

Analyzing Bacterial Conjugation with Graphical Models: A Model Comparison Approach

by

Nat Kendal-Freedman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2024

© Nat Kendal-Freedman 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The entirety of this thesis was written by Nat Kendal-Freedman under the supervision of Dr. Brian Ingalls.

The work presented in Chapters 3-6 of this thesis is the product of a joint project in preparation with Joseph Meleshko, Aaron Yip, and Brian Ingalls. Section 3.1 includes an overview of the experimental work done by Aaron Yip. The model formulation and implementation described in Chapter 3 was conceptualized by Nat Kendal-Freedman, with input from Joseph Meleshko. Nat Kendal-Freedman wrote the python implementation of the code described in Section 4.1; Joseph Meleshko wrote the python implementation of the code described in Sections 4.2-4.4. Nat Kendal-Freedman both developed the model versions and analyzed the results in Chapter 5. Joseph Meleshko contributed to the computation of the results.

Abstract

Conjugation is a mechanism for horizontal gene transfer that allows microbes to share genetic material with nearby cells. It plays an important role in the spread of antibiotic resistance in bacteria and is used as a tool for genetic engineering. Understanding which factors affect conjugation frequency is an ongoing challenge due to the stochastic nature of cell-cell interactions. In this thesis, we present a proof of concept of a model comparison approach for analyzing experimental data of bacterial conjugation. We develop a Bayesian network structure to model the interactions within a single experimental trial. We model different versions of biological mechanisms by assigning different conditional probability distributions to those structures. Identifying distributions that predict events consistent with the experimental results provides insight into the mechanisms governing conjugation. We compare 12 model variations for each of 6 experimental trials. Our results suggest that individual cell features and contact quality both impact the likelihood of conjugation. We also provide insight into the length of the delays involved in conjugation. These results are consistent when compared across multiple trials and metrics.

Acknowledgments

This thesis could not have been completed without the support of many people.

First, I am grateful for the guidance, encouragement, and abundance of thought-provoking questions provided by my supervisor, Dr. Brian Ingalls. Most of all, thank you for supporting my many side-quests and providing me with the freedom to direct my research. I would also like to thank my undergraduate supervisors, Dr. Jay Newby and Dr. Thomas Hillen, for their continued support and advice. To my committee members, Dr. Matthew Scott and Dr. Chris Bauch, thank you for taking the time to read and provide feedback on this thesis.

My friends and family have been a great source of strength, inspiration, and encouragement. In particular, I would like to thank the following individuals:

Joseph Meleshko – Collaborating with you has been a highlight of my degree, but it is your friendship I value most. Thank you for your insightfulness, generosity, and candor.

Aiden Huffman – A great deal of the work that went into this thesis was completed while on silent calls with you. Thank you for your companionship and for reminding me not to take myself too seriously.

Atiyeh Ahmadi – I cannot count the number of teas we shared over the past two years. Thank you for bringing some much-needed positivity into my life.

Noah Gergel – You have always been there for me when I needed a friend. Thank you for reminding me to prioritize myself.

Ian DeHaan – Words cannot express how grateful I am for your unwavering friendship. Thank you for believing in me and for helping me fulfill my destiny of being the world's first pure applied mathematician.

Finally, thank you to my family – especially my parents and my brother – for your love and support throughout this journey.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgments	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Horizontal Gene Transfer	2
1.1.1 Plasmids & Conjugation	3
1.2 Relevance	4
1.3 Experimental Approaches	5
1.3.1 Approaches for the Single-Cell Level	6
1.3.2 Previous Experimental Studies	7
1.4 Previous Modeling Approaches	9

2	Probabilistic Graphical Models	10
2.1	Bayesian Networks	11
2.2	Conditional Independence in a Bayesian Network	13
2.3	Querying a Bayesian Network	17
3	Model Formulation	19
3.1	Experimental Setup	19
3.1.1	Image Processing	22
3.1.2	Synchrony	23
3.2	Model Setup	24
3.2.1	Assumptions	24
3.2.2	Handling Inconsistent Data	27
3.2.3	Graph Structure	29
3.2.4	Modeling Decisions	32
3.2.5	Conditional Probability Distributions & Edge Weights	34
3.3	Model Comparison	38
3.3.1	Model Ranking Systems	40
4	Model Implementation	42
4.1	Initial Calculations and Data Processing	42
4.2	Building the Graph Structure	43
4.3	Query Setup	44
4.4	Query Evaluation	47
5	Models & Results	52
5.1	Models	52
5.1.1	Expression Delay	52
5.1.2	Maturation Delay	55
5.1.3	Contact Quality	57
5.2	Results	59
5.2.1	Rankings	61

6	Conclusions & Future Plans	67
6.1	Future Plans	67
	References	69

List of Figures

1.1	Diagram of Conjugation	3
1.2	Image of Microfluidic Traps	6
1.3	Delays in the Conjugative Process	7
2.1	Graph Representation of Medical Diagnosis Example	12
2.2	Trails with Three Nodes	14
2.3	Effect of Downstream Evidence on a Trail	16
3.1	Diagram of a Microfluidic Trap	20
3.2	Sample Frame from an Experiment	21
3.3	Depiction of Alterations to Inconsistent Data	28
3.4	Edges between Consecutive Cells	31
3.5	Edges from Gene to Color or Maturation Nodes	31
3.6	Edges from Maturation to Gene Nodes	32
3.7	Depiction of a Noisy OR Statement	35
3.8	Gene to Color Node Edge Weights	36
3.9	Critical Region for a Conjugation Event	39
4.1	Splitting a Query	45
4.2	Relevant Nodes for a Query	50
5.1	CDFs for Expression Delay	55

5.2	CDFs for Maturation Delay	57
5.3	Bounding Polygons for Cell Contact	58
5.4	Depiction of Contact Quality Functions	59
5.5	Tracking Error Leading to a Zero Probability Query	61

List of Tables

3.1	Sum of Square Errors Fail Case	40
5.1	Estimations of Expression Delay	54
5.2	Description of Experimental Trials	60
5.3	Description of Queries for each Trial	60
5.4	Average Trial Rankings	63
5.5	Average Query Rankings	63
5.6	Total Query Probabilities	64
5.7	Query Probability Rankings	64
5.8	Comparison of Rankings	65

Chapter 1

Introduction

Bacteria are known for their ability to adapt and evolve rapidly. This trait is due in part to the ways in which they share genetic information. Gene transfer in bacteria can be classified based on whether genetic information is passed ‘vertically’ between generations or ‘horizontally’ within a generation. Vertical gene transfer (VGT) occurs through cell division, while horizontal gene transfer (HGT) includes several distinct methods of gene transfer between cells. This work focuses on conjugation, a type of HGT which plays an important role in prokaryotic evolution. It also has the potential to be a powerful tool for bio-engineering.

Although bacterial conjugation has been researched extensively, it is difficult to study and model at the single-cell level [38, 41]. The rapid growth of bacterial populations makes it hard to distinguish if gene spread is due to HGT or VGT. The scale at which conjugation occurs makes it technically difficult to detect individual conjugation events, even with the aid of a microscope. Challenges for modeling population-level behavior include the stochastic nature of cell-cell interactions, the rapid cell division rate, and the need to model numerous individual cells. While most previous work is focused on population dynamics, recent advances in microfluidics and agent-based modeling create opportunities to study conjugation at the single-cell level.

Understanding the mechanisms that govern conjugation requires us to identify the specific cells involved in gene transfer. These cells are often referred to as a ‘mating pair.’ Many studies to date rely on ad hoc, manual identification of mating pairs from images [3, 21, 22, 23, 34]. This process is both time-consuming and limited to clear-cut situations. Therefore, this thesis presents a novel computational approach which aims to provide:

- (i) insight into the mechanisms governing conjugation at the single-cell level, and

(ii) a method of inferring which cells formed successful mating pairs.

This approach was designed in parallel with an experimental study and image-processing software. Together, we propose a pipeline for analyzing interactions between individual cells from imaging data. Our portion of the pipeline is concerned with inferring information from the processed data. We use a model comparison approach because the system we are modeling is difficult to reproduce with a generative model.

We create a single graph representation for each experimental trial. The edge weights are determined by probability distributions that represent mechanisms involved in conjugation. To create multiple models, we consider different sets of distributions. We then evaluate which model most accurately explains the experimental results. In this way, model comparison provides insight into the mechanisms underlying conjugation.

The chapters of this thesis are organized as follows. Chapters 1 and 2 provide background information on conjugation and Bayesian networks, respectively. Details of our experimental data and model formulation are in Chapter 3, with an overview of the implementation in Chapter 4. The specific models we compare are described in Chapter 5, along with the results of the comparison. The final chapter includes our plans for future work on this project.

1.1 Horizontal Gene Transfer

The three main mechanisms of HGT in bacteria are transformation, transduction, and conjugation [9, 35]. Transformation and transduction are both consequences of other biological processes and generally involve DNA fragments. Bacteria may excrete DNA fragments or release them during cell death. Transformation is the process by which bacteria take up environmental DNA fragments and incorporate them into their genome. Specific enzymes are required for bacteria to incorporate DNA in this manner.

In contrast to transformation, which does not require a transmission vector, transduction occurs when genes are transferred by bacteriophages. Bacteriophages are viruses that infect bacteria. They replicate within a host, release themselves into the environment, and then infect a new host. On rare occasions during replication, a bacteriophage incidentally absorbs DNA fragments from its host. When that bacteriophage injects its DNA into a new host, it can also transfer the fragments.

Conjugation is unique because it allows bacteria to share genes directly and requires a physical connection between two cells. This connection allows for the transfer of an extra-chromosomal DNA molecule called a plasmid.

1.1.1 Plasmids & Conjugation

The main distinction between chromosomes and plasmids is that chromosomes contain necessary genes, whereas plasmids contain accessory genes [5, 10]. Necessary genes regulate critical processes like protein synthesis and cell division. Accessory genes provide evolutionary advantages that may only be useful in specific environmental conditions. For instance, antibiotic resistance genes are beneficial for survival only when an antibiotic is present. Maintaining a plasmid uses cellular resources, so plasmids that do not convey substantial benefits are less likely to persist within a population [28, 35].

Both chromosomal and plasmid DNA are passed on during cell division, but only plasmid DNA is transferred through conjugation. Cellular machinery facilitates and regulates conjugation. However, conjugative plasmids are considered self-replicating because they contain genes that initiate conjugation and code for the necessary machinery [9, 28, 39]. The majority of plasmids are non-conjugative and cannot be transferred via conjugation. Some mobilizable plasmids can transfer with ‘assistance’ from another conjugative plasmid.

A conjugation event is the successful transfer of a plasmid from a donor cell to a recipient cell, as depicted in Figure 1.1. The process is generally described as follows [28, 9]:

1. The donor cell extends a tube-like appendage called a pilus.
2. The pilus facilitates a physical connection with the recipient cell.
3. A copy of the plasmid transfers from the donor cell to the recipient cell.
4. The connection is severed and the cells separate.

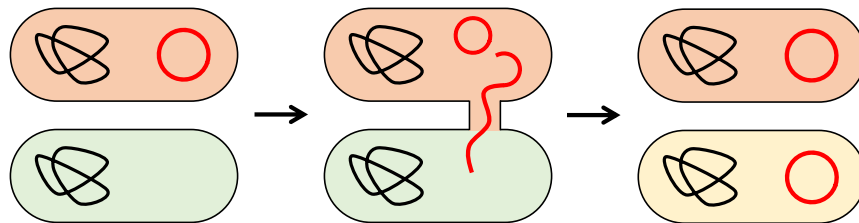


Figure 1.1: A cell transfers a plasmid to another cell, causing it to change phenotype.

At this point, the recipient cell has a complete copy of the plasmid and is referred to as a ‘transconjugant.’ The plasmid then initiates the development of conjugative machinery in the new cell. It may eventually act as a donor and transfer the plasmid to new recipients. While conjugation has been observed to occur on a timescale of minutes, there can be substantial delays before the functional and conjugative genes on the plasmid are expressed [26]. These delays pose interesting experimental and modeling challenges, and are discussed further in Section 1.3.1.

Notably, the details of the conjugation process differ significantly between types of plasmids. The model system for conjugation is the F-plasmid system, characterized primarily in *E. coli* [39]. F-pili are relatively long and flexible, so cell wall contact is not required for conjugation [12, 15]. Researchers have observed conjugation events between bacteria as far as 12 microns apart, over twice the length of a typical *E. coli* cell [3]. Another commonly studied system is the P-plasmid system. It is characterized by short, rigid pili and relies more heavily on direct cell wall contact [17, 33]. Thus, F-plasmids transfer well in variable environments with fewer cells (e.g. fluids), and P-plasmids are suited to densely packed environments (e.g. surfaces).

1.2 Relevance

One motivation for studying bacterial conjugation is learning about its role in evolution. Plasmids are ubiquitous because they self-propagate rapidly through microbial populations. Conjugation is best known for its role in the spread of antibiotic resistance. It is also responsible for the prevalence of important genes in a variety of contexts [9]. For example, genes for nitrogen fixation, heavy metal resistance, and lactose usage are found on plasmids. Conjugation occurs in environments ranging from soil to seawater to animal intestines. It is believed to play an important role in the formation and maintenance of biofilms [25, 39]. Additionally, bacteria can conjugate to cells from different species and to organisms belonging to different kingdoms, such as yeast [16].

Conjugation also has the potential to be a valuable tool for microbial genetic engineering [29, 41]. Plasmids are good targets for bio-engineering because it is relatively simple to modify them to contain different accessory genes. Their ability to spread horizontally via conjugation makes it possible to affect an entire population by introducing a small number of plasmid-bearing cells. The combination of these features makes conjugation particularly valuable for applications that require modification of an environmental population, such as environmental remediation, wastewater treatment, and agriculture. In fact, many plasmids

already contain genes that facilitate the breakdown of chemical compounds [9]. It is likely that conjugation can be used to introduce more effective versions of these genes.

There are significant risks involved in releasing modified, mobilizable genetic elements to the environment [29]. It is unclear if these elements will persist in the environment, and if so how they will spread. If the goal is environmental remediation, it would not be desirable for the plasmid to spread to populations outside the affected area. Likewise, if a modified plasmid is introduced to specific bacterial populations, it may be problematic if it spreads to other microbes. Other concerns may arise on a longer timescale, such as how engineered genes could mutate or recombine with existing genes. These applications are in their infancy, and significant policy decisions and regulations need to be developed before they are implemented.

1.3 Experimental Approaches

When designing plasmids for engineering purposes, it is desirable to maximize the frequency of conjugation and the spread throughout the community. Understanding the factors which lead to these outcomes requires us to analyze both individual conjugation events and population dynamics. Most studies to date focus on population dynamics because it is extremely challenging to detect individual conjugation events. Common methods focus on population-level properties rather than individual cell traits [38]. Studies have investigated properties including conjugation frequency, optimal ratios of donor to recipient cells, and ideal environmental conditions [18].

Experiments that study conjugation differentiate cells based on if and when they acquire a plasmid. Because plasmids are passed down during cell division, cells are classified in terms of their lineage. Donors are from lineages that contained the plasmid at the start of the experiment and recipients are from lineages that did not. Transconjugants are cells from recipient lineages that received the plasmid via conjugation. One method of distinguishing cell types is using different strains of bacteria for the donors and the recipients. In this case, transconjugants are the cells from the recipient strain that have the plasmid. Another option is to include an additional, non-conjugative marker plasmid in the donor or recipient population. To detect conjugation without a genetic assay, the plasmids contain reporter genes that cause observable phenotypic changes.

Antibiotic resistance is routinely used as a selection marker when adding engineered plasmids to bacterial strains [38]. After the populations are allowed to interact, an antibiotic is introduced to the media. The antibiotic kills cells without the plasmid, effectively

distinguishing which cells received it. This method does not provide information about when cells were conjugated to, making it is impossible to determine which cells were involved in conjugation events. Furthermore, these experiments are often carried out in liquid media and mating filters, neither of which provide longitudinal data on individual cells. The lack of information about individual cells also makes it impossible to determine whether any observed increase in the transconjugant population is due to conjugation or to cell division.

1.3.1 Approaches for the Single-Cell Level

Studying conjugation at the single-cell level involves tracking the phenotype and position of individual cells over time. There are two advancements that make this possible: fluorescent proteins and microfluidic traps. Plasmids can be modified to contain genes coding for fluorescent proteins, so that cells produce an optical signal after receiving the plasmid [38, 41]. The different populations can now be distinguished in imaging data. Time-lapse imaging provides spatial data across multiple time points. The time at which a cell begins fluorescing provides information about when its lineage was conjugated to.

Ideally, images taken under the fixed focal plan of a microscope capture the state and location of every cell in the population. This setup requires bacteria to grow in a single layer. Liquid media and mating filters are clearly unsuitable, and traditional agar plates do not provide the smooth surface required for imaging. One solution is microfluidic traps, such as the ones shown in Figure 1.2. They have a height just greater than that of a bacterium, which forces cells to grow in a single layer [24, 40]. At least one end of the trap is open to a flow of media. The media provides nutrients to the cells and washes away excess bacteria and waste products. (Because the population grows exponentially, it is not feasible to make a trap large enough to contain the entire population.)

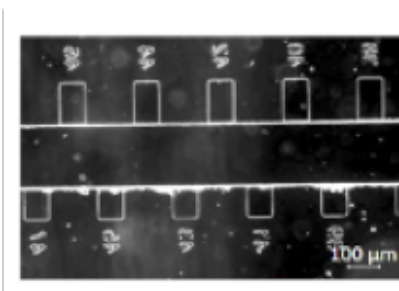


Figure 1.2: Cells grow in the rectangular traps off the middle channel. The channel allows media into the traps and washes excess cells away. Image taken by Sara Haghayegh.

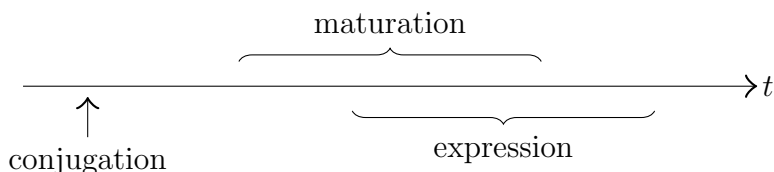


Figure 1.3: A timeline of delays in the conjugative process. Brackets denote potential ranges of the delays.

These methods make it possible to study conjugation at the single-cell level and to observe how a plasmid spreads through a population spatially. However, the variable delays in the conjugative process make it difficult to determine mating pairs. These delays are due to the time it takes to replicate DNA and to build cellular machinery for conjugation. The most relevant delays occur during gene expression and plasmid maturation [26, 34]. The expression delay makes it difficult to determine the time of the conjugation event, while the maturation delay makes it difficult to determine the set of potential donors. A potential timeline for these delays is shown in Figure 1.3.

The ‘expression delay’ is the time between conjugation and the expression of the functional gene on the plasmid. For plasmids with fluorescent proteins, this delay corresponds to the time before a cell’s lineage changes color. (It includes the time necessary for the fluorescent proteins to develop and become visible.) The expression delay may be different than the ‘maturation delay.’ Mature cells are those that are able to donate the plasmid. The maturation delay is the time between conjugation and the development of the conjugative machinery. This discrepancy means that transconjugants may donate the plasmid before they are detected. Both delays are often longer than the life span of a single cell, meaning that a conjugation event only leads to visible or functional changes in the recipient’s descendants. Thus, many studies only analyze events where the transconjugant’s ancestors came into contact with a single donor cell [3, 21, 22, 23, 34].

1.3.2 Previous Experimental Studies

Some studies use manually identified pairs to study conjugation at the single-cell level. In [21], the authors investigated the intracellular localization of a plasmid in *E. coli*. They used a type of fluorescent protein which binds to double-stranded DNA (dsDNA). This approach causes fluorescent foci appear around each established plasmid. They found that plasmids were evenly spaced and located at the center or quarter positions of the cell. Moreover, the authors considered the orientation of successful mating pairs. Most were

aligned such that their lateral (long) sides were in contact; however, transfer appears to be possible from any point along the cell wall.

The most similar approach to our experimental setup, as described in Section 3.1, is from [34]. They used time-lapse imaging of microfluidic traps to study plasmid invasion in *P. putida*. The plasmid codes for fluorescent proteins to allow for visualization of the different populations. Manual identification of mating pairs was used to investigate factors including the distance between and orientation of successful mating pairs. As in [21], they found that laterally aligned pairs were most common. An additional experiment that suggested recipient cells are most likely to be conjugated to at later stages of the cell cycle. Their analysis also suggests that elongating cells are more likely to conjugate than non-growing cells.

Another approach to studying conjugation at the single-cell level involves using fluorescent proteins that bind to single-stranded DNA (ssDNA). During conjugation, a single strand of the plasmid DNA is transferred. (A second strand is replicated in both the donor and recipient.) Because plasmid DNA is only single-stranded during transfer, these proteins cause the plasmids to fluoresce only during conjugation. Concurrent appearance of conjugative foci in a nearby donor and recipient cell indicate a mating pair. Notably, this method also allows for the detection of multiple conjugation events involving a single recipient. Including a second color of fluorescent protein which binds only to dsDNA allows for the continued detection of plasmids after the second strand has been replicated. Both [6] and [12] use multiple types fluorescent proteins which bind to ssDNA or dsDNA to study F-plasmid systems in *E. coli*.

The intracellular dynamics of conjugation is investigated in [6]. The authors consider processes including the conversion of ssDNA to dsDNA after conjugation, duplication of plasmids within a cell, and intracellular localization of plasmids. Analysis of mating pairs indicates that ssDNA is most likely to exit the donor from the lateral side and enter the recipient from the poles. The authors also found that conjugation can occur at any point during the cell cycle, in contrast to the results from [34]. Finally, the authors of [12] study conjugation between cells which are physically distant. Maleimide labeling is used for visualization of cell walls and pili. The authors characterize properties such as the number, length, and location of pili. They observe relatively little differences in the pili throughout the cell cycle. Most pili are on the lateral side of the cell, which is consistent with the location at which [6] observed ssDNA exit from donor cells.

1.4 Previous Modeling Approaches

Both deterministic and stochastic approaches are used to model conjugation at the population level. These models consider how various factors affect the proportion of plasmid-bearing cells within a well-mixed population. In [26], the authors develop a delay-differential equation model to study various delays involved in the conjugative process. A stochastic differential equation (SDE) model is used to analyze experimental data in [30]. The authors couple the SDE to an observation equation and use it to better estimate parameter values. Another paper uses multiple stochastic models to investigate plasmid persistence within a population [31]. They consider how plasmid loss, maintenance cost, and conjugation affect persistence. A different approach is required to model conjugation in filter matings. In [20], the authors model interactions between colonies by assuming each colony contains a single type of cell (donor, recipient, or transconjugant).

Agent based models (ABMs), also known as individual based models (IBMs), are a type of computational model. They are often used to model conjugation at the single-cell level. ABMs model interactions within a large group of autonomous individuals by representing each individual with a separate set of equations. This approach makes ABMs suitable for systems with high degrees of stochasticity and heterogeneity. However, running these models is computationally expensive.

There are several platforms for agent based modeling of bacterial populations. These models often include cell growth, intracellular processes, and environmental conditions. DiSCUS (Discrete Simulation of Conjugation Using Springs) is an ABM developed specifically to study conjugation [13]. CellModeller is a more general platform for modeling populations of rod-shaped cells [32]. It was originally designed to study biofilms, but includes options to model conjugation. Other ABMs were designed for more specific modeling goals. For instance, COSMIC-rules (Rule-based Computing System for Microbial Interactions and Communications) was developed to study the spread of antibiotic resistance plasmids [14]. An ABM for plasmid invasion of biofilms is presented in [27]. Generally, these models do not vary the probability of conjugation based on the properties of the mating pair. They include either a fixed probability of conjugation or sample from a distribution and choose a recipient arbitrarily.

Chapter 2

Probabilistic Graphical Models

Our goal is to infer information about interactions within a network of cells, each with several relevant properties. We have information about some properties (e.g. fluorescence) and we want to determine the likely value of other ones (e.g., having the plasmid). Additionally, we know the causal relationships between these properties; for instance, receiving the plasmid eventually causes changes in fluorescence. It is useful to think of each property as a random variable. This representation allows us to think of our network of cells as a set of random variables which are dependent on each other.

Suppose we want to calculate the probability that each variable is equal to a particular value, called an assignment. Determining this probability requires us to consider the joint distribution of all the variables. When working with large networks, it is infeasible to enumerate the probability of all possible assignments. (Even for binary variables, the joint distribution of n variables requires the consideration of 2^n possibilities.) Instead, we factorize the probability distribution into the product of conditional probability distributions (CPDs). Each CPD is the probability distribution for a single variable. It is conditioned on the other variables that directly influence its value. (We will describe precisely what we mean by a direct influence later in the chapter.) Factorizing the joint distribution into a CPD for each variable reduces it to a product of n CPDs. In this way, we can consider events which could have been caused by a large number of low-probability interactions.

Partial information is common in experiments, where we can only observe part of a system or collect indirect evidence. Suppose that the numerical values of a subset of the random variables are known. It is useful to understand which unknown variables are independent given the values of the known variables. (Independence simplifies the calculations needed to determine the probability of an assignment of the variables.) While the CPDs

provide a complete characterization of the network, they do not provide an intuitive understanding of how evidence affects whether two variables are dependent. To understand dependencies, we construct a graph in which the variables are nodes and the direct dependencies are edges. This graph representation of a joint probability distribution is a probabilistic graphical model (PGM). It provides a way of visualizing when two variables in our network are independent. Our discussion of PGMs draws from Chapter 3 of [19].

This chapter describes how we represent a set of conditional probability distributions as a graph. We demonstrate that this graph can provide additional insight into the joint probability distribution. In our case, the graph is directed because each edge represents causation and it is acyclic because every edge points forward in time. (The properties of a cell are only impacted by its past state and past interactions.) When the graph structure of a PGM is directed and acyclic, we call it a Bayesian network. Since our network is so complicated, we begin by illustrating the relevant concepts with a simple example.

2.1 Bayesian Networks

A classic example of a PGM arises when trying to diagnose an illness based on symptoms. A simple case is constructed using two illnesses (allergies, virus), three symptoms (rash, headache, fever), and one relevant external factor (season). We think of each of these factors as a random variable. Illnesses and symptoms are binary variables representing whether the patient has them, while the season is a variable with four states. As in our network, we know the causal relationships between these variables: spring and summer lead to an increased prevalence of allergies, fall and winter lead to an increased prevalence of viruses, allergies can cause rash and headache, and viruses can cause headache and fever. By representing each variable as a node and each cause as an edge, we get the graph shown in Figure 2.1. Notably, there are no cycles in this graph.

In this example, we use the set of dependencies to build a graphical representation of the variables. Independence is the opposite of dependence, so this representation also encodes information about independence. More precisely, the graph provides a way of categorizing when two variables are conditionally independent. We say that X and Y are (conditionally) independent given Z if $P(X, Y|Z) = P(X|Z) \cdot P(Y|Z)$, or, equivalently, $P(X|Y, Z) = P(X|Z)$. (Note that X , Y , and Z may represent either a single variable or a set of variables.) Intuitively, conditional independence says that X and Y do not influence each other if Z is known. Another interpretation of this statement is that X and Y only impact each other *indirectly* through Z . This interpretation is what we refer to when we

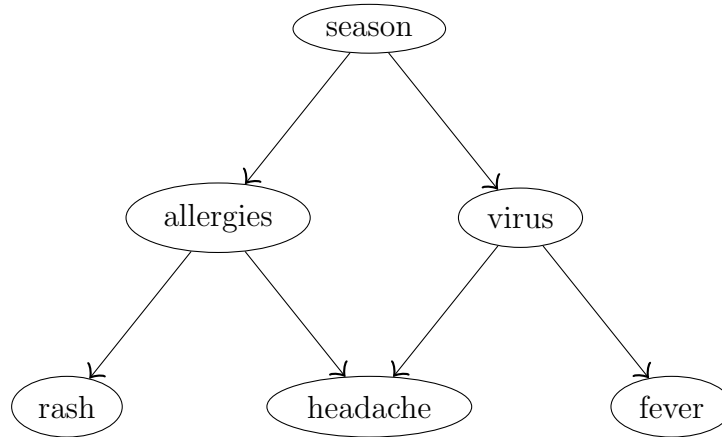


Figure 2.1: A graphical representation of the relationship between various illnesses and symptoms. The edges point in the direction of causation. Adapted from [19].

say two variables ‘directly influence’ each other - X impacts the value of Y , without first influencing some other variable(s) Z .

We frame our definitions with respect to the process of constructing a graph from a set of variables and the dependencies between them.

Definition 1 Let $G = (V, E)$ be a graph where $V = \{X_1, \dots, X_n\}$ correspond to random variables and the directed edge (X_i, X_j) is in E iff X_i directly influences X_j . Then, $G = (V, E)$ is a **Bayesian network structure** if it is acyclic.

A Bayesian network is the combination of this structure with a joint probability distribution that satisfies certain requirements. To understand how the graph structure and probability distribution relate, we return to the previous example. The example depicted in Figure 2.1 only has 6 nodes, but the joint probability distribution includes $4 \cdot 2^5 = 128$ possible assignments to the variables. However, we can factorize the joint distribution into a product of conditional probabilities based on the above causal relationships. Representing each variable by its first letter for clarity, we get

$$P(R, H, F, A, V, S) = P(R|A) \cdot P(H|A, V) \cdot P(F|V) \cdot P(A|S) \cdot P(V|S) \cdot P(S)$$

The probability of any assignment is the product of just 6 values. $P(R|A)$, $P(H|A, V)$, and $P(F|V)$ are the probability distributions for a symptom, conditioned on having an illness which can cause it. $P(A|S)$ and $P(V|S)$ are the probability distributions for illnesses, conditioned on the season.

Although there are only a few CPDs in this example, it is not immediately clear why the probability can be written as such. This factorization is easier to understand and verify if we reference the graphical representation in Figure 2.1. Each CPD corresponds to the probability distribution for a node, conditioned on the values of its parents. In this way, the graphical representation encodes two equivalent ideas:

- (i) the factorization of the joint probability distribution, and
- (ii) the dependencies between variables.

More formally, factorization is defined as follows:

Definition 2 Let $G = (V, E)$ be a directed, acyclic graph where $V = \{X_1, \dots, X_n\}$ correspond to random variables. A probability distribution P over V **factorizes** G if

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa_{X_i}^G),$$

where Pa_X^G is the set of all parents of X in G .

We build a graphical representation not just from a set of dependencies between variables, but also from the CPDs that describe those dependencies. A Bayesian network is the combination of that graph, or Bayesian network structure, and the corresponding probability distribution.

Definition 3 Let $G = (V, E)$ be a Bayesian network structure, and let P be a probability distribution over V which factorizes G . Then, $B = (G, P)$ is a **Bayesian network**.

2.2 Conditional Independence in a Bayesian Network

The key observation when analyzing a Bayesian network structure is that most of the relationships are indirect. Intuitively, two variables are dependent if they can influence each other through a trail in the graph. (A trail is the equivalent of a path, except that the direction of the edges is irrelevant.) Knowing the values of some variables can affect the flow of influence. That is, whether two variables are independent may depend on what other variables are known. Many sets of CPDs can factorize the same graph, so it is

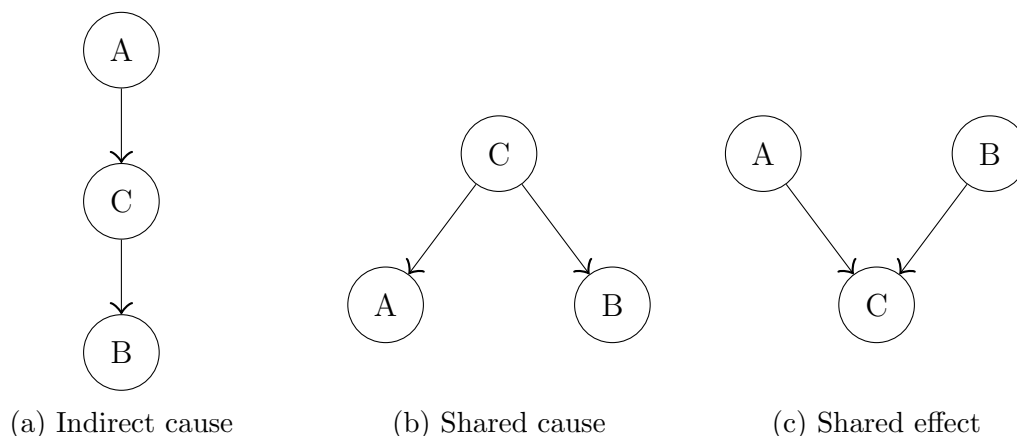


Figure 2.2: All graph structures in which two nodes are related via a third.

useful to know what statements about conditional independence are true regardless of the exact CPDs. In this section, we will explore how a Bayesian network structure encodes information about conditional independence.

We begin describing why variables connected by an edge can never be guaranteed to be conditionally independent. Suppose A has a causal effect on B , so there is an edge $A \rightarrow B$. Let A take on values determined by any random distribution which is independent of its parents. If B is defined to have the same value as A , then A and B are dependent, regardless of what information is known about other variables. (To see why A must be independent of its parents, suppose A has the same value as its parent C . Then, B would also equal C , making it so that $P(B|A, C) = P(B|C)$.)

We now consider cases in which A and B are related indirectly via a trail. The simplest case is when the trail has just one other node C . Figure 2.2 shows the three ways A and B can be related via C . It turns out that the analysis of these cases can be generalized to any two nodes in a Bayesian network structure.

We start with Figure 2.2a. This structure can be interpreted in two ways: A is an indirect cause for B or B is indirect evidence for A . For instance, recall the trail $\text{season} \rightarrow \text{virus} \rightarrow \text{fever}$ from Figure 2.1. Season is an indirect cause of fever because it impacts how likely someone is to have a virus, which in turn impacts how likely they are to have a fever. Conversely, fever is indirect evidence of season. Fever is often caused by a virus, and viruses are most prevalent in a cold season. In either case, knowing whether a person has a virus removes this indirect causation or evidence; once we know whether they have a virus, knowing the season does not affect the probability they have a fever,

and vice versa. Knowing the value of the middle node makes the end nodes independent. We conclude that for this type of trail, A and B are independent given C .

Figure 2.2b can be interpreted as A and B sharing a common cause. This trail arises in the previous example in the form of headache \leftarrow virus \rightarrow fever. If someone has a fever, then they are more likely to have a virus; if someone is more likely to have a virus, then they are more likely to have a headache. In the absence of information about whether a person has a virus, headache and fever are dependent. Once we know whether someone has a virus, knowing they have a fever gives no additional information about their likelihood of having a headache. We again conclude that knowing the value of the middle node makes the end nodes independent. A and B are independent given C for this type of trail.

The final case, shown in Figure 2.2c, can be interpreted as A and B sharing a common effect. It was seen in the example as allergies \rightarrow headache \leftarrow virus. Suppose we know that someone has a headache. This information increases the likelihood that they have at least one illness. If we also know that the person has allergies, then it is likely the headache was caused by the allergies. This implication also means it is less likely to be caused by a virus. In this way, knowing the value of C allows influence to flow between A and B . If we do not know whether someone has a headache, then having allergies does not give any information about the likelihood they have a virus. We conclude that *not* knowing the value of the middle node makes the end nodes independent. In this type of trail, A and B are independent only if we do *not* know the value of C .

An active trail is one which allows information to flow between the nodes on either end. For 3-node trails, the above analysis tells us that

- $A \rightarrow C \rightarrow B$ is active iff C is unknown,
- $A \leftarrow C \rightarrow B$ is active iff C is unknown, and
- $A \rightarrow C \leftarrow B$ is active iff C is known.

If a trail is part of a larger graph, it is not sufficient to consider only A , B , and C . Consider the graph shown in Figure 2.3. If we know the value of Z , then we have information about C , which makes the trail between A and B active. Indirect information about other types of 3-node trails does not prevent them from being active, because the value of the middle node is still uncertain.

We can view a longer trail as a series of 3-node trails and evaluate whether each one is active. For the endpoints to influence each other, information must be able to flow through every ‘intermediate’ 3-node trail. We formalize the idea of an active trail as follows. Note that the symbol \rightleftharpoons is used to denote an edge in a trail.

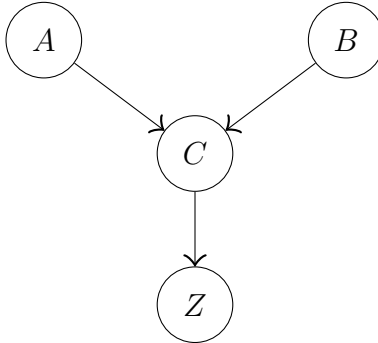


Figure 2.3: The trail $A \rightarrow C \leftarrow B$ is active if Z is known.

Definition 4 Let G be a Bayesian network structure and let Y be a set of variables whose values are known. Then, the trail $X_1 \rightleftharpoons \dots \rightleftharpoons X_n$ is active given Y if the following conditions hold.

- (i) If there is a 3-node trail of the form $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, for $i \in \{2, \dots, n-1\}$, then X_i or one of its descendants is in Y .
- (ii) No other nodes from the trail are in Y .

A trail is active whenever there is information about every shared effect node in the trail and there is not information about the remaining nodes in the trail. The absence of active trails guarantees the conditional independence of two sets of variables. The following theorem confirms the intuition that we have been building in this section - conditional independencies in the graph of a Bayesian network imply conditional independencies in the probability distribution.

Theorem 1 Let $B = (G, P)$ be a Bayesian network and let A , B , and C be sets of nodes in G . If for every pair of nodes $a \in A$ and $b \in B$, there is no active trail between a and b given C , then A is conditionally independent of B , given C .

For a proof of this theorem, see [19]. The important consequence for us is that we can use different CPDs to create many models of one system. As long as the set of direct dependencies are the same, the underlying Bayesian network structure is the same. Thus, the same independence statements hold. Leveraging this observation also simplifies the calculations we make when computing a query.

2.3 Querying a Bayesian Network

We now consider the probabilities that we wish to calculate. A query is the term for a probability calculation made using a PGM. Experiments often provide information about a subset of the variables of interest, perhaps through the form of indirect observations. In this case, it is natural to ask questions about the variables which lack direct evidence, conditioned on the experimental evidence. A query generally involves three sets of variables:

- E : Evidence variables have been given an assignment e ,
- X : Query variables have values that are of interest, and
- H : Hidden variables make up all remaining variables in the model.

Note that there cannot be any overlap between the sets of variables. Hidden variables are required because a joint probability distribution includes *all* variables in the model. Finding a conditional probability distribution of the form $P(X|E = e)$ requires the calculation

$$P(X|E = e) = \sum_h P(X, H = h|E = e).$$

The need to marginalize over all the hidden variables causes this to be an expensive process. Taking advantage of known conditional independencies allows us to speed up the process: if a hidden variable is independent of the query variables given the evidence, we factor it out of the summation.

Returning to the diagnosis example from Figure 2.1, the evidence variables are the symptoms a patient has and the query variables are the potential illnesses. If the season is not known, then it is a hidden variable. One question we might ask is the probability a patient has various illnesses, given their symptoms. This question corresponds to a type of query called a conditional probability query.

Definition 5 A *conditional probability query* is the calculation of the distribution $P(X|E = e)$, where E is the set of evidence variables and X is the set of query variables.

The names suggest that the evidence variables are those we observe and the query variables are those we do not. However, it may be interesting to let some of the known variables be the query variables. We can then calculate the probability of the observed values given a subset of the evidence. In this case, it suffices to calculate the probability of

a specific outcome, $P(X = x|E = e)$. This computation is much faster than determining the entire distribution.

As an alternative to a conditional probability query, we can calculate the most likely assignment of the query variables. In the diagnosis example, this query might find the set of illnesses a patient is most likely to have. This task corresponds to a marginal MAP (maximum a posteriori probability) query.

Definition 6 *A marginal MAP query is the calculation of the variable assignment x which yields $\max_x P(X = x|E = e)$, where E is the set of evidence variables and X is the set of query variables.*

A marginal MAP query finds the most likely joint assignment for the query variables. This assignment may be different than the set of most likely assignments for each individual variable. There may also be multiple assignments which yield the same greatest probability.

Chapter 3

Model Formulation

In this chapter, we describe the experimental data and how a Bayesian network can be used to represent it. Each trial is characterized by a single graph representation and each model is given by a selection of probability distributions. We cover how each representation is constructed, including the graph structure and edge weights. Finally, we explain how queries are used to compare the different models. By comparing different model variants across multiple trials, we gain insight into which distributions most accurately describe conjugation.

3.1 Experimental Setup

Our group conducts experiments in which bacteria grow and conjugate in a single layer in the focal plane of a microscope. Specific microfluidic traps were designed for the purpose of maintaining a single layer of cells. The bacteria used in these experiments are *Escherichia coli*, with an individual cell being approximately 3–6 μm in length and 0.65 μm in diameter. Each trap is 0.74 μm high. The length and width of each trap is 80 μm , which provides space for hundreds of cells. One end of each trap is open so that media can flow in and excess cells and waste products can flow out. A schematic of this design is shown in Figure 3.1. Experiments are run for a set of traps, so each trap can be thought of as a separate trial.

The experimental data comes from work done by Aaron Yip. He is investigating how bacteria can be used to produce PETase, an enzyme which facilitates the breakdown of PET plastic. Specifically, he is interested in how conjugation can be used to modify

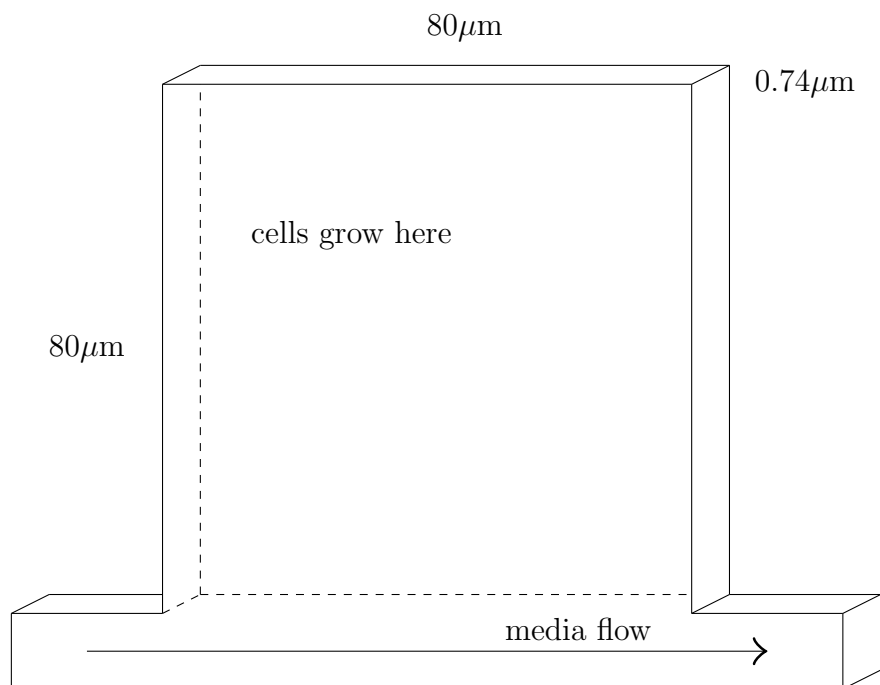


Figure 3.1: A schematic of a single microfluidic trap. Note that the traps lie flat, and the flow channel is not drawn to scale.

environmental populations to express PETase, for eventual applications in wastewater treatment and environmental remediation. Some of the plasmids involved in the experiment contain genes coding for a version of PETase. (The inclusion of this gene is not relevant for our analysis.) Additionally, the plasmids contain reporter genes coding for fluorescent proteins, eventually allowing us to observe which cells have each plasmid. At the start of the experiment, a small number of cells from two populations of *E. coli* are loaded into each trap:

- Donor cells contain a conjugative P-plasmid system which codes for red fluorescent protein (RFP).
- Recipient cells contain a non-conjugative plasmid which codes for green fluorescent protein (GFP).

As cells interact, the RFP plasmids are transferred to the recipient population via conjugation, leading to the formation of transconjugants. The concurrent expression of GFP

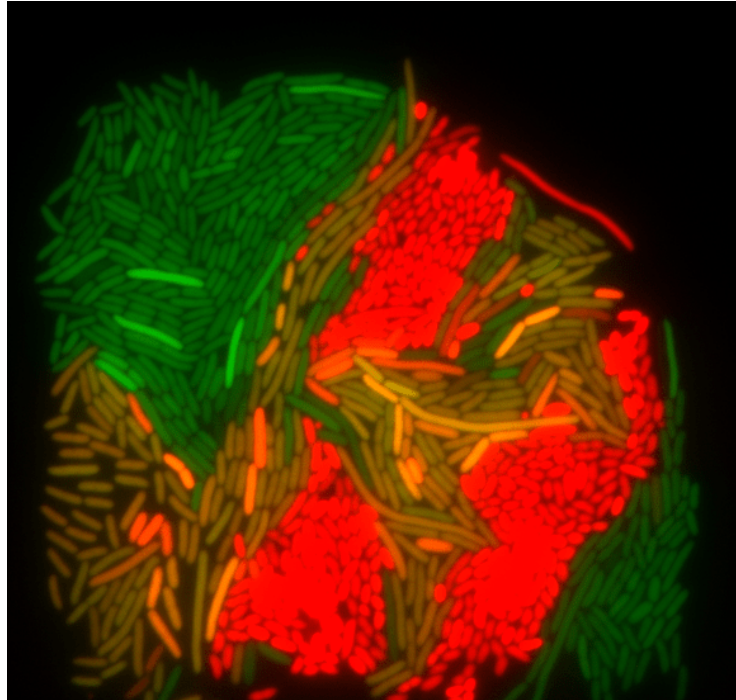


Figure 3.2: A sample frame from one of Aaron Yip’s experiments in which groups of recipients (green), donors (red), and transconjugants (yellow) are all visible. Relative fluorescence levels have been adjusted for visibility.

and RFP by transconjugant cells causes them to appear yellow-orange in the overlay of green and red channels. After a lag corresponding to the expression delay, we can visually distinguish the transconjugants from donor and recipient cells. Because the term donor also refers to any cell that can donate the plasmid (including mature transconjugants), it is convenient to refer to populations via their color. We refer to donors, recipients, and transconjugants cells as red, green, and yellow, respectively. (An ambiguity of this terminology is that green cells include those transconjugants which have received the plasmid but not yet changed color, i.e. are not yellow.)

Time-lapse microscopy images of the traps are taken in 5 minute intervals, over a span of 20-24 hours. Because the doubling time of our strain of *E. coli* is approximately 40 minutes, the microfluidic trap quickly becomes packed (within the first half of the experiment). The microscope has specific filters to pick up red and green fluorescence. A representative frame from an experiment is shown in Figure 3.2.

3.1.1 Image Processing

Data from the experiments is in the form of time-lapse images of each trap, so several processing steps are needed to extract information. These steps are outside the scope of our work. We include a brief overview of them because they are necessary for the experiment to analysis pipeline. There are three main steps in this process: data extraction, lineage corrections, and transconjugant identification.

First, preexisting software packages (Omnipose and CellProfiler) are used to extract information about cell features and lineages [8, 36]. From the raw imaging data, they create a csv file in which each row corresponds to a cell at a time point and each column corresponds to a feature. This process involves three main steps:

- (i) Cell Segmentation determines which groups of pixels represent a cell. This step separates cells from each other and the background, and outputs a set of objects for each image.
- (ii) Cell Tracking determines corresponding cell objects in consecutive images. This step creates either a track link or parent link for each related pair, depending on whether it divided between frames.
- (iii) Feature Assignment determines properties of individual cells. This step identifies relevant properties of each cell, such as its length and fluorescence.

One important feature of CellProfiler is that it fits an ellipse to each cell object. The features it records in the third step, such as length, are based on the fitted ellipse. CellProfiler also assigns an ID number to each cell object, which persists between frames. The data we use for each cell is summarized in the following list.

- Spatial Data: position and orientation
- Cell size: length and width
- Fluorescence intensity: red and green levels
- Lineage Tracking: cell and parent cell IDs

While using packages such as CellProfiler is standard practice, they are not perfect and often provide output which includes segmentation and tracking errors. Tracking errors

stem from assigning a link incorrectly or missing a link entirely. Biological quirks, such as cells that do not resemble ellipses, contribute to these errors. Accurate lineage tracking is critical for our ability to track the spread of the plasmid through the population. Our colleague Atiyeh Ahmadi is working on an algorithm to refine the results from CellProfiler. While the algorithm is not complete at the time of writing this thesis, it will be integrated into the pipeline at a later date.

One process CellProfiler is not designed for is detecting changes in fluorescence, which is how we identify newly formed transconjugants. Specifically, we need to know when a previously green cell first starts expressing RFP. Background fluorescence from nearby cells makes it challenging to set a threshold for detecting this color change. Our colleague Aaron Yip trained Ilastik, a machine learning tool for image classification, to identify newly formed transconjugant cells [4]. His approach detects and flags cells as transconjugants for the first few time steps after this color change. (It occasionally flags a lineage for much longer.) To determine the entire transconjugant population, we assume that every descendant of a transconjugant also has the plasmid. A binary value indicating whether a cell was flagged by this process is added to the CellProfiler output.

3.1.2 Synchrony

The time delay between conjugation and gene expression is on the same scale as the time between cell divisions. This overlap means that transconjugants often divide after receiving the plasmid but before expressing RFP. In this case, all of the original transconjugant's descendants are expected to change color at roughly the same time. Conversely, if we observe that every descendant of a cell changes color 'synchronously,' we infer that their common ancestor was conjugated to. (Because conjugation is a rare event, it is less likely that every descendant was conjugated to at approximately the same time.)

We do not have a good way of encoding this phenomenon into our representation directly. As with the example in Chapter 2, our graph is built on causal effects. Synchrony is not a cause of conjugation. To make our analysis more accurate, we check for this phenomenon in advance. We use the results as evidence that some cells have the plasmid. See Section 4.1 for details.

3.2 Model Setup

3.2.1 Assumptions

We begin with a description of the necessary assumptions. We accept that the processed data we receive accurately represents the experiments. When there are errors in lineage tracking and transconjugant detection, we have no good way of determining what actually occurred in the experiment. We take the data we receive as accurate unless it is incompatible with the experimental methods. (In particular, we preclude donor cells from becoming recipients or transconjugants and vice versa.) Additionally, we make the following biological assumptions:

- (i) Conjugation, maturation, and other cell properties are governed by the same rules for each type of cell.
- (ii) Every lineage that receives the plasmid begins expressing RFP within a fixed time range.
- (iii) Every lineage that receives the plasmid matures within a fixed time range.
- (iv) Only recipient cells can be conjugated to.
- (v) Conjugation events are not instantaneous.
- (vi) There is no plasmid loss.
- (vii) Conjugation only occurs between adjacent cells.

Assumptions (i) - (iv) are necessary for our model formulation, while assumptions (v) - (vii) are based on the specifics of the experimental data. The latter set could be changed for other experiments.

Assumption (i): Cell Properties. We use the same governing functions for each type of cell. This decision is justified by experimental evidence showing that all donor cells conjugate at the same rate [1, 2, 7]. These studies demonstrate that conjugation events are not due to a subset of the donor population. Note that this assumption does *not* mean every donor-recipient pair is equally likely to result in conjugation. The probability of conjugation occurring depends on the features of the cells and the quality of the contact.

Assumption (ii): Expression Delay Range. Cells that are conjugated to begin expressing RFP within a fixed time range. This time range is motivated both by biology and

by computational necessity. There is a minimum amount of time necessary for transcription and translation of genes contained on a plasmid. This time corresponds to the minimum expression delay. Requiring a minimum delay of one time step is realistic and ensures that all edges point forward in time. (If all edges point forward in time, the graph is guaranteed to be acyclic.) Likewise, it is reasonable to expect that there is maximum amount of time the process can take. The maximum delay is necessary because it allows us to be certain that many cells do *not* have the plasmid; any cell whose descendants remain green for a sufficiently long time is guaranteed not to have it. Without a maximum expression delay, any cell which contacted a mature cell would have a non-zero chance of having (and donating) the plasmid. Including these possibilities makes it difficult to formulate and evaluate queries. See Section 5.1.1 for a description of how we chose the range.

Assumption (iii): Maturation Delay Range. Cells that are conjugated to mature within a fixed time range. This assumption is the equivalent of assumption (ii) for the maturation process. In this case, the minimum delay corresponds to the time necessary for plasmid replication and the development of conjugative apparatus. It must be at least one time step to ensure that all edges point forward in time. The maximum delay allows us to be certain that transconjugants can eventually act as donors; any cell which received the plasmid sufficiently long ago is guaranteed to be mature. Without a maximum maturation delay, any transconjugant has a non-zero chance of being unable to act as a donor. Accounting for those possibilities makes it difficult to evaluate queries. See Section 5.1.2 for a description of how we chose the range.

Assumption (iv): Potential Recipients. Donors and transconjugants do not act as recipients. Because the only way to detect conjugation is through a change in cell color, there is no way to detect secondary conjugation events to donors and transconjugants. Thus, there is no way to validate the inclusion of secondary conjugation events. We only evaluate our models on probabilities related to the first time a lineage expresses RFP. See Section 3.3 for details.

Assumption (v): Time For Conjugation. Transferring a plasmid takes time. Conjugation has been observed to occur within about five minutes [1, 2], which corresponds to one time step in our experiments. We include this assumption as follows: suppose cell A and cell B are in contact in frame t . This contact has a probability of causing cell B's *descendants* in frame $t + 1$ to have the plasmid. This assumption is not necessary to guarantee the graph is acyclic as long as the maturation delay is at least one time step.

Assumption (vi): No Plasmid Loss. Every lineage that receives the plasmid will have it for the remainder of the experiment. We also assume that once a lineage expresses RFP or matures, it never loses that property. While plasmid loss is a known phenomenon

[5, 28, 37], we cannot detect it from the processed data. If a lineage stops expressing RFP, then it likely lost the plasmid. We have no way to detect if transconjugants stop expressing RFP. (Transconjugants are only flagged when they begin expressing RFP, and transconjugant status is then passed down the lineage.) Due to inconsistencies in the data, we assigned color information to cells based on their lineage. This leads to the same issue when attempting to detect plasmid loss in donor lineages. See Section 3.2.2 for details.

Assumption (vii): Adjacency during Conjugation. Only adjacent cells form mating pairs. Recall from Section 3.1 that the experiment uses a P-plasmid system. The pili are short, rigid, and likely unable to extend around an entire cell. Because the cells are tightly packed except during the start of the experiment, we constrain the maximum conjugation distance to be less than one cell width. While this may exclude a small number of potential contacts during the start of the experiment, it eliminates many unrealistic contacts later in the experiment. See Section 5.1.3 for a description of how we determine cell contact. Note that it is necessary to restrict the set of potential contacts for computational speed, but it is not necessary to restrict it specifically to adjacent cells.

We do not make any assumptions about green cells which die or fall out of the trap. Depending on their contacts, these cells have varying probabilities of having the plasmid while they are visible. We neither place restrictions on the number of times a cell can conjugate within one time frame nor include a recovery period after transfer. (Similarly, there is no restriction on the total number of conjugation events in each time frame.)

The second set of assumptions are related to the properties of cells at the start of the experiment. We use the following initial conditions:

- Green cells at the start of the experiment do not have the plasmid.
- Yellow cells in the first time frame are mature.
- Red cells in the first time frame are mature.

There is a delay between the setup of the microfluidic traps and the start of imaging, so it is possible for conjugation events to occur before imaging. These events should be relatively uncommon, but we have no way of measuring them. We assume no green cells have the plasmid in the first time frame. When a transconjugant is detected in the first time step, we assume it is already mature. (Expression and maturation occur on similar time scales.) Donor cells are grown well in advance, so they should be mature at the start of the experiment.

3.2.2 Handling Inconsistent Data

We discovered that some of the data was inconsistent with the experiment or the assumptions listed in the previous section. First, the color of some lineages changed in ways which are biologically impossible. Some recipient and transconjugant cells had children that were donor cells and vice versa. These cases are due to errors in lineage tracking and transconjugant detection. It does not make sense to include biologically impossible events in our modeling, and we have no way to correct the lineage tracking. So, we use the procedure described in Algorithm 1 to assign color information based on lineages. This approach ensures that our input is consistent with the experimental setup.

Algorithm 1 Setting cell color

```
for all cells  $i$  do
  if  $i$  has an ancestor  $j$  then                                     ▷ cells with ancestors
    if color( $j$ ) is red or yellow then
      color( $i$ )  $\leftarrow$  color( $j$ )                                     ▷ red and yellow always passed on
    else if color( $j$ ) is green then
      if  $i$  is flagged transconjugant then                             ▷ green may become yellow
        color( $i$ )  $\leftarrow$  yellow
      else
        color( $i$ )  $\leftarrow$  green
    else                                                               ▷ cells without ancestors
      if  $i$  is flagged transconjugant then
        color( $i$ )  $\leftarrow$  yellow
      else
        check fluorescence data
```

A second issue arises from assumption (ii), which imposes a time range for RFP expression after conjugation. Suppose a green cell divides into two children. If the first child changes color before the minimum expression delay, then we assume the parent has the plasmid. If the second child does *not* change color past the maximum expression delay, then we assume the parent does *not* have the plasmid. This scenario leads to a contradiction. (See Figure 3.3 for a representation of this situation.)

We have no way to correct errors in the data, so we only consider biological explanations when resolving this situation. There is a minimum time required to synthesize fluorescent proteins, so the color change is evidence that the parent cell received the plasmid. If we assume the parent has the plasmid, one explanation for the second lineage remaining green

is that it did not receive a copy during cell division. This possibility is particularly likely if the parent was conjugated to shortly before dividing and has not yet replicated the plasmid. Another possibility is that the second child is unable to express RFP. In either case, it is likely the green lineage will not cause detectable conjugation events. Breaking the lineage link between the parent cell and the green child is consistent with these scenarios and does not otherwise affect the data. (The only information passed along lineages is the probability that a cell has the plasmid and is mature.)

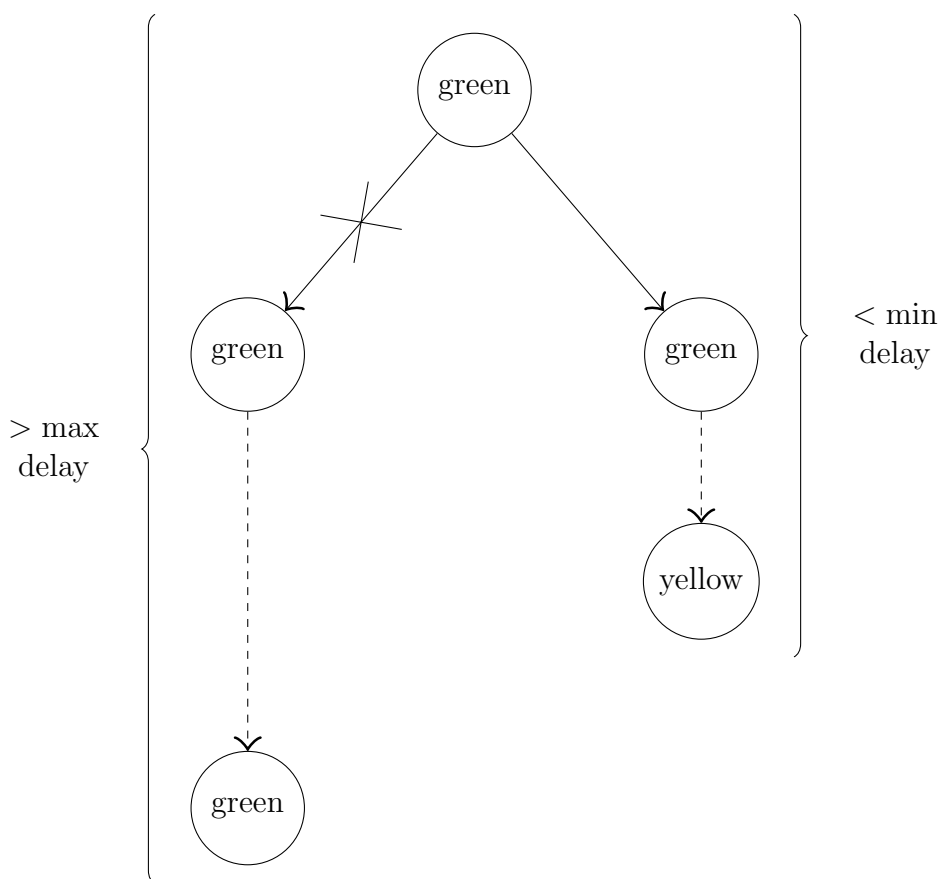


Figure 3.3: A cell splits into one lineage that changes color before the minimum expression delay and another lineage that does not change color past the maximum expression delay. This scenario contradicts assumption (ii), so we remove the link to the first child in the green lineage.

3.2.3 Graph Structure

At this point, we can see why a Bayesian network is an appropriate way to model our system. We have a large network of interdependent cells, each of which can be represented by a few key properties. We also have a complete understanding of how the properties affect each other; for instance, we can describe every potential contact between cells. It is natural to think of cell properties as variables governed by conditional probability distributions, where the distribution for each variable is conditioned on the other cells that directly interact with it. This description is equivalent to constructing a graph in which each node is a property and each edge is a direct causal effect. We get a directed graph which is factorized by conditional probability distributions (CPDs). Because all the edges represent causation, they point forward in time, guaranteeing that the graph is acyclic. This representation of our system is a Bayesian network. Each cell is affected by relatively few other cells, so the degree of the graph is computationally manageable.

We now describe the nodes and edges that make up the graph representation of one trial. The nodes correspond to the properties of each cell that are given as evidence or will be inferred. We use information about cell color to infer whether each cell has the plasmid. To determine which cells may have been conjugated to, we need to know which cells can act as donors. So, we also infer which transconjugants are mature. We include three nodes for each cell which represent whether it has the plasmid, whether it is mature, and what color it is. (While we have additional information about cells, such as their spatial positioning, these properties are neither evidence of conjugation nor values that must be inferred.)

Because we are working with discrete images, it is convenient to use the term ‘cell’ to mean ‘cell at a time point’ moving forward. For each cell i , our graph includes a triple of nodes (g_i, m_i, c_i) which represent whether it has the plasmid (gene), whether it is mature, and its color, respectively. Note that the fewer values each variable can take, the faster we can compute queries. It is clear that g_i and m_i are binary variables representing whether the cell has that property. While our cells can have three colors, additional structure from our problem allows us to use a binary variable for c_i . The color nodes provide evidence about whether cells contains the plasmid. Because we assume that donors have it throughout the trial, we will only check the value of color nodes that represent recipients and transconjugants. Of these two types of cells, only transconjugants express RFP. It suffices to let c_i indicate whether a cell is expressing RFP.

To simplify the construction of the graph, every type of cell is represented by the same set of nodes. The nodes for every cell i are given by:

- $g_i = \begin{cases} 0 & \text{cell } i \text{ does not have the plasmid (gene)} \\ 1 & \text{cell } i \text{ has the plasmid (gene)} \end{cases}$
- $m_i = \begin{cases} 0 & \text{cell } i \text{ is not mature} \\ 1 & \text{cell } i \text{ is mature} \end{cases}$
- $c_i = \begin{cases} 0 & \text{cell } i \text{ is not expressing RFP} \\ 1 & \text{cell } i \text{ is expressing RFP} \end{cases}$

The edges in our graph correspond to direct causal effects between nodes. The set of edges pointing into each node correspond to the variables its CPD is conditioned on. The weights of those edges depend on the exact probability distributions. We first describe which edges exist. The following list summarizes all direct implications, or ways that cells can gain each property:

- (i) The properties of a cell affect the properties of its direct descendant.
- (ii) Receiving the plasmid causes a cell's descendants to change color and mature.
- (iii) Conjugation events cause a cell to receive the plasmid.

When considering effects within a single lineage, it is convenient to use the convention that cell i has direct descendant $i + 1$. This notation allows us to associate the label with a time step. It only allows us to describe the case where each cell has a single descendant. In the case of cell division, the edges we describe are duplicated for each child cell. Duplicating the edges also allows each child lineage to be affected independently.

We start by describing how the properties of each cell affect its direct descendant. Each node for cell i directly impacts its counterpart in cell $i + 1$, as depicted in Figure 3.4. These edges pass on information about the state of a lineage through time.

Receiving the plasmid causes a change in the state of the lineage. Suppose cell i is the first in its lineage to have the plasmid. (In Figure 3.5, this is cell 0.) From assumption (ii), we know that its descendants will change color at some time between the minimum and maximum expression delay. From assumption (iii), we know that its descendants will mature at some time between the minimum and maximum maturation delay. The gene node for each cell has a direct causal effect on the color and maturation nodes of every descendant within the respective time ranges, as depicted in Figure 3.5.

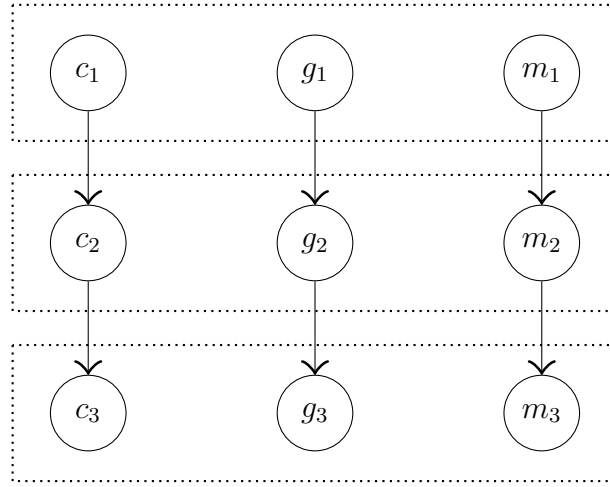


Figure 3.4: All edges representing the causal effect each cell has on its direct descendant. This diagram depicts all such edges, supposing that cell i is the direct ancestor of cell $i + 1$. Dotted boxes indicate nodes belonging to the same cell.

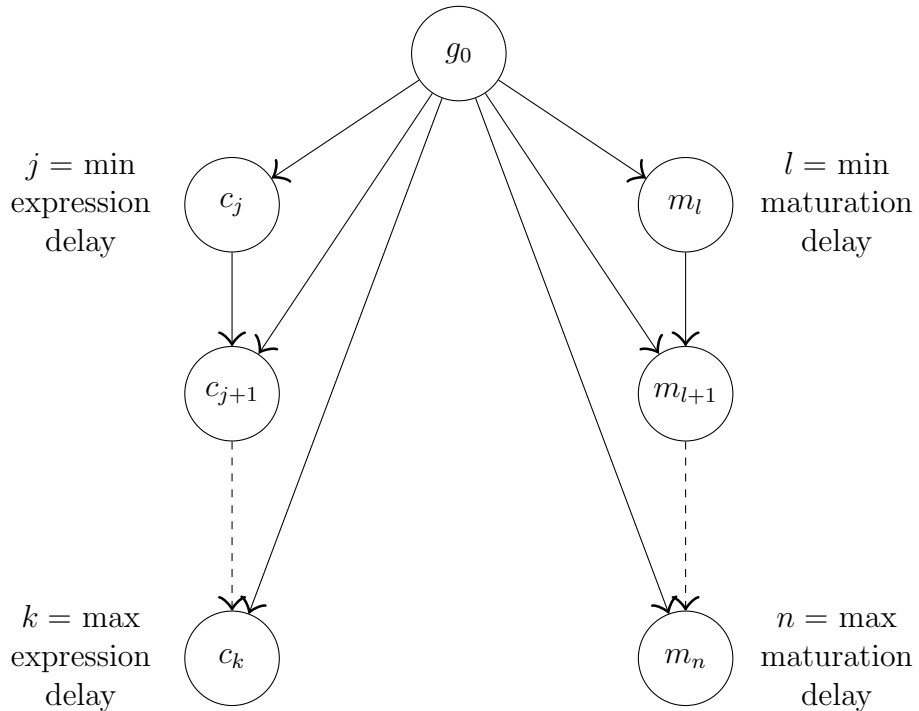


Figure 3.5: All edges pointing from a gene node to the color or maturation nodes of its descendants. Edges between consecutive color and maturation nodes are also depicted.

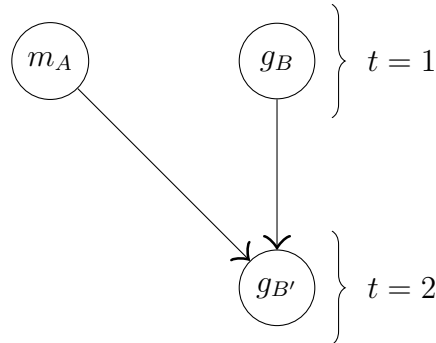


Figure 3.6: If cell A and cell B are in contact, there is an edge pointing from the maturation node of cell A to the gene node of B’s direct descendant B’. The edge from B to B’ is also depicted.

It remains to consider how conjugation is represented in the graph. Only mature cells can donate the plasmid, so edges representing conjugation point from maturation nodes into gene nodes. Because conjugation takes a few minutes, these edges point into the direct descendant(s) of the recipient cell, as depicted in Figure 3.6. To simplify the construction of the graph, these edges are added for every pair of cells which are in contact, regardless of their type. There are (at least) two edges added for every contact, including edges pointing from green cells into red cells.

3.2.4 Modeling Decisions

The goal of this thesis is to compare several models of the same process. The data and assumptions determine the graph structure for each trial. We vary the conditional probability distributions that determine the edge weights. Specifically, we consider the distributions governing the:

- (a) Expression Delay,
- (b) Maturation Delay, and
- (c) Contact Quality.

While assumptions (ii) and (iii) enforce a range on the expression and maturation delays, they do not prescribe the range itself or the probability distribution within it. The minimum and maximum delays determine which edges exist between gene and color or

maturation nodes, while the distributions determine the edge weights. We use the same ranges for each model but vary the distributions. Similarly, assumption (vii) restricts the set of possible conjugation events to adjacent cells. It does not prescribe the probability that a particular contact leads to conjugation. The contact quality function determines the weights of the edges from maturation nodes to gene nodes.

(a) Expression Delay. The time between a when a cell is conjugated to and when it begins expressing the genes on the plasmid appears to be highly variable. Manually observing experimental data provides some information on the range, as described in Section 5.1.1. These observations provide little information on the exact delays or their distribution. A cumulative distribution function (CDF) f representing this delay encodes how likely a cell is to start expressing RFP after a certain time. That is, $f(t)$ is the probability that a cell has started expressing RFP within t minutes of receiving the plasmid. We can choose any monotonically increasing function $f : \mathbb{R}^+ \rightarrow [0, 1]$ which satisfies

$$f(t) = \begin{cases} 0 & t < \text{min expression delay} \\ 1 & t \geq \text{max expression delay.} \end{cases}$$

(b) Maturation Delay. There is relatively little experimental evidence describing how long it takes for a cell to mature after receiving the plasmid. It would be interesting to investigate how long maturation takes and how variable the process is. If g is a CDF representing the maturation delay, then $g(t)$ is the probability that a cell has matured within t minutes of receiving the plasmid. We can again choose any monotonically increasing function $g : \mathbb{R}^+ \rightarrow [0, 1]$ which satisfies

$$g(t) = \begin{cases} 0 & t < \text{min maturation delay} \\ 1 & t \geq \text{max maturation delay.} \end{cases}$$

(c) Contact Quality. The quality of the contact between a mature donor cell and a recipient cell determines the probability that it leads to conjugation. The contact quality can account for both the relative spatial positioning of the cells and their individual traits. Our function choices are based on experiments which investigated what types of contact are mostly likely to facilitate conjugation.

We can use any function $h : (d, r) \rightarrow \mathbb{R}^{\geq 0}$, where d and r are the properties of the donor and recipient, respectively. While the value $h(d, r)$ should represent the probability that d conjugates to r , we lack the data to estimate it accurately. So, h is a measure of the *relative* likelihood of conjugation. If $h(d_1, r_1)$ is twice $h(d_2, r_2)$, this indicates that the

contact between d_1 and r_1 is twice as likely to lead to conjugation as the contact between d_2 and r_2 . (Because the values are relative, we normalize them when building the model and do not have to restrict the range to $[0, 1]$. See Section 4.3 for details.)

The exact functions we consider in this work are included in Section 5.1.

3.2.5 Conditional Probability Distributions & Edge Weights

We have described both the graph structure and the functions which determine the values of the CPDs. It remains to cover the formulation of the CPDs and how they determine the edge weights. Recall that a node represents whether a cell has a particular property, and the edges pointing into it represent the ways it can gain that property. So, each CPD represents multiple potential causes of the same effect. We only need to consider the probability that *at least one* parent node led the cell to gain a property. Because the variables are binary, we use noisy OR statements to define our CPDs, as described below.

A regular OR statement, such as $Y = X_1 \vee X_2 \vee \dots \vee X_n$, is true if at least one $X_i = 1$. (Each X_i is a binary random variable.) A noisy OR statement generalizes this idea to the situation where there is uncertainty. In a noisy OR statement, Y has a *probability* of being true if at least one $X_i = 1$. The graph representation of a noisy OR statement is depicted in Figure 3.7.

A regular OR statement is only false if $X_i = 0$ for all i . The probability that $Y = X_1 \vee X_2 \vee \dots \vee X_n$ is false is given by the product

$$P(Y = 0) = \prod_{i=1}^n P(X_i = 0).$$

The same idea works for a noisy OR statement. Our distributions are conditioned on the values of the parent nodes X_1, \dots, X_n . Given the values of each X_i , we take the product of the probability that each X_i failed to cause Y . There are two ways this can occur: either $X_i = 0$, or $X_i = 1$ but does not cause Y . The probability that a noisy OR statement is false is given by the product

$$P(Y = 0 | X_1, \dots, X_n) = \prod_{X_i=1} (1 - \alpha_i) = \prod_{i=1}^n (1 - \alpha_i X_i).$$

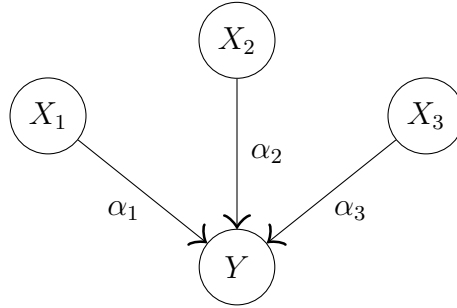


Figure 3.7: A graph representation of a noisy OR statement. Y has a probability of being true if at least one $X_i = 1$. The edge weights are given by $\alpha_i = P(Y = 1|X_i = 1)$.

We can make this simplification because

$$(1 - \alpha_i X_i) = \begin{cases} 1 & X_i = 0 \\ 1 - \alpha_i & X_i = 1 \end{cases}$$

is the probability that X_i does not cause Y . The probability that $Y = 1$ is found by taking the complement. It can be thought of as the expected value of Y . The simplification also allows us to calculate this expected value if we let $X_i \in [0, 1]$.

We formulate the CPD for every node in our graph using noisy OR statements. The values of each α_i depend on the functions we use to model the biological processes. Note that the weight of the edge $X \rightarrow Y$ is *not* always the value $P(Y = 1|X = 1)$. The weight will be different if there are multiple paths between X and Y . (Multiple paths correspond to multiple ways in which $X = 1$ can cause $Y = 1$.) To explain the edge weight calculations, we return to the convention that cell i has descendant $i + 1$. We again associate each label with a time step.

We begin with the edges from a cell to its direct descendant. The only path between a node in cell i and its counterpart in cell $i + 1$ is the edge directly connecting them. Edges from cell i to cell $i + 1$ represent the probability that cell $i + 1$ has a property because cell i does. There is no plasmid loss (or loss of expression or maturation), so this probability is always 1. We assign a weight of one to the edges

- $c_i \rightarrow c_{i+1}$,
- $g_i \rightarrow g_{i+1}$, and
- $m_i \rightarrow m_{i+1}$.

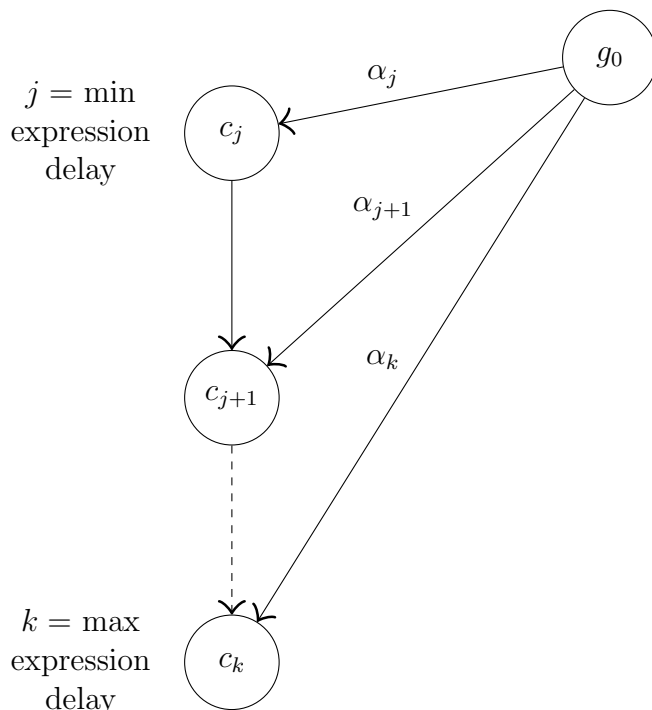


Figure 3.8: A reference for calculating edge weights from gene to color nodes.

We next consider the edges from gene nodes to color or maturation nodes. Since the construction of the edge weights is identical for both color and maturation, it suffices to describe the process for color nodes. Let $f(t)$ be the probability distribution representing the expression delay, j be the minimum expression delay, and k be the maximum expression delay. For clarity, we refer to Figure 3.8.

For each color node c_i , the probability that it expresses RFP due to g_0 receiving the plasmid equals $f(i)$. The probability that it does not express RFP despite g_0 receiving the plasmid is $1 - f(i)$. An equivalent characterization of this probability is that every path from g_0 to c_i fails to cause c_i to change color. (Failing refers to the case in which the first node in the path has value 1, but the last node has value 0.) This characterization is a noisy OR statement. For each node c_i , we set the probability from the noisy OR calculation equal to $1 - f(i)$. We then solve the equality corresponding to c_i for the edge weight α_i , as follows.

One key observation is that whenever c_i does not express RFP, it cannot cause c_{i+1} to express RFP. (This corresponds to the fact that the edge weight is 1.) For any $s \in \mathbb{N}$, the

probability that a path of the form $g_0 \rightarrow c_i \rightarrow \dots \rightarrow c_{i+s}$ fails is the probability that the edge $g_0 \rightarrow c_i$ fails, which is $(1 - \alpha_i)$.

We begin with c_j . The only path from g_0 to c_j is the edge $g_0 \rightarrow c_j$. The probability that it fails is $(1 - \alpha_j)$, so we get the equality

$$1 - f(j) = 1 - \alpha_j.$$

Solving this equality yields

$$\alpha_j = f(j).$$

There are two paths from g_0 to c_{j+1} : $g_0 \rightarrow c_j \rightarrow c_{j+1}$ and $g_0 \rightarrow c_{j+1}$. The probability that both paths fail is $(1 - \alpha_j)(1 - \alpha_{j+1})$, so we get the equality

$$1 - f(j+1) = (1 - \alpha_j)(1 - \alpha_{j+1}).$$

Solving this equality yields

$$\alpha_{j+1} = 1 - \frac{1 - f(j+1)}{1 - \alpha_j}.$$

Continuing inductively, the probability that all paths from g_0 to c_m fail is given by $\prod_{i=j}^{m-1} (1 - \alpha_i)$, so we get the equality

$$1 - f(m) = \prod_{i=j}^m (1 - \alpha_i).$$

Solving this equality yields

$$\alpha_m = 1 - \frac{1 - f(i)}{\prod_{i=j}^{m-1} (1 - \alpha_i)}.$$

This formula holds for $j \leq m \leq k$, which is the range of possible expression delays. It always gives $\alpha_k = 1$, corresponding to the fact that the cell must start expressing RFP by the maximum expression delay.

Finally, we consider the edges representing conjugation events. The edge $m_A \rightarrow g_{B'}$ exists whenever A is in contact with B , where B is the parent of B' . The only path between m_A and g'_B is the edge directly connecting them. So, the weight of the edge is the probability that cell A conjugated to cell B . This value is given directly by the contact quality function $h(A, B)$.

3.3 Model Comparison

So far, we described the construction of the graph representation and how each model determines the edge weights. We now consider the best way to compare different models. We formulate queries that provide insight into how accurately each model explains the observed results.

We know that a conjugation event occurred whenever a recipient lineage begins expressing RFP. The timing of the color change determines the set of cells which could have been conjugated to. Suppose a cell is green at time k and its descendant is yellow at time $k + 1$, as in Figure 3.9. If the maximum expression delay is k , then the lineage could not have received the plasmid at or before time 0, so $g_0 = 0$. If the minimum expression delay is j , then the lineage must have received the plasmid by time $k - j + 1$, so $g_{k-j+1} = 1$. Therefore, the lineage received the plasmid between time 1 and time $k - j + 1$. We call the set of gene nodes that may have received the plasmid the ‘critical region’ for a conjugation event. In Figure 3.9, this corresponds to the gene nodes g_1, \dots, g_{k-j+1} . (It is possible for the lineage to have been conjugated to at exactly time $k - j + 1$, so we include that cell in the critical region.)

There are many edges from maturation nodes pointing into each critical region. Each edge represents a contact between the lineage and a potential donor. We know that in the experiment, one of these contacts led to a conjugation event. We do not know which contact was successful. So, we evaluate each model based on the probability that *any* contact into the critical region led to conjugation. We query the model for the probability that the lineage was conjugated to within the critical region. This value measures how accurate the model is at determining when conjugation *does* occur. (Details of how we handle cells dividing during the critical region are included in Section 4.3.)

One downside of defining queries in this way is that different ranges for the expression delay lead to different queries. Model comparison is most meaningful when done query-by-query, so we use the same range for all models. Another factor that needs to be consistent for every model is the baseline probability of conjugation. We do not know the actual probability of conjugation between a given pair of cells. The contact quality functions are relative, and some produce higher values than others. To avoid biasing towards functions with higher values, we normalize the weights of the contact edges as described in Section 4.3. As a result, we do not need to consider the probability that lineages which remain green were *not* conjugated to. Models which assign more probability to edges outside of a critical region necessarily assign less probability to edges pointing into them. These models already perform more poorly on our queries.

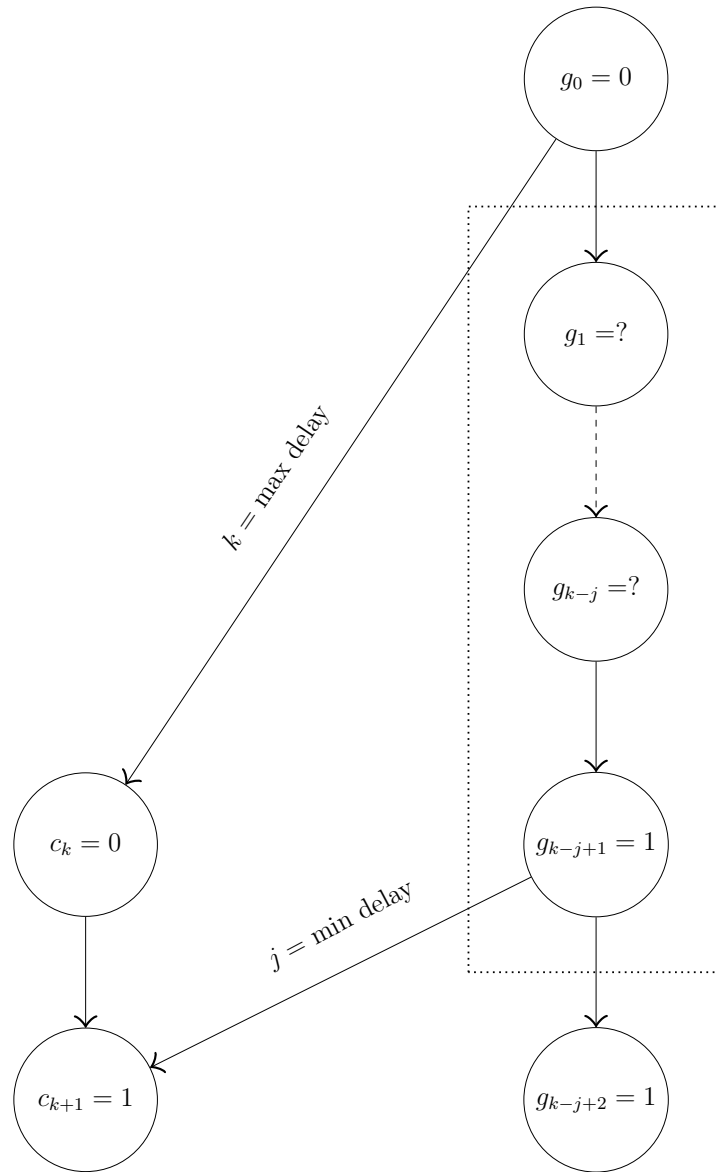


Figure 3.9: The critical region is the set of gene nodes that could have received the plasmid. It is determined by the time at which the lineage started expressing RFP.

Finally, we note that the individual probabilities generated by querying a single model are not directly meaningful. Each model includes a large number of low probability events. Even if the functions we choose are perfect descriptors of biological processes, the model should assign a low probability to any given event. Moreover, the weights of contact edges are normalized, so they are not direct measures of the probability of conjugation. Instead, the relative values are useful in comparing model formulations, as we next describe.

3.3.1 Model Ranking Systems

Each trial includes a large number of conjugation events. We need a metric which ranks each model based on its overall performance across the queries. We also assess which model is most accurate across multiple trials.

Recall that the result of a query corresponds to the probability that a lineage was conjugated to within the critical region. A better model will produce higher probabilities. One way to rank models is by comparing probabilities on a query-by-query basis. We found probabilities ranging in order of magnitude from 10^{-68} to 10^{-3} . As we next illustrate, this variation causes a typical approach, such as sum of squared errors, yields meaningless conclusions. Consider the scenario shown in Table 3.1. Here, model A is twice as accurate for Query 1 but model B is twice as accurate for Query 2. They should be considered equally good, but the sum of squared errors suggests that model A is better.

Model	Query 1	Query 2	SSE
A	0.01	0.2	1.6201
B	0.02	0.1	1.7704

Table 3.1: A scenario in which sum of squared errors leads to an arbitrary conclusion.

We developed three metrics for model comparison.

(i) Average Trial Ranking: Here, we use a group tournament ranking of the models. Suppose we have k models. Consider a single trial t with n queries. For each query, we rank the models based on the probabilities they return for it. The model with the highest probability has rank 1, and the model with the lowest probability has rank k . If two models tie for rank j , then both are ranked j and no model is ranked $j + 1$. The average rank for model m on trial t is

$$\mu_t(m) = \frac{1}{n} \sum_{i=1}^n \text{rank}_i(m),$$

where $rank_i(m)$ is the rank of model m on query i . The best model for the trial has the lowest average rank. We then order the models by their average rank over the trials. The average trial ranking for model m is

$$avg_T(m) = \frac{1}{l} \sum_{t=1}^l \mu_t(m),$$

where l is the number of trials. The best model has the lowest average. This metric is an ‘average of the average rankings.’ It weighs each trial equally when comparing models.

(ii) Average Query Ranking: Trials have different numbers of queries. Averaging the ranks by trial weighs individual queries differently. The more queries in the trial, the less weight is given to them individually. To weigh queries equally, we average across all the queries at once. Let n be the total number queries across all trials. The average query ranking for model m is

$$avg_Q(m) = \frac{1}{n} \sum_{i=1}^n rank_i(m),$$

where $rank_i(m)$ is the rank of model m on query i . The best model has the lowest average.

(iii) Query Probability Ranking: A different way to compare models is based on the total probability they assign to queries. For each trial, we rank the models from highest to lowest total probability. The model with the highest probability has rank 1, and the model with the lowest probability has rank k . If two models tie for rank j , then both are ranked j and no model is ranked $j + 1$. The query probability ranking for model m is

$$avg_P(m) = \frac{1}{l} \sum_{t=1}^l rank_t(m),$$

where $rank_t(m)$ is the rank of model m on trial t . The best model has the lowest average.

When using a ranking, we consider two ways of averaging the results: by trial (i) and in aggregate (ii). When using a ranking, it does not make sense to consider the average over multiple trials. The edge weights are normalized separately for each trial, and depend on the number of edges in the graph representation of the trial. Some trials will have overall lower probabilities than other trials. Therefore, we do not consider a version of (iii) in which the probabilities are averaged across all the trials at once.

Chapter 4

Model Implementation

The code required to implement and evaluate our models was written in python. This chapter summarizes the design decisions and major steps of building each model. The steps are

1. Initial Calculations and Data Processing,
2. Building the Graph Structure,
3. Query Setup, and
4. Query Evaluation.

The majority of these steps were described in Chapter 3, so this section focuses on filling in technical details and organizing the process. Details of the code implementation which are not relevant to the modeling problem are omitted. The code will be made available as part of the forthcoming manuscript.

4.1 Initial Calculations and Data Processing

The first step in building a graph representation is organizing the output from CellProfiler into a more appropriate format. CellProfiler outputs a single csv file for each trial, with rows corresponding to cells and columns corresponding to features. We begin by assigning a unique identifier (UID) to each cell at a time point. For each relevant feature, we create a

dictionary that uses the UIDs as the keys. Most values are taken directly from CellProfiler, but two features required additional calculations: cell color and potential neighbors.

As described in Section 3.2.2, we determined cell color based on lineage. Algorithm 1 describes how cell color is determined for cells with an ancestor. For cells in the first frame, or those without an ancestor, we use the red and green fluorescence levels to determine if it is a donor or a recipient.

We also generate a list of potential neighbors. The contact quality functions involve relatively intensive computations, so we first find potential neighbors. We only calculate the contact quality between two cells if their centers are within 15 microns. This range is well over twice the length of a cell, so it should not exclude any contacts.

The next step in the initial calculations determines which cells are certain to have the plasmid because their descendants change color synchronously, as described in Section 3.1.2. We check synchrony for each green cell which divides before the next frame, because we only know that the lineage received the plasmid sometime before that division event. For each such cell, we check if all of its descendants are first flagged as transconjugants within 5 frames of each other. (This value was recommended as a starting place by Aaron Yip, and other values will be considered in the future.) If so, we assume it has the plasmid. This information is added to our model as evidence in the next step. Adding evidence does not change what queries are asked, but it may simplify the queries by reducing the length of the critical range we have to consider.

Finally, we build function objects corresponding to each expression delay, maturation delay, and contact quality function. These functions represent the conditional probability distributions (CPDs) and are used to determine the edges and edge weights.

4.2 Building the Graph Structure

There are four steps involved in setting up the graph structure of a model: setting up the nodes, determining the edges, assigning the edge weights, and adding the evidence. To simplify this process, evidence is not added until *after* the graph structure is determined. We include the same set of nodes and edges for every type of cell, even if they are not used in our calculations. See Section 3.2.3 for details on how the nodes and edges are determined.

Setting up the nodes: For each UID, a corresponding color, gene, and maturation node are created.

Determining the edges: For each color, gene, and maturation node, an edge pointing into the corresponding node of its descendant is added to the graph. The edges from gene nodes to color and maturation nodes depicted in Figure 3.5 are also added. The dictionary of potential neighbors is used to determine which cells are (potentially) adjacent. The edges from maturation to gene nodes depicted in 3.6 are added.

Assigning the Edge Weights: The CPDs are added using the functions built in the first step. If a conjugation edge is found to have a weight of zero, it is removed from the model. (All other edges are guaranteed to have a non-zero weight.) Also, all edges pointing from a cell to its descendant are assigned weight 1.

Adding the Evidence: The two sources of evidence are color and synchrony. The color node of each red and yellow cell is set to 1, and the color node of each green cell is set to 0. Certainty about gene nodes is determined as follows:

- Suppose a gene node points into a color node that has value 0. If the nodes are separated in time by at *least* the maximum expression delay, then the gene node is set to 0.
- Suppose a gene node points into a color node that has value 1. If the nodes are separated in time by at *most* the minimum expression delay, then the gene node is set to 1.

At this point, we check for contradictions in the model, as described in Section 3.2.2, and remove edges via the process depicted in Figure 3.3. Next, we consider the cells which were determined to have the gene from synchrony. The gene node of these cells is also set to 1. Certainty about maturation nodes is then determined similarly:

- Suppose a gene node with value 0 points into a maturation node. If the nodes are separated in time by at *least* the maximum maturation delay, then the maturation node is set to 0.
- Suppose a gene node with value 1 points into a maturation node. If the nodes are separated in time by at *most* the minimum maturation delay, then the maturation node is set to 1.

4.3 Query Setup

There are three steps required to set up the queries: identifying the queries, normalizing the conjugation edge weights, and calculating a preliminary approximation of each query.

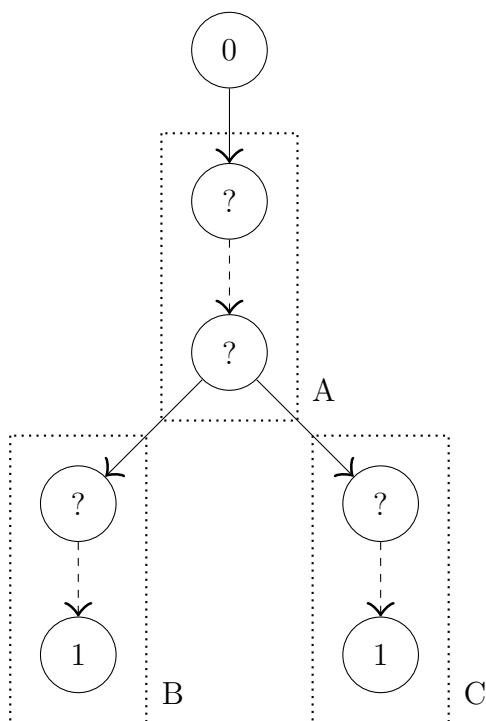


Figure 4.1: The critical region is split when a lineage divides. Dotted rectangles indicate the critical regions of the sub-queries.

Identifying the Queries: Each query is determined by the set of gene nodes comprising its critical region, as depicted in Figure 3.9. The start of a critical region is defined by a gene node with value 0 pointing into a gene node with unknown value. The end(s) of a critical region are defined by a gene node with unknown value pointing into a gene node with value 1. Because cells may divide within a critical region, we split queries into sub-queries at each cell division. One sub-query corresponds to the lifetime of an individual cell, as depicted in Figure 4.1. If the cells divide again, this process is repeated.

If a lineage within the critical range does not lead to a gene node with value 1, the corresponding sub-query is not considered. This situation occurs when a cell dies or falls out the trap. If no lineages within the critical range lead to a gene node with value 1, the entire query is disregarded. Green cells near the end of an experiment fall into this category. There are no future green cells to guarantee that they do not have the plasmid. Since there is no evidence in either of these situations, we do not calculate the probability that conjugation occurred.

A list of all sub-queries is generated, along with the set of sub-queries that belong to each original original critical region. Note that for every critical region, there is a corresponding set of maturation nodes whose values are uncertain. We also identify these sets of nodes because they are relevant when evaluating queries. Note that yellow cells near the end of the experiment may have uncertain maturation nodes.

Normalizing the Conjugation Edge Weights: We consider ‘relevant’ edges that correspond to contacts where we can assess whether conjugation occurred. These are edges into critical ranges and edges into gene nodes with value 0. The former are contacts that may have led to conjugation and the latter are contacts that could not have led to conjugation. The normalization factor is the sum of the weights of all relevant edges. The weight of each relevant edge is then scaled by the normalization factor. This process also ensures that all edge weights used in the following calculations are less than 1.

Calculating Naive Probabilities: There are conjugation edges pointing from uncertain maturation nodes to uncertain gene nodes. It is computationally infeasible to determine an assignment over all these nodes at once, so we compute a ‘naive’ approximation for each uncertain node. When evaluating a query, we use the naive probability for the maturation nodes from different lineages. Naive probabilities are calculated for uncertain gene nodes because they affect the uncertain maturation nodes.

There are two types of cells with uncertain maturation nodes. They either belong to a critical region of a query or to a cell that died, fell out the trap, or is near the end of the experiment. To simplify the following discussion, we refer to both situations as queries.

We call these probabilities naive because they only consider the parents of each node. For each query, we start at the first time step with an uncertain node. We use a noisy OR calculation to determine the probability that each uncertain node has value 1. This probability is the expected value of the node. We repeat the process for each consecutive time step, so that we always know the actual or expected value of the parent nodes.

Finally, we normalize the values for queries corresponding to critical regions. Conjugation is rare, so the naive probabilities are low until the last node in the critical region. Normalizing keeps the increase relative over time and ensures that the probabilities increase to one more smoothly. (We normalize to one because we know there was a conjugation event.) To normalize, we do the noisy OR calculation for the last node. We know the value of that node is one, so we use the expected value as the normalization factor. The value of each node in the query is divided by the expected value. If there are sub-queries, the normalization factor is the lowest value across all last nodes. The naive probabilities are then capped at 1.

One complication in this process is ‘linked’ queries. Queries A and B are ‘linked’ if

Query A points into Query B *and* Query B points into Query A. That is, an uncertain maturation node corresponding to Query A points into an uncertain gene node for Query B and vice versa. If either query is linked to other queries, the entire set is calculated together. For each set of linked queries, we do the following:

- (i) Compute the naive probabilities as if there were no edges between the linked queries. These values correspond to the probabilities that each received the plasmid from elsewhere.
- (ii) Recompute the naive probability for each query. Include the edges pointing into it from the linked queries and use the naive probabilities calculated in step (i). These values now include the probability that each query received the plasmid from a linked query, given that the linked query got it elsewhere.

This process does not generate cycles.

There is one situation in which this method can lead to a zero probability outcome. Suppose there are four queries A,B,C,D such that A and B are linked, B and C are linked, and C and D are linked. If the only way for Query D to get the plasmid is a path from A to B to C to D, the model returns probability zero for D. When calculating the first step, only Query A has a non-zero probability. When calculating the second step, Query B has a non-zero probability of receiving it from Query A. When evaluating the full queries, Query C has a non-zero probability of receiving it from Query B. There are no further computations, so Query D returns probability 0. (Note that a zero probability query corresponds to an observed conjugation event that our model cannot explain.)

4.4 Query Evaluation

In this section, we describe the process of evaluating the queries. A query is defined by a critical region of a single lineage. We refer to this lineage as the query lineage. It also involves nodes which impact the critical region or are impacted by the critical region. These nodes include ancestors and descendants of the cell in question and nodes within other queries' critical regions. A query computes probability of the combination of the following events:

- (i) The colors of the query lineage are as observed.
- (ii) The query lineage was conjugated to within the critical region.

Moreover, we evaluate the query based solely on information about interactions with surrounding cells. Information about the future of the query lineage that directly implies the event occurs is not used as evidence. (Including this evidence would result in every query evaluating to 1.) For instance, we do not condition on the value of color nodes pointed into from the critical region or the value of gene nodes past the critical region.

We first describe all relevant nodes. The subscript indicates whether they are evidence (E), query (Q), or hidden (H) variables. Recall that a conditional probability query evaluates the probability of an assignment to the query variables, given the values of the evidence variables. The hidden variables are marginalized out. (See Section 2.3 for a general description of queries.)

The following nodes from the query lineage are relevant. The edges between them are depicted in Figure 4.2.

- C_Q is the set of color nodes pointed into by any node from the critical region.
- G_Q is the set of gene nodes in the query region.
- G_E is the set of gene nodes before the query region that are certain to be 0.
- M_H is the set of maturation nodes corresponding to the query region.

C_Q is queried to determine the probability that the colors match the data, and G_Q is queried to determine the probability that the lineage was conjugated to during the critical region. G_E is evidence that the query lineage was not conjugated to before the critical region. M_H is hidden because we have no evidence to check their values against.

The following nodes are relevant because they are impacted by maturation nodes from the query lineage.

- Z_E is the set of gene nodes pointed into by any node in M_H that are certain to be 0.
- R_E is the set of gene nodes pointed into by any node in M_H that belong to a different lineage's critical region.
- P_E is the set of gene nodes pointed into by maturation nodes past M_H that belong to a different lineage's critical region.

Z_E is evidence that the query lineage was in contact with cells it did not conjugate to. It is indirect evidence that the query lineage was not yet mature. R_E and P_E are indirect evidence that the query lineage was mature and acted as a donor.

The following nodes from other lineages impact the query lineage.

- K_E is the set of gene nodes that point into G_Q and are certain.
- R_H is the set of gene nodes that point into G_Q and belong to a different lineage's critical region.

K_E and R_H are potential causes of the query lineage receiving the plasmid in the critical region. Nodes in K_E are certain, but may be either 0 or 1. R_H is hidden because we have to consider the impact of every possible assignment to it.

Finally, all other nodes U pointing into the relevant queries are treated as evidence. If the values of any of these nodes are uncertain, we use the naive probability calculation. Any other node in the graph is treated as independent. They are at least one query removed from the critical region and have minimal impact. Their values were also considered during the calculation of the naive probabilities.

To speed up the computations, we do not consider every node in R_E and P_E separately. For the purpose of the current query, it only matters *if* they are conjugated to. (The exact node in its critical region does not matter.) Each query in R_E and P_E is treated as a single node. Every edge that pointed into the critical region now points into it. The probability that the lineage was conjugated to is a single noisy OR calculation.

Another speedup is that we do not consider every 'possible' assignment to the critical region. Once the lineage receives the plasmid, it does not lose it. We only consider assignments that are zeroes followed by ones. If the critical region G_Q is the nodes (g_1, \dots, g_n) , we evaluate the probability of the assignments in the set

$$\{(0, \dots, 0, 1), (0, \dots, 1, 1), \dots (1, \dots, 1, 1)\}.$$

We refer to these as valid assignments. This approach reduces the possibilities we have to consider from 2^n to n . We take a similar approach for each other series of nodes (e.g. C_Q).

The probability evaluated for each query is

$$P(C_Q \in V_C, G_Q \in V_G | G_E = 0, P_E = 1, Z_E = 0, R_E = 1, K_E, U),$$

where V_C and V_G are the valid assignments for C_Q and G_Q , respectively.

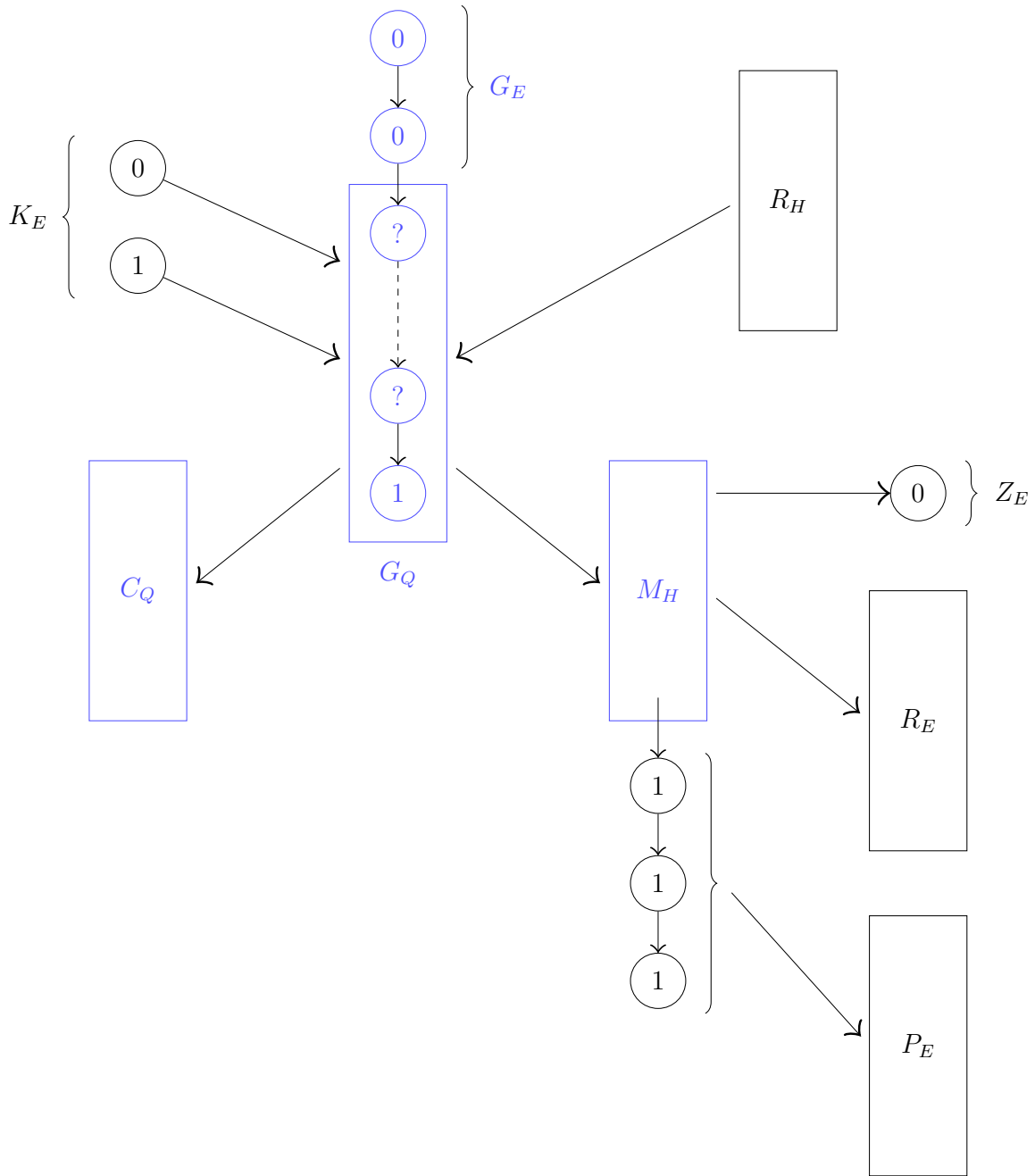


Figure 4.2: A depiction of all sets of nodes relevant for evaluating a query. Rectangles represent nodes corresponding to a critical region of a query. Blue nodes represent the query lineage. U is omitted for readability. See above for descriptions of each type of node.

Recall that some cells divide within their critical region. When evaluating the queries for cells that divided in the critical region, we consider each sub-query separately. We assume the cell does not have the plasmid before the start of the sub-query we are actively evaluating. When evaluating the model, we consider the entire critical range together. We compute the probability that each sub-lineage received the plasmid at some point in the critical region. For the scenario in Figure 4.1, there are two possibilities. Either

- (i) conjugation occurs in the critical region for sub-query A, or
- (ii) conjugation occurs in the critical regions for sub-queries B *and* C.

The total probability corresponds to $P(A) + P(\neg A) \cdot P(B) \cdot P(C)$. If there are more than two sub-queries, we generalize this process in the natural way.

This method combines multiple conjugation events into a single query. It is ideal for each query to represent a single event, so that all events are weighed equally when comparing models. Most split queries are the result of a single event because conjugation is rare. We include split queries because they are possible; there are cases in the data where it was more likely for B and C to be conjugated to. Overall, combining sub-queries should not significantly bias our comparisons.

Chapter 5

Models & Results

5.1 Models

This section describes and motivates the specific models we consider. As described in Section 3.2.4, there are three aspects of each model that we can vary: the expression delay, the maturation delay, and the contact quality. This work serves as a proof of concept, so we test every combination of a small set of functions. We chose two expression delay functions, two maturation delay functions, and three contact quality functions for a total of 12 models. We plan to refine our functions after the lineage tracking software is complete. (Results from these models are less meaningful because we had to make corrections to the data due to inconsistencies, as described in Section 3.2.2.)

5.1.1 Expression Delay

There are two main considerations for the expression delay: the range of possible delays and the distribution within that range. Including a larger range allows us to assess more possibilities with our model by increasing the critical region for the queries. However, it also makes the queries more computationally expensive. We chose a relatively large range, but constrain it based on the evidence we have. Because it determines the critical regions of the queries, we use the same range for both functions.

To determine approximate bounds for the expression delay, we manually observed the imaging data. We assume that a cell was conjugated to if all of its descendants change color at roughly the same time and at least one of its sibling's descendants does not change

color for a long time. The first condition suggests that all of its descendants received the plasmid at the same time. Because conjugation is a rare event, one event involving a common ancestor is more likely than multiple events occurring at nearly the same time. The second condition suggests that the cell's parent did not receive the plasmid. (If it had, we expect the sibling's descendants to change color at the same time.) If a cell changes color but its sibling does not, the same logic suggests that it received the plasmid directly. This method does not require us to hypothesize about which cell acted as a donor.

Once we identify the cell that was conjugated to, we use its lifetime to bound the potential expression delay. The maximum delay is the time between when the cell is born and when its last descendant changed color; the minimum delay is the time between when the cell divided and when its first descendant changed color. For consistency, we used Ilastik's output to judge when a cell became a transconjugant.

For instance, in one trial a cell lived from frame 14 to 28. Its two children were flagged as transconjugants at frames 30 and 32. Its sibling had descendants which were not flagged past time frame 40. We calculate that the expression delay was between 2 to 18 time frames, or 10 to 80 minutes. Another cell was born at frame 21 and was flagged at frame 36, but its sibling was not flagged for many more frames. The expression delay for this cell was no more than 75 minutes. A summary of all the events we considered are included in Table 5.1.

The range we use for the expression delay should overlap with every observed range, and ideally would include the maximum delay for most events. Our minimum delay must be shorter than the lowest maximum observed (50 minutes) and our maximum delay must be longer than the highest minimum observed (80 minutes). Note that the observed ranges are likely to be underestimations of the actual delay. If a lineage begins expressing red fluorescent protein (RFP) shortly after being conjugated to, there are fewer descendants to consider. The more descendants that have to change color, the more likely it is for at least one to have a variable delay. We are more likely to catch events with shorter delays.

We chose a minimum expression delay of 30 minutes because it is well below the maximum delays observed. We chose a maximum expression delay of 150 minutes because it is greater than nearly all the observed maximums, and is well above the average (92 minutes).

# Cells	Minimum Delay	Maximum Delay
1	25	50
1	–	55
2	10	65
1	–	70
1	–	70
1	–	75
1	–	75
2	15	75
2	40	75
2	25	85
4	50	80
2	10	90
3	45	90
2	20	95
2	20	100
1	–	105
4	60	115
3	45	115
2	60	135
6	80	150
1	–	155

Table 5.1: Estimations of the expression delay (in minutes) from manually observing imaging data. Each row corresponds to one event, and the number of cells indicates how many descendants lit up concurrently. The average maximum delay is 92 minutes.

We now consider the distribution within this range. A natural starting point is a normal distribution. We chose a wide range and expect delays near the extremes to be less likely. The middle of our range is 90 minutes. To include two standard deviations above and below the average into our range, we need a standard deviation of 30 minutes. Our first expression delay cumulative distribution function (CDF) is given by

$$f_1(t) = \begin{cases} 0 & t < 30 \\ \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{t - 90}{30\sqrt{2}} \right) \right) & 30 \leq t < 150 \\ 1 & t \geq 150. \end{cases}$$

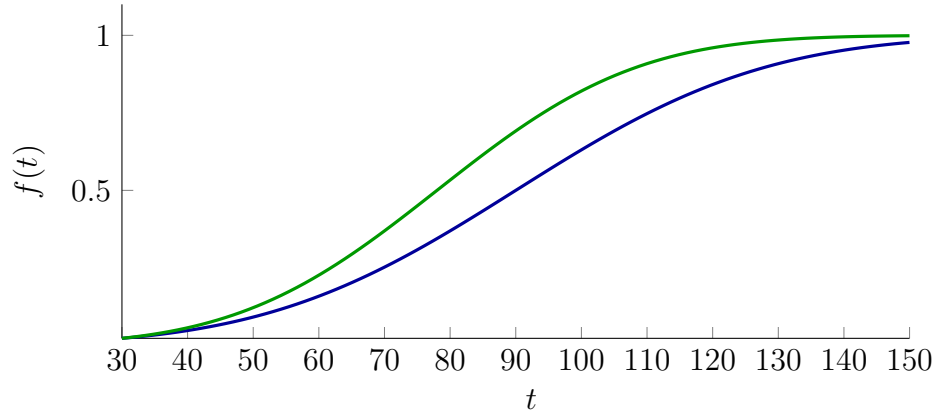


Figure 5.1: The two cumulative distribution functions we use for the expression delay. f_1 is blue and f_2 is green.

Although we expected our observations to be underestimates, as explained above, an average delay corresponding to the average maximum observed delay may be too high. We consider a second normal distribution with a lower average. Because we used a high maximum delay, we do not expect a lot of events to occur towards the upper end. So, we consider a distribution which is ‘skewed’ towards shorter delays. One way to do this within the predefined range is to include two standard deviations below the average and three above it. For our range, this results in an average of 78 minutes and a standard deviation of 24 minutes. Our second expression delay CDF is given by

$$f_2(t) = \begin{cases} 0 & t < 30 \\ \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{t - 78}{24\sqrt{2}} \right) \right) & 30 \leq t < 150 \\ 1 & t \geq 150. \end{cases}$$

A comparison of the two expression delay functions is shown in Figure 5.1.

5.1.2 Maturation Delay

For maturation, we again consider both the range of possible delays and the distribution within that range. Recall that the maturation delay is the time between when a cell receives the plasmid and when its descendants are able to donate it. As with the expression delay, we wish to assess as many possibilities as is computationally reasonable. The maturation

delay does not affect the critical range of the queries, and choosing a shorter delay for maturation does not exclude any conjugation events. So, we chose a shorter range for the maturation delay.

It is difficult to estimate the maturation delay from our experiments. We were only able to identify one case in which a transconjugant was guaranteed to act as a donor shortly after receiving the plasmid. In one trial, a colony of recipients first came into contact with donors in frame 22. Two adjacent cells were flagged as transconjugants in frame 33, but only one was ever in contact with a donor cell. Because the transconjugants were not siblings, this scenario suggests the following sequence of events: a donor conjugated to the adjacent recipient, which then conjugated to the other recipient. (While it is uncommon for a transconjugant to change color at the same time as a cell it donated to, the variance in the expression delay makes this situation possible.) Both events happened within a 55 minute window, so we surmise that a cell can mature within 55 minutes. This observation gives us a rough idea of how long maturation takes, and is consistent with results from the literature [1, 2, 7].

Other than this observation, we rely on estimates from the literature. We chose distributions corresponding to the results from studies on the kinetics of conjugation. One study using F-plasmid systems estimated a delay of 40-90 minutes [7]. Another study of both F and P-plasmid systems estimated a delay of 40-80 minutes [1]. Because our range imposes a strict minimum, we use a slightly larger range of 30-90 minutes. We fit a normal distribution to this range, including two standard deviations above and below the mean (60 minutes). Our first maturation delay CDF is given by

$$g_1(t) = \begin{cases} 0 & t < 30 \\ \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{t - 60}{15\sqrt{2}} \right) \right) & 30 \leq t < 90 \\ 1 & t \geq 90. \end{cases}$$

A third study of a P-plasmid system found that ‘most’ cells were mature by about 40-45 minutes [2]. We chose a second distribution that has a mean of 45 minutes. The paper did not provide a range, so we decided to keep the same range of 30-90 minutes for consistency. (Using the same range ensures that the same conjugation events are possible in each version of our model.) Keeping a standard deviation of 15 minutes allows us to include one standard deviation below the mean and three above it in this range. Our

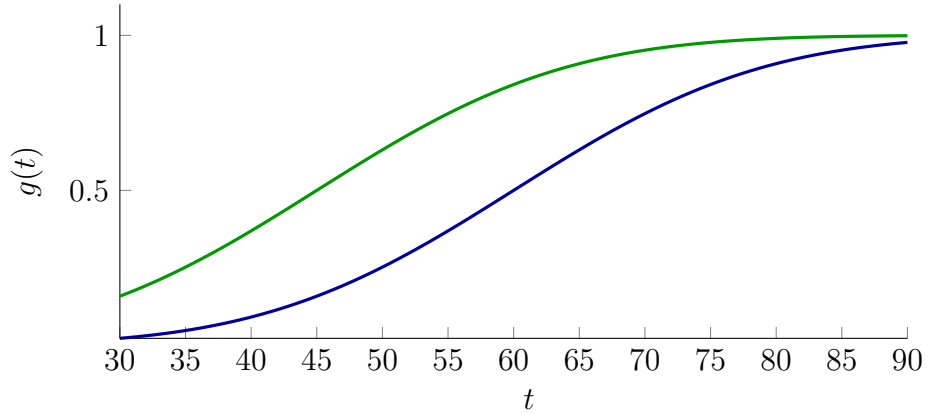


Figure 5.2: The two cumulative distribution functions we use for the maturation delay. g_1 is blue and g_2 is green.

second maturation delay CDF is given by

$$g_1(t) = \begin{cases} 0 & t < 30 \\ \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{t - 45}{15\sqrt{2}} \right) \right) & 30 \leq t < 90 \\ 1 & t \geq 90. \end{cases}$$

A comparison of the two maturation delays is shown in Figure 5.2.

5.1.3 Contact Quality

The contact quality represents how likely conjugation is to occur between two cells. At this stage, we consider only the spatial positioning of the cells. We plan to include other factors, such as growth rate, as we refine our models.

Recall that conjugation can occur between cells which are not physically touching because the donor cell extends a pilus to the recipient. We consider any potential donor-recipient pair where the recipient is within some ‘contact range’ of the donor. Conjugation is not symmetric. A contact quality function $h : (A, B) \rightarrow [0, 1]$ represents the probability that A conjugates to B . It does *not* account for the probability that A is mature, as that probability is determined separately. Suppose there are two green cells in contact, each of which may have the plasmid. In this case, we will calculate both $h(A, B)$ and $h(B, A)$.

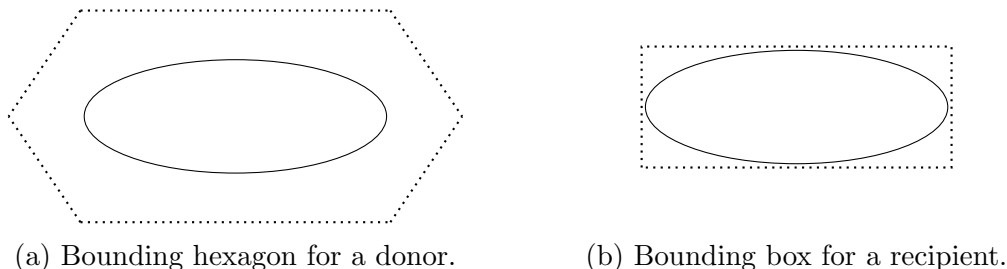


Figure 5.3: Bounding polygons used when determining contact quality.

Because we are working with 2D imaging data, each cell is represented by a 2D shape. In the data we receive, each cell is represented by a fitted ellipse and specified by its endpoints and diameter (minor axis). It would be ideal to use ellipses when checking which cells are in contact, but calculations involving ellipses are computationally expensive. We fit a polygon around each cell, as shown in Figure 5.3. We use a hexagon to represent the contact range of each potential donor, since it provides a good estimation of an elliptical range. Because we do not want to include a range around a potential recipient, we represent it with a tight bounding box. (This decision was consistent with the recommendations provided by Aaron Yip.)

The first contact quality function we consider assigns the same probability p to every contact. This function is a useful baseline because it only assesses if two cells are in contact. Note that the value of p is not important because it is normalized during model construction. The function is given by

$$h_1(A, B) = \begin{cases} p & B \text{ is within contact range of } A \\ 0 & \text{otherwise.} \end{cases}$$

That is, $h_1(A, B) = p$ if the hexagon representing A intersects the box representing B , and zero otherwise.

The second contact quality function we consider assigns probability based on the area of B within the contact range of A . This approach is a standard way of considering the contact quality. It is correlated with both how close and how aligned two cells are. Since the values are normalized during model construction, we let $h_2(A, B)$ equal the area of the box representing B which is within the hexagon representing A . See Figure 5.4a for a visual depiction of this function.

The last contact quality function we consider assigns probability based on the boundary of B within the contact range of A . Because the pilus connects to the surface of the

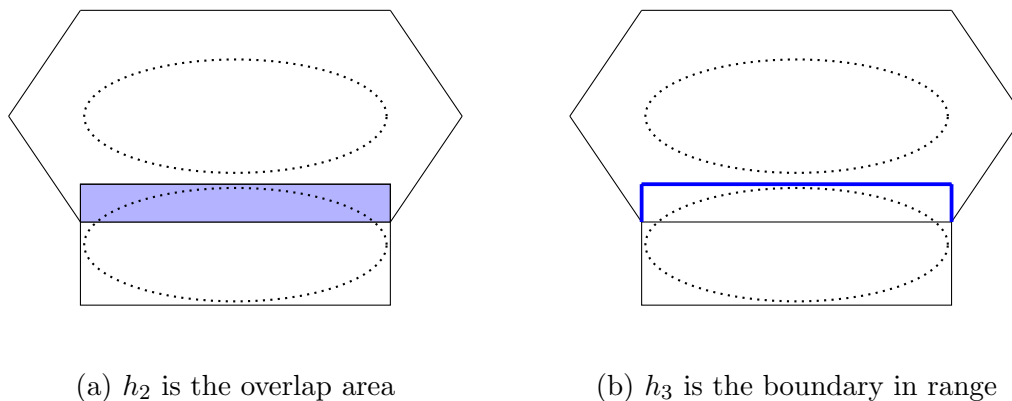


Figure 5.4: Depictions of the contact quality functions, with the overlapping area or boundary in blue.

recipient, it might be more accurate to consider how much of B 's surface is within range of A . We let $h_3(A, B)$ equal the boundary of the box representing B which is within the hexagon representing A . See Figure 5.4b for a visual depiction of this function.

We currently use the same contact range of $0.4 \mu\text{m}$ for every metric. It is less than the diameter of one cell (approximately $0.65 \mu\text{m}$) to ensure that only adjacent cells are considered to be in contact. It also ensures that h_3 only accounts for the boundary of the recipient cell that is nearest the donor cell.

5.2 Results

We tested our models on 6 different trials from the same experiment. Each trial corresponds to the time-lapse images taken of a single trap from the experiment. We chose trials that had a larger number of transconjugants, so that there are more events to query. The number of cells and ratio of cell types vary between trials, as described in Table 5.2. (Note that Trial 4 had one yellow cell in the first frame. No other trials had yellow cells in the first frame.) While all trials ran for the same length of time, frames from the end of the trial were omitted if the image quality deteriorated. The number of frames used from each trial is included in Table 5.2.

Trial	Number of Frames	Start			End			
		% Red	% Green	Cells	% Red	% Green	% Yellow	Cells
1	266	59	41	93	0	95	5	1152
2	290	66	34	136	17	62	21	919
3	198	66	34	321	23	53	24	1020
4	198	70	30	240	32	11	57	930
5	197	73	27	59	33	59	8	1001
6	197	46	54	37	0	75	25	1103

Table 5.2: Information about each experimental trial. Cell color was determined using Algorithm 1.

Trial	Number of Queries	Number of Zeroes	Minimum Probability	Maximum Probability
1	194	18	1.13e-35	2.18e-4
2	175	10	3.15e-36	2.74e-4
3	293	16	2.46e-31	6.33e-4
4	175	14	1.04e-31	1.97e-3
5	85	3	4.45e-68	1.72e-4
6	101	10	1.00e-39	3.24e-4

Table 5.3: Information about the queries for each trial. The minimum excludes queries that returned 0. All values are rounded to three significant digits.

Recall that each query corresponds to an observed conjugation event. The total number of queries in each trial ranged from 85 to 293. Between four and ten percent of the queries in each trial had zero probability. Table 5.3 includes the number of queries and the range of probabilities for each trial.

A zero probability query means there was an observed event our models cannot explain. We investigated some of these events to determine why our model missed them. When comparing the data to the images, we found that some zero probability queries were due to errors in transconjugant detection and lineage tracking. Our model identified multiple queries corresponding to conjugation events that were artifacts of errors in the data.

Each query corresponds to the first time a cell is flagged as a transconjugant by Ilastik. We found that ‘first’ cell in the imaging data, and tracked its lineage forward and backward in time. Some of the first cells from zero probability queries never visibly changed color. They were likely mislabeled.

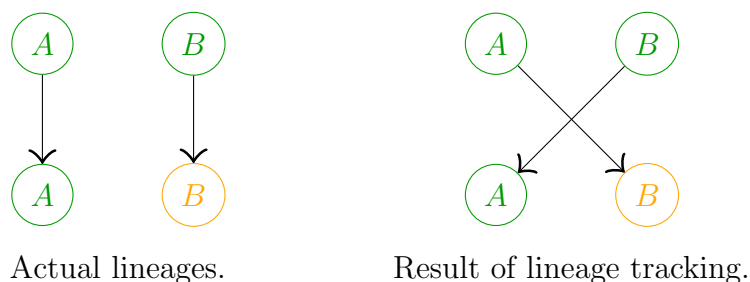


Figure 5.5: A tracking error which can lead to a zero probability query.

When a transconjugant is flagged, the expression delay determines a set of its ancestors that could have received the plasmid. Some first cells did not belong to the same lineage as the last ancestor that could have been conjugated to. There was a tracking error between the time the lineage received the plasmid and the time it was flagged. One case which leads to this type of error is depicted in Figure 5.5.

We manually checked ten zero probability queries from the first trial to estimate the number of zeroes caused by each issue. Three were caused by incorrectly flagged transconjugants and five were caused by tracking errors. Two more queries involved both incorrectly flagged transconjugants and tracking errors. One of those two queries also could have been caused by synchrony. All descendants of one cell were flagged over three hours in the future (one was from the wrong lineage and one was incorrectly flagged). To avoid this problem, we will modify synchrony to ignore cases in which the common ancestor is too distant.

5.2.1 Rankings

We rank our models according to the three metrics described in Section 3.3.1. The results of the Average Trial ranking are shown in Table 5.4, the results of the Average Query Ranking are shown in Table 5.5, and the results of the Query Probability Ranking are shown in Table 5.7. We also include the total probability assigned to queries in Table 5.6. (It was used to calculate the Query Probability Ranking.) Finally, Table 5.8 compares how the different metrics ranked the models. All values are rounded to two decimal places.

We use the convention `Contact_Expression_Maturation` to label each model. We refer to the functions described in Section 5.1 as follows.

- `Int` refers to h_1 , which checks if the bounding polygons intersect.
- `Area` refers to h_2 , which calculates the overlap area of the bounding polygons.

- Bound refers to h_3 , which calculates the boundary of the recipient within contact range of the donor.
- Norm refers to f_1 and g_1 , which are normal distributions for expression and maturation, respectively.
- Skew refers to f_2 and g_2 , which are shifted normal distributions for expression and maturation, respectively.

Based on the Average Trial Rankings in Table 5.4, the best model uses intersection to determine contact quality and skewed distributions for both the expression and maturation delays. Across different distributions, intersection is consistently ranked higher than boundary and boundary is consistently ranked higher than area. Likewise, skewed distributions for both delays are consistently ranked higher than normal distributions. The maturation delay has the greatest impact on the ranking; the top 6 models are all skewed.

The Average Query Rankings in Table 5.5 agree with the Trial Query Rankings. This result is expected because the trials are all part of the same experiment. The dynamics should be the same across the different trials, so the models should perform the same across all the queries. Weighing the queries unequally will not affect the overall ranking if the models perform the same across all of them. This result demonstrates that our models are consistent.

The total probability assigned to queries by each model is shown in Table 5.6. Because the edge weights are normalized, we expect these values to vary. They range from $3.14 \cdot 10^{-4}$ to $9.02 \cdot 10^{-3}$. We use these probabilities to calculate the rankings in Table 5.7. Intersection is again the best contact quality function, and the skewed distributions are again the best delay functions. The most impactful factor remains the maturation delay.

The comparison shown in Table 5.8 shows that all the ranking systems produce similar results. This similarity indicates that the models which perform best on a query-by-query basis also assign the highest total probability to queries. This result is a good consistency check because a good model should perform well on both metrics. (Consider a model which assigns high probability to a few events. It would do well in the Query Probability Ranking but poorly in the query-by-query rankings.)

Model	Trial						Average
	1	2	3	4	5	6	
Int_Skew_Skew	2.25	1.94	1.93	2.07	2.50	2.43	2.19
Bound_Skew_Skew	2.92	2.69	2.91	2.86	2.93	2.58	2.82
Int_Norm_Skew	3.11	2.78	2.74	2.92	3.36	3.35	3.04
Bound_Norm_Skew	3.79	3.52	3.73	3.71	3.80	3.53	3.68
Area_Skew_Skew	4.20	4.75	4.61	4.46	3.96	3.67	4.28
Area_Norm_Skew	5.07	5.59	5.43	5.31	4.81	4.55	5.13
Int_Skew_Norm	7.11	7.02	7.04	6.94	7.92	7.57	7.27
Bound_Skew_Norm	7.94	7.88	8.11	7.90	8.27	7.78	7.98
Int_Norm_Norm	7.95	7.84	7.87	7.79	8.80	8.48	8.12
Bound_Norm_Norm	8.80	8.71	8.93	8.76	9.14	8.68	8.84
Area_Skew_Norm	9.06	9.85	9.71	9.34	9.25	8.84	9.34
Area_Norm_Norm	9.90	10.67	10.53	10.18	10.08	9.72	10.18

Table 5.4: The average ranking of each model across all the queries in a single trial. The last column is the Average Trial Ranking.

Model	Trial						Sum	Average
	1	2	3	4	5	6		
Int_Skew_Skew	437	339	565	363	210	245	2159	2.11
Bound_Skew_Skew	567	471	852	501	246	261	2898	2.83
Int_Norm_Skew	604	486	804	511	282	338	3025	2.96
Bound_Norm_Skew	735	616	1093	649	319	357	3769	3.68
Area_Skew_Skew	815	832	1350	780	333	371	4481	4.38
Area_Norm_Skew	983	978	1592	930	404	460	5347	5.23
Int_Skew_Norm	1379	1228	2063	1215	665	765	7315	7.15
Bound_Skew_Norm	1541	1379	2375	1383	695	786	8159	7.98
Int_Norm_Norm	1542	1372	2307	1364	739	856	8180	8.00
Bound_Norm_Norm	1708	1524	2617	1533	768	877	9027	8.82
Area_Skew_Norm	1758	1724	2844	1634	777	893	9630	9.41
Area_Norm_Norm	1920	1867	3086	1782	847	982	10484	10.25

Table 5.5: The sum of the rankings for each model over all the queries in each trial. The last column is the Average Query Ranking.

Model	Trial					
	1	2	3	4	5	6
Int_Skew_Skew	0.81	1.48	6.07	7.30	0.75	0.71
Int_Norm_Skew	0.80	1.45	5.98	7.17	0.73	0.70
Bound_Skew_Skew	0.79	1.44	5.28	7.52	0.71	0.89
Area_Skew_Skew	0.80	1.13	4.70	9.02	0.59	0.91
Bound_Norm_Skew	0.78	1.42	5.21	7.38	0.70	0.88
Area_Norm_Skew	0.79	1.11	4.63	8.86	0.59	0.90
Int_Skew_Norm	0.56	0.91	4.00	4.04	0.36	0.41
Bound_Skew_Norm	0.56	0.89	3.36	3.83	0.38	0.58
Area_Skew_Norm	0.58	0.68	2.96	3.99	0.32	0.61
Int_Norm_Norm	0.55	0.89	3.95	3.97	0.36	0.41
Bound_Norm_Norm	0.55	0.87	3.32	3.76	0.37	0.58
Area_Norm_Norm	0.58	0.66	2.93	3.91	0.32	0.61

Table 5.6: The total probability assigned to queries by each model. All values are scaled by 10^3 for readability.

Model	Trial						Average
	1	2	3	4	5	6	
Int_Skew_Skew	1	1	1	5	1	5	2.33
Int_Norm_Skew	2	2	2	6	2	6	3.33
Bound_Skew_Skew	5	3	3	3	3	3	3.33
Area_Skew_Skew	3	5	5	1	5	1	3.33
Bound_Norm_Skew	6	4	4	4	4	4	4.33
Area_Norm_Skew	4	6	6	2	6	2	4.33
Int_Skew_Norm	10	7	7	7	9	11	8.50
Bound_Skew_Norm	9	9	9	11	7	9	9.00
Area_Skew_Norm	7	11	11	8	11	7	9.17
Int_Norm_Norm	12	8	8	9	10	12	9.83
Bound_Norm_Norm	11	10	10	12	8	10	10.17
Area_Norm_Norm	8	12	12	10	12	8	10.33

Table 5.7: The ranking of each model for each trial using the total probability assigned to queries. The last column is the Query Probability Ranking.

Model	Average Trial	Average Query	Query Probability
Int_Skew_Skew	1	1	1
Bound_Skew_Skew	2	2	2
Int_Norm_Skew	3	3	2
Bound_Norm_Skew	4	4	5
Area_Skew_Skew	5	5	2
Area_Norm_Skew	6	6	5
Int_Skew_Norm	7	7	7
Bound_Skew_Norm	8	8	8
Int_Norm_Norm	9	9	10
Bound_Norm_Norm	10	10	11
Area_Skew_Norm	11	11	9
Area_Norm_Norm	12	12	12

Table 5.8: A comparison of the different ranking systems.

Finally, we discuss the biological significance of these results. The skewed distributions have lower averages than the normal distributions, suggesting that RFP expression and maturation occur towards the lower end of the proposed range. However, the model construction may favor an earlier average. The probability of each contact donating the plasmid is the product of two values: the contact quality and the probability the donor is mature. If cells mature earlier, the overall probability of conjugation is greater. This increase only benefits the model if the increased probability corresponds to critical regions. We will investigate this possibility and adjust the model formulation or comparison if necessary. One possible solution is changing the normalization for edge weights. We could account for the probability each edge corresponds to a mature node when determining the normalization factor. (See Section 4.3 for a description of the current normalization.)

The most interesting result is the ranking of the contact quality functions. There is experimental evidence that the type of contact between cells impacts the probability of conjugation [6, 12, 21, 34]. Our models are somewhat consistent with this observation. The boundary function approximates surface contact more precisely than area, and it performs at least as well as area across all distributions. However, we expected both boundary and area to perform better than intersection. (Intersection does not measure the quality and was intended to be a baseline.) One explanation is that donors ‘select’ recipients based on other features. One study investigated conjugation of F-like plasmids to multiple recipient species [11]. They found that donors select recipients based on proteins expressed on the cell surface. While our experimental setup is different, we may be observing a

similar phenomenon. Recipient cells may have different concentrations of surface proteins. Alternatively, studies suggest factors including growth rate, the stage of the cell cycle, and the relative orientations of cells affect conjugation frequency [6, 12, 21, 34]. We will test this possibility by including additional factors in future iterations of our models.

Chapter 6

Conclusions & Future Plans

In this thesis, we presented a proof of concept for a model comparison approach to investigating the mechanisms underlying bacterial conjugation. We argued that a Bayesian network is an appropriate model for the data from our experiments. We demonstrated how such a model is constructed and explored variations that account for different mechanisms. Our models behave consistently across multiple trials and methods of evaluating them. The results suggest that recipients are selected based on features as well as contact quality with the donor. They also indicate that shorter expression and maturation delays are more likely than longer ones.

6.1 Future Plans

Once the image processing refinements are complete, we plan to reconstruct the models presented in this work. Improved cell tracking and transconjugant detection should improve the accuracy of our models and result in fewer zero probability queries. These improvements will also allow us to use color evidence directly from the data, including evidence of plasmid loss. Removing the assumption that there is no plasmid loss should further improve our results.

Our approach can easily be refined to include different features and adapted to different experiments. We will test additional distributions and incorporate cell features into the contact quality functions. Moreover, we hope to model other experiments with similar setups. It would be interesting to investigate what factors differ across species of bacteria and types of plasmids.

A secondary goal of this project is identifying the most likely donor corresponding to each observed transconjugant. To do so, we add nodes representing if transmission occurs between two cells. A marginal MAP query over those nodes would require one transmission node to have value 1, thereby returning the most likely donor. It will be interesting to see if different models predict the same donors. We could also manually observe those pairs in the imaging data. If the pairs share features we did not account for, we will add them to our distributions and evaluate the new model. Iterating this process could lead to further insights about the mechanisms governing conjugation.

References

- [1] L. Andrup and K. Andersen. A comparison of the kinetics of plasmid transfer in the conjugation systems encoded by the f plasmid from *Escherichia coli* and plasmid pcf10 from *Enterococcus faecalis*. *Microbiology*, 145(8):2001–2009, 1999.
- [2] L. Andrup, L. Smidt, K. Andersen, and L. Boe. Kinetics of conjugative transfer: A study of the plasmid pXo16 from *Bacillus thuringiensis* subsp. *israelensis*. *Plasmid*, 40(1):30–43, 1998.
- [3] A. Babić, A.B. Lindner, M. Vulić, E.J. Stewart, and M. Radman. Direct visualization of horizontal gene transfer. *Science*, 319(5869):1533–1536, 2008.
- [4] S. Berg, D. Kutra, T. Kroeger, C.N. Straehle, B.X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J.I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F.A. Hamprecht, and A. Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, 2019.
- [5] A.C. Carroll and A. Wong. Plasmid persistence: costs, benefits, and the plasmid paradox. *Canadian Journal of Microbiology*, 64(5):293–304, 2018.
- [6] A. Couturier, C. Virolle, K. Goldlust, A. Berne-Dedieu, A. Reuter, S. Nolivos, S. Bigot, Y. Yamaichi, and C. Lesterlin. Real-time visualisation of the intracellular dynamics of conjugative plasmid transfer. *Nature Communications*, 14(1):294, 2023.
- [7] J.N. Cullum, J.F. Collins, and P. Broda. Factors affecting the kinetics of progeny formation with flac in *Escherichia coli* K12. *Plasmid*, 1(4):536–544, 1978.
- [8] K.J. Cutler, C. Stringer, T.W. Lo, L. Rappez, N. Stroustrup, S.B. Peterson, P.A. Wiggins, and J.D. Mougous. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature Methods*, 19(11):1438–1448, 2022.

- [9] J. Davison. Genetic exchange between bacteria in the environment. Plasmid, 42(2):73–91, 1999.
- [10] F. Dionisio, I.C. Conceição, A.C. Marques, L. Fernandes, and I. Gordo. The evolution of a conjugative plasmid and its ability to increase bacterial fitness. Biology letters, 1:250–252, 2005.
- [11] G. Frankel, S. David, W.W. Low, C. Seddon, J.L.C. Wong, and K. Beis. Plasmids pick a bacterial partner before committing to conjugation. Nucleic acids research, 51(17):8925–8933, 2023.
- [12] K. Goldlust, A. Ducret, M. Halte, A. Dedieu-Berne, M. Erhardt, and C. Lesterlin. The f pilus serves as a conduit for the dna during conjugation between physically distant bacteria. Proceedings of the National Academy of Sciences, 120(47):e2310842120, 2023.
- [13] A. Goñi-Moreno and M. Amos. Discus: A simulation platform for conjugation computing. In Unconventional Computation and Natural Computation, pages 181–191. Springer International Publishing, 2015.
- [14] R. Gregory, J.R. Saunders, and V.A. Saunders. Rule-based modelling of conjugative plasmid transfer and incompatibility. Biosystems, 91(1):201–215, 2008.
- [15] L.C. Harrington and A.C. Rogerson. The f pilus of escherichia coli appears to support stable dna transfer in the absence of wall-to-wall contact between cells. Journal of bacteriology, 172:7263–7264, 1990.
- [16] J.A. Heinemann and G.F. Sprague. Bacterial conjugative plasmids mobilize dna transfer between bacteria and yeast. Nature, 340(6230):205–209, 1989.
- [17] J. Jass, S. Schedin, E. Fällman, J. Ohlsson, U.J. Nilsson, B.E. Uhlin, and O. Axner. Physical properties of escherichia coli p pili measured by optical tweezers. Biophysical Journal, 87(6):4271–4283, 2004.
- [18] A.R. Johnsen and N. Kroer. Effects of stress and other environmental factors on horizontal plasmid transfer assessed by direct quantification of discrete transfer events. FEMS Microbiology Ecology, 59(3):718–728, 2007.
- [19] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge, Massachusetts, 2009.

- [20] C. Lagido, I.J. Wilson, L.A. Glover, and J.I. Prosser. A model for bacterial conjugal gene transfer on solid surfaces. FEMS Microbiology Ecology, 44(1):67–78, 2003.
- [21] T.D. Lawley, G.S. Gordon, A. Wright, and D.E. Taylor. Bacterial conjugative transfer: visualization of successful mating pairs and plasmid establishment in live escherichia coli. Molecular Microbiology, 44(4):947–956, 2002.
- [22] B. Li, Y. Qiu, Y. Song, H. Lin, and H. Yin. Dissecting horizontal and vertical gene transfer of antibiotic resistance plasmid in bacterial community using microfluidics. Environment International, 131:105007, 2019.
- [23] L Liu, Q. Zhang, and R. Li. In situ and individual-based analysis of the influence of polystyrene microplastics on escherichia coli conjugative gene transfer at the single-cell level. Environmental Science & Technology, 57(42):15936–15944, 2023.
- [24] Q. Luan, C. Macaraniag, J. Zhou, and I. Papautsky. Microfluidic systems for hydrodynamic trapping of cells and clusters. Biomicrofluidics, 14(3):031502, 2020.
- [25] J.S. Madsen, M. Burmølle, L.H. Hansen, and S.J. Sørensen. The interconnection between biofilm formation and horizontal gene transfer. FEMS Immunology Medical Microbiology, 65(2):183–195, 2012.
- [26] A. Massoudieh, A. Mathew, E. Lambertini, K.E. Nelson, and T.R. Ginn. Horizontal gene transfer on surfaces in natural porous media: Conjugation and kinetics. Vadose Zone Journal, 6(2):306–315, 2007.
- [27] B.V. Merkey, L.A. Lardon, J.M. Seoane, J. Kreft, and B.F. Smets. Growth dependence of conjugation explains limited plasmid invasion in biofilms: an individual-based modelling study. Environmental Microbiology, 13(9):2435–2452, 2011.
- [28] A. Norman, L.H. Hansen, and S.J. Sørensen. Conjugative plasmids: vessels of the communal gene pool. Philosophical transactions of the Royal Society of London, Series B, Biological sciences, 364(1527):2275–2289, 2009.
- [29] F. Padilla-Vaca, F. Anaya-Velázquez, and B. Franco. Synthetic biology: Novel approaches for microbiology. International microbiology : the official journal of the Spanish Society for Microbiology, 18(2):71–84, 2015.
- [30] K.R. Philipsen, L.E. Christiansen, H. Hasman, and H. Madsen. Modelling conjugation with stochastic differential equations. Journal of Theoretical Biology, 263(1):134–142, 2010.

- [31] J.M. Ponciano, L. De Gelder, E.M. Top, and P. Joyce. The Population Biology of Bacterial Plasmids: A Hidden Markov Model Approach. Genetics, 176(2):957–968, 2007.
- [32] T.J. Rudge, P.J. Steiner, A. Phillips, and J. Haseloff. Computational modeling of synthetic microbial biofilms. ACS Synthetic Biology, 1(8):345–352, 2012.
- [33] A.L. Samuels, E. Lanka, and J.E. Davies. Conjugative junctions in rp4-mediated mating of *Escherichia coli*. Journal of Bacteriology, 182(10):2709–2715, 2000.
- [34] J. Seoane, T. Yankelevich, A. Dechesne, B. Merkey, C. Sternberg, and B.F. Smets. An individual-based approach to explain plasmid invasion in bacterial populations. FEMS microbiology ecology, 75(1):17–27, 2011.
- [35] S. Soucy, J. Huang, and J. Gogarten. Horizontal gene transfer: building the web of life. Nature Reviews Genetics, 16(8):472–482, 2015.
- [36] D.R. Stirling, M.J. Swain-Bowden, A.M. Lucas, A.E. Carpenter, B.A. Cimini, and A. Goodman. Cellprofiler 4: improvements in speed, utility and usability. BMC Bioinformatics, 22(1):433, 2021.
- [37] D.K. Summers. The kinetics of plasmid loss. Trends in Biotechnology, 9(1):273–278, 1991.
- [38] S.J. Sørensen, M. Bailey, L.H. Hansen, N. Kroer, and S. Wuertz. Studying plasmid horizontal transfer in situ: a critical review. Nature Reviews Microbiology, 3(9):700–710, 2005.
- [39] C. Virolle, K. Goldlust, S. Djermoun, S. Bigot, and C. Lesterlin. Plasmid transfer by conjugation in gram-negative bacteria: From the cellular to the community level. Genes, 11(11):1239, 2020.
- [40] H. Yin and D. Marshall. Microfluidics for single cell analysis. Current Opinion in Biotechnology, 23(1):110–119, 2012.
- [41] A. Yip, J. Smith-Roberge, S. Haghayegh Khorasani, M.G. Aucoin, and B.P. Ingalls. Calibrating spatiotemporal models of microbial communities to microscopy data: A review. PLOS Computational Biology, 18(10):e1010533, 2022.