

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

CAMPUS DI CESENA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA BIOMEDICA

TESI DI LAUREA

in

Bioingegneria molecolare e cellulare LM

**PHAGE THERAPY: A SOFTWARE SYSTEM FOR
PHAGE QUANTIFICATION
AND
KINETIC MODEL INFERENCE**

CANDIDATA:
Nan Papili Gao

RELATORE:
Prof. Emanuele D. Giordano

CORRELATORE:
Prof. Alfonso Jaramillo

Sessione III
Anno Accademico 2013/2014

Abstract

One of the most serious problems of the modern medicine is the growing emergence of antibiotic resistance among pathogenic bacteria. In this circumstance, different and innovative approaches for treating infections caused by multidrug-resistant bacteria are imperatively required. Bacteriophage Therapy is one among the fascinating approaches to be taken into account. This consists of the use of bacteriophages, viruses that infect bacteria, in order to defeat specific bacterial pathogens. Phage therapy is not an innovative idea, indeed, it was widely used around the world in the 1930s and 1940s, in order to treat various infection diseases, and it is still used in Eastern Europe and the former Soviet Union. Nevertheless, Western scientists mostly lost interest in further use and study of phage therapy and abandoned it after the discovery and the spread of antibiotics. The advancement of scientific knowledge of the last years, together with the encouraging results from recent animal studies using phages to treat bacterial infections, and above all the urgent need for novel and effective antimicrobials, have given a prompt for additional rigorous researches in this field. In particular, in the laboratory of synthetic biology of the department of Life Sciences at the University of Warwick, a novel approach was adopted, starting from the original concept of phage therapy, in order to study a concrete alternative to antibiotics. The innovative idea of the project consists in the development of experimental methodologies, which allow to engineer a programmable synthetic phage system using a combination of directed evolution, automation and microfluidics. The main aim is to make “the therapeutics of tomorrow individualized, specific, and self-regulated” (Jaramillo, 2015). In this context, one of the most important key points is the Bacteriophage Quantification. Therefore, in this research work, a mathematical model describing complex dynamics occurring in biological systems involving continuous growth of bacteriophages, modulated by the performance of the host organisms, was implemented as algorithms into a working software using MATLAB. The developed program is able to predict different unknown concentrations of phages much faster than the classical overnight Plaque Assay. What is more, it

gives a meaning and an explanation to the obtained data, making inference about the parameter set of the model, that are representative of the bacteriophage-host interaction.

This thesis is dedicated to my parents.
For their endless love, support and encouragement.

Table of contents

Abstract	1
List of Figures	7
Introduction	10
1 The protagonists of Bacteriophage therapy	11
1.1 Bacteria	11
1.1.1 Bacterial growth	12
1.1.2 Escherichia Coli as phage host.....	14
1.2 Bacteriophages.....	16
1.2.1 Morphology of bacteriophages.....	16
1.2.2 Classification of bacteriophages.....	17
1.2.3 T7 Bacteriophages	18
1.3 Comparison between bacteriophage therapy and antibiotics	19
1.4 Overview of the presented research.....	21
2 Phage quantification and parameters inference	26
2.1 The initial idea of the project.....	26
2.2 Models for the bacteriophage-host interactions.....	27
2.2.1 Model formulations	28
2.2.2 The first data and the final model.....	31
3 Materials and methods	34
3.1 Experimental methods	34
3.1.1 Enumeration of bacteria	34
3.1.2 Enumeration of bacteriophage particles	36
3.1.3 Multiplicity of infection	37

3.1.4 Phage-host interaction experiment.....	38
3.1.5 Optical density - CFU/mL calibration.....	42
3.1.6 Definition of delay model parameters.....	43
3.2 Software implementation.....	50
3.2.1 Plot and summary of the OD measurement results.....	50
3.2.2 Plot and summary of the OD measurement results.....	53
3.2.3 First fitting attempt and parameter research.....	55
3.2.4 Final parameter optimization.	58
3.2.5 Model prediction	62
4 Results and Discussion.....	66
4.1 CFU/mL.....	66
4.2 PFU/mL	67
4.3 MOI.....	68
4.4 A1mainPLOT.M.....	71
4.5 A2odCFUcalibration.m	72
4.6 A3mainFIT.m and A4mainCROSS.m.....	74
A.7 Experimental parameters	75
4.8 Comparison between experimental and model optimized parameter values	80
4.9 A5mainPREDITCION.m-Validation step.....	80
5 Conclusion and future directions	83
5.1 Conclusion	83
5.2 Future directions	83
Appendix-Matlab Codes.....	86
1 A1mainPLOT.m	86
1.1 sigmFITdata.m	92
1.2 sigmoide.m	94
A2odCFUcalibration.m	95

3	A3mainFIT.m	97
	3.1 helper.m	101
	3.2 cairns.m	102
	3.3 MyEventFunction.m.....	103
A4	mainCROSS.m.....	104
	4.1 helper4.m	107
5	A5mainPREDICTION.m.....	108
6	Other MATLAB code used:	112
	6.1 rgb.m.....	112
	6.2 herrorbar.m	112
	References	113

List of Figures

- Figure 1: Most bacteria have diameters in the range of 1 μm (micrometre) to 5 μm . From left to right: Spherical (cocci) bacteria, rod-shaped (bacilli) bacteria, Spiral bacteria.
(<http://www.ppdictionary.com/gnbac.htm>) 11
- Figure 2: Bacteria during the process of binary fission.
(http://www.sflorg.com/sciencenews/scn061606_01.html) 12
- Figure 3: The typical bacterial growth curve. (Alcamo I. Edward, 2001)
..... 14
- Figure 4: Difference between F-positive (on the left) and F-negative (on the right) cells. (<http://leavingbio.net/bacteria%20page.htm>) 15
- Figure 5: The typical structure of a bacteriophage.
(<http://en.wikipedia.org/wiki/Bacteriophage>)..... 16
- Figure 6: The lytic cycle of a T7 bacteriophage.
(<http://textbookofbacteriology.net/phage.html>) 18
- Figure 7: Overview of the PACE system.
(http://www.nature.com/nature/journal/v472/n7344/fig_tab/nature09929_F1.html) 23
- Figure 8: The accessory plasmid pet24_gp5 with Kanamycin resistance.
..... 24
- Figure 9: Phage and host evolution. (Hassall, J., Jaramillo, A. (2014). Engineering synthetic bacteriophage cocktails. Poster, University of Warwick) 25
- Figure 10: Optical density measurement over time of the concentration of bacteria infected by different dilution of phages..... 32
- Figure 11: Serial dilution.
(http://2014.igem.org/Team:CSU_Fort_Collins/Notebook/KillSwitch/Sep) 35
- Figure 12: Infinite® F500 Tecan microplate reader 39

- Figure 13: Standard disposition in a 96-well plate.....40
- Figure 14: The use of a multichannel pipette for the dispensation of phages and medium in the cells.41
- Figure 15: Log_{10} of the cell number over time.
<http://colinmayfield.com/biology447/modules/intro/bacterialgrowthrate.htm>44
- Figure 16: a-Latent period experiment on the stir plate. b- Double filtering.....47
- Figure 17: Standard protocol for the OD_{600} measurement in a microplate reader.....51
- Figure 18: Example of a logistic function (red) and an “inverted” logistic function (blue).52
- Figure 19: A template for the OD-CFU/mL calibration results. $X=\text{OD}$; $Y=\text{CFU/mL}$53
- Figure 20: a-Matlab Curve Fitting application. b-Matlab Ginput command.....54
- Figure 21: The hold-out data split. (<http://scott.fortmann-roe.com/docs/MeasuringError.html>)58
- Figure 22: K-fold Cross Validation with $k=3$ and dataset of 30 elements.
(http://homepages.inf.ed.ac.uk/pmartin/tutorial/case_studies.html)59
- Figure 23: The leave one out split. Every fold represents only one sample and $k=\text{number of samples}$.
(<https://www.packtpub.com/books/content/learning-how-classify-real-world-examples>).....59
- Figure 24: Scatter plot of the variables X and Y.
(<http://math.tutorvista.com/statistics/correlation-and-regression.html>) 63
- Figure 25: Bacterial colonies after an overnight incubation at 37°C66
- Figure 26: Viral Plaques after an overnight incubation at 37°C . Plaques are circular and clear zones formed in a lawn of cells due to lysis by phage.67
- Figure 27: Percentage of infected cells at different MOI.....68

- Figure 28: Curves with the same phage dilution (the same colour) but with different starting concentration of cells. X=time; Y=OD. 70
- Figure 29: Curves after normalization, every colour represents a different dilution of phage. Solid lines: starting OD (0.2) value achieved directly from the growth of a low bacterial concentration (0.04). Dotted lines: starting OD (0.2) value achieved diluting a high bacterial concentration (0.7)..... 70
- Figure 30: The plot over time of a single replicate with all its samples. 71
- Figure 31: The command window of *AImainPLOT.m*..... 71
- Figure 32: The summary of the data in mean curves. Each black line represents the standard deviation of the respective curve..... 72
- Figure 33: OD-CFU/mL calibration results..... 73
- Figure 34: The plot of the linear fitting equation..... 73
- Figure 35: The curve fitting with the optimized parameter set..... 75
- Figure 36: Experimental growth rate. 76
- Figure 37: One-step growth for the Latent Period determination..... 77
- Figure 38: One-step growth for the Binding Rate determination. 78
- Figure 39: Decay Rate Experiment..... 79
- Figure 40: The results of the prediction in the command window. 81
- Figure 41: Examples of the prediction for different mean curves. 81
- Figure 42: The variability for the samples of every phage dilution..... 82
- Figure 43: The curve fitting results of the interaction between MG1655 and T7 Δ gp5. 84
- Figure 44: An example of the growth curve as result of the interaction of 100 cells and 10 phages..... 85

Introduction

Phage therapy is the use of bacteriophages, also known as phages, as antimicrobial agents for the treatment of pathogenic bacteria and other problems. In recent years, the recognition of a phenomenon antibiotic resistance as a major healthcare issue has led to renew interest in alternative therapies including bacteriophage therapy (Merril & Adhya, 2003; Hanlon GW, 2007). The project of Professor Alfonso Jaramillo's laboratory at the University of Warwick is part of this research orientation. In particular, its main purpose is the continuous production and directed evolution of synthetic bacteriophage cocktails using techniques and technologies that come from Biology, Chemistry and Engineering. In this context, one of the issues and urgencies is the phage quantification, in other words, the determination of the phage concentration. The classical method for the count of the number of phages, or simply the Phage Titer, is the Bacteriophage Plaques Assay. However, in order to have some results with this technique, it is necessary to await up to 24 hours.

In this thesis, a delay model was implemented and solved in Matlab environment, in order to obtain a faster prediction and quantification of a specific unknown concentration of phages and to make inference about the model parameters. The first chapter provides a succinct description of bacteria and bacteriophages, the basic interactions among them and the reason why the scientific research has focused the attention again on the phage therapy. In the second chapter, mathematical models which represent the dynamics of Bacteriophage-Host interaction are described in detail. The experimental techniques adopted and the architecture of the software implementation are examined in the third chapter. The experimental details and results are reported in the fourth chapter. Finally, the conclusion of the work are summarised in the last chapter.

Chapter 1:

1 The protagonists of Bacteriophage therapy

1.1 Bacteria

Due to the presence of a rigid wall, bacteria maintain a definite shape. However, bacteria come in a wide variety of shapes, sizes and structures and most of them are very small, in the order of a few micrometers (10^{-6} meters). The most common shapes are the rod-like (Bacillus), the spherical (Coccus), and the spiral (Spiral) (Figure 1).

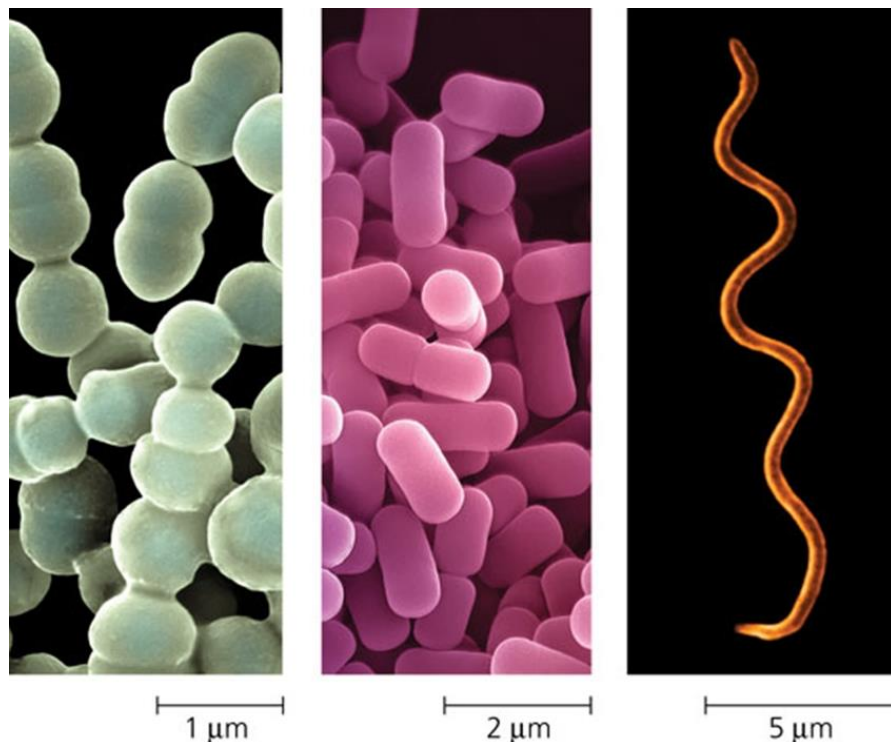


Figure 1: Most bacteria have diameters in the range of 1 μm (micrometre) to 5 μm . From left to right: Spherical (cocci) bacteria, rod-shaped (bacilli) bacteria, Spiral bacteria. (<http://www.ppdictionary.com/gnbac.htm>)

A second major criterion for distinguishing bacteria is based on the cell wall structure. Using a series of stains and reagents called the Gram stain, the cells wall can give different staining characteristics. In particular, with this method of differentiation, bacteria that have a thin layer and an outer membrane stain red are called Gram negative while bacteria with a thicker wall layer, lacking the outer membrane, stain violet are called Gram positive.

1.1.1 Bacterial growth

Most bacteria share one major characteristic: they reproduce by an asexual process called “binary fission” (Figure 2). This means that the cell elongates and grows to about double its original size and then splits, after the equal division of the nuclear materials, into two genetically identical daughter cells called clones.

Growing bacteria on a solid surface as agar on a Petri dish, it is possible to distinguish different colonies. A colony is composed by all the progeny of a single original cell.



Figure 2: Bacteria during the process of binary fission.
(http://www.sflorg.com/sciencenews/scn061606_01.html)

It is simple to note that the mathematical series describing bacterial growth is: 1, 2, 4, 8, 16, etc. It can be also written as a series in base 2: 2^0 , 2^1 , 2^2 , 2^3 , 2^4 , etc. For this reason, bacteria show an exponential growth since the number that increases in the series is the exponent. In reality, exponential growth is only one phase of the bacterial life cycle.

When a given number of bacteria is inoculated in a close system (or batch culture), like a test tube with fresh medium, it is possible to monitor the population growth trend over time. The graph that plots cell number versus time is called bacterial growth curve and usually the cell number is expressed as a log. In a growing culture, the plot gives a characteristic curve in which four phases of the bacterial growth cycle are distinguished: lag phase, exponential (log) phase, stationary phase, and death phase (Figure 3).

The lag phase is the first phase observed. Immediately after inoculation of bacteria into a new growth medium, there is no change in the cell number. However, the population can grow in volume or mass, synthesizing proteins, RNA, etc., and increase in metabolic activity in preparation for the cell division. The lag phase may be short or very long; its duration depends on several factors including recovery time necessary after the shock and/or damage in the transfer; time required for the production of division factors; and the quantity of the starting inoculum.

During the second phase called exponential or log phase, the cells begin DNA replication and shortly after they divide by binary fission. This is the period where the cells can grow most quickly, with a geometric progression. The time necessary for the culture to double is called “generation (or doubling) time”. This is represented as a fairly constant rate and it can be easily obtained from the graph. The exponential growth leads to rapidly increasing population but it cannot last forever in a batch culture because of the exhaustion of available nutrients; accumulation of end products and inhibitory metabolites; lack of “biological space” due to the increase of the population density.

The third phase in the bacterial growth is the stationary phase, that is not necessarily a period of quiescence. During this phase bacteria can divide slowly for a time, but soon stop dividing completely. Viable cells counted maintain a slow metabolic activity; if they are diluted into fresh medium they can quickly adopt again an exponential growth.

In the last phase, if there is no addition of new medium, the number of viable cells progressively decreases. Like the log phase, the death phase follows an exponential trend and within hours the

culture may not have any living cells. It is important to underline that most of dead bacteria look identical to live cells, so normal spectrophotometer measurements or microscopic count are not an indication of the observation of live cells. Hence, in order to have a culture with the maximum number of viable cells, it is best to grow bacteria only until early stationary phase and then refresh them with new medium.

The duration of the phases of the growth curve can be slightly different for different conditions of pH, oxygen, salt concentration, temperature, nutrients and for different types of bacteria.

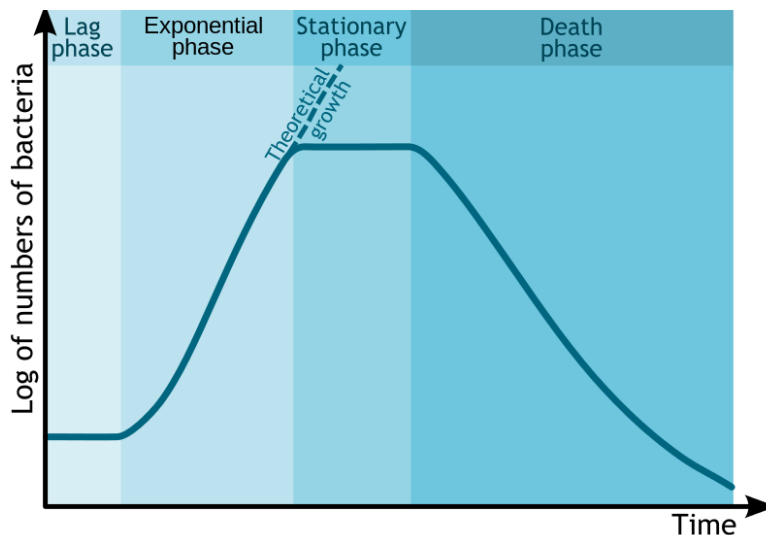


Figure 3: The typical bacterial growth curve. (http://en.wikipedia.org/wiki/Bacterial_growth)

1.1.2 *Escherichia Coli* as phage host

Escherichia Coli (or simply called *E. Coli*) is a gram negative, rod-shaped bacterium. It is a common inhabitant of the lower intestinal part of man and warm-blooded animals. Most strains of *E. Coli* are harmless and they are part of the normal microflora of the gut. Their principal jobs are the suppression of harmful bacteria and the production of vitamins. Nevertheless, some species of *E. Coli* can be responsible of foodborne illness in their hosts.

In this project, a particular strain of *E. Coli* was adopted as bacteriophage host. There are many reasons that can support this

choice. Firstly, in the last 60 years E. Coli has been the subject of intensive research and now it is the most widely studied prokaryotic organism. In particular, it is considered as important host specie in Biotechnology and Biology where it is commonly used for the manipulation of recombinant DNA. Secondly, this type of bacteria can survive outside the body, also at freezing temperatures, so they can easily grow in a laboratory environment. Lastly, E. Coli represents a powerful model system thanks to its ability to grow in chemically prepared media and its high grow rate. Indeed, under favourable conditions of temperature, pH, etc., its doubling time is roughly 20-30 minutes.

Another important point to highlight is that the bacterial strain chosen as phage host is an F- (F minus) mutant. Usually, some E. Coli strains can carry an F-plasmid or also called fertility factor, because thanks to the production of the sex pilus, the bacterial conjugation can take place. It is an episome, in other words a plasmid that can integrate itself into bacterial chromosome by homologous recombination. There can be only one copy of the F-plasmid in a given bacterium and bacteria that possess a copy are called F-positive or F+ (F plus) while cells that lack F-plasmid are called F-negative or F- (F minus) (Figure 4). In order to avoid biofilm creation, a mutant F- strain of E. Coli was adopted in this work.

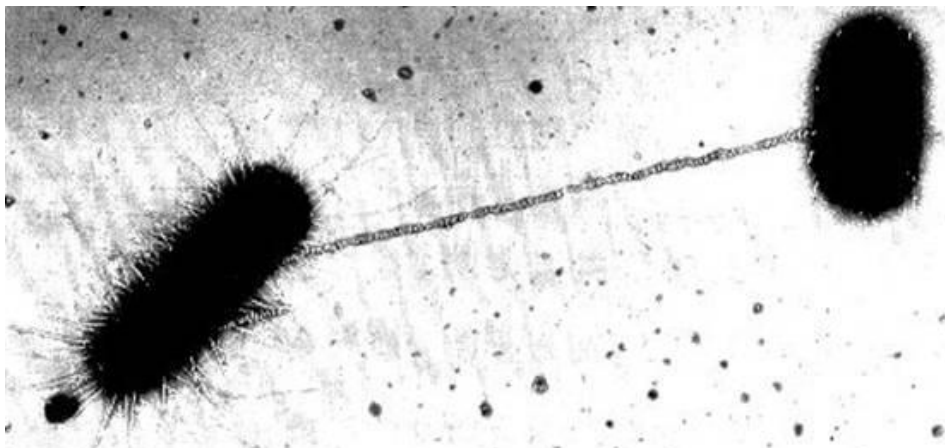


Figure 4: Difference between F-positive (on the left) and F-negative (on the right) cells. (<http://leavingbio.net/bacteria%20page.htm>)

1.2 Bacteriophages

During the 1890s, many bacteriologists had observed something that seemed to limit bacterial growth and activity. In particular the British Ernest Hanbury Harkin reported the presence of an unknown substance with an evident antibacterial activity against Cholera disease in the waters of rivers in India (Harkin,1896). But the official discover of Bacteriophages occurred only in 1915, when the English Frederick Twort observed a growing bacterial culture killed by an small agent and hypothesized that it could be a virus. The name Bacteriophage was coined by a Canadian biologist called Félix d’Hérelle two years later, in 1917, when he discovered “an invisible, antagonistic microbe of the dysentery bacillus” (d’Herelle, 1917). The meaning of the term “Bacteriophage” is “bacteria-eater” from the Greek word “phagein” which means “to devour”. So Bacteriophages, also known as phages simply, are viruses that can attack and kill bacteria. They are parasites and for this reason they need a bacterial host in order to replicate themselves. Phages, like bacteria, are easily findable in nature. For instance, they can be isolates from sewage, soil and feces.

1.2.1 Morphology of bacteriophages

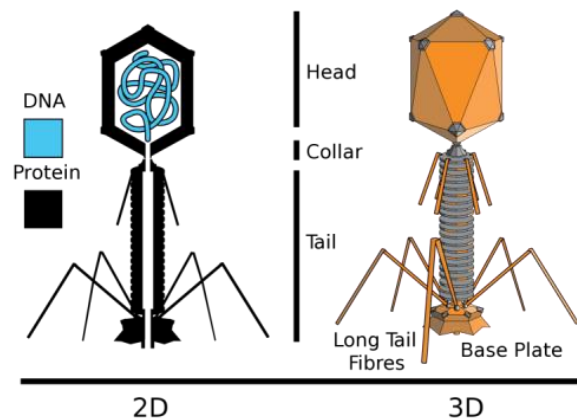


Figure 5: The typical structure of a bacteriophage. (<http://en.wikipedia.org/wiki/Bacteriophage>)

It is possible to find phages with many different sizes and shapes but many of them share some common features. Like other viruses, bacteriophages are simple organisms which consist of a head, or capsid, and a tail (Figure 5). The capsid is composed by one or more proteins and its job is to coat and protect the genetic material (nucleic acid) that may be either DNA or RNA, double- or single-stranded. Most phages have also a tail that is a narrow tube indispensable for the viral DNA injection in bacterial hosts. Some of more complex phages have one or more tail fibers which ease the coupling of the phage to a bacterial cell.

1.2.2 Classification of bacteriophages

A wide range and variety of phages exists in nature, they can be classified in at least 12 different groups considering several factors such as viral morphology, genome type, auxiliary structures and their host preference. Bacterial viruses are very specific; they may infect only one or a limited number of bacteria. Therefore, they are usually named according to the bacteria group, strain that they infect. For instance, phages that infect the bacterium *E. Coli* are called coliphages.

A second criterion for the classification of bacteriophages is based on how they infect host cells and reproduce themselves. Indeed, it is possible to distinguish two principal categories of phages according to their life cycle: lytic (virulent) or lysogenic (temperate). The main difference between these two types of cycle is that during a lytic infection, after the injection of the genetic material into the host, phages multiply and kill the cell by lysis in order to release new viruses. Conversely, during the lysogenic cycle, phages are able to include their DNA or RNA into the cell chromosome and replicate it without bursting the host. Only the lytic phages are a good choice for developing therapeutic phage preparation because lysogenic phages may not destroy bacteria immediately and in addition they may transfer virulence genes and those mediating antibiotic-resistance to other bacteria (Sandeep, 2006).

1.2.3 T7 Bacteriophages

T7 phages are also called coliphages in so far as they can infect most strains of E. Coli. Their genetic material is composed by double-stranded DNA and this viral chromosome is coated and protected by a protein capsid. In addition to the head, T7 phages also possess a tail and some auxiliary structures. Bacteriophage T7 was discovered in 1945 and as other six members of the group T-phages, it has a lytic life cycle because it always brings about the death of its hosts by lysis after the infection (Figure 6).

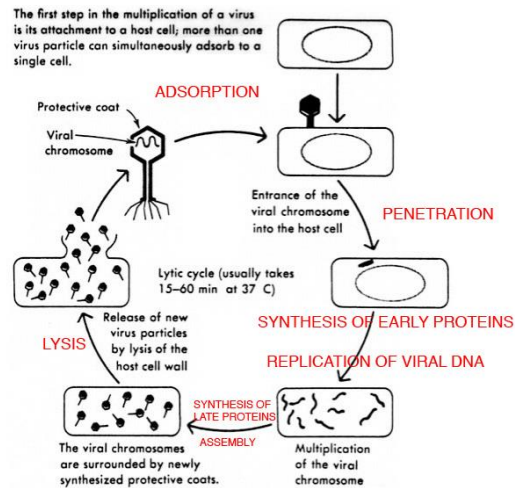


Figure 6: The lytic cycle of a T7 bacteriophage. (<http://textbookofbacteriology.net/phage.html>)

In particular, the first step of the replication is the “adsorption”. During this phase, T7 phage recognizes specific receptors on the bacterial surface then adheres to those sites using the tail fibers. The irreversible binding of the phage to the host is followed by a second phase called “penetration” where T7 phage injects its viral DNA by means of its hollow tail that work like a syringe while the capsid remains outside the cell. Soon after the penetration phase, the phage life cycle enters in “eclipse” period. This phase represents the interval between the injection of viral genome into E. Coli and the release of new phages. The “eclipse” phase is dedicated firstly, to the synthesis of a set of early proteins required for the replication of the phage DNA; secondly to the production of several copies of phage nucleic acid; lastly to the synthesis of late proteins which are mainly structural proteins of the capsid and the

tail. After the production of all these parts, the assembly phase, better known as “maturation”, takes place. Viral genome is packaged into the capsid and the tail is linked to it, new mature infective phages are accumulated inside the host until a limit, called also the burst size. This is the last phase of the phage life cycle, it induces the “lysis” of the host and the release of intracellular bacteriophages that are able to infect and spread to nearby cells. All these phases take about 12-35 minutes to complete, so T7 phage has a short life cycle and an excellent ability to expand exponentially. Consequently, it is able to defeat pathogenic bacteria rapidly. In addition to its fast growth rate, T7 phage has shown a high flexibility and adaptivity that suggest it befitting for exploring evolutionary principles and for developing new tools to overcome the antibiotic resistance issue. (Qimron et al, 2010).

The points listed above are some of the several properties that demonstrate the reason why T7 phage was adopted in this project.

1.3 Comparison between bacteriophage therapy and antibiotics

Phage therapy and traditional antibiotics have the same goal: defeat and kill pathogenic bacteria. Despite that, they have different behaviours and working principles, therefore different pros and cons. In particular, adopting phages as therapeutic agents, it is possible to list several advantages over antibiotics and the mains are (Chhibber et al., 2012; Sandeep, 2006):

- 1) Phages have a very high specificity and thanks to this they are not harmful for useful bacteria that live in and on the human body. Consequently there are not side effects like secondary infections or intestinal disorders that are typical cases of many antibiotic treatments. Furthermore, due to their high specificity, it is unlikely to obtain a selection for phage resistance in non-target bacteria while using

antibiotics with a broad spectrum activity is possible to select for resistance of many mutated pathogenic bacteria.

- 2) It is very easy to find phages throughout the nature, it is estimated that there are about 10^{31} phages on earth and as a result of that, viruses are the most abundant life form. Using new phages is possible to overcome the issue of phage-resistant bacteria. Conversely, the development of new antibiotic in order to defeat antibiotic-resistant bacteria requires many years to accomplish.
- 3) Phages have a bactericidal action and usually a single dose of them is sufficient to kill pathogens because they are an 'intelligent' drug: they have an exponentially growth and they can multiply at the site of infection as long as there are bacteria to destroy. Once they have reached the purpose, phage too will die because they cannot grow without their specific host. Antibiotics can be bactericidal but also bacteriostatic, and usually it is necessary to administer repeated doses of the drug.
- 4) Phage therapy may become a useful alternative for patients that are allergic to antibiotics and it can be administered in different way in the form of pills, injections, etc.

Considering the features above, it is possible to look at phage therapy as a valid alternative approach for the treatment of superbugs, bacteria that are resistant to multiple antibiotics, and in general of bacterial infections. However it is necessary to overcome some problems associated with the use of phages (Chhibber & Kumari, 2012; Sandeep, 2006):

- 1) Their specificity can be a disadvantage when the exact species of infecting bacteria is unknown or in presence of a multiple infection. Because of this downside, it is necessary an identification prior of the pathogenic infection in the laboratory. Conversely, one type of antibiotic can defeat many different species of bacteria.
- 2) Cases of bacteriophage ineffective action were reported, maybe because of an incorrect diagnosis or choice of the method of phage administration. In particular oral phage administration could be neutralized by the gastric acidity; phage injected into the bloodstream are recognized by the immune system that can produce antibodies against these

viruses after a certain period; bacteriophages cannot be used against intracellular bacteria because this host is not available for the interaction.

- 3) Bacteria can also develop resistance to a certain type of phages, in this case it is necessary the adoption of new species.
- 4) A well-defined lytic phage in vitro environment could show a different behaviour in vivo and adopt a lysogenic cycle in some circumstances.

1.4 Overview of our research

One of the most critical and tricky problem related to the use of bacteriophages in order to defeat bacterial pathogens is the fact that bacteria can also become resistant to phages. But fortunately, phage-resistance is not even remotely as troubling as antibiotic-resistance. Indeed, one hypothetic solution for this issue is the use of new types of phages. When bacteria become resistant to a specific kind of phages, they continue to be susceptible to other types and it is very easy to find them in the nature that is an endless source of phages. But if each newly isolated phage needs approval, this process could take a very long time and become too expensive. An alternative to the previous solution could be the attempt to exploit the phage evolution. In other words, like their bacterial hosts and unlike antibiotics, phages are able to mutate, to evolve themselves, and then they can fight phage-resistant bacteria (Matsuzaki et al., 2005). Furthermore, usually phages can attack bacteria by attaching receptors on the bacterial surface that are virulence factors, but the development of phage-resistance can bring to an alteration of these receptors and consequently make new mutants less pathogenic than susceptible bacteria so they can be defeated by the human immune system (Inal, 2003). In addition, it is also possible to prevent the evolution and growth of phage-resistant bacteria, using during the treatments a cocktail of phages (a preparation containing different types of phages) and/or in combination with antibiotics.

The research conducted by the team of Professor Alfonso Jaramillo Lab at the School of Life Sciences of the University of Warwick forms part of this scenario. The main aim of the work is the development of a methodical procedure in order to produce engineering synthetic bacteriophage cocktails against specific bacteria, forestalling possible phage-resistant mutants. This innovative idea is inspired to the principle of the system called PACE (phage-assisted continuous evolution) developed in the laboratory of Professor David Liu at the University of Harvard. Using the PACE platform (Figure 7), it is possible to evolve gene-encoded molecules that may be associated to protein production in *E. Coli*, continuously without the human intervention and 100 time faster than previous traditional methods (Carlson et al., 2014; Esvelt et al, 2011). To accomplish this, they have linked the desired activity to the production of infectious progeny phage, in particular exploiting the expression of protein III (or pIII that is encoded by gene III). According Esvelt et al., “phage infection requires protein III which mediates F pilus binding and host cell entry. Phage lacking pIII are approximately 10^8 -fold less infectious than wild-type phage”. In order to couple pIII production to the desired activity, gene III was deleted from the phage genome and inserted into an accessory plasmid (AP) present in *E. Coli* host cells. These cells continuously flow through a fixed-volume called “lagoon” containing a replicating population of phages called “selection phage” (SP) encoding the gene of interest. Any mutation of host cells has a minimal impact on the outcome of the selection over many rounds of phage replication because the lagoon dilution rate is so fast that only the evolving selection phage population can replicate. In this way, only the selection phage with an activity of interest can induce the production of a sufficient amount of pIII from the accessory plasmid and survive in the lagoon. In theory, it is possible to apply PACE system to any gene that can be linked to pIII production in *E. Coli*. There are many activity of interest at transcriptional, translational or post-translational levels, for instance the evolution of T7 RNAP. This polymerase is very specific only for its promoter but using the PACE system the T7 RNAP was evolved to recognize the T3 promoter and to reject the initial sequence target. A second plasmid called “Mutagenesis Plasmid” (MP) has been included in the cell hosts. This arabinose-inducible mutagenesis plasmid promotes the evolution elevating the

error rate during DNA replication in particular suppressing proofreading and enhancing error-prone lesion bypass.

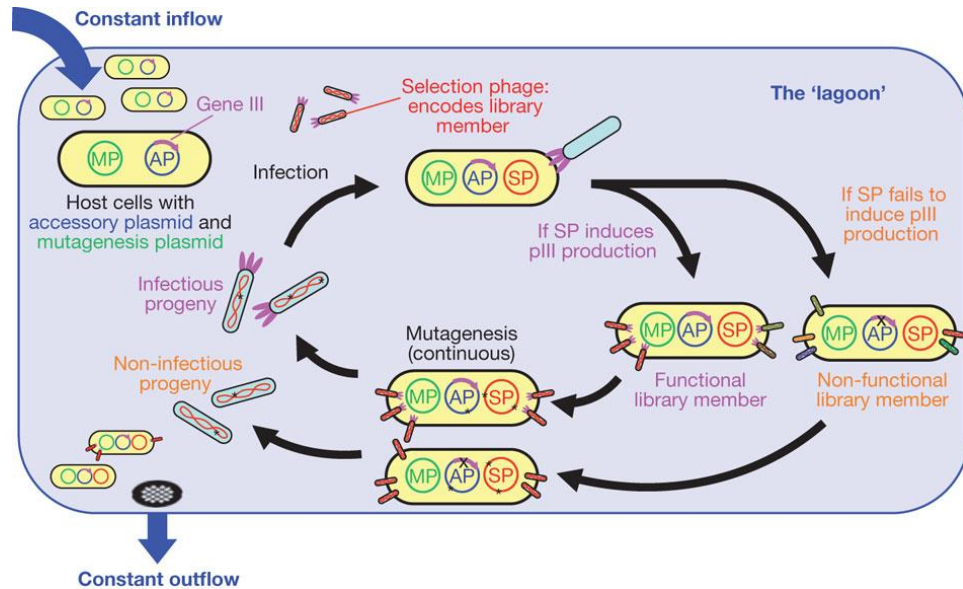


Figure 7: Overview of the PACE system.

(http://www.nature.com/nature/journal/v472/n7344/fig_tab/nature09929_F1.html)

Following a similar principle also the evolution of phages and bacteria can take place. Exploiting this idea, it is possible to achieve the goal of the research to get phage cocktails. Indeed, it is necessary to evolve bacteria and phages, one at time and repeat the procedure iteratively. The evolutions can occur using a system such as a bioreactor and typical tools of synthetic biology and bioengineering. In this work, a particular strain of E. Coli was adopted as phage host: MG1655 Δ fimA-H; Δ flu; Δ matB::P2_luxCDABE. This is an F minus (F-) strain that underwent the deletion of three specific genes in order to avoid the formation of biofilm. T7 bacteriophage was adopted as selection phage, in particular the T7 Δ gp5 phage that lacks of the gene 5 in its genome. The gene 5 protein (gp5) of bacteriophage T7 is a DNA polymerase that is fundamental for the phage replication and growth. The deleted gp5 was inserted in an accessory plasmid (AP) and included in the F- strain through electroporation process. Consequently, the T7 Δ gp5 phage can reproduce itself only attacking the F- that possesses the AP (Figure 8) with gp5.

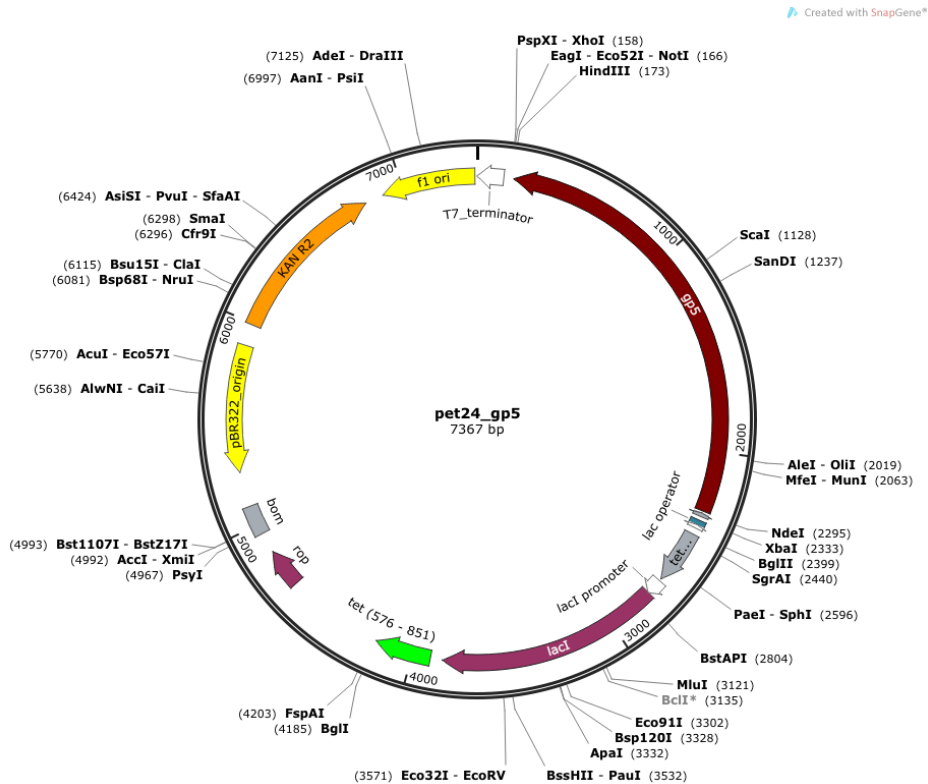


Figure 8: The accessory plasmid pet24_gp5 with Kanamycin resistance.
(Courtesy of J. Hassall, University of Warwick)

The host evolution takes place using the original F- strain (without AP) and including an arabinose-induced mutagenesis plasmid (MP) into it, which increases the probability of mutations. The cells are continuously pumped into the cellstat of a bioreactor in presence of a high concentration of phages (MOI>8, see below), in this way only the bacteria that develop phage-resistance can survive (Figure 1.9). New resistant bacteria will be used for the phage evolution using the reservoir method and mixing two types of host: one with AP and the other with MP (Figure 9). Cycling these two processes, it will be possible to get a set of different types of phages that could be used as cocktail.

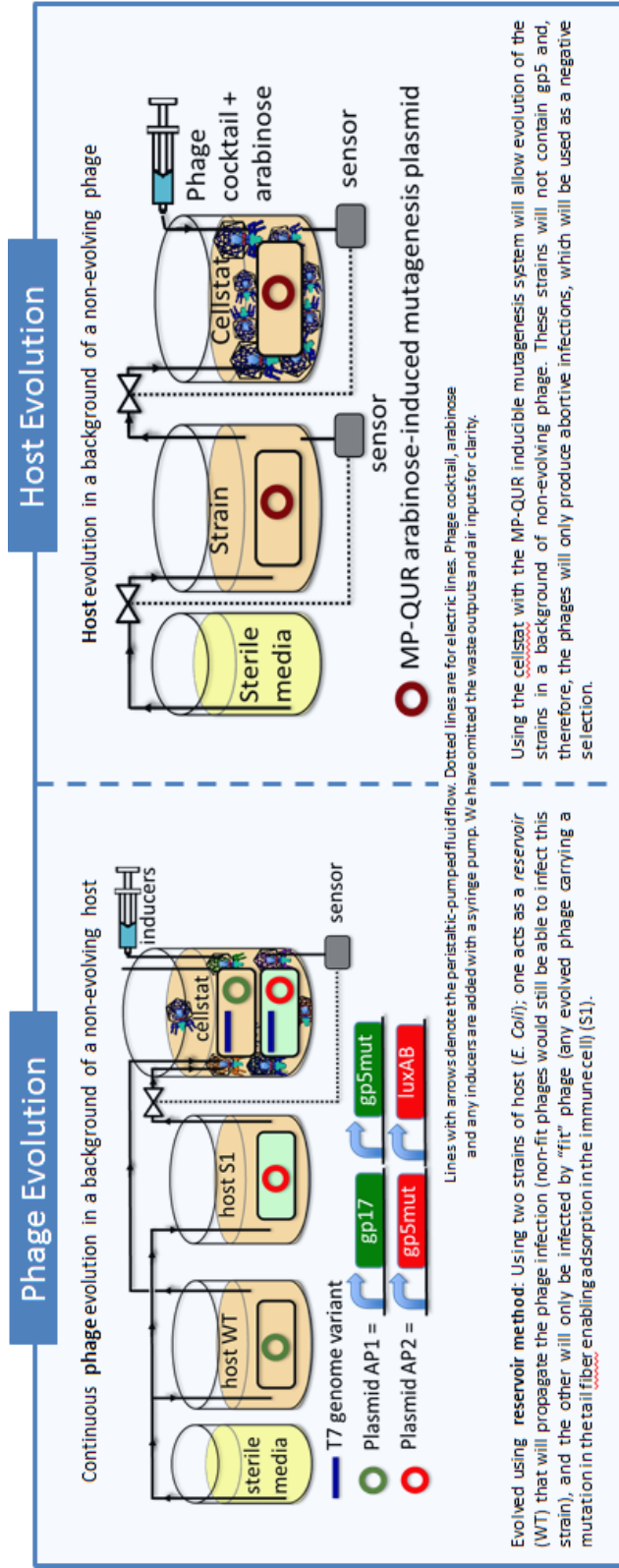


Figure 9: Phage and host evolution. (Hassall, J., Jaramillo, A. (2014). Engineering synthetic bacteriophage cocktails. Poster, University of Warwick)

Chapter 2:

2 Phage quantification and parameters inference

2.1 The initial idea of the project

The traditional method for the phage quantification, also known as phage titration, is the plaque assay. This technique is used for the detection of the concentration of viruses in a sample. Initially the procedure was developed for the titers of bacteriophage stocks, then after the modification realized by Renato Dulbecco in 1952, it was widely exploited for the quantification of many different viruses as well (Dulbecco-Vogt, 1953). It is considered a reliable technique easy to implement as its protocol is composed by few simple passages (see Chapter 3). The main disadvantage of the procedure is the time to wait for the results. Indeed, this method is based on the count of the number of plaques, circular zones produced by the spread of new infectious viral progeny from infected cells on a Petri dish and these can be visible only after a period of incubation that lasts usually about 24 hours or at least overnight. The goal of this work was the development of an alternative method for phage quantification, able to predict the unknown concentration of a sample as quickly as possible and avoiding culture techniques with Petri plates. The task was accomplished taking advantage of the software implementation of a model, that is representative of phage-host interactions. In the context of continuous evolution and production of synthetic bacteriophage cocktails, this choice was also motivated by the need to control, over the time, the concentration of free-phages as outcome of the cellstat. In addition to the phage quantification, the programs developed for this new method have also a second but no less important purpose: the inference of unknown model parameters using optimization criteria. These parameters reveal some important information concerning the characteristics of phages and bacteria.

2.2 Models for the bacteriophage-host interactions

The main purpose of the scientific research developed in the field of bacteriophage therapy is the possibility of considering the latter, in the next future, as a science, a concrete alternative to antibiotics in the treatment of pathogens. All this requires a wide knowledge of the interactions between bacteriophages and their hosts. Indeed, the dynamics that describe the behaviour of phages and cells play an important role in the result of the treatment and only the fully understanding of these intrinsic dynamical properties will be the key to the success. Unfortunately, it is easily comprehensible that the quantitative dynamics existing between prey (bacteria) and predator (bacteriophage) is too complex and elaborate only for human intuition, underlining the need to draw fully on the use of mathematical models. It is useful to develop models in order to reproduce conditions or theories in a simple way. There are many potential advantages and purpose in the use of mathematical models of dynamics, from providing a quantitative fitting to empirical data, to predicting future unknown outcomes and explaining complicated scientific material.

The interactions between bacteriophages and their hosts have been investigated for many years (Delbruck, 1940; Krueger & Northrop, 1930) and some interesting items about their features are still subject of scientific debates. Mathematical models, which have been proposed for a better understanding of these dynamical properties, cover many different area of interest, including ecological environment (Levin et al., 1977; Middleboe, 2000), food industry (Cattoën, 2003; Mudgal et al., 2006), therapeutic field (Cairns et al, 2009; Kasman et al., 2002; Levin & Bull, 1996; Payne & Jansen, 2000; Payne et al., 2000) and molecular evolution (Husimi, 1989). Some of them are mainly based on the classical Lotka-Volterra equations, known also as the predator-prey equations, used to describe the dynamical interaction of two general biological species. Only few models describing phage-host dynamics have also been validated or analysed mathematically (Mudgal et al., 2006). In this work, three different phage-host models have been considered and examined. The final best choice will be presented after a brief review of their main features and the

reasons of selection. It is also important to underline that these models are a description of *in vitro* bacteriophage-host interaction so they could not capture the real dynamics which occur *in vivo* environment where some parameters may be critical to treatment success. Therefore in the future it will be necessary a better understanding of the *in vivo* system in order to identify a relationship between *in vivo* and *in vitro* phage-host behaviour (Bull & Gill, 2014).

2.2.1 Model formulations

The following models have been chosen because they are easy to understand and their parameters are highly meaningful and purposeful. All state variables are time-dependent, but for the sake of simplicity this dependence will be omitted in the equations described.

a) In the paper “Understanding bacteriophage therapy as a density-dependent kinetic process” (Payne & Jansen, 2000), the authors present a simple and generic model which describes the variation over time of phage and host populations using three differential equations:

$$\frac{dx}{dt} = ax - bvx - H(t)x$$

$$\frac{dy}{dt} = ay + bvx - ky - H(t)y$$

$$\frac{dv}{dt} = kLy - bvx - mv - h(t)v$$

- $x(t)$ represents the concentration of uninfected bacteria
- $y(t)$ the lytic bacteria
- $v(t)$ the free phage
- a is the replication coefficient of the bacteria
- b is the transmission coefficient
- k is the lysis rate coefficient
- L is the burst size
- m is the decay rate of free phage
- $H(t)$ is the host response against the bacteria
- $h(t)$ is the host response against the phage

The model has been developed exploiting the kinetic properties of phage replication and it is able to show as outcome, a range of qualitative values for the variables. The work explains paradoxical and unpredictable results obtained with the use of bacteriophages in the therapeutic field splitting the outcome in different categories and considering some critical thresholds. The study illustrates the yielding points of bacteriophage biology which can be engineered, opening up the path for phage therapy.

b) The second model is described in the following paper: “Quantifying the Significance of Phage Attack on Starter Cultures: a Mechanistic Model for Population Dynamics of Phage and Their Hosts Isolated from Fermenting Sauerkraut” (Mudgal et al., 2006). Here the authors concern the possible use of a mechanistic mathematical model, in the food industry, in order to quantify the growth of phage and host population for different initial conditions. The model, validated with two different phage-host systems, is composed by 4 delay differential equations:

$$\frac{dS}{dt} = \alpha S \left(1 - \frac{S + I + M}{C} \right) - K(t)SP$$

$$\frac{dI}{dt} = K(t)SP - H(t - L)K(t - L)S(t - L)P(t - L)$$

$$\frac{dP}{dt} = -K(t)SP + BH(t - L)K(t - L)S(t - L)P(t - L)$$

$$\frac{dM}{dt} = \gamma M \left(1 - \frac{S + I + M}{C} \right)$$

- S(t) represents the density of susceptible bacteria
- I(t) the infected bacteria
- P(t) the free phage
- M(t) the resistant bacteria
- α is the growth rate of susceptible cells
- γ is the growth rate of resistant cells
- C the maximum cell density
- L is the latent period
- B is the burst size
- K is the adsorption rate coefficient

Considering phage-host kinetic properties, the model includes an important delay term ($t - L$) which represents phage maturation. Furthermore, the authors have introduced another change: a step function H for a better description of the adsorption rate that varies with time. The model results show a systematic difference between the parameters obtained from model optimization and those acquired from experiments and a possible dependence of some parameters on phage or cells densities.

c) The last model considered is described in the paper "Quantitative Models of In Vitro Bacteriophage–Host Dynamics and Their Application to Phage Therapy" (Cairns et al, 2009). The authors focus their attention on the non-linear kinetics of bacteriophage–host interactions combining experimental and model results. Their kinetic model sharing the assumption of Payne and Jansen (i.e. mass-action law), includes a delay term and a variable that represents resistant bacteria as in the second paper (Mudgal et al., 2006):

$$\frac{dS}{dt} = aS - fS - bSV$$

$$\frac{dR}{dt} = aR + fS$$

$$\frac{dI}{dt} = bSV - bS(t - K)V(t - K)$$

$$\frac{dV}{dt} = hbS(t - K)V(t - K) - bSV - mV$$

- $S(t)$ represents the concentration of susceptible bacteria (CFU)
- $I(t)$ the infected bacteria (CFU)
- $V(t)$ the free phage (PFU)
- $R(t)$ the resistant bacteria (CFU)
- a is the growth rate of susceptible and resistant bacteria
- f is the mutation rate of bacteria
- b is the binding rate of phages to susceptible bacteria
- K is the latent period (time between infection and lysis)
- h is the burst size at lysis
- m is the phage decay rate due to thermodynamic and other effects.

The presence of a delay term ($t - K$) is fundamental for the explanation of a phenomenon: after a period K , the burst of infected bacteria releases new phage progeny, in particular about h viruses per cell. In this way, at time t it is necessary to consider not only the current dynamics but also the concentration of free phage and susceptible bacteria at time $(t - k)$. The research represents one of the first phage therapy study tested using data acquired on an *in vitro* system and it can be seen as a precursor for the development of *in vivo* phage therapy.

2.2.2 The first data and the final model

The main purposes in the use of a mathematical model are the fitting of experimental data and the attempt to give them a meaning through the study of the parameter set. An initial fresh cell culture (MG1655) with an optical density of 0.14 was divided in different tubes and each one was infected by a different concentration of phages (T7). The samples were shaken and incubated at 37°C. The optical density (OD_{600}) was measured manually every three minutes taking 1mL of each sample and using the spectrophotometer. The preliminary experimental results of this project are shown below (Figure 10). The coloured curves represent the different phage-host samples. They show that in a first moment, the concentration of bacteria increases following the same trend of the negative control, bacteria without the presence of bacteriophages, (black line) but after a period, slightly different for each of them, the optical densities decrease until zero as consequence of cells death caused by phage infection. The curves obtained are similar to the data described by Krueger & Northrop in 1930 about “the kinetics of the bacterium-bacteriophage reaction”.

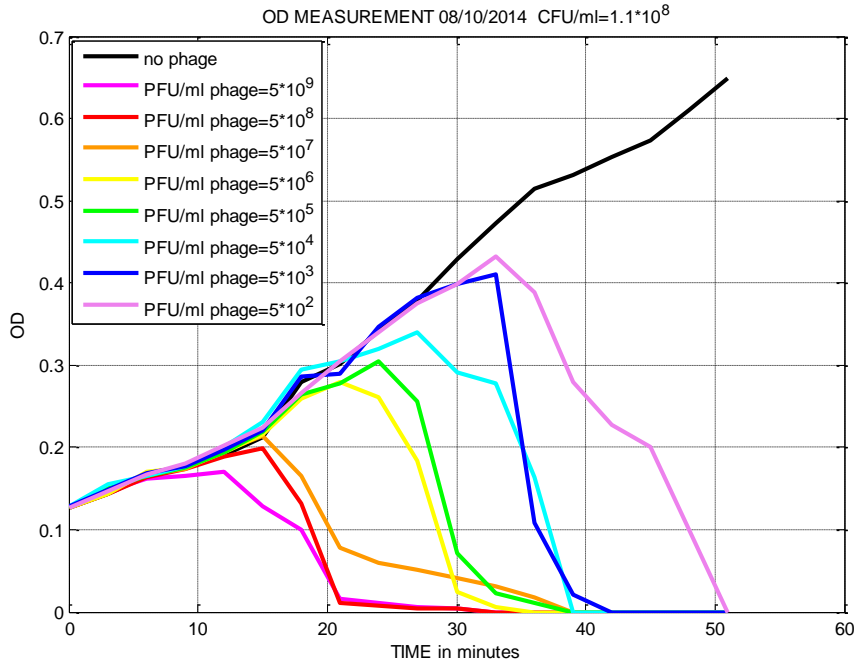


Figure 10: Optical density measurement over time of the concentration of bacteria infected by different dilution of phages.

The previous models chosen as possible candidates were implemented in MATLAB environment in order to try to use them for the fitting of these experimental data. The third model (c) has shown the best behaviour for our initial problem thanks to its flexibility, simplicity and stability. In addition, this mathematical model was simplified deleting one differential equation and one parameter. This has been possible because this project avoids the evolution of susceptible bacteria in resistant mutants (for details, see below). Below one will find the correct model adopted with $R(t)=0$ and $f=0$:

$$\frac{dS}{dt} = aS - bSV \quad (1)$$

$$\frac{dI}{dt} = bSV - bS(t - K)V(t - K) \quad (2)$$

$$\frac{dV}{dt} = hbS(t - K)V(t - K) - bSV - mV \quad (3)$$

There are three state variables S, I, V which represents the concentration of susceptible, infected bacteria and free phage particles respectively. Five parameters complete the description of the model:

- a is the growth rate of susceptible bacteria [min^{-1}]
- b is the binding rate of phages to susceptible bacteria [$\text{CFU}^{-1}\text{min}^{-1}$]
- K is the latent period (time between infection and lysis) [min]
- h is the burst size at lysis [PFU]
- m is the phage decay rate due to thermodynamic and other effects [min^{-1}].

Chapter 3

3 Materials and methods

In this chapter all the materials and methods adopted during the project will be illustrated. It is divided in two main sections: the first one includes a detailed description of the experimental procedures and protocols, while in the latter the programs implemented in MATLAB are presented.

3.1 Experimental methods

3.1.1 Enumeration of bacteria

The number of viable bacteria in a sample is commonly measured in colony-forming unit (CFU). A colony represents an aggregate of cells derived from a single progenitor cell and becomes visible upon a proper incubation time. Their count represents a rough estimate of only living cells which are able to replicate themselves via binary fission. This is in contrast with the direct microscopic counts that include also dead cells. The results can be usually reported in CFU/ml or CFU/g for liquid or solid cultures respectively. The determination of the number of bacteria in an unknown sample is obtained using serial dilutions, plating and counting living cells. This method was invented by Robert Koch and used for the study of water quality as described in the paper “About Detection Methods for Microorganisms in Water” in 1883. The first step is the serial dilution (Figure 11), it is necessary because usually the concentration of cells in a culture sample exceeds the accurate detection range of the assay. It is very important to keep in mind the dilution factor adopted in order to mathematically compensate for it at the end of the experiment. Another relevant point: there is a limit for the number of dilutions

that is possible to do. The quantification of bacteria is more accurate if the number of colonies counted falls in the range of 30-300 per Petri dish. Therefore, when the concentration of the starting culture is unknown, it is recommended to plate different dilutions and have triplicates in order to average the counts together.

The protocol:

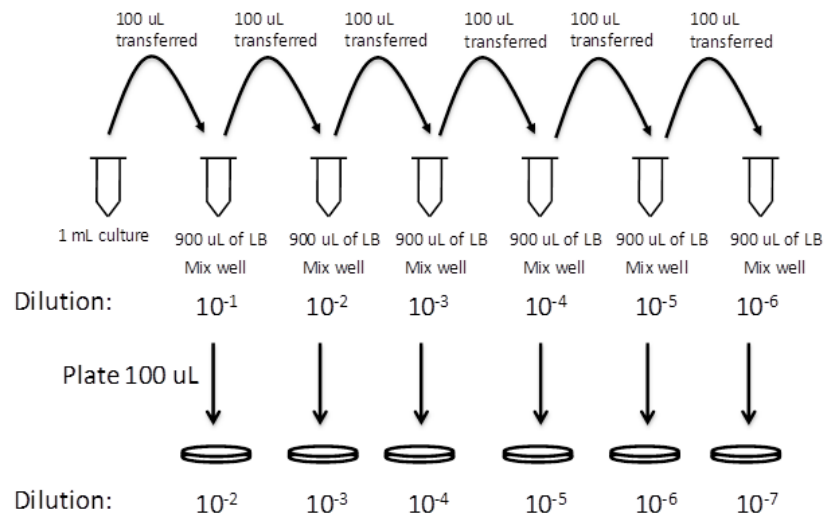


Figure 11: Serial dilution for the bacterial enumeration.
http://2014.igem.org/Team:CSU_Fort_Collins/Notebook/KillSwitch/Sep

After the preparation of six small, sterile test tubes labelled from 10^{-1} to 10^{-6} , 0.9 mL of Luria-Bertani (LB) broth were added to each one. 0.1 mL of the original liquid culture were inoculated into the first test tube and mixed thoroughly using the vortex before proceeding. The procedure was repeated for each test tube until the last one, withdrawing 0.1 mL of diluted bacterial suspension from every last tube and pipetting that into the next one. 0.1 mL of every serially diluted specimen that is significant for the counting (30-300 colonies so usually 10^{-4} , 10^{-5} , 10^{-6}), were plated onto different LBA (or LB agar + antibiotic where it was opportune) plates using spreaders or glass beads. Minimum three replicates for every dilution. After an overnight incubation at 37°C of the plates, upside down, the number of colony-forming units (CFU) was counted and averaged for each dilution. Finally the concentration of bacteria in the original suspension (in CFU/mL) was calculated using the following formula:

$$\frac{\text{CFU}}{\text{mL}} = \frac{\text{Number of colonies counted in the plate}}{\text{Volume plated (in mL)} * \text{Total dilution factor}} \quad (4)$$

For example: if the number of colonies counted in a plate with 0.1 mL of the sample with dilution 10^{-6} is 60, the CFU/ml is:

$$\frac{\text{CFU}}{\text{ml}} = \frac{60}{0.1 * 10^{-6}} = 6 * 10^8$$

3.1.2 Enumeration of bacteriophage particles

The traditional procedure for the measurement of the concentration of viruses in a sample is the Plaque Assay. The basic concept is similar to the method previously illustrated for the determination of CFU/ml. The final results represent only the number of virus particles able to form plaques per unit volume and not the total quantity of particles. The benchmark is in PFU/mL (plaque-forming unit/millilitre). A viral plaque is a circular region generated within a solid cell culture by the replication and the spreading of bacteriophage viruses and consequently the cell bursts. Usually, the final count involves only the plates that contain a number of plaques between 10 and 100 in order to minimize the error.

The protocol:

To perform a plaque assay, 10-fold dilutions of a virus stock were prepared. 0.1 mL aliquots were inoculated into a tube containing 3mL of soft agar and 0.3mL of fresh bacteria at an OD value of 0.2 (or approximately with a CFU of 10^7). The mix was plated, in triplicates, and incubated at 37°C overnight. After the count of the number of plaques, the PFU/mL was calculated using the following formula:

$$\frac{\text{PFU}}{\text{mL}} = \frac{\text{Number of Plaques counted in the plate}}{\text{Volume plated (in mL)} * \text{Total dilution factor}} \quad (5)$$

3.1.3 Multiplicity of infection

The multiplicity of infection, or simply MOI, is an important parameter to take into consideration in virology since it was introduced 70 years ago (Ellis et al., 1939). It is the ratio of infectious virions to viable cells in a sample. In other words, the MOI is the average number of bacteriophages per bacterium:

$$MOI = \frac{(PFU/mL) * mL \text{ of phages added}}{(CFU/mL) * mL \text{ of cells added}} \quad (6)$$

This number could be 0.001, 0.01, 0.1, 1, 10, 100, etc., and its value can be decided at the beginning of an experiment or deduced in a second moment, if the starting concentration of phages or bacteria is unknown, using some prediction tools. An essential key point is the understanding of the concept of the multiplicity of infection. For instance, an MOI of one (e.g. 1000 phages added to 1000 cells) does not mean that each cell is infected by one virion. In fact, the number of phages that infects each bacterium at different MOI can be described by the Poisson distribution:

$$P(n) = \frac{m^n * e^{-m}}{n!} \quad (7)$$

where m is the MOI so the average number of viral agents per target, $P(n)$ is the probability to have bacteria infected by exactly n phages. In particular:

- $P(0) = e^{-m}$ is the probability to get uninfected cells
- $P(1) = m * e^{-m}$ is the probability to get cells infected by one phage
- $P(n > 1) = 1 - [P(0) + P(1)] = 1 - [e^{-m} * (m + 1)]$ is the probability to get cells multiply infected
- $P(n > 0) = 1 - P(0) = 1 - e^{-m}$ is the probability to get infected cells.

There is an assumption which underpins this statistical description: an equal probability for all cells of a culture to be infected.

Therefore, for a better understanding of the MOI's interpretation, another important concept has to be introduced. When considering two samples with the same number of phages and bacteria, then the same MOI, but in different volume of medium (e.g. 5mL and 10mL), even if the final number of infected cells in both cultures would be the same, the duration of the adsorption period will show different (much longer in the bigger volume) (Racaniello, 2014). The MOI is easy to calculate and simple to understand but it could not be sufficient to describe all the experimental conditions on its own (Shabram & Aguilar-Cordova, 2000). Furthermore, it has been demonstrated that the use of MOI cannot be appropriate in some conditions and an alternative parameter has been proposed (Kasman et al, 2002).

In this work, considering what described above, the MOI was used in order to keep a record of the concentration of phages and cells in the samples and only as a qualitative descriptor of the experimental situations.

3.1.4 Phage-host interaction experiment

The first experiments were conducted using a spectrophotometer. In particular, a fresh cell culture with an optical density of 0.1- 0.2 was equally divided in different tubes, and a different dilution of phages was inoculated in each tube. The specimens were incubated at 37°C with shaking. Every 5 minutes, 1mL of each sample was withdrawn and its optical density was measured manually. This procedure has several disadvantages. Firstly, it is not reliable and accurate because of the many delays and imprecisions: no optimal growth condition for the culture (they were taken away from the incubator every 5 minutes); the need for the operator to fill different cuvettes and measure them one at a time; interoperator variability. Secondly, there is a waste in terms of time, resources and material. Lastly, there is an operative limit for the maximum number of samples measurable every 5 minutes. In order to overcome these issues, a new automatic procedure was adopted thanks to the use of a microplate reader (Figure 12). This instrument is able of

measuring automatically the absorbance over time of several samples, controlling the internal temperature and shaking in different ways. The cells (MG1655 Δ fimA-H; Δ flu; Δ matB::P2_luxCDABE, with Kanamycin resistance) and the phages (T7 Δ Gp5) used during these experiments have been previously engineered in order to avoid any contamination problem. In other words, this type of bacteriophage is able to replicate only infecting the cells which possess the Gp5- plasmid.



Figure 12: Infinite® F500 Tecan microplate reader

Protocol:

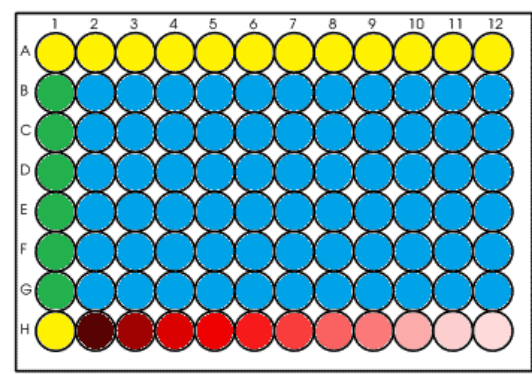
The growth of fresh culture was obtained inoculating cells of an overnight culture into a tube with the antibiotic Kanamycin and LB medium considering the following proportions:

$$\frac{\text{Kanamycin}}{\text{LB}} = \frac{1}{1000}$$

$$\frac{\text{Bacteria}}{\text{LB}} = \frac{2}{100}$$

the cells were incubated at 37°C and shaken until the achievement of an OD₆₀₀ of 0.2 or better a CFU/ml of about 6*10⁷ (see Chapter 4). This initial condition value was settled considering two key points. Firstly, in 2002, Kasman and his colleagues demonstrated

that the common definition of MOI is inadequate for experimental conditions in which the cell concentration is minor than $10^7/\text{mL}$, and as a solution to this issue, they proposed an alternative parameter called $\text{MOI}_{\text{actual}}$ that considers the cell concentration, as well as the adsorption time. In order to avoid this problem and simplify the picture, the minimum concentration of starting cells taken into account is $10^7/\text{mL}$. Secondly, the purpose of this work is parameter inference and prediction of an unknown bacteriophage dilution as soon as possible. The time required to kill all the phage hosts, is shorter for a low concentration of cells than for a high one, considering a same amount of bacteriophages inoculated ($\text{MOI}_{\text{highCONCENTRATION}} < \text{MOI}_{\text{lowCONCENTRATION}}$). Consequently, this represents an upper boundary so that the starting concentration of bacteria is approximately $10^7/\text{mL}$. It is very important to take a record of the starting optical density value because a same amount of fresh cells will be used for the prediction of unknown concentrations of phages. A sample that comes from the original bacteriophage stock was serially diluted from 10^0 to 10^{-10} . The standard disposition of bacteria and phages is shown in the figure:



- **Blank**
- **Negative control (no bacteriophages)**
- **Bacteria+Bacteriophages**
- **Bacteriophages serially diluted:**
- 10^0
- 10^{-1}
- ...
- 10^{-10}

Figure 13: Standard disposition in 96-well plate.

A 96-well plate was used in the experiment for the measurement of optical density (Figure 13). Firstly, 0.2 mL of LB medium were dispensed in each yellow well; secondly the same amount of each different dilution of phages, from 10^0 to 10^{-10} (dilution factor), were inoculated into the respective wells; lastly 0.18 mL of cell suspension was distributed in every green and light blue well. After that, using a 12 well multichannel pipette and withdrawing from the last line (H) of the plate, a total of 0.02 mL of T7 Δ Gp5 phages with different dilution was added to the cell suspension in the light blue wells (0.02 mL of LB into the green wells) (Figure 14).

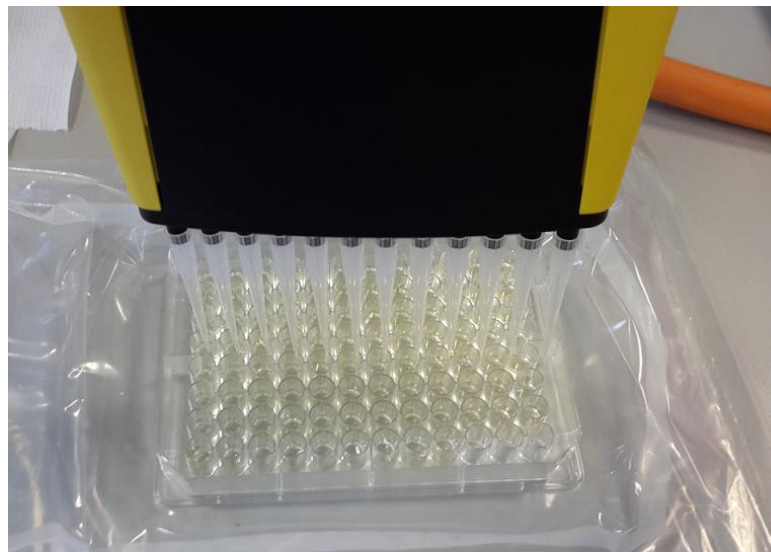


Figure 14: The use of a multichannel pipette for the dispensation of phages and medium in the cells.

The plate was closed with the lid, previously maintained at 37°C, in order to decrease the formation of condensation during the experiment, and inserted in the pre-warmed microplate reader. The program was settled with a kinetic cycle that consisted in: the control of the temperature at 37°C; the shaking of the plate for three minutes each cycle; the OD₆₀₀ measurement of every well except the last line sacrificed for the phage dispensation. In particular the optical density values were measured in four fixed point of every well, three times each, in order to get a better accuracy, so the final OD₆₀₀ is the average of these values. The protocol adopted is a modified procedure for the *Kinetic Lysis Assays* described by Qimron and colleagues in the paper “Genomewide screens for *Escherichia coli* genes affecting growth of T7 bacteriophage” in 2006.

3.1.5 Optical density - CFU/mL calibration

The OD-CFU/mL calibration can be seen as a link between the experimental part and the software environment. Indeed, all the collected data, which represent the evolution over time of the number of cells, are OD values. In order to have a meaningful model parameter inference and phage quantification, it is necessary to translate these values in CFU/mL. The relationship between OD and CFU/mL can be considered linear.

Protocol:

A fresh cells culture was incubated at 37°C and shaken. 4 or 5 sample were collected from it during the growth and considering an OD₆₀₀ range of 0.1-0.8 (e.g. 0.1 0.2 0.4 0.6 0.8). Each sample was immediately serially diluted and the appropriate dilutions (e.g. 10⁻⁴, 10⁻⁵, 10⁻⁶) were plated in triplicates and incubated at 37°C overnight. After the count of the number of colonies, three CFU/mL values were determined for every OD₆₀₀. The linear fitting equation of this relationship was obtained with the use of MATLAB. The general equation is:

$$\frac{CFU}{ml} = p1 * OD_{600} + p2 \quad (8)$$

where p1 and p2 are parameters of the straight line and they were determined by the best fitting, obtained by the least square method. All the optical density values were measured using the spectrophotometer. For experimental results derived from the TECAN microplate reader, a conversion of OD_{tecan} in OD_{spectrophotometer} values has been necessary. In particular, a fresh cell culture was incubated at 37°C and shaken, at regular intervals. A sample was withdrawn and its optical density was measured using both the spectrophotometer and the microplate reader.

3.1.6 Definition of delay model parameters

a) Growth Rate

As described in Chapter 1, the replication process of bacterial populations is called binary fission. Bacterial growth is represented by the variation of the cell number over time and it follows an exponential trend called also doubling. Indeed, during the replication, a bacterium can divide into two daughter cells, that are genetically identical to the mother cell if no mutation occurs. Thus, each bacterial generation is, theoretically, twice the number of the previous population but it is important to highlight that not necessarily all cells survive. Considering the first equation of the model adopted (1), after a simplification, it represents the first-order chemical reaction that occurs during the Log-phase of a bacteria culture growth, the only one useful for the growth rate determination (Hall et al., 2013).

$$\frac{dS}{dt} = aS \quad (1.a) \quad \text{with } V = 0 \text{ (no bacteriophages)}$$

The equation describes the increment of cell population in time (dS/dt) proportional to the number of bacteria present at that time and the constant of proportionality a is called growth rate. Its value, expressed in h^{-1} or, as in this work, in min^{-1} , can be easily determined solving the differential equation:

$$\int_{S_{t_0}}^{S_t} \frac{dS}{S} = \int_{t_0}^t a dt \quad (9)$$



$$\ln S_t - \ln S_{t_0} = a * (t - t_0) \quad (10)$$

Thus, the difference between the natural logarithm of the number of susceptible cells at time t and the natural logarithm of the number of susceptible cells at time t_0 is directly proportional to the time interval through the constant growth rate a .

For a simpler interpretation, the equation can be converted in \log_{10} :

$$\log_{10}S_t - \log_{10}S_{t_0} = \frac{a}{2.303} * (t - t_0) \quad (11)$$

Plotting different concentrations of cells (CFUs) present at various times and using a \log_{10} scale for the y axis, the result is a straight line:

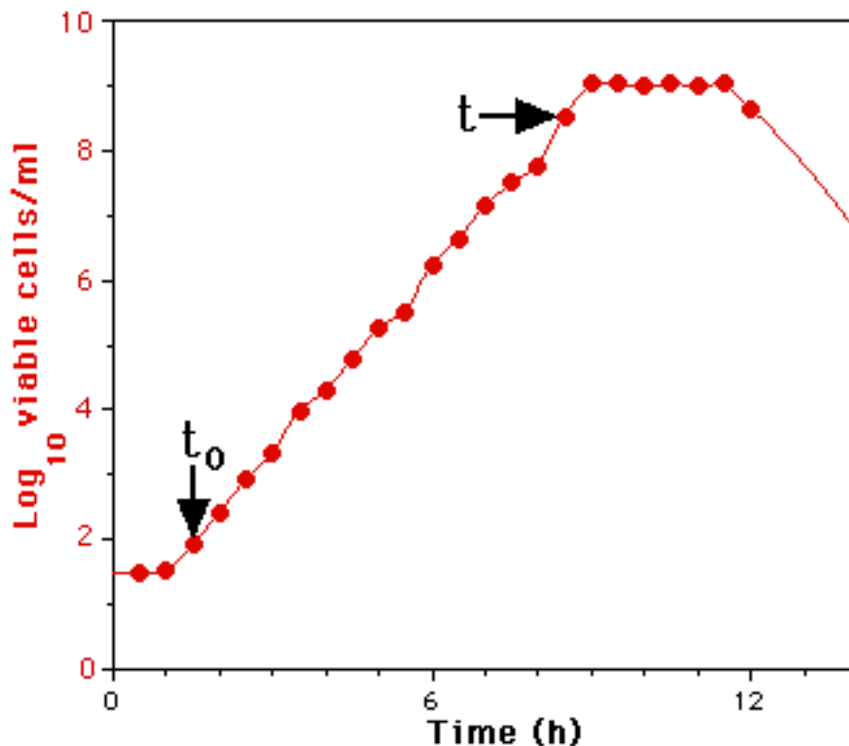


Figure 15: \log_{10} of the cell number over time.

<http://colinmayfield.com/biology447/modules/intro/bacterialgrowthrate.htm>

The growth rate value represents the slope of the line, namely how quickly a particular type of bacterium grows in a particular media. It can be calculated making a explicit from the previous equation:

$$a = \frac{(\log_{10}S_t - \log_{10}S_{t_0}) * 2.303}{(t - t_0)} \quad (12)$$

There is a relationship between the growth rate and the doubling time or the mean generation time. In particular, the latter is the

average time interval required for a population to double and divide by binary fission. Considering the equation (5):

$$\left(\begin{array}{l} \text{if } S_t = 2 * S_{t_0} \\ \text{the cell number at time } t \text{ is} \\ \text{the double of the cell number at time } t_0 \end{array} \right)$$

$$\text{then } (t - t_0) = d \text{ or doubling time}$$



$$\ln 2 = a * d = 0.693 \quad (13)$$

$$d = \frac{0.693}{a} \quad (14)$$

Protocol:

Before the advent of microplate readers, the growth rate was measured manually. The cultures were grown in a temperature controlled environments such as incubators, warm baths, etc. and opportunely shaken. At regular intervals of a few minutes, the optical density of the samples was measured using the spectrophotometer and plotted considering the time in the x axis (Hall et al., 2013). With the development of the microtiter plate readers the previous procedure can be automatically performed by the machine. In this project, considering the phage-host interaction experiment previously described (see Paragraph 3.1.4), the useful data for the growth rate determination are the mean results of the negative controls (only the green wells in Figure 13). It is important to underline that the curves used represent only the exponential growth of bacteria avoiding the first lag phase. After the conversion of the optical density values of these data in CFUs, the parameter was automatically calculated by the MATLAB software system implemented, using the equation (7).

b) Phage Decay Rate

The bacteriophage decay rate m is a constant value which describes the number of phage particles lost because of random degradation. That can occur due to thermodynamic or other effects (Cairns et al., 2009). The decay rate is usually expressed in h^{-1} or min^{-1} . In order to determine the parameter m , a 7-days experiment was conducted in absence of a host.

Protocol:

2mL of T7ΔGp5 bacteriophages (PFU/mL= $3 \cdot 10^{10}$) was added to 18mL of LB rich medium in a 50mL Falcon Tube and hermetically closed with Parafilm. They were incubated at 42°C with shaking at 100rpm for 7 days. At regular intervals, samples were withdrawn and serially diluted and inoculated into fresh cell culture (CFU= about 10^7) and soft agar for the determination of PFU/mL. Each sample was plated in triplicates. The final results were plotted in MATLAB environment and fitted considering the solution of a simplified equation (3):

$$\frac{dV}{dt} = -mV \quad (3. a) \quad \text{with } S = 0 \text{ (no bacteria)}$$

The solution of this equation is:

$$\int_{V_0}^{V_t} \frac{dV}{V} = - \int_0^t m dt \quad (15)$$



$$\ln V_t = \ln V_0 - m * t \quad (16)$$

The equation (11) represents a straight line and the value of m was obtained as its slope.

c) Latent Period

The latent period or incubation period K , is the time interval elapsed from the attachment of a phage to the cell wall and the viral genome injection until the release of the first new progeny upon the burst of bacteria. This parameter is a measure of the duration of the lytic phage infection, its replication and assembly, and it is expressed in h or min (Adams, 1959).

Protocol:

The latent period was determined using a one-step growth experiment. A fresh cell culture ($\text{CFU/mL}=6 \times 10^7$) was added in a 100mL conical flask and mixed with bacteriophages considering a $\text{MOI}=0.01$ (Cairns et al., 2009; Hyman & Abedon, 2009). The entire system is controlled in temperature (at 37°C) and shaken using a stir plate (Figure 16a). Samples from the culture were withdrawn at periodic intervals and immediately double filtered in order to get free phage removing bacteria and bound phages (Figure 16b). All the specimens were serially diluted and the aliquots were plated with fresh bacteria, in triplicates. The determination of this parameter value is simple to understand: the number of counted plaques, roughly constant at the beginning, increases progressively after the latent period due to the lysis of a bacterial population (Delbruck, 1942).

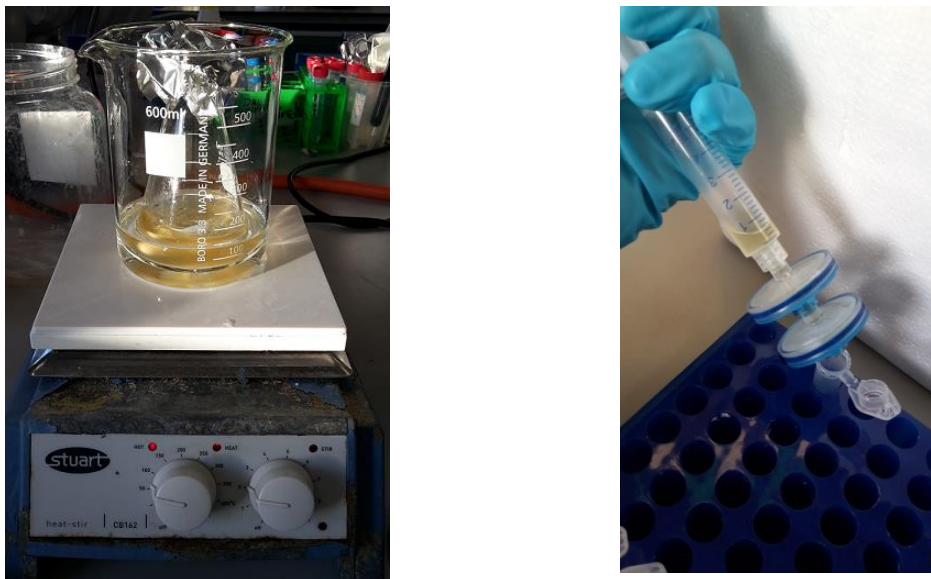


Figure 16: a-Latent period experiment on the stir plate. b- Double filtering.

d) Burst Size

In the lytic infections, the burst size is the number of new phage particles released per infected bacterium. This concept is strictly linked to the burst event, in other words, the lysis of the cell. Usually the burst size is determined as population averages (called also average burst size) (Adams, 1959). In 1980, Gadagkar and Gopinathan described in their paper “Bacteriophage burst size during multiple infections”, a simple formula for the determination of the burst size:

$$\mathbf{Burst\ size} = \frac{\mathbf{E}}{[\mathbf{B} - (\mathbf{C} - \mathbf{D})]} \quad (17)$$

Considering (*A*) as the starting number of bacteria (CFUs) and (*B*) the total number of phages added, the *MOI* is (*A*)/(*B*). At the end of the adsorption period, (*C*) represents the number of infected bacteria plus the number of free phages remaining, and (*D*) the number of infected bacteria. Thus, (*C-D*) is the number of unadsorbed free phage. After the latent period, (*E*) is the number of phages released by bacteria lysis.

A similar formula is described in the paper “Lysis Timing and Bacteriophage Fitness” (Wang, 2006):

$$\mathbf{Burst\ size} = \frac{\mathbf{F}}{(\mathbf{T} - \mathbf{U})} \quad (18)$$

Where *T* is the starting concentration of total phage: unadsorbed phages (free-floating phages) plus infective cells (bacterial cells infected with phages); *U* is the number of unadsorbed phage; *F* is the final concentration of phages after the latent period.

Protocol:

For the determination of the burst size, the previous formulas were modified and adapted to this work. The number of plaques was counted from the same experiment of the latent period evaluation.

e) Binding Rate

The binding rate, or known as adsorption rate, is a measure of the combination of bacteriophages with susceptible bacteria, which is the first step of phage growth and reproduction. Typically this constant considers the portion of irreversible adsorption to cells rather than the reversible one. The rate is proportional to the number of phages and bacteria (Delbruck, 1940) but it also depends on the affinity between them, the bacterial target size, and other factors. This rate of phage-host attachment is first order with respect to both the concentration of free phage and the concentration of bacteria (Krueger, 1930). It can be calculated solving the following equation (Ellis & Delbruck, 1938):

$$\frac{dV}{dt} = -bSV \quad (19)$$

This relationship can be easily obtained from the delay model equation (3) considering $h=0$ and $m=0$. The experiment for the determination of the binding rate value consists of the measurement of the free phage decline from time zero, when bacteriophages are inoculated into a bacterial culture, to a time immediately before the start of bacterial lysis represented by the latent period. Thus, the duration of the experiment is minor than the latent period. For these reason the previous assumptions can be taken into consideration: no lysis so the burst size $h=0$; the experiment lasts less than one hour so the decay rate $m=0$. The solution of the equation (14) is:

$$V = V_0 * e^{-b*S_0*t} \quad (20)$$

Where V and V_0 are ending and starting number of phages respectively, b is the binding rate constant, t is the established time over which the phage adsorption takes place, (Hyman & Abedon, 2009) and S_0 is the concentration of bacteria at time zero. In this work, the number of bacteria and bacteriophages are expressed in CFU and PFU, the time in minutes, so the binding rate is measured in $\text{CFU}^{-1}\text{min}^{-1}$. Considering another way to write the equation (20):

$$-\ln\left(\frac{V}{V_0}\right) = S_0 * b * t \quad (21)$$

The latter is similar to the equation of a straight line ($Y=m*x$), where the slope is S_0*b .

Protocol:

The number of plaques, which represents the decline of free phage over time, was obtained from the same binding experiment of the latent period and burst size calculation, considering the results from time zero to lysis time (excluded). The data were plotted in MATLAB environment and b was obtained by the fitting using the equation (16).

3.2 Software implementation

All data processing was performed off-line using a commercial software package (MATLAB 8.1.0.604, The MathWorks Inc., Natick, MA, 2000). The software system implemented is composed by five “user-friendly” main programs which are based on different secondary functions. In this section, their principal features will be briefly described. Furthermore, all the information reported and inserted by the user will be specified by the abbreviation UI (User’s Information).

3.2.1 Plot and summary of the OD measurement results

The main purpose of the MATLAB program *AImainPLOT.m* is the plot and the summary of the results obtained from the microplate reader for the phage-host interaction experiment. The final data, recorded usually in a Microsoft Excel file, can be elaborated and managed considering the standard protocol adopted for the optical density measurement (Figure 17). The first line (white wells) represents the blank, usually the medium adopted (e.g. LB); the last line (black wells) is not taken into consideration because it was “sacrificed” for the phage serial dilution; the first column (green

wells) is the negative control (no bacteriophages, only bacteria); the remaining coloured columns are different replicates of the same sample (0.18mL cells + 0.02mL phages) but each one (or each colour) with a different dilution of phages (e.g. the red column represents 6 replicates of cells with undiluted phages, the orange one with 10^{-1} diluted phages, etc.). It is not necessary to use all the wells (e.g. also only three replicates and phage dilutions from 10^{-2} to 10^{-6}) and when the blank value is known, it is possible to use the first line for a further replicate. Nevertheless, it is mandatory to follow this disposition, filling replicates from up to down and dilutions from left to right in descending order.

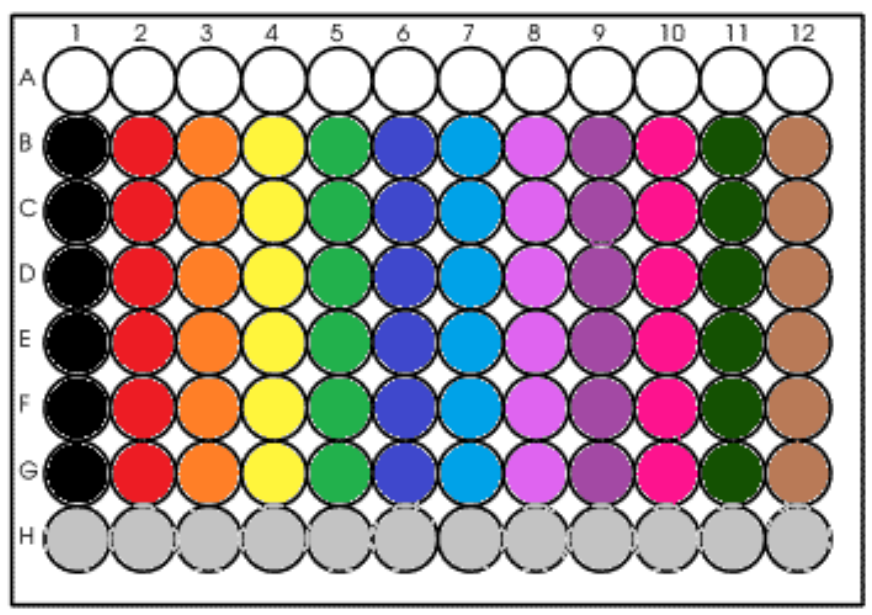


Figure 17: Standard protocol for the OD₆₀₀ measurement in a microplate reader.

AImainPLOT.m description:

The first step is the choice of the excel file, the user can simply select it from the right folder. The program considers the blank value and shows the plots of the different dilutions of phages, one figure for each replicate (UI). After the selection of the only useful data (UI) and the time vector to take into consideration, the script is able to order automatically the curves with the same dilution of phages (also the negative control) and calculate their mean. In order to describe the dispersion of the data from the mean curves, the program, using the file *sigmFITdata.m*, can fit the descending part

of the curves with a sigmoidal function and calculate the coordinates of the midpoint. The dispersion of the midpoints of different replicates, for every dilution, is displayed through the function *herrorbar.m* on the plot of the mean curves. The general Sigmoidal, or better logistic, equation used by the *sigmFITdata.m* is (in blue):

$$Y(t) = K + A - \left(A + \frac{K - A}{1 + \left(e^{-\frac{t-M}{S}} \right)} \right) \quad (22)$$

with $A = 0$

Where the four parameters are A , K , M and $1/S$. These are the lower asymptote, the upper asymptote, the time of the maximum growth and growth rate respectively. The lower asymptote equals zero because the optical density data to fit are always non negative, thus, a negative value for A would be meaningless. Furthermore, the equation represents an “inverted” S-shape curve, it was formulated considering the generalised logistic function (in red, Figure 18).

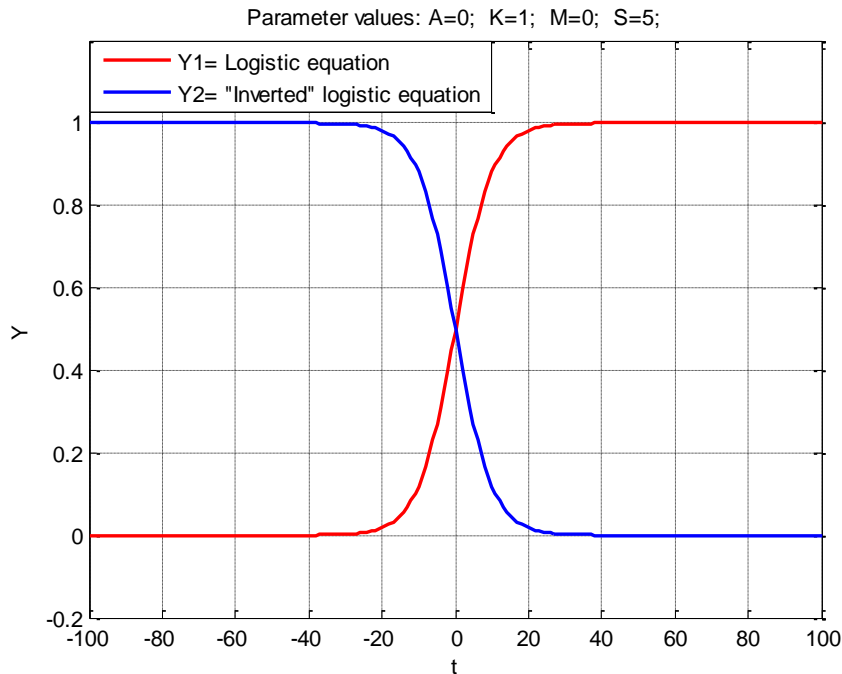


Figure 18: Example of a logistic function (red) and an “inverted” logistic function (blue).

Analysing the functions:

$$\begin{array}{ll} \text{when} & t \rightarrow \text{Inf} \quad \text{then} \quad Y1 \rightarrow K; \quad Y2 \rightarrow 0 \\ & t \rightarrow -\text{Inf} \quad \quad \quad Y1 \rightarrow 0; \quad Y2 \rightarrow K \end{array}$$

The midpoints of the data fitting curves are automatically obtained thanks to the implementations of a second function “helper” called *sigmoide.m*.

3.2.2 Plot and summary of the OD measurement results

The second program implemented in MATLAB environment is called *A2odCFUcalibration.m*. Considering the results of the OD-CFU/mL calibration experiment, the program is able to return the best fit or relationship between these two variables. The conversion is of fundamental importance for the phage-host delay model taken into account, where the number of susceptible and infected bacteria is expressed in CFU and the concentration of phages in PFU.

A2odCFUcalibration.m description:

The first step is the data import: the user can choose the excel file with the final results come from the OD-CFU/mL calibration (Figure 19).

	A	B	C	D	E	F	G
1	OD-CFU/mL DATA						
2	X	0	0	0	0.15	0.15	0.15
3	Y	0	0	0	5.20E+07	6.50E+07	5.00E+07
4							
5							

Figure 19: A template for the OD-CFU/mL calibration results. X=OD; Y=CFU/mL.

The best fitting of the data can be obtained automatically or manually (UI). In the first case, the default MATLAB function used is *fit.m*, and the equation adopted for the fitting is a linear polynomial curve with the robust regression method called LAR (Least Absolute Residuals). The normal linear least-squares fitting has the disadvantage to be sensitive to outliers (extreme values).

They have a large influence on the fit because squaring the residuals magnifies the effects of these extreme data points. The LAR method finds a curve that minimizes the absolute difference of the residuals, rather than the squared differences. Therefore, extreme values have a lesser influence on the fit. With the second option, the user has the possibility to choose the best function and options for the fitting using the Matlab toolbox *cftool* and save the results in the workspace (Figure 20a). Nevertheless, a first-order fitting function is recommended. The last passage of the program consists of the CFU/ml determination for the initial amount of cells, in other words, the concentration of bacteria at the beginning of the microplate reader experiment. The user can manually select its value if it is known, otherwise inserting the starting OD value, the program can automatically calculate the CFU/mL with the use of the fitting relationship or graphically through the *ginput.m* command (UI) (Figure 20b).

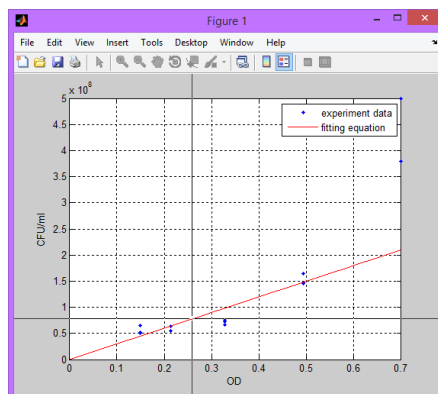
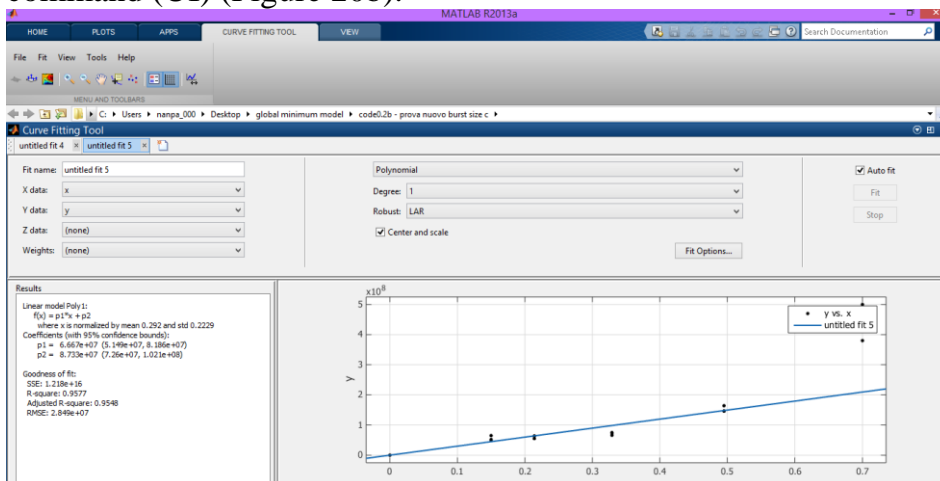


Figure 20: a-Matlab Curve Fitting application. b-Matlab Ginput command.

3.2.3 First fitting attempt and parameter research

The program *A3mainFIT.m* represents the third passage of the software system implemented. It depends on the information about the mean curves of the phage-host interaction and the calibration of optical density and CFU/mL, collected from *A1mainPLOT.m* and *A2odCFUcalibration.m* respectively. Its main aim is the research of the optimal values for the five model parameters, which give the best fitting with the data. This optimization research is accomplished for each sample (or mean curve), obtaining in this way, a series of parameter sets equal to the number of curves taken into account. The procedure leads to a situation called “overfitting”, but the results are only partial: they represent the starting point for the research and optimization of a final parameter set values, described in the next section. The main reason why it is preferable, firstly, optimize different parameter sets for every single sample rather than one set for all the samples, is the saving in terms of time and computational complexity. Indeed, the research of parameter values requires the minimization of an error function, and the solution of the differential delay equations, which represents the best fitting between data and model, might be only a local minimum. In order to overcome this issue, a global minimum research is adopted, considering different initial value for the parameter set. The idea of this software system is to conduct the global research for each sample, one at time, so as to limit the range of the possible values, and at a later stage, to obtain the ultimate parameters.

A3mainFIT.m description:

After the loading of the data coming from the execution of the previous programs, the user can insert the PFU/mL values for the current bacteriophages, considering this value for the highest concentration adopted during the experiment (e.g. if the first dilution is 10^{-1} then the PFU/mL is $3 \cdot 10^9$; if the first dilution is 10^{-2} then the PFU/mL is $3 \cdot 10^8$). The second step is the growth rate calculation. This value can be obtained using the formula previously illustrated (Equation 12) and applying it to the negative

control (no bacteriophages) mean curve. In particular the two points necessary for the calculation can be detected automatically as the initial and the end points of the experiment, or manually with the use of *ginput.m*. After the automatic definition of the initial conditions for bacteria and bacteriophages, the global minimum research takes place in a structured “for” cycle. It requires the use of different user-defined and standard MATLAB functions. Firstly, the model delay differential equations are implemented in the function *cairns.m*, this is recalled by an helper function *helper.m* which is able to solve the system using the built-in MATLAB solver *dde23.m*, specialized in the resolution of delay differential equations with constant delays. Secondly, the function *lsqcurvefit.m* allows the research of the best fitting between data and model. It is able to solve the nonlinear curve-fitting problems in least-squares sense. In other words, considering *xdata* as the starting values (the initial guess) of the parameter set, and *ydata* as the observed data, it finds the coefficients x that solve the problem:

$$\begin{aligned} & \min_x \|F(x, xdata) - ydata\|_2^2 \\ & = \min_x \sum_i (F(x, xdata_i) - ydata_i)^2 \end{aligned} \quad (23)$$

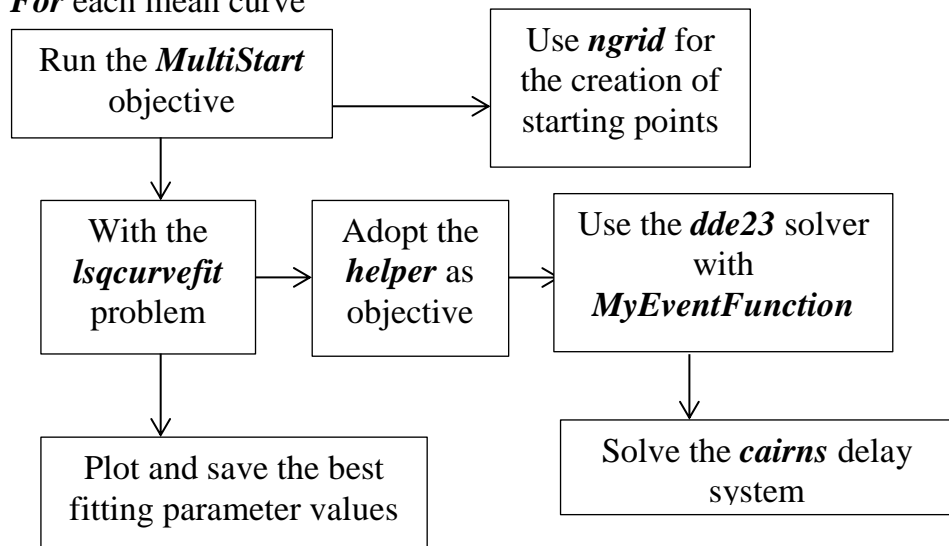
Where the research of x values is limited by a lower and an upper bounds, LB and UB respectively. In particular, the LB and UB for the growth rate are established considering its experimental value ($\text{Bound}_{growthR} = \text{Experimental}_{growthR} \pm \frac{\text{Experimental}_{growthR}}{5}$) while the bounds for the other parameters are fixed in a meaningful range. Lastly, in order to obtain the global minimum value, avoiding local minimum points, the research has to be extended to the entire range between LB and UB . This is possible thanks to the adoption of the *MultiStart.m* and the creation of a minimum optimization problem with *helper.m* as the objective of *lsqcurvefit.m*. The different starting points, which represent the initial parameter sets, are created using *ndgrid.m* and considering all the possible combinations of defined values between LB and UB . For example, considering $LB = [0.0137 \ 0 \ 0 \ 0 \ 0.0001]$ and $UB = [0.0205 \ 1e-9 \ 500 \ 30 \ 0.1]$, the bounds for growth rate, binding rate, burst size, latent period and decay rate respectively, and $\text{comNUMpar} = [3 \ 4 \ 3 \ 4 \ 1]$, the vector representing the number of

values between *LB* and *UB* taken into account as starting points (e.g. growth rate=[0,0137 0,0171 0,0205], binding rate=[0 3.3e-10 6.7e-10 1.0e-9], etc.), *ngrid.m* returns all the possible combinations of these values:

0,0137	0	0	0	0,10
0,0171	0	0	0	0,10
0,0205	0	0	0	0,10
0,0137	3.33E+04	0	0	0,10
0,0171	3.33E+04	0	0	0,10
0,0205	3.33E+04	0	0	0,10
0,0137	6.67E+04	0	0	0,10
0,0171	6.67E+04	0	0	0,10
0,0205	6.67E+04	0	0	0,10
0,0137	1.00E+05	0	0	0,10
0,0171	1.00E+05	0	0	0,10
...				

In order to avoid the model instability issue for some range of starting values, through the *ddeset.m*, it is possible to create a DDE (delay differential equation) option structure, in particular an “Event Location Property” called *MyEventFunction.m*. If the solver detects such event during the solution of a problem (i.e. the run time exceeds three seconds), the function will terminate the execution of the *dde23.m* solver. Summarizing:

For each mean curve



End.

3.2.4 Final parameter optimization.

The main purpose of the program *A4mainCROSS.m* is the identification of the best parameter set, valid for all the previous samples. It considers the results of different sets obtained by the execution of *A3mainfit.m*, as starting point. The model parameter optimization is possible by means of the Cross Validation method. The Cross Validation (or abbreviated CV) is a statistical method for the evaluation of accuracy and validity of a model (Refaeilzadeh, 2009). Indeed, in statistics, a typical goal is to learn a model from the available data. Consequently, evaluating the prediction capability of the model on the training data adopted, the model might show a good behaviour, but this is not an indication of its predicting performance for the future unseen data. The idea of CV, originated in the 1930s (Larson, 1931), is the estimation of the generalization performance and the model selection. In particular, the available dataset is divided into two segments: one used to learn a model (training set) and the other adopted to validate the model (validation or testing set). There are different types of CV, the main ones are:

a) *The holdout method*

It represents the simplest CV procedure. The dataset is globally divided into only 2 non-overlapped sets. The model fitting is performed considering the training set and the function approximator so obtained has the goal to predict the remaining unseen data of the testing set. The method has the advantage to be better than the normal residual evaluations, because of the presence of the testing set, and to be not computationally complex. The downside is the high variance of its evaluation, which depends heavily on how the available dataset is divided into the two different sets.

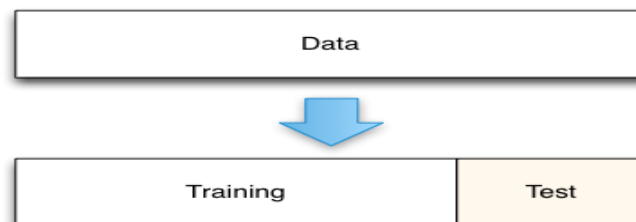


Figure 21: The hold-out data split. (<http://scott.fortmann-roe.com/docs/MeasuringError.html>)

b) *K-fold cross validation*

This method represents the next development of the previous one. At the first, the dataset is partitioned into k equal (or nearly equal) subsets or folds. The process considers one of the k subset as validation set and the other $k-1$ subsets as training set (Figure 21). This procedure is iteratively repeated k times and at each iteration, a different fold of the data is considered. In this way, every sample gets to be in a testing set once and in a training set $k-1$ times. With the increase of k value, the method shows a lower variance, but at the expense of the execution time and the computational complexity.

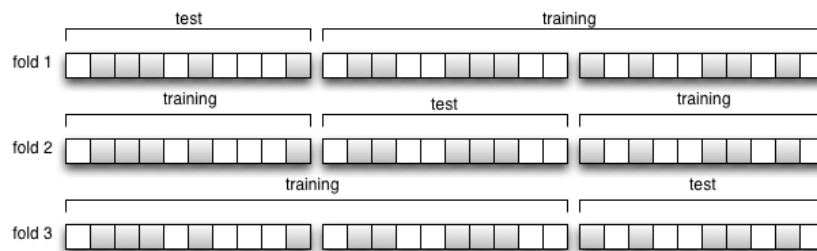


Figure 22: K-fold Cross Validation with $k=3$ and dataset of 30 elements. (http://homepages.inf.ed.ac.uk/pmartin/tutorial/case_studies.html)

c) *Leave-one-out cross validation*

This method, also abbreviated as LOOCV, is a special case of the K -fold CV with k equals to the number of data points N . Thus, during the N iterations, each time only one point is used for the testing while the remaining $N-1$ data are considered for the determination of the function approximator. The LOOCV presents a very accurate performance estimation but also a higher computational cost. It is widely used when only a small number of data is available.

Dataset	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	Test	Train	Train	Train	Train
2	Train	Test	Train	Train	Train
3	Train	Train	Test	Train	Train
4	Train	Train	Train	Test	Train
5	Train	Train	Train	Train	Test

Figure 23: The leave one out split. Every fold represents only one sample and k =number of samples. (<https://www.packtpub.com/books/content/learning-how-classify-real-world-examples>)

The CV method can be applied in three main contexts: firstly, for *the performance estimation* of the learner model from available data using one algorithm, in other words, to quantify the generalizability of an algorithm; secondly, for *the model selection*, the comparison of the behaviour of different algorithms considering the available data; lastly, for *the tuning of the model parameters*, considering the performance of two or more variants of a parameterized model and trying to achieve the best results with the dataset taken into account (Refaeilzadeh, 2009). The third and last case represents the main purpose of this work, as described at the beginning of this section. The available dataset considered is composed by the averaged curves of different replicates, they are not simple points but they represent the evolution of cell population in time, as a consequence of the interaction with infecting bacteriophages. The leave-one-out cross validation method was adopted, considering the small number of the experimental data used.

A4mainCROSS.m description:

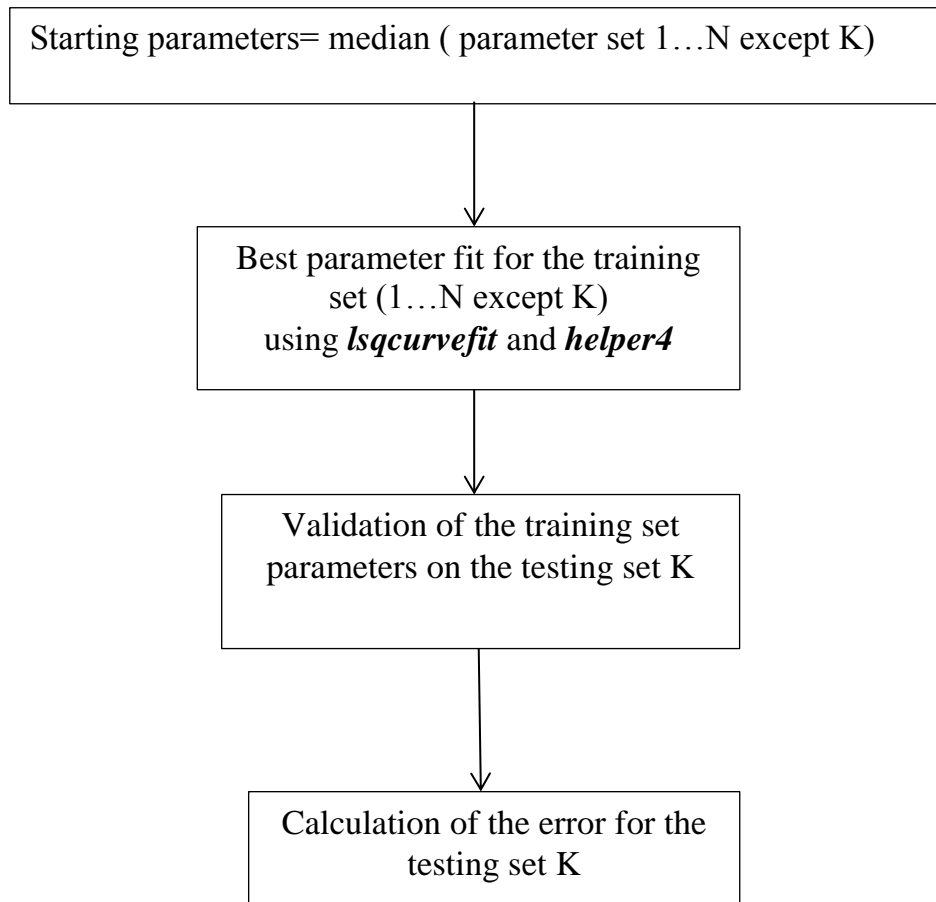
After the loading of the data coming from the previous software programs, the execution of *A4mainCROSS.m* is mainly based on the *for* cycle that represents the implementation of the LOOCV method. In particular, during each iteration, one mean curve is held-out for validation (test) while the remaining ones are used for learning (training). The procedure repeated cyclically adopts, as starting parameters, the median values of the parameter sets, obtained in *A3mainFIT.m*, but only those attributed to the training set. Using *lsqcurvefit.m* and *helper4.m*, it is able to find the best fitting parameter values for the training set taken into consideration and to test this set on the validation mean curve. The following error is recorded each time:

$$\mathbf{Error}_k = \sum (Y_{model\ k-1} - \hat{Y}_{experiment})^2 \quad (24)$$

In other words, the error for the k th sample is represented by the sum of the squared difference between the fitting curve using the best parameter set obtained for the training set $k-1$ and the experimental data of the k th mean curve. At the end of the

execution of all the iterations, there are k parameter sets, with k equals to the number of available samples, but only the one with the minimum error accumulated, represents the final optimized parameters. The latter was adopted for the final fitting and plotting of all the data. Summarizing the iterative process:

For each mean curve k (with $k=N$, the number of samples)



End

3.2.5 Model prediction

The last program implemented *A5mainPREDICTION.m* has the purpose to collect the data elaborated by the previous scripts and to use them for the model prediction. This is realizable thanks to the conversion of the new data, expressed in optical densities, in CFU/mL and the normalization of them, as well as of the previous model “training” dataset, to their respective starting OD values. Indeed, one of the two main aims of this project is the prediction of one or more unknown concentrations of bacteriophages. In order to get new curves over time that are comparable to the model training dataset, it is important to follow the same procedure adopted previously, thence the use of a microplate reader, the measurement of the OD₆₀₀ every five minutes, etc., in other words, to recreate the same experimental condition. In addition, considering the model, it is necessary to maintain the same starting bacterial concentration of the previous experiment and to infect it with an unknown dilution of phages of interest. Only in this way it is possible to obtain a curve over time that describes the variation of the cell population in CFUs and to make predictions. One of the issues of this procedure is the presence of variability in the starting OD values of different experiments. It is highly unlikely the adoption of the same concentration of bacteria every time. Thus, the normalization of the data is a way to “synchronise” them, in order to get a better comparison.

The correlation coefficient is adopted as criterion of comparison between model and unknown curves. In general, the correlation quantifies the extent to which two variables, X and Y, vary in related way. The relationship between them can be easily found considering the scatter plot of the data (Figure 24). A positive correlation exists when high values of X are associated with high values of Y. Conversely, a negative correlation exists when high values of X are associated with low values of Y. There is no correlation if the values of X are not at all predictive of values of Y. The degree of the correlation is quantified by the correlation coefficient *R*, which is a unit-free term. The value of *R* always lies between -1 and +1. The calculated *R* assumes a positive value when the slope on the graph rises from left to right and a negative value

when the slope rises from right to left. Ideally, $R=+1$ or $R=-1$ if all the points of a scatter plot fall on a line with upward or downward incline respectively. Such perfect correlation is seldom encountered (e.g. when the two variables compared are identical $Y=X$). However, the correlation coefficient defines the correlational strength, and strong correlations are associated with scatter clouds that adhere to the imaginary trend line. Therefore, the closer R value is to $+1$, the stronger the positive correlation. The closer R is to -1 , the stronger the negative correlation.

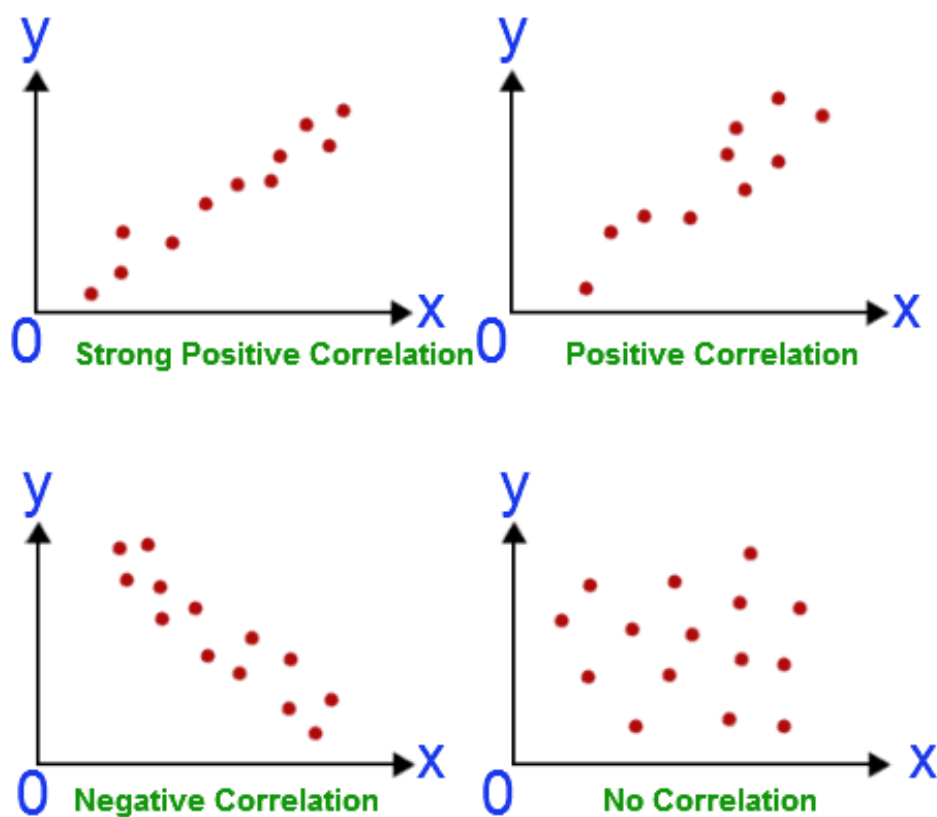


Figure 24: Scatter plot of the variables X and Y. (<http://math.tutorvista.com/statistics/correlation-and-regression.html>)

The common formula for the measure of the linear correlation between X and Y is called Pearson product-moment correlation coefficient and it is defined as:

$$R = \frac{SS_{XY}}{\sqrt{SS_{XX} * SS_{YY}}} \quad (25)$$

It requires the calculation of three different sums of squares (SS):

-the sum of squares for variable X

$$SS_{XX} = \sum (x_i - \bar{x})^2 \quad (26)$$

-the sum of squares for variable Y

$$SS_{YY} = \sum (y_i - \bar{y})^2 \quad (27)$$

-the sum of their cross-products

$$SS_{XY} = \sum (x_i - \bar{x}) * (y_i - \bar{y}) \quad (28)$$

In this work, the linear correlation is considered as the yardstick for the sample but it is important to underline that a value of R close to 0 means that there is not a linear correlation and not necessary that there is not a relationship between them.

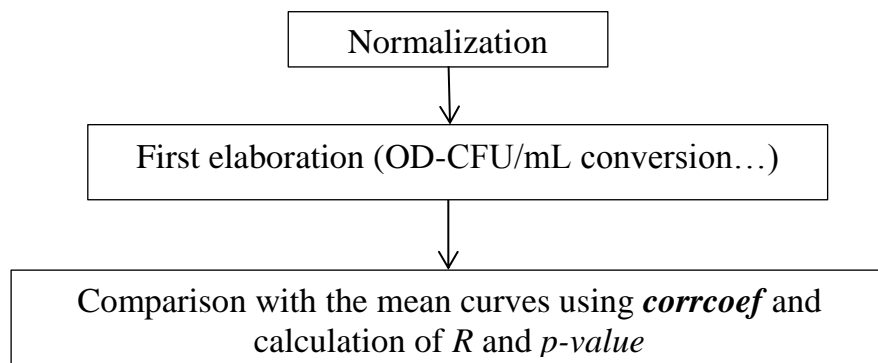
Another significant variable to take into consideration is the *p-value*. This is a method, adopted in statistics, which gauges the “significance” of the accomplished analyses. In this case, it is possible to consider the no correlation between the variables X and Y as null hypothesis and their correlation as the alternative hypothesis. The *p-value* is a number that lies in the interval [0,1] and it can be interpreted in the follow way:

- A small *p-value* (usually ≤ 0.05) suggests that the observed data are inconsistent with the assumption that the null hypothesis (no correlation) is true, and thus that hypothesis must be rejected and the alternative hypothesis (X and Y are correlated) is accepted as true;
- A large *p-value* (usually > 0.05) indicates weak evidence against the null hypothesis, so the null hypothesis cannot be rejected.

A5mainPREDICTION.m description:

After the loading of the data previously elaborated, the user can select the excel file containing the new data to predict. Both the mean curves and the unknown samples are normalized. A second step of elaboration consists of the cutting or the interpolation of the new data, depending on their size, and the OD conversion in CFU/mL, in order to have all the data with the same length, then comparable. Using the MATLAB function *corrcoef.m*, the correlation coefficient and the *p-value* are calculated, as result of the comparison between each single new sample and each mean curve. An unknown sample is more similar to the mean curve that returns the highest value for *R* and with a *p-value* <0.05. In particular, a strong correlation is present when $|R|>0.75$, and a moderate correlation when $0.75>|R|>0.67$. In addition, if the $|R_{max}/R_{second}|\leq 0.03$, where R_{max} is the maximum *R* value and R_{second} the coefficient that represents the correlation between the unknown sample and a second mean curve, then the prediction is between two results. Knowing the starting concentration of bacteria, the final outcome of the program is the prediction of the starting unknown concentration of phages for all the new data. Summarizing the *for* cycle for the comparison:

For each new data



End

Chapter 4

4 Results and Discussion

4.1 CFU/mL

The CFU/mL value for the MG1655 Δ fimA-H; Δ flu; Δ matB::P2_luxCDABE bacterial strain was calculated considering an $OD_{\text{spectrophotometer}}$ of about 0.2. For the bacterial enumeration, two experiments were conducted in different days and for each one, the specimens were plated in three replicates. The final value represents the average result of the count of colonies (figure 25).

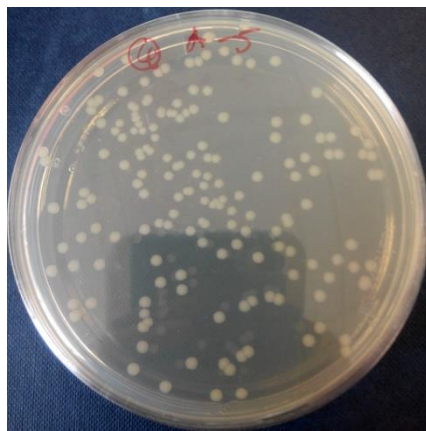


Figure 25: Bacterial colonies after an overnight incubation at 37°C.

CFU/mL	Replicate1	Replicate2	Replicate3
Experiment 1	61*10E+6	62*10E+6	57*10E+6
Experiment 2	64*10E+6	63*10E+6	55*10E+6



from the equation (4) $\frac{CFU}{mL} = 6 * 10^7$

4.2 PFU/mL

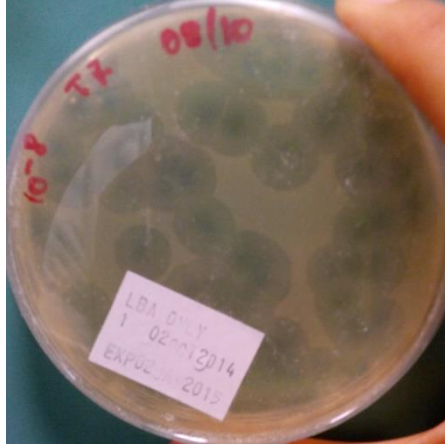


Figure 26: Viral Plaques after an overnight incubation at 37°C. Plaques are circular and clear zones formed in a lawn of cells due to lysis by phage.

The PFU/mL value for the T7Δgp5 bacteriophages was calculated considering the Plaque Assay result. Following the Plaque Assay protocol, 0.1mL of phages were inoculated in 0.3mL of bacteria with a $OD_{\text{spectrophotometer}}$ of about 0.2 ($CFU=0.3 \cdot 6 \cdot 10^7=1.8 \cdot 10^7$). After an overnight incubation at 37°C, the number of plaques (Figure 26) was counted from each one of the three replicates plated and averaged, obtaining in this way the final value:

PFU/mL	Replicate 1	Replicate 2	Replicate 3
Plaque Assay-OD=0.2	$30 \cdot 10E+9$	$35 \cdot 10E+9$	$32 \cdot 10E+9$



from the equation (5)
$$\frac{PFU}{mL} = 3 \cdot 10^{10}$$

In order to see which could be the dependence of the PFU/mL value on the concentration of cells used, a second Plaque Assay experiment was accomplished considering a starting $OD_{\text{spectrophotometer}}$ of about 0.8 ($CFU/mL \gg 6 \cdot 10^7$) and no significant differences were founded:

PFU/mL	Replicate 1	Replicate 2	Replicate 3
Plaque Assay-OD=0.8	$35 \cdot 10E+9$	$36 \cdot 10E+9$	$40 \cdot 10E+9$

4.3 MOI

As previously described (section 3.1.3) the number of phages that infects each bacterium at different MOI can be described by the Poisson distribution. Considering the probability to get infected cells:

$$P(n > 0) = 1 - P(0) = 1 - e^{-m}$$

As the MOI increases, the percentages of cells infected with at least one viral particle also increases:

MOI	% Infected
0.01	0.10%
0.1	9.50%
1	63.20%
2	86.50%
3	95.00%
4	98.20%
5	99.30%
6	99.80%
7	99.90%
8	~100.0%

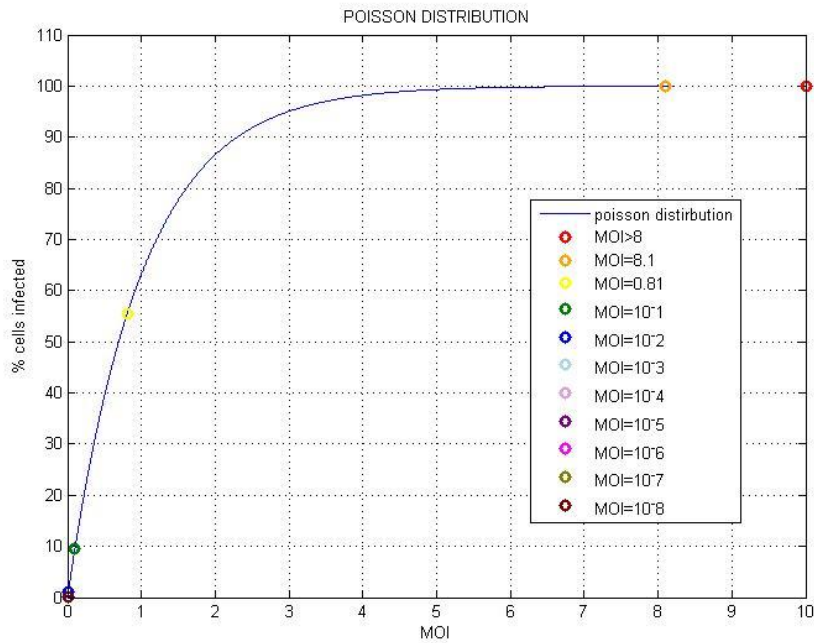


Figure 27: Percentage of infected cells at different MOI.

Considering the CFU/mL, PFU/mL and the phage-host interaction protocol (section 3.1.4), which consists of the addition of 0.02mL of phages and 0.18mL of bacteria in each well, the MOI was calculated as:

$$\text{from the equation (6)} \quad \text{MOI} = \frac{3 * 10^{10} * 0.02}{6 * 10^7 * 0.18} \approx 60$$

This MOI value is valid for the undiluted concentration of phages. Consequently, considering the concentration of cells constant, for each serial diluted sample, the MOI would be:

Dilution	MOI
10 ⁰ -undiluted	60
10 ⁻¹	6
10 ⁻²	0.6
10 ⁻³	0.06
10 ⁻⁴	0.006
10 ⁻⁵	0.0006
10 ⁻⁶	0.00006
...	...

It is strictly necessary maintaining the same starting OD or CFUs both for the “training” dataset and for the future unknown sample. Only in this way the prediction of the concentration of bacteriophages is possible. Indeed, the curves obtained by the evolution over time of different sample with a same dilution of phages but different starting OD, are different, so the prediction could not be reliable (Figure 28). Another observation is that there is also a difference between achieving the correct starting OD directly and obtaining it upon dilution of the bacterial culture. For instance, if the starting OD chosen is 0.2, then growing a fresh culture from a very low concentration until 0.2 is different from reaching this value diluting a culture at an OD of 0.7 with LB medium. In particular, the result of the second method is the presence of a delay for the curves (Figure 29), whose entity increase with the dilution factor.. Thus, only the first growth method was adopted.

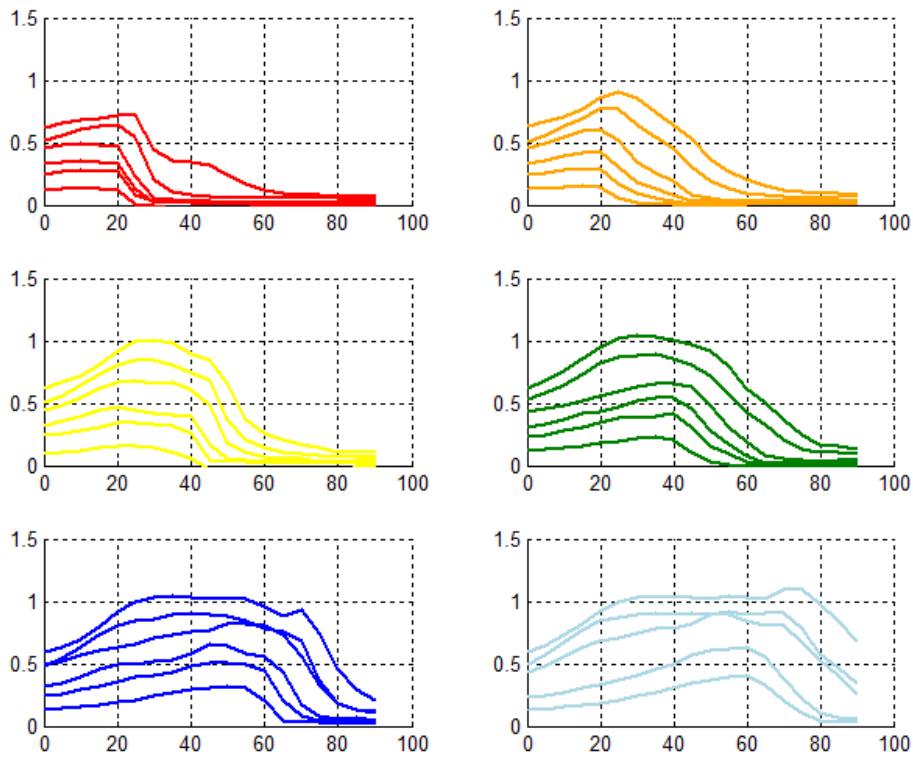


Figure 28: Curves with the same phage dilution (the same colour) but with different starting concentration of cells. X=time; Y=OD.

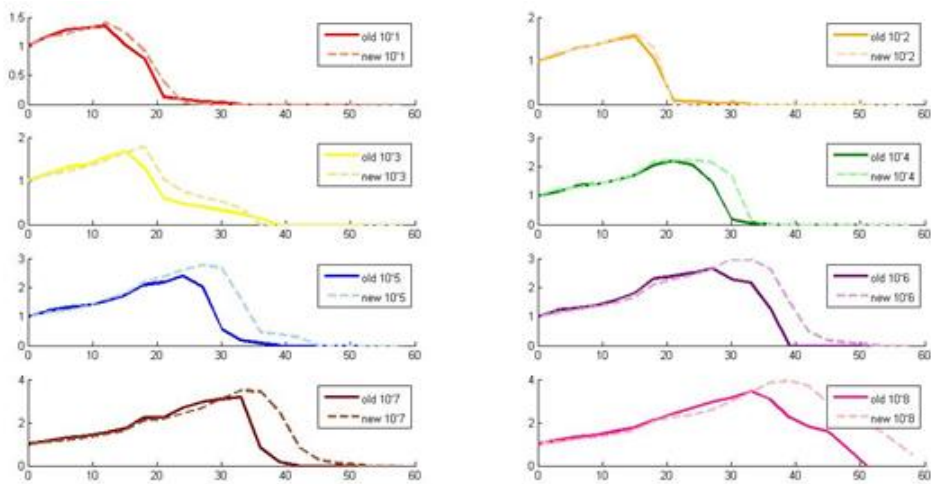


Figure 29: Curves after normalization, every colour represents a different dilution of phage. Solid lines: starting OD (0.2) value achieved directly from the growth of a low bacterial concentration (0.04). Dotted lines: starting OD (0.2) value achieved diluting a high bacterial concentration (0.7).

4.4 *AImainPLOT.M*

The *AImainPLOT.m* program plots, at first, the general results coming from the microplate reader experiment, generating a figure for each replicate (Figure 30). Once the user has selected the data of interest and the cutting time (Figure 31), the program is able to elaborate and summarize them considering the average and the standard deviation of their values in time (Figure32).

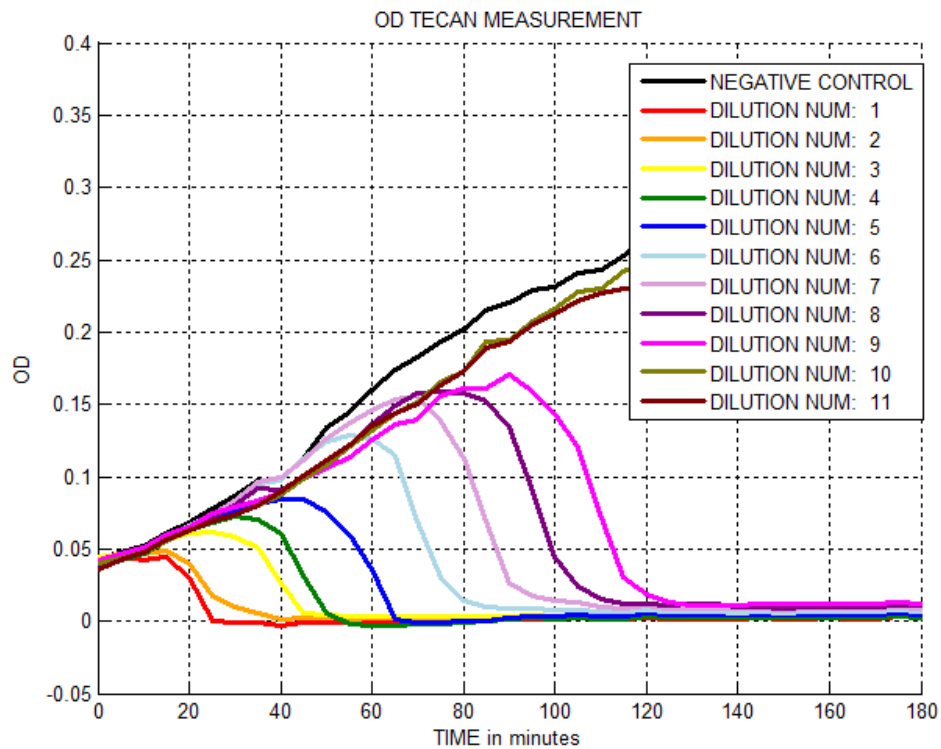


Figure 30: The plot over time of a single replicate with all its samples.

```
Command Window
****ATTENTION: Select the excel file with ONLY the microplate reader OD values!!****

****PRESS Ctrl+C to exit from the program****

*Do you want to setup the blank manually (press 1)
or automatically using the first line of the microplates (press2)? 1
*Write the OD tecan value for the blank (for example 0.08 (LB) or 0.07 (2xYT)): 0.08
*Key the number of replicates for every dilution (for example 7): 7
*Write the number of dilutions for every line, including the negative control (for example
*Do you want to proceed with the mean curves calculation? YES(1) NO(2) 1
*Key the number in minutes for the cutting (for example 90): 90
*Press 1 for automatical training set creation,
2 for manual traning creation (only for experts): 1
*Interval of replicates (for example [5 7]): [5 7]
*Interval of dilutions (for example [3 7]): [3 7]
*****END*****
fx >>
```

Figure 31: The command window of *AImainPLOT.m*.

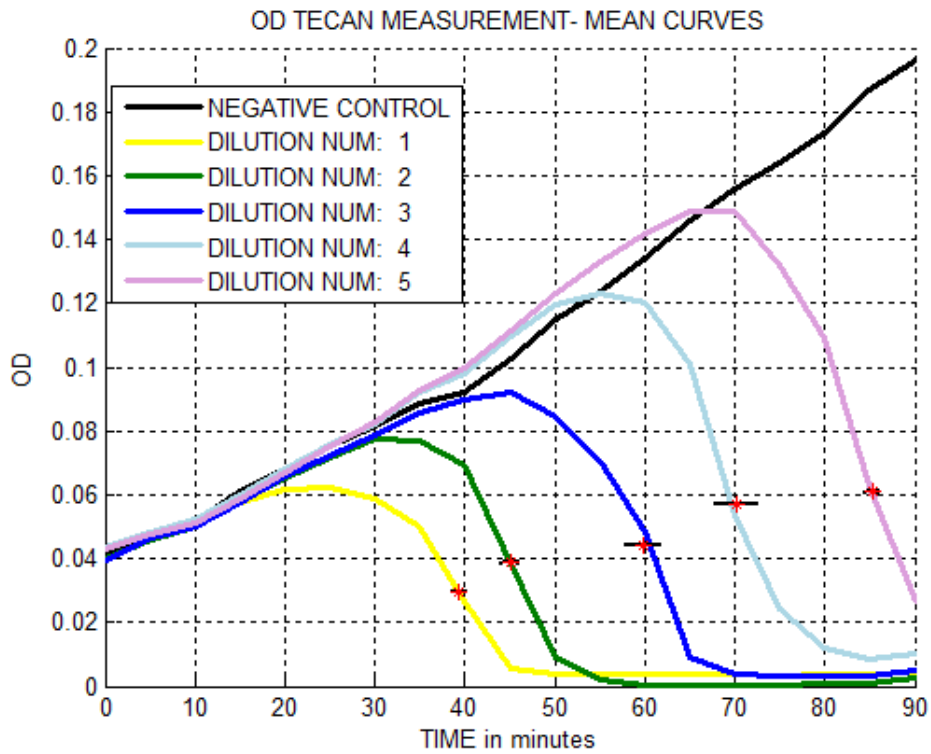


Figure 32: The summary of the data in mean curves. Each black line represents the standard deviation of the respective curve.

4.5 A2odCFUcalibration.m

The experimental results of the OD-CFU/ml calibration are summarized in the following table:

OD-CFU/mL calibration	Replicate 1	Replicate 2	Replicate 3
OD: 0	0	0	0
OD: 0.150	5.20E+07	6.50E+07	5.00E+07
OD: 0.214	6.30E+07	6.40E+07	5.50E+07
OD: 0.329	7.30E+07	6.60E+07	7.50E+07
OD: 0.495	1.64E+08	1.47E+08	1.46E+08
OD: 0.700	3.80E+08	5.00E+08	4.20E+08

Where the number of colonies was counted from each replicate after an overnight incubation at 37°C and consequently, the CFU/mL was calculated taken into account the dilution factor. The linear model that represents the best fitting of the data was automatically calculated by *A2odCFUcalibration.m* (Figure 33) and plotted (Figure34).

```

Command Window
****ATTENTION: Select the excel file with ONLY spectrophotometer OD values!!!!****

****PRESS Ctrl+C to exit from the program****

*Press 1 for automatical calibration, 2 for manual calibration (only for experts): 1

fittedmodel =

Linear model Poly1:
fittedmodel(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 2.991e+08 (2.31e+08, 3.672e+08)
p2 = 2.707e-09 (-2.475e+07, 2.475e+07)

*CFU/mL immission:
for MANUAL press 1;
with GINPUT press 2;
from the STARTING OD INFO press 3: 1
*Key the CFU/ml (for example 6*10^7 or 60000000): 6*10^7
*****END*****
fx >>

```

Figure 33: OD-CFU/mL calibration results.

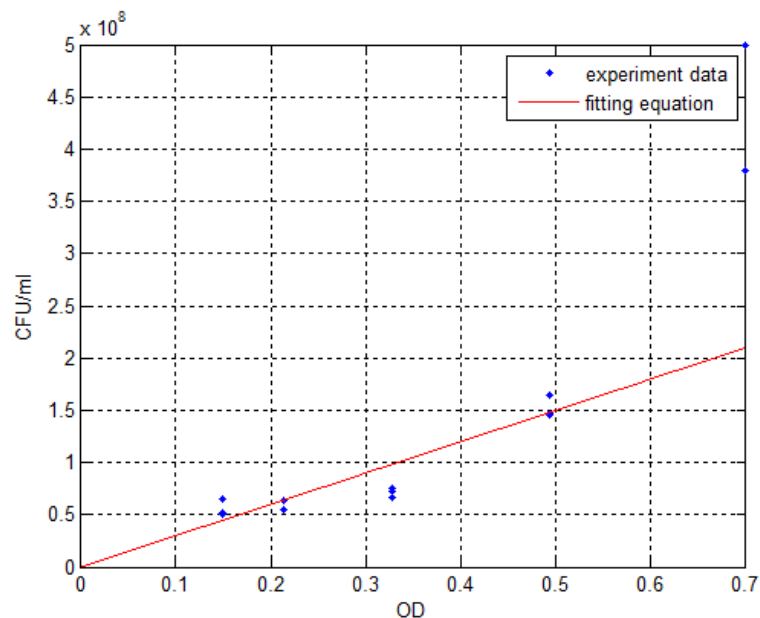


Figure 34: The plot of the linear fitting equation.

Considering the presence of a bias between $OD_{\text{spectrophotometer}}$ and OD_{tecan} , an additional experiment was performed in order to obtain a conversion value:

OD spectr	OD tecan	ODspectr/ODtecan
0.15	0.051	2.94
0.214	0.056	3.82
0.329	0.101	3.25
0.495	0.143	3.46
0.7	0.197	3.55



$$OD_{\text{spectrophotometer}} \approx 3.4 * OD_{\text{tecan}}$$

4.6 *A3mainFIT.m* and *A4mainCROSS.m*

The results of the *A3mainFIT.m* program are shown below:

Curve (PFU/mL)	Growth Rate [min ⁻¹]	Binding Rate [CFU ⁻¹ min ⁻¹]	Burst size [PFU]	Latent period [min]	Decay Rate [min ⁻¹]
3*10 ⁸	0.0175	1.81E-10	433	9	0.0091
3*10 ⁷	0.022	1.78E-10	483	8	0.0088
3*10 ⁶	0.022	1.96E-10	520	10	0.0091
3*10 ⁵	0.213	1.45E-10	520	8	0.0077
3*10 ⁴	0.0224	2.05E-10	520	10	0.0089

In particular, the global minimum research, performed for each sample, requires an execution time in the order of hours. This computational period is strictly depending on the number of data considered, the bounds and the starting points adopted. The parameter optimization was accomplished using the Leave-One-Out Cross Validation method and the final values for the parameter set are:

Final Parameter set	Growth Rate [min ⁻¹]	Binding Rate [CFU ⁻¹ min ⁻¹]	Burst size [PFU]	Latent period [min]	Decay Rate [min ⁻¹]
	0.0213	1.75E-10	500	9	0.0089

Afterwards, the optimized values were adopted in order to solve the delay differential equations of the model and to obtain the ultimate curve fitting (Figure 35).

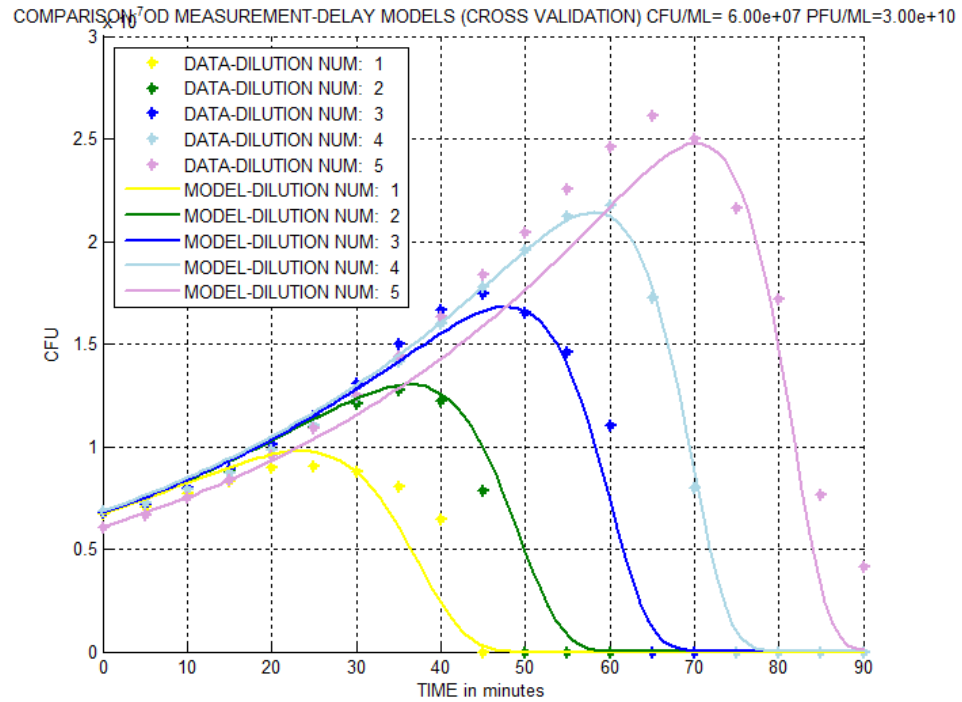


Figure 35: The curve fitting with the optimized parameter set.

A.7 Experimental parameters

a) Growth Rate

The growth rate value is automatically calculated by the program *A3mainFIT.m* using the mean curve of the negative controls (Figure 36):

$$\text{Experimental Growth Rate} = 0.0219 \text{ [min}^{-1}\text{]}$$



$$\text{Experimental Doubling Time} = \frac{0.693}{0.0219} \approx 32 \text{ [min]}$$

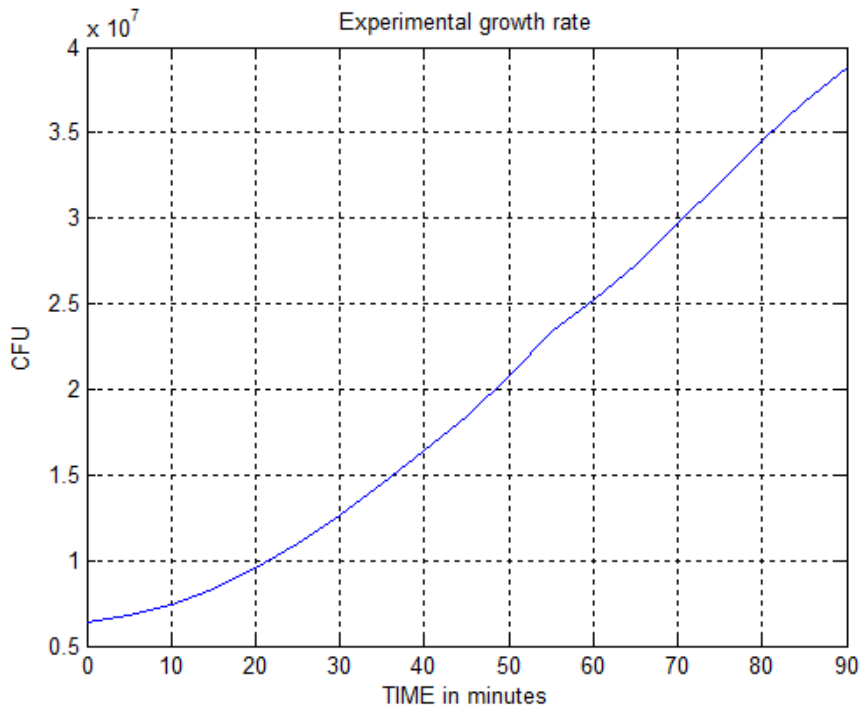


Figure 36: Experimental growth rate.

b) Latent Period

The one-step growth experiment results are shown in the following table:

One-step growth experiment (Number of plaques *10⁴)	Replicate 1	Replicate 2	Replicate 3
0 ⁻ min	50	60	40
0 ⁺ min	36	34	31
2 min	28	34	31
4 min	36	30	31
6 min	33	30	30
8 min	27	26	24
10 min	21	23	27
11 min	25	18	16
12 min	28	24	20
14 min	600	460	390
16 min	560	520	440

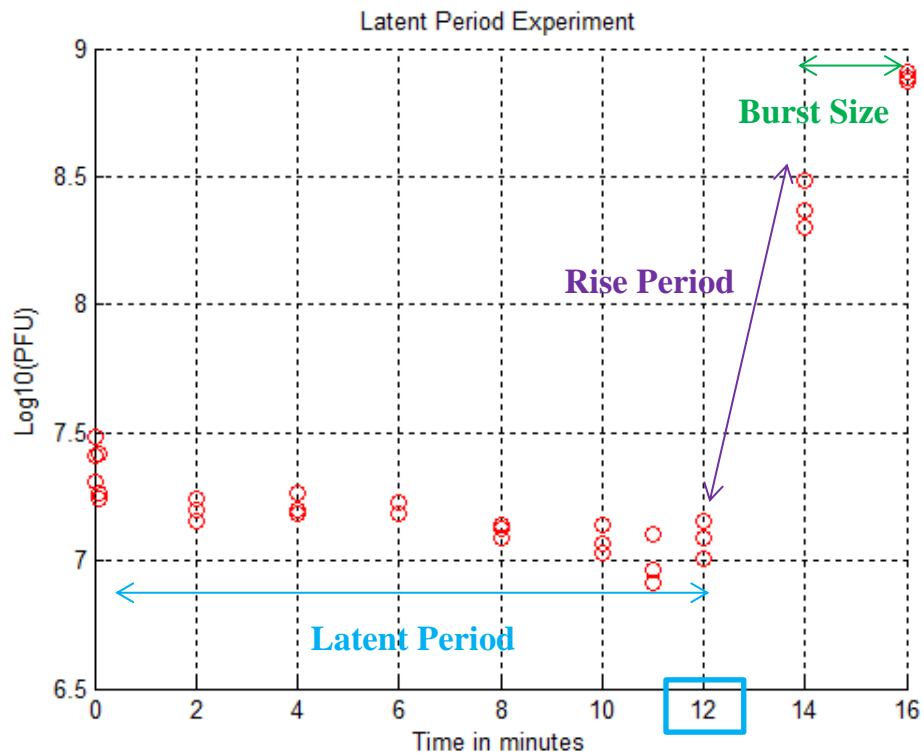


Figure 37: One-step growth for the Latent Period determination.

From the plot of these values in Log_{10} (PFU) (Figure 37), it is possible to observe a remarkable increase in the number of plaques after about 12 minutes. The experiment was repeated in different days in order to get a better accuracy and it showed the same result every time. Thus, the latent period is:

$$\text{Experimental Latent Period} \approx 12 \text{ [min]}$$

c) Binding Rate

Considering the equation (21) and the one-step growth result until 10 minutes (before the lysis of bacteria), the curve that fits the data is a straight line (Figure 38) whose slope is $S_0 * b$, with S_0 the starting number of susceptible bacteria. Hence, the binding rate b is (slope/ S_0):

$$\text{Experimental Binding Rate} = 1.92 * 10^{-11} [\text{CFU}^{-1} * \text{min}^{-1}]$$

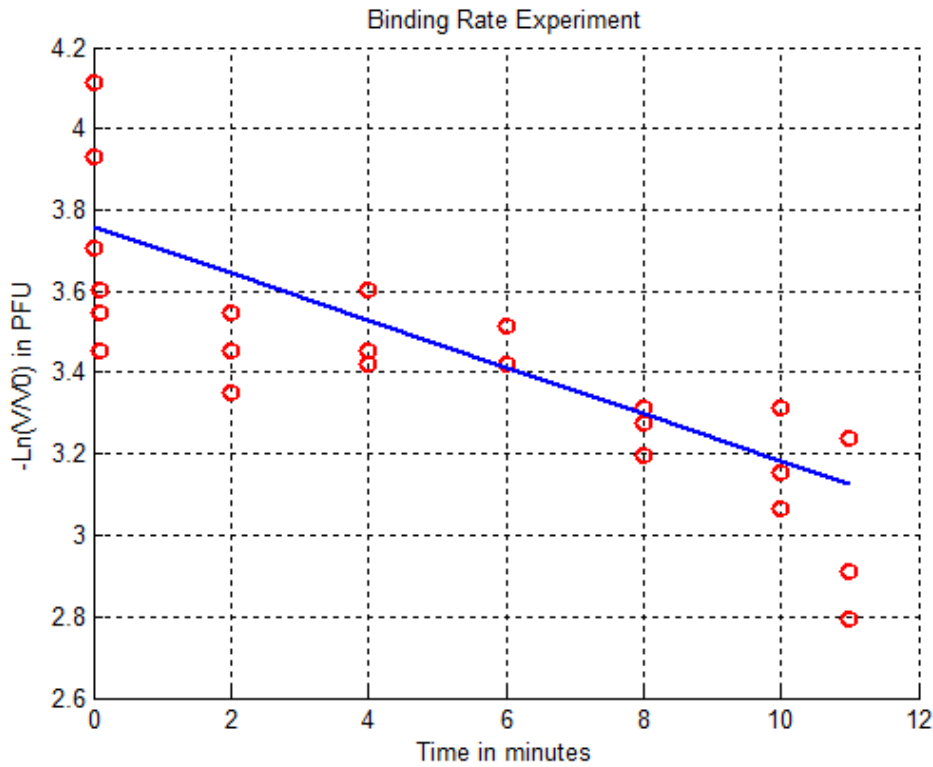


Figure 38: One-step growth for the Binding Rate determination.

d) Burst Size

The final formula taken into account for the determination of the burst size value is:

$$\mathbf{Burst\ size} = \frac{F_{14-16min}}{(T_{0+min} - U_{4min})} \quad \mathbf{(18. a)}$$

Where T is the starting concentration of total phage immediately after the inoculation of phages in bacteria; U is the number of unadsorbed phage; F is the final concentration of phages after the latent period. The number of plaques (PFUs), as function of the time, can be considered roughly constant until 12 minutes (latent period), so the difference between T_{0+min} and U_{4min} in the number of plaques counted onto the plates, is approximated to 1. The burst size value is:

$$\mathbf{Experimental\ Burst\ Size} = \frac{484 \sim 507 * 10^4}{(34 - 33) * 10^4} \approx 484 \sim 507 \mathbf{ [PFU]}$$

e) Decay Rate

The 7-day decay experiment results are shown in the following table:

Decay Rate Experiment (PFU/mL)	Replicate 1	Replicate 2	Replicate 3
0 hours	2.30E+09	1.60E+09	1.60E+09
29 hours	6.00E+08	3.00E+08	3.00E+08
99 hours	7.40E+07	5.40E+07	6.10E+07
121 hours	3.10E+07	3.70E+07	3.70E+07
147 hours	2.10E+07	2.50E+07	2.50E+07

Using the equation (16) for the fitting, the phage decay rate represents the slope of the straight line (Figure 39):

$$\text{Experimental Decay Rate} = 0.001 \text{ [min}^{-1}\text{]}$$



Figure 39: Decay Rate Experiment.

4.8 Comparison between experimental and model optimized parameter values

Comparison Parameter Values	Growth Rate [min ⁻¹]	Binding Rate [CFU ⁻¹ min ⁻¹]	Burst size [PFU]	Latent period [min]	Decay Rate [min ⁻¹]
Model	0.0213	1.75E-10	500	9+3	0.009
Experiment	0.0219	1.92E-11	484-507	12	0.001

The previous table summarizes the parameter set determined by model optimization and that determined from experiments. The values reported do not show significant differences demonstrating that the automatic global minimum research and the CV method, implemented in the software system, are able to return, as a result of the optimization, parameter values that are meaningful and realistic. In other word, they are close to the reality, a good representation of the phage-host interaction experimentally observable. In particular, the growth rates are very similar because the bounds adopted for the parameter optimization are in the range of the experimental values. Taking into consideration the starting period (3 minutes) of shaking, programmed by the microplate reader experiment in order to make the optical density measurement uniform, even the latent period values coincide. While the slight differences displayed for the remaining values may be determined by two factors: first, due to the presence of a high variability in both observed and predicted value, especially in the count of PFUs; second, the model may need further modifications, for example, adsorption rate and latent period may depends on the initial host density or some parameters may vary with time (or better with the physiological condition of the host).

4.9 A5mainPREDICTION.m-Validation step

The script *A5mainPREDICTION.m* was used at first for the validation of the implemented software system. Considering the optimized model parameters and the training data, the program is able to predict the starting concentration of phages in unknown

samples. The test realized for the model validation consists of the prediction of various specimen sets coming from five different microplate reader experiments. The total testing set is composed by 235 samples in which the starting concentration of phages is known. The testing data were obtained using the same protocol (section 3.1.4) and cell starting OD₆₀₀ of the training set. As result, the program is able to return the CFU/ml, PFU/mL, MOI and the plot of every sample (Figure 40-41).

```

Command Window

*****PREDICTION FOR THE SAMPLE NUM 1!*****

PFU/ml prediction: 3.00e+08
CFU/ml prediction: 4.68e+07
MOI prediction: 7.13e-01

*****PREDICTION FOR THE SAMPLE NUM 2!*****

PFU/ml prediction: 3.00e+07
CFU/ml prediction: 4.21e+07
MOI prediction: 7.92e-02

*****PREDICTION FOR THE SAMPLE NUM 3!*****

PFU/ml prediction: 3.00e+06 |
CFU/ml prediction: 3.75e+07
MOI prediction: 8.90e-03
  
```

Figure 40: The results of the prediction in the command window.

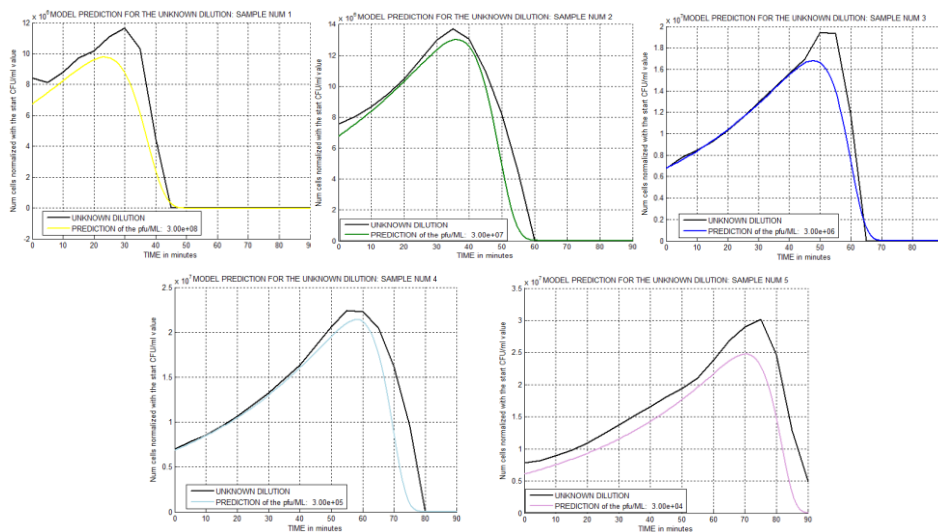


Figure 41: Examples of the prediction for different mean curves.

The validation of the model parameter value comparing predicted and theoretical concentrations of phages has shown a probability of the 91.9% to get correct results. In particular, the 8.1% of error involves the prediction of mean curves having lower MOI values (<0.006), this means lower concentrations of phages considering the cell number roughly constant. A possible explanation for that might be the presence of a high variability in the curves over time of the samples with low phage concentration, as the reflection of an issue at an operative point of view (Figure 42). Indeed, when the bacteriophages are strongly diluted, the probability to withdraw and inoculate the same number of phages in the different samples is low. This could be also the reason why the 8.1% of no-correct prediction may not be considered as error, also due to the fact that the phage dilutions for the testing set are known only theoretically (from the serial dilution), so the variation in the number of phages for each sample could determine a different prediction. Another observation is that the error in the prediction may be determined by the use of a starting OD value different from that of the training set and the error probability grows with the increase of the gap between the initial conditions for training and testing sets.

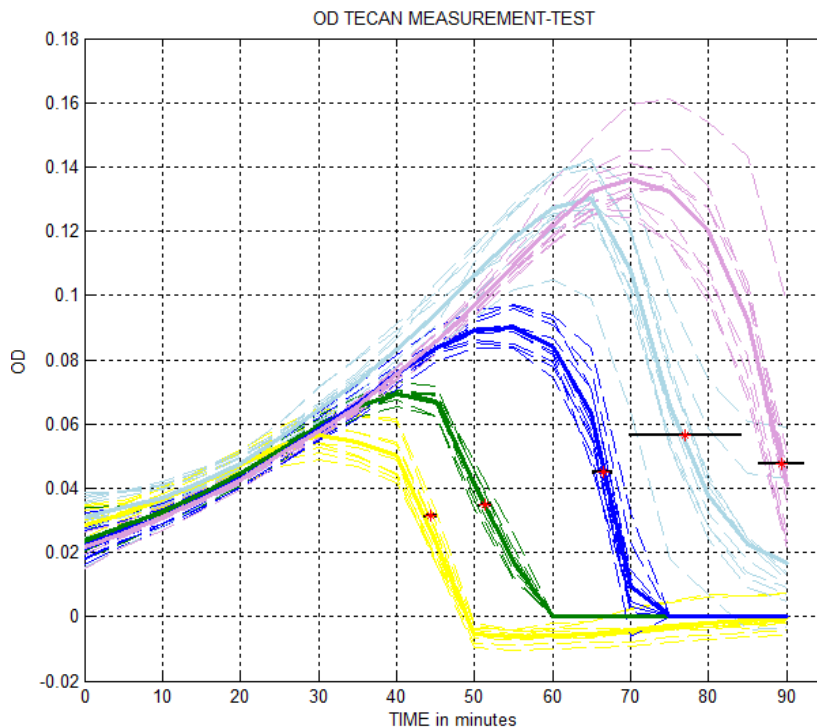


Figure 42: The variability for the samples of every phage dilution.

Chapter 5

5 Conclusion and future directions

5.1 Conclusion

The software system developed demonstrates to have many advantages. Firstly, it is capable of predicting, with high reliability, one or more concentrations of phages more quickly than the traditional Plaque Assay method. Indeed, the waiting time for the results is not in the order of up to 24 hours but about 90 minutes. Secondly, it shows to be good and useful for differential comparisons. For instance, it could be adopted for the monitoring of a variable over time, measuring the concentration of phages at different time intervals and comparing the results. Lastly, the software system is able to give a meaning to the obtained experimental data, making inference about the unknown model parameter set values which describe the kinetic interaction between phage and their host.

5.2 Future directions

Future work will be aimed at testing the software system with different types of cells and/or phages. As proof of this concept, the flexibility of the program was evaluated using a different strain of E. Coli (MG1655) with T7 Δ gp5 phages. The results after cross validation (Figure 43) show a good fitting of the experimental data and the prediction program was tested on 15 samples obtaining a probability of correct prediction of 93%. As mentioned above, it will be important, in particular, to use as training and testing set, for each evolution cycle, the data from the interaction of new evolved bacteria that are resistant to the starting phages and new mutated phages capable of infecting and defeating the described resistant hosts.

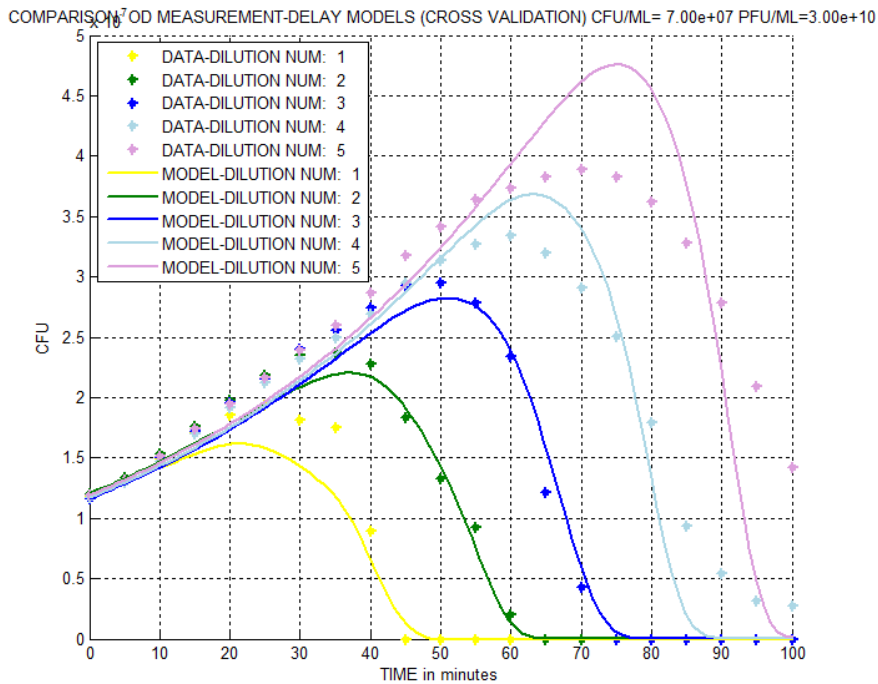


Figure 43: The curve fitting results of the interaction between MG1655 and T7Agp5.

A second important future extension of the project will involve the implementation and adaption of the software system on Arduino electronic platform for the Bioreactor output monitoring and evaluation. In particular, creating an optical density reader and adding it at the outflow of the Bioreactor (cellstat) opportunely connected with a fresh cell culture (chemostat), it will possible to automatically obtain the experimental curves of phage-host interaction, without the use of an external microplate reader. In this way, the prediction of the unknown concentration of phages will be in near real-time.

One last future orientation will be the modification of the model for a single-cell level. Indeed, using microfluidic systems and microscopy techniques, it will be possible to observe the dynamic behaviour between phages and bacteria taking in consideration a limited number of them (Figure 44).

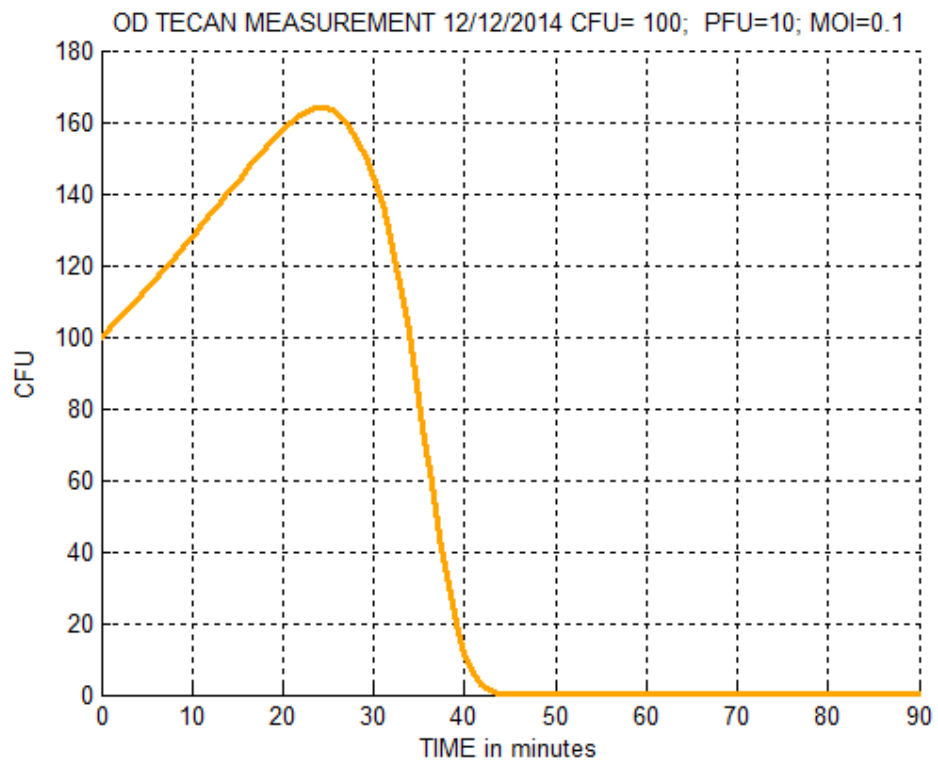


Figure 44: An example of the growth curve as result of the interaction of 100 cells and 10 phages.

Appendix-Matlab Codes

1 AlmainPLOT.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      AlmainPLOT.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Rules for the plot%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%1) qimron protocol
%2) starting od from the spectrophotometer of 0.2-0.3
%3) microplate: every row is a replicate, every column a
different
%dilution, the first one is the negative control (no
phages) after
%%10^-1...10^-8
%4) MOI no over 8!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PLOT results with Qimron's protocol

clear all
close all
clc
warning('off','all')

%% DATA IMPORT
fprintf('****ATTENTION: Select the excel file with ONLY the
microplate reader OD values!!!!****\n\n ****PRESS Ctrl+C to
exit from the program****\n\n')
[FileName,PathName,FilterIndex] = uigetfile('*.');
%%10/12/2014

data= xlsread(strcat(PathName, FileName));
[r,c]=size(data);
t=0:300:300*(c-1);

%% BLANK
media=input('*Do you want to setup the blank manually
(press 1)\n or automatically using the first line of the
microplates (press2)? ');
```



```

if media==1

blank=input('*Write the OD tecan value for the blank (for
example 0.08 (LB) or 0.07 (2xYT)): ');

%%%CONTROL%%%
if (blank>1 || length(blank)>1 || isnumeric(blank)==0)
    error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

shiftBLANK=0;
else if media==2
    blank=mean(data(1:12,1));
    shiftBLANK=12;
    else error('***ERROR: NO VALID INPUT; PLEASE PRESS 1 OR
2!*** \n')
    end
end

colori={'Red','Orange','Yellow','Green','Blue','LightBlue',
'Plum','Purple','Magenta','Olive','Maroon'};

%% NUMBER OF REPLICATIONS
numREP=input('*Key the number of replicates for every
dilution (for example 5): '); %7

%%%CONTROL%%%
if (numREP>8 || length(numREP)>1 || isnumeric(numREP)==0)
    %test if numREP is a number, a single value, <=8!
    error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

replicates=[];

line1=2+shiftBLANK; %negative control is the first line
line2=input('*Write the number of dilutions for every line,
including the negative control (for example 12): '); %12

%%%CONTROL%%%
if (line2>12 || length(line2)>1 || isnumeric(line2)==0)
    %test if numREP is a number, a single value, <=8!
    error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
shiftLINE=line2;
line2=line2+shiftBLANK;

```

```

for i=1:numREP
    replicates(:, :, i)=data(line1:line2, 1:c);
    line1=line1+shiftLINE;
    line2=line2+shiftLINE;
end

%% NEGATIVE CONTROLS, NO PHAGE
NC=[];
nc=1;
for i=1:numREP
    NC(:, i)=data(nc, 1:c);
    nc=nc+shiftLINE;
end

%% BLANK SUBSTRACTION
NC=NC-blank;
replicates=replicates-blank;

%% PLOT
[m, n, d]=size(replicates);
t=t/60; %in minutes

for i=1:numREP
    figure(i)
    title('OD TECAN MEASUREMENT')
    hold on
    plot(t, NC(:, i), 'color', rgb('Black'), 'LineWidth', 2.5)
    legendInfo{1}=['NEGATIVE CONTROL'];
    for j=1:m
        plot(t,
            replicates(j, :, i), 'color', rgb(colori(j)), 'LineWidth', 2.5)
        legendInfo{j+1} = ['DILUTION NUM: ' num2str(j)]; % or
        whatever is appropriate
    end

    xlabel('TIME in minutes')
    ylabel('OD')
    grid on
    hold off
    legend(legendInfo)
end
startTRAINING=input('*Do you want to proceed with the mean
curves calculation? YES(1) NO(2) ');

if startTRAINING==1
    %% CUTTING TIME (90 MIN)
    minutesCUT=input('*Key the number in minutes for the
cutting (for example 90): '); %min

```

```

%%%CONTROL%%%
if (minutesCUT>t(end) || length(minutesCUT)>1 ||
isnumeric(minutesCUT)==0)
    %test
    error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cut=(minutesCUT+5)/5;
t=t(1:cut);
replicates=replicates(:,1:cut,:);
NC=NC(1:cut,:);

%% 2D MATRIX CREATION
%the first rows are the red curves in time (e.g. dilution
10^0)

curves=[];
shift=0;

for y=1:m % size(replicates)
    for i=1:d % numREP
        curves(i+shift,:)=replicates(y,:,i);
    end
    shift=shift+d;
end

% figure; %test plot
% a=1;
% b=d;
% for y=1:m
%     subplot(6,2,y)
%     plot(t,curves(a:b,:))
%     a=a+d;
%     b=b+d;
% end

% create a 2d matrix only with good data

    setCHOOSING=input('*Press 1 for automatical training set
creation,\n 2 for manual traning creation (only for
experts): ');

    if (setCHOOSING==1)
        int1=input('*Interval of replicates (for example [1
5]): '); % [4 6]

        %%%CONTROL%%%
        if (int1(2)<int1(1) || length(int1)~=2 ||
isnumeric(int1)==0)
            %test

```

```

error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

int2=input('*Interval of dilutions (for example [3
7]): '); %[1 7]

%%%CONTROL%%%
if (int2(2)<int2(1) || length(int2)~=2 ||
isnumeric(int2)==0)
%test
error('***Wrong number of input arguments***')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

goodDATA=[];
ii=(int2(2)+1)-int2(1);
a=(int2(1)-1)*d+int1(1);
b=(int2(1)-1)*d+int1(2);
meanNC=mean(NC(:,int1(1):int1(2))');

% d is the numer of replicates
for i=1:ii

    goodDATA=[goodDATA;
        curves(a:b, :)];
    a=a+d;
    b=b+d;
end
replicates4CURVE=((int1(2)+1)-int1(1))*ones(ii,1);

else if (setCHOOSING==2)

    goodDATA=input('*Key the curves for the training set
manually: (for example [curves(5:7, :);\n curves(12:14, :);\n
curves(19:21, :);\n curves(26:28, :);\n curves(33:35, :);\n
curves(40:42, :);\n curves(47:49, :)]:\n ');
    replicates4CURVE=input('*Digit the number of replicates
for every dilution of the training set: (for example [3;\n
3;\n 3;\n 3;\n 3;\n 3]:\n ');

    else error('\n***ERROR!!!!!!!!!!!! Input no valid!
Please run the program again!***\n\n')
    end
end

figure; %test plot
plot(t,goodDATA)
title('Good data')

%% DATA SUMMARY AND MEAN CURVES CALCULATION

```

```

% fit data from max to min with a sigmoidal function
[infoMIDPOINT] = sigmFITdata(goodDATA,t);

% Average of the different curves
meanCURVES=[];
stdMIDPOINT=[]; %standard deviation of the midpoints
from infoMIDPOINT
meanMIDPOINT=[]; % mean value of the midpoints from
infoMIDPOINT
linea1=1;
linea2=replicates4CURVE(1);

    for i=1:length(replicates4CURVE)

meanCURVES(i,:)=mean(goodDATA(linea1:linea2,:),1);

stdMIDPOINT(i,1)=std(infoMIDPOINT(linea1:linea2,1));

stdMIDPOINT(i,2)=std(infoMIDPOINT(linea1:linea2,2));

meanMIDPOINT(i,1)=mean(infoMIDPOINT(linea1:linea2,1));

meanMIDPOINT(i,2)=mean(infoMIDPOINT(linea1:linea2,2));
        linea1=linea1+replicates4CURVE(i);
        if (i<length(replicates4CURVE))
            linea2=linea2+replicates4CURVE(i+1);
        else
            linea2=linea2+replicates4CURVE(end);
        end
    end

% MIDPOINT for the mean curves
[infoMIDPOINTmean] = sigmFITdata(meanCURVES,t);

figure;

title('OD TECAN MEASUREMENT- MEAN CURVES')
grid on
hold on
plot(t, meanNC,'color',rgb('Black'),'LineWidth',2.5)
[mm,nn]=size(meanCURVES);
color=int2(1);
legendInfo{1}=['NEGATIVE CONTROL'];
for i=1:mm
plot(t,
meanCURVES(i,:), 'color', rgb(colori(color)), 'LineWidth', 2.5)
legendInfo{i+1} = ['DILUTION NUM: ' num2str(i)];
color=color+1;
end
legend(legendInfo)

```

```

xlabel('TIME in minutes')
ylabel('OD')
%plot(meanMIDPOINT(:,1),meanMIDPOINT(:,2),'g*')
herrorbar(infoMIDPOINTmean(:,1),infoMIDPOINTmean(:,2),stdMIDPOINT(:,1))
% the standard deviation reported on the midpoint of the
mean curves (x,y)
hold off

%% DATA SAVING
save('tecan12122014','t','goodDATA','NC','meanCURVES','mean
NC','infoMIDPOINT','infoMIDPOINTmean','stdMIDPOINT','blank'
,'colori','int1','int2')

else if startTRAINING==2
    error('*****END*****')
    else error('\n***ERROR!!!!!!! Input no valid! Please
run the program again!***\n\n')
    end
end
disp('*****END*****')

```

1.1 *sigmFITdata.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigmFITdata.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [infoMIDPOINT] = sigmFITdata(matrixDATA,t)

%%input: matrixDATA with different curves as rows and the
time as columns
[row,col]=size(matrixDATA);

% find the maximum point of every curve
maxCURVES=max(matrixDATA(:,3:end),[],2);

    for i=1:length(maxCURVES)

indexMAX(i)=max(find(matrixDATA(i,3:end)==maxCURVES(i)))+2;
    end

%% FITTING

```

```

% general sigmoidal equation

f = @(p,x) p(1)-((p(1) ./ (1 + exp(-(x-
p(2))/p(3)))));

opt=optimset('TolX',1e-
6,'MaxIter',300,'MaxFunEvals',1e2); %'Display','iter',

infoMIDPOINT=[];

for i=1:row

% consider only the descending part of the curves
sigmaDATA=matrixDATA(i,indexMAX(i):end);
tDATA=t(indexMAX(i):end);

midPOINT=(max(sigmaDATA)-min(sigmaDATA))/2;

global yo p
yo=midPOINT;
middleTIME=ceil(length(t)/2);
xo=t(middleTIME); %starting point for the F(xo)
research
xi=tDATA(1):0.01:tDATA(end); %increase points for
the fitting

p = nlinfit(tDATA,sigmaDATA,f,[0 20 50 5],opt);
%%[] is p0 starting coefficient

%[curve,goodness]=fit(tDATA',sigmaDATA','a + b ./ (1
+ exp(-(x-m)/s))','start',[0 20 50 5])

xx=fzero(@sigmoide,xo,opt);
% the function sigmoide is the fitting sigmoidal
curve shifted!
% x0=80; initial point for the research

infoMIDPOINT(i,1)=xx; %% x value
infoMIDPOINT(i,2)=midPOINT; %% y value

% test plot %find the max points and from there I start
with the sigmoidal fitting
% 3 means that I start to consider the maximums after 10
minutes, (at time 0 no max considerable)
% % % % % % % % % % % % figure; %test plot
% % % % % % % % % % % % hold on
% % % % % % % % % % % % grid on
% % % % % % % % % % % %
plot(t(3:end),matrixDATA(:,3:end))

```

```

% % % % % % % % % % % %
% % % % % % % % % % % %
plot(t(indexMAX),maxCURVES,'r*')
% % % % % % % % % % % %
% % % % % % % % % % % %           plot(tDATA,sigmaDATA)
% % % % % % % % % % % %
line(xi,f(p,xi),'color','r')
% % % % % % % % % % % %           plot(xx,yo,'k*')
% % % % % % % % % % % %           plot([xx xx],[0 yo])
% % % % % % % % % % % %           hold off
% % % % % % % % % % % %           refresh
% % % % % % % % % % % %           pause()

end

end
end

```

1.2 sigmoide.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigmoide.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [y] = sigmoide(x)
global yo p
y=p(1)-((p(1) ./ (1 + exp(-(x-p(2))/p(3)))));
y=y-yo; % the curve is shifted of yo.
end

```


A2odCFUcalibration.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% A2odCFUcalibration.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc
warning('off','all')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%new calibration tecan OD CFU/ML

%% DATA IMPORT
fprintf('****ATTENTION: Select the excel file with ONLY
spectrophotometer OD values!!!****\n\n ****PRESS Ctrl+C to
exit from the program****\n\n')
[FileName,PathName,FilterIndex] = uigetfile('*.');
%odCFUdata 12-12-2014
data= xlsread(strcat(PathName, FileName));
x=data(1,:);
y=data(2,:);

%od=[0 0.150 0.214 0.329 0.495 0.7]; %spectrophotometer
% y=[0 0 0 5.2*10^7 6.5*10^7 5.0*10^7 6.3*10^7 6.4*10^7
5.5*10^7 7.3*10^7 6.6*10^7 7.5*10^7 1.64*10^8 1.47*10^8
1.46*10^8 3.8*10^8 2.5*10^8];
% x=[0 0 0 0.150 0.150 0.150 0.214 0.214 0.214 0.329 0.329
0.329 0.495 0.495 0.495 0.7 0.7];

%% CALIBRATION
choice=input('*Press 1 for automatical calibration, 2 for
manual calibration (only for experts): ');

if choice==1

fittedmodel=fit(x,y,'poly1','Robust','LAR')
P=coeffvalues(fittedmodel);

% % % extract the *interpolated* curve from the figure
% % X=get(h,'XData');
% % Y=get(h,'YData');
else if choice==2
    cftool(x,y)
    %%% save here the cftool results!!
    disp('*It is advisable to use a LINEAR function!')
    disp('*PLEASE save the cftool fitting session\n
(from the maximized cftool window,\n choose "Save to
Workspace" from the "fit" menu\n then click OK');
end
```

```

        disp('*Press a button to continue after the
fitting!')
        pause()
        fittedmodel
        goodness
        output
        P=coeffvalues(fittedmodel);

        else error('\n***ERROR!!!! INPUT NO VALID!! Please run
the program again!!*** \n\n')
        end
end
figure;
h=plot(fittedmodel,x,y);
grid on
legend('experiment data','fitting equation','FontSize',18)
xlabel('OD','FontSize',18)
ylabel('CFU/ml','FontSize',18)

%% CFU/mL DETERMINATION
c=input('\n\n*Input CFU/mL :\n for MANUAL press 1;\n with
GINPUT press 2;\n from the STARTING OD INFO press 3: ');
if c==1
CFUml=input('*Key the CFU/ml (for example 6*10^7 or
60000000): '); %6*10^7
else if c==2
    [startOD, CFUml]=ginput(1);
    else if c==3
        startOD=input('*Write the starting
SPECTROPHOTOMETER OD value (es 0.2): ')

        %%CONTROL%%
        if (startOD<0 || startOD>1 || length(startOD)>1 ||
isnumeric(startOD)==0)
            error('***Wrong number of input arguments***')
        end
        %%%%%%%%%%%
        CFUml=fittedmodel(startOD);
        else error('\n***ERROR!!!! INPUT NO VALID!! Please
run the program again!!*** \n\n')
        end
    end
end
end

save('odCFUcalibration','P','fittedmodel','CFUml')
disp('*****END*****')

```

3 *A3mainFIT.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      A3mainFIT.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all
close all
clc
warning('off','all')
%%%%%%%%%% FITTING MICROPLATE READER DATA TO THE MODEL

%% DATA LOADING
load('tecan12122014')
warning('off','all')

wellCELLS=0.18; %%qimron protocol
wellPHAGE=0.02;

%% EQUATION FOR OD-CFU/mL CALIBRATION

choice=input('*Do you use the previous OD-CFU/ml
calibration? Digit 1 (YES), 2 (NO): ');

if choice==2
    fprintf(' \n ***Please run the A2odCFUcalibration
program!!*** \n')
else if choice==1
    load('odCFUcalibration.mat')
    [mm,nn]=size(meanCURVES);
    p1 =P(1);
    p2 =P(2);
PFUml=input('*Key the PFU/ml for the HIGHEST concentration
considered(for example 3*10^10 or 3000000000): ');
%3*10^10
%PFUml=3*10^10;
MOI=(PFUml*wellPHAGE)/(CFUml*wellCELLS); %for the highest
concentration (lowest dilution)
conversionSPECTRtecan=3.4; %OD difference between the
spectrophotometer and the tecan

od1=(meanCURVES>=0.02).*meanCURVES*conversionSPECTRtecan;
od2=(meanNC>=0.02).*meanNC*conversionSPECTRtecan;
for i=1:mm
cfuMEANcurves(i,:)=(wellCELLS+wellPHAGE)*fittedmodel(od1(i,
:));
end
cfuMEANcurves=(cfuMEANcurves>=0).*cfuMEANcurves;
```

```

cfuMEANnc=(wellCELLS+wellPHAGE)*fittedmodel(od2);
cfuMEANnc=(cfuMEANnc>=0).*cfuMEANnc;

%% GROWTH RATE DETERMINATION
figure;
plot(t,cfuMEANnc)
title('Experimental growth rate')
grid on
xlabel('TIME in minutes')
ylabel('CFU')
points=input('*Growth rate: choose two points
automatically, the first and the last, (press 1)\n or
manually, with ginput (press 2): ');

if points==1
t0=t(1);
N0=cfuMEANnc(1);%ginput(1);
t1=t(end);
N1=cfuMEANnc(end);%ginput(1);
else if points==2
[t0,N0]=ginput(1);
[t1,N1]=ginput(1);
else
fprintf('*** ERROR!! No VALID INPUT!!***\n')
end
end

growthRATE = ((log10(N1)-log10(N0))*2.303)/(t1 - t0) %in
min^-1
grow1=growthRATE-(growthRATE/5);
grow2=growthRATE+(growthRATE/5);

%% MAIN, FIRST STEP: parameter optimization for each
dilation

global hystory

figure(100);

    grid on
    hold on

    color=int2(1);
for i=1:mm
plot(t,
cfuMEANncurves(i,:), '*','color',rgb(colori(color)), 'LineWidth
h',1.5)

legendInfo{i} = ['DILUTION NUM: ' num2str(i)];
color=color+1;
end

```

```

str2 = sprintf('OD TECAN MEASUREMENT 12/12/2014 CFU/ML=
%1.2e PFU/ML=%1.2e', CFUml, PFUml);
title(str2)
legend(legendInfo)
xlabel('TIME in minutes')
ylabel('CFU')
hold on

%% solve the model system

opt=optimset('TolX',1.0e-
5,'MaxIter',15,'MaxFunEvals',1e4);%,'OutputFcn',@PlotIterat
es); %'Display','iter',

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Initial conditions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
esp=int2(1)-1;
for i=1:mm
    S0=cfuMEANcurves(i,1); %CFU
    V0=PFUml*wellPHAGE/10^esp; %PFU
    I0=0;
    %ratio=1/(10^6);
    R0=0;%S0*ratio;

    hystoryMATRIX(:, :, i)=[S0; R0; I0; V0];
    esp=esp+1;
end

integrationTIME=[0 t(end)];
tspan=integrationTIME;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%GLOBAL
SEARCH%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%load('mainFITA3b')

color=int2(1);
for k=1:mm

hystory=hystoryMATRIX(:, :, k);

if k==1
a=growthRATE; % growth rate: up and down
f=0; % mutation rate of bacteria
K=6.1055; % latent period: left and right
m=0.0001038; % phage decay rate
b=0.280019e-09; % binding rate of phage: amplitude of the
curve
h=150.1286; % burst size
P0=[a b h K m]; % starting value for the parameters

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%LOWER AND UPPER BOUNDS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LB=[growl 0 0 0 0.0001];

```

```

% UB=[grow2 1e-9 500 30 0.1];
LB=[grow1 0 0 0 0.00000001];
UB=[grow2 1e-9 800 30 0.01];

%NUMBER OF VALUES TAKEN IN CONSIDERATION FOR EACH LOWER AND
UPPER BOUND%
comNUMpar=[3 4 3 4 2]; %288 starting points 144

else
P0=ParNEW(1,:);
LB=P0-P0/5;
UB=P0+P0/5;
LB(1)=grow1;%P0(1)-P0(1)/10; %%the interval for the growth
rate
UB(1)=grow2;%P0(1)+P0(1)/10;
comNUMpar=[3 3 3 3 1]; %108 starting points 81
end

for i=1:length(LB)
    comb{i}=linspace(LB(i),UB(i),comNUMpar(i));
end

[X,Y,Z,W,Q] =
ndgrid(comb{1},comb{2},comb{3},comb{4},comb{5});
startP=[X(:),Y(:),Z(:),W(:),Q(:)];
custpts = CustomStartPointSet(startP);

problem =
createOptimProblem('lsqcurvefit','x0',P0,'objective',@helpe
r,'lb',LB,'ub',UB,'xdata',t,'ydata',cfuMEANcurves(k,:),'opt
ions',opt);
%gs = GlobalSearch('Display','iter','MaxTime',60);
ms =
MultiStart('UseParallel','always','StartPointsToRun','bound
s','Display','iter','PlotFcns',@gsplotbestf);%,'MaxTime',60
);%
[PmultiBest,errormultiBest,exitflag] =
run(ms,problem,custpts)
ParNEW(k,:)=PmultiBest;

%%% delay model solution%%%
% 1. define the state
state=+1;
% 2. register this function as an event function
options = ddeset('Events',@MyEventFunction);
% 3. start a stopwatch timer, if you already use one,
define a new one: tic(ticID)
tic;

sol=dde23('cairns',ParNEW(k,4),hystory,tspan,options,ParNEW
(k,:));

```

```

cells=sol.y(1,:)+sol.y(3,:);
cells=(cells>=0).*cells;
% phage=sol.y(3, :)' +sol.y(4, :)' ;
time=sol.x;
figure(100)
hold on
plot(time,cells,'color',rgb(colori(color)),'LineWidth',2);
%tint,phage);
color=color+1;
end

save('mainFITA3b','cfuMEANcurves','PFUml','wellCELLS','well
PHAGE','hystoryMATRIX','growthRATE','ParNEW','LB','UB','con
versionSPECTRtecan')

else
    fprintf('***ATTENTION! ERROR: NO VALID INPUT.
Please run the programe again!')
end
end

disp('*****END*****')

```

3.1 *helper.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      helper.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [cells2,VALUE]=helper(Par,t)
global hystory
tt=t;
%% 1. define the state
state=+1;
%% 2. register this function as an event function
options = odeset('Events',@MyEventFunction);

%% 3. start a stopwatch timer, if you already use one,
define a new one: tic(ticID)
tic;

%% Run the model

```

```

sol=dde23('cairns',Par(4),hystory,tt,options,Par);

load('v.mat')
if VALUE>0
    load('defaultDDEsolERR')

end

time=sol.x;
cells=sol.y(1,:)+sol.y(3,:);
cells2=interp1(time,cells,tt);
cells2=(cells2>=0).*cells2;

end

```

3.2 *cairns.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   cairns.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function v = cairns(t,y,Z,Par)
S=y(1);
R=y(2);
I=y(3);
V=y(4);

a=Par(1);
f=0;
b=Par(2);
h=Par(3);
m=Par(5);

%% delay
ylag=Z;
v=zeros(4,1);

v(1)=a*S-f*S-b*S*V;
v(2)=a*R+f*S;
v(3)=b*S*V-b*ylag(1)*ylag(4);
v(4)=h*b*ylag(1)*ylag(4)-b*S*V-m*V;

end

```


3.3 *MyEventFunction.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MyEventFunction.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Define the event function
function [VALUE, ISTERMINAL, DIRECTION] =
MyEventFunction(t, y, Z, state)
%%The event function stops the intergration is VALUE == 0
and
%%ISTERMINAL==1

%%a. Define the timeout in seconds
TimeOut = 3; %sec
%%
%%b. The solver runs until this VALUE is negative (does not
change the sign)
    VALUE = toc-TimeOut;
    save('v','VALUE')

%%c. The function should terminate the execution, so
ISTERMINAL = 1;

%%d. The direction does not matter
DIRECTION = 0;

end
```

A4mainCROSS.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      A4mainCROSS.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Parameter optimization

clear all
close all
clc
warning('off','all')
%%%%%%%%a=0.017; growth rate
%%%%%%%%m=0.0001

%% DATA LOADING
choice=input('Do you use the previous fitting parameters?
Digit 1 (YES), 2 (NO): ');

if choice==2
    fprintf(' \n ***Please run the A3mainFIT program!!***
\n')
else if choice==1
    load('odCFUcalibration.mat')
    load('tecan12122014.mat')
    load('mainFITA3b')
    PParNEW=ParNEW;

global hystory trainHYSTORY testHYSTORY K

Kvect=PParNEW(:,5);

Ydata=cfuMEANcurves;

figure(100);

    grid on
    hold on

[mm,nn]=size(meanCURVES);
color=int2(1);
for i=1:mm
    plot(t,
cfuMEANcurves(i,:), '*','color',rgb(colori(color)), 'LineWidt
h',1.5)

legendInfo{i} = ['DILUTION NUM: ' num2str(i)];
```

```

color=color+1;
end
    str2 = sprintf(' COMPARISON: OD MEASUREMENT-DELAY MODELS
(CROSS VALIDATION) CFU/ML= %1.2e PFU/ML=%1.2e', CFUml,
PFUml);
title(str2)

xlabel('TIME in minutes')
ylabel('CFU')

    opt=optimset('TolX',1.0e-
5,'MaxIter',15,'Display','iter','MaxFunEvals',1e4);%,'OutputFcn',@PlotIterates);

    Par=[];
hystory=hystoryMATRIX;
integrationTIME=[0 t(end)];
tspan=integrationTIME;

%% CROSS VALIDATION

x_sample=t;
y_sample=cfuMEANcurves;
n_sample=size(y_sample,1);
K=n_sample;
index=crossvalind('kfold',n_sample,K)

ParMATRIX=[];
error=[];
CellMATRIX=[];
tic
for i=1:K
    %iTESTsample=find(index==i); % i è il numero del
campione preso come test in leave one out!
    test = (index==index(i)); train = ~test;
    %train_x = x_sample(train,:);
    train_y = y_sample(train,:);
    trainHYSTORY=hystory(:, :, train);
    ParINIT=median(PParNEW(train,:)) % Starting parameter set:
median of the training set parameters.

    %test_x = x_sample(test,:);
    test_y = y_sample(test,:);
    testHYSTORY=hystory(:, :, test);

[ParNEW, resnorm, residual, exitflag, output] =
lsqcurvefit(@helper4, ParINIT, t, train_y, LB, UB, opt);
ParMATRIX(index(i), :)=ParNEW
sol=dde23('cairns', ParMATRIX(index(i), 4), testHYSTORY, tspan,
[], ParMATRIX(index(i), :));

z=interp1(sol.x, sol.y(1, :), t);

```

```

error(index(i))=sum((y_sample(index(i),:)-z).^2)
tint = linspace(integrationTIME(1), integrationTIME(2));
yint = deval(sol,tint);
cells=yint(1,:);
CellMATRIX(index(i),:)=cells';

end
toc

MINerror=min(error);
optimalPAR=ParMATRIX(find(error==MINerror),:);
save('crossRESULTS2median','ParMATRIX','optimalPAR','error'
,'hystory','legendInfo')

%%%%%%%%%% after cross validation %%%%%%%%%%%

%% PLOT
load('crossRESULTS2median')

ParNEWmainA4CROSS=optimalPAR;

tint = linspace(integrationTIME(1),
integrationTIME(2),100);
cells=[];

for i=1:K
sol=dde23('cairns',ParNEWmainA4CROSS(4),hystory(:,:,i),tspa
n,[],ParNEWmainA4CROSS);
yint = deval(sol,tint);
cells(:,i)=yint(1,:)+yint(3,:);
cells=(cells>=0).*cells;
NORMcells(:,i)=cells(:,i)/cells(1,i);
end

color=int2(1);
figure(100)
hold on
l=length(legendInfo);
for i=1:mm
plot(tint,
cells(:,i),'color',rgb(colori(color)),'LineWidth',1.5)
legendInfo{i+1} = ['MODEL-DILUTION NUM: ' num2str(i)];
color=color+1;
end
xlabel('TIME in minutes')
ylabel('CFU')
legend(legendInfo)
f=figure(100);
saveas(f,'crossMEDIAN.fig')

else

```

```

        fprintf('***ATTENTION! ERROR: NO VALID INPUT.
Please run the programe again!')
    end
end

disp('*****END*****')

```

4.1 helper4.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  helper4.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function cells=helper4(Par,t)
global trainHYSTORY testHYSTORY K
tt=t;

for i=1:(K-1)
sol=dde23('cairns',Par(4),trainHYSTORY(:, :, i),tt,[],Par);
tint = linspace(tt(1),tt(end), length(tt));
yint = deval(sol,tint);
cells(i, :)=yint(1, :)' + yint(3, :)' ;
end

end

```

5 *A5mainPREDICTION.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      A5mainPREDICTION.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear all
close all
clc
```

```
%% DATA LOADING
choice=input('Do you use the previous optimized fitting
parameters? Digit 1 (YES), 2 (NO): ');
```

```
if choice==2
    fprintf(' \n ***Please run the A3mainFITprogram!!***
\n')
else if choice==1
    load('odCFUcalibration.mat')
    load('tecan12122014.mat')
    load('mainFITA3b')
    load('crossRESULTS2median')
```

```
[mm,nn]=size(cfumeANcurves);
integrationTIME=[0 t(end)];
tspan=integrationTIME;
ParNEWmainA4CROSS=optimalPAR;
tint = linspace(integrationTIME(1),
integrationTIME(2),100);
cells=[];
```

```
%% DATA IMPORT
```

```
fprintf('****ATTENTION: Select the excel file with ONLY the
tecan OD values!!!****\n\n')
[FileName,PathName,FilterIndex] = uigetfile('*.xlsx');
```

```
unknown= xlsread(strcat(PathName, FileName));
unknown=unknown-blank;
[MM,NN]=size(unknown);
```

```
%% PREVIOUS MODEL FITTING
openfig('crossMEDIAN.fig')
```

```
for i=1:mm
```

```

%plot(t, cfuMEANcurves(i,:), '*','color',rgb(colori(i)))
%%%data
sol=dde23('cairns',ParNEWmainA4CROSS(4),hystory(:, :,i),tspa
n,[],ParNEWmainA4CROSS);
yint = deval(sol,tint);
cells(:,i)=yint(1,:)+yint(3,:);
NORMcells(:,i)=cells(:,i)/cells(1,i);
%plot(tint,
cells(:,i),'color',rgb(colori(i)),'LineWidth',1.5);
%%%model

end

%%% NORMcells and cfuUNKNOWN must have the same length
if(NN>=nn)
unknown=unknown(:,1:nn); %cutting, in order to avoid
artefacts
else
tt=0:300:300*(NN-1);
for j=1:MM
unknown(j,:)=interp1(tt,unknown(j,:),t);
end
end

%% OD-CFU/mL CALIBRATION
conversionSPECTRtecan=3.4;
p1 = P(1);
p2 = P(2);

od=(unknown>=0.02).*unknown*conversionSPECTRtecan;
for i=1:MM
cfuUNKNOWN(i,:)=(wellCELLS+wellPHAGE)*fittedmodel(od(i,:));
end;
% cfuUNKNOWN=(wellCELLS+wellPHAGE)*(p1*od + p2); %in CFU

cfuUNKNOWN=(cfuUNKNOWN>=0).*cfuUNKNOWN;
cfuUNKNOWN2=cfuUNKNOWN;

%% NORMALIZATION

for i=1:MM
cfuUNKNOWN(i,:)=cfuUNKNOWN(i,:)./cfuUNKNOWN(i,1);
end

%load('tecanFIT2results')

figure;

title('COMPARISON BETWEEN THE UNKNOWN DILUTION''S CURVE AND
THE MODEL AFTER NORMALIZATION')
xlabel('TIME in minutes')

```

```

ylabel('Num cells normalized with the start CFU/ml value')

color=int2(1);
for i=1:mm
plot(tint,
NORMcells(:,i),'color',rgb(colori(color)),'LineWidth',1.5);
hold on
grid on
color=color+1;
end
plot(t,cfuUNKNOWN,'k','LineWidth',1.5);

%% PREDICTION

% %%%%%%%%%%%%% Find the best coefficient of
correlation %%%%%%%%%%%%%

coeff=[];

pfu=squeeze(hystory(4,1,:)/wellPHAGE);

for j=1:MM

    cfuUNKNOWNinterp(j,:)=interp1(t,cfuUNKNOWN(j,:),tint);
%%interpolation with 100 points-the same length of
NORMcells
    for i=1:mm
        [R,p]=corrcoef(cfuUNKNOWNinterp(j,:),NORMcells(:,i));
        coeff(j,i)=R(1,2);
        pVALUE(j,i)=p(1,2);
    end
end
maxCORR=max(coeff,[],2);
pVALUE

for j=1:MM
legendINFO=[];
indexMAXcorr(j)=find(coeff(j,:)==maxCORR(j));
pVALUEmaxCORR(j)=pVALUE(j,indexMAXcorr(j));
cc=indexMAXcorr(j)+(int2(1)-1);
if (maxCORR(j)>=0.67 && pVALUEmaxCORR(j)<0.05)
    count=j+2;
figure(count)
sampl = sprintf('%d',j);
    hold on
grid on
% plot(tint,cfuUNKNOWNinterp(j,:),'k','LineWidth',2)
%
plot(tint,NORMcells(:,indexMAXcorr(j)),'color',rgb(colori(c
c)),'LineWidth',2);
plot(t,cfuUNKNOWN2(j,:),'k','LineWidth',2)

```



```

plot(tint,cells(:,indexMAXcorr(j)), 'color',rgb(colori(cc)),
'LineWidth',2);
str = sprintf('MODEL PREDICTION FOR THE UNKNOWN DILUTION:
SAMPLE NUM %s', sampl);
title(str)
xlabel('TIME in minutes')
ylabel('Num cells in CFU/ml')
legendINFO{1} = ['UNKNOWN DILUTION'];

PFUml1=pfu(indexMAXcorr(j));
CFUml=cfuUNKNOWN2(j,1)/wellCELLS;
MOI1=(PFUml1*wellPHAGE)/cfuUNKNOWN2(j,1);
results1 = sprintf('%1.2e',PFUml1);
% stringal = sprintf('PREDICTION of the pfu/ML: %s',
results1);
% (stringal)
legendINFO{2} = ['PREDICTION of the pfu/ML: ' results1];

fprintf('*****PREDICTION FOR THE SAMPLE NUM
%d!*****\n\n',j)

secondMAX= max((coeff(j,:)<maxCORR(j)).*coeff(j,:));
if (abs(secondMAX-maxCORR(j))<=0.04)
    indexSECMAXcorr=find(coeff(j,)==secondMAX);
    ccc=indexSECMAXcorr+(int2(1)-1);

plot(tint,cells(:,indexSECMAXcorr), 'color',rgb(colori(ccc))
,'LineWidth',2);
    PFUml2=pfu(indexSECMAXcorr);
    MOI2=(PFUml2*wellPHAGE)/cfuUNKNOWN2(j,1);
    results2 = sprintf('%1.2e',PFUml2);
%    stringa2 = sprintf('SECOND PREDICTION of the pfu/ML:
%s', results2);
%    legappend(stringa2)
    legendINFO{3} = ['SECOND PREDICTION of the pfu/ML: '
results2];

    fprintf('PFU/ml prediction: %1.2e - %1.2e
\n',min(PFUml1,PFUml2),max(PFUml1,PFUml2))
    fprintf('CFU/ml prediction: %1.2e \n',CFUml)
    fprintf('MOI prediction: %1.2e - %1.2e
\n\n\n\n',min(MOI1,MOI2),max(MOI1,MOI2))

else fprintf('PFU/ml prediction: %1.2e \n',PFUml1)
    fprintf('CFU/ml prediction: %1.2e \n',CFUml)
    fprintf('MOI prediction: %1.2e \n\n\n\n',MOI1)

end
legend(legendINFO)
    if (maxCORR(j)<0.75 && maxCORR(j)>=0.67)

```

```

        fprintf('*ATTENTION! THE PREDICTION COULD BE NOT
PRECISE FOR THIS DATA: SAMPLE NUM %d!*\n\n\n\n',j)
        end

else fprintf('*****ERROR!! NO PREDICTION FOR THIS
DATA: SAMPLE NUM %d!*****\n\n\n\n',j)
end
pause()
end

        else
                fprintf('***ATTENTION! ERROR: NO VALID INPUT.
Please run the programe again!')
                end
end
end

```

6 Other MATLAB code used:

6.1 rgb.m.

(<http://www.mathworks.com/matlabcentral/fileexchange/24497-rgb-triple-of-color-name--version-2/content/rgb.m>)

6.2 herrorbar.m

(<http://www.mathworks.com/matlabcentral/fileexchange/3963-herrorbar/content/herrorbar.m>)

References

Adams, M. H. (1959). Bacteriophages. Interscience Publishers, Inc., New York.

Alcamo Edwards I. (2001). Fundamentals of microbiology. Jones & Bartlett Pub. Sixth edition.

Bull, J. J., and Gill, J. J. (2014). The habits of highly effective phages: population dynamics as a framework for identifying therapeutic phages. *Front. Microbiol.* 5:618. doi: 10.3389/fmicb.2014.00618

Cairns, B. J., Timms, A. R., Jansen, V. A. A., Connerton, I. F., and Payne, R. J. H. (2009). Quantitative Models of In Vitro Bacteriophage–Host Dynamics and Their Application to Phage Therapy. *PLoS Pathogens*, 5(1), e1000253. doi:10.1371/journal.ppat.1000253

Carlson, J. C., Badran, A. H., Guggiana-Nilo, D. A. , and Liu, D. R. (2014). Negative Selection and Stringency Modulation in Phage-Assisted Continuous Evolution. *Nature Chemical Biology*. Epub 2014 Feb 2.

Chhibber, S., and Kumari, S. (2012). Application of Therapeutic Phages in Medicine, Bacteriophages. Dr. Ipek Kurtboke (Ed.), ISBN: 978-953-51-0272-4, InTech, DOI: 10.5772/34296. Available from:
<http://www.intechopen.com/books/bacteriophages/therapeutic-bacteriophages>

Cattoën, C. (2003). Bacteria-Phage mathematical model applied to the cheese industry. Biomathematics Project, Massey University.

D'Hérelle, F. (1917). Sur un microbe invisible antagoniste des bacilles dysentériques. *Comptes rendus Acad Sci Paris*. 165, pp. 373–5. On an invisible microbe antagonistic toward dysenteric bacilli, pp. brief note by Mr. F. D'Herelle (2007). *Research in microbiology* Vol.158, No.7, pp. 553–4.

Delbruck, M. (1940). Adsorption of bacteriophage under various physiological conditions of the host. *J. Gen. Physiol.* 23:631–642.

Delbruck, M. (1940). The growth of bacteriophage and lysis of the host. *J. Gen. Physiol.* 23:643–660.

Delbruck, M. (1942) Bacterial viruses (bacteriophages). *Adv. Enzymol.* 2, 1–32.

Dulbecco, R., and Vogt, M. (1953). Some problems of animal virology as studied by the plaque technique. *Cold Spring Harbor Symp. Quant. Biol.*, 18, 273-279

Ellis, E. L., and Delbruck, M. (1939). The growth of bacteriophage. *J. Gen. Physiol.* 22: 365–384.

Esvelt, K. M., Carlson, J.C., and Liu, D. R. (2011). A System for the Continuous Directed Evolution of Biomolecules. *Nature*. 2011 Apr 28; 472(7344):499-503. Epub 2011 Apr 10.

Gadagkar, R., and Gopinathan, K. P. (1980). Bacteriophage burst size during multiple infections. *J. Biosci.*, Vol. 2, Number 3, pp. 253-259.

Hall G. B., Acar H., Nandipati A., and Barlow M. (2013). Growth Rates Made Easy. *Mol. Biol. Evol.* 31(1):232–238 doi:10.1093/molbev/mst187

Hanlon, G.W. (2007). Bacteriophages: an appraisal of their role in the treatment of bacterial infections. *Int J Antimicrob Agents* 30: 118–128.

Hankin, E. H. (1896). L'action bactericide des eaux de la Jumna et du Gange sur le vibron du cholera (in French). *Annales de l'Institut Pasteur* Vol. 10, pp. 511–523.

Husimi, Y. (1989). Selection and evolution of Bacteriophages in cellstat. *Adv. Biophys.* Vol. 25, pp. 1-43.

Hyman P., Abedon S. T. (2009). Practical Methods for Determining Phage Growth Parameters. *Bacteriophages. Methods in Molecular Biology™* Volume 501, 2009, pp 175-202.

Inal, J. (2003). Phage therapy: a reappraisal of bacteriophages as antibiotics. *Arch. Immunol. Ther. Exp.* 51, 237–244.

Jaramillo, A. (2015). Jaramillo Lab, Available from: <http://jaramillolab.issb.genopole.fr/display/sbsite/Jaramillo+Lab>. Last accessed 21th Jan 2015.

Kasman, M. L., Kasman, A., Westwater, C., Dolan, J., Schmidt, J. M. and Norris, J. S. (2002). Overcoming the phage replication threshold: a mathematical model with implications for phage therapy. *J. Virol.* 76:5557–5564.

Krueger, A. P. (1931). *J. Gen. Physiol.*, 1931, 14, 493.

Krueger, A. P. and Northrop J. H. (1930). The kinetics of the bacterium-bacteriophage reaction. *J Gen Physiol.* 1930 Nov 20; 14(2): 223–254.

Larson, S. (1931). The shrinkage of the coefficient of multiple correlation. *J. Educat. Psychol.*, 22:45–55,1931.

Levin, B. R., and Bull, J. J. (1996). Phage therapy revisited: the population biology of a bacterial infection and its treatment with bacteriophage and antibiotics. *Am. Nat.* 147:881–898.

Levin, B. R., Stewart, F. M., and Chao, L. (1977). Resource-limited growth, competition, and predation: a model and experimental studies with bacteria and bacteriophage. *Am. Nat.* 111:3–24.

Matsuzaki, S., Rashed, M., Uchiyama, J., Sakurai, S., Ujihara, T., Kuroda, M., Ikeuchi, M., Tani, T., Fujieda, M., Wakiguchi, H., and Imai, S. (2005). Bacteriophage therapy: a revitalized

therapy against bacterial infectious diseases. *J. Infect. Chemother.* 11, 211–219.

Merril, C. R., Scholl, D., and Adhya, S. L. (2003). The prospect for bacteriophage therapy in Western medicine. *Nat Rev Drug Discov* 2: 489–497.

Middleboe, M., (2000). Bacterial growth and marine virus-host dynamics. *Microb. Ecol.* 40:114–124.

Mudgal, P., Breidt, F., Lubkin, S. R., and Sandeep, K. P. (2006). Quantifying the Significance of Phage Attack on Starter Cultures: a Mechanistic Model for Population Dynamics of Phage and Their Hosts Isolated from Fermenting Sauerkraut . *Applied and Environmental Microbiology*, 72(6), 3908–3915. doi:10.1128/AEM.02429-05

Payne, J. H., and Jansen, A. A. (2000). Understanding bacteriophage therapy as a density dependent kinetic process. *J. Theor. Biol.* 208:37–48.

Payne, R. J. H. , Phil, D., & Jansen, V. A. A. (2000). Phage therapy: the peculiar kinetics of self-replicating pharmaceuticals. *Clinical pharmacology and therapeutics*, 68(3), 225-30. 10.1067/mcp.2000.109520

Qimron U., Marintcheva B., Tabor S., and Richardson C.C. (2006). Genomewide screens for *Escherichia coli* genes affecting growth of T7 bacteriophage. *Proc Natl Acad Sci USA* 103: 19039–19044

Qimron, U., Tabor, S., and Richardson, C.C. (2010). New Details about Bacteriophage T7-Host Interactions. *Microbe* Vol. 5, N°3.

Racaniello, V. (2014). Virology question of the week: What matters more, multiplicity of infection or virus concentration? Available from: <http://www.virology.ws/2014/05/06/virology-question-of-the-week-what-matters-more-multiplicity-of-infection-or-virus-concentration/> Last accessed 28th Jan 2015.

Refaeilzadeh P., Tang L. and Liu H. (2009). Cross Validation. In Encyclopedia of Database Systems, Editors: M. Tamer Özsu and Ling Liu. Springer.

Sandeep, K. (2006). Bacteriophage precision drug against bacterial infections. Current Science, Vol.90, pp 361 – 363, ISSN 0011-3891

Shabram, P., and Aguilar-Cordova, E. (2000). Multiplicity of infection/multiplicity of confusion. Mol. Ther., 2, pp. 420–421

Wang, I. N. (2006). Lysis Timing and Bacteriophage Fitness. Genetics. 172(1): 17–26. doi: 10.1534/genetics.105.045922