FITC) was laid by Rutili et al. in 1976 with the usage of FITC-dextran [35]. A few years later Inulin was combined with FITC to become fluorescent without losing the properties of a good tracer for GFR [36]. Without the room temperature water solubility property of inulin it was just a matter of time for FITC-sinistrin to replace it. FITC labeled sinistrin was produced by Pill et al. [37] which retained the room temperature water solubility property and did not show signs of accumulation in organs. However, it is imperative to acknowledge the substantial intrinsic auto-fluorescence emitted by living tissue, particularly when employing markers with emission wavelengths shorter than 600 nm, such as FITC-sinistrin, as it can potentially introduce measurement inaccuracies [38]. Furthermore, penetration depth increases with longer wavelengths. Consequently, it is advantageous to employ markers with higher emission wavelengths, specifically within the near-infrared (NIR) range (650–900 nm). In this regard, in the same research facility, Huang et al. developed and produced several other zwitterionic NIR fluorescent agent [39,40], two of which were used extensively in experiments and represent a substantial portion of the data used in this work. The differences between the two markers, FITC–HPγCD and ABZWCY–HPβCD, are mainly in admission and absorption wavelength and their composition. The former being in the blue light spectrum is a polyfructosane, while the latter in the infrared spectrum is a cyclodextrin. Huang et al. also published an overview of several fluorescent markers and compared them [41].

2.2.2 Compartment Model

A further simplification of the determination of renal function using fluorescence-labeled markers is the transcutaneous real-time determination of the excretion kinetics of the markers. This was demonstrated by Rabito et al. 2005 with the transcutaneous determination of GFR with carbosityril-124-DTPA-EU with a laser-induced fluorescence measuring system [42]. A miniaturized low-cost device, developed by the group led by Prof. Gretz [43], is capable of reading the markers concentrations, while the animal is moving freely [44]. With such a device, the manual extractions of urine and plasma samples become obsolete. Not only it avoids a procedure that otherwise would inflict significant stress upon the animal but furthermore, the subsequent time-consuming laboratory analyses of these samples are avoided.

In the case of just a few samples, a one compartment model, sometimes also referred to as mono-exponential, can be used to fit the data. Only the exponential regression curve after reaching the equilibrium is taken into account for calculating the integral under the curve, thus simplifying the integral. The determination of the second function is omitted because it represents only a very small part of the surface area and the resulting error in the GFR calculation is considered to be acceptable [45]. In principle, it can determine the GFR relatively accurately with only two plasma samples, if the first sample was taken after reaching the equilibrium between plasma and extracellular fluid. However, it is extremely vulnerable to measurement errors [46].

Despite collective efforts to streamline classical GFR determination methods, such as the infusion or bolus techniques, which seek to mitigate the dependency on urine and plasma sampling [47–49], this aspect remains indispensable. On the other hand, when a marker is administrated in a bolus fashion and the concentration is measured frequently, the one-compartment model is often not suitable for the early time course of the marker concentration in plasma because of a multi-phasic decline of the concentration-time curve. This is due to the delay caused by perfusion or diffusion rate limited processes during the distribution phase of the substance to reach equilibrium in the tissue. Spontaneous equilibrium of a substance between plasma and tissues is rarely observed.

By introducing additional compartments for the substance to distribute itself from the central compartment, usually blood, to one or several peripheral compartments the model becomes more accurate but also more complex. These are the multi-compartment models, which include the more slowly equilibrating peripheral compartments caused by tissues with a surrounding protective membrane or with poor perfusion. Three-compartment model have, additionally to the blood and interstitial fluid compartment of the two-compartment model, an extra cellular compartment.

As an example for a bi-exponential decline after an intravenous bolus dose, a two-compartment model is shown as:

$$C = A \cdot e^{-\alpha \cdot t} + B \cdot e^{-\beta \cdot t}$$

where C is the total concentration curve, which is composed as the sum of two lines, when data are plotted on semi-logarithmic scale. These two lines can be interpreted as the equation for each compartment and have an initial concentration represented by A and B respectively. The specific slope of each line are defined by α and β respectively as shown in Fig. 2.2. Their corresponding half-lives are estimated as $t_{1/2\alpha} = \frac{ln(2)}{\alpha}$ and $t_{1/2\beta} = \frac{ln(2)}{\beta}$.

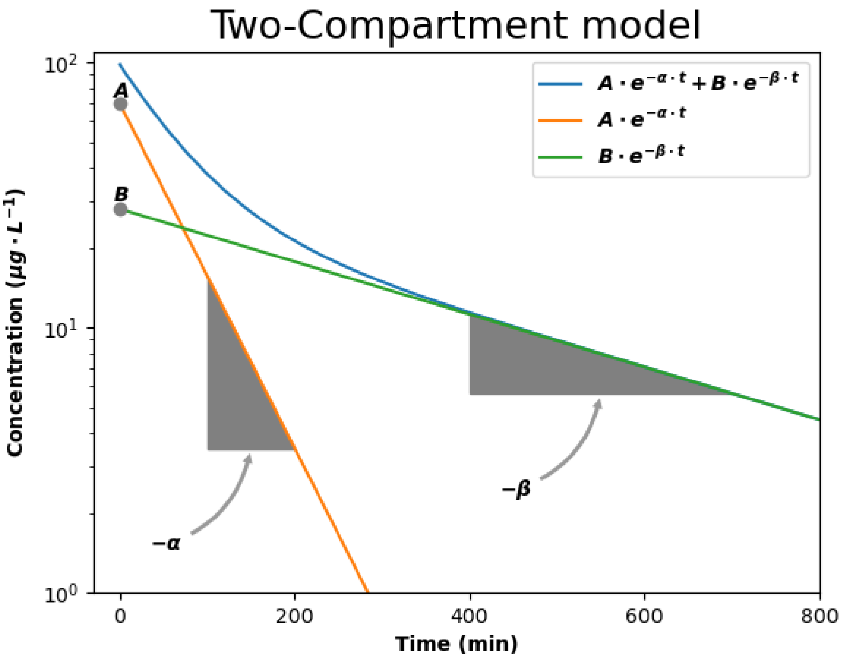


Figure 2.2: Example of a two-compartment model (blue). Central compartment (orange) and the peripheral compartment (green) with their respective initial concentrations A and B as well as their respective slopes $-\alpha$ and $-\beta$.

2.3 Material & Methods

This chapter describes methods used for analyzing the data generated by the transcutaneous approach to assess renal function in conscious, freely moving animals, utilizing a near infrared device in conjunction with the fluorescent renal marker. A comprehensive exposition of the the data itself is also provided, explaining the possible differences and nuances.

2.3.1 Device, marker and animal preparation

The innovative methodology used to acquire the data employs a compact electronic device to non-invasively monitor the excretion of the exogenous fluorescent renal marker, through the animal's skin [1]. The animal is given anesthesia on a warming plate to counteract the potential hypothermic effects of it. Once the animal reaches an appropriate level of anesthesia a patch of fur is removed from its back to accommodate the adhesive patch where the device will be attached to.

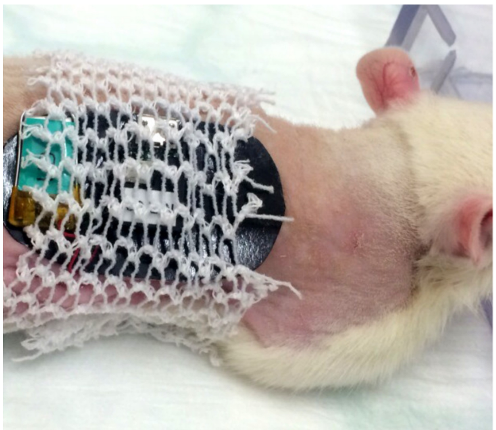


Figure 2.3: A picture showing the animal ready for the injection [44]. On its shaved back the visible device and battery are attached to its back with double-sided adhesive and protected by a mesh to further stabilize the setting. The battery is situated on the left with its turquoise color while to its right lies the device with the silver usb+port showing through the mesh.

The device comprises of an optical component featuring two light-emitting diodes (LEDs) and a photodiode as well as a rechargeable battery acquired from MediBeacon (MediBeacon GmbH, Mannheim, Germany). To prevent noise from ambient light falsely perceived by the photodiode as infrared light coming from the fluorescent marker, a black plastic foil is put on the device. This device and the battery are effortlessly affixed to the shaved dorsal area of the animals via a simple double-sided adhesive patch and protected by a meshed band as shown in Fig. 2.3. Importantly, the animals remain awake and unrestrained throughout the recording period, which spans approximately 2 hours permitting sequential measurements within the same animal.

The fluorescent agent ABZWCY-HPβCD, produced by CYANAGEN (CYANAGEN S.r.l, Bologna, Italy), is combined with the Deltajonin physiologic solution (DeltaSelect GmbH, Rimbach, Germany) and injected through the tail vein. Following intravenous administration of the renal marker, the LEDs stimulate it, and the ensuing fluorescence is captured by the photodiode. The data generated are recorded and stored in the device's internal memory, subsequently facilitating the creation of an elimination kinetics curve for the marker.

To minimize the duration of anesthesia, the ABZWCY-HPβCD dose and the set-up of the device with the battery onto the patch is prepared beforehand. Once the animal has been anesthetized, the complete procedure of affixing the device to the animal and administering ABZWCY-HPβCD typically takes approximately five minutes. After the animal has been left alone for the duration of the readings the device is removed and connected to a computer through a USB cable and the results are saved as a .csv file.

2.3.2 Data

The data saved is a csv file containing three columns. The first value is the initial time the experiment started, the second is the time passed since the beginning in seconds and the last value represents the concentration readings ranging from zero to 2047. Each row is equivalent to approximately a second and each csv file contains as many rows as the duration of the device being attached to the battery. When plotted, it forms a kinetic curve of the animal as shown in Fig. 2.4.

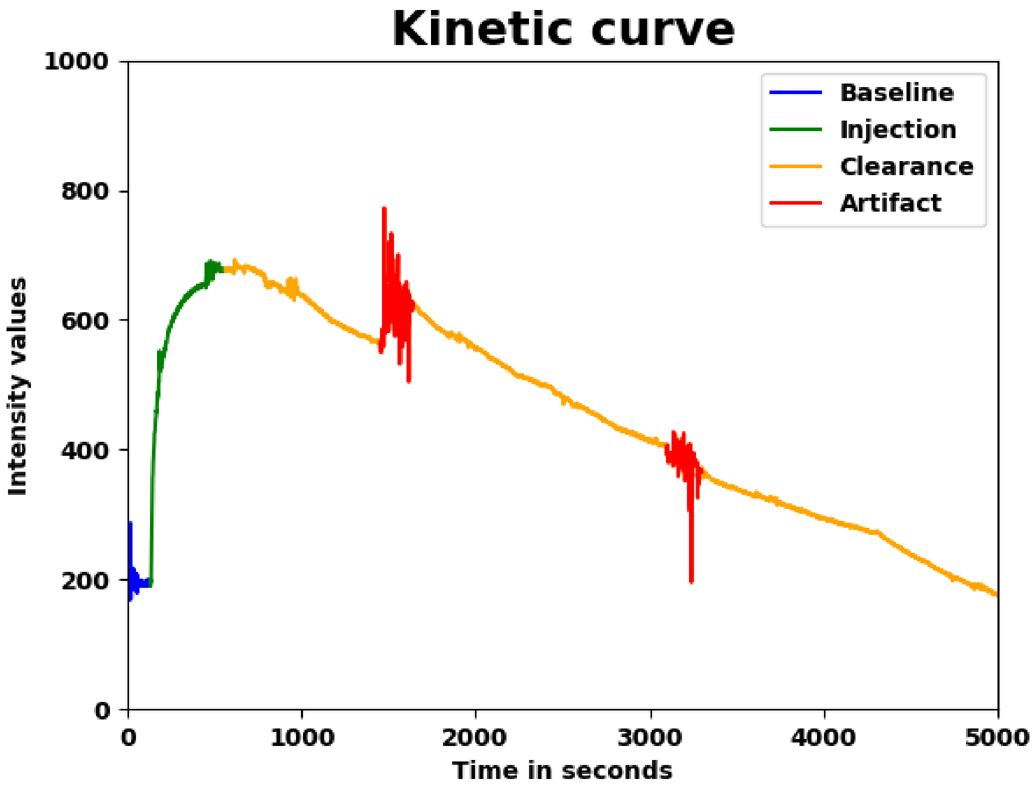


Figure 2.4: Example of one of the intensity readings of an animal. The baseline (blue) is before then injection phase (green) which is followed by the clearance phase (yellow). Artifacts (red) can occur when the animal moves or tries to remove the device from its back.

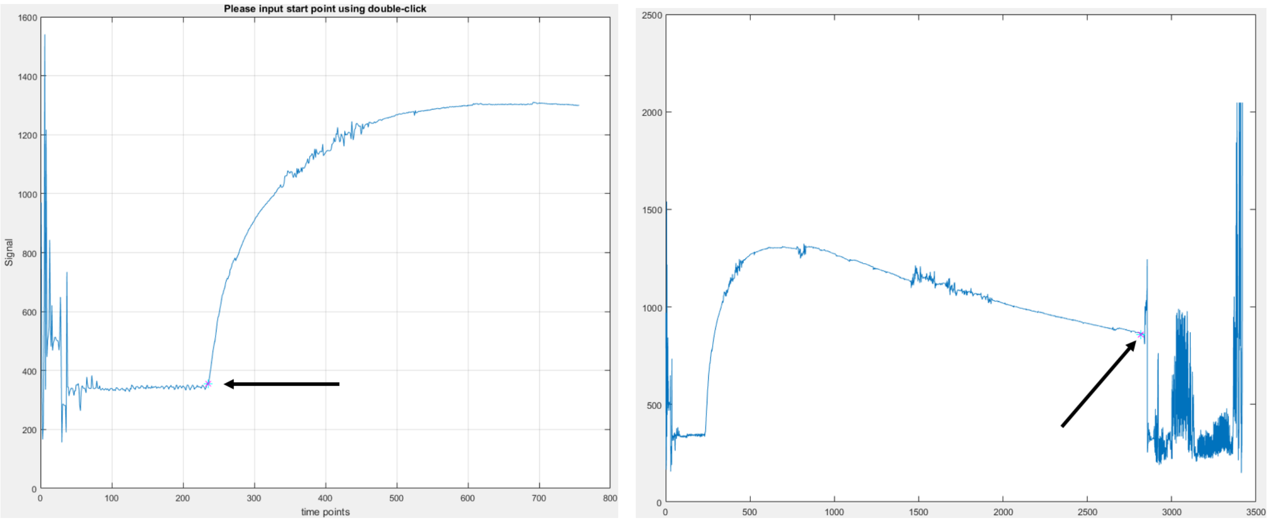
The curve can be subdivided in the different phases: baseline (blue), injection (green) and clearance (orange). Sometimes if the animal moves or tries to dismount the device, artifacts as seen in the plot in red can occur. These are most prevalent towards the end of the curve when the device is being dismounted from the animal. The curve shown in Fig. 2.4 is one example of the kinetic curve shapes, which will be explored more in detail later on. A total of 14854 csv files were found in the repositories of the research department.

2.3.3 GFRmeasure

For a preliminary analysis and labeling of GFR curves, visualization and interactive annotation methods have been developed and released as open-source software tool GFRmeasure on GitHub [3].

The curve plotted by the software offers an initial assessment of data quality. It is imperative to closely scrutinize the shape of the curve, with particular attention to the injection's phase peak, signifying the end of the administration of the fluorescence marker.

A good selection of injection and cutoff points is crucial in order to evaluate the half-life correctly. For humans it is a trivial task to immediately recognize patterns and see if a curve is usable, when the exact moment of the injection happened, as well as to see when to cut the data to avoid noise or artifacts. After loading the data, the user is asked to manually click on the time point that could resemble the injection point as shown in Fig. 2.5a. Similarly, the user is asked to select a cutoff point where the experiment is finish, as seen in Fig. 2.5b.



(a) (b)

Figure 2.5: Manual selection of the injection point (a) and the cutoff point (b). The point of injection can be clearly seen as the moment of increase from the baseline. The cutoff point is typically chosen to be the last point before movement artifacts occur.

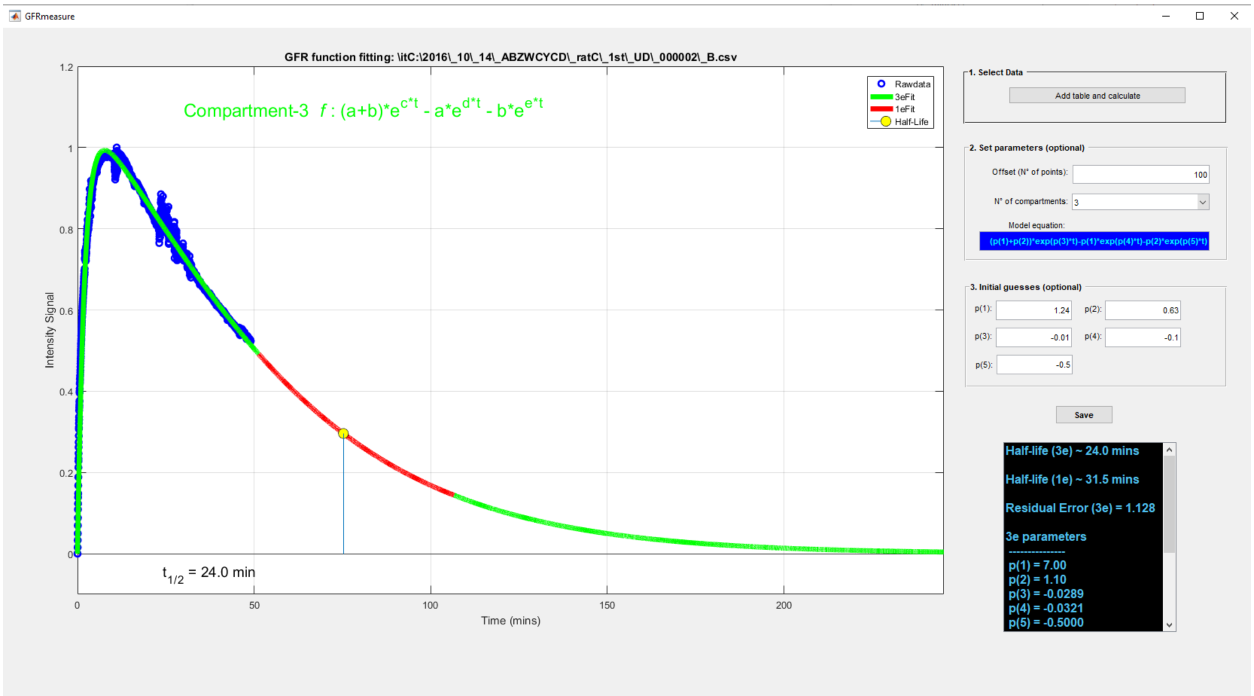
GFRmeasure allows evaluating the half-life of the fluorescent agent ABZWCY-HPβCD but

---

[3] https://github.com/AngeloTorelli/GFRmeasure. Last visited: 15.09.2023.

can also be used with any generated high-frequency kinetic curve. The excretion kinetics of ABZWCY-HPβCD is characterized by fitting the device-measured signal to one- to three-compartment models. A 3-compartment model is selected by default but can be changed by using the ”No of compartments” tab. A reasonable initial guess for curve fitting is already provided as shown in the field of ”3. Initial guesses (optional)” in Fig. 2.6. However, if the model does not fit well and residual error is high, these parameters can be adapted to the specific curve.

This software with its graphical user interface (GUI), shown in Fig. 2.6, was programmed entirely with MATLAB and tested for Windows and Linux and is freely accessible, intuitive, and designed to be user-friendly, even for those new to its operation. It was also used to create a ground truth dataset for comparison of some of the features.



2. Set parameters (optional)

Offset (N° of points): 100

N° of compartments: 3

Model equation: (p(1)+p(2))*exp(p(3)*t)-p(1)*exp(p(4)*t)-p(2)*exp(p(5)*t)

3. Initial guesses (optional)

p(1): 1.24
p(2): 0.63
p(3): -0.01
p(4): -0.1
p(5): -0.5

```
Half-life (3e) ~ 24.0 mins

Half-life (1e) ~ 31.5 mins

Residual Error (3e) = 1.128

3e parameters
 --------------
 p(1) = 7.00
 p(2) = 1.10
 p(3) = -0.0289
 p(4) = -0.0321
 p(5) = -0.5000
```

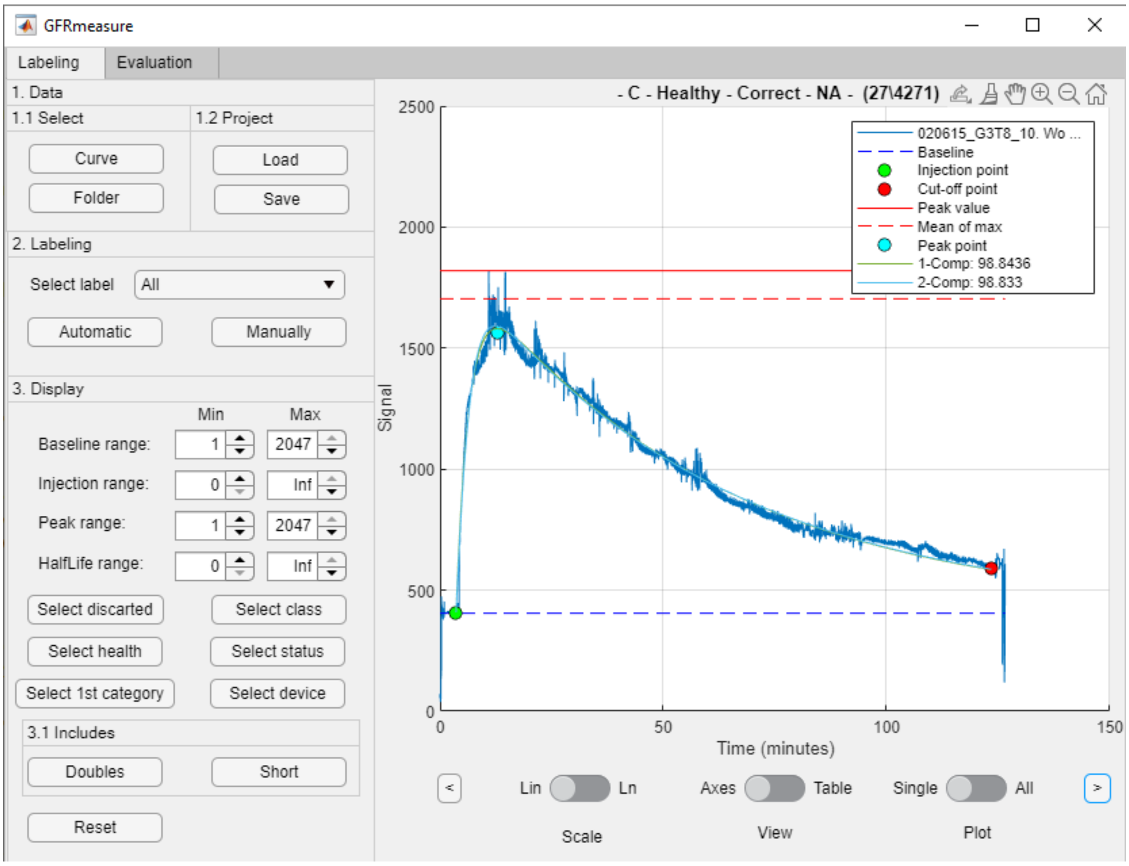
Figure 2.6: The GFRmeasure GUI after loading and fitted a kinetic curve with compartment model. It shows all the results such as the half-life time in minutes on the bottom left and all the other parameters and the residual error of the chosen compartment model can be found in the black box to the right. The results are saved as images and a csv files in the user-selected folder.

Due to the complexity and the sheer amount of data, there is a need for automatic methods helping to differentiate the different phases, fit the kinetic curve with one of the compartment models, calculate the half-life, categorize the data in the different health state of the kidney and filter through the data.

2.3.4 GFRmeasure+

Managing and enabling research on extensive repositories of GFR measurements, acquired and processed over the course of several years in the context of individual research projects, requires a robust approach for automating the analysis and categorization of diverse kinetic curves. Such an approach needs to address the challenges of maintaining a coherent and organized record of which curve corresponded to what became increasingly daunting.

As the need for a comprehensive, collective analysis of this substantial dataset emerged, the research group recognized the pressing necessity for a robust solution. This is where GFRmeasure+ (see Fig. 2.7) came into play, designed to enhance GFRmeasure even further.



Select label: All

Baseline range: 1 2047

Injection range: 0 Inf

Peak range: 1 2047

HalfLife range: 0 Inf

Figure 2.7: The GFRmeasure+ GUI after loading and fitted a kinetic curve with compartment model. It shows all the results in the legend on the top right of the plot. On the left side there are options to load and save the data including results as well as manually correcting the results. Additionally it is also possible to filter through the data and see the results as a table.

It embarked on the task of addressing the intricate challenges inherently associated with high-frequency data sampling, as well as adapting to the ever-evolving landscape of new measurement devices. The ultimate objective was to develop a sophisticated software package capable of automating the analysis and categorization of the diverse kinetic curves. This software aimed to bring order to the extensive dataset, enabling swift and accurate identification of individual curves and facilitating comprehensive research endeavors that rely on the collective analysis of this wealth of data. In essence, GFRmeasure+ was developed to streamline the process of harnessing valuable insights from the substantial body of kinetic data amassed over the years.

Peak detection

The initial challenge entails the segmentation of the curve into the aforementioned phases, namely, the baseline, injection, and clearance phases. To accomplish this, it is imperative to precisely identify the peak after the injection phase, which serves as a crucial reference point in this process. This peak localization is achieved through a procedure involving median filtering of the intensity curve. Specifically, a median filter with an order equal to five percent of the curve's length is applied. Subsequently, all local maxima present in the filtered curve are identified as shown in Fig. 2.8. However, not all local maxima are deemed relevant. Those local maxima falling below the average value of the entire curve and those occurring after the halfway point in terms of time are discarded, ensuring that only significant peaks are considered. Among the remaining candidate peaks, the first occurring is selected. This approach forms the cornerstone of the curve subdivision process.

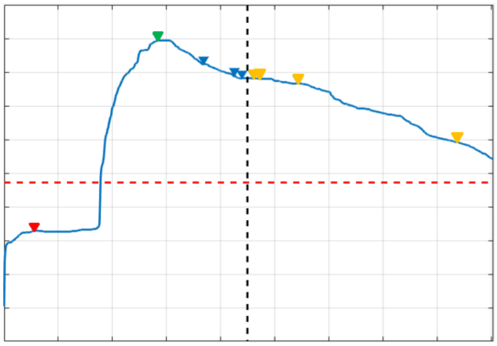
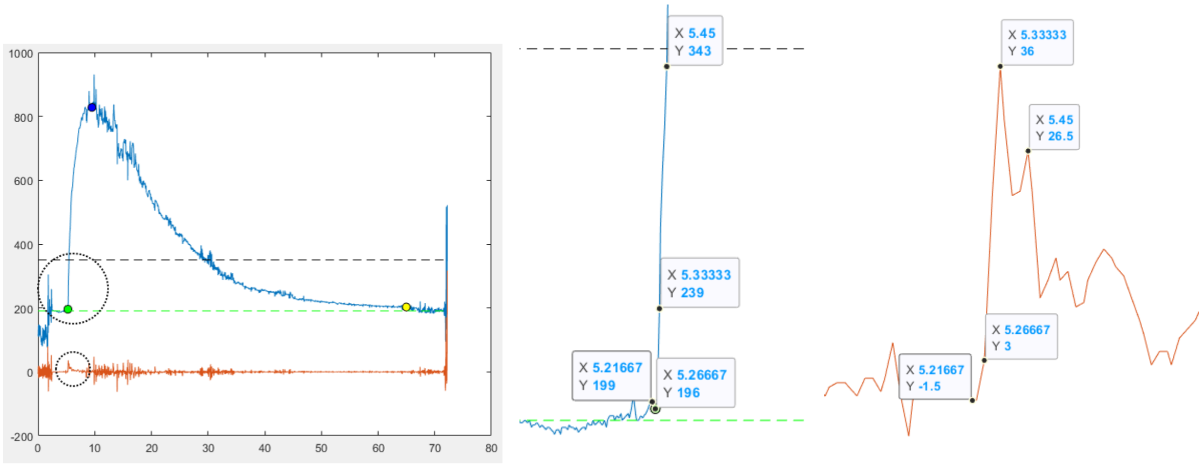


Figure 2.8: Depiction of the peak point detection process. The local maxima (triangles) are identified on a median filter with an order equal to five percent of the curve's length (blue line). The plot is divided by half the time of the curve (black line) and the average intensity value of the entire curve (red line). Local maxima are discarded if they are below the average intensity (red triangle) or after half the time of the curve (yellow triangles). From the remaining possibilities (blue triangles) the first occurring (green triangle) is chosen.

Injection point detection, baseline calculation and duplicate identification

After successfully identifying the peak in the kinetic curve, the subsequent challenge revolved around pinpointing the precise moment of injection. Among the three critical points, injection, peak, and cutoff, the determination of the injection point held paramount importance as it significantly influences the curve fitting process. Achieving accuracy in identifying the injection point was crucial for optimizing curve fitting outcomes. Therefore, this step necessitated the utmost precision and reliability among the findings.

To discern the exact moment of injection, the following approach was developed. To ensure that only relevant points were evaluated, only points occurring prior the identified peak point were considered as potential candidates for the injection point, which is exemplified in Fig. 2.9a.



(a) (b) (c)

Figure 2.9: The automatic process of selecting the precise injection point. Plot (a) of the kinetic curve (blue) and its gradient (orange) with the injection point (green dot), baseline (green line), mean intensity value (black line), peak point (dark blue dot) and cutoff point (yellow dot). Initiating the search from the zenith and directing it leftwards, only data points boasting intensity values below the mean intensity of the entire curve fall within the purview, as depicted by the expansive dotted circle in (a). The respective points along the gradient of the curve are being analyses in such that first the local maxima is found in the small dotted circle in (a) or in the magnified version (c). From this summit, the journey descends to the left, in tandem with the gradual diminishing of intensity values along the curve (b). Any deviation from these criteria instantly halts the search, ensuring the precision and reliability of the injection point determination.

Here the peak point is marked as a blue dot. The two dotted circles in Fig. 2.9a are regions which are magnified to better illustrate the method. Moreover, the injection point had to exhibit intensity values lower than the average intensity of the entire kinetic curve marked as the dash line in Fig. 2.9b. Additionally the one-dimensional numerical gradient of the curve is taken into account as represented by the orange curve in Fig. 2.9a and in its zoomed in version Fig. 2.9c. The gradient can be thought of as a collection of vectors pointing in the direction of increasing values of the intensity function.

The exact moment of injection was identified when the following criteria converge: The algorithm detects the first local maximum in the gradient prior the peak which in the example shown in Fig. 2.9c happens to be at five minutes and twenty seconds or $x = 5.33$. This gradient peak signifies a notable surge in intensity which most likely is the injection phase. The samples just before this gradient peak is of particular interest because it is where the injection point will be. These identified samples are further validated by ensuring that, moving leftward from this peak point, the intensity values also continues to diminish. This reduction in intensity, in conjunction with the gradient's behavior, corroborated the precise injection moment making the injection point at five minutes and sixteen seconds or $x = 5.26667$. Continuing to follow the gradient leftwards it can be seen that the value keeps on diminishing as seen in point at five minutes and thirteen seconds or better $x = 5.21667$ but at the same time it increase in the intensity curve stopping the algorithm.

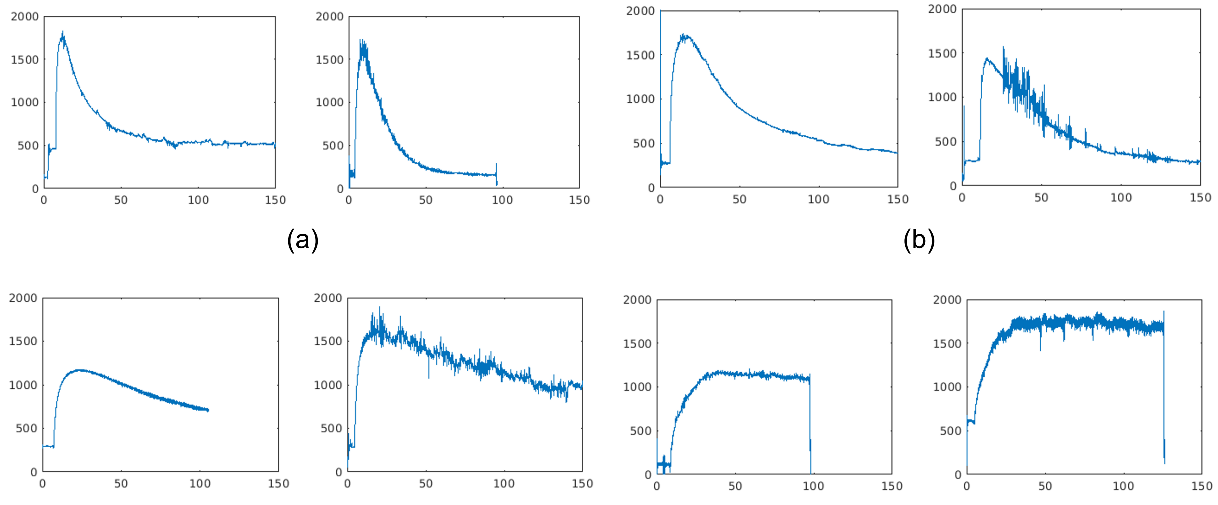
Based on the injection point the baseline is calculated as the median intensity value from the start of the curve up to the injection point. This part is classified as the baseline phase. The baseline also serves as a reference for further calculations. One of which was used in conjunction with the injection and peak point to identify duplicate curves. This process involved sifting through all available curves to pinpoint instances where the timestamp of these points align. In cases where these criteria match, a further comparison of the intensity values between the injection and peak points with the curves that were marked as potential duplicates was conducted, ensuring higher accuracy in the identification process.

Cutoff point detection

The wrong choice of the cutoff point can significantly impact the accuracy of curve fitting, with the quality of the data playing a pivotal role in this regard. Specifically, if opting for the cutoff point on artifacts that frequently manifests toward the end as a result of the device's removal process rather than the genuine intensity curve, the impact on curve fitting is drastic and the accuracy of the half-life time determination diminishes. On the other hand when done right the precision persist even when the curve undergoes significant truncation. For this reason the last ten percent of the data is not taken into account. From the remaining points the last intensity value above the baseline and below the intensity value on the peak point is selected.

Health classification

The shape is tied to the health of the animal. A few examples are presented in Fig. 2.10.



(a)

(b)

(c)

(d)

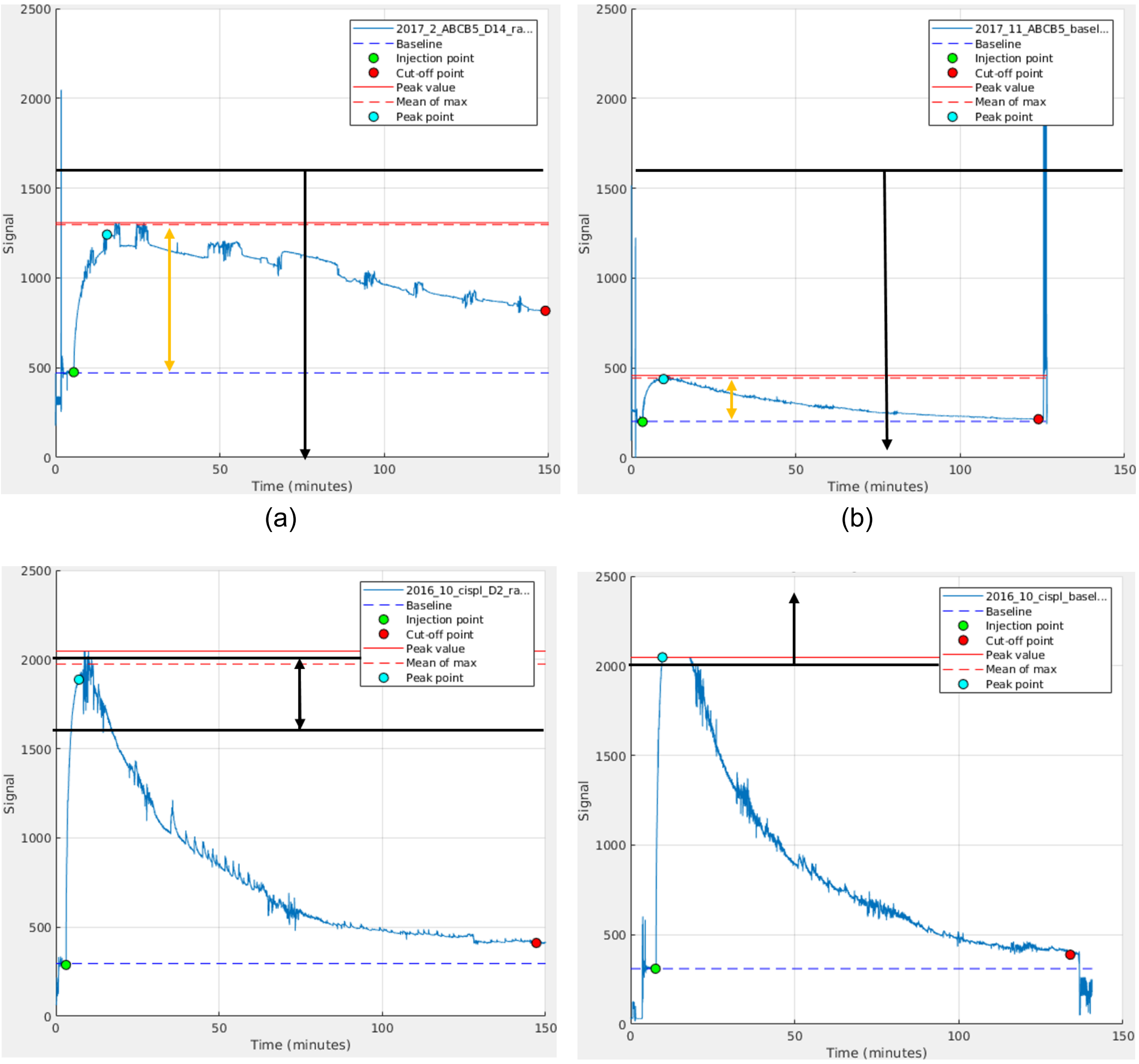
Figure 2.10: Examples for different type of kinetic curves grouped according to the health of the animal. From very healthy (a) where the half-life clearance is around ten minutes and does not surpass the twenty minute mark, to all the way to kidney failure (d) where clearance does not occur but instead the marker gets distributed in the body. A healthy kidney (b) manages to have a half-life clearance between twenty and forty minutes while the sick animal (c) already needs more than forty minutes to do so.

If the kidneys of the animal are healthy (Fig. 2.10a) it has a rapid clearance. On the other hand the more compromised the kidney is, the slower the clearance and therefore the slope of the kinetic curve flattens as seen in Fig. 2.10d. While it is feasible to compute the half-life times from these kinetic curves, a crucial limitation arises when attempting to directly compare the results obtained from different experiments. This limitation stems from the absence of information regarding the exact quantity of the injected marker. Nonetheless it is of high value to be able to categorize the curve into its clearance potential.

Compared to the human CKD classification mentioned in Table 2.2 there is no clear reference for the different types of animal. For the purpose of convenience the curves are categorized according to their half-life times as follows: very healthy with a half-life up to twenty minutes, healthy betwenn twenty and forty, sick between forty and one hour and kidney failure above the hour mark.

ABCD quality scoring system

Optimal curves exhibit peak intensity values within the range of approximately 400 to 2000. Within this range in mind, a quality scoring system has been established to categorize these curves effectively.



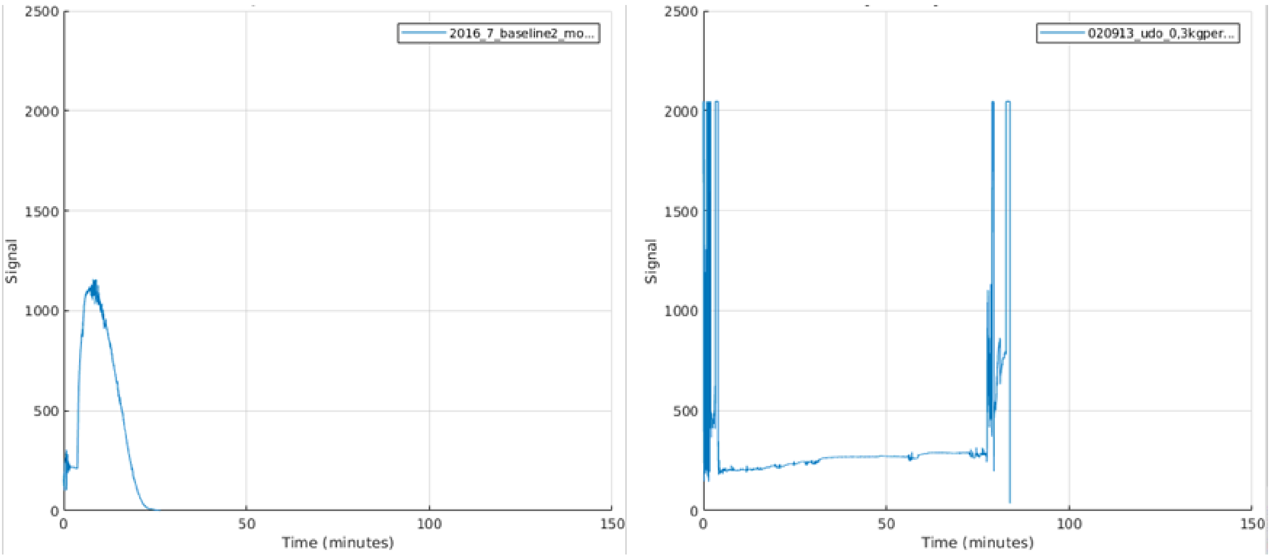
(a) (b)

(c) (d)

Figure 2.11: One example for each classification according to the proposed ABCD quality scoring system of kinetic curves. Black lines and arrows indicate the threshold and range in which the maximum intensity should lie while the orange arrows show the spread between the baseline and peak.

Curves with peak intensities below 1600, and a considerable spread between the baseline and the peak during the injection phase, exceeding 300 units, are grouped under class A (Fig. 2.11a). Conversely, those curves falling below this intensity range are classified as part of class B (Fig. 2.11b). Moving up the intensity scale, curves that reach values surpassing 1600 but remain below the 2000 mark are designated as class C (Fig. 2.11c). Values exceeding 2000 present a challenge for accurate reading, primarily because the device's photodiode saturates at this level. Consequently, such curves are classified as class D (Fig. 2.11d).

Furthermore, for curves falling within higher range of the 2000 mark, the consideration of quenching effects becomes essential. In cases where the device only register values around 400, distinguishing the signal from background noise becomes challenging. Therefore curves that fall below 400, are too short or have no peak are deemed unsuitable and necessitate exclusion. An example for too short can be seen in Fig. 2.12a as well as an example of a mix of below 400 and no peak is displayed in Fig. 2.12b.



(a) (b)

Figure 2.12: Examples of discarded curves. A short curve with values surpassing the baseline and reaching zero (a). A noisy curve with no peak with genuine intensity values below the 400 mark (b).

The initial selection of an appropriate marker dosage tailored to the specific animal under investigation represents the foundational stride toward attaining robust and reliable curves.

Ground truth data

A ground truth dataset of hundred randomly chosen samples was created with the help of GFRmeasure. The default setting were only used when the fitting was correct, otherwise the parameters were adjusted to obtain better results. All the parameters as well as the injection, peak and cutoff points were saved. Additionally the half-life time was calculated and saved. This was done to be able to compare the automatic features of GFRmeasure+.

2.4 Results

This chapter lays out the finding of the available data with GFRmeasure+. Additionally it shows the comparison of GFRmeasure with the results of the manually labeled dataset with two other software solutions. The first being the go-to pharmacokinetic solution existing for the past twenty years called WinNonlin™ from Certara (Certara Inc., Princeton, New Jersey, USA). The second program in question is the accompanying software to the reading device from Medibeacon. In order to ensure the correct utilization, the data was analyzed with WinNonLin by Prof. Dr. med. David Czock from the department of clinical pharmacology and pharmacoepidemiology at the Heidelberg university hospital. The results were returned and compared.

2.4.1 Evaluation of GFRmeasure+ automation

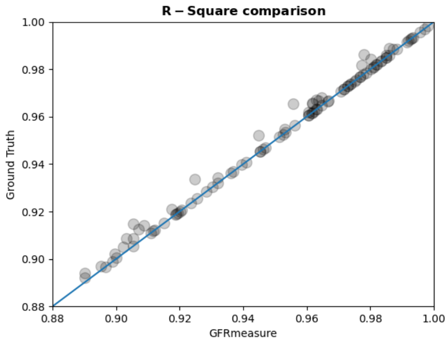


Figure 2.13: R-square error comparison between GFRmeasure+ (x-axis) and the ground truth dataset (y-axis). In this example the plot is zoomed in in the range between 0.88 and one, with one being a perfect fit. Blue line indicates the symmetry line. Each sample is plotted with a twenty percent transparency, enabling visibility of the underlying sample.

As described in the methods section, GFRMeasure+ is able to automatically detect the injection, peak and cutoff points therewith fully automatically fit the compartment models. To test this fully automatic mode, GFRmeasure+ was tested against the ground truth dataset. Instead of comparing the difference between the timestamps of the injection and cutoff points of the ground truth and GFRmeasure+, the R-square error of the fittings were taken into account instead as shown in Fig. 2.13. While both the injection and cutoff points do exert a substantial influence on the fitting process, the comparison of the R-square error holds even more significance. This is because the R-square error can remain consistent even when the specific points used for the analysis differ. As previously mentioned, this phenomenon is akin to the earlier example in which the tail end of a curve is preferably truncated to mitigate artifacts. Out of the hundred samples, seventy had identical fittings, the others were influences mostly by the cutoff point, when including too much noise. The half-life difference between the two deviated by a maximum of 24 seconds, which is very low considering the typical half-life times between 10 and 90 minutes. This resulted in a mean difference of 0 and a ±1.96 standard deviation of ±15 seconds as shown in the Bland-Altman plot in Fig. 2.14.

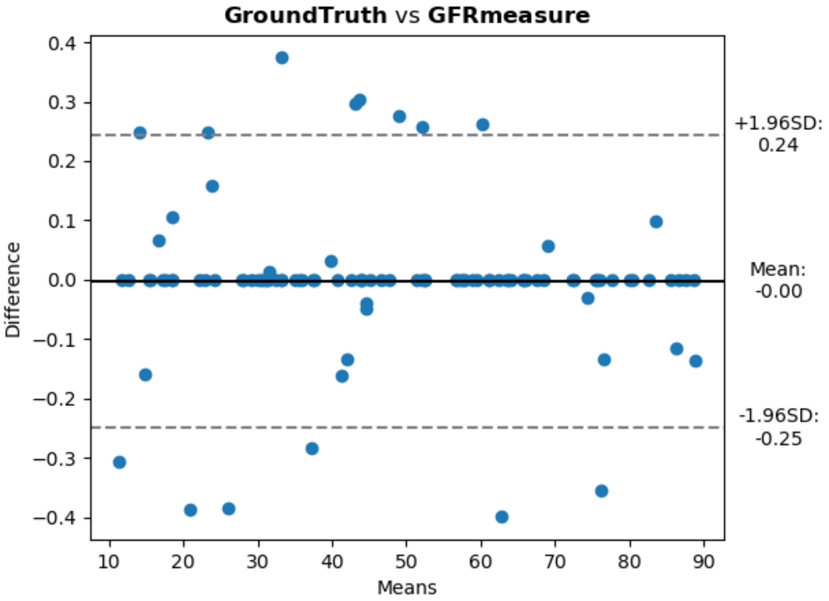


Figure 2.14: Bland-Altman plot comparing half-life times of the ground truth dataset against GFRmeasure+. All values in both plots are in minutes. The y-axis shows the difference in half-life times of the compared solutions for each of the hundred kinetic curve analyzed while the x-axis shows the mean value of the two. The dotted lines are the ±1.96 times the standard deviation.

2.4.2 Evaluation of available data

Thanks to the automatic abilities of GFRmeasure+, it was able to evaluate all 14854 csv files found. Only 4815 curves were identified as unique. Comparing the curves with their duplicates it showed that many curves were shortened at the beginnings or the ends or both and some times several times at different length of truncation to remove artifacts manually. This was done to increase the accuracy of the software from Medibeacon, which was at that time the only solution at the researcher's disposal. As visualized in Fig. 2.15, another 711 were discarded completely due to:

- having the wrong layout or coding in the csv file, with 429 cases;
- having not the right date and or time formatting, representing a total of 43 files or
- having no peak at all such as the curve shown in Fig. 2.12b, being 239 occurrences.

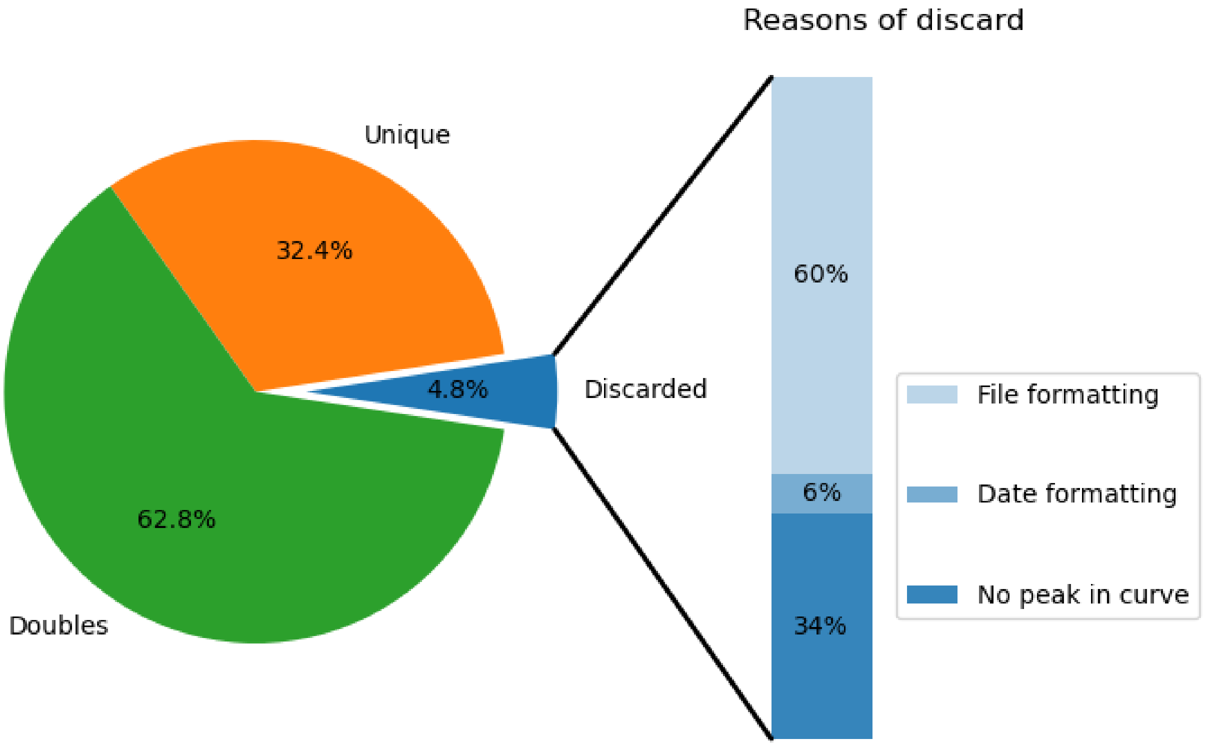
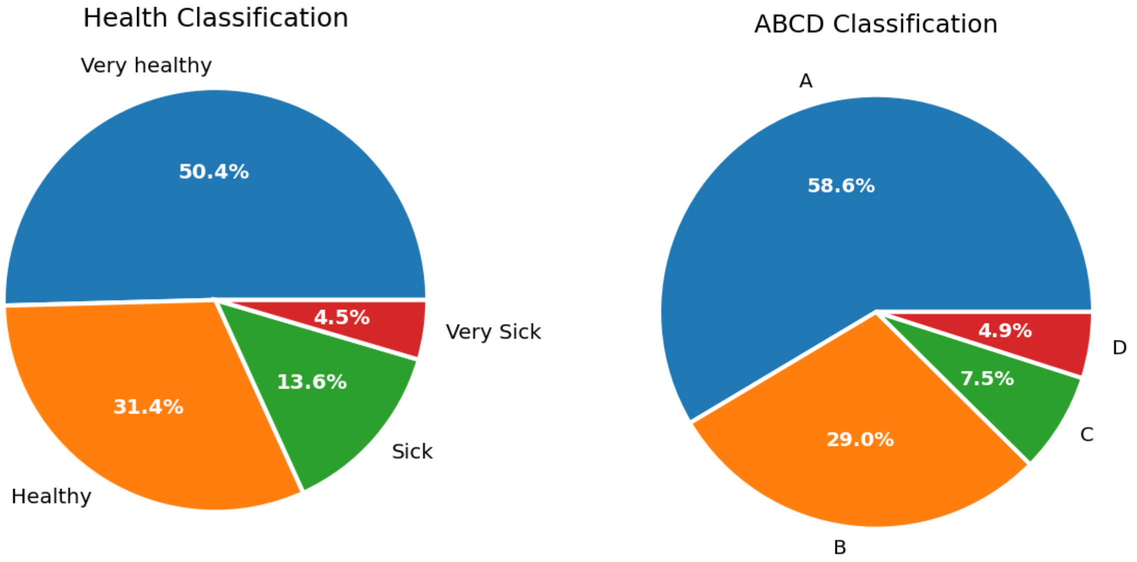


Figure 2.15: Distribution of the available data. The data is subdivided in doubles (green), uniques (orange) and discarded (blue). The latter has a subdivision regarding the reason of being discarded: most occurring with 60% being having a wrong formatted csv file (light blue), followed by curves without peaks at 34% (dark blue) and files with wrongly formatted date or time with 6% (medium dark blue).

Thanks to GFRmeasure+'s capability to display filtered curves based on specific criteria, including ranges of baseline, injection and peak points, half-life time, as well as health categories, ABCD scoring system, device used, curve creator, and the exclusion of duplicates or shorter curves, it became straightforward to investigate distinct scenarios.

When examining the diverse health conditions represented by the unique curves, the data revealed that 2428 curves were classified as ”Very healthy,” 1511 were categorized as ”Healthy,” 657 were denoted as ”Sick,” and only 219 fell into the ”Very sick” category, as illustrated in Fig. 2.16a.



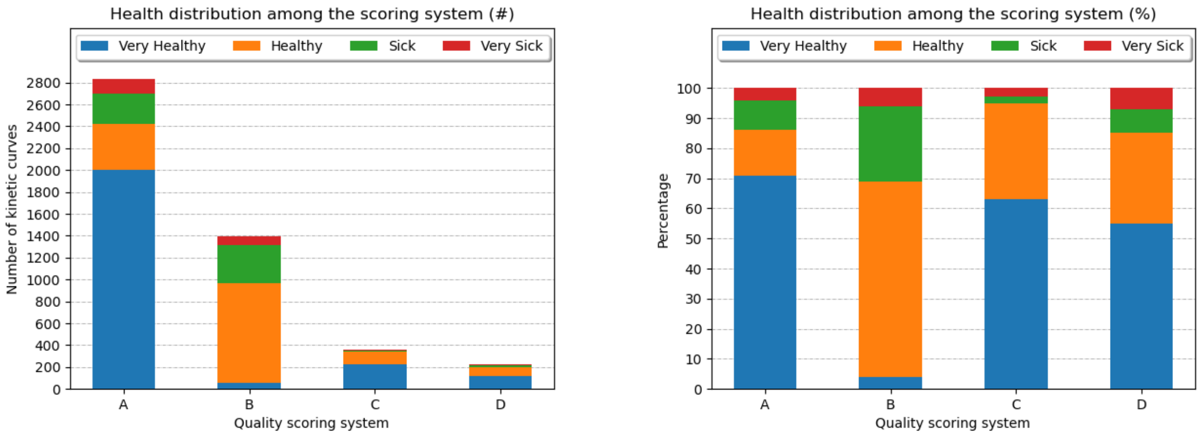
(a) (b)

Figure 2.16: Distribution of unique curves divided according to their health classification (a) and to the introduced ABCD scoring system (b). In (a) very healthy curves (blue) are the most common. Second most common are healthy (orange) followed by sick curves (green) ending with the very sick category (red). Similar is the distribution of the ABCD scoring system. Class A (blue) is the most represented followed by B (orange), C (green) and D (red).

This disproportion of very healthy and healthy to the sick and very sick can be hypothesized as follows. In most experiments or trials the test group is compared to a control group which is in a healthy state. Additionally, the animals in the test group are closely monitored and subjected to multiple follow-up tests. Furthermore, it may take some time for the treatment to show its effects. Therefore, healthy cures accumulate and appear more often than sick ones.

Finally once a sick state is reached it is usually the case that the experiment is brought to an end if it proves the statement of the experiment, to avoid further suffering of the animal.

A different categorization of the unique curves is with the introduced ABCD scoring system. Most represented category is A with 2820 curves, followed by B with 1397. The other two categories are not as strongly represented with only 360 in C and 238 in D as seen in Fig. 2.16b. Despite the similarity in distribution observed in the pie charts, it is purely coincidental as shown in Fig. 2.17. Fig. 2.17a shows the distribution in absolute numbers and Fig. 2.17b shows the distribution in percentage.



(a) (b)

Figure 2.17: Health distribution among the ABCD quality scoring system in absolute numbers (a) and in percentage (b). The colors are maintained as in Fig. 2.16a with blue, orange, green and red corrisponding to ”Very healthy”, ”Healthy”, ”Sick” and ”Very sick” respectively.

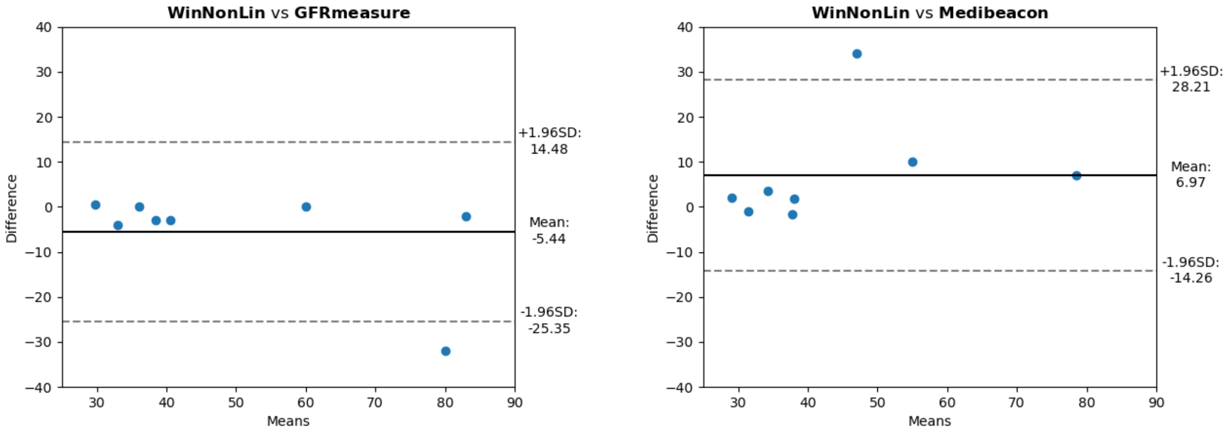
Whenever a file is saved, the device's name is automatically incorporated into the file's name, unless the user decides to rename it. The identifier of the device is two letters and six digits separated by a underscore. A total of 134 unique device identifiers were found. By examining the curves for each device it was noted that two devices were malfunctioning. Since these devices are very cheap to produce, it is more convenient to replace it than to try to fix it, hence the number of devices used.

2.4.3 Comparison with other solutions

For this comparison the fitting curves and the half-life times were compared between the software solutions. The data sample used with all three packages including WinNonLin was small with only eight samples. This was due to the limited time available Prof. Dr. med. David Czock was able to allot for this project.

First WinNonLin was compared with GFRmeasure and then with the Medibeacon software. Comparisons of two methods are often done by the Bland-Altman plots, which are also known as Tukey mean-difference plot. These plots allow to visually assess the agreement or concordance between two quantitative measurement methods or devices. It was introduced by British statisticians Douglas G. Altman and J. Martin Bland in 1983 [50].

When looking at Fig. 2.18a the mean difference over all half-life times between WinNonLin and GFRmeasure is -5.44 minutes. This not only indicates an overestimation of the half-life but shows how close the calculations are to WinNonLin. A similar results can be seen with the comparison with Medibeacon in Fig. 2.18b, only that range tends to be more towards the positive values like in the case of the mean differences of 6.97 minutes. This is due the fact that Medibeacon underestimates the times in compare to WinNonLin.



(a) (b)

Figure 2.18: Bland-Altman plot comparing half-life times of WinNonLin against GFRmeasure (a) and against Medibeacon (b). All values in both plots are in minutes. The y-axis shows the difference in half-life times of the compared solutions for each kinetic curve analyzed while the x-axis shows the mean value of the two. The dotted lines are the ±1.96 times the standard deviation. GFRmeasure overestimated the half-life times while Medibeacon underestimates them, as shown by the black line with the mean value on the right side. Both maintain the zero value between the ±1.96 SD interval.

To further compare Medibeacon with GFRmeasure, the R-square values of the fitting models where compared. Fifty curve were selected at random with the same distribution as the health classification as shown in Fig. 2.16a. The curves were first fitted with the default compartment models of each software and compared by the R-square error as seen in Fig. 2.19. GFRmeasure has a mean value of 96.57 and a median value of 97.94 in contrast to Medibeacon with mean and median value of 88.46 and 96.20 respectively. The two outliers at the bottom are curves of very sick animals. The labeled sample in Fig. 2.19 has been chosen to show the difficulties Medibeacon has to adapt to these kind of curves (see Fig. 2.20a).

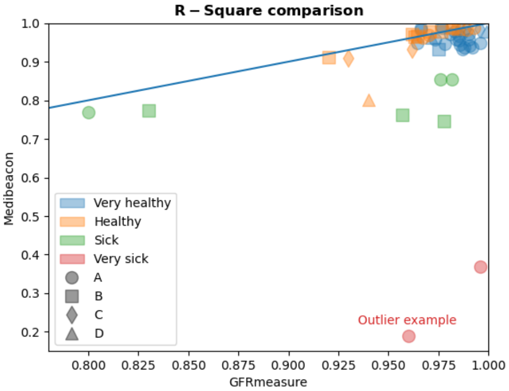
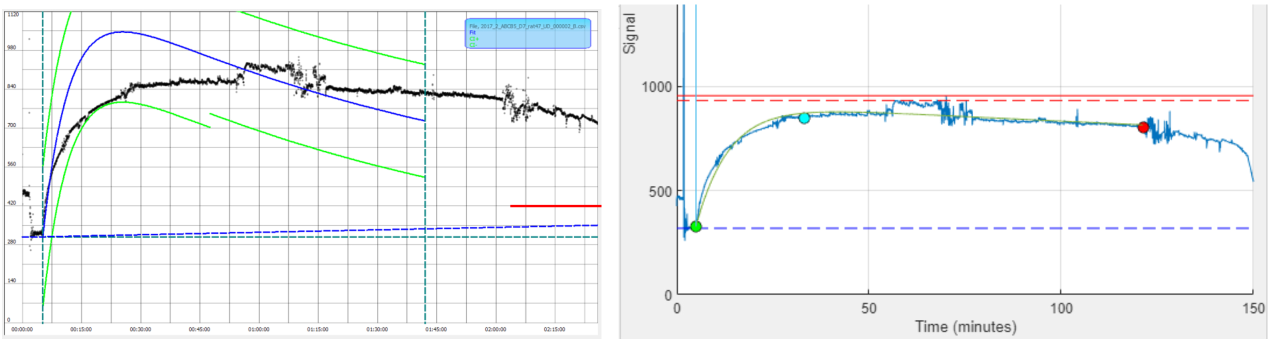


Figure 2.19: R-square error comparison between GFRmeasure (x-axis) and Medibeacon (y-axis), where the value range from zero to one, with one being a perfect fit. Blue line indicates the symmetry line, where the R-square error obtained by both methods is equal. Each ABCD scoring category has its own shape while the health has different colors as depicted by the legend. The labeled sample at the bottom has been chosen as an example shown in Fig. 2.20a.



(a) (b)

Figure 2.20: Comparison of model fitting on a very sick animal. More specifically, curve from the labeled sample in Fig. 2.19. Outlier from the curves used to compare Medibeacon (a) with GFRmeasure (b).

2.5 Discussion

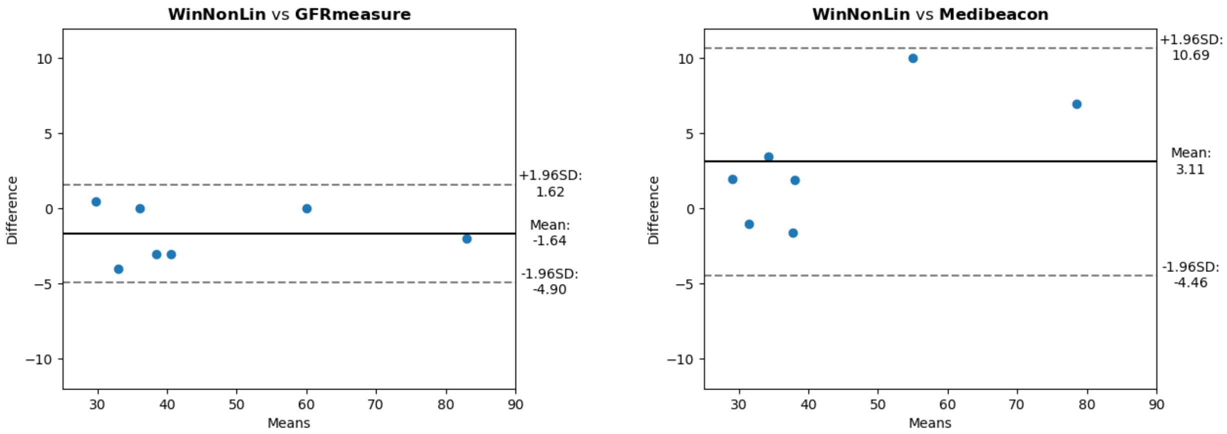
The distinction between low-frequency sampling through blood tests and high-frequency experiments involving infrared devices and fluorescent markers should ideally have minimal impact when comparing final outcomes like half-life times. However, substantial differences do emerge in terms of data acquisition, challenges, and execution processes. In this case, most facets involved were enhanced, encompassing the amelioration of animal well-being, reduction of the execution duration, and the implementation of high-resolution sampling. Conversely, new challenges surfaced, including the management of vast data volumes, distortions introduced by artifacts affecting the outcomes, device constraints, and unanticipated physical phenomena like quenching, among others.

This work offered solutions for the following challenges. It introduced accurate methodologies for processing data on a large scale and unearthing concealed associations through careful examination of curve distribution quality specific to a subject, like the identification of malfunctioning devices. Additionally, it facilitated the identification of issues in experimental setups for new users, whether through self-assessment or under the guidance of a supervising project overseer, leading the examination based on user profiles.

Another aspect involved analyzing the distribution of health in the newly introduced ABCD quality scoring system. It stands out that class B curves has the fewest occurrences of ”Very healthy” curves. A postulated argument for this behavior can be the fact that experiments featuring minimal marker concentration can’t reach a peak and therefore often fall under the B class. The opposite is also true where when excessive marker concentration is injected the device saturates classifying such curves as class D.

The study involved a comparison of the obtained results with other solutions available in the market, one of which can be considered the gold standard called WinNonLin. The sample size used to test the solutions is too small to establish statistical significance. However, upon closer examination of each individual example, the observed differences become noteworthy and meaningful. It is worth to highlight a notable distinction in the presence of the one outliers in the two Bland-Altman plots. While the presence of this outlier in GFRmeasure would not affect its classification as ”Very sick” of the kinetic curve, a different scenario unfolds with Medibeacon. For the kinetic curves, see Fig. 2.20.

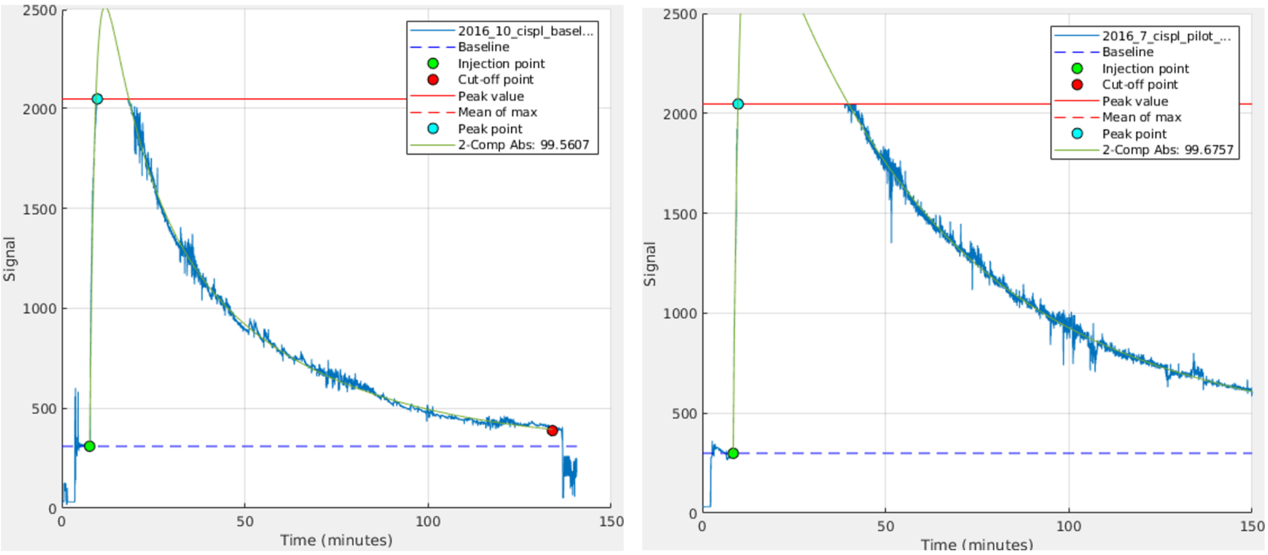
This can be observed as the lower data point in Figure 2.18a and the upper data point in Figure 2.18b. The half-life time calculated by WinNonLin was 64 minutes, whereas GFRmeasure yields a value of 96 minutes, both leading to the correct classification of the curve as ”Very sick.” However, Medibeacon significantly underestimates the half-life time, providing a value of only 30 minutes, resulting in the classification of the curve as ”healthy.” This discrepancy can be attributed to Medibeacon’s inability to accurately fit the model to exceptionally abnormal curves. It’s worth noting that although it might not be advisable, removing this outlier from both Bland-Altman plots results further in favor of GFRmeasure, leasing to the following outcomes:



(a) (b)

Figure 2.21: Bland-Altman plot comparing half-life times of WinNonLin against GFRmeasure (a) and against Medibeacon (b) without the outlier. All values in both plots are in minutes. The y-axis shows the difference in half-life times of the compared solutions for each kinetic curve analyzed while the x-axis shows the mean value of the two. The dotted lines are the ±1.96 times the standard deviation.

Another surprising discovery was the ability to calculate half-life times in curves under the class D, where the device saturation was reached, allowing for more results than previously imagined as shown in Fig. 2.22.



(a) (b)

Figure 2.22: Two examples of successfully fitting a two-compartment model on kinetic curve of the D class.

When comparing established tools, such as WinNonLin, to newer options, certain key features emerge that underscore the importance of increased automation, especially for researchers who are new to this field. Table 2.3 enumerates essential features that researchers at the Center for Medical Research in Mannheim, Germany, deemed significant, and it compares the three tools used based on these particular features.

Features - Programs	**MediBeacon**	**WinNonLin**	**GFRMeasure+**
Automation	✗	✗	✓
Batch processing	✗	✓	✓
Beginner friendly	✓	✗	✓
Classification	✗	✗	✓
Open-Source	✗	✗	✓
Live correction	✗	✗	✓
Robustness	✗	✓	✓

Table 2.3: Comparison of the most relevant features and the performance between the used software solutions. ✓ depicts the presents of the feature while ✗ represents a lack of it.

3

Morphological data analysis

The second part of this work focuses on morphological data analysis. Part of it is based on the methods and results were already published [51]. The individual steps taken to reach those results by means of the proposed approach called AutoCellSeg as well as additional methods and experiments are described in detail.

3.1 Background

In the laboratory, biologists often perform many tests under differing conditions based on cell or bacteria culture. Depending upon the type of experiment being performed and the effects under observation, these varying conditions typically lead to a huge accumulation of data. The manual analysis of a large volume of image data is very inefficient.

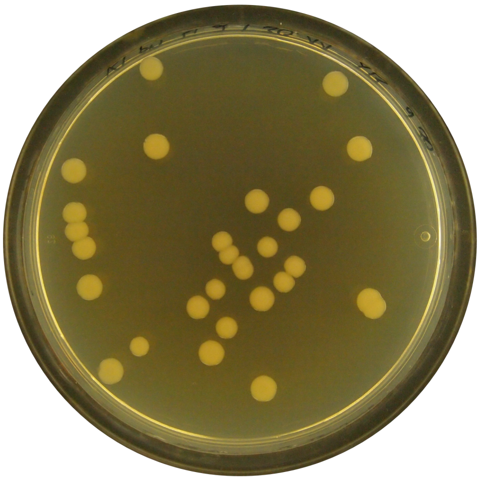


Figure 3.1: Typical petri dish with several colony forming units (CFUs). In this case from the *Klebsiella pneumoniae* family.

The image data produced during microbiological assays is often comprised of cells or colony forming units (CFUs) as shown in Fig. 3.1. It has therefore been a long-standing interest in automated colony segmentation and counting techniques by the microbiology community [52]. A variety of cell segmentation and counting methods as well as software solutions for this purpose have been proposed ever since.

Counting the cells or CFUs manually is expensive and time consuming but it is still the status quo. Besides the sheer count of the cells/CFUs, it is desirable in some experiments to quantify other aspects, for instance, to examine the change in the cell/CFU morphology and size according to the effects inflicted upon the cells/CFUs under both control and test conditions. To make a quantitative analysis based on other CFU features is a non-trivial task, and doing so is more complicated than simply enumerating all the colonies.

It becomes imperative to determine the size and number of CFUs/cells with great accuracy whenever the difference in the observed change is only marginal. Additionally the differences could have been caused not only by the difference between the control and the test, but instead by the experiment setup leaving users with unsatisfying or skewed results. This is especially the case with inexperienced users. This is why an automatic, high-accuracy analysis with a certain degree of quality management is highly desirable.

An algorithm that performs segmentation automatically, differentiating between the colonies themselves and the unwanted background, need to be able to obtain CFU boundaries. Using this segmentation, the counting of CFUs can be performed, and, in addition, features of interest such as size, shape etc. can be extracted. A typical CFU image analysis pipeline requires noise removal, image segmentation, feature extraction, feature selection and object counting/classification. Each algorithmic step has parameters which have to be appropriately set to obtain the targeted colonies in individual images.

Methods that require abstract a priori knowledge about colony characteristics such as size in pixels, approximate number count, metrics of shape and other features will restrict its practical use. This is especially true when analyzing entire data sets where colonies in individual images may vary greatly from one another in size, morphology and intensity, requiring the user's input each time. Instead the parameters should, from a user's standpoint, be asked once and be intuitive, such as setting the minimum and maximum size of colonies by graphically clicking on them in the image. The evaluation throughout entire datasets should be adaptable to facilitate the process and minimize the user interactions.

3.2 Related Work

Significant problems hindering a good CFU analysis include high CFU density, imperfections and impurities in the suspension medium, inherent background acquisition noise, inconsistent illumination artifacts, luminous reflectance and other visible artifacts on the container's boundary, close proximity between CFUs or with container boundaries and low image resolution. There are many different methods to tackle the complex task of segmenting and analyzing CFUs, such as deep learning, fuzzy algorithms, circular Hough-transform to name a few. The following review of related work focuses on existing comprehensive software solutions rather than methodological approaches to individual steps in the required processing pipeline.

Only some commercial tools, such as UVP ColonyDoc-It™ (Analytik Jena AG, Jena, Germany) [53], STEMvision™ (STEMCELL Technologies Inc., Vancouver, Canada) [54] etc. are available. Due to the limited market, these commercial solutions are expensive. While these commercial tools offer a graphical user interface (GUI) to help guide the user, it is often the case that the underlying algorithm stays undisclosed.

CHiTA [55] and NICE [56] are two examples of open source CFU counting solutions implemented in MATLAB (The MathWorks Inc., Natick, Massachusetts, USA). Although MATLAB is a commercial software, programs based on it are quite popular as MATLAB offers an immense library of already existing well documented methods that can be deployed free of charge on all major operating systems (Windows/Mac/Linux) using their run-time compiler.

CHiTA applies segmentation based on circular Hough-transform and makes it possible to discard eccentric segments. NICE employs a combination of extended minima and thresholds, but is not suited to handle CFUs that differ significantly in size and shape as compared to the ones given in their sample data. Additionally, these solutions are not able to detect the container boundaries enclosing the CFUs and thus consider the whole image as the searching area for CFUs. Although additional cropping can improve the outcome, many erroneous segments are detected nevertheless. Another MATLAB-based software [57] is based on a combination of an adaptive and a global threshold. This method was developed to work in conjunction with a special hardware for image acquisition, which allows for uniform light settings. However, the provided hardware allows to process on single images only and it is bound to perform worse for data sets acquired other than in this specialized setting.

Open source tools enable the user to write or adapt programs to their own need. A very popular choice is OpenCFU [58], which executes quickly and does so on a variety of images with simple parameter settings. Additionally, results with OpenCFU show very few false positives. Due to its restrictive CFU selection criteria, however, it is prone to underestimate the number and the size of colonies, as it searches for only round objects. OpenCFU is written in C++ using OpenCV and is stand-alone application available for both Linux and Windows operating systems, with compilation instructions for GNU/Linux only [59].

Another popular open-source software is called CellProfiler [60] that allows users to define and share custom-made pipelines. One of which was constructed in CellProfiler by Choundhry et al. called Cell Colony Counting [61]. It is used to detect, count and quantify CFUs. Another commonly used platform with its own GUI is ImageJ [62], which is Java-based. Cai et al. provided a macro, commonly known as IJM, that works with ImageJ to analyze CFUs [63]. Another macro based on edge detection algorithm called Cell Colony Edge was developed by Choundhry et al. mentioned earlier in ImageJ to adapt to particular images [61].

Machine learning-based approaches have become popular in recent years [64]. Ilastik [65] is a beginner-friendly framework to classify image pixels into user-defined classes. This is done with an interactive graphical pixel labeling step which allows the user to select individual pixels and Ilastik classifies the rest of the image based on that selection. Even though it is not restricted to a specific algorithm it uses mainly a Random Forest classifier [66]. Users can select from 37 predefined features for the classification process. A specific solution for bacteria colony counting based on convolution neural networks was proposed by Ferrari et al. [67]. It is able to classify colony conglomerations into different classes containing at least one and up to six colonies per aggregation. Additionally it uses an outliers class for rejecting false positives to improve the accuracy of the results. This solution requires partial retraining on other datasets by creating a ground truth mask for some of the new data for evaluation and testing. A missing element, however, is that it does not extract and compare colony features and it does not offer an option for adapting the parameters for experienced users to improve the results in cases where it is needed.

FastER is another software package that claims to provide a user-friendly, fast and robust cell segmentation [68]. It makes use of maximally stable extremal regions [69] for blob detection and estimates the likelihood of the detection being correct by applying a support vector machine (SVM) with a Gaussian kernel. FastER is programmed in C++ and provides a stand-alone GUI, which is freely available to the public. Unlike some of the aforementioned software systems, fastER tends to yield a lot of false positives when there is a colony container visible in the image data. Additionally, it has no option for any parameter selection, nor for post-processing the obtained results.

Automatic dish extraction is rarely used [70,71], which in these cases was achieved by applying a contrast-limited adaptive histogram equalization in combination with morphological filters, often leading to false positives if post-filtering was not employed. Other programs may offer manual methods such as creating a circle by selecting 3 points along the dish [72] or like in the case of CellProfiler several different pre-defined shapes are available for cropping most petri dish silhouettes.

To generalize a segmentation program otherwise specific to an experiment or setup is burdensome, as one must have the will, expertise and time to adapt the existing solution to various experiment settings. Many software solutions for CFU detection suffer from a lack of quick and intuitive parameter selection. Moreover, even when a program does offer a more extensive parameter selection process, it still demands a wider user knowledge to be able to take advantage of all its capabilities. There is an undeniable need for more user-friendly software that is appealing to advanced and normal/new users alike.

Unknown acquisition conditions and object locations can furthermore adversely affect the results, even with finely-tuned parameters. Therefore, it is desirable that an interactive, flexible, intuitive and responsive correction step is integrated in any image analysis pipeline. To locate objects in an image, one could use feedback-based segmentation mechanisms, with an intuitive and interactive selection of a priori features. It is often helpful to have a comparison in cell/CFU count and size without the need of transferring the results to a different tool. Therefore, one aim of this work was to develop a software solution that contains interactive and easy-to-use feature selection for post-segmentation editing and analysis to address the mentioned problems.

3.3 Material & Methods

This chapter presents the datasets used for the characterization and evaluation of the proposed method and software solution AutoCellSeg [51]. It also lists the programs used for comparison with AutoCellSeg for each dataset. Afterwards it explains AutoCellSeg in more details with the various approaches used to overcome problems, automatize the program and surpass the competition. This is followed by a quality management method developed to insure a correct execution of an experiment.

3.3.1 Datasets

For this work several datasets were used for an in depth comparison between the software solutions. In order to show the robustness and applicability of AutoCellSeg, it is reasonable to try datasets of well-segmented images with different foreground and background. Moreover, it is necessary to run AutoCellSeg in different operation modes, in order to demonstrate its full performance.

Dark background

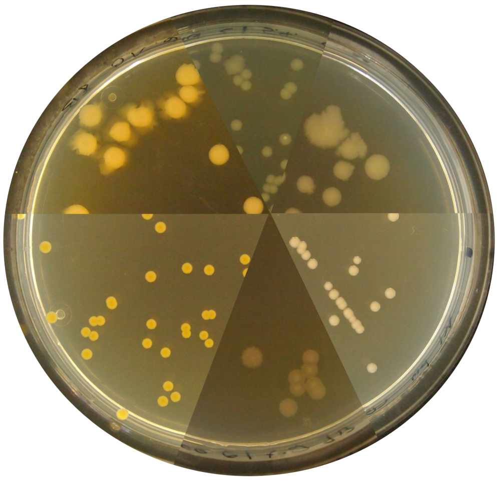


Figure 3.2: A collage made of six out of the twelve images collected. Two from the *E.coli*, two from the *Klebsiella pneumoniae*, one from the *Pseudomonas aeruginosa* and one from the *Staphylococcus aureus* images were chosen to showcase the vast spectrum of variability in the images.

Twelve images, acquired with the support of Prof. Dr. Matthias Mack and Jasmin Baumgartl at the laboratories of Mannheim University of Applied Sciences, were labeled as ground truth. The images are comprised of three images each of different bacterial species including *E.coli*, *Klebsiella pneumoniae*, *Pseudomonas aeruginosa* and *Staphylococcus aureus*, part of which can be seen as a collage image (see Fig. 3.2). For all cases the background is darker than the colony itself. The images have a resolution of 4032 pixels in width and 3024 pixels in height and were labeled in Adobe Photoshop (Adobe Inc., San Jose, California, USA), by delineating boundaries of colonies. Afterwards, via MATLAB, masks were created by filling out the boundaries and labeling them as binary. These ground truth images are then used to extract the colony count, and the size of each individual colony in pixels [73]. For this data set AutoCellSeg was compared against OpenCFU, CellProfiler as well as the combination of the latter with Ilastik in batch and individual mode.

Inverted background

Nineteen high quality images [74], provided by Quentin Geissmann at the OpenCFU project website make up the inverted background benchmark. This data set is based on *Staphilococcus aureus* colonies on luria broth agar plates. Each of the nineteen images have a lighter background than the colony itself. The images have a resolution of 1538 pixels in width and 1536 pixels in height. Four images were selected to show some examples of the possible colonies distributions found in this data set, as seen in Fig. 3.3. This data set was used to compare AutoCellSeg with OpenCFU on the data OpenCFU was trained for. Unfortunately no ground-truth data was provided alongside the images. For this reason the results of both programs were compared directly to each other.

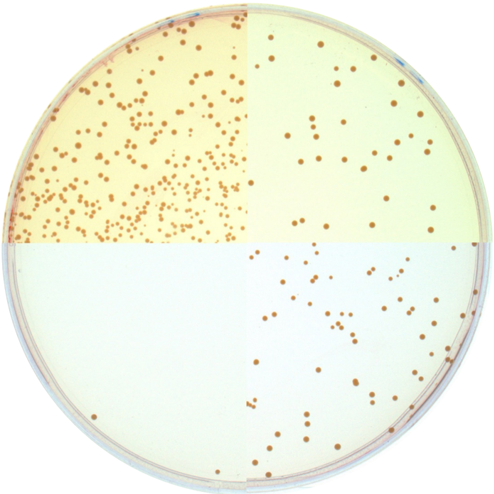


Figure 3.3: A collage made of four quarters out of the nineteen images. The chosen images are number 3, 6, 12 and 19.

Batch processing

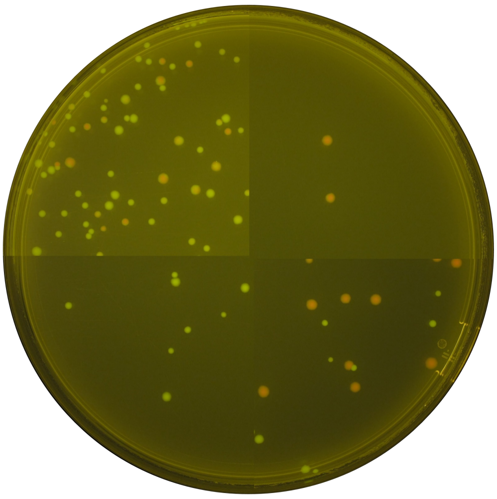


Figure 3.4: A conglomeration of four quarters out of the fourty-nine images. The name of the chosen images end with 09, 14, 18 and 23.

Another fourty-nine high quality images, available at the OpenCFU website [75], were used in this work. The data set contains images with a resolution of 2592×1944 pixels of *E.coli* colonies on agar plate expressing a red and a green fluorescent protein. The color of the agar plate background has also a greenish tone and the background beyond the plate is black. To show the variation between images in the data set four images were selected as shown in fig. 3.4. This data set was primarily chosen to evaluate the performance of AutoCellSeg in batch processing mode (fully automated option) by comparing the results with OpenCFU. Unfortunately no ground-truth data was provided alongside the images. For this reason each CFU in each image was counted manually and the results of both programs were compared directly to each other.

Fluorescent cell segmentation

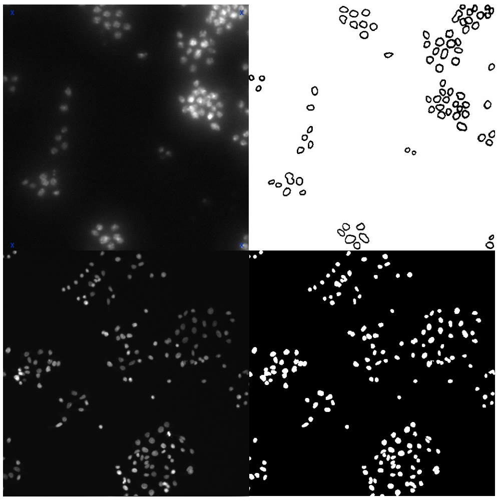


Figure 3.5: Left side shows one example of each data set and on the right it's respective mask. The image from the top is from *BBBC007* while the bottom one is from *BBBC008*. Their respective names are AS_09125_040701150004_A02f00d0 and AS_09125_050116000001_A24f00d0 _slice1_ channel1.

Among the existing data sets with ground truth offered by the Broad Bioimage Benchmark Collection from the Broad Institute [76], two data sets were used. One of the two data sets is referenced as *BBBC007* which is a collection of drosophila Kc167 cell images and the other data set referenced as *BBBC008* contains human HT29 colon-cancer cells images (see Fig. 3.5). While the latter has a total of twelve images with a resolution of 512×512, the former data set has a total of sixteen images with various resolutions: eleven have a resolution of 400×400 pixels, four have a width and height of 450 pixels and one has a resolution of 512×512 pixels. For both data sets, only the images of the first channel (DNA) were used for the comparison between AutoCellSeg and OpenCFU while excluding the third channel which stains actin in the cytoplasm. While the *BBBC008* had already the ground truth images as masks, the other had black and white ground truth images where the cell boundaries are marked in black while the rest is white [77]. To create masks from the ground truth images provided as seen in the top right corner of Fig. 3.5, the background was removed by using the built-in MATLAB function *imclearborder*. Additionally, the outlines when thicker than one pixel was not taken into account and no other changes were made to the ground truth images. The purpose to use these data sets is to demonstrate the performance of AutoCellSeg on a segmentation challenge containing animal and human cells.

Randomly selected images

AutoCellSeg was put through another performance test with three additional images from different published experiments to see the performance on images unrelated to the once previously tested. Two images and their respective manual count of the colonies for a comparison were provided by Choundhry et al. [61]. One of which is a low resolution image, with only 246×236 pixels, of twenty-two bacterial colonies and the other an image of fifty-one clonogenic colonies with a resolution of 1028×1000 pixels. The last image was taken randomly from a data set containing aggregates of one to six colonies of bacteria on blood agar plates provided by Ferrari et al. [67] and compared with the provided label.

CFU growth over time

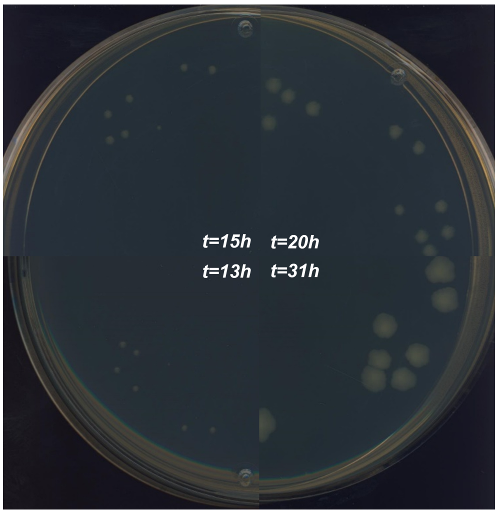


Figure 3.6: Four pictures taken from one of the CFU growth over time sets. This set is with fewer colonies and the images are taken at random petri dish position. From the bottom left clockwise the times are 13, 15, 20 and 31 hours as shown in white in each partial image, making it into one collage.

For biological experiments it is of high importance to know if the experiment setup was done appropriately to ensure that the results show what the scientist expected it to be instead of being a skewed result because of external causes. A way of doing this is to monitor cell growth over time. A quality management data set containing various images of the same petri dish recorded over a certain period of time was created in the laboratories of the Medical Research Center with the help of Yifei Mu. The images of this data set have all different sizes and resolutions. They are cropped out in a square shape with a resolution of approximately 1750 pixels on each side. The background is of a darker bluish gray color inside the petri dish (the medium) and almost black outside the petri dish. The *E.coli* colonies are a faint change to the medium color with a yellow-green touch. Thirteen sets have images which were taken at 8, 12, 23, 27, 31, 35 and 46 hours with similar position of the petri dish. Twelve sets have images taken hourly starting from 11 hours to 20 hours and one at 31 hours with random position of the petri dish. As an example of a set with fewer colonies, four pictures at 13, 15, 20 and 31 hours, were made into a collage which follows the same colonies of five and two as can be seen in Fig. 3.6. The amount of colonies in the images may vary from just a couple to hundreds. The purpose of the data set is solely to follow the growth of each colony separately. The reasons can be for example confirming the experiment is set correctly or understanding if bacteria have different growth curve depending on their neighboring surroundings.

3.3.2 Overview of AutoCellSeg

AutoCellSeg [51] implements a robust image segmentation method, can be applied in different operation modes, enables the user to select intuitive object features to guide the segmentation process, allows the user to correct results with a graphical interface and provides results statistics and diagrams. In this section, AutoCellSeg is outlined from a user perspective. The following section presents the algorithmic details and design decisions.

Using AutoCellSeg is as simple as selecting an individual or multiple images as well as entire folders of images without having a restriction in number. The user subsequently chooses a process depending on the level of interaction with AutoCellSeg, see Fig. 3.7.

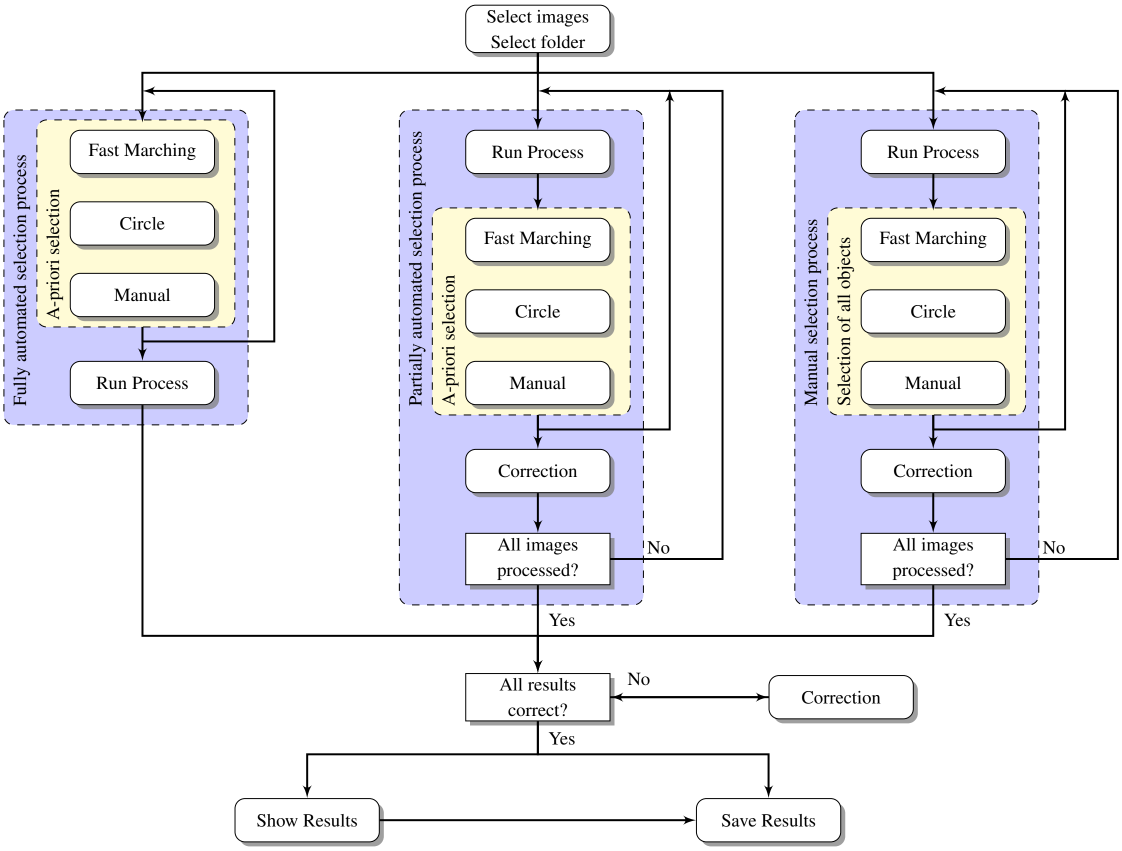
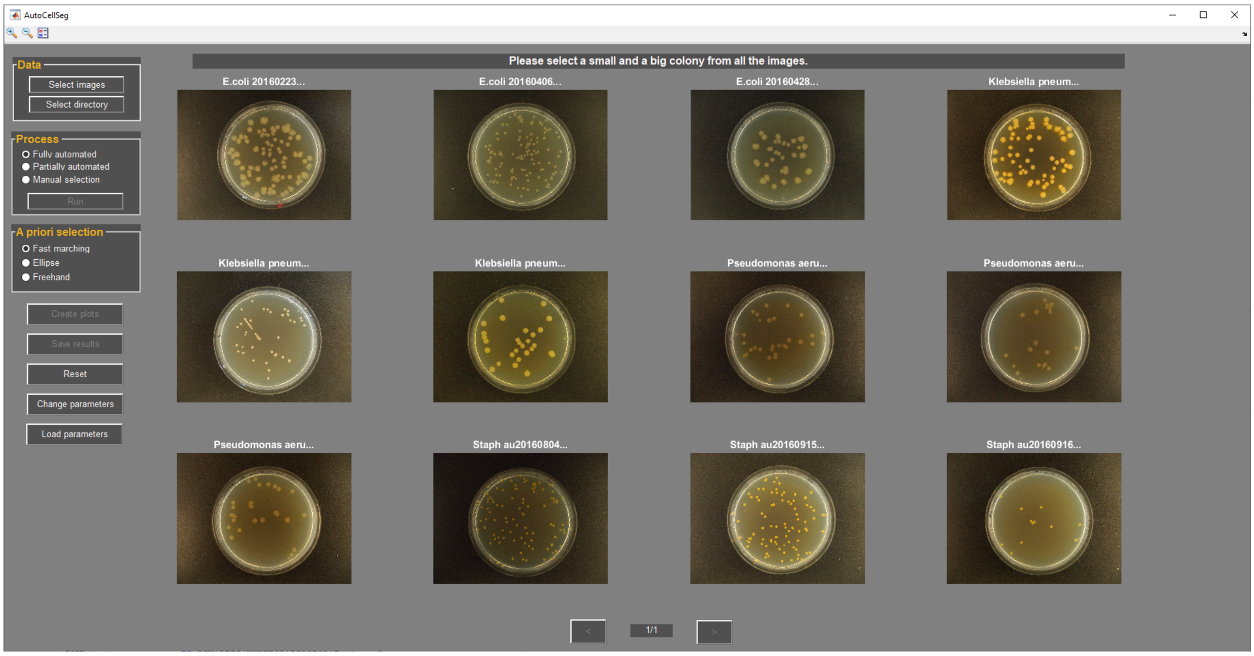


Figure 3.7: The pipeline for AutoCellSeg showing each individual process depicted by using a blue colored box. The left column represents the “fully automated” process, the central path shows the “partially automated” process and the right one illustrates the “manual labeling” process. The yellow boxes inside the blue boxes indicate the different methods that the user can choose from to select the a priori information. Some of the processes allow for a correction after each image but all allow for the user to correct the results by adding new or removing unwanted segments at the end of the entire process. Once the user is satisfied with the outcome, plots can then be created and saved for comparisons between experiments. The image was taken unchanged from the work already published [51].

The “fully automated” process having the least and “manual selection” having the most interaction steps. The only a priori information needed for “fully automated” consist of the selection of the approximately smallest and biggest CFU/cell between all images before the complete evaluation can begin without further user input. Similarly, the a priori information must be selected prior to the “partially automated” process, but instead of using that information for all the images it is selected for each image before continuing to the next. For the “manual selection” the user has to select all the objects, for example CFU or cells, individually for all the images. A correction is possible in all three modes, either after each image for the case of the “partially automated” and “manual selection” processes as well at the end of all three procedures.

Once the results are satisfactory, the user can let AutoCellSeg create a plot showing the change of total colony sizes, an overlaid kernel density estimation (KDE) plot as well as a blob count plot for a comparison between both test and control images in each individual case. The categorization between test and control is purely linked by their file name. At this stage the user can save all of these plots along with all the parameters used, the binary masks, the outlined overlaid images and two cvs files containing all the colony features such as area, eccentricity, radius and mean intensity.



Please select a small and a big colony from all the images.

Figure 3.8: The interactive AutoCellSeg GUI after loading images from the dark background data set.

AutoCellSeg with its GUI, shown in Fig. 3.8, was programmed entirely with MATLAB and tested for all major operating systems: Windows, Ubuntu and MacOSX. The tool, the created benchmark with the results as well as the entire source code are open source and freely available at GitHub [1]. AutoCellSeg has few requirements such as the required MATLAB runtime being installed and the images to be of the common image format.

3.3.3 Image Analysis Pipeline

The following sections describe the individual sequential steps in details with pictures showing the change in each step, some of which are explained more elaborately since they were part of a new development resulting from this work. AutoCellSeg is based on fuzzy a priori size and optionally intensity information extracted from the user's object selection using one of the three available methods (fast marching, ellipses or freehand drawing). A supervised multi-threshold image segmentation step makes use of this a priori information. The image is then pre-processed based on channel information. Afterwards it is passed on to an adaptive threshold segmentation step, where a binary mask is created containing the individual detected blobs. All the segmentation results, created by the different intensity thresholds, are overlapped iteratively over a search space of successive threshold values between 0 and 1. A plausibility check is performed over the resulting image where any blob having a size outside the size derived from the fuzzy a priori knowledge is either removed if smaller or if bigger passed onto feedback-based watershed segmentation step, adapted from Vincent et al. [78]. The feedback-based watershed segmentation adapts the value of H-maxima transform for regional maxima of the image, which are then used as seed points for watershed segmentation. After this step, the user is then able to correct the results if necessary, which is done with slightly different fast marching methods.

---

[1] https://github.com/AngeloTorelli/AutoCellSeg.Last visited: 15.09.2023

Fuzzy a priori information

For most software solutions, once the image is loaded, a priori information is required, making the method supervised. Currently, popular solutions with or without GUIs like CellProfiler, OpenCFU or ImageJ require the user input of the a priori information values, which are given manually or by moving a sliding bar, making it not intuitive for the user. Most users have no prior knowledge about the parameters such as precise size in pixels and circularity factor, which can be challenging to estimate even for an expert user.

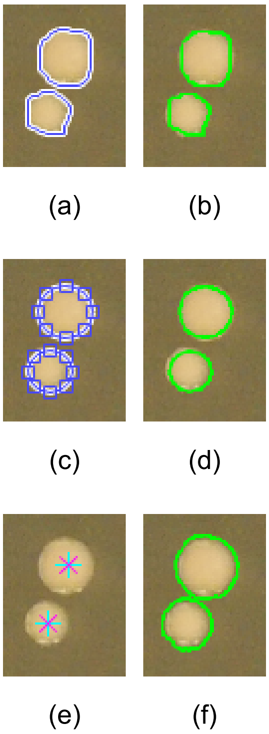


Figure 3.9: The three rows of images show the freehand, the ellipse and the fast marching selection process respectively and their results.

In AutoCellSeg a more intuitive approach is offered, by letting the user manually select the CFU/cell graphically instead of the demanding trial and error search of optimal a priori information input. For this, the user has the choice between three interactive methods (see Fig. 3.9 for an overview). The first two are by freehand drawing (Fig. 3.9a) or by overlaying an ellipse around the border of a CFU/cell (Fig. 3.9c) making those regions into a binary mask. The last available method, the default method, is just by simply clicking on the approximate center of an CFU/cell (Fig. 3.9e) in the image displayed in the GUI.

The user selected points inside the object boundaries are used as seed points for a fast marching segmentation [79], by applying the MATLAB function imsegfmm, to create a mask. For cases where the colonies are similar to the background it is better to increase the contrast. For that reason, independently of the chosen method, a weight array based on gray scale intensity differences is applied on the image using the built-in MATLAB function graydiffweight.

Additionally AutoCellSeg allows the user to remove the previously selected seed point in case of an accidental mistake by using the backspace or the delete button. The recommendation is to select at least one of the smallest and one of the biggest alone standing blobs with average shape and color. Here, the overview of all images in the AutoCellSeg main window can serve its purpose to search for images containing such criterion. In case of uncertainty a larger selection is advantageous to make the outcome more accurate. The a priori information of minimum and maximum area size is extracted along with the average eccentricity from the selected blobs. An example of a priori size extraction using fast marching is shown in Fig. 3.10a.

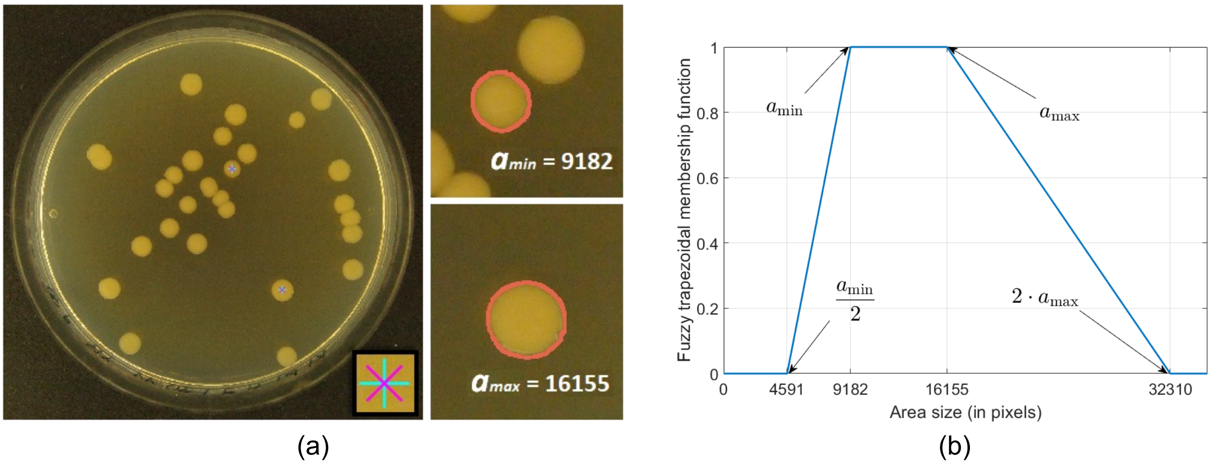


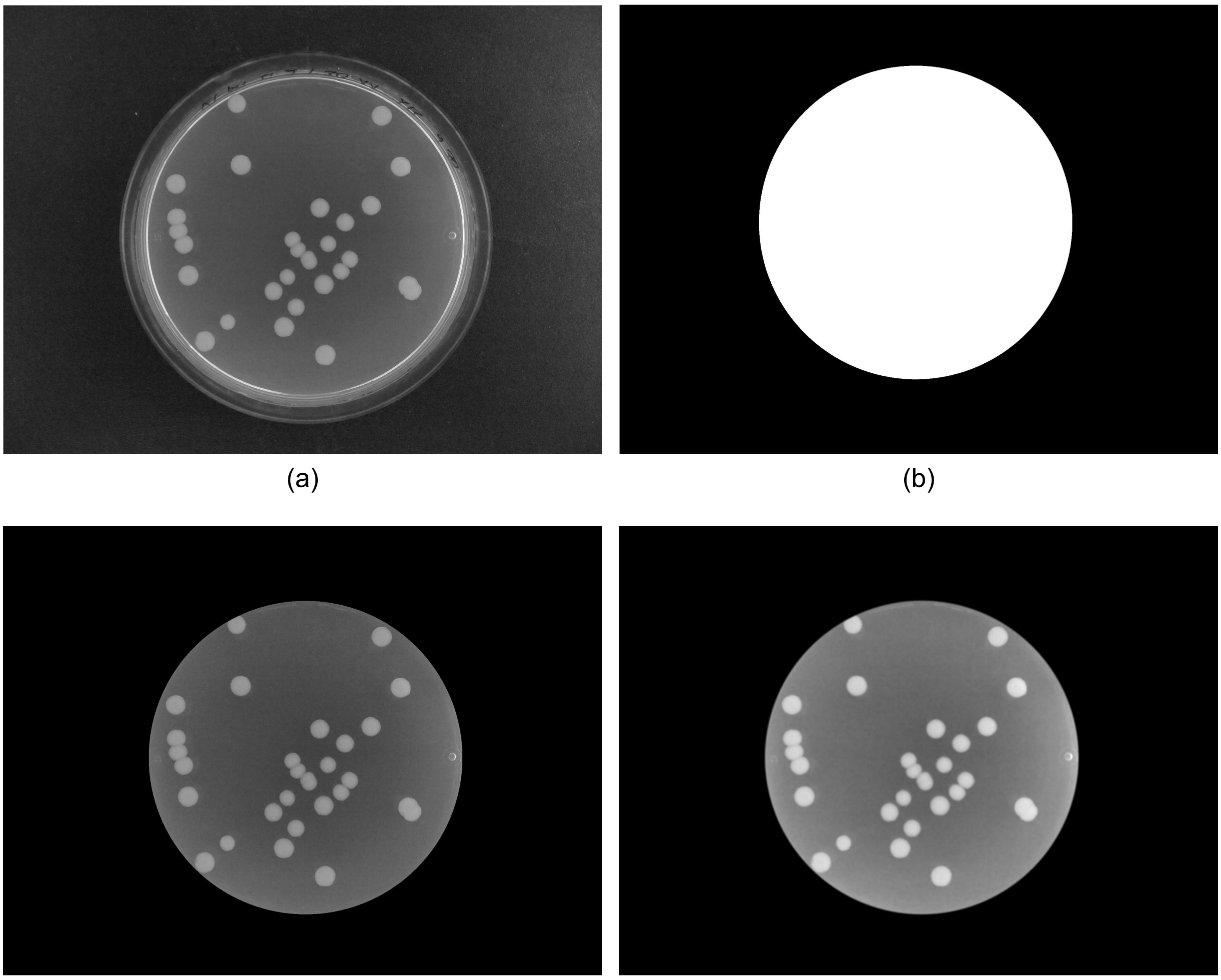
Figure 3.10: The extraction process of a priori information from the AutoCellSeg GUI. The image on the left in (a) shows the two colonies selected from the user. The center of asterisk marker as shown magnified on the bottom right corner is the user click position. The two smaller images on the right side show the minimum a_{min} and maximum a_{max} colony sizes using the fast marching method indicated by the pink colored delineation. (b) is an example of a fuzzy trapezoidal membership function. In this case the resulting edges of the size trapezoid are indicated by black arrows are equal to $(\frac{a_{min}}{2}, a_{min}, a_{max}, 2 \cdot a_{max}) = (4591, 9182, 16155, 32310)$.

Optionally the intensity information is also taken into account. For both size and intensity a fuzzy trapezoidal membership function is used as filtering criteria. The edges of the trapezoid are half of the minimum, minimum, maximum and twice the maximum size as presented in Fig. 3.10b. Similarly for intensity the trapezoid has almost the same edges: half of minimum, minimum, maximum and one and a half times the maximum capped at 1. When the a priori information is calculated, the image is pre-processed before being segmented.

Image pre-processing

The pre-processing stage can be explained in several steps as follow. First, in case of the presence of multiple channels the image is converted into gray-scale (see Fig. 3.11a).

The conversion to gray-scale can be manually modified, between a specific channel or the mean of all channels (default) in the option mask. The image is normalized between 0 and 1 as follows: $\frac{p_i - p_{min}}{p_{max} - p_{min}}$ where each intensity pixel value of the image is represented by p_i and p_{min} and p_{max} are equivalent to the minimum and maximum intensity values of the whole image.



(a) (b)

(c) (d)

Figure 3.11: An example to show some of the steps of the pre-processing stage. The image is converted to a gray-scale image (a). By default AutoCellSeg extracts the petri dish as a mask (b) and applies it to the image to have only the contents of the petri dish (c). A Gaussian filter with the default size of 5 by 5 is applied (d).

If a petri dish is in the image, AutoCellSeg will by default recognize it and only concentrate on the inner content of it. This is done by searching the most occurring intensity value iv, usually being the medium, and using it to create two slightly different masks. One mask is created with values below $iv + 0.01$, while the other with values above $iv - 0.01$, which, combined with a logical AND operator, result into only the inner region of the petri dish.

This way the information regarding the inner radius of the petri dish is extracted to create a perfect circle inside the petri dish (see Fig. 3.11b). This is by default activated and can be turned off in the option, but if left activated it will be multiplied with the image (see Fig. 3.11c). Afterwards a Gaussian filter is applied to minimize the noise in the medium before the background removal step (see Fig. 3.11d), which by default is of size five by five but can be also be changed in the options.

Background removal is performed using morphological top hat filtering with a disk of radius ninety as a structuring element (see Fig. 3.12a). At this stage, if activated by the user, a contrast-limited adaptive histogram equalization (with the help of the already existing MATLAB function *adapthisteq*) is used to enhance the contrast of the image by transforming the intensity values on small regions, rather than the entire image (see Fig. 3.12b). Optionally, one can additionally use an edge detection step which is then combined with the previous steps. From the resulting image, the lower and upper boundaries of the intensity histogram are clipped to the 1% and 99% quantiles, respectively before being passed to the multi-threshold segmentation step.

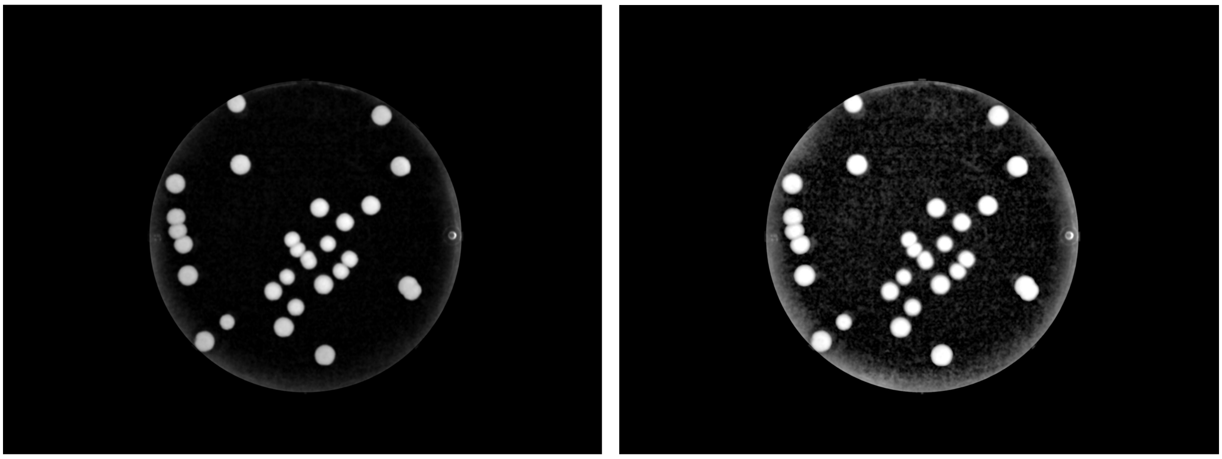
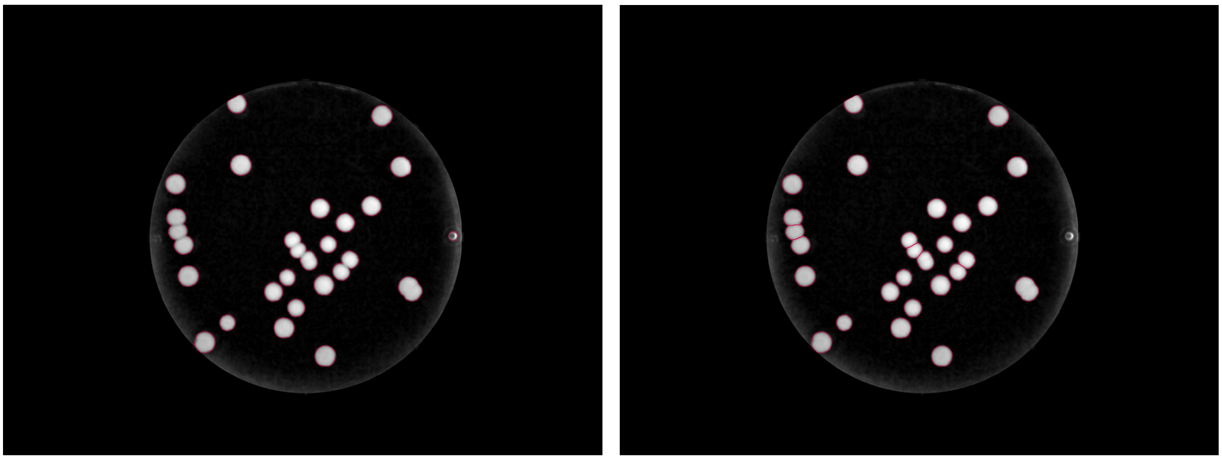

(a) (b)

Figure 3.12: Subsequent steps of the pre-processing stage. The background is then removed by using a morphological top hat filtering with a disk of radius ninety as a structuring element (a). A contrast-limited adaptive histogram equalization method is used to enhance the contract of the image (b).

Multi-threshold segmentation

An automatic multi-threshold segmentation is then applied over an intensity search space (see Fig. 3.13a). By removing the background and the petri dish, it not only helps diminishing the false positives, but also increases the sensibility of the intensity search space for relevant objects by removing unnecessary low and high intensity values of the background and reflective petri dish. The size of an incremental step, which by default is 0.1, can be varied manually in the option dialog to alter the outcome. The lower the value the finer the results but at the same the longer the processing time. At each step a segmentation image is obtained and each blob is checked for plausibility. Blobs with a size outside the fuzzy trapezoidal membership function are filtered out. After all the threshold levels are executed, the results are combined using the logical OR operator. The image is now segmented into individual blobs and clusters where the latter need to be processed further to split them into individual clells/CFUs. This is done with a feedback-based watershed segmentation, which is explained in detail in the next section. The only exception is with the graphically selected segments, which are kept unchanged if the new segmentation area only deviates with approximately 10% of area encapsulated by initial segments.



(a) (b)

Figure 3.13: Image (a) is the result of the multi-threshold segmentation step and image (b) is the result of the feedback-based automatic watershed segmentation step using the same example as in the image pre-processing pipeline.

Splitting of conglomerates by feedback-based automatic watershed segmentation

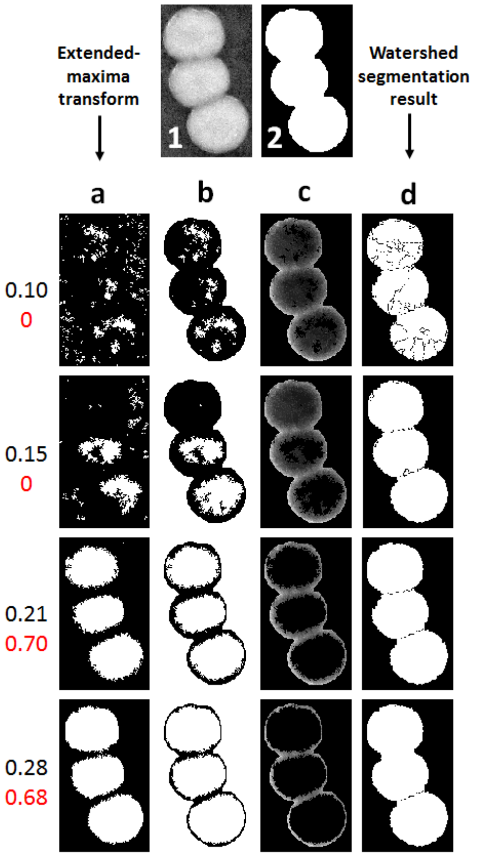


Figure 3.14: The parameter adaptation for h maximizing the quality criterion Q. Image 1 and 2 represent the original grayscale image of the chunk of colonies and the initial blob detected respectively. Four different H-maxima transforms with $h = (0.1, 0.15, 0.21, 0.28)$ were selected shown in black with their respective quality of segmentation $Q = (0, 0, 0.7, 0.68)$ in red. Column (a) shows the result of applying extended-maxima transform followed by a hole filling operation. Column (b) shows the representation of an OR operation between the selected seed points image with the complement of the mask. Column (c) shows the result of imposing the complement of the previous mask on the greyscale image. The final result of applying watershed algorithm from the image obtained from preceding step is shown in the last column (d). The result with the highest value of Q is selected as the final segmentation (second last row).

A similar method [80], where it was applied to adapt intensity threshold for segmentation, has been modified for this work. The watershed segmentation [78] is useful in separating conglomerates of blobs that satisfy the criteria of eccentricity greater than 0.5 and total size greater than the maximum CFU/cell area. The watershed segmentation separates a bigger bundle based on the defined catchment basins or the seed points using the intensity information where dark and light pixels represent low and high elevation respectively (see Fig. 3.13b). However, detecting the seed points may not be straightforward as the objects of interest to be segmented have varying intensity maxima. Here, the seed points are controlled by regional maxima with a height of at least h using the H-maxima intensity transform, which may vary the produced result greatly depending on the chosen intensity value. Regional maxima refer to connected groups of pixels that share a consistent intensity value, and where all the outer boundary pixels possess a lower intensity value. The regional maxima of the H-maxima transform is called extended-maxima transform. Therefore, the range can be selected manually in the options (default is from 0.1 to 0.3 with steps of 0.01) and are tuned iterative by feedback-based information using a quality criterion at each step. The quality criterion Q is the multiplication of three fuzzy trapezoidal membership functions for evaluation of the expected count, the eccentricity and the size. The edges of the three function are: count being [1, n_x, $2 \cdot n_x$, $3 \cdot n_x$-1] with n_x being the size of the cluster divided by the minimum blob size, eccentricity being [0, 0.1, 0.5, 0.85] and size being set same as before $[\frac{a_{min}}{2}, a_{min}, a_{max}, 2 \cdot a_{max}]$. The end result uses the optimal parameter, which has the highest quality value. Based on the left conglomerate in Fig. 3.13 a detailed example of inner process of the feedback-based parameter adaptation for the watershed segmentation is shown in Fig. 3.14.

Interactive correction of segmentation

Even with the best software, results may lack precision, which is why a correction feature should always be offered. At various time points in the available type of segmentation processes, the user has the chance to correct the results of each image. Either after each image is processed in “semi” and “manual” segmentation or at the end of “fully automatic” by clicking on the image thumbnail displayed on the main GUI. The image is then opened in a new window where the user is able to add undetected or remove erroneous segments. Both are very important, non-trivial post manipulation features, that only AutoCellSeg is offering.

In the following, the most occurring challenges in segmentation will be mentioned as a showcase of how flexible the post correction features are. An overview of all challenges can be seen in Fig. 3.15. These challenges can all be broken down into false positives and false negatives.

The typical false positives are falsely recognized colonies that only need to be deleted. When examining clusters, false positives can be wrongly separated bundles or segments that are falsely identified as conglomerates, which usually occur when the intensity of a CFU/cell varies a lot or when its size is bigger than the average. Another type of false positive is additional recognition, which is when some background is additionally recognized to an CFU/cell as part of one.

False negatives are colonies that are only recognized partially or are not recognized at all, such as in the case of extremes, where colonies could be mistaken with the similar intensity of a background such as the petri dish border. A conglomerate without any separation at all can also be interpreted as a false negative.

The user can add seed points where they need to enlarge the borders or delete and add the cell again in cases where a single CFU/cell is not segmented perfectly. For all problems with conglomerates, the user needs to delete all the falsely recognized colonies if any and add as many seed points as the number of colonies forming that conglomeration in their center.

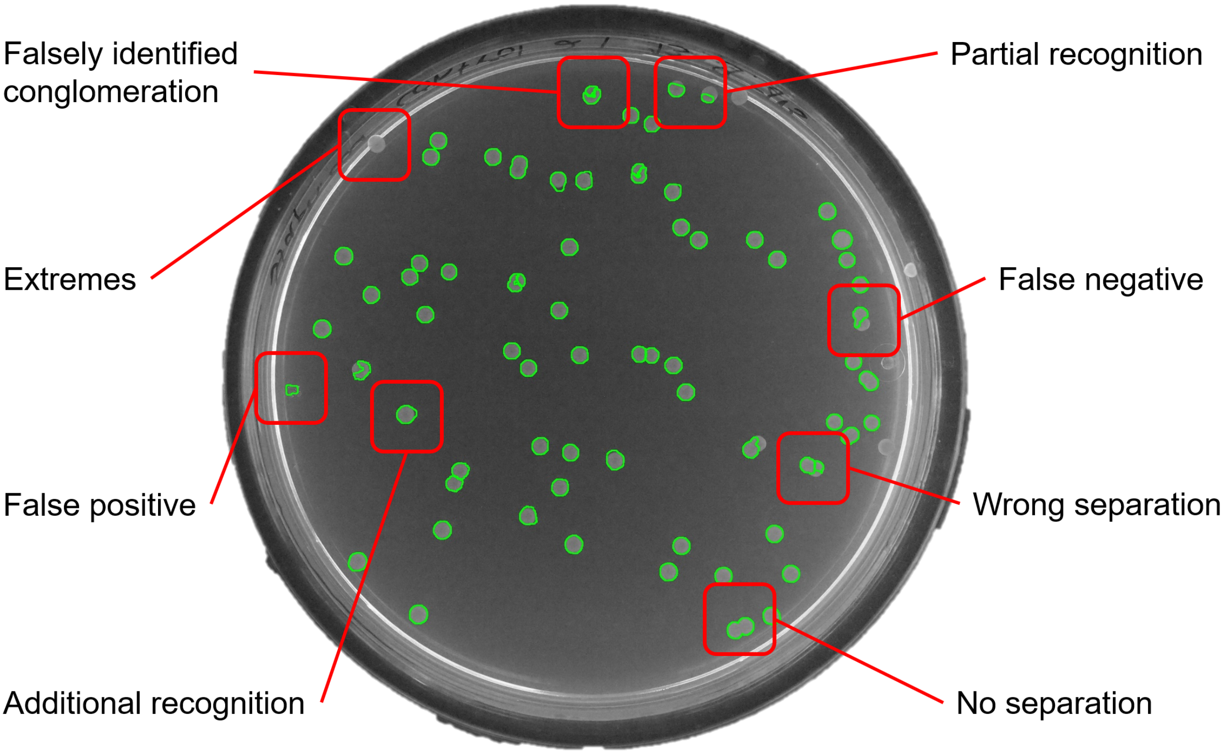
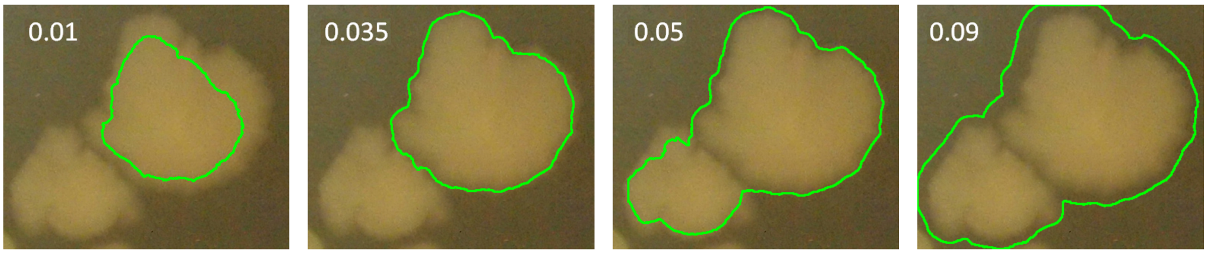


Figure 3.15: An overview of segmentation challenges based on an actual example. For the sake of this image, each challenge was given a name. All of these challenges can be categorized as false positives and false negatives. Falsely identified conglomeration, false positive and additional recognition shown on the left side are all false positive as they segment either a background as a colony or a border inside a colony which does not exist. On the contrary when not detecting a cell, being partially or fully, or a border correctly, then it can be categorized as a false negative as for the rest of the challenges.

In the segment removal step, activated with right mouse click, the coordinates of graphical inputs are compared to the center coordinates of their nearest segments. The segment closest to an input is then deleted. In the segment addition step, centers are inserted graphically by the user by using the left mouse button. These centers must be defined within the actual boundaries of the CFU visible in the image. In the correction phase, the adding function is internally processed in almost the same manner as a priori selection using the fast marching method. The only difference is that instead of giving all the seed points at once and running the function through the entire image, this function uses only one seed point at a time and runs it on a cropped part of the image which is of the same size as the biggest selected colony in the a priori selection phase. Moreover, the threshold value is set to 0.02 as a default. The threshold parameter for the addition of new segments with fast marching can be changed in the options dialog. To demonstrate the effect of the different values for the threshold parameter a two colony aggregate example was used as shown in Fig. 3.16. The parameter used to affect the outcome here is a threshold value that can range between 0 and 1. This value specifies the level at which the function thresholds the normalized geodesic distance map of the fast marching method to obtain the output binary image. If there are more centers for a single blob, then these reference points contained inside the blob are used as seed points for watershed segmentation instead of adopting the seed points that are controlled by regional maxima of the H-maxima intensity transform. Once the user is done adjusting the result, closing the window will end the correction phase.



(a) (b) (c) (d)

Figure 3.16: A showcase of fast marching method results using different threshold values. A small value such as 0.01 can cause the colony to be selected only partially as shown in (a). With a value of 0.035, the segmentation is almost perfect besides some of the faint border on the right as shown in (b). At a threshold of 0.05 the contour of the colony is completely delineated but part of the other colony is wrongfully included as well as (c) shows. As (d) shows increasing the value too high to 0.09 results in an undesired single aggregate of both colonies. The image was taken from the work already published [51].

3.3.4 Pipeline for Quality Management

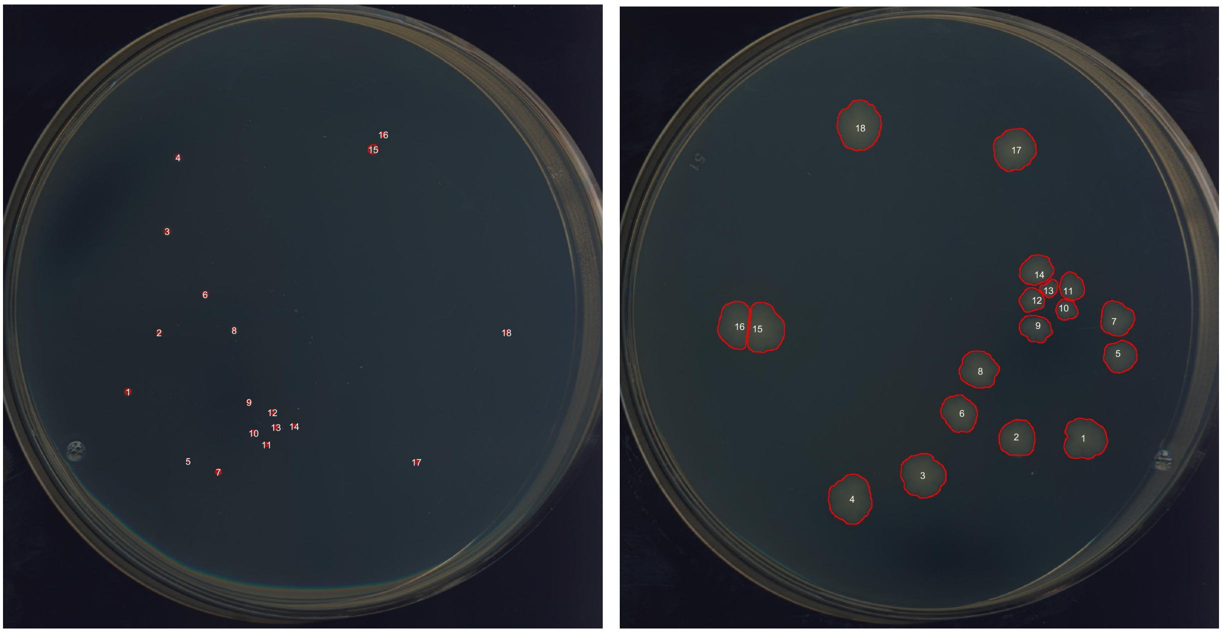
Based on a typical bacterial growth curve in culture medium there is an initial lag, followed by an exponential growth until it reaches an equilibrium and then dies again [81]. It is important to note that when assessing the quality of experiments, it is not necessarily the entire curve of interest. Instead, the focus should be on ensuring that as many colonies as possible are in the desired rising phase. Colonies in the exponential growth phase are the most affected by their surroundings and experiment setting. In this context, an automatic method for detecting this specific phase is of interest and presented in the following.

Ideally, this method should also provide a quality measure that quantifies how well the experiment is progressing. For instance, a measure of 80% would imply that 80% of the cells or colonies have reached the desired phase. The method presented here offers the capability to calculate this measure locally, allowing for a detailed analysis at the level of individual cells or colonies. This stands in contrast to a global assessment, which considers only average trends.

The proposed methods are evaluated on the last dataset ”CFU growth over time” that was presented in section 3.3.1. The results are reported in section 3.4.8. The functionality of the method presented in this chapter are not part of AutoCellSeg but could easily be integrated in the software.

Localization and Tracking

The initial challenge lies in ensuring the accurate tracking of colonies or cells across successive time points. While a scientist may take meticulous measures to maintain the petri dish’s consistent orientation, it is quite possible for the petri dish to undergo spontaneous rotations in various directions, when placed under the camera. Consequently, the colonies or cells occupying a specific position on the petri dish in one image may not necessarily maintain that exact location in a subsequent image of the same petri dish.



(a) (b)

Figure 3.17: Example of a rotated petri dish image (b) with all colonies successfully localized and tracked from the colonies of the first image in that series (a). Each colony is assigned a unique identifier that remains associated with that particular colony throughout the entirety of the series.

To compensate for such rotations, the following method for registering the images to the image acquired at the first time point is proposed. First, the center points of all segmented CFUs are used as a point cloud, which is compared with a point cloud from a later time stamp's segmentation. This approach is rooted in the observation that, despite the colonies' growth over time, their centers tend to maintain a relatively consistent position in relation to all other colony centroids.

The process involves passing these two point clouds to an Iterative Closest Point (ICP) algorithm [82,83]. This algorithm uses a k-dimensional tree for the search and yields a transformation matrix encompassing both rotation and translation, along with the root mean square error (RMSE). Experiments showed that for larger rotations the ICP was incapable of finding the correct orientation. In these cases the RMSE returned by the ICP was greater than 50, triggering the proposed method to rotate the cloud points clockwise by 30 degrees. The shifted point cloud is subsequently subjected to the ICP algorithm once more. This iterative procedure continues until an RMSE value lower than fifty is achieved.

Upon successfully registering the colonies of the rotating image (see Fig. 3.17b) with the colonies of the initial timestamped image (see Fig. 3.17a), a comprehensive analysis of each colony is conducted over the whole duration of the experiment. In addition to the calculation of the colony sizes and shape characteristics over the different time points of the experiment, an extra parameter is calculated, which can be interpreted as the neighboring space available for growth.

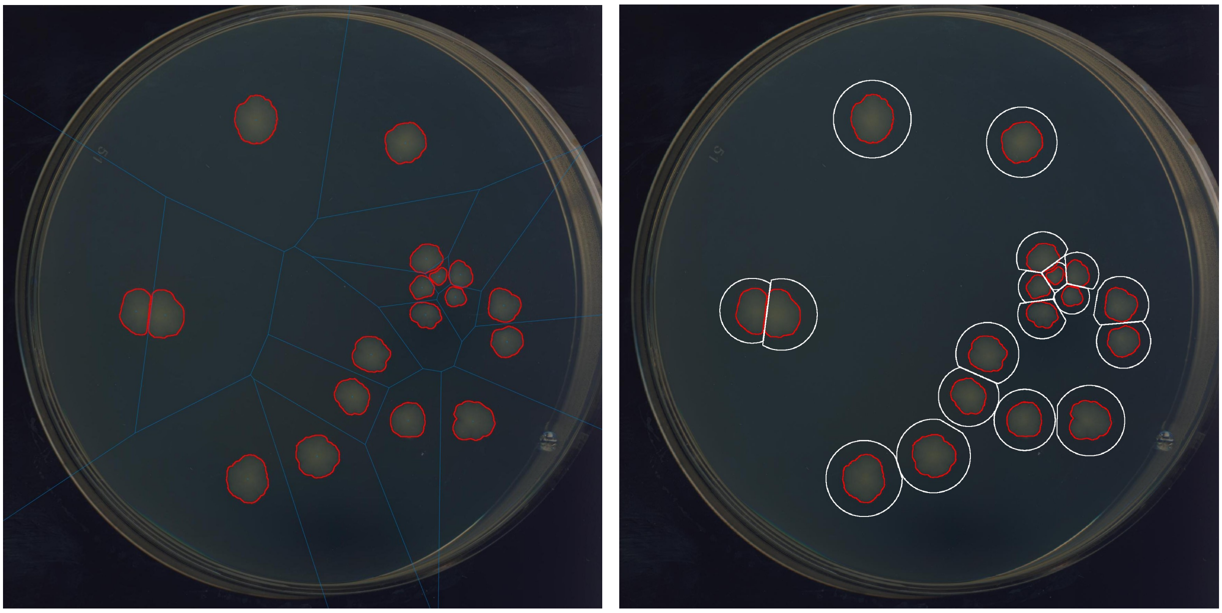

(a) (b)

Figure 3.18: Example of Voronoi diagram (a) on the same rotating image as in Fig. 3.17b. Based on three times the area of each colony a growing space area is calculated which are subsequently cut along the line of their Voronoi regions (b).

The foundation for this parameter is that in the competitive environment on the petri dish, where colonies compete for the limited resources on the agar plate, colony growth can be affected if restricted. The first step of the proposed method involves computing a Voronoi diagram [84] by partitioning the entire space surrounding each point from the point cloud into a designated region of influence. This region of influence is referred to as a Voronoi region, and the collection of all these Voronoi regions collectively forms the Voronoi diagram as shown in Fig. 3.18a. To determine the required space for growth, an area three times of the current surface area of each colony around their center is calculated and then multiplied by the corresponding Voronoi region associated with that colony as shown in Fig. 3.18b.

The probability of growth diminishes proportionally to the available space and can serve as an indicator. As the number of colonies in a petri dish increases, the likelihood of growth restrictions also rises. To inform users about whether the experiment is conducted without such limitations, a metric was introduced. This metric indicates the percentage of colonies that have unhindered growth, considering the available space capped at 65%. This value is derived from experimental data, which demonstrates that at approximately this level of available space, colony growth begins to slow down.

3.4 Results

This chapter introduces a detailed description of the evaluation methodology used and presents the results from new benchmark data, as well as the results from other data sets used in previous works. It demonstrates the robustness of AutoCellSeg by showing its performance with various data sets with different backgrounds and foregrounds as well as the difference between the available operation modes including batch processing on randomly selected images. Additionally we demonstrate that AutoCellSeg is also capable of segmenting more than just colonies with fluorescent mammalian cells. Here, the quality measures resulted from AutoCellSeg are compared to those in other software tools.

3.4.1 Evaluation Methodology

Specificity and sensitivity of tools used for colony counting were previously based predominantly on the total count alone. An analysis based solely on the total count, gives less insight into the plausibility of colonies, especially due to the lack of size detected. In many experiments, individual colony size could be of equal importance. Henceforth, an additional quality criterion based on segmentation precision should also be incorporated. Based on the work of Khan et al. [80], the quality measure Q, was selected to additionally evaluate the outcome of the segmentation process. Q takes two factors into account. Firstly, the relative deviation in segment count with respect to ground truth as:

$$q_1 = \frac{N_O}{N_{TT}}$$

where N_O is the number of overlapping objects and N_{TT} is the total number of objects in the ground truth image. Each blob of the segmented image is overlapped with the according ground truth blob. If the maximum overlap exceeds the heuristically chosen value of $\frac{1}{3}N_O$, the set of correctly counted segmented objects increases by one and the blob from the set of all blobs from the ground truth image is deleted. This is done to prevent multiple counting of the same object, meaning either when a blob is falsely recognized as two smaller ones or when multiple close blobs are recognized as a bigger one. Secondly, similar to the Rand Index [85], the number of misclassified segment pixels is calculated as:

$$q_2 = \frac{\sum_{i,j} |sign(\delta_{truth,ij} - \delta_{seg,ij})|}{\sum_{i,j} sign(\delta_{truth,ij} + \delta_{seg,ij})}$$

The a priori knowledge for these measures are handled using a fuzzy membership functions in order to describe it in the range between 0 and 1. The fuzzy function μ, which was used in both cases for q_1 and q_2, is given as:

$$\mu(q) = \begin{cases} 1 & q \leq 0 \\ 1 - 2(q)^2 & 0 < q \leq \frac{1}{2} \\ 2(q-1)^2 & \frac{1}{2} < q < 1 \\ 0 & q \geq 1 \end{cases}$$

Together they form the quality measure Q, valid for binary ground truth objects, as follows:

$$Q = \mu(q_1) \cdot \mu(q_2)$$

When ground truth data was not available, AutoCellSeg was directly compared to OpenCFU with a normalized difference in colony counts, denoted as σ, employed as follows:

$$\sigma = \frac{n_{acs} - n_{ocfu}}{max(n_{acs}, n_{ocfu})} \tag{3.1}$$

Here, n_{acs} and n_{ocfu} represent the number of colonies detected by AutoCellSeg and OpenCFU respectively. Negative values would indicate that AutoCellSeg underestimated the colony count compared to OpenCFU, while positive values indicate overestimation. A similar equation can also be used to compare any software solution against a manual count:

$$\sigma_{mc} = \frac{n_{counter} - n_{mc}}{n_{mc}} \tag{3.2}$$

where n_{mc} are the actual number of colonies counted manually and $n_{counter}$ is the number of colonies detected by a CFU counter solution. Positive values would indicate that the CFU counter solution overestimated the colony count, while negative values indicate underestimation.

3.4.2 Dark background benchmark

Various tools mentioned previously were compared using the Q measure introduced in section 3.4.1 with the dark background benchmark, see Fig. 3.19.

The four contestants for this benchmark were: AutoCellSeg, OpenCFU, CellProfiler as well as the combination of the latter with Ilastik in batch and individual mode. Both OpenCFU and AutoCellSeg were operated in semi-automatic mode as explained in the previous chapter. For CellProfiler separate pipelines were created to be used for each species. These can be found in the according folder of the repository [2]. For instance the CellProfiler pipeline and results for E.coli can be found under the folder “E.coli/CellProfiler” [3].

In the case of Ilastik only one image was taken for each bacterial specie for pixels labeling of the background and colonies. All available features were utilized until the resulting segmentation was close to the ground truth. The results of the segmentation were exported as tiff files and were then combined with the CellProfiler pipeline for quantification. The results of the segmentation were obtained as binary masks directly from the AutoCellSeg and CellProfiler pipelines. As is the case of OpenCFU, the center and the radius of the colonies are the only features available for a quality comparison of the colonies.

This is yet another drawback of using OpenCFU, making it impossible to directly compare the count and size of colonies with ground truth without a binary mask. Via the use of

---

[2] github.com/AngeloTorelli/AutoCellSeg/tree/master/DATA/Benchmark.Last visited: 15.09.2023

[3] github.com/AngeloTorelli/AutoCellSeg/tree/master/DATA/Benchmark/E.coli/CellProfiler. Last visited: 15.09.2023

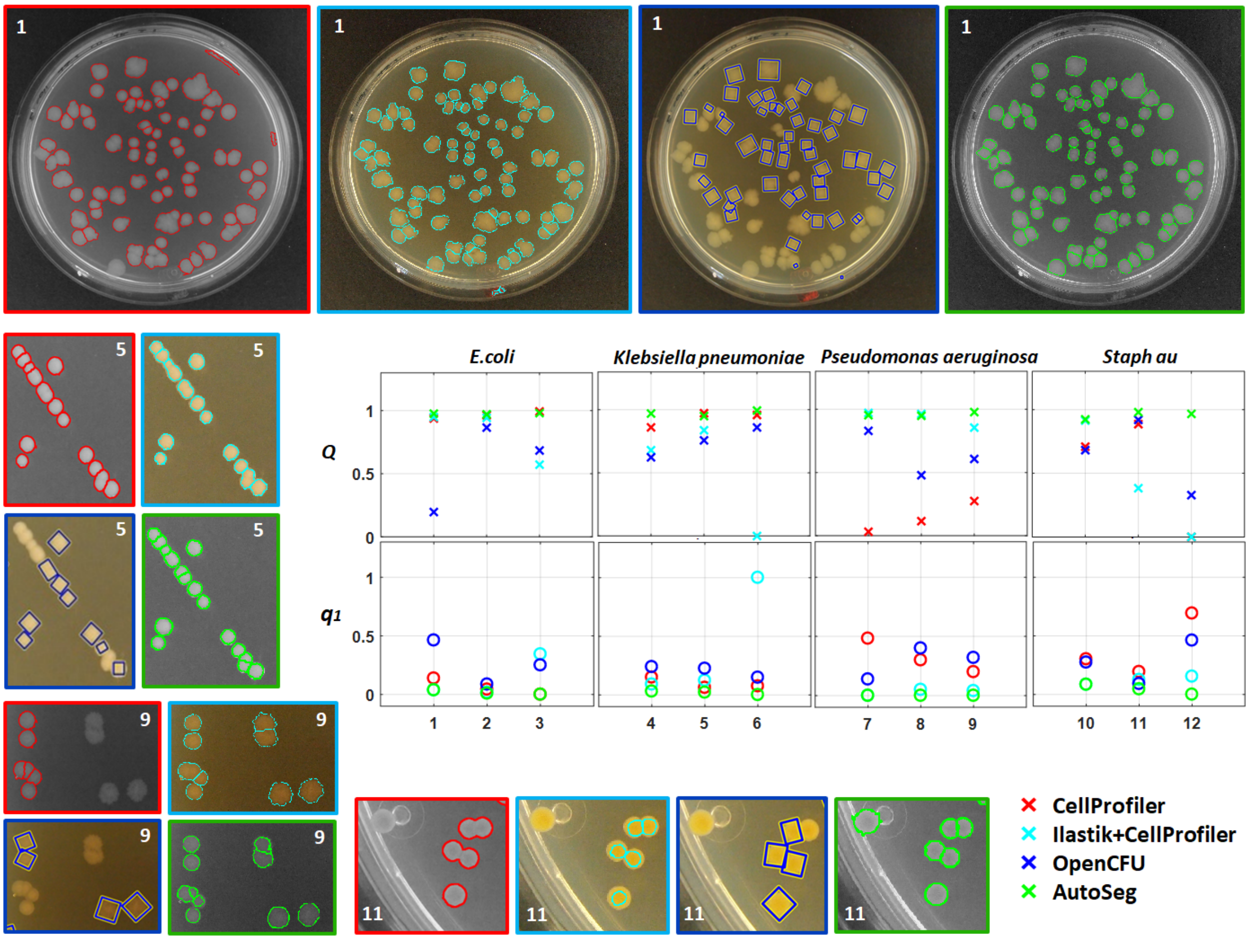


Figure 3.19: The colors red, cyan, blue and green are used to represent CellProfiler, Ilastik+CellProfiler, OpenCFU and AutoCellSeg respectively. One full example of the results from the dark background benchmark image data set are shown in the top row. The file name for that image is "E.coli 20160223 plate 30min 48h control 3 48h.JPG". The number in the corner shows which column it represents from the results graphs. One image from each species is selected to demonstrate some of the problematic occurring for each solution. The graphs on the top row shows the results for Q, which incorporates the measures for both size and count of CFUs, for each species. While the second row shows the relative deviation from manual count q1. This visual comparison uses an average Q value (Qm) and an average q1 value (q1,m). For the definition of Q and q1 see section 3.4.1. Image was taken from the work already published [51].

the center and radius of colonies, binary circles were constructed to emulate the colonies, and watershed segmentation was used to separate the overlapping colonies. This was performed to match the number and size of the colonies, as they were extracted from the OpenCFU software. All three binary masks acquired were compared against the ground truth for the total count and size of the colonies using the Q benchmark.

Due to its careful colony selection criteria, OpenCFU avoids false detection. In OpenCFU the user has to carefully specify the range of the colony size by manually using the slider bar, which can be cumbersome when trying to be precise. In addition to detecting noise both inside and outside of the CFU container, OpenCFU produces many false positives if the lower radius is very low. However, if the lower radius is set to a more accurate value (i.e. 25-35, in case of Staphylococcus aureus), the estimated colony count is widely underestimated. In order to get a good segmentation result, one must first try several values for smaller radii. Even for an expert user, extensive trial and error may still be necessary if different data sets with varying resolutions are to be evaluated. In AutoCellSeg, this issue is rectified, because the user must only click on one small and one big colony, in order for the smaller and larger radii to be automatically extracted using a fast marching method.

The second problem with OpenCFU, is that there is no fully-automated batch throughput solution. In order to advance, the user must manually click on the GUI's > tab in order to process the next image. Even when global parameters for segmentation and counting could be set, the fact that one must manually click through data, significantly hinders OpenCFU to be used for larger data sets. In some cases, CellProfiler delivers a satisfying segmentation result.

The results of Ilastik with CellProfiler in batch operation was not significantly better than OpenCFU or CellProfiler. Nevertheless, the combination of the two programs produces results comparable to AutoCellSeg, when it comes to labeling all of the images, which does take much time and effort. In terms of quality, Q, of the segmentation outcome, AutoCellSeg outperforms the other solutions. The graphs in Fig. 3.19 show the individual comparison between the various solutions.

Software	**Petri dish extraction**	**Post-processing**	**Benchmark evaluation Q_m**	**Deviation in count $q_{1,m}$ (%)**
AutoCellSeg	Automatic	Addition + removal	0.97	1.9
OpenCFU	ROI mask	Removal	0.65	25.8
CellProfiler (CP)	Manual	N/A	0.64	22
Ilastik + CP (Batch)	N/A	N/A	0.67	17.2
Ilastik + CP (Manual)	N/A	N/A	0.95	5.4

Table 3.1: Quality comparison of AutoCellSeg, OpenCFU, CellProfiler, Ilastik and CellProfiler in batch and manual mode using the dark background benchmark images. The Q_m and $q_{1,m}$ values shown in the last two columns are the average values for all 12 images used. Table was taken from the work already published [51].

Table 3.1 depicts the overall comparison between AutoCellSeg, OpenCFU and CellProfiler and the combination of Ilastik and Cellprofiler. In contrast, AutoCellSeg does not demand manual work, and allows users to enjoy a fully-automated process, without human intervention during its running process. Even if colonies vary greatly in size AutoCellSeg is able to analyze the images because it draws a priori information from the input examples automatically. In this way, AutoCellSeg can also easily be used for other data sets, as well as images containing other cell structures. When structures are not circular, AutoCellSeg takes potential eccentricity of objects into account via graphical, a priori knowledge input. AutoCellSeg also extracts other a priori features based on intensity, in case of the detection of false positives.

3.4.3 Inverted background benchmark

Since AutoCellSeg was originally created for images with dark background and light foreground, one of the challenges would be to apply the same algorithm on a data set with a light background and darker colonies to see how well AutoCellSeg can adapt to the new type of images. For this challenge the inverted background dataset was used which was previously described.

Only AutoCellSeg and OpenCFU were compared against each other by running both of them in the partially automated mode. A big disadvantage using OpenCFU is the lack of a fully automated batch mode, since the user has to advance to the next image by clicking on the GUI after processing each image. In OpenCFU global parameters, or a priori knowledge, is based on trial and error since the framework does not offer any overview of the complete dataset. OpenCFU was configured with a minimum radius of 1 pixel as a global parameter to account for the presence of very small colonies in several images. For the same reason, the Guassian filter mask was reduced to a size length of three and the edge detection was turned on in AutoCellSeg.

Figure 3.20 displays the results for one of the densely populated dishes. OpenCFU effectively identifies almost all colonies in automated mode. In the first image of the second row, it is evident that OpenCFU failed to detect a cluster of colonies. Moreover, OpenCFU

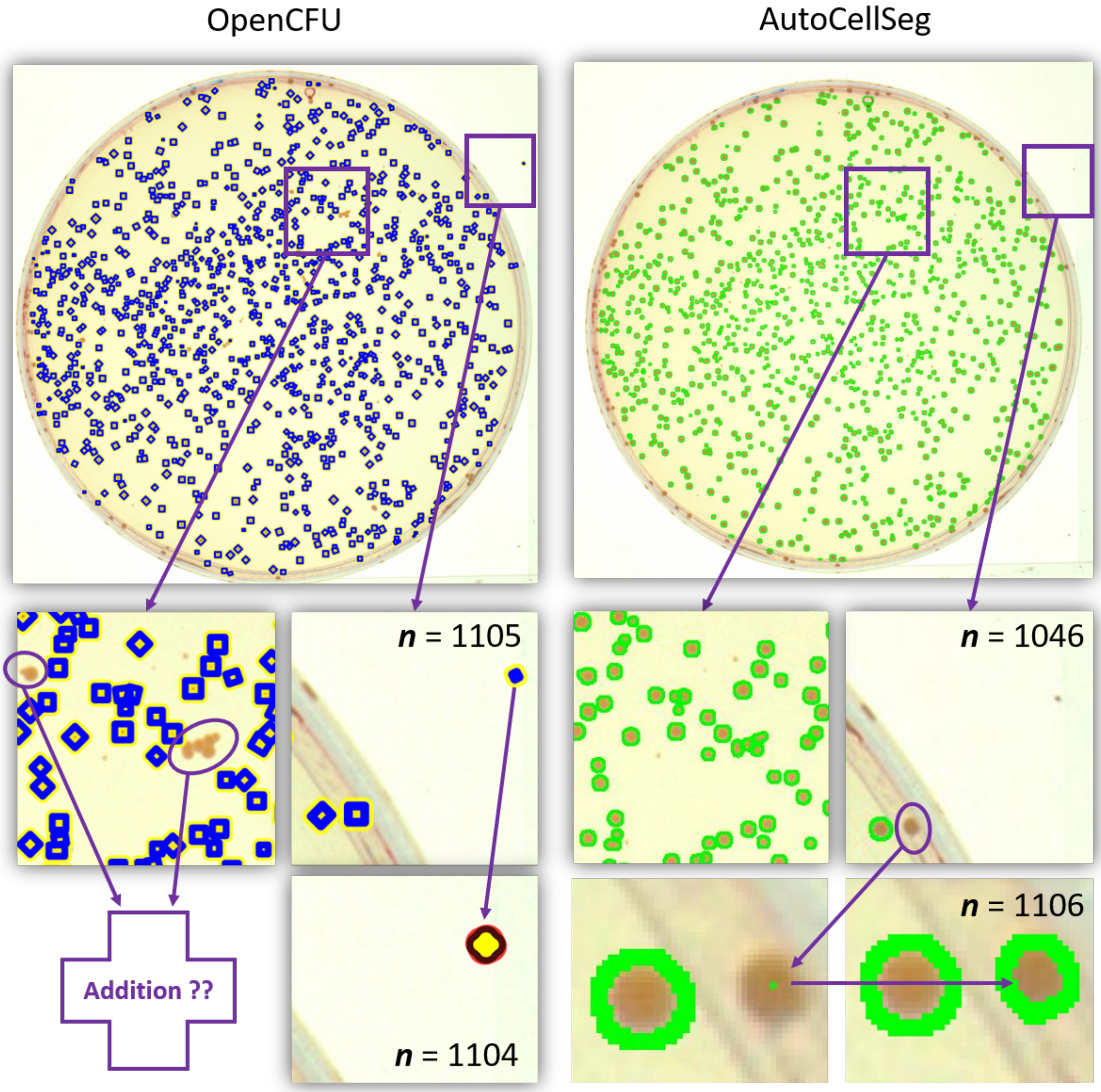


Figure 3.20: This presents the segmentation results of AutoCellSeg and OpenCFU for one of the images characterized by a lighter background and darker colonies. The top-left image displays the detection results from OpenCFU, while the top-right image shows the segmentation achieved by AutoCellSeg. The second row provides zoomed-in views of selected areas from the first row images without any post-editing, and the last row shows the results after applying the correction step. Image was taken unchanged from the work already published [51].

mistakenly identified some noise outside the dish, especially when choosing a very small value for the lower radius. Conversely, AutoCellSeg has an advantage as it automatically extracts the container and therefore successfully segmented the same section as shown in the third image of the second row. The underestimation of colonies by AutoCellSeg in this case is attributed to the automatic exclusion of colonies on the petri dish's boundaries in fully automated mode without correction, as seen in the last image of the second row. Moving on to the correction step, OpenCFU allows for the removal of erroneously detected segments outside the dish, as demonstrated in the second image of the last row. In contrast, AutoCellSeg additionally offers an interactive solution where undetected segments can be manually added, as illustrated in the third image of the last row. As mentioned before, AutoCellSeg employs a fast marching method to segment undetected colonies based on user-defined seed points, as shown in the last image of the last row. However, AutoCellSeg may occasionally miss smaller colonies depending on user selections during the initial extraction phase, a situation that can be addressed easily in the correction step.

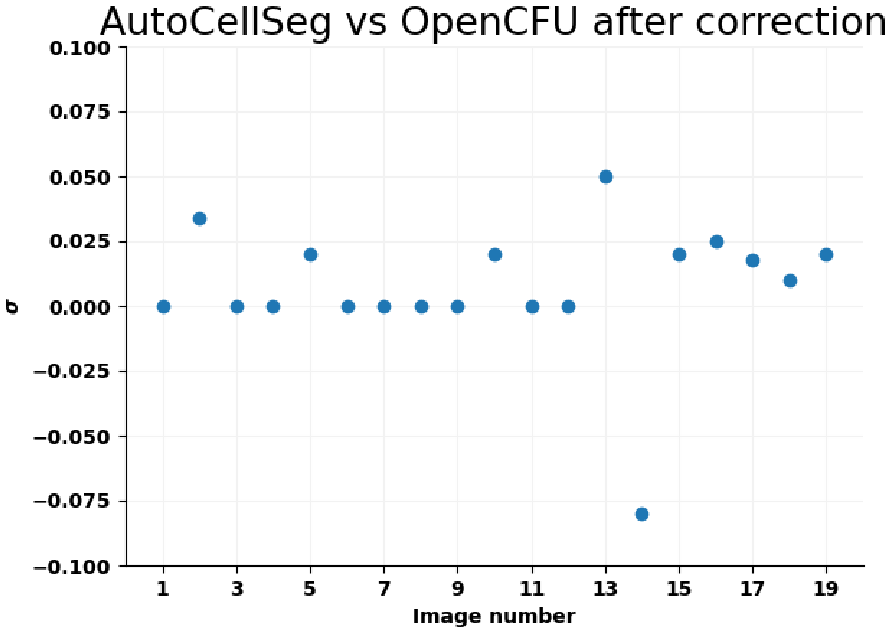
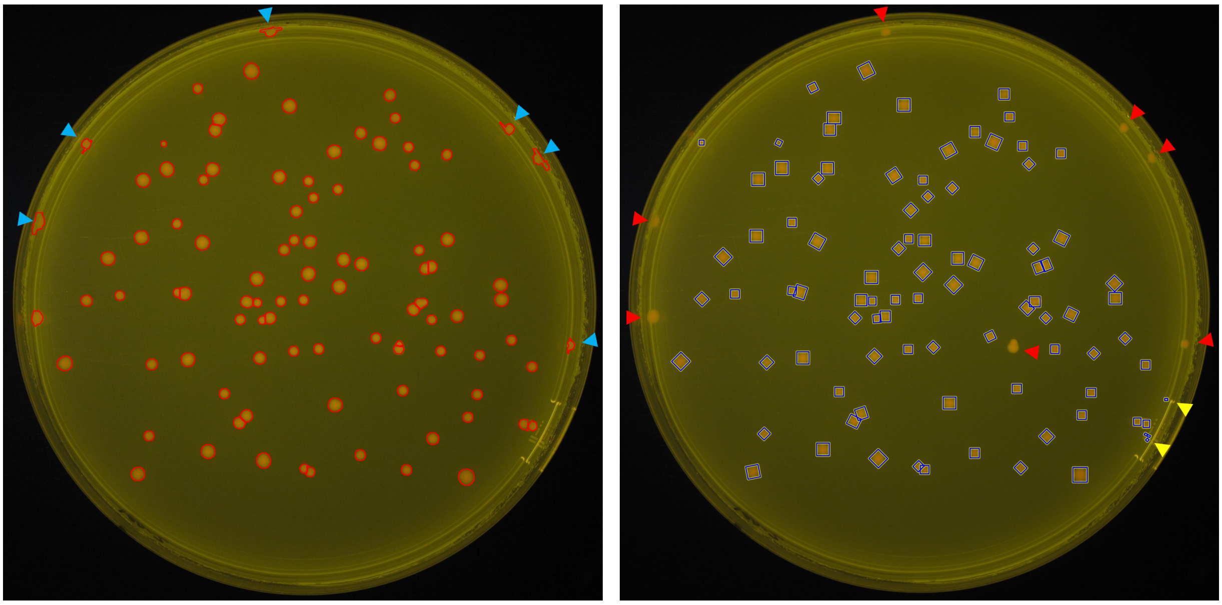


Figure 3.21: The assessment of similarity between AutoCellSeg and OpenCFU relies on the disparity in colony counts. The x-axis denotes the image numbers, while the y-axis represents the normalized difference in colony count, denoted as σ in accordance with criterion (3.1). This assessment is conducted in two modes: one without correction (orange) and the other with correction (blue).

Like many other CFU counting solutions, OpenCFU fails to offer the extraction of the colony container. During post-processing, OpenCFU only allows for the removal of false segments, as indicated by the arrow in the second image of the last row in Fig.3.20. Notably, in cases where segments go undetected, users cannot manually add new segments, as shown in the first image in the second row of Fig.3.20. In contrast, AutoCellSeg offers full flexibility by allowing users to add undetected segments and remove falsely detected ones, as seen in the third image in the second and third rows of Fig.3.20. The goal here is to compare colony counts for all images using both software solutions, taking into consideration the size of the detected colonies. The final colony counts obtained from both software packages exhibited remarkable similarity, as evidenced in Fig. 3.21. It is worth noting that the correction step effectively reduces the disparities between the two tools. AutoCellSeg may exhibit a slight tendency to more accurately estimate the colony number.

3.4.4 Batch benchmark

As mentioned in the previous section the advantage in AutoCellSeg is that after running the process in any mode, one can also add undetected colonies as well as removing the incorrectly detected colonies. The latter is also the case with OpenCFU. Since the default settings of OpenCFU failed to recognize most CFUs, it was executed in bilateral threshold mode with a value of ten and with a minimum radius of four selected as the global parameter settings. In contrast, AutoCellSeg utilized default parameters. Initially, both programs operated in fully automated mode. Subsequently, manual correction was carried out for both, allowing for an assessment of the degree of refinement each software affords the user in improving the results. Furthermore, users can readily observe distinctions in the detected colony boundaries as can be seen in Fig. 3.22a.



(a) (b)

Figure 3.22: The 36th image from the batch processing dataset named ”IMG_1033.jpg” was considered as an example, along with the corresponding segmentation results generated by both software solutions. The outcomes of AutoCellSeg (a) and OpenCFU (b) were obtained in fully-automated with a subsequent correction. The blue arrowheads highlight manually added CFUs, which were initially missed by AutoCellSeg’s segmentation. The red arrowheads indicate false negatives that could not be added by OpenCFU. The yellow arrowheads show instances of false positive which were easily deleted.

As an example the 36th image from the batch processing dataset named ”IMG_1033.jpg” was chosen. AutoCellSeg has automatic petri dish extraction which causes false negatives around the border of the petri dish. Thanks to the ability to add missed CFUs in the correction phase in AutoCellSeg, they can be seen delineated by the blue arrow heads in Fig. 3.22a. Nevertheless one can see that the delineation of these cases is not precise because the reflection of the glass is similar to the intensity values of the CFUs. OpenCFU has eight false negatives, six extremes around the border and one conglomerate in the middle of the petri dish as indicated by the red arrowheads as seen in Fig. 3.22b. The lack of the capability to add new segments during the correction step for OpenCFU makes it impossible to add the conglomerate in the center as well as the six missed CFUs on the border, even when changing the global parameters. The yellow arrow heads indicate instances of false positive which were easily deleted.

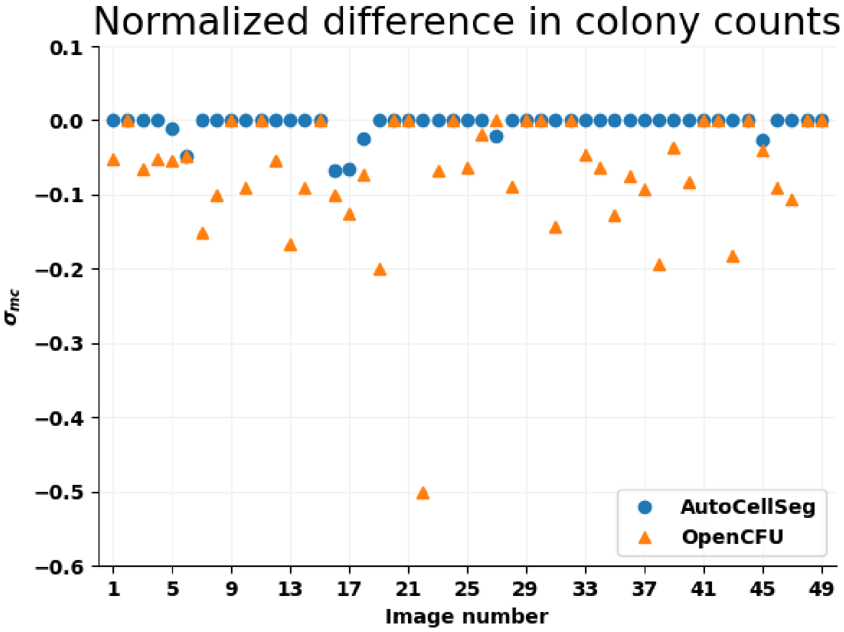


Figure 3.23: The assessment of similarity between manual count and both AutoCellSeg (blue) and OpenCFU (orange) relies on the disparity in colony counts. The x-axis denotes the image numbers, while the y-axis represents the normalized difference in colony count, denoted as σ_{mc} in accordance with criterion (3.2).

Since there was no ground truth present with the data set the count of each CFU in each image was done manually. The results of AutoCellSeg and OpenCFU were compared with the manual count, as shown in Fig. 3.23. For the normalized difference in colony count the same σ in accordance with criterion (3.1) was use as before. The results not only shows that with AutoCellSeg, a comparable count to OpenCFU can be obtained, but that AutoCellSeg is closer to the ground truth count as with OpenCFU. This shows the versatility and applicability of AutoCellSeg on different data sets.

3.4.5 Fluorescent cell segmentation

OpenCFU was run with auto threshold mode selecting a minimum radius of 10 (global parameter setting). Both tools were used in fully automated mode with the manual correction. Due to OpenCFU's limitation of saving a segmentation binary mask, the approach involved constructing binary circles based on the colony centers and radii, followed by applying watershed segmentation to effectively separate overlapping colonies. The segmentation quality of OpenCFU and AutoCellSeg was evaluated against the ground truth provided using eight different evaluation measures. Apart from criterion (Q_m), $\mu(q_1)_m$ and $\mu(q_2)_m$, five other metrics were chosen from the 2018 IEEE International Symposium on Biomedical Imaging Challenges. These metrics are:

- local pixel-wise boundary prediction error (pixel error e_p),
- minimum mean square error between the pixels of the target segmentation and the topology-preserving warped source segmentation called warping error (e_w) [86],
- frequency with which two masks disagree over whether a pair of pixels belongs to the same or different objects, called rand error (e_r), which is closely related to rand index [85],
- a scoring metric particularly useful in applications where the accurate delineation of objects is critical by focusing on the key foreground elements while minimizing the impact of background noise or irrelevant information called the foreground-restricted rand scoring (V^R) [87] and
- a criterion regarding the quantification of information and uncertainty called information theoretic scoring (V^I) [87].

For the sake of completeness, these established evaluation measurements were used even though they come with flaws. For example pixel error considers only whether or not a given pixel was correctly classified as a boundary pixel, without concerning the ultimate effect of that prediction on the resulting image segmentation [87]. V^R and V^I which were introduced to compensate these flaws and to compare the results better in the presence of challenges, are the reason for their inclusion in the comparison.

Data set	**Tool**	Q_m	$\mu(q_1)_m$	$\mu(q_2)_m$	e_p	e_w	e_r	V^R	V^I
BBBC007	1	**0.49**	**0.98**	**0.50**	**0.33**	2.2	**0.06**	**0.97**	**0.99**
	2	0.10	0.64	0.15	0.68	**0.6**	0.55	0.90	0.45
BBBC008	1	**0.72**	**0.99**	**0.73**	**0.21**	2.5	**0.13**	**0.93**	**0.98**
	2	0.21	0.57	0.35	0.42	**0.6**	0.57	0.62	0.89
Dark BG	1	**0.97**	**0.98**	**0.97**	**0.06**	0.4	**0.11**	**0.97**	**0.99**
	2	0.65	0.74	0.76	0.21	**0.2**	0.28	0.74	0.91

Table 3.2: Comparison between AutoCellSeg (tool 1) and OpenCFU (tool 2) using three different benchmark data sets: BBBC007 and BBBC008 from the Broad Institute, as well as the dark background benchmark data set created for this work. The evaluation measures are: average total quality (Q_m), average count ($\mu(q_1)_m$), average overlapping pixels ($\mu(q_2)_m$), local pixel-wise boundary prediction error (e_p), warping error (e_w unit in 10^{-4}), rand error (e_r), foreground-restricted rand scoring (V^R) and information theoretic scoring (V^I). The better results between the two software are highlighted in bold. The table was taken from the work already published [51].

The results are shown in Tab. 3.2, where it can be seen that AutoCellSeg has a higher score and lower error than OpenCFU in almost all metrics. Only the warping error from the masks created from OpenCFU were better because the perfect circles affect this measure to an extremely lesser extent. From the results, it can be seen that measure μ_2 is somewhat more strict than e_p since it takes into account each individual pixel of each object instead of the entire image. For a comparison, the results of the dark background dataset was also added.

3.4.6 Single images benchmark

For the three randomly selected images [61,67] presented in 3.3.1, AutoCellSeg successfully segmented all CFUs as can be seen in Fig. 3.24. The count of AutoCellSeg followed by the manual count can be seen in white in the corner of the images. The top left image has a count of twenty-two which is the same as the segmentation of AutoCellSeg. The fifty-one CFUs in the image with clonogenic colonies vary greatly in size and are blue colored on a light background. All fifty-one CFUs were segmented correctly as shown in the big image on the right side of Fig. 3.24. AutoCellSeg effectively segmented the central cluster of five colonies, aligning with the mask designated as a five-colony aggregate in accordance with the ground truth. Additionally, it recognized also the other four colonies around the conglomerate.

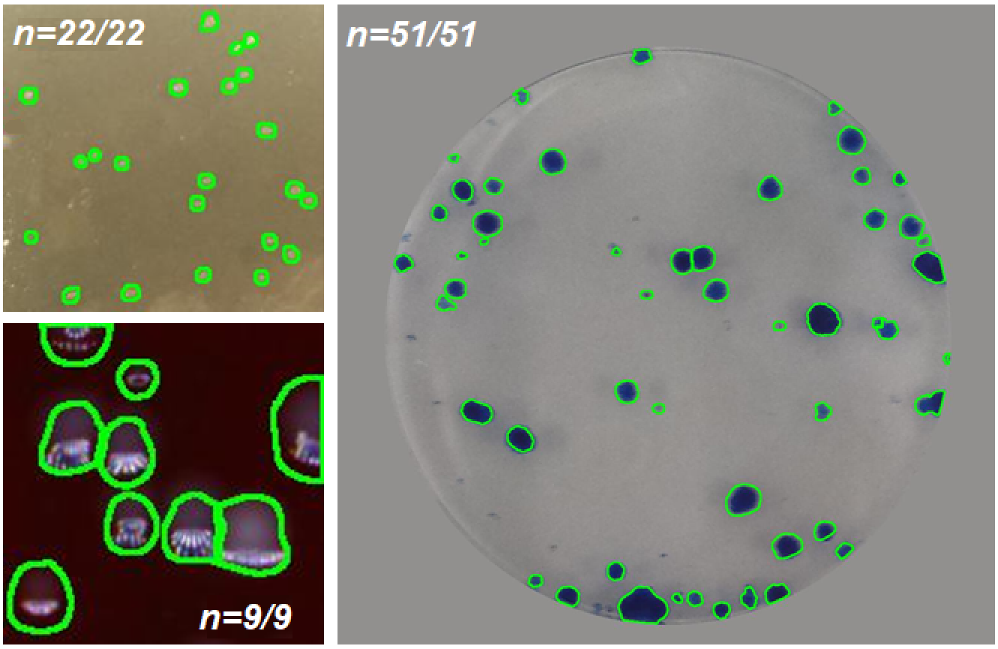


Figure 3.24: Segmentation done with AutoCellSeg of different images using partially automated mode with correction. The image at bottom left was randomly selected from Ferrari et al. [67] and the other two images are taken from Choundhry et al. [61]. The colony count provided by AutoCellSeg as well as the manual colony count for each image is shown in white on the corners of the images. The image was taken unchanged from the work already published [51].

3.4.7 Control/ test analysis

AutoCellSeg offers a unique capability compared to other CFU analysis software by enabling the comparison of different experiment settings. For instance, in microbiological experiments, it allows users to observe changes in CFU morphology and quantity during various treatments. This feature is exemplified in a test involving the irradiation of *E.coli* colonies with blue light to monitor alterations in colony size and count. AutoCellSeg facilitates this experimental comparison by offering colony size and count comparison plots as well as Kernel Density Estimation (KDE) functions plots. In practice, users load the images into AutoCellSeg and configure the name of control and test images in the parameters accordingly. After processing all images, users can choose to display comparison plots. The software also permits corrections at the individual image level. The resulting segment outlines, represented in light blue for control CFUs and red for test experiment CFUs, showcase the detection outcomes in Fig. 3.25. Each experiment is represented by a specific color shown on the top left corner of each image in the first row.

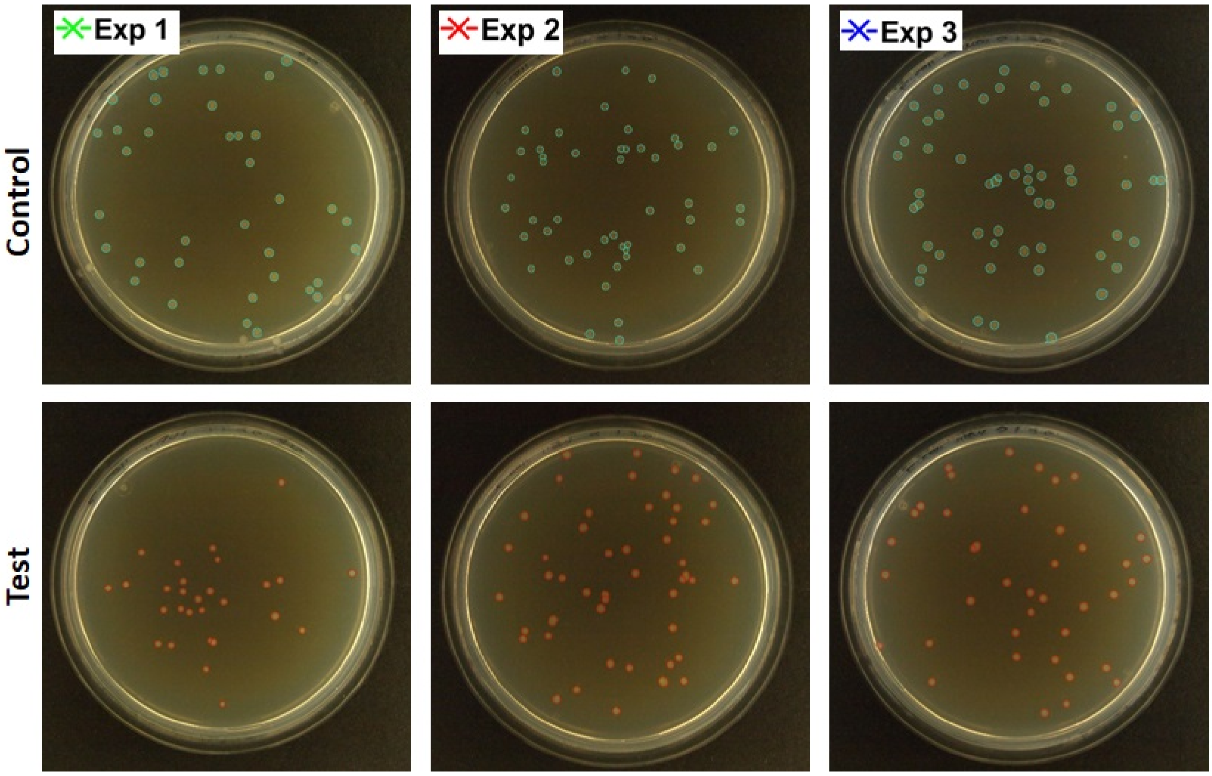


Figure 3.25: First and second row show the segmentation results for control (light blue) and test (red) images respectively. Each column represents the experiment number. In this case the control/test pair have no dependencies between each other but are only linked by their file name.

Kernel Density Estimation (KDE) functions are then plotted for each pair of datasets, providing probability density estimates of CFU size based on normal kernel functions (see Fig. 3.26).

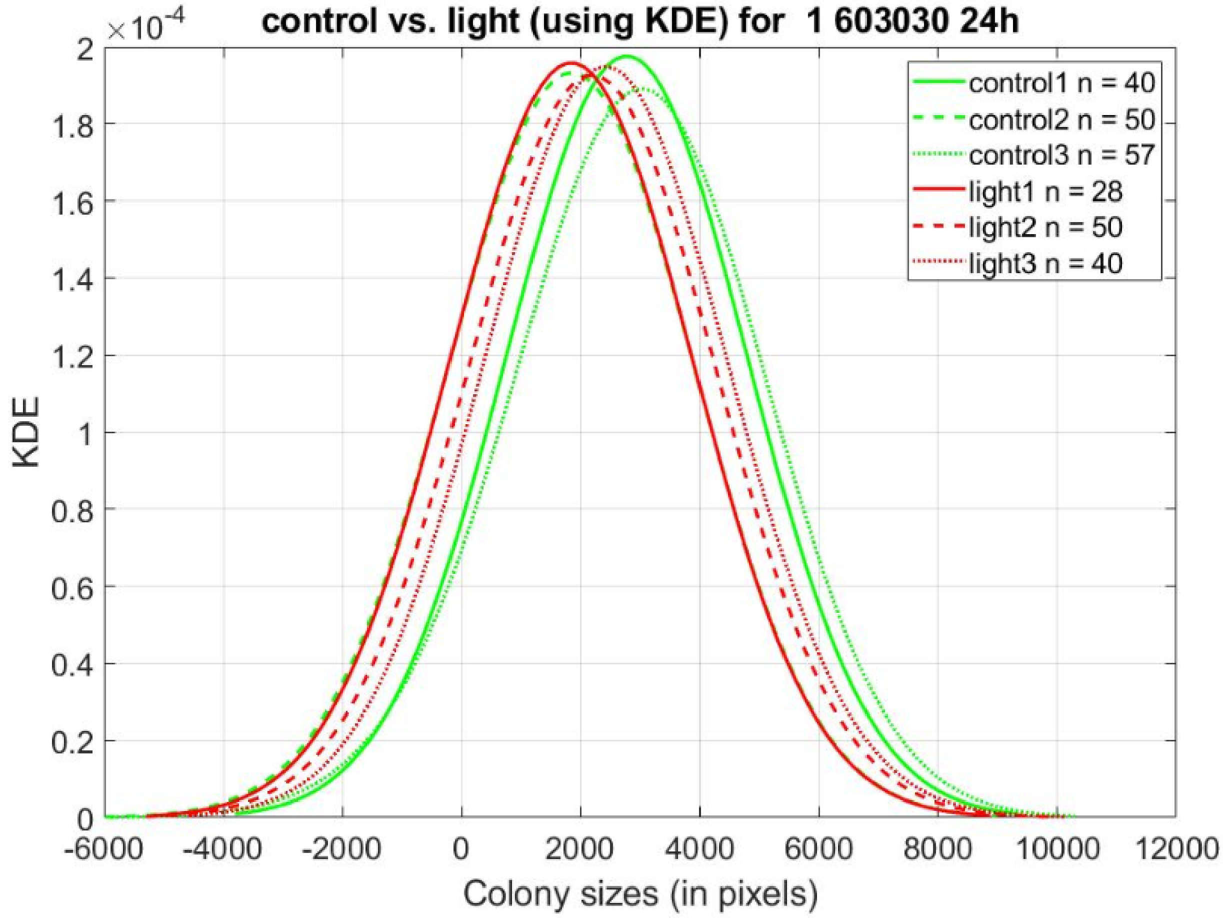
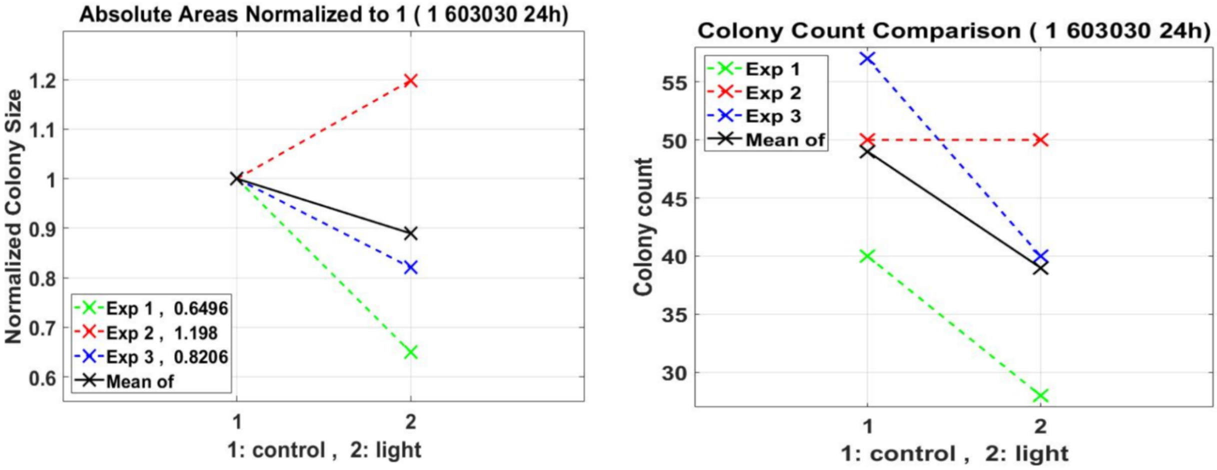


Figure 3.26: Kernel Density Estimation (KDE) plot is displayed on y-axis for colony sizes in pixels (x-axis). The green lines represent control experiment results, where as red lines represent test experiment results. The number of colonies found in each image is denoted by n in the legend. Negative values on the x-axis should not be interpreted as indicating colonies with negative CFU areas; rather, they arise from the extrapolation of the smoothing function.

The green curves represent the size distribution in control colonies, while the red curves illustrate the size distribution (in pixels) of CFUs following light irradiation. The smoothness of these curves is controlled by a bandwidth parameter. Negative values on the x-axis do not signify the presence of colonies with negative CFU area; instead they result from the extrapolation of the smoothing function.

AutoCellSeg also allows users to observe changes in CFU area and count after test experiments. Since in this case each experiment is actually not related to each other and only linked by the name, it can be deceiving to see some experiment increase and some decrease in size. When the control/test pairs share dependencies, the line of interest is the mean value, represented by the black line. Fig.3.27a displays changes in CFU areas, normalized to 1 for control conditions, with different colors representing various experiments. In this specific setup, CFU sizes appear to decrease after light irradiation. Fig.3.27b shows the absolute CFU count for both control and test conditions, indicating an average decrease in this experiment.



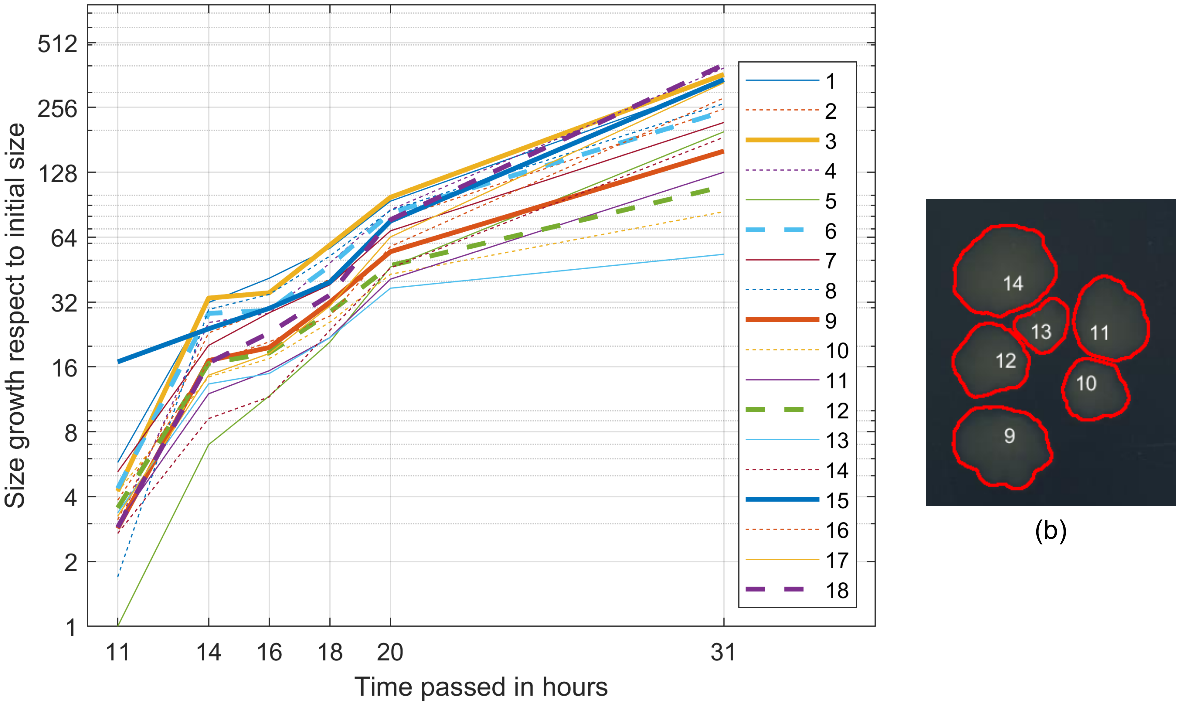
(a) (b)

Figure 3.27: In this case the control/test pairs are independent, making the mean value the line of interest, which is represented by the black line. The left plot in (a) shows the change in absolute colony sizes normalized to 1 for each individual control/test pair. This is shown to express the change in size for each pair of experiments (in color) and the total mean change between control and test in black. The right plot in (b) is the absolute count of colonies detected in each experiment. The black line shows the average colony count.

AutoCellSeg effectively extracts a priori information from input examples, even when colony size exhibits significant variation. This adaptability extends to different cell structures, even when they are not perfectly round, thanks to AutoCellSeg's consideration of object eccentricity through graphical a priori knowledge input. The software also extracts additional a priori features based on intensity to address false positives.

3.4.8 CFU growth over time

It is often challenging for the human eye to perceive subtle growth impairments, which can be influenced by physical and chemical factors. These impairments may not be immediately apparent but over time it can be noticeable. AutoCellSeg offers the ability to quantify these changes and can therefore be used as a quality management for petri dish experiments.



(a)

(b)

Figure 3.28: The growth curve of each of the eighteen colonies found the in Medium I series (a). The bottom six curves, number 9 to 14, with diminishing growth rates are from a cluster of colonies shown in (b).

The plate series described at the end of section 3.3.1 were categorized as few, medium or many depending on how many colonies were present in the plate throughout the series. Few was defined as ten or less, more than fifty was considered many and anything in between was classified as medium. The *E.coli* were not visible at the eight hour timestamp. Since all colonies start from zero pixels when the petri dish was freshly prepared, it is safe to assume that the different sizes, when the colonies start to be visible, mean that each colony has a different starting growing point. For that reason the smallest colony area when first visible is taken as reference. All the values seen in Fig. 3.28a are the size growth ratio in respect to that initial size.

The value introduced in Section 3.3.4 concerning the reference value indicating the space available for a colony can be computed for the entirety of the petri dish and subsequently presented in the title, as depicted in Figure 3.29. This plot shows the growth factor from the timestamp twenty hours to thirty-one hours in relations to it's growing space at those times. The orange line is the best fit function from all the points and the red line shows the cut off of 65% of available space. The green dots are colonies with higher available space as 65%, while the blue one are beneath that threshold, which are the same six colonies that show diminishing growth rates in 3.28b.

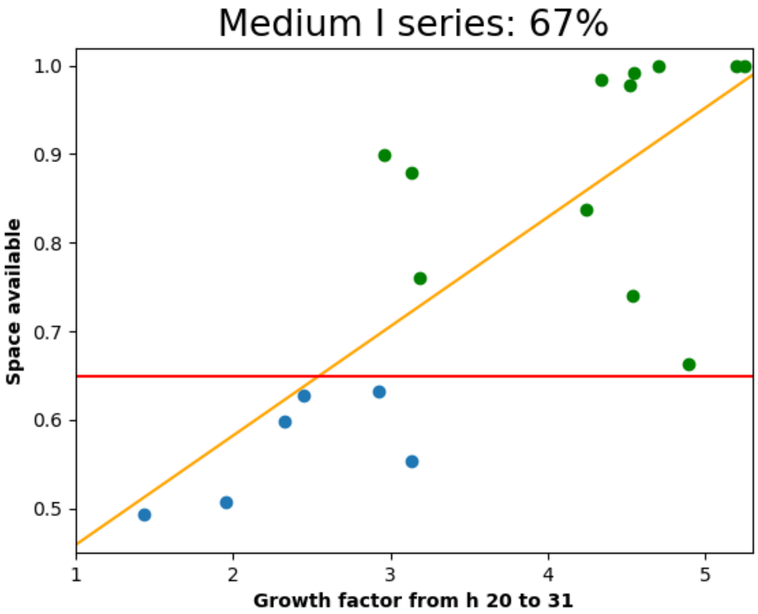


Figure 3.29: Relation between growing factor and available space scatter plot. The threshold of 65% available space (red) separates colonies beneath it (blue) and those above that threshold (green). The orange line is the best fit function. The 67% is the percentage of colonies that have more than 65% of the space available to grow in the future, which in this case is twelve out of eighteen.

3.5 Discussion

AutoCellSeg comes with a GUI which allows the user to select images or an entire folder, change the parameters, choose the mode of operation, select the a priori information, correct the results graphically, create comparison plots and save all the results including individual CFUs features. The main advantage in using AutoCellSeg lies in its ability to adjust the results in an intuitive and interactive fashion using simple graphical selection. Especially, the adding of missing CFUs is a unique feature not found in other solutions.

It is versatile and robust enough to cope with different image variations, adapts to various backgrounds and can cope with untypical morphological cell structures in a variety of data sets. This has been demonstrated on a several benchmark sets based on bacteria colonies and established benchmarks for cell segmentation in contrast to common CFU counter solution such as CellProfiler and OpenCFU, outperforming the competition. Since AutoCellSeg saves also the entire masks of the segmentation, which other CFU counters lack, it can easily be used to create benchmark datasets, which can be then shared with the public.

The ability to track colonies along rotating petri dishes and presenting the user with a quality value, showing the percentage of unrestricted colonies, has also been found to work successfully helping the user furthermore in the setup of experiments.

AutoCellSeg provides new features and more flexibility in comparison to existing software tools for a better user experience as seen in Table 3.3.

Automatic Petri-dish extraction is a feature not performed by CellProfiler and OpenCFU. In order to extract the information that is only present inside the container, OpenCFU allows the user to select a region of interest (ROI) in the form of a 3-point circle, or a complex polygon. Disappointingly, the ”Auto-Petri” function mentioned in the work of Geissmann [58] is nowhere to be seen on its GUI, and it also fails to run adequately, as there were several results with false positives caused by the noise beyond the Petri dish borders.

Features - Programs	Auto-CellSeg	Open-CFU	Cell-Profiler	+ Ilastik (batch)	+ Ilastik (indiviual)
A-priori labeling	✓✓✓	✓	✓✓	✓✓✓	✓✓✓
Accuracy	✓✓✓	✓✓	✓✓	✓✓	✓✓✓
All-in-one solution	✓✓✓	✓✓	✓✓	✓	✓
Data analysis	✓✓✓	✓	✓✓	✓✓	✓✓
Degree of automation	✓✓✓	✓✓	✓✓	✓	✓
Execution speed	✓✓	✓✓✓	✓	✓	✓
Post-processing	✓✓✓	✓	✓	✓	✓
User-friendly	✓✓✓	✓✓	✓	✓	✓

Table 3.3: Comparison of the most relevant features and the performance between the used frameworks. ✓ depicts the baseline among the used frameworks. ✓✓ represents an improvement in compared to the baseline and ✓✓✓ illustrates the best option between the tools. The table was taken from the work already published [51].

CellProfiler and OpenCFU may not be the best solution for a high throughput and fully automated segmentation, because one must manually select the Petri dish as well as of the manual tuning of the parameters. However, the segmentation outcomes produced by CellProfiler and OpenCFU were the closest to AutoCellSeg and were better compared to all the other software solutions tested. Thus, they were included for comparison of some datasets.

Based on the work done by Choundhry et. al. [61] a custom-made pipeline in CellProfiler as well as a combination with Ilastik was developed to have a better comparison with OpenCFU and AutoCellSeg. Ilastik was included, because, pixel-wise, it delivers reasonable predictions, via supervised machine learning.

Regarding the other mentioned programs, most of them were discarded because of too many disadvantages or the results not being satisfactory in initial tests. For example ImageJ Edge and IJM both failed when a CFU container was in the image. There were two additional ImageJ Macro that promised great advantages in their published papers. Siragusa et al. created a lightbox for reproducibility and predefined plate positioning purposes and developed a ImageJ Macro called CoCoNut, to automatically recognize the cells tested in their laboratories [88]. A more recent ImageJ macro created by Brzozowska et al., called countPHICS [89] was published in combination with a program written with QT Creator which translates the output text file and plots histograms of colony size distribution.

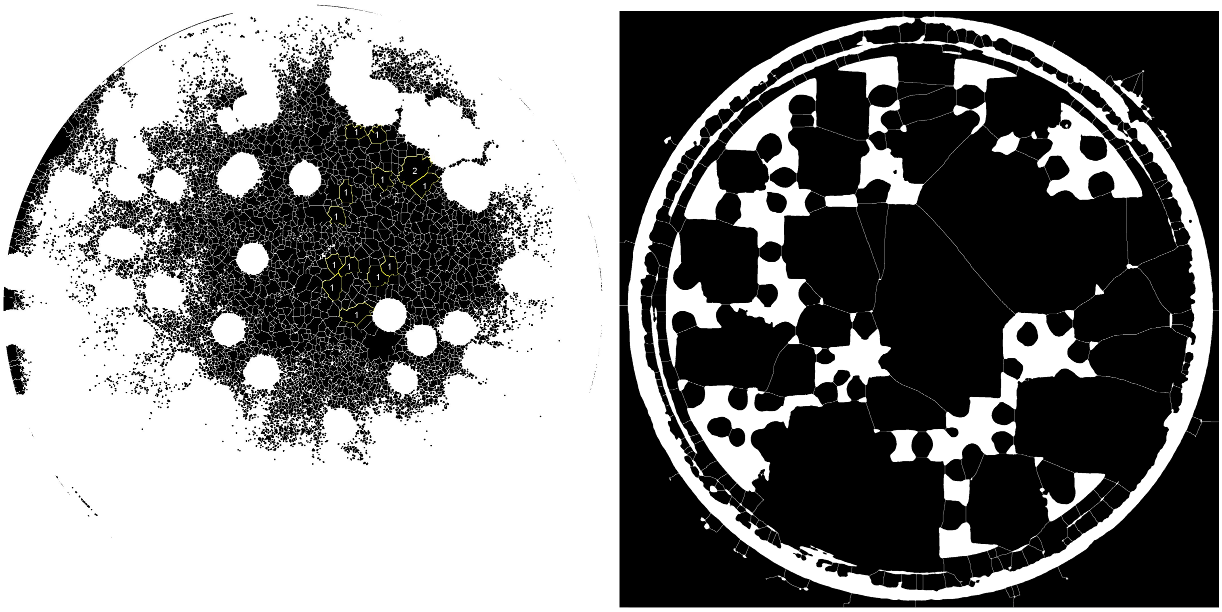
Custom made pipelines and macros mentioned earlier have the main problem of being often highly dependent on the image material for which they were originally customized for, resulting in a poor performance when working with a range of different images. In these instances, users are compelled to do some modifications in the parameter values, and also in the number and order of steps involved. Similar obstacles arise from using software that has little flexibility in their settings.

For CoCoNut the only hypothetical advantage is the possible improvement in accuracy when using their developed hardware with the exact type of cells they used. This statement can only be envision to be the case, since the hardware was not available for testing. The drawbacks on the other hand are obvious after the first use. It requires a random picture to be selected as a background and depending on the selected image the result varies immensely. The user needs to select a region of interest (ROI) with ImageJ which will be used for all the images expecting all the petri dishes to be at the exact same position. Aside from minor usability issues, the only parameter the user can adjust is the radius of the smallest CFU making it impossible for the user to influence CoCoNut in case it deviates from the correct path. This is especially true in the case of lighter cells compared to the background.

Similar to CoCoNut the user has a few requirements to fulfill but has very little influence to steer countPHICS at his will. In this case the user can change values for rolling ball radius, minimum colony size, circularity and sigma. Nevertheless the results of both are unusable as shown in the Fig. 3.30a for CoCoNut and in Fig. 3.30b for countPHICS. Another paper [90] shows a similar approach to the one showed in this work but does not disclose the code for testing their acclaimed abilities.

Others have just recently included deep learning approaches as well as offered the code based on the programming language python [91,92]. Here AutoCellSeg could be used to generate training data for the underlying deep learning model which requires labeled data as well as used complementary to correct the results.

Commercial solutions were not taken into consideration for this work as they are not ac-



(a) (b)

Figure 3.30: Results of CoCoNut (a) and countPHICS (b) on the dark background dataset created for this work. The image in question is ”Klebsiella pneumoniae Kp20161026 24h 60min 24h Kp control 1 24h 60min 24h Kp.JPG”.

cessible to the majority of the users.

4

Summary

Data analysis plays a pivotal role in biological research, empowering scientists to extract meaningful insights from complex datasets. Throughout the collaborative doctoral program titled ”Tissue analytics for stem cell-based diagnostics”, a substantial volume of data has been generated across multiple projects. Manual analysis of this data proved to be resource-intensive and error prone.

The focus of this thesis is on automatic functional and morphological data analysis, as well as quality management. More specifically, it addresses data analysis challenges associated with functional data from transcutaneous measurements of glomerular filtration rate and morphological data from colony forming units.

Transcutaneous measurement of glomerular filtration rate allows for the animal to stay conscious during the non-intrusive procedure and avoids the time consuming serial blood and urine sample collection previously required. The software solutions for the analysis of the acquired data proposed in this work yield outcomes akin to the gold standard, while maintaining intuitiveness and robustness and necessitating fewer interactions. The experimenter is assisted by calculating the elimination kinetics curve after the data is collected. Furthermore, the proposed solution encompasses quality management features, including live correction and duplicate detection functionalities, filtering and overview capabilities as well as automatic curve quality scoring.

A comprehensive analysis was conducted on a dataset comprising over 14000 curves. Out of this extensive pool, 4815 unique curves were identified for subsequent in-depth examination, including health classification and assessment using the introduced ABCD scoring system. The fitting performance was evaluated in comparison to the proprietary software provided by the device manufacturer, revealing a higher degree of consistency across the variability of the dataset. Furthermore, the automation capabilities were assessed using a ground truth dataset, yielding outcomes that closely aligned with the manually obtained references.

Data analysis methods for counting and characterization of colony forming units (CFUs) are developed in order to replace the manual counting with automatic petri dish extraction and detection of each individual CFU. The proposed approach, named AutoCellSeg, not only outperforms the competition in terms of accuracy and usability but also introduces supplementary methods including post-processing, correction, analysis of growth over time, quality management, plotting capabilities, and the ability to be used to generate ground truth datasets for testing or for the learning processes of supervised deep learning models.

AutoCellSeg was tested on five CFU and cell datasets with different characteristics, comprising a total of 111 images, and consistently demonstrated superior performance compared to widely used competitors in this domain. Furthermore, a distinct method allowing for automatic tracking of the growth of CFUs on rotated petri dishes was developed and evaluated successfully on a separate dataset comprising 211 images.

This study has demonstrated that the proposed intuitive solutions can effectively process data either automatically or semi-assisted, thereby enabling users to achieve their objectives more accurately, swiftly, and/or with fewer interactions.

Bibliography

1. Daniele, C., Nardozi, D., Torelli, A., Khan, A.U.M., and Gretz, N. (2020). Transcutaneous Measurement of Glomerular Filtration Rate in Rodents. In Diabetic Nephropathy, L. Gnudi and D. A. Long, eds. (Springer US), pp. 129–137. 10.1007/978-1-4939-9841-8_9.

2. Levey, A.S., Atkins, R., Coresh, J., Cohen, E.P., Collins, A.J., Eckardt, K.-U., Nahas, M.E., Jaber, B.L., Jadoul, M., Levin, A., et al. (2007). Chronic kidney disease as a global public health problem: Approaches and initiatives – a position statement from Kidney Disease Improving Global Outcomes. Kidney Int *72*, 247–259. 10.1038/sj.ki.5002343.

3. Jager, K.J., Kovesdy, C., Langham, R., Rosenberg, M., Jha, V., and Zoccali, C. (2019). A single number for advocacy and communication—worldwide more than 850 million individuals have kidney diseases. Kidney Int *96*, 1048–1050. 10.1016/j.kint.2019.07.012.

4. Bello, A.K., Alrukhaimi, M., Ashuntantang, G.E., Basnet, S., Rotter, R.C., Douthat, W.G., Kazancioglu, R., Köttgen, A., Nangaku, M., Powe, N.R., et al. (2017). Complications of chronic kidney disease: current state, knowledge gaps, and strategy for action. Kidney Int Suppl (2011) *7*, 122–129. 10.1016/j.kisu.2017.07.007.

5. Plantinga, L.C., Tuot, D.S., and Powe, N.R. (2010). Awareness of Chronic Kidney Disease Among Patients and Providers. Adv Chronic Kidney Dis *17*, 225–236. 10.1053/j.ackd.2010.03.002.

6. Naghavi, M., Abajobir, A.A., Abbafati, C., Abbas, K.M., Abd-Allah, F., Abera, S.F., Aboyans, V., Adetokunboh, O., Afshin, A., Agrawal, A., et al. (2017). Global, regional, and national age-sex specific mortality for 264 causes of death, 1980–2016: a systematic analysis for the Global Burden of Disease Study 2016. The Lancet *390*, 1151–1210. 10.1016/S0140-6736(17)32152-9.

7. Jha, V. (2004). End-stage renal care in developing countries: the India experience. Renal Failure *26*, 201–208. 10.1081/jdi-120039516.

8. Silva Junior, G.B. da, Oliveira, J.G.R. de, Oliveira, M.R.B. de, Vieira, L.J.E. de S., and Dias, E.R. (2018). Global costs attributed to chronic kidney disease: a systematic review. Revista da Associação Médica Brasileira *64*, 1108–1116. 10.1590/1806-9282.64.12.1108.

9. Dalrymple, L.S., Katz, R., Kestenbaum, B., Shlipak, M.G., Sarnak, M.J., Stehman-Breen, C., Seliger, S., Siscovick, D., Newman, A.B., and Fried, L. (2011). Chronic Kidney Disease and the Risk of End-Stage Renal Disease versus Death. J Gen Intern Med *26*, 379–385. 10.1007/s11606-010-1511-x.

10. Upadhyay, A., Earley, A., Haynes, S.M., and Uhlig, K. (2011). Systematic Review: Blood Pressure Target in Chronic Kidney Disease and Proteinuria as an Effect Modifier. Ann Intern Med *154*, 541. 10.7326/0003-4819-154-8-201104190-00335.

11. Schiffl, H., Lang, S.M., and Fischer, R. (2002). Stopping smoking slows accelerated progression of renal failure in primary renal disease. J Nephrol *15*, 270–274.

12. Robinson-Cohen, C., Littman, A.J., Duncan, G.E., Weiss, N.S., Sachs, M.C., Ruzinski, J., Kundzins, J., Rock, D., de Boer, I.H., Ikizler, T.A., et al. (2014). Physical Activity and Change in Estimated GFR among Persons with CKD. J. Am. Soc. Nephrol. *25*, 399–406. 10.1681/ASN.2013040392.

13. Navaneethan, S.D., Yehnert, H., Moustarah, F., Schreiber, M.J., Schauer, P.R., and Beddhu, S. (2009). Weight Loss Interventions in Chronic Kidney Disease: A

Systematic Review and Meta-analysis. Clin. J. Am. Soc. Nephro. *4*, 1565–1574. 10.2215/CJN.02250409.

14. Ruggenenti, P., Perna, A., Gherardi, G., Gaspari, F., Benini, R., and Remuzzi, G. (1998). Renal function and requirement for dialysis in chronic nephropathy patients on long-term ramipril: REIN follow-up trial. The Lancet *352*, 1252–1256. 10.1016/S0140-6736(98)04433-X.

15. Brenner, B.M., Cooper, M.E., de Zeeuw, D., Keane, W.F., Mitch, W.E., Parving, H.-H., Remuzzi, G., Snapinn, S.M., Zhang, Z., and Shahinfar, S. (2001). Effects of Losartan on Renal and Cardiovascular Outcomes in Patients with Type 2 Diabetes and Nephropathy. N. Engl. J. Med. *345*, 861–869. 10.1056/NEJMoa011161.

16. Dalal, R., and Sehdev, J.S. (2019). Physiology, Renal Blood Flow and Filtration. In StatPearls (StatPearls Publishing).

17. Lote, C.J. (2012). Principles of Renal Physiology (Springer New York) 10.1007/978-1-4614-3785-7.

18. van der Velde, M., Woodward, M., Levey, A.S., de Jong, P.E., Coresh, J., Gansevoort, R.T., and Consortium, C.K.D.P. (2010). Association of estimated glomerular filtration rate and albuminuria with all-cause and cardiovascular mortality in general population cohorts: a collaborative meta-analysis. Lancet *375*, 2073–2081. 10.1016/S0140-6736(10)60674-5.

19. National Kidney Foundation (2002). K/DOQI clinical practice guidelines for chronic kidney disease: evaluation, classification, and stratification. American Journal of Kidney Diseases: The Official Journal of the National Kidney Foundation *39*, S1-266.

20. Cockcroft, D.W., and Gault, M.H. (1976). Prediction of creatinine clearance from serum creatinine. Nephron *16*, 31–41. 10.1159/000180580.

21. Traynor, J., Mactier, R., Geddes, C.C., and Fox, J.G. (2006). How to measure renal function in clinical practice. BMJ (Clinical research ed.) *333*, 733–737. 10.1136/bmj.38975.390370.7C.

22. Levey, A.S. (1999). A More Accurate Method To Estimate Glomerular Filtration Rate from Serum Creatinine: A New Prediction Equation. Ann Intern Med *130*, 461. 10.7326/0003-4819-130-6-199903160-00002.

23. Michels, W.M., Grootendorst, D.C., Verduijn, M., Elliott, E.G., Dekker, F.W., and Krediet, R.T. (2010). Performance of the Cockcroft-Gault, MDRD, and New CKD-EPI Formulas in Relation to GFR, Age, and Body Size. Clin. J. Am. Soc. Nephro. *5*, 1003–1009. 10.2215/CJN.06870909.

24. Levey, A.S., Stevens, L.A., Schmid, C.H., Zhang, Y.L., Castro, A.F., Feldman, H.I., Kusek, J.W., Eggers, P., Van Lente, F., Greene, T., et al. (2009). A new equation to estimate glomerular filtration rate. Ann Intern Med *150*, 604–612. 10.7326/0003-4819-150-9-200905050-00006.

25. Schwartz, G.J., Haycock, G.B., Edelmann, C.M., and Spitzer, A. (1976). A simple estimate of glomerular filtration rate in children derived from body length and plasma creatinine. Pediatrics *58*, 259–263.

26. Rule, A.D., Larson, T.S., Bergstralh, E.J., Slezak, J.M., Jacobsen, S.J., and Cosio, F.G. (2004). Using Serum Creatinine To Estimate Glomerular Filtration Rate: Accuracy in Good Health and in Chronic Kidney Disease. Ann Intern Med *141*, 929. 10.7326/0003-4819-141-12-200412210-00009.

27. Saleem, M., Florkowski, C.M., and George, P.M. (2008). Comparison of the Mayo Clinic Quadratic Equation with the Modification of Diet in Renal Disease equation and radionuclide glomerular filtration rate in a clinical setting. Nephrology *13*, 684–688. 10.1111/j.1440-1797.2008.01045.x.

28. Levey, A.S., Coresh, J., Balk, E., Kausz, A.T., Levin, A., Steffes, M.W., Hogg, R.J., Perrone, R.D., Lau, J., and Eknoyan, G. (2003). National Kidney Foundation Practice

Guidelines for Chronic Kidney Disease: Evaluation, Classification, and Stratification. Ann Intern Med *139*, 137. 10.7326/0003-4819-139-2-200307150-00013.

29. Filler, G., Witt, I., Priem, F., Ehrich, J.H., and Jung, K. (1997). Are cystatin C and beta 2-microglobulin better markers than serum creatinine for prediction of a normal glomerular filtration rate in pediatric subjects? Clin Chem *43*, 1077–1078.

30. Soveri, I., Berg, U.B., Björk, J., Elinder, C.-G., Grubb, A., Mejare, I., Sterner, G., and Bäck, S.-E. (2014). Measuring GFR: A Systematic Review. Am. J. Kidney Dis. *64*, 411–424. 10.1053/j.ajkd.2014.04.010.

31. Ritter, J., Flower, R.J., Henderson, G., Loke, Y.K., MacEwan, D.J., Robinson, E.S.J., and Fullerton, J. (2024). Rang and Dale’s pharmacology Tenth edition. (Elsevier).

32. Cole, B.R., Giangiacomo, J., Ingelfinger, J.R., and Robson, A.M. (1972). Measurement of Renal Function without Urine Collection: A Critical Evaluation of the Constant-Infusion Technic for Determination of Inulin and Para-Aminohippurate. N. Engl. J. Med. *287*, 1109–1114. 10.1056/NEJM197211302872202.

33. Schwartz, G.J., and Furth, S.L. (2007). Glomerular filtration rate measurement and estimation in chronic kidney disease. Pediatr. Nephrol. *22*, 1839–1848. 10.1007/s00467-006-0358-1.

34. Levey, P.D.A.S. (1990). Measurement of renal function in chronic renal disease. Kidney Int *38*, 167–184. 10.1038/ki.1990.182.

35. Rutili, G., Arfors, K.-E., and Ulfendahl, H.R. (1976). Fluorescence measurements in nanolitre samples. Anal Biochem *72*, 539–545. 10.1016/0003-2697(76)90564-9.

36. Lorenz, J.N., and Gruenstein, E. (1999). A simple, nonradioactive method for evaluating single-nephron filtration rate using FITC-inulin. Am. J. Physiol-renal. *276*, F172–F177. 10.1152/ajprenal.1999.276.1.F172.

37. Pill, J., Kraenzlin, B., Jander, J., Sattelkau, T., Sadick, M., Kloetzer, H.-M., Deus, C., Kraemer, U., and Gretz, N. (2005). Fluorescein-labeled sinistrin as marker of glomerular filtration rate. Eur J Med Chem *40*, 1056–1061. 10.1016/j.ejmech.2005.03.020.

38. Huang, J., Gretz, N., and Weinfurter, S. (2016). Filtration markers and determination methods for the assessment of kidney function. Eur J Pharmacol *790*, 92–98. 10.1016/j.ejphar.2016.06.060.

39. Huang, J., Weinfurter, S., Pinto, P.C., Pretze, M., Kränzlin, B., Pill, J., Federica, R., Perciaccante, R., Ciana, L.D., Masereeuw, R., et al. (2016). Fluorescently Labeled Cyclodextrin Derivatives as Exogenous Markers for Real-Time Transcutaneous Measurement of Renal Function. Bioconjug Chem *27*, 2513–2526. 10.1021/acs.bioconjchem.6b00452.

40. Huang, J., Weinfurter, S., Daniele, C., Perciaccante, R., Federica, R., Della Ciana, L., Pill, J., and Gretz, N. (2017). Zwitterionic near infrared fluorescent agents for noninvasive real-time transcutaneous assessment of kidney function. Chem Sci *8*, 2652–2660. 10.1039/C6SC05059J.

41. Huang, J., and Gretz, N. (2017). Light-Emitting Agents for Noninvasive Assessment of Kidney Function. ChemistryOpen *6*, 456–471. 10.1002/open.201700065.

42. Rabito, C.A., Chen, Y., Schomacker, K.T., and Modell, M.D. (2005). Optical, real-time monitoring of the glomerular filtration rate. Appl. Opt. *44*, 5956. 10.1364/AO.44.005956.

43. Schock-Kusch, D., Xie, Q., Shulhevich, Y., Hesser, J., Stsepankou, D., Sadick, M., Koenig, S., Hoecklin, F., Pill, J., and Gretz, N. (2011). Transcutaneous assessment of renal function in conscious rats with a device for measuring FITC-sinistrin disappearance curves. Kidney Int *79*, 1254–1258. 10.1038/ki.2011.31.

44. Zeneida, H., Weinfurter, S., and Gretz, N. (2016). Transcutaneous assessment of renal function in conscious rodents. J. Visualized Exp., e53767. 10.3791/53767.

45. Frennby, B., and Sterner, G. (2002). Contrast media as markers of GFR. Eur Radiol *12*, 475–484. 10.1007/s003300100864.

46. Thomaseth, K. (1998). Optimal design of a two-sample test for assessing [125I]iothalamate plasma clearance in peritoneal dialysis. Nephrol. Dial. Transpl. *13*, 2265–2270. 10.1093/ndt/13.9.2265.

47. Reinhardt, C.P., Germain, M.J., Groman, E.V., Mulhern, J.G., Kumar, R., and Vaccaro, D.E. (2008). Functional immunoassay technology (FIT), a new approach for measuring physiological functions: application of FIT to measure glomerular filtration rate (GFR). Am. J. Physiol-renal. *295*, F1583–F1588. 10.1152/ajprenal.90354.2008.

48. Rieg, T. (2013). A High-throughput Method for Measurement of Glomerular Filtration Rate in Conscious Mice. J. Visualized Exp., 50330. 10.3791/50330.

49. Shmarlouski, A., Schock-Kusch, D., Shulhevich, Y., Buschmann, V., Rohlicke, T., Herdt, D., Radle, M., Hesser, J., and Stsepankou, D. (2016). A Novel Analysis Technique for Transcutaneous Measurement of Glomerular Filtration Rate With Ultralow Dose Marker Concentrations. IEEE Trans. Biomed. Eng. *63*, 1742–1750. 10.1109/TBME.2015.2501544.

50. Altman, D.G., and Bland, J.M. (1983). Measurement in Medicine: The Analysis of Method Comparison Studies. The Statistician *32*, 307. 10.2307/2987937.

51. Khan, A. ul M., Torelli, A., Wolf, I., and Gretz, N. (2018). AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques. Sci Rep *8*, 7302. 10.1038/s41598-018-24916-9.

52. Mansberg, H.P. (1957). Automatic Particle and Bacterial Colony Counter. Science *126*, 823–827. 10.1126/science.126.3278.823.

53. Analytik, J.A. (1990). Website of ColonyDoc-It by UVP: www.uvp.com/colony-doc-it. Last visited: 15.09.2023.

54. STEMCELL, T. (1993). Website of STEMCELL Technologies product: www.stemcell.com/products/product-types/instruments/stemvision.html. Last visited: 15.09.2023.

55. Bewes, J.M., Suchowerska, N., and McKenzie, D.R. (2008). Automated cell colony counting and analysis using the circular Hough image transform algorithm (CHiTA). Phys Med Biol *53*, 5991–6008. 10.1088/0031-9155/53/21/007.

56. Clarke, M.L., Burton, R.L., Hill, A.N., Litorja, M., Nahm, M.H., and Hwang, J. (2010). Low-cost, high-throughput, automated counting of bacterial colonies. Cytometry. Part A: The Journal of the International Society for Analytical Cytology *77*, 790–797. 10.1002/cyto.a.20864.

57. Brugger, S.D., Baumberger, C., Jost, M., Jenni, W., Brugger, U., and Mühlemann, K. (2012). Automated counting of bacterial colony forming units on agar plates. PloS One *7*, e33695. 10.1371/journal.pone.0033695.

58. Geissmann, Q. (2013). OpenCFU, a new free and open-source software to count cell colonies and other circular objects. PloS One *8*, e54072. 10.1371/journal.pone.0054072.

59. Geissmann, Q. (2013). Website of OpenCFU: opencfu.sourceforge.net. Last visited: 15.09.2023.

60. Carpenter, A.E., Jones, T.R., Lamprecht, M.R., Clarke, C., Kang, I.H., Friman, O., Guertin, D.A., Chang, J.H., Lindquist, R.A., Moffat, J., et al. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol *7*, R100. 10.1186/gb-2006-7-10-r100.

61. Choudhry, P. (2016). High-Throughput Method for Automated Colony and Cell Counting by Digital Image Analysis Based on Edge Detection. PLOS ONE *11*, e0148469. 10.1371/journal.pone.0148469.

62. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. Nat Methods *9*, 676–682. 10.1038/nmeth.2019.

63. Cai, Z., Chattopadhyay, N., Liu, W.J., Chan, C., Pignol, J.-P., and Reilly, R.M. (2011). Optimized digital counting colonies of clonogenic assays using ImageJ software and customized macros: comparison with manual counting. Int J Radiat Biol *87*, 1135–1146. 10.3109/09553002.2011.622033.

64. Kan, A. (2017). Machine learning applications in cell image analysis. Immunol Cell Biol *95*, 525–530. 10.1038/icb.2017.16.

65. Sommer, C., Straehle, C., Kothe, U., and Hamprecht, F.A. (2011). Ilastik: Interactive learning and segmentation toolkit. In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (IEEE), pp. 230–233. 10.1109/ISBI.2011.5872394.

66. Breiman, L. (2001). Random Forests. Mach. Learn. *45*, 5–32. 10.1023/A:1010933404324.

67. Ferrari, A., Lombardi, S., and Signoroni, A. (2017). Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging. Pattern Recognit *61*, 629–640. 10.1016/j.patcog.2016.07.016.

68. Hilsenbeck, O., Schwarzfischer, M., Loeffler, D., Dimopoulos, S., Hastreiter, S., Marr, C., Theis, F.J., and Schroeder, T. (2017). fastER: a user-friendly tool for ultrafast and robust cell segmentation in large-scale microscopy. Bioinformatics *33*, 2020–2028. 10.1093/bioinformatics/btx107.

69. Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput. *22*, 761–767. 10.1016/j.imavis.2004.02.006.

70. Chen, W.-B., and Zhang, C. (2009). An automated bacterial colony counting and classification system. Inform. Syst. Front. *11*, 349–368. 10.1007/s10796-009-9149-0.

71. Zhang, C., Chen, W.-B., Liu, W.-L., and Chen, C.-B. (2008). An Automated Bacterial Colony Counting System. In 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008) (IEEE), pp. 233–240. 10.1109/SUTC.2008.50.

72. Ates, H., and Gerek, O.N. (2009). An image-processing based automated bacteria colony counter. In 2009 24th International Symposium on Computer and Information Sciences (IEEE), pp. 18–23. 10.1109/ISCIS.2009.5291926.

73. Torelli, A. (2016). Github repository of AutoCellSeg benchmark data: github.com/AngeloTorelli/AutoCellSeg/tree/master/DATA/Benchmark. Last visited: 15.09.2023. GitHub.

74. Geissmann, Q. (2013). Website for OpenCFU S. aureus dataset: sourceforge.net/projects/opencfu/files/samples/EcoliRobEgbert.zip. Last visited: 15.09.2023.

75. Geissmann, Q. (2013). Website for OpenCFU RFP/GFP E. coli dataset: sourceforge.net/projects/opencfu/files/samples/plosPicHQ.zip. Last visited: 15.09.2023.

76. Broad, I. (2004). Website of Broad Bioimage Benchmark Collection: https://data.broadinstitute.org/bbbc/image_sets.html. Last visited: 15.09.2023.

77. Ljosa, V., Sokolnicki, K.L., and Carpenter, A.E. (2012). Annotated high-throughput microscopy image sets for validation. Nat Methods *9*, 637–637. 10.1038/nmeth.2083.

78. Vincent, L., and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell *13*, 583–598. 10.1109/34.87344.

79. Sethian, J.A. (1996). A fast marching level set method for monotonically advancing fronts. Proc. Natl. Acad. Sci. *93*, 1591–1595. 10.1073/pnas.93.4.1591.

80. Khan, A. ul M., Mikut, R., and Reischl, M. (2015). A Benchmark Data Set to Evaluate the Illumination Robustness of Image Processing Algorithms for Object Segmentation and Classification. PLOS ONE *10*, e0131098. 10.1371/journal.pone.0131098.

81. Wang, L., Fan, D., Chen, W., and Terentjev, E.M. (2015). Bacterial growth, detachment and cell size control on polyethylene terephthalate surfaces. Sci Rep *5*, 15159. 10.1038/srep15159.

82. Chen, Y., and Medioni, G. (1992). Object modelling by registration of multiple range images. Image Vision Comput. *10*, 145–155. 10.1016/0262-8856(92)90066-C.

83. Besl, P.J., and McKay, N.D. (1992). A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell *14*, 239–256. 10.1109/34.121791.

84. Jones, T.R., Carpenter, A., and Golland, P. (2005). Voronoi-Based Segmentation of Cells on Image Manifolds. In Computer Vision for Biomedical Image Applications Lecture Notes in Computer Science., D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, et al., eds. (Springer Berlin Heidelberg), pp. 535–543. 10.1007/11569541_54.

85. Rand, W.M. (1971). Objective Criteria for the Evaluation of Clustering Methods. J Am Stat Assoc *66*, 846–850. 10.1080/01621459.1971.10482356.

86. Jain, V., Bollmann, B., Richardson, M., Berger, D.R., Helmstaedter, M.N., Briggman, K.L., Denk, W., Bowden, J.B., Mendenhall, J.M., Abraham, W.C., et al. (2010). Boundary Learning by Optimization with Topological Constraints. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (IEEE), pp. 2488–2495. 10.1109/CVPR.2010.5539950.

87. Arganda-Carreras, I., Turaga, S.C., Berger, D.R., Cireşan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J.M., et al. (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. Front Neuroanat *9*. 10.3389/fnana.2015.00142.

88. Siragusa, M., Dall'Olio, S., Fredericia, P.M., Jensen, M., and Groesser, T. (2018). Cell colony counter called CoCoNut. PLOS ONE *13*, e0205823. 10.1371/journal.pone.0205823.

89. Brzozowska, B., Gałecki, M., Tartas, A., Ginter, J., Kaźmierczak, U., and Lundholm, L. (2019). Freeware tool for analysing numbers and sizes of cell colonies. Radiat Environ Biophys *58*, 109–117. 10.1007/s00411-018-00772-z.

90. Boukouvalas, D.T., Prates, R.A., Lima Leal, C.R., and de Araújo, S.A. (2019). Automatic segmentation method for CFU counting in single plate-serial dilution. Chemom. Intell. Lab. Syst. *195*, 103889. 10.1016/j.chemolab.2019.103889.

91. Zhang, L. (2022). Machine learning for enumeration of cell colony forming units. Vis Comput Ind Biomed Art *5*, 26. 10.1186/s42492-022-00122-3.

92. Nunn, C.J., Klyshko, E., and Goyal, S. (2023). petiteFinder: an automated computer vision tool to compute Petite colony frequencies in baker's yeast. BMC Bioinformatics *24*, 50. 10.1186/s12859-023-05168-5.

Curriculum Vitae

Personal Information

Name: Angelo Felice Torelli

Date of birth: May 29, 1987

Place of birth: Barquisimeto, Venezuela

Nationality: German

Marital status: Divorced, two children

Education

09/97 - 07/05	European Baccalaureate European School in Karlsruhe, DE
10/06 - 09/08	Diploma in Computer Science Karlsruhe Institute of Technology, DE
10/08 - 06/12	Bachelor of Science in Computer Science, Thesis: Influence of Agility on Technical Implementations, Karlsruhe University of Applied Science, DE
03/11 - 05/11	Semester abroad for a collaboration project, University of Puget Sound in Tacoma, Washington, USA
10/12 - 05/14	Master of Science in Medical Computer Science, Thesis: Computer Simulation of Urothelial Cells, University of Applied Science in Mannheim, DE
05/16 -	PhD in Medical Computer Science, Dissertation: New tools for functional and morphological data analysis and quality management in biological research University of Heidelberg, DE

Military Service

10/05 - 06/06	ABC-defense soldier, barracks, Bruchsal, DE

Language Proficiency

Danish: Beginner

Spanish: Conversant

English: Fluent

German, Italian: Native speaker

Technical Proficiencies

Suite: Microsoft Office, LibreOffice, Google G Suite

OS: Windows, Ubuntu, MacOS, FreeNAS

Softwares: ImageJ, FIJI, CellProfiler, LAS-X, CompuCell3D, PhotoShop, Ilastik

Languages: Python, MATLAB, Java, JavaScript, LateX, SAS, HTML, CSS, XML, C#, PHP, SQL, C++

Frameworks: Tensorflow, Keras, PyTorch, Caffe 2, Theano, Fast.ai, PyCharm, IntelliJ, Sublime Text 2

Professional Background

03/09 - 07/09	Teaching assistant, Maths 2, Karlsruhe University of Applied Sciences, DE
08/09 - 12/09	Software engineer intern, Topic: Creation of dynamic web pages based on Java, ZK, CSS and JavaScript Forschungszentrum Informatik, Karlsruhe, DE
10/13 - 02/14	Lab instructor, Algorithms and Data Structures, Mannheim University of Applied Sciences, DE
10/13 - 02/14	Lab instructor, Techniques of Program Development, Mannheim University of Applied Sciences, DE
06/14 - 04/16	Research assistant, Topic: Simulations of the urothelim Mannheim University of Applied Sciences, DE
10/20 - 10/21	Machine Learning Specialist, Cercare Medical, Aarhus, DK

Publications

Angelo Torelli, Fabian Siegel, Philipp Erben, Markus Gumbel. “Modeling of the Urothelium with an Agent Based Approach”. Lecture Notes in Computer Science, Vol. 9044. Springer, 2015.

Arif ul Maula Khan, Angelo Torelli, Ivo Wolf, Norbert Gretz. “AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques”. Scientific Reports, 8(7302), 2018.

Carolina Gamez, Arif ul Maula Khan, Angelo Torelli, Barbara Schneider-Wald, Norbert Gretz, Ivo Wolf, Karen Bieback, Markus Schwarz. “Image processing workflow to visualize and quantify MSCs in 3D”. German Congress of Orthopedic and Trauma Surgery (DKOU 2018), Nov. 6, 2018.

Cristina Daniele, Daniela Nardozi, Angelo Torelli, Arif ul Maula Khan, Norbert Gretz. “Transcutaneous measurement of glomerular filtration rate in rodents”. Pages 129-137. Luigi Gnudi, David A. Long. Diabetic Nephropathy, Methods and Protocols. Springer US, 2020.

September 30, 2023