

UC Merced

UC Merced Electronic Theses and Dissertations

Title

Multiscale Modeling and Deep Learning for Complex Microbial Colonies

Permalink

<https://escholarship.org/uc/item/4381215s>

Author

Collignon, Jordan

Publication Date

2024

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, MERCED

MULTISCALE MODELING AND DEEP LEARNING FOR
COMPLEX MICROBIAL COLONIES

*A dissertation submitted in partial satisfaction of the requirements for
the degree of Doctor of Philosophy*

in

APPLIED MATHEMATICS

by

JORDAN COLLIGNON

Committee in charge:

Professor Suzanne Sindi, Chair

Professor Erica Rutter

Professor Emily Jane McTavish

2024

Copyright

Chapter 2 © 2020 Multidisciplinary Digital Publishing Institute

Chapter 3 © 2024 Springer-Verlag

This is to certify that I have examined a copy of the dissertation by

Jordan Collignon

and found it satisfactory in all respects, and that any and all revisions
required by the dissertation committee have been made.

Applied Mathematics
Dissertation Committee Chair:

Professor Suzanne Sindi

Dissertation Committee:

Professor Erica Rutter

Dissertation Committee:

Professor Emily Jane McTavish

Date

Contents

Signature Page	iii
List of Symbols	vii
List of Figures	x
List of Tables	xii
Acknowledgements	xiii
Curriculum Vitae	xiv
Abstract	xx
1 Introduction and Background	1
1.1 Overview	1
1.2 Richness of Microbial Colonies	2
1.3 Irreversible Phenotypes: Prion Proteins in the Yeast <i>Saccharomyces cerevisiae</i>	2
1.4 Reversible Phenotypes: White-Opaque Switch in the Yeast <i>Candida albicans</i>	4
1.5 Computational Approaches to Studying Yeast Colonies	6
1.5.1 Colony Growth Simulations	6
1.5.2 Image Analysis and Deep Learning Applications	7
1.6 Dissertation Structure	9
2 Quantifying the Biophysical Impact of Budding Cell Division on the Spatial Organization of Growing Yeast Colonies	11
2.1 Introduction	11
2.2 Materials and Methods	15
2.2.1 Computational Model	15
2.2.2 Colony Metrics	22
2.2.3 Statistical Analysis	28
2.3 Results	29
2.3.1 Nutrient-Rich Growth: Budding Division Impacts Local Colony Organization in Simulated Yeast Colonies	29
2.3.2 Nutrient-Limited Growth: Differential Growth Rates Impact Global Organization of Yeast Colonies	36
2.4 Discussion	45
2.5 Conclusions	50

3	<i>[PSI]-CIC: A Deep-Learning Pipeline for the Annotation of Sec-tored <i>Saccharomyces cerevisiae</i> Colonies</i>	51
3.1	Introduction	51
3.1.1	Yeast as a Model System	52
3.1.2	The Role of Image Analysis	54
3.2	Methods	55
3.2.1	<i>[PSI]-CIC</i> Algorithm	55
3.2.2	Training (Image Segmentation)	62
3.2.3	Evaluation	63
3.3	Results	63
3.3.1	Training Images	63
3.3.2	Testing Images	64
3.4	Discussion	69
3.5	Conclusion	72
4	A Deep Learning Framework for <i>Candida albicans</i> Colony Classifi-cation	74
4.1	Introduction	74
4.2	Methods	75
4.2.1	<i>Candida</i> Classification Pipeline	75
4.2.2	Colony Detection	76
4.2.3	Neural Network Architectures	79
4.2.4	Label Prediction	81
4.2.5	Ground Truth Annotation	82
4.2.6	Training	83
4.3	Results	84
4.3.1	Single Input Architectures: Quantitative Performance	84
4.3.2	Single Input Architectures: Qualitative Performance	88
4.3.3	Dual Input Architectures: Quantitative Performance	88
4.3.4	Dual Input Architectures: Qualitative Performance	93
4.3.5	Robustness of Model Performance	93
4.4	Discussion	96
4.5	Conclusion and Future Work	98
5	Counting Microbial Colonies	100
5.1	Introduction	100
5.2	Analyzing Performance of Circular Object Detection for Colony Images	101
5.2.1	More about the CHT Method	102
5.2.2	CHT Implementation: <i>[PSI⁺]</i> Colony Images	103
5.2.3	Measuring Accuracy of CHT	105
5.2.4	U-Net + CHT: Addressing a Detection Problem in Bacteria Colony Images	110
5.3	Generalizing <i>Candida albicans</i> Colony Classification using Additional Datasets	114

5.3.1	Performance on the Extended CHROMagar Dataset	114
5.3.2	Performance on the SDA Dataset	116
5.3.3	Performance on the Combined CHROMagar + SDA Datasets	117
5.3.4	Discussion on Extended Data	119
6	Conclusions and Future Work	126
6.1	Summary	126
6.2	Future Directions	127
6.2.1	Embedding Prion Aggregation into the Center-based Model .	127
6.2.2	Deep-Learning for Sectorized Colony Image Classification	130
6.2.3	Developing a Center-Based Model of <i>Candida albicans</i> Colony Growth	132
6.2.4	Final Thoughts	133
A	Data Curation and [PSI]-CIC Implementation	134
A.1	Image Acquisition and Pre-processing	134
A.2	Synthetic Image Generation	135
A.3	Colony Extraction	137
A.4	Implementation	139
B	Image Acquisition for Candida Pipeline	141
B.1	Experimental Images	141
B.2	Data Augmentation	141
B.3	Implementation	142
C	Image Acquisition for Other Colony Images	143
C.1	Additional Experimental Data	143
C.2	Synthetic Bacterial Colony Image Generation	143
C.2.1	Colony Location and Size	143
C.2.2	Mask Generation	146
C.2.3	Color Selection	146
C.2.4	Image Generation	147
C.3	Performance Metrics of <i>C. albicans</i> Classification Pipeline on Addi- tional Experimental Data	148
C.3.1	Extended CHROMagar Dataset	148
C.3.2	SDA Dataset	151
C.3.3	Combined CHROMagar + SDA Dataset	159

List of Symbols

Chapter 2

R_i	radius of cell i
\vec{x}_i	position of the center of cell i .
$d_{i,j}$	the Euclidean distance between two cells i and j .
$E_{i,j}^{CC}$	modified Hertz potential between two neighboring cells i and j .
E^{MB}	Linear spring constant describing the attachment of a bud on a mother cell.
$E_{i,j}^{adh}$	the adhesive interaction between two neighboring cells i and j .
σ	Poisson ratio describing the incompressibility of yeast cells.
E	Young's Modulus describing the mechanical property of yeast cell walls.
ϕ	receptor surface density describing the density of surface adhesion molecules in the contact area.
W_s	single bond binding energy.
η	damping coefficient describing the viscosity of the growth media.
$G2_{\text{avg}}$	average length in time for the G2 phase of cells.
$G1_{\text{avg}_{\text{daughter}}}$	average length in time of the G1 phase for new daughter cells.
$G1_{\text{avg}_{\text{mother}}}$	average length in time of the G1 phase for mother cells.
R_{avg}	average mature radius size of a cell.
$M_{j,\text{max}}$	carrying capacity indicating the maximal possible biomass for each subdomain in the biophysical model.
$D_i(t)$	the area of subdomain i with respect to time t .

r	the rate of maximum cell cycle adjustment, controlling the amount the cell cycle is adjusted at each timestep.
Δt	the length of one timestep.
p	p-value for hypothesis testing.
α	significance level for hypothesis testing.

Chapter 3

$[R, G, B]$	vector of intensities for the red, green, and blue channels respectively.
R_i	the i^{th} red region of a colony.
W_i	the i^{th} white region of a colony.
a	the total number of red regions for a given colony.
b	the total number of white regions for a given colony.
$N(R_i, red), N(R_i, white)$	the number of red or white pixels respectively inside the i^{th} red region of a colony.
$N(W_i, red), N(W_i, white)$	the number of red or white pixels respectively inside the i^{th} white region of a colony.
$p(R_i, red), p(R_i, white)$	function defining the purity of region R_i in a given colony with respect to the red or white pixels respectively in that region.
$p(W_i, red), p(W_i, white)$	function defining the purity of region W_i in a given colony with respect to the red or white pixels respectively in that region.
$\mu(R_i), \mu(W_j)$	weight function denoting the proportion of pixels in the colony occupied by region R_i and region W_j respectively.
p_w	function denoting the weighted purity of a colony with respect to both red and white regions.

Chapter 4

$r_{i,j}$	the radius of colony i found in plate j in pixels.
R_j	the radius of plate j in pixels.
$c_{i,j}$	the fraction between the radius of colony i in plate j and the radius of plate j .
$d_{i,j}$	the value of $c_{i,j}$ normalized to the interval $[0, 1]$ across all values of $c_{i,j}$.
α	the weight-balancing factor in the categorical focal cross-entropy loss function.

γ the focusing parameter in the categorical focal cross-entropy loss function.

Chapter 5

\vec{c}_i the vector representing the center of synthetic colony i .

r_i the radius of synthetic colony i .

δ the minimum distance tolerance threshold between any two neighboring colonies.

\vec{x} a point in a distribution.

$\vec{\mu}$ the mean of a distribution.

Σ the covariance of a distribution.

d_M the Mahalanobis distance between and point and a distribution.

List of Figures

1.1	Phenotypic Variants in <i>S. cerevisiae</i> and <i>C. albicans</i>	3
1.2	Yeast prion phenotypes are the result of multiscale processes.	5
2.1	Spatial Phenotypes are the Consequence of Processes at Different Scales	13
2.2	Biophysical Model of Cell-Cell Interactions	16
2.3	Selecting a Bud Site	19
2.4	Cell Cycle Length	21
2.5	Colony Sparsity and Expanse	25
2.6	Colony Spatial Graphs	27
2.7	Constructing Subcolony Graphs	28
2.8	Simulated Yeast Colonies in Nutrient-Rich Conditions and their Corresponding Lineage Relationships	30
2.9	Population Growth, Expanse and Sparsity of Simulated Yeast Colonies in Nutrient-Rich Conditions	31
2.10	Age and Spatial Organization of Cells in Nutrient-Rich Colonies	33
2.11	Colony Connectivity in Nutrient-Rich Colonies	35
2.12	Subcolony Structure and Organization in Nutrient-Rich Colonies	37
2.13	Population Growth, Expanse and Sparsity of Simulated Yeast Colonies in Nutrient-Limited Conditions	38
2.14	Age and Spatial Organization of Cells in Nutrient-Limited Colonies	40
2.15	Colony Connectivity in Nutrient-Limited Colonies	42
2.16	Subcolony Structure and Organization in Nutrient-Limited Colonies	44
2.17	Nutrient Limitation Drives Spatial Organization of Cells	46
3.1	Yeast prion phenotypes are the result of multiscale processes.	53
3.2	Illustration of the proposed pipeline.	56
3.3	Novel annotation and sector counting procedure.	58
3.4	Plate Level Segmentations	64
3.5	Accuracy of colony-level predictions on quantifiable colony data	67
3.6	Purity correction improves classification.	70
4.1	Candida Detection Pipeline	76
4.2	Distributions of Colony sizes	78
4.3	Neural Network Architectures	80
4.4	Preprocessing images for training	82

4.5	Example colony predictions	85
4.6	Confusion Matrices for the Single Input Toy Model	86
4.7	Confusion Matrices for the Single Input Resnet 34 Model	87
4.8	Precision-Recall and ROC curves for the Single input toy model . . .	89
4.9	Precision-Recall and ROC curves for the Single Input Resnet 34 model	90
4.10	Confusion Matrices for the Dual Input Toy Model	91
4.11	Confusion Matrices for the Single Input Resnet 34 Model	92
4.12	Precision-Recall and ROC curves for the Dual input toy model	94
4.13	Precision-Recall and ROC curves for the Dual input Resnet 34 model	95
4.14	Distribution of prediction scores for the single input toy model without data augmentation	97
5.1	Visualizing steps in the circle Hough transform	104
5.2	Colony Counting Flowchart	105
5.3	Colony counting accuracy	107
5.4	Accuracy analysis pipeline	108
5.5	Pipeline for extracting annotations from the images	109
5.6	Analyzing false positive and false negative colony detections	111
5.7	Finding the optimal sensitivity parameter	112
5.8	Using <code>imfindcircles</code> may be ineffective without preprocessing . . .	113
6.1	Representing mechanisms of prion aggregation at multiple scales . . .	129
6.2	Expected output of using deep-learning methods for colony analysis .	131

List of Tables

2.1	Parameter Values Used in ABM	23
3.1	Isolating Quantifiable Colonies	65
3.2	Classification performance	66
4.1	Counts of extracted images from each plate	83
4.2	Prediction accuracy of each model over multiple runs	96
5.1	Error analysis for colony detection	108
5.2	Area under the curve (AUC) metrics for model performance on each class of images in the CHROMagar dataset.	115
5.3	Accuracy analysis of the performance of the four deep learning models on the testing images in the extended CHROMagar dataset	116
5.4	Area under the curve (AUC) metrics for model performance on each class of images in the SDA dataset.	117
5.5	Accuracy analysis of the performance of the four deep learning models on the testing images in the extended SDA dataset	118
5.6	Area under the curve (AUC) metrics for model performance on each class of images in the combined CHROMagar + SDA dataset.	119
5.7	Accuracy analysis of the performance of the four deep learning models on the testing images in the combined CHROMagar + SDA dataset	120
5.8	Accuracy analysis of single input toy model performance on each testing set	121
5.9	Accuracy analysis of dual input toy model performance on each testing set	122
5.10	Accuracy analysis of single input Resnet 34 performance on each testing set	123
5.11	Accuracy analysis of dual input Resnet 34 performance on each testing set	124

Acknowledgements

I wish to express thanks to Dr. Tricia Serio at the University of Washington and Dr Wesley Naeimi at the University of Massachusetts, Amherst for providing the experimental data necessary for the work detailed in Chapter 3. I also wish to express thanks to Dr. Clarissa Nobile, Dr. Aaron Hernday, Dr. Ruihao Li, Austin Perry, Namkha Nguyen, and Daravuth Cheam at the University of California, Merced for providing the experimental data necessary for the work detailed in Chapters 4 and 5. I also want to thank the National Science Foundation and the National Institutes of Health for the grant funding that supported this work. I also want to thank the National Science Foundation at UC Merced for the National Research Traineeship in Intelligent Adaptive Systems fellowship, the UC Merced Fletcher Jones fellowship, the UC Merced Fred & Mitzie Ruiz Fellowship, and the UC Merced Chancellor's Graduate Fellowship for providing the funding to support the completion of the work described in this dissertation.

The text of Chapter 2 of this dissertation is a reprint of the material as it appears in the journal *Dynamics Models in Biology and Medicine, Volume II*. The co-author listed in this publication made substantial contributions to the development of the work presented in this chapter of this dissertation.

JORDAN COLLIGNON

Email: jcollignon@ucmerced.edu
Website: <https://jcollignon.wordpress.com>
LinkedIn: <https://linkedin.com/in/jordan-collignon-936698104>
Portfolio: <https://portfolio.com/jcollignon>

EDUCATION

University of California, Merced

Doctor of Philosophy, Applied Mathematics Expected Spring 2024

Master of Science, Applied Mathematics Fall 2022

Field of Study: Mathematical Biology

Dissertation Topic: Multiscale Modeling and Deep Learning for Complex Microbial Colonies

Advisor: Dr. Suzanne Sindi

Supervisors: Dr. Suzanne Sindi, Dr. Erica Rutter, Dr. Emily Jane McTavish

Address: 5200 North Lake Road, Merced, CA 95343

California State University, Monterey Bay

Bachelor of Science, Mathematics, Pure Concentration. Spring 2017

Address: 100 Campus Ctr, Seaside, CA

Major GPA: 4.0

TECHNICAL SKILLS

- MATLAB
- Python
- LaTeX
- R
- C++
- Netlogo
- Inkscape
- Adobe Photoshop
- Microsoft Office

PROFESSIONAL EXPERIENCE

Teaching Fellow + Course Designer Fall 2023 – Present

University of California, Merced

- Under the supervision of Dr. Arnold Kim, delivered the SPARK seminar “SPRK 010: Opportunities in Data Science” to incoming first-year undergraduate students.

- Under the supervision of Dr. Suzanne Sindi, delivering the SPARK seminar “SPRK 010: Thinking with Data” as a course designer and occasional lecturer.

Instructor

Summer 2021

Hybrid; University of Northern Colorado, Greeley, CO

- Developed and taught a two-week summer course as part of the Frontiers of Science Institute curriculum.
- Introduced Python coding with applications to image processing for advanced middle and high school students using Google Colaboratory.

Graduate Student Researcher and Fellow

Fall 2018 – Present

University of California, Merced, Merced, CA

- Developing a deep-learning pipeline to analyze images of plated sectored yeast colonies to obtain insight into the mechanisms behind prion disease.
- Developing a biophysical model of a growing budding yeast colony that can replicate sectoring behavior observed in real yeast colonies.
- Submitted a paper quantifying the biophysical impact that budding has on yeast colonies and how this contributes to organized colony structures.

Teaching Assistant

Fall 2017 – Summer 2018

University of California, Merced, Merced, CA

- Led discussion sections of MATH 032: Probability & Statistics under the supervision of Haik Stepanian, with greater emphasis on theory and application.
- Led discussion sections of MATH 032: Probability & Statistics under the supervision of Dr. Suzanne Sindi with greater emphasis on R programming and sampling methods.

Summer@ICERM

Summer 2016

Brown University, Providence, RI

- Applied methods of mathematical modeling methods to predict bodily lead propagation.
- Developed and analyzed a system of coupled partial differential equations corresponding to blood and two types of bone as the body’s lead storages.

Valparaiso Experience in Research for Undergraduate Mathematicians

Summer 2015

Valparaiso University, Valparaiso, IN

- Used statistical analysis to estimate different ecological factors contributing to the passenger pigeon’s survival such as its historic range, food requirements, and average lifespan.

- Developed an agent-based model showing how these elements interact in a simulated forest environment where the passenger pigeon is reintroduced.

PRESENTATIONS

- Multiscale Modeling and Deep Learning for Complex Microbial Colonies, Dissertation Defense Presentation, University of California, Merced, Merced, CA, April 19, 2024.
- *[PSI]-CIC*: A Deep-learning Pipeline for the Analysis and Annotation of Sectored Yeast Colonies, Society for Mathematical Biology, Columbus, OH, July 18, 2023.
- A Machine Learning Pipeline for the Analysis of Sectored Yeast Colonies, Society for Mathematical Biology Annual Conference, Virtual Conference, June 15, 2021.
- Modeling and Analysis of Sector-like Formations in Yeast Colonies, Society for Industrial and Applied Mathematics (SIAM) Dynamical Systems, Virtual Conference, May 25, 2021.
- Developing an Image Processing Technique to Quantify Prion Sectoring in Yeast, Mathematical Biology SMART Team, University of California, Merced, Merced, CA, October 14, 2020.
- Modeling Yeast Colony Structure to Study Sectored Prion Phenotypes, Society for Mathematical Biology, Université de Montreal, Montreal, Quebec, July 23, 2019.
- Propagation of Lead in the Human Body, MAA Golden Section Meeting, Santa Clara University, Santa Clara, CA., March 4, 2017.
- Mathematical Modeling in Ecology: Simulating the Reintroduction of the Extinct Passenger Pigeon, 30th Annual California State University Student Research Competition, CSU Bakersfield, Bakersfield, CA., April 29, 2016.
- An Agent-based Model Simulating the Reintroduction of the Extinct Passenger Pigeon under Varying Resource Availability, Joint Mathematics Meetings, Washington State Convention Center, Seattle, WA., January 8, 2016.

PUBLICATIONS

- **J. Collignon**, W. Naeimi, T. R. Serio, and S. Sindi. *[PSI]-CIC*: A Deep-Learning Pipeline for the Annotation of Sectored *Saccharomyces cerevisiae* Colonies. *Bulletin of Mathematical Biology*. (Under review, February 2024)
- **J. Collignon**. A High-throughput Pipeline for the Analysis and Annotation of Sectored Yeast Colonies. Capstone Report, eScholarship. (January 2023)
- D. Elzinga, E. Boggess, **J. Collignon**, A. Riederer, and A. Capaldi. An Agent-based Model Determining a Successful Reintroduction of the Extinct Passenger Pigeon (*Ectopistes migratorius*). *Natural Resource Modeling*. (October 2020)

- M. Banwarth-Kuhn, **J. Collignon**, S. Sindi. Quantifying the Biophysical Impact of Budding Cell Division on the Spatial Organization of Growing Yeast Colonies. Applied Sciences: Dynamic of Biology and Medicine Volume II. (September 2020)
- M. Morrissey, **J. Collignon**, V. Ciocanel, and T Kapitula. Propagation of Lead in the Human Body. Society for Industrial and Applied Mathematics Online. (July 2017)

PROFESSIONAL MEMBERSHIPS

- American Mathematical Society (AMS)
- Society for Industrial and Applied Mathematics (SIAM)
- Society for Mathematical Biology (SMB)

CERTIFICATES

- Advanced Pedagogy, University of California, Merced Teaching Commons (Spring 2023)
- Principles of Pedagogy, University of California, Merced Center for Engaged Teaching and Learning (Fall 2021)
- Bobcat Leadership Seminar, Leader Development Program at UC Merced, University of California, Merced Margo F. Souza Student Leadership Center (Fall 2020)

HONORS AND AWARDS

UC Merced Fletcher Jones Fellowship 2022 – 2023

This fellowship is awarded to select Ph.D. candidates at UC Merced each year to facilitate further research and dissertation progress.

Outstanding Senior Math Major Award 2017

Each year, the mathematics faculty nominate one student from each of the four grade levels based on their GPA, attitude towards mathematics, and departmental contributions.

Barry Goldwater Scholarship and Excellence in Education Program 2016

This scholarship is given to college students who plan to pursue a research career in their chosen field.

National Alliance Scholar for Doctoral Studies in the Mathematical Sciences 2016

This program nominates students aiming to pursue a Ph.D. in the mathematical sciences.

Posters on the Hill Honorable Mention 2016

A competitive program giving students the opportunity to present their research to Congress members.

Undergraduate Research Opportunities Center (UROC) Scholar's Program 2015 – 2017

This two-year program helps students who want to pursue a Ph.D. in their field develop their applications for graduate school and fellowships.

Pay it Forward Scholarship and Mentoring Program 2013 – 2017

A four-year scholarship given to first-generation college students from Monterey County schools.

LEADERSHIP AND SERVICE

Applied Mathematics Graduate Student Leadership Committee 2023 – Present

- Served as a senior member of the inaugural committee.
- Assisted in the development and implementation of the annual UC Merced Graduate Visitation Weekend events.
- Co-authored a simpler guide for travel reimbursement proceedings that Applied Mathematics graduate students can refer to for assistance with submitting reimbursement materials for conference related travel.
- Co-authored committee guidelines and standard operating procedures to guide future iterations of this committee.

Graduate Student Association at the University of California, Merced 2019 – Present

- Advocated for proper representation of graduate students in university decision making.
- Maintained the Association's website for graduate students to stay informed.
- Worked as the recording secretary for all official council meetings.
- Served as the graduate group Delegate for two years.
- Implemented new bylaws and election structure during the 23-24 academic year.

Data Analyst Volunteer, Legal Services for Seniors

Spring 2016

- Served as part of the upper division service learning Mathematics Consultants course at California State University, Monterey Bay
- Organized client information and created visual representations of data.
- Provided bigger pictures of who this organization serves and areas to expand service.

Pay it Forward Mentor, The First Tee of Monterey County

2013 – 2017

- Served as a mentor and tutor to a student in Alisal Union School District.
- Participated in monthly meetings with colleagues aimed at changing the college experience.

Abstract

Title: Multiscale Modeling and Deep Learning for Complex Microbial Colonies

Name: Jordan Collignon

Degree: Applied Mathematics Ph.D.

University: University of California, Merced

Year: 2024

Committee Chair: Professor Suzanne Sindi

It is challenging to build meaningful models of biological systems that are calibrated to experimental data. For microbial colonies, chromogenic assays are key tools for distinguishing phenotypic variants in growing colonies over time and provide additional useful information for colony quantification. In microbial colonies, the presence of multiple phenotypes indicate a functional change that is dependent on the colony species. One example in *Saccharomyces cerevisiae* is that the presence of multiple phenotypes indicates a loss of infectious agents within the cells. Another example in *Candida albicans* is that the presence of multiple phenotypes tells us about cell maturing strategies. At present, we lack both a model that explains the formation of these sectorized phenotypes and a method for validating such a model from experiments. In addition, we lack a framework which couple both approaches to make meaningful insights related to the driving of multiple phenotypes in microbial colonies. In this dissertation, I seek to address this gap with a data-driven approach that integrates experimental data of sectorized yeast colonies with the construction of an agent-based model of growing budding yeast colonies. I first discuss my previously published work developing an agent based model of budding yeast and apply mathematical techniques to analyze colony structure. Next, I explain my ongoing work to use image analysis techniques to create a high-throughput pipeline allowing us to extract information about the size and structure of colonies found in image data. Finally, I present ongoing work with using this framework for larger and more diverse datasets and discuss limitations on their use in automated colony quantification. Together these projects create a framework allowing us to adapt our agent based modeling framework to the conditions observed in experiments so that the model both captures sectoring behavior and allows us to predict the mechanisms driving sectoring behavior.

Chapter 1

Introduction and Background

1.1 Overview

In this dissertation we primarily focus on the study and analysis of biological yeast colonies such as *Saccharomyces cerevisiae* and *Candida albicans* which exhibit heterogeneous structures due to inherent features present within the cells that comprise colonies. This dissertation covers three central objectives towards the simulation and analysis of complex microbial colonies.

- First, we develop a framework for the study of microbial colonies where simulation-based models and experimental data can be used to make inferences about the underlying dynamics driving heterogeneous colony phenotypes.
- Second, we develop novel data augmentation methods and couple them with deep-learning frameworks for the annotation and classification of single and multi-phenotype yeast colonies: Chapter 3: $[PSI^+]/[psi^-]$ colonies in *Saccharomyces cerevisiae* and Chapter 4: white/opaque colonies in *Candida albicans*.
- Third, we develop computational pipelines to count the number of circular colonies and apply them to accelerate the discovery process over traditional approaches of counting by hand.

Throughout, yeast is used as a model system. In the first objective, we consider the forward problem where colony growth is simulated. The second and third objectives consider the analysis of colony level experimental data. While the spread of prion proteins serves as a major motivator for this work, particularly Chapters 2 and 3, the methods developed are general and applicable towards the study of the emergence of microbial level phenotypes more broadly. Their generalizability will be tested and discussed in Chapter 5.

1.2 Richness of Microbial Colonies

In the context of cell biology, a cell is the smallest “live” unit of any living organism. The use of single celled organisms to study biology is a common approach to studying big questions in the field ranging from the onset of disease to genomic sequencing. Through continuous research, tools are developed to aid in efficiently providing the information necessary to answer new and more complex questions [178].

An alternative to single-cell approaches for answering complex questions is through experiments on biological colonies. A colony consists of populations of cells such that all cells are descendants of a single parent or “founder” cell. In both scales, researchers can examine physical characteristics of interest of cells and colonies which indicate their phenotype. In general, we can define a phenotype as a set of observable characteristics that can include its interaction with the local environment [176]. A couple features indicative of a cell’s phenotype are its shape or morphology, such as whether a cell is spherical or ellipsoidal, and whether a cell is currently undergoing division (see Figure 1.1). Similarly, a colony of cells has its own phenotype, such as its shape with respect to its own environment and the presense of structures within a subset of the colony which exhibit distinct features from other parts of the colony. At the colony scale, different morphologies within a colony of the same species are signs of phenotypic variants. A phenotypic variant as defined by Frobose et al [56] are colonies of the same species which exhibit different morphologies [138]. Specific examples include pseudohyphael growth at the colony periphery [14] or different pigmentations in subsets of colonies in controlled experiments [83]. Chromogenic assays also allow experimentalists to view phenotypic variants at the colony level by tracking changes in color throughout colony growth. Studying phenotype variants at the colony level is capable of providing additional insight into a particular experiment.

Through the analysis of phenotypic variants at a colony level, we are interested in studying the heterogeneity of a colony and treatments that give rise to heterogeneity. For this dissertation, we primarily detail methods that both analyze and leverage colony heterogeneity in *Saccharomyces cerevisiae* and *Candida albicans*. These are two species of yeast to which experimental manipulations have revealed unique morphologies which we attempt to model and quantify throughout this dissertation. We aim to leverage this to answer complex biological questions related to changes in colony morphology and behavior during proliferation over time. We break this down further based on whether a phenotype is reversible or irreversible.

1.3 Irreversible Phenotypes: Prion Proteins in the Yeast *Saccharomyces cerevisiae*

Protein folding is a vital process for regulating development [180] of a living organism. Failure of the protein folding process is associated with neurodegenerative diseases [180]. One class of fatal neurodegenerative diseases arises due to the presence

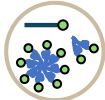
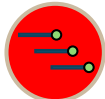
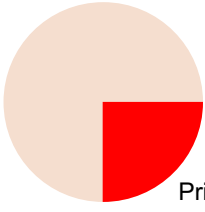
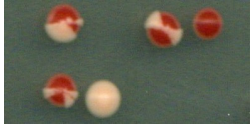

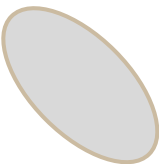


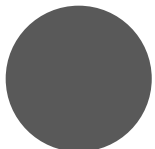
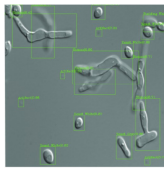
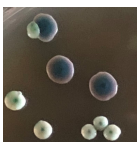
	Cell	Colony	Experiments
<p><i>S. cerevisiae</i> [PSI⁺] /[psi⁻]</p>	<p>Prion - state </p> <p>Prion - free </p>	<p>Prion - state region </p> <p>Prion - free region</p>	<p>Colonies </p>
<p><i>C. albicans</i> White- Opaque</p>	<p>Round </p> <p>Elongated </p>	<p>White (small) </p> <p>Sectored </p> <p>Opaque (large) </p>	<p>Cells </p> <p>Colonies </p>

Figure 1.1: **Phenotypic Variants in *S. cerevisiae* and *C. albicans*.** Examples of phenotypes in *S. cerevisiae* (top) and *C. albicans* (bottom). Cell-level phenotypes (left) include color and local environmental interactions as a collection of physical characteristics. Within *S. cerevisiae* cells the presence of misfolded proteins defines another phenotype indicating whether the cells is in the prion state or is prion free. Cell morphology serves as another characteristic to define a phenotype in *C. albicans*, where white cells are more round than opaque cells which appear elongated. In colonies, the collective body of cells can constitute a colony-level phenotypes (middle), where experimental manipulations help establish colony-level phenotypes observable in experiments (right). Image of *C. albicans* cells comes from Bettauer et al [11].

of infectious misfolded proteins capable of inducing further misfolding called “prions”, a term originally coined by Prusiner [126, 127] to describe these as proteinacious infectious particles (PrP) in humans. Prion diseases have not been formalized until the 1960s, when Prusiner discovered that misfolded proteins serve as the infectious agents behind the spread of all prion diseases, a key component in the proposition of the prion hypothesis [165]. To this day however, the study of biological processes behind prion disease and the search for appropriate solutions to eradicate them remains an active area of research.

Prions are not exclusive to mammals though. Proteins that share the templating and aggregation properties are also present in yeast such as *Saccharomyces cerevisiae* and have been studied extensively. At least eight naturally occurring yeast prion proteins [26, 93, 175] have been studied in this species alone and have helped set the stage for screening potential candidates for anti-prion drugs [76]. One of the first and most widely studied prion protein in yeast is Sup35, an essential release factor in the translation-termination process [99, 164]. Aggregates of misfolded Sup35 protein have the ability to self-propagate within yeast populations [120]. At the colony scale, experimental manipulations that rely on chromogenic assays allow for visualization of colony regions containing aggregates. These regions of aggregated Sup35 are indicative of a particular phenotype, namely $[PSI^+]$, whereas regions absent such aggregates are indicative of the $[psi^-]$ phenotype. In colonies the $[PSI^+]$ phenotype can be lost through multiple generations of cell division, resulting in sector-like regions, each with their own phenotype.

Currently, models used for the purpose of studying the spread of prions proteins in yeast are limited due the complexity of the system. The dynamics of Sup35 protein interaction is an intracellular process, while the process of cell reproduction and colony expansion is largely intercellular. Therefore, an understanding of the dynamics of misfolded proteins across an entire colony involves an interplay of dynamics in multiple spatial scales (Figure 1.2). To overcome this limitation, we attempt to get a better understanding of how collective cellular behavior in a growing colony contributes to the formation of well-defined phenotype structures. The case of how a growing budding yeast colony affects the formation of these structures is detailed in Chapter 2. The inverse problem of using experimental $[PSI^+]$ and $[psi^-]$ colonies to gain insight into prion protein dynamics and phenotype formation is discussed in Chapter 3.

1.4 Reversible Phenotypes: White-Opaque Switch in the Yeast *Candida albicans*

Candida albicans is a yeast specifically found with the human gut microbiota [118] and often forms close community groups known as biofilms. These biofilms consist of two cell types of interest, namely cells that are round (white) and those that have elongated (opaque) or budding morphologies [118]. These two cell types were physi-

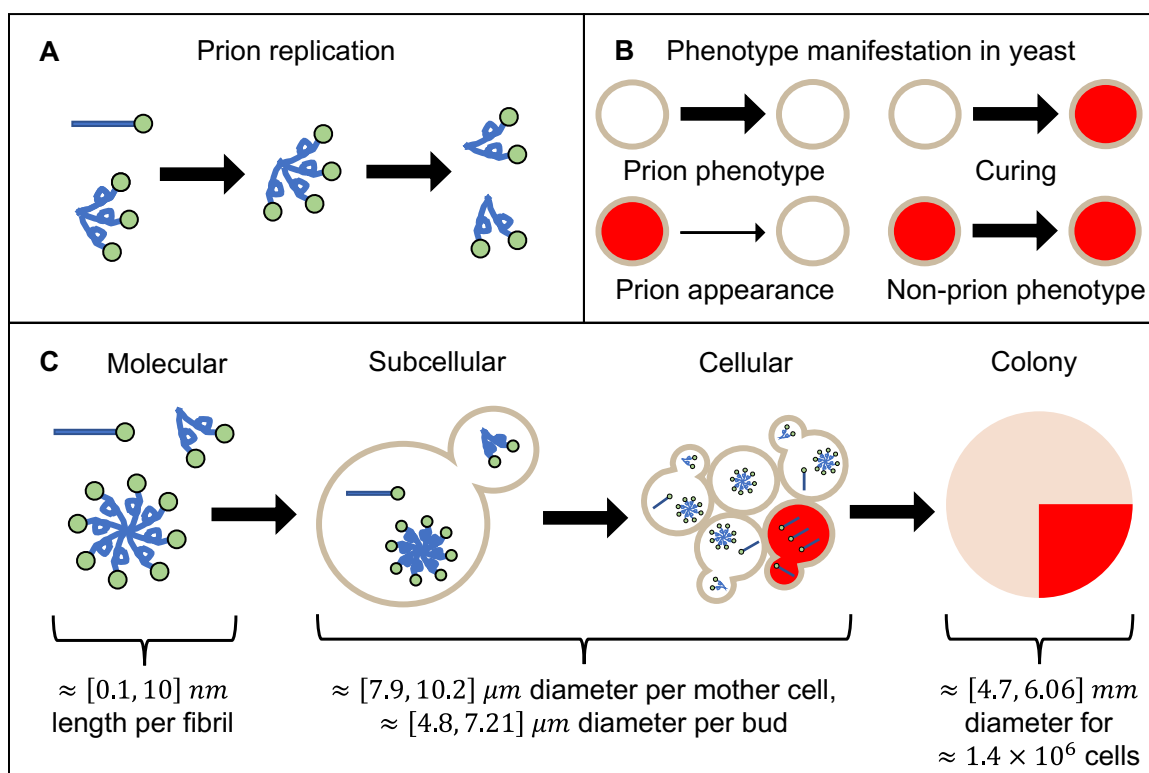


Figure 1.2: **Yeast prion phenotypes are the result of multiscale processes.**

A: At the molecular scale, alternatively folded proteins (twisted) act as templates that convert normally folded proteins (straight) into the alternatively folded form and assemble into aggregates. The aggregates then split into smaller segments (fragmentation) which increases the number of aggregates. B: At the cellular scale, the presence of prion aggregates inside individual cells (represented as circles) are responsible for their white color, while the absence of prions allows pigment generation and gives them their red color. The prion phenotype could be lost sporadically, resulting in cured cells, while in rare instances—1 in 10^6 —(indicated by a thinner arrow) the prion phenotype appears spontaneously. C: Phenotype expression in yeast involves multiscale processes. The dynamics inherent in protein misfolding are found at the molecular level (A). At the subcellular level, since prions are also found in yeast which undergo their own process of reproduction, allowing transmission of prions between attached cells. At the cellular level, the presence of prions within a cell in turn determines their phenotype (B). At the colony level, the collection of intercellular interactions that occur on the scale of a cell results in structured regions of one phenotype within the colony. Molecular scale was visually estimated from image data in [81]. Subcellular and cellular scales were estimated using data from [179]. A rough estimate for the colony scale was obtained using the minimum and maximum averaged surface area measurements of a mother cell in [179], multiplied by the approximate number of cells in colonies from data in [80].

cally observed as shiny and round to dull and flat respectively while also experiencing several other differing phenotypes such as metabolic and mating preferences [67]. In both *in vivo* and *in vitro* studies, both of these cell types exhibits multiple distinct physical characteristics.

In colonies that have grown over a longer period of time, differences in sizes of colonies of white and opaque cells are increasingly noticeable. Specifically, colonies consisting of mostly white cells tend to be much smaller than colonies of opaque cells. Using CHROM media for *in vitro* study of *C. albicans* colony growth allows for visual distinction between the two colonies by pigmenting opaque colonies as dark. However, a major difference is that the phenotype of a *C. albicans* cell is reversible; it can freely switch between the white and opaque state. Moreover, just like with $[PSI^+]$ and $[psi^-]$ phenotypes in *S cerevisiae*, colonies of *C. albicans* exhibit both phenotypes simultaneously, either as the presence of sectors or through imperfect circular growths.

Quantification of white-opaque switching is often done in small colonies where cells are individually imaged. Quantification of larger colonies on the other hand are less common in current studies. Due to the stochastic nature of living organisms including cells, many replicates are often needed to control for stochasticity. This in turn means many colonies have to be grown with the same treatment in order for inferences to be reliable. However, quantification of colonies is an expensive process, and is even more so when attempting to quantify phenotypes within colonies. Chapter 4 will detail a computational framework to more efficiently quantify many *C. albicans* colonies undergoing a white-opaque phenotypic switch.

1.5 Computational Approaches to Studying Yeast Colonies

1.5.1 Colony Growth Simulations

The formation of colony level phenotypes arises from the collective interaction of multiple cells. Complex phenotype organization at the colony level arising from a founder cell or from a small cluster of cells provides rich information that will help uncover relationships between molecular processes, individual cell behaviors, and phenotype transitions. Most biological studies do not provide a rigorous quantification of shape, size and structure between different sectoring phenotypes. However, the interplay between cellular spatial organization due to collective cell behavior has been emphasized in previous studies [57, 101]. For example, patterns of polarized growth and division have been shown to impact overall cellular organization, and cell-cell adhesion forces have been shown to impact how the population expands outward and how cells divide [33]. The effects of nutrient limited growth on growing microbial colonies have been studied to analyze the effect of low nutrient availability on filamentous growth [15], declining growth activity [72], the influence of diffusing

nutrients on colony morphology [163], and loss of diversity in colony regions with higher nutrient concentrations [109]. Given the body of evidence suggesting nutrient availability has a significant influence on colony-scale growth patterns, it is unclear why few experimental models of growing microbial communities attempt to accurately implement this feature in order to replicate experimentally observed colonies. A modeling framework for simulating phenotype heterogeneity in a growing colony that accounts for both collective cell interactions and nutrient availability is needed to further our understanding of how different colony phenotypes emerge and expand over time.

Agent based modeling is one approach for researchers to study colony growth and heterogeneous structures. A few models accounting for collective cell interaction have been proposed to provide insight into colony-level organization. The most notable of these studies comes from Wang et al [169] which considers the effect of cell aging on reproduction and colony structure. Smith et al [151] which relates the shape of growing *E coli* cells to the formation of close-knit structures of cells with the same phenotype. Nadell et al [112] generalizes this idea by investigating the impact of physical and biological parameters on spatial distributions of genetic lineages within a growing colony and how these structures are maintained over time.

One pitfall with modeling large colonies is that the computational cost significantly increases at a rate proportional to the number of cells in the colony. However, some colony behaviors are either unnoticeable or very minimal in small colonies compared to larger colonies. For example, in dense colonies, nutrient limitation becomes a significant factor in the formation of quiescent and necrotic zones where cells stop dividing or die respectively [41]. Moreover, pseudohyphal growth in bacterial colonies becomes noticeable after a few days of colony growth as a result of limited nutrients [14]. Modeling studies aimed at accounting for unusual behaviors present in large colonies should be feasible enough to simulate large numbers of cells with this colony behavior present.

An important feature not emphasized in these studies is the effect of budding on colony structure. Models for colony growth have largely considered fission as the reproductive process for individual cells. However, the biophysical role of budding, especially for *S. cerevisiae*, has not been explicitly featured in yeast modeling simulations where budding is the primary reproductive mechanism, nor have previous studies investigated the effect that budding has on colony-level organization. The model proposed in Chapter 2 of this dissertation is one of the first large scale colony models to explicitly incorporate budding as a biological component of colony growth to study the formation of colony-level phenotypes.

1.5.2 Image Analysis and Deep Learning Applications

With the availability of greater processing power today, methods and software for microbial colony quantification are able to handle greater quantities of data and analyze them more efficiently. Such methods include software and image-based meth-

ods to automate microbial colony counting [29, 87, 162], edge detection [24] and for circular objects, the circle Hough transform [6, 73]. In addition, machine learning methods are a popular new direction for studying microbial cells and colonies. Deep learning is a useful inverse problem to study phenotype formation because it allows the resulting experimental data to be used as input and a means of training a model to learn features of interest present in the data.

Deep learning is an approach more commonly used in computer vision for analysis of image data. These methods when applied to image data typically have one or two objectives. One class of methods, namely image classification, takes in an image as input and maps that image to a set of predefined labels as output. These methods are useful for labeling images with specific user-defined features [107, 147, 156]. The second class of methods, image segmentation, is a generalization of the first, where the output of such methods include pixel-wise label assignment [65, 136] or a transformation of the original image into a new image. Unlike image classification, image segmentation opens a wider window to the problem of counting objects by including spatial context such as location. Both methods are applicable in series such as the model proposed by Carl et al [25] which uses segmentation to remove background noise before performing classification on the regions of interest. Combining both methods effectively allows for robust feature extraction of complex data from images that traditional methods fail to capture.

A major problem in any machine learning application is that the quantity of data must be sufficient enough for a machine learning model to reliably segment or classify real data. When the quantity of data is insufficient, data generation may be employed to address the shortage of available data. In the case where data is in the form of images, either traditional methods or other deep learning methods such as generative adversarial networks can create realistic images that are similar to the actual image data. This is a simpler transfer learning approach, where models trained on synthetic data can be applied to real data. Combining synthetic data with real data has been used previously in training image classification models [123] and has shown improvement over models traditionally trained with only the real data.

Deep learning has been applied to images of cells to efficiently distinguish between various cell types. For *C. albicans* this involves locating white and opaque cells [11]. However, little published work exists for eye-level colony quantification with deep learning approaches; the data available is highly specialized for cell-level or small-scale colony-level analyses.

This dissertation will cover two computational frameworks aimed at performing efficient analyses on large-scale colony-level image data to uncover mechanisms driving colony level-phenotypes in experiments. The deep learning framework discussed in Chapter 3 of this dissertation proposes a way to quantify sectoring patterns in images of [*PSI*⁺] and [*psi*⁻] colonies in order to gain insight into the formation of colony phenotypes due to the presence of Sup35 aggregates in individual yeast cells. The deep learning framework discussed in Chapter 4 will present a different and simpler approach which is directly aimed at quantifying phenotypic switching in images of *C.*

albicans colonies.

1.6 Dissertation Structure

This dissertation investigates work on using agent based modeling of a growing yeast colony to study the growth of sector-like structures and related these growths to experimental colony phenotypes. I will also describe the deep learning frameworks used to study rich image datasets of colony phenotypes in different microbial colonies and address issues with colony quantification from the image data available.

In Chapter 2 I discuss the construction of a novel two-dimensional cell-based model for studying the growth, movement, structure, and spatial organization of a growing colony of yeast cells. Here, we also emphasize that the role of the budding process has a significant effect on colony structure and growth and provide reasons for why budding should be included in models exhibiting yeast colony growth. We also show that a nutrient limited environment for controlling cell reproduction also has a significant impact on the organization of colony substructures and can help explain the formation of sectors which can be observed at the colony level. We then show how these insights help provide new interpretations about observed sectored phenotypes in yeast.

In Chapter 3 I discuss the construction of a computational pipeline for analysis of colony-level yeast data. I present a computational pipeline called the [PSI] Colony Image Classifier ([PSI]-CIC) for segmenting and quantifying individual colonies of *S. cerevisiae* found in image data using both deep learning and conventional detection tools. I show that utilizing deep learning as a preprocessing step aids in overcoming a challenge of isolating colonies using traditional edge detection methods when multiple phenotypes are present. I then show we can accurately quantify sectoring in images where prion curing is induced by heat shock. To conclude the chapter, we discuss the potential impacts of [PSI]-CIC on the use of image segmentation in the context of studying prion dynamics in yeast.

In Chapter 4 I begin discussing how the tools developed up to this point can be applied to images of *Candida albicans*. I present a similar pipeline for more efficient quantification of colonies that undergo a white-to-opaque phenotype switch. Unlike the classification structure of [PSI]-CIC from Chapter 3, this pipeline aims at using traditional detection methods for isolating colonies, then using a different deep learning approach for colony classification. Two deep learning models have been constructed to allow for an image and metadata to be added as input and tested on a series of images across different experimental setups. I show that each of these methods achieves a high classification accuracy overall on *Candida albicans* colony images, then conclude with a discussion on generalizing the model to larger datasets.

In Chapter 5 I continue the work discussed in Chapter 4 by putting the generalizability of this model to the test on a more diverse dataset. Furthermore, I will discuss progress on addressing challenges in colony counting with traditional detection methods. This chapter explores the use of multiple yeast colony images to address

what elements of images make it challenging for detection methods to successfully locate colonies. In particular, we show that a colony's position, including in a cluster of other colonies, lighting aberrations, and size play roles in hiding features that algorithms use to isolate a colony from an image. We then show how the inclusion of deep learning can help address some of these issues while also pointing out their shortcomings.

I then conclude my dissertation in Chapter 6 with a summary of the key findings presented in each chapter and a discussion of open questions that arise as a result of what is uncovered in these studies. To facilitate further research, I will also provide suggested future directions for these projects.

Chapter 2

Quantifying the Biophysical Impact of Budding Cell Division on the Spatial Organization of Growing Yeast Colonies

This chapter is the text of the paper I co-authored with Dr. Mikahl Banwarth-Kuhn and Dr. Suzanne Sindi which was published in the Journal of the Dynamic Models in Biology and Medicine Volume 2 [8]. I provide more details about the contributions of this chapter below.

Both Dr. Mikahl Banwarth-Kuhn and I contributed equally to writing the final versions of the Introduction (Section 2.1), Discussion (Section 2.4), and Conclusion (Section 2.5) respectively at the time of submission.

I led the development of the colony metrics (Section 2.2.2) used to analyze the colony structure and organization over time. I also formally defined the notion of a subcolony and applied the method of partitioning cells into their respective subcolonies in the biophysical model.

My co-author Dr. Mikahl Banwarth-Kuhn wrote, tested, and executed the scripts that simulated the biophysical model of a growing yeast colony (Section 2.2.1). This includes the formalization of the intercellular forces, cell cycle, and nutrient limitation governing the spatial cellular mechanisms over time. Dr. Mikahl Banwarth-Kuhn also created the majority of the final versions of the text, plots, figures, and tables shown in the Results section of this chapter (Section 2.3).

2.1 Introduction

The morphological characteristics exhibited by growing microbial communities arise from complex interactions between genetic, epigenetic, environmental and cellular determinants [2, 3, 9, 42, 55, 57, 63, 64, 78, 83, 94, 109, 131, 144] (Figure 2.1). For example, the emergence of large regions of a single genotype in bacterial colonies is

most often associated with the chance loss of certain genes as individuals die or do not reproduce due to nutrient limitation [63, 64, 109]. In this scenario, the survival or extinction of an individual depends on relative fitness or physical interactions with neighboring cells [57]. In yeast colonies, sectoring patterns appear when cells transition between different phenotypic states [55, 83, 84, 90, 97, 106, 119, 132, 177]. One example in *Saccharomyces cerevisiae* is the appearance of sectoring in yeast prion phenotypes. In the non-prion state ($[psi^-]$), cells establish red colonies; however, when prion aggregates are present ($[PSI^+]$), *S. cerevisiae* colonies exhibit different colors ranging from white (strong) to shades of pink (weak). Under certain experimental conditions, changes in protein aggregation dynamics between neighboring cells result in sectors corresponding to loss of the prion phenotype (Figure 2.1 D). Other examples of sectoring in yeast colonies include spontaneous mitotic crossover [84, 90] and the white-opaque switch in *Candida albicans* [55, 97, 106, 132, 177]. In each case, the complex phenotypic organization that arises from an initially small group of cells, provides a rich data set that can be used to uncover relationships between molecular processes, individual cell behaviors and phenotypic transitions at the colony level (Figure 2.1). At this point, most biological studies do not provide rigorous quantification of shape, size and structure between different sectoring phenotypes. As such, characterizing the role of individual cell behaviors in directing spatial organization of cells, as well as quantifying their impact on overall heterogeneity and disease progression within microbial communities is an underexplored opportunity for discovery. In this study, we propose a novel mathematical and computational framework that depicts realistic biophysical division processes and the effect of nutrient limitation and use our model to study how these processes impact colony organization.

In yeast, and other microbial colonies, cells grow closely together, and the cumulative effect of mechanical interactions at the microscopic scale impacts the overall shape and organization of growing colonies [57, 101]. For example, patterns of polarized growth and division have been shown to impact cellular organization, and cell-cell adhesion forces have been shown to impact how the population expands outward and how cells divide [33]. An interesting feature of *S. cerevisiae* is that cells undergo an asymmetric division process called budding (Figure 2.1 and Figure 2.4) [21, 27, 48, 85, 117]. During budding division, new daughter cells form as protrusions on the surface of the mother cell and stay attached until they reach a mature size and physically separate. After separation the resulting mother and daughter cells are unequal in size and the daughter cell does not inherit the replicative age of the mother. The creation of a large mother-daughter cell pair during budding division results in distinct biophysical properties from fission (i.e. non-budding division), a process by which cells split symmetrically into two daughters. Moreover, this difference leads to small perturbations in the physical interaction between neighboring cells and could act as a mechanism driving emergent patterns of organization. Thus, in the case of *S. cerevisiae*, understanding the impact of budding division and other individual cell behaviors on spatial relationships between cells may be paramount to understanding the evolution of phenotypic organization such as prion sectors.

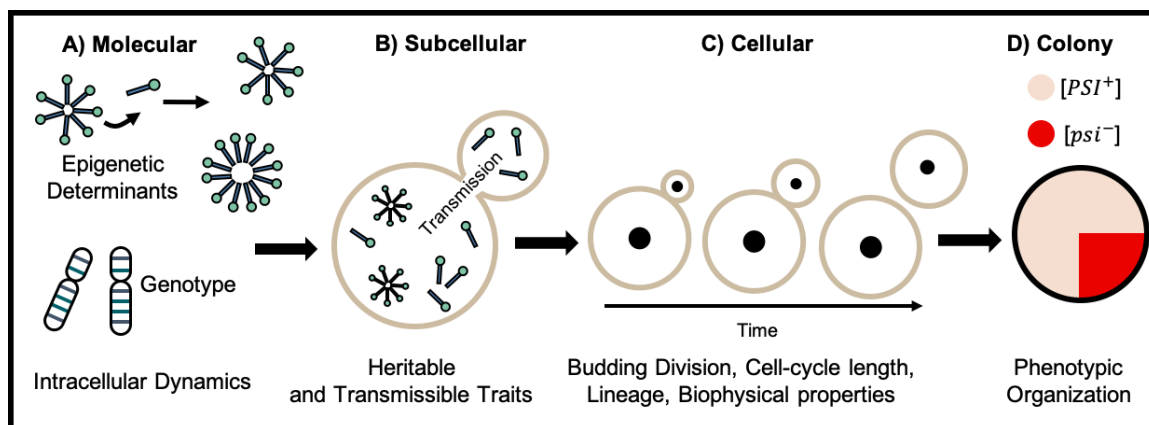


Figure 2.1: **Spatial Phenotypes are the Consequence of Processes at Different Scales.** (A) Cells transition between different phenotypic states due to genetic mutations or epigenetic determinants. For example, alternative conformations of the prion protein in *S. cerevisiae* can function as epigenetic determinants of transmissible phenotypes. (B) Daughter cells inherit their phenotype from their mother. In some cases, inefficient transmission of different intracellular constituents (i.e. prion aggregates) can lead to loss of phenotype. (C) Individual cell behaviors impact the propagation, loss and spatial arrangement of phenotypes within the colony. In this chapter we investigate the impact of budding division in *S. cerevisiae* on overall shape, size and spatial organization of cells. During budding division, the new daughter cell forms as a bud on the mother cell and remains attached until it reaches a mature size and they physically separate. (D) The outcome of processes at the molecular, subcellular and cellular scales lead to different morphological traits such as sector-like regions in *S. cerevisiae* colonies where all cells have lost the prion phenotype.

In addition to biophysical properties of cells, an environmental factor that impacts cell behaviors is nutrient limitation [59, 64, 109, 112, 158, 161]. Namely, the availability of required nutrients limits cell growth progression consequently slowing or stopping reproduction [64]. Regulatory pathways governing growth and quiescence in yeast cells are well-studied [18, 104]. Combined experimental and computational studies suggest that nutrient limitation is a key mechanism driving the emergence of organizational structures in growing microbial communities [109, 112]. For example, Mitri et al. [109] observed that *Pseudomonas aeruginosa* colonies with abundant resources expand more quickly and maintain large unstructured regions while their low nutrient counterparts have more spatio-genetic structuring. In addition, Nadell et al. [112] used an agent-based model (ABM) to investigate the impact of physical and biological parameters, including nutrient availability, on the spatial distribution of genetic lineages within microbial colonies. Recent studies provide further evidence that biophysical properties of cells can influence cell-cell interactions and change organizational dynamics within the colony [57, 82, 125, 151]. For example, Giometto et al. [57] showed that physical interactions of cells prolong the survival of less-fit strains at the growing frontier of *S. cerevisiae* colonies. While these studies provide compelling evidence that mechanical properties of cells and nutrient limitation serves as a combined mechanism driving spatial organization in microbial colonies, quantifying their individual impact in experiments is very difficult.

Mathematical and computational models have served as successful tools for investigating the role of individual cell behaviors and mechanical properties of cells on emergent patterns in multicellular tissues and growing microbial colonies. For example, cell-based models have been successfully used to capture passive biomechanical properties of cells during tissue development (for reviews see [58, 167]) as well as microbial biofilm formation [60]. However, computational models that focus on the impact of individual cell behaviors in directing spatial organization of yeast colonies is somewhat limited. Jönsson et al. [79] proposed an ABM to study the effect of cell division patterns and growth inhibition by neighboring cells on variations in the size and shape of growing *S. cerevisiae* colonies. In addition, Wang et al. [169] developed an ABM with several important biological processes including budding division, mating, mating type switch, consumption of nutrients, and cell death. They used their model to study the impact of different budding patterns and nutrient limitation on mating probabilities, colony development and colony expansion. In each of these previous studies, results focused on the colony as a whole – size, shape and expansion – and not how the colony itself was organized. An additional set of studies used agent based models (ABMs) to investigate the impact of individual cell growth and reproduction times on colony expanse as well as study the relationship between cell generation and birth location in the colony [1, 4, 5, 103, 130]. However, likely for computational simplicity, these prior studies focused primarily on populations of a few hundred or few thousand cells. To our knowledge, this represents the first biophysical model designed to study budding colonies that considers populations of more than 10,000 cells and emphasizes the impact of biophysical properties of cells on colony

organization.

In this Chapter, we study the impact of budding cell division on overall shape, size and spatial organization of growing yeast colonies. To do this, we develop a 2D ABM that explicitly includes the mechanical interactions that arise when the daughter cell is growing and physically connected to its mother. Spatial rearrangement of cells in our synthetic colonies depends on cell-cell interaction and our model incorporates several other important biological processes including, asymmetric cell cycle lengths between the mother and daughter cell and the impact of nutrient limitation. In addition, we develop specific metrics to quantify the spatial organization of cells that emerges due to different biophysical properties in budding and non-budding colonies. We then adapt our model to simulate colony growth in a nutrient-rich and nutrient-limited environment and discuss how nutrient limitation impacts global colony organization as it relates to sector-like regions formed by individual subcolonies. In Section 2.3.1 we analyze the impact of budding alone by considering colonies grown in an environment with an inexhaustible supply of nutrients. In Section 2.3.2 we study how a more realistic nutrient limited environment acts in concert with biophysical forces created by budding division to further impact colony organization. We find that (1) budding does not impact large-scale properties of the colony such as shape and size; (2) budding does impact local spatial organization of cells with respect to spatial layout of mother-daughter cell pairs and connectivity of subcolonies; (3) nutrient limitation further promotes local spatial organization of cells; (4) nutrient limitation changes global colony organization by driving variation in subcolony sizes. In Section 2.4 we discuss the implications of our work in understanding the appearance of sectoring patterns in growing yeast colonies and more broadly outline extensions of our model to further study prion sectors in *S. cerevisiae*. In Section 2.2, a detailed description of our model and the metrics we developed to quantify spatial organization of cells are given in the methods section. Finally, Section 2.5 offers our concluding remarks.

2.2 Materials and Methods

In this section, we develop the 2D, off-lattice, center-based model we use to simulate the growth of *S. cerevisiae* colonies (Figure 2.2) as well as describe the metrics we use to analyze simulation output. In the model, each cell is represented by an elastic sphere that moves, grows, buds and divides according to biophysical and cell-kinetic model parameters estimated from experiments (Table 2.1). We simulate artificial yeast colonies under two different division formulations (budding and non-budding) and growth conditions (nutrient rich and nutrient limited).

2.2.1 Computational Model

Below we develop our computational ABM for studying yeast colony growth and structure. In simulations of budding colonies we explicitly model the mechanical interactions during budding cell division by modeling the daughter cell as a growing

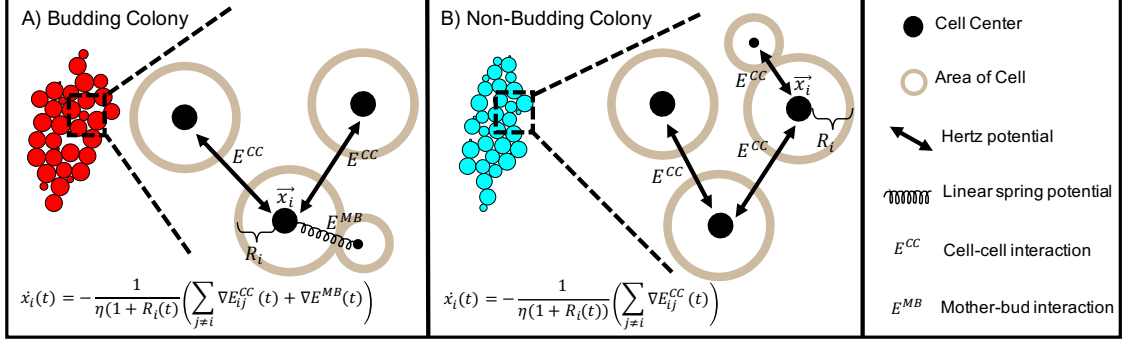


Figure 2.2: **Biophysical Model of Cell-Cell Interactions.** We simulate budding (A) and non-budding (B) colonies using a 2D center-based modeling approach where cells interact through different potentials. (A) For budding colonies, the mechanical interactions of all cell pairs (E^{CC}) are governed by a combination of repulsive and attractive interactions using a modified Hertz-model described in Equation (2.1). We use a linear spring to model the additional adhesive force between mother cells and new daughter cells during the budding phase (E^{MB}) as in Equation (2.4). (B) For non-budding colonies, the mechanical interactions are similar to that of cells in budding colonies except that the adhesive force between mother cells and new daughter cells during the budding phase (E^{MB}) is neglected (see Section 2.2.1 for details).

circle attached to the mother cell with a stiff spring (Figures 2.2 A and 2.4). In contrast, in simulations of non-budding colonies we treat mechanical interactions of mother-bud pairs as identical to other cell-cell pairings (Figure 2.2 B).

Cell-Cell Interaction and Spatial Arrangement of Cells

We assume the resting shape of individual yeast cells is circular, and track the size and location of cell i at time t by its radius $R_i(t)$ and center $\vec{x}_i(t) = (x_i(t), y_i(t))$ (Figure 2.2). We track the total number of cells at time t , $N(t)$, and index cells by their birth order $i \in \{1, 2, \dots, N(t)\}$. Since all cells are of the same type, the mass of each cell, $m_i(t)$, is proportional to the area of each cell with the same constant. Yeast cells interact through different potentials that we use to represent biologically-relevant processes seen in experiments (Figure 2.2). For example, yeast cells in physical contact form adhesive bonds that result in an attractive force [5, 19, 43, 79, 169]. However, due to the incompressibility of their cell wall, yeast cells also resist compression from neighboring cells with a repulsive force [5, 79, 149, 155, 169]. We represent the combination of repulsive and attractive interactions between cells using a modified Hertz-model, as has been previously done [22, 45, 47, 71], where the potential $E_{ij}^{CC}(t)$

between two cells i and j is given by:

$$E_{ij}^{CC}(t) = \frac{(R_i(t) + R_j(t) - d_{ij}(t))^{5/2}}{5\tilde{E}_{ij}} \sqrt{\frac{R_i(t)R_j(t)}{R_i(t) + R_j(t)}} + E_{ij}^{\text{adh}}(t). \quad (2.1)$$

The first term of Equation (2.1) depicts the repulsive interaction between two cells and $d_{ij}(t) = \|\vec{x}_i(t) - \vec{x}_j(t)\|$. In this equation $\tilde{E}_{ij}(t)$ is defined by:

$$\tilde{E}_{ij} = \frac{3}{2} \left(\frac{1 - \sigma^2}{E} \right) \quad (2.2)$$

where E and σ are the Young's moduli and Poisson ratios of cells, respectively. The second term of Equation (2.1) models the adhesive interaction between cells and is given by:

$$E_{ij}^{\text{adh}}(t) = \phi W_s A_{ij}(t) \quad (2.3)$$

where ϕ is the density of surface adhesion molecules in the contact area, W_s is the single bond bind energy and $A_{ij}(t) = (R_i(t) + R_j(t)) \times 0.5$ is the contact area between cells i and j .

In simulations of non-budding colonies, the mechanical interactions of all cell pairs are given by Equation (2.1). However, in simulations of budding colonies, we explicitly model an additional force between mother cells and new daughter cells during the budding phase (Section 2.2.1, Figure 2.2 A and Figure 2.4). To do this, we represent the adhesive interaction caused by attachment of the new daughter cell to its mother using a linear spring potential given by:

$$E_{mb}^{\text{MB}}(t) = K_{\text{bud}} (d_{mb}(t) - (R_m(t) + R_b(t)))^2 \quad (2.4)$$

where $d_{mb}(t) = \|\vec{x}_m(t) - \vec{x}_b(t)\|$, $R_m(t)$ is the radius of the mother cell at time t , $R_b(t)$ is the radius of daughter cell at time t and K_{bud} is a spring constant chosen large enough to ensure that the new daughter cell b remains attached to its mother for the duration of the budding phase and is not pushed away due to forces from other neighboring cells.

In addition, we assume that cells are in an overdamped regime so that inertial forces acting on the cells are neglected [51,86,114]. This leads to the following equation of motion describing the movement of an individual yeast cell i in a budding colony:

$$(\eta(1 + R_i(t)/2)) \dot{x}_i(t) = \begin{cases} - \left(\sum_{i \neq j} \nabla E_{ij}^{CC}(t) + \nabla E_{ij}^{\text{MB}}(t) \right) & i \text{ is a bud} \\ - \left(\sum_{i \neq j} \nabla E_{ij}^{CC}(t) + \nabla E_{ij}^{\text{MB}}(t) \right) & i \text{ is a mother with bud} \\ - \left(\sum_{i \neq j} \nabla E_{ij}^{CC}(t) \right) & \text{else} \end{cases} \quad (2.5)$$

where j indexes the other cells in the colony at time t and η is the damping coefficient that represents viscosity of the growth media and is scaled by $(1 + R_i(t)/2)$. The equation of motion describing the movement of all cells in a non-budding colony simplifies to only the third case in Equation (2.5).

The equation of motion of a cell is discretized in time using the forward Euler method, and the position $\vec{x}_i(t)$ of cell i at time t is given by:

$$\vec{x}_i(t + \Delta t) = \vec{x}_i(t) - \left(\sum_{i \neq j} \nabla E_{ij}^{CC}(t) + \nabla E_{ij}^{MB}(t) \right) \frac{\Delta t}{\eta(1 + R_i(t)/2)} \quad (2.6)$$

where Δt is the time step size. The same discretization technique is used for all cells in each simulation.

Budding Cell Division

S. cerevisiae cells undergo budding cell division [21,117]. During this process, one large mother-daughter cell pair is formed by the appearance of a bud on the mother. The bud (or new daughter) remains attached while it gradually grows into a larger cell (Figure 2.4). At the time of division, the mother cell and new daughter become physically separated resulting in two unevenly sized cells. After division, a bud scar is left on the surface of mother cell at the location where the new daughter was formed, and no subsequent buds can be formed at that site (Figure 2.3). Similarly, a birth scar is left on the surface of the new daughter cell.

The location of the bud on the surface of the mother cell can be chosen according to two distinct patterns, axial or bipolar [48,79,85,117,169]. In our model we follow [169] and model budding cell division with the the following pattern: mother cells are equally likely to choose a new bud location adjacent to or opposite from the previous bud location, and daughter cells always bud opposite to their birth scar (Figure 2.3). To ensure no bud/birth scars are used twice, we keep track of all previous bud/birth sites for every cell. If the next choice for a bud site falls on a previously used location we adjust the location of the new bud site by increments of 10° in either the clockwise (probability = .5) or counterclockwise (probability = .5) direction until we arrive at a location with no previous bud/birth scar (Figure 2.3 (Left)). The budding location of the founder cell’s first daughter is chosen randomly and uniformly along its boundary.

Cell Growth and Cell Cycle Length

We follow the standard model of eukaryotic cell division and consider the cell cycle to have two distinct growth phases: $G1$ and $G2$. At the time of separation, mother and new daughter cells are unequal in size. Thus, new daughter cells undergo an extended $G1$ phase in order to grow to a mature adult size before producing their own bud [17] (Figure 2.4). In our model, the average cell cycle length for mother cells is ≈ 90 minutes (~ 15 minutes in $G1$ and ~ 75 minutes in $G2$) and the average cell cycle length for new daughters cells is ≈ 120 minutes (~ 75 minutes for the “Budding” phase while attached to their mother and ~ 45 minutes growing on their own). To depict more realistic cell cycle dynamics, we introduce an element of stochasticity to the cell division times.

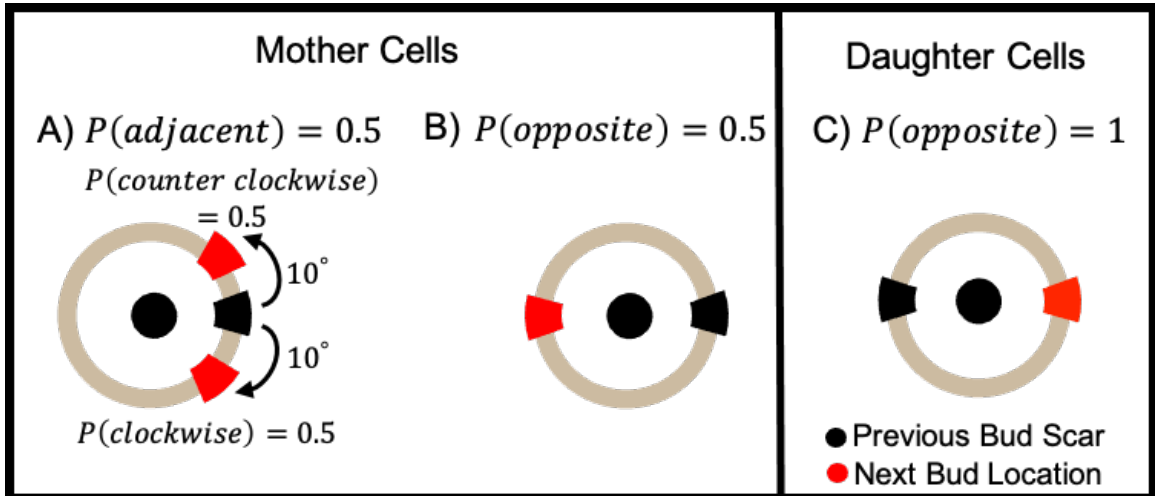


Figure 2.3: **Selecting a Bud Site.** The choice of the next bud location for a cell depends on whether it is a mother cell (left) or a new daughter cell (right). (Left) For mother cells, the next bud location will be chosen either adjacent to the previous bud scar with probability 0.5 (A) or opposite to the previous bud scar with probability 0.5 (B). In the case when the location of the new bud site overlaps with a previous bud site (A), we adjust the location of the new bud site by increments of 10° in either the clockwise (probability 0.5) or counterclockwise (probability 0.5) direction until we arrive at a location with no previous bud scar. (Right) For new daughter cells the next bud location (red) is chosen opposite to the previous birth scar (black) with probability 1. New daughter cells that have successfully completed a full cell cycle are considered mother cells for the remainder of the simulation. (See Section 2.2.1.)

The variable **Cell Progress** ($CP \in [0, 1]$) is used to track the progress of individual cells through the $G1$ and $G2$ phases. In our model, the progress of cell i at time t is given by:

$$CP_i(t) = CP_i(t - \Delta t) + CI_i \times \Delta t \quad (2.7)$$

where $CI_i = (G1_i + G2_i)^{-1}$. The length of $G2_i$ is computed once for all cells:

$$G2_i = (1 + \mathbf{U}[-0.1, 0.1]) G2_{\text{avg}}.$$

(In the previous expression $\mathbf{U}[-0.1, 0.1]$ is a uniformly distributed random variable on the interval $[-0.1, 0.1]$.) To represent the longer $G1$ phase of daughter cells, the length of $G1_i$ is assigned once upon creation of a new bud

$$G1_{i_{\text{new daughter}}} = (1 + \mathbf{U}[-0.1, 0.1]) G1_{\text{avgdaughter}}$$

and then updated once the new daughter cell completes its first $G1$ phase and forms a bud of its own

$$G1_{i_{\text{mother}}} = (1 + \mathbf{U}[-0.1, 0.1]) G1_{\text{avgmother}}.$$

In the $G1$ phase, mother cells have already reached their adult size, so the $G1$ phase is simply a waiting time until entering the $G2$ phase where they form a bud (or new daughter). Every new bud is initiated with a radius of size $0\mu\text{m}$ and grows for ≈ 75 minutes while attached to its mother. (This 75 minutes of attachment accounts for the entire $G2$ phase of the mother and make up the first part of the $G1$ phase for the daughter.) After this phase, the mother and bud are physically separated resulting in a new daughter cell. At separation, the mother cell enters the $G1$ phase, and the new daughter cell stays in its $G1$ phase and continues to grow for ≈ 45 minutes until it reaches its adult radius size (Figure 2.4). At this time, the daughter cell transitions into a mother cell and begins to produce its first bud.

The adult size, corresponding to a maximum radius R_{max} , is assigned to each cell upon creation and set to

$$R_{i,\text{max}} = (1 + \mathbf{U}[-0.1, 0.1]) R_{\text{avg}}.$$

The radius of cell i at time t is given by:

$$R_i(t) = \begin{cases} R_i(t - \Delta t) + \frac{R_{i,\text{max}}}{G1_i} \times \Delta t & R_i(t) \leq R_{i,\text{max}} \\ R_{i,\text{max}} & \text{else.} \end{cases} \quad (2.8)$$

Nutrient Limited Growth

Until now, we assumed that the environment cells were in contained an inexhaustible nutrient supply and cell maturation and division occurs at the same rate no matter how many cells were present. We now revisit this assumption by modeling the growth of individual yeast cells as dependent on a local nutrient supply. That is, a depletion in nutrient concentration slows down individual cell growth

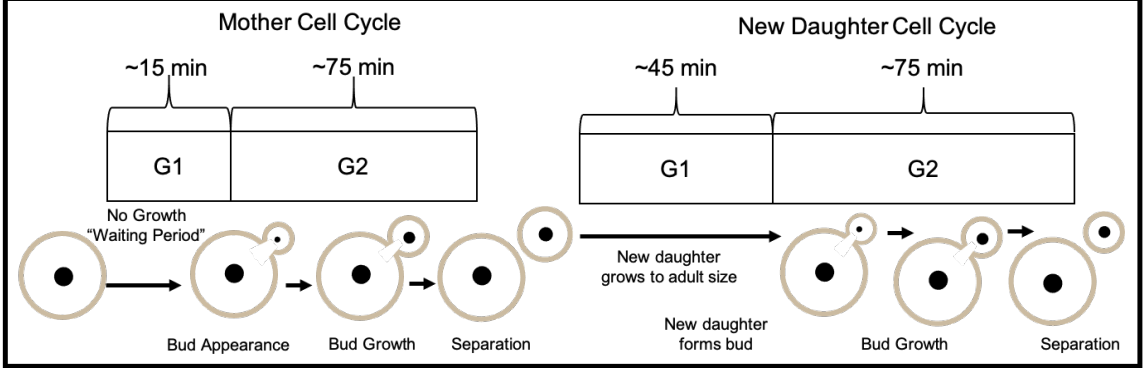


Figure 2.4: **Cell Cycle Length.** (Left): The $G1$ phase for mother cells is approximately 15 min. Since mother cells have already reached their adult size, the $G1$ phase serves as a waiting period before the mother cell enters $G2$ and forms a bud. When the mother cell enters $G2$, the new daughter cell forms as a bud and stays attached for ~ 75 min as it grows. After ~ 75 min, the mother and new daughter physically separate resulting in two unevenly sized cells. At this time, the mother cell enters $G1$ and begins a new cell cycle. (Right:) The new daughter cell continues to grow until it reaches its adult size (~ 45 min) and forms its own bud. Under nutrient limited conditions, the length of the $G1$ and $G2$ phases are increased for both mother and daughter cells (see Section 2.2.1 for details).

by prolonging the cell cycle length. Previous studies have incorporated the effect of enzyme and/or nutrient concentration on individual cell behaviors in ABM models of microbial colony growth [44–47, 72, 109, 112, 169, 170]. The majority of these studies use reaction-diffusion equations that include the uptake of growth substrate by each cell to compute spatial gradients of enzyme or nutrient concentration. For simplicity, we consider each region of our simulation domain to have a maximal possible biomass (i.e. carrying capacity). We divide the simulation domain into smaller subdomains and adjust the cell cycle progression $CP(t)$ for cells in each subdomain j at time t as follows.

First, for each cell i we track the subdomain the cell is in denoted $D_i(t)$ and compute the total mass of cells in each subdomain j where $m_i(t) = \pi R_i(t)^2$:

$$M_j(t) = \begin{cases} \frac{\sum_{i=1}^{N(t)} m_i(t) \mathbb{I}_j(D_i(t))}{M_{j,\max}}, & \text{if } \sum_{i=1}^{N(t)} m_i(t) \mathbb{I}_j(D_i(t)) \leq M_{j,\max} \\ 1 & \text{else} \end{cases} \quad (2.9)$$

where \mathbb{I}_j is an indicator variable that equals 1 if $D_i(t) = j$ and 0 otherwise. Note that in practice $M_{j,\max}$ is chosen large enough that $\frac{M_j(t)}{M_{j,\max}}$ is always less than 1.

Next, we define a growth-rate adjustment factor for each subdomain that is initialized to 1 at the beginning of simulations and decreases in time according to the

following equation:

$$GR_{\text{adjust}}(t) = \begin{cases} GR_{\text{adjust}}(t - \Delta t, j) - rM_j(t)\Delta t, & \text{if } \frac{GR_{\text{adjust}}(t - \Delta t, j)}{\Delta t} \geq -rM_j(t) \\ 0 & \text{else} \end{cases} \quad (2.10)$$

where $M_{j,\text{max}}$ is the carrying capacity for the j -th subdomain and r is the per capita rate of decrease of GR_{adjust} . We then re-scale the cell cycle increment:

$$\widetilde{CI}_i(t) = CI_i(t) \times GR_{\text{adjust}}(t, D_i(t)) \quad (2.11)$$

and thus the cell progression in nutrient limited growth becomes

$$CP_i(t) = CP_i(t - \Delta t) + \widetilde{CI}_i(t) \times \Delta t. \quad (2.12)$$

As the colony grows, cells move, new cells are born and cells are displaced into new subdomains. To account for this we calculate the growth rate adjustment factor for each subdomain at each timestep, and use it to update $\widetilde{CI}_i(t)$ for each cell according to the unique subdomain it occupies at time t .

Simulation Run Time

In our simulations, the number of cells in synthetic yeast colonies reaches $\approx 15,000$. Since our model requires computing the force between all cell pairs, the number of computational operations is proportional to the square of the number of cells. In order to decrease the computational cost of our simulations, we use a search algorithm that makes the number of computational operations asymptotically linear to the number of simulated cells. To do this, the total area occupied by cells is divided into S square subdomains. (Note these are the subdomains used for the nutrient model as described in Section 2.2.1). The size of the subdomains in simulations is determined based on the longest distance at which two cells can interact with each other. Since cell-cell adhesion and repulsion interactions are short range, the search algorithm for computing cell-cell interaction forces is limited to only neighboring subdomains.

Since there are eight neighboring subdomains for each unique subdomain S_i , this algorithm reduces the total number of operations. In addition, the code for this work was implemented in C++ using OpenMP for parallelization. As a result, the total run-time of one simulation is ≈ 4 hours on a 20-core node.

2.2.2 Colony Metrics

In this Section, we define the metrics used in Section 2.3 to analyze yeast colony morphology and organization. We first use two previously defined metrics to describe overall colony size and shape [5, 79, 169] and later introduce new metrics to characterize spatial organization of cells within the colony.

Table 2.1: **Parameter Values Used in ABM.** Descriptions of the biophysical and biological processes corresponding to these variables are detailed in Section 2.2.

Parameter	Symbol	Value	Units	Meaning	Ref
Poisson ratio	σ	.3		Incompressibility of yeast cells	[46, 149, 167]
Young's Modulus	E	1000	kPa	Mechanical property of yeast cell walls	[46, 149, 167]
Receptor Surface Density	ϕ	10^{15}	m^{-2}	Density of surface adhesion molecules in the contact area	[46, 167]
Single Bond Binding Energy	W_s	$25k_B T$			[46, 167]
E^{MB} Linear Spring Constant	K_{bud}	25	$nN/\mu m$	Attachment of bud on mother cell	calibrated
Damping Coefficient	η	2.5	$Ns/\mu m^2$	Viscosity of the growth media	[46, 167]
Average Length of G2 phase	$G2_{avg}$	75	min		[39, 166]
Average Length of G1 phase (new daughters)	$G1_{avg,daughter}$	120	min		[39, 166]
Average Length of G1 phase (mothers)	$G1_{avg,mother}$	15	min		[39, 166]
Average Mature Radius Size	R_{avg}	2.58	μm		[105]
Carrying Capacity	$M_{j,max}$	$18\pi R_{avg}^2$	μm^2	Maximal possible biomass for each subdomain	calibrated
Subdomain Size	$D_i(t)$	25	μm^2	Area of each subdomain	calibrated
Rate of Maximum Cell Cycle Adjustment	r	.003		Controls the amount cell cycle is adjusted at each timestep	calibrated
Timestep	Δt	0.00144	min		calibrated

Colony Shape Metrics

The first two metrics are used to quantify the shape of the colony as it grows in time. The **colony expanse** quantifies how large the colony is, while the **colony sparsity** quantifies how circular the colony is. Both depend on the center of mass of the colony (Figure 2.5). Let $N(t)$ be the number of cells in the colony at time t , each of which has position $\vec{x}_i(t) = (x_i(t), y_i(t))$ and radius $R_i(t)$. Since we assume all cells are of the same type, the mass of each cell, $m_i(t)$, is proportional to the area of each cell with the same constant. As such, the center of mass of the colony at time t is given by the 2D point, $\vec{C}(t)$, defined by:

$$\text{Center of Mass: } \vec{C}(t) = \frac{\sum_{i=1}^{N(t)} m_i(t) \vec{x}_i(t)}{\sum_{i=1}^{N(t)} m_i(t)} = \frac{\sum_{i=1}^{N(t)} \pi R_i^2(t) \vec{x}_i(t)}{\sum_{i=1}^{N(t)} \pi R_i^2(t)}. \quad (2.13)$$

The colony expanse is defined as the largest distance between any cell boundary and the center of mass of the colony. That is:

$$\text{Colony Expanse: } E(t) = \max_{1 \leq i \leq N(t)} \left\{ \|\vec{x}_i(t) - \vec{C}(t)\| + R_i(t) \right\}. \quad (2.14)$$

The colony sparsity compares the area of the colony to the area of the circle with radius equal to the colony expanse. Notice that this circle need not be the smallest enclosing circle, as the smallest enclosing circle need not have its center at the colony center of mass, which is a modification of the colony sparsity used by Jönsson [79] and colony radius used by Aji [1]. In our simulations, cells do not overlap, so we define the area of the colony as follows:

$$\text{Area of Colony: } A_{\text{colony}}(t) = \sum_{i=1}^{N(t)} \pi R_i^2(t). \quad (2.15)$$

Thus, the colony sparsity is defined as:

$$\text{Colony Sparsity: } S(t) = \frac{\pi E(t)^2}{A_{\text{colony}}(t)}. \quad (2.16)$$

Colony Organization Metrics

Next we introduce new metrics related to the organization of cells within the colony. We define a graph $G(V, E)$ as a set of nodes V and edges E . In each graph, all cells are represented as nodes. To analyze the colony organization, we consider two undirected graphs which evolve dynamically along with the colony (Figure 2.6). The first graph, G_S , is based on the Delaunay triangulation [53, 89] and encodes spatial relationships between cells (Figure 2.6 A). The second graph, the lineage graph which we denote as G_L , encodes mother-daughter relationships between cells (Figure 2.6 B)

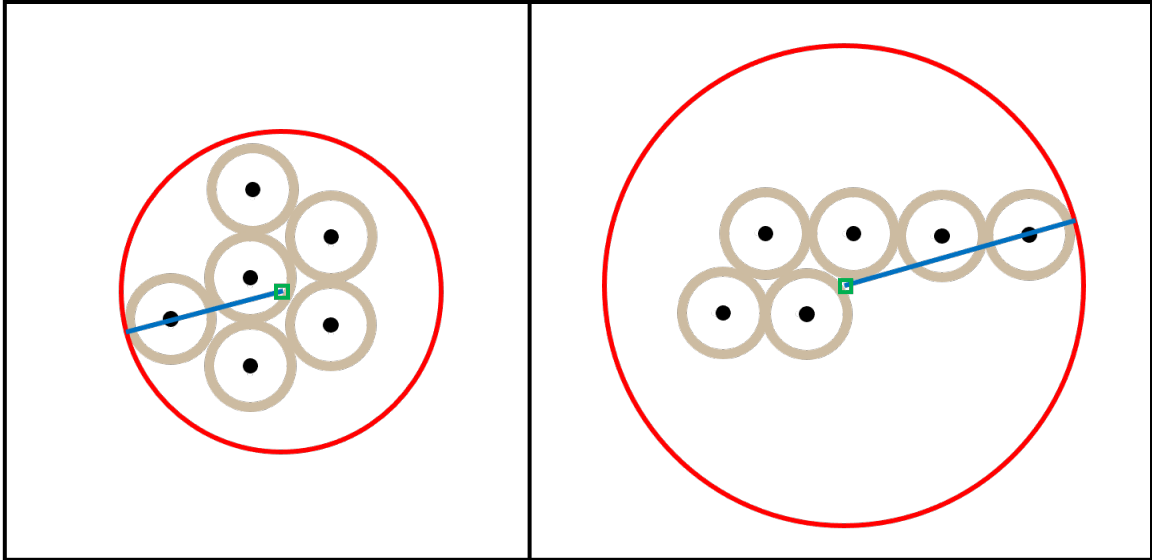


Figure 2.5: **Colony Sparsity and Expanse.** We first compute the colony center of mass (represented as a green square) using Equation (2.13). Then we determine the radius (length of the blue line) of the smallest circle which surrounds the entire colony centered at the center of mass (shown in red) using Equation (2.14). The colony sparsity is then computed using Equation (2.16) with the result of Equation (2.14), the area of the circle, and the total area of the cells using Equation (2.15). (Left): The space within the circle is more dense, thus covering more area within the circle with radius equal to the colony expanse, resulting in a small colony sparsity. (Right): The space that cells occupy within the circle is less dense, resulting in a higher colony sparsity.

Both graphs are constructed from the same vertex set, namely, all cells in the colony at time t :

$$V = \{c_1, c_2, \dots, c_{N(t)}\}.$$

The edge set E_S is constructed based on the Delaunay triangulation. The Delaunay triangulation is the dual graph of the Voronoi diagram for cell centers which consists of all points in the plane that are equidistant to their two nearest sites [172]. In our case, the edge set E_S is defined as:

$$E_S = \{(c_i, c_j) \mid \text{cell } i \text{ and } j \text{ share an edge in the Delaunay triangulation.}\}. \quad (2.17)$$

The edge set E_L consists of only edges between immediate mother-daughter cell pairs. Each cell, c_i , has a unique mother, $m(c_i)$. Thus, the edge set E_L consists of the following edges:

$$E_L = \{(c_i, m(c_i))\} \quad \text{for } i \in \{2, \dots, N(t)\}. \quad (2.18)$$

(Note that $m(c_1)$ is not defined because c_1 is the founder cell.)

Together $G_S = \{V, E_S\}$ and $G_L = \{V, E_L\}$ can be used to quantify how closely the spatial organization of the colony relates to mother-daughter cell pair interactions within the colony. To do this, we first define the intersection graph, G_I (Figure 2.6 C). G_I is constructed from the same vertex set containing all cells in the colony, but the edge set, E_I , only includes edges belonging to both E_S and E_L , namely:

$$E_I = E_S \cap E_L. \quad (2.19)$$

Our first colony organization metric, **colony connectivity**, is the fraction of mother-daughter edges that are also in the intersection graph:

$$\text{Colony Connectivity} := \frac{|E_I|}{|E_L|}. \quad (2.20)$$

Our second organization requires us to establish a few concepts related to colony structure. First, we define a **subcolony** to be the subset of all cells whose common ancestor is an immediate daughter of the founder (Figure 2.7). We index the daughters of the founder cell by d_1, d_2, \dots, d_F where d_k denotes the k -th daughter of the founder cell. Note, every cell in the colony belongs to one of the subcolonies founded by an immediate daughter of the founder cell. Thus, for each $c_i \in V \setminus \{c_1\}$, we define $\text{Sub}(c_i)$ to be the subcolony that cell i belongs to. Moreover, each daughter of the founder is considered to be the founder of its own colony (i.e. a subcolony of the original colony) denoted $\text{Sub}(d_k)$ and the total number of subcolonies is equal to F , the total number of immediate daughters of the founder cell.

We next define a graph associated with each subcolony. Let V_{sub, d_k} be the set of all cells that are in $\text{Sub}(d_k)$. That is,

$$V_{\text{sub}, d_k} = \{c_i \in V \setminus \{c_1\} \mid \text{Sub}(c_i) = \text{Sub}(d_k)\} \\ \text{for } i \in \{2, 3, \dots, N(t)\} \text{ and } k \in \{1, 2, \dots, F\}. \quad (2.21)$$

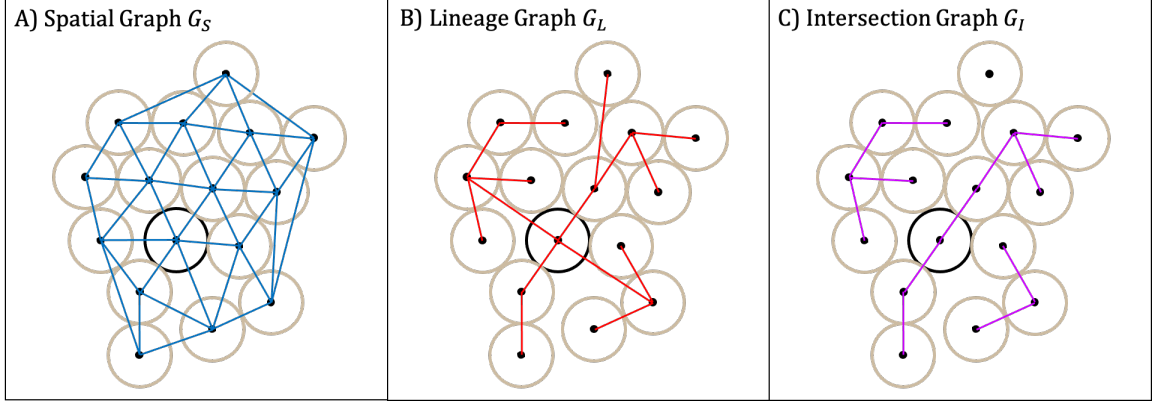


Figure 2.6: **Colony Spatial Graphs.** The vertex set for all three colony graphs is the same (all cell centers). The founder cell is designated in black. The edge set differs depending on the relationships between cells. (A): The edge set for the spatial graph G_S (blue edges) are those induced by the the Delaunay triangulation applied to the cell centers (Equation (2.17)). (B): The edges for the lineage graph G_L (red edges) correspond to mother-daughter pairs (Equation (2.18)). (C): The edge set for the intersection graph G_I (purple edges) include those edges that belong to the two previous edge sets (Equation (2.19)).

Let E_{sub,d_k} be the the set of all edges in E_S that join two cells in $\text{Sub}(d_k)$. Namely,

$$E_{\text{sub},d_k} = \{(c_i, c_j) \in E_S \mid \text{Sub}(c_i) = \text{Sub}(c_j) = \text{Sub}(d_k)\} \quad (2.22)$$

for $i, j \in \{2, 3, \dots, N(t)\}$ and $i \neq j$.

We define G_{sub,d_k} to be the subgraph of G_S whose vertex set is V_{sub,d_k} and whose edge set is E_{sub,d_k} (see Figure 2.7 (B) and (C)). Note that E_{sub,d_k} for each k partitions the larger edge E_{sub} defined as:

$$E_{\text{sub}} = \bigcup_{k=1}^F E_{\text{sub},d_k}. \quad (2.23)$$

Similarly, V_{sub,d_k} for each k partitions the larger vertex set V_{sub} defined to be:

$$V_{\text{sub}} = \bigcup_{k=1}^F V_{\text{sub},d_k}. \quad (2.24)$$

We then define the **subcolony graph** $G_{\text{sub}}(V_{\text{sub}}, E_{\text{sub}})$, where

$$G_{\text{sub}} = \bigcup_{k=1}^F G_{\text{sub},d_k}. \quad (2.25)$$

We define our second colony organization metric to be the number of **connected components** of G_{sub,d_k} for each subcolony. A connected component is defined to

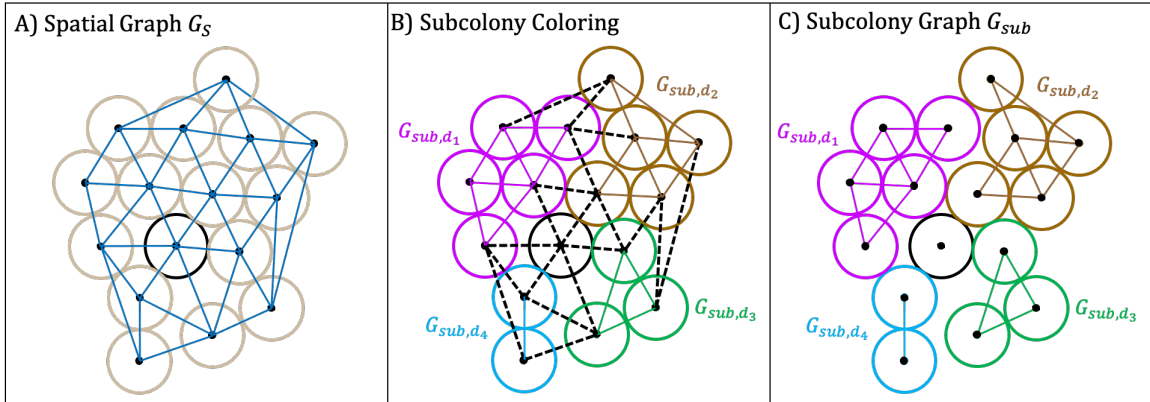


Figure 2.7: **Constructing Subcolony Graphs.** Subcolony graphs are constructed from partitions of the spatial graph G_S according to the following procedure. (A): We generate the the same spatial graph (G_S) using the Delaunay triangulation (Equation (2.17)). (The founder cell is indicated in black.) (B): We define a subcolony as a subset of cells in the lineage graph consisting of a daughter of the founder cell along with all of its descendants (Equation (2.21)). Edges are colored based one which subcolony each cell belongs to (Equation (2.22)). We then remove edges from the spatial connecting cells from different subcolonies (dotted black edges). (C): Removing these edges results in the subcolony graph G_{sub} , a set of subgraphs of G_S that we index by daughter cells: G_{sub,d_k} . These graphs preserve the spatial relationship between cells within the same subcolony (Equation (2.25)).

be any maximal subgraph $G_{connect} \subseteq G_{sub,d_k}$ such that any two vertices in $G_{connect}$ are connected by a path and not connected to any other vertices in G_{sub} . The total number of connected components for the k^{th} subcolony is the total number of maximal subgraphs that partition the k^{th} subcolony graph G_{sub,d_k} .

2.2.3 Statistical Analysis

To analyze the impact of budding and nutrient limited growth on size, shape, and emergent patterns of spatial organization of cells, we generated 50 simulations of each colony type (budding/non-budding, nutrient rich/nutrient limited). To compare expanse, sparsity, connectivity and the number of connected components for each of the first five subcolonies between budding and non-budding colonies we used independent t-tests implemented using the `statannot` package in python [173]. In addition, we performed Kaplan-Meier survival analysis and generated Kaplan-Meier survival curves using the `lifelines` library in python [34]. The survival function defines the probability that a death event (i.e. loss of a mother-daughter edge in G_S or a given subcolony splitting into more than 15 connected components) has not occurred yet at time t , or equivalently, the probability of surviving past time t [49]. We then used the log-rank test available in the `lifelines` library to compare the survival curves be-

tween budding and non-budding colonies in all cases (Figure 2.11 D, Figure 2.12 D, Figure 2.15 D and Figure 2.16 D). Finally, we computed the Kolmogorov-Smirnov statistic using the SciPy library in python [168] to compare the probability distributions of birth location between nutrient rich and nutrient limited colonies.

2.3 Results

To study the impact of budding division on the spatial organization of *S. cerevisiae* colonies, we developed a 2D computational model to compare morphological characteristics and spatial arrangement of cells between budding and non-budding colonies in both nutrient-rich and nutrient-limited conditions (see Section 2.2 for a detailed discussion of our computational model and the metrics we use to evaluate our colonies). Note, by non-budding colonies we mean colonies where mother-bud pairs are not physically attached while the new daughter grows to a mature size. In the model, we represent each cell by an elastic sphere that moves, grows, buds and divides according to biophysical and cell-kinetic model parameters chosen to match experimentally derived values (Section 2.2.1, Table 2.1, Figure 2.2, Figure 2.3 and Figure 2.4). Each simulation begins with a single newly born founder cell. We allow the colony to grow for ≈ 24 hours until there are $\approx 15,000$ cells. We compared our synthetic yeast colonies grown under different conditions (budding/non-budding, nutrient-rich/nutrient-limited) with metrics designed to capture the overall colony growth, shape, and spatial organization of cells (see Section 2.2.2). Results below are based on 50 simulations for each of these four conditions. Typical output from budding and non-budding colonies in nutrient-rich conditions is shown in Figure 2.8 (middle and bottom rows) and typical output from budding and non-budding colonies in nutrient-limited conditions is shown in Figure 2.17 A, B.

As we next discuss in greater detail, we find that metrics corresponding to the overall growth and shape of colonies are not impacted by budding division. However, we observe significant differences in metrics characterizing the local spatial organization and connectivity of budding versus non-budding colonies. Finally, we find that in addition to further impacting local spatial organization and colony connectivity, nutrient limitation changes the global organization of growing yeast colonies.

2.3.1 Nutrient-Rich Growth: Budding Division Impacts Local Colony Organization in Simulated Yeast Colonies

First, we consider colonies growing in “nutrient-rich” conditions. As described in Section 2.2.1, for budding colonies, we explicitly model the mechanical interactions that arise due to the formation of a new daughter cell from budding (Figure 2.2 A). When simulating non-budding colonies, we treat mechanical interactions of mother-bud pairs and all other cell-cell pairs identically (Section 2.2.1 and Figure 2.2 B).

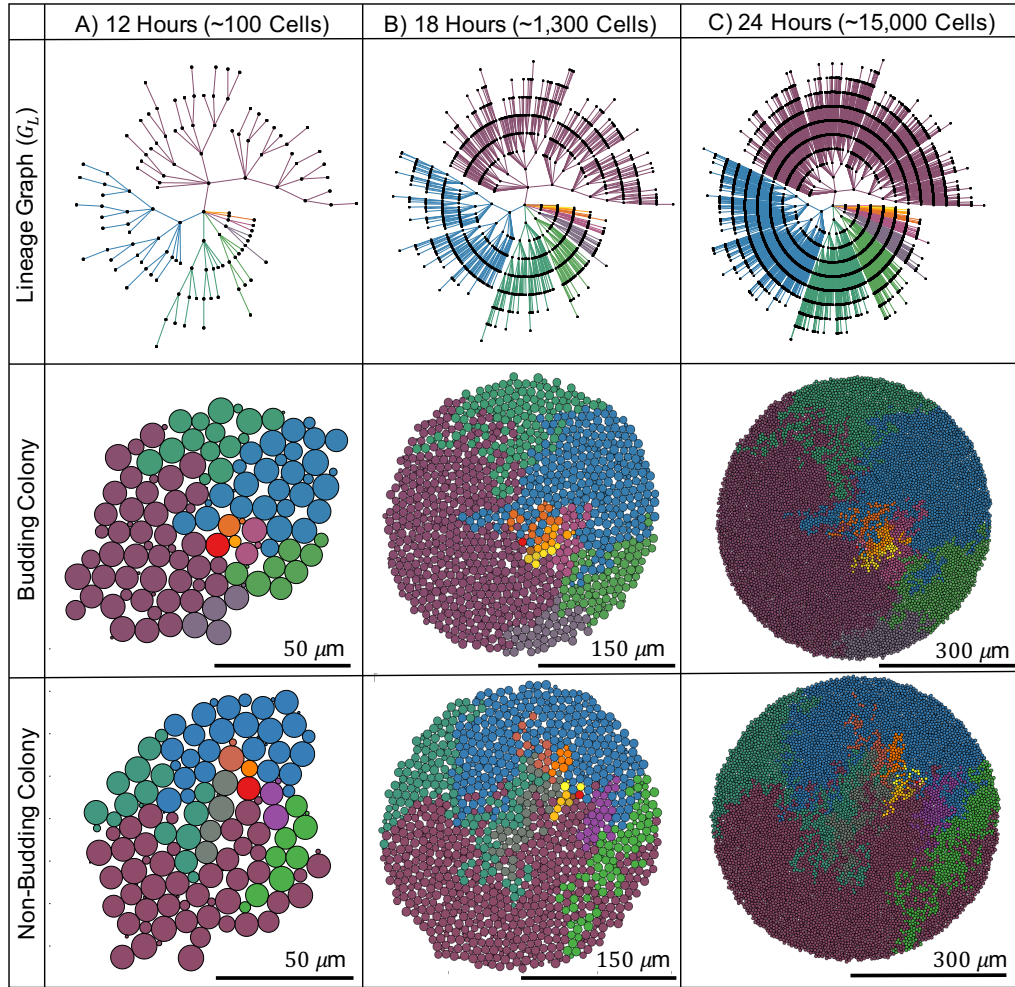


Figure 2.8: **Simulated Yeast Colonies in Nutrient-Rich Conditions and Corresponding Lineage Relationships.** We compared overall growth, shape and spatial organization between budding (Middle) and non-budding (Bottom) colonies. Colonies are depicted at three time points (A) 12 h (~100 cells), (B) 18 h (~1,300 cells) and (C) 24 h (~15,000 cells). In addition to the physical layout of cells, we analyzed two different networks associated with our colonies, the lineage graph (G_L , Top) and the spatial graph (G_S , see Section 2.2.2). (Top row) The lineage graph (G_L) represents mother-daughter relationships and does not consider cell position in space. As such, the lineage graph is the same for the two different colonies depicted below (budding and non-budding). Cells in colonies and edges in the lineage graph are colored according to the unique subcolony each cell belongs to, where a **subcolony** is the subset of all cells whose common ancestor is the same immediate daughter of the founder cell: Founder (red), Subcolony 1 (maroon), Subcolony 2 (blue), Subcolony 3 (dark green), Subcolony 4 (light green), Subcolony 5 (lavender), Subcolony 6 (purple), Subcolony 7 (dark orange), Subcolony 8 (gold), Subcolony 9 (yellow), Subcolony 10 (rust), Subcolony 11 (magenta), Subcolony 12 (light pink) and Subcolony 13 (grey). Lineage graphs display the first five subcolonies only. (See Section 2.2.2 for details.)

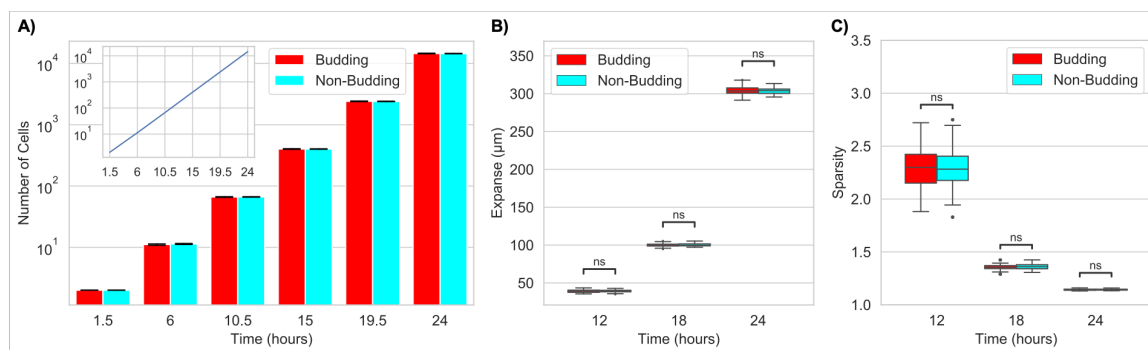


Figure 2.9: Population Growth, Expanse and Sparsity of Simulated Yeast Colonies in Nutrient-Rich Conditions. (A) Colony growth is exponential with doubling time ~ 105 min (inset). Bar plots represent average number of cells in each colony across all 50 simulations for both budding and non-budding colonies under nutrient-rich conditions calculated at 4.5 h intervals. (B) Colony expanse increases over time for both budding and non-budding colonies. (C) Colony sparsity decreases to 1 (implying that the colony is becoming more circular) as the size of the colony increases over time for both budding and non-budding colonies. (See Section 2.3.1 for details.)

Budding Does Not Impact Large-Scale Colony Growth or Structure (Expanse or Sparsity)

In the absence of nutrient limitation, colony growth is exponential with a doubling time of ~ 105 minutes. More specifically, colony growth is exponential with a doubling time of ~ 90 minutes for mother cells and ~ 120 minutes for new daughter cells (Figure 2.9 A and Figure 2.4). We assume no difference in growth rate or cell cycle length between budding and non-budding colonies. Thus, as expected, we see no difference in total population between budding and non-budding colonies (Figure 2.9 A).

Next, we compared the size and shape of budding and non-budding colonies under nutrient-rich conditions by calculating the expanse and sparsity of each colony (Figure 2.9 B, C). The expanse quantifies how large the colony is with respect to the average distance of each cell to the colony center of mass, while colony sparsity is a measure that quantifies the roundness of each colony (Section 2.2.2). As the colony grows, the colony expanse increases and the colony sparsity decreases towards 1, implying that the colony is becoming more circular as the number of cells in the colony increases. We observe no difference between colony expanse or colony sparsity between budding and non-budding colonies (Figure 2.9 B, C). Thus, budding does not impact the appearance of the colony when viewed as a whole.

Budding Does Not Change Global Age and Spatial Structure but Impacts Local Connectivity

Next we analyzed the relationship between cell age and spatial location within the colony after 24 hours of growth. First we computed the empirical probability density function for the ages of cells in the colony (Figure 2.10 A). We find that over 93% of cells are less than 8 hours old, and thus we focus on this subpopulation. (Note, because cell cycle timing is unchanged between budding and non-budding colonies, as expected we find no difference in the distribution of cell ages between these two conditions). Next, we quantified the birth location of cells within the colony (Figure 2.10 B). To do this, we analyzed the empirical probability density function for the normalized distance from the colony center of mass to the birth location of cells born in the last hour of colony growth. We observe that in both conditions, cells are more likely to be born closer to the perimeter of the colony. We note that this empirical distribution has a nearly linear increase, consistent with a uniform probability per area of the colony, with the exception of the decreased probability at the moving front of the colony. Similarly, we find no difference in birth location of cells between budding and non-budding colonies demonstrating that the budding mechanism has no impact on birth locations within the colony. Finally, to ensure that this distribution itself is not age structured, we examined the distance from the colony center of mass (Figure 2.10 C) and note that both budding and non-budding cells display uniform probabilities with respect to distance from the colony center of mass for all ages.

While there are no differences in the large-scale age structure between budding and non-budding colonies, we note that there are significant differences within the local neighborhood of a cell. More specifically, we observed a small, but statistically significant (non-overlapping 95% confidence intervals) difference in the distance a cell less than 8 hours old was from its mother (Figure 2.10 D). Although this difference is quite small (less than the average radius of a cell), because most cells in the colony fall into this age group, this suggests significant differences in the local spatial arrangement and organization of budding versus non-budding colonies. Note, for both average distance from the colony center of mass (Figure 2.10 C) and average distance from cells to their mother (Figure 2.10 D), average values were determined for different age groups using a sliding window with a fixed window size of 20 minutes. In addition, 95% confidence intervals are shown for each window.

Budding Division Maintains Closeness between Mothers and Daughters after Physical Separation

To better quantify the differences in local spatial arrangement and organization between budding and non-budding colonies, we analyzed the spatial graph (G_S) and lineage graph (G_L) for each colony. The spatial graph has an edge between cells that are adjacent according to the Delaunay triangulation of their centers (Figure 2.6 A). The lineage graph has an edge between mother and daughter cell pairs regardless of their spatial position (Figure 2.8 (top row) and Figure 2.6 B). We construct both

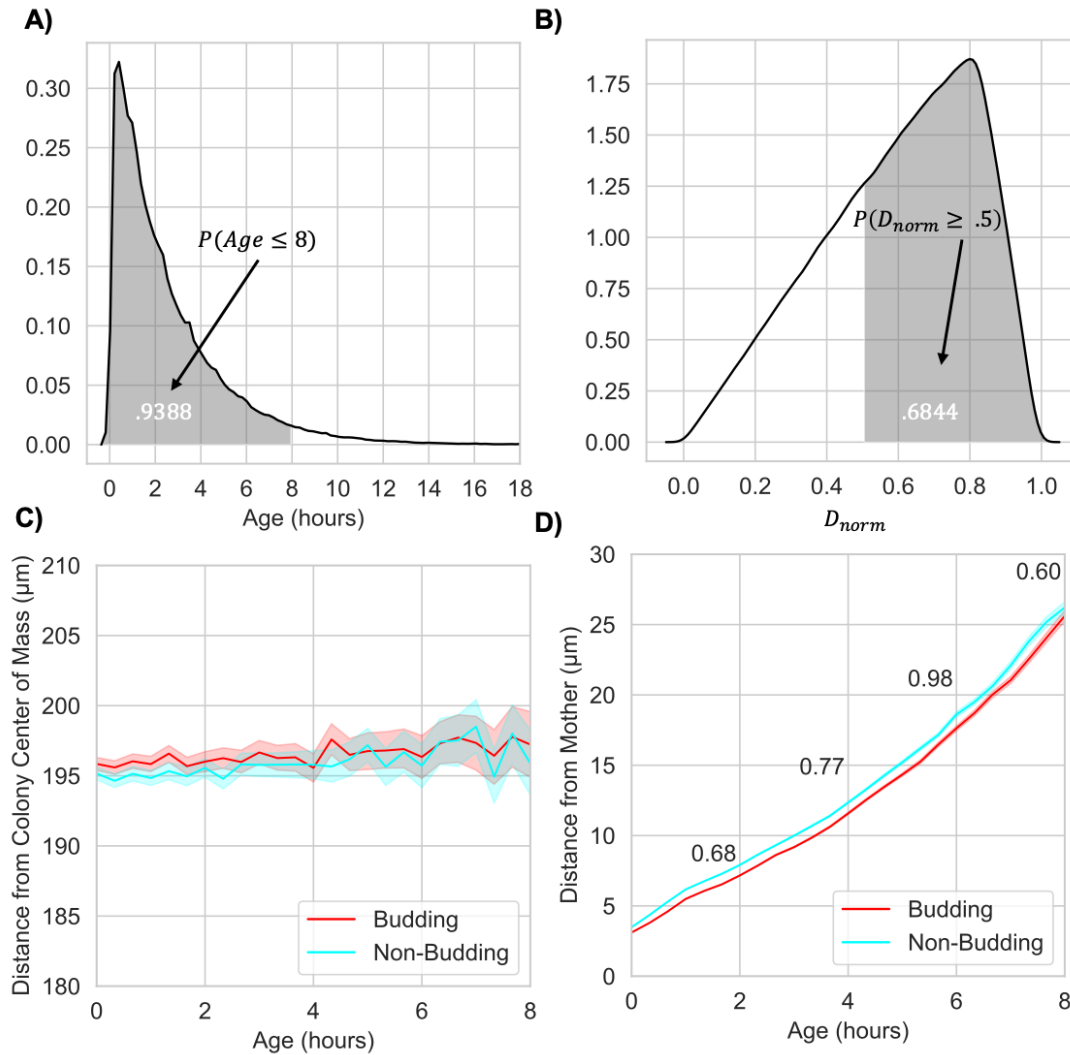


Figure 2.10: **Age and Spatial Organization of Cells in Nutrient-Rich Colonies.** (A) Empirical probability density function for cell ages in synthetic colonies after 24 h of colony growth. We observe that the probability that a given cell is ≤ 8 h old is .9388. (B) Empirical probability density function for the normalized distance from the colony center of mass to the birth location of cells born in the last hour of colony growth (D_{norm}). We observe that the probability that D_{norm} is $\geq .5$ is .6844. (C) Cell age (h) versus cell distance from the colony center of mass for budding (red) and non-budding (blue) colonies at the 24 h time point. In both (C) and (D), average values were determined for different age groups using a sliding window with a fixed window size of 20 min. We conclude that cell distance from the colony center of mass is not impacted by cell age or budding division since 95% confidence intervals overlap. (D) Cell age (h) versus cell distance from its mother for budding (red) and non-budding (blue) colonies. The difference between budding and non-budding colonies is given for age groups 2,4,6,8. We conclude that this difference is significant since 95% confidence interval do not overlap. (See Section 2.3.1 for details).

graphs from our simulation output following the procedures defined in Section 2.2.2. Together, these graphs provide a framework to quantify how spatial relationships between mother-daughter cell pairs dynamically evolve during colony growth.

Colony connectivity is a measure of how many mother-daughter cell pairs are adjacent in the colony. We determine this by looking at the fraction of mother-daughter edges in G_S (See Section 2.2.2 for more details.) First, we compared colony connectivity between budding and non-budding colonies after 12, 18 and 24 hours of growth (Figure 2.11 A). While colony connectivity decreases over time for both budding and non-budding colonies, it is significantly higher in budding colonies starting after 18 hours of growth. We hypothesized that since cells in budding colonies remain physically attached to their mother until separation, high connectivity of the cells still attached to their mother could explain the observed difference. To investigate this hypothesis, we compared the connectivity between mother-daughter cell pairs in two distinct phases: when they are attached during budding cell division (“budded cells”, Figure 2.11 B) and after separation (“un-budded cells”, Figure 2.11 C). We observe that the connectivity of budding colonies is significantly higher in each phase (Figure 2.11 B, C). Namely, in the case of budded cells, connectivity stays close to 1 during the entire 24 hour time period for budding colonies whereas it decreases between 12-18 hours and then again between 18-24 hours for non-budding colonies (Figure 2.11 B). Surprisingly, in the case of un-budded cells, the difference in connectivity is significantly different between budding and non-budding colonies starting at 18 hours (Figure 2.11 C). In fact, the absolute difference in mean connectivity between budding and non-budding colonies remains unchanged from the overall colony connectivity computed for all cells.

To further address the mechanism driving our observed differences in colony connectivity between budding and non-budding colonies, we hypothesized that mother-daughter cell pairs remain physically close for a longer period of time after separation in budding colonies. To investigate this hypothesis, we generated Kaplan-Meier survival curves for the lifetime of mother-daughter edges in G_S (lifetime being the length of time after separation). We observe that the probability that a given mother-daughter edge will remain in G_S for longer than t hours after separation is higher for budding colonies (Figure 2.11 D). In addition, the restricted mean survival time for a given mother-daughter edge in G_S is 80 minutes compared to 66 minutes for non-budding colonies. Furthermore, we observe distinctly different behavior between the two survival curves from 0-50 minutes after separation, whereas after 50 minutes, the two curves display more similar behavior. This change in behavior is marked by a “sharp” decrease in survival probability for both types of colonies. The timing of this sharp decrease is consistent with the appearance of a second bud from the mother cell and we conjecture that the new bud “pushes away” the previous daughter.

Budding Division Promotes Subcolony Connectivity

Next, we considered the impact of budding division on subcolony structure and organization. We define a **subcolony** to be the subset of all cells whose common

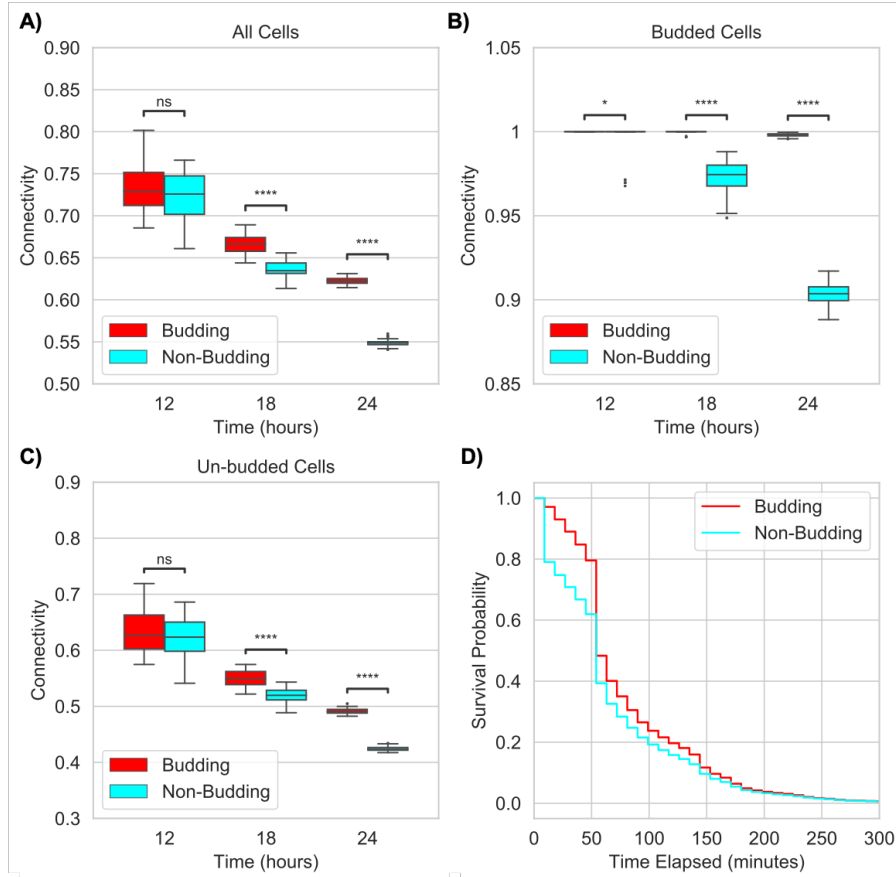


Figure 2.11: Colony Connectivity in Nutrient-Rich Colonies. Comparison of colony connectivity between budding (red) and non-budding (blue) colonies at 12, 18 and 24 h. We compared connectivity for all cells in (A), as well as cells in two distinct phases: during the time when cells are attached for budding colonies in (B) and after separation in (C). (See Section 2.2.2 for how we define connectivity.) (A) We observe a statistically significant difference in colony connectivity at 18 h ($p = 1.423e-27$) and 24 h ($p = 2.008e-99$). (B) We observe a rapid decrease in connectivity between both 12-18 and 18-24 h in non-budding colonies. This decrease leads to a statistically significant difference between budding and non-budding colonies at 12 h ($p = 3.387e-02$), 18 h ($p = 6.521e-34$) and 24 h ($p = 2.867e-103$). (C) We observe a statistically significant difference in connectivity at 18 h ($p = 3.684e-19$) and 24 h ($p = 7.615e-89$). (p -values for connectivity were computed using independent t-tests.) (D) Kaplan-Meier survival curves for the edge connecting mother-daughter cell pairs in G_S for budding (red) and non-budding (blue) colonies. The y-axis is the probability that a given mother-daughter edge will remain in G_S for longer than t hours after separation, where time is on the x-axis. We observe that the survival curves are different between the two groups ($p = 7.217e-25$), indicating that the probability that a mother-daughter edge remains in G_S for longer than t hours is greater for budding colonies. (The p -value comparing survival curves was calculated using a log-rank test as described in Section 2.2.3.) (See Section 2.3.1 for details.)

ancestor is the same immediate daughter of the founder cell (Section 2.2.2 and Figure 2.7). Note that a colony has as many subcolonies as immediate daughters of the founding cell. We then analyze how well each of these subcolonies is connected in terms of the spatial layout of the colony. To do this, we considered colonies at the final time point (24 hours) and compared the number of connected components of the first five subcolonies between budding and non-budding division conditions. We found that the average number of connected components was significantly lower for budding colonies (Figure 2.12 A). This demonstrates that budding division acts as a mechanism to increase spatial adjacency within subcolonies as well as impact overall subcolony connectivity.

Because every subcolony begins with a single cell, which is necessarily a single connected component, we studied at what time a given subcolony would break apart. To do this, we computed the average time elapsed from creation of a given subcolony until it splits into more than 15 connected components. We observe that the average time was significantly lower in non-budding colonies for subcolonies 1, 2 and 3 (Figure 2.12 B). These results suggest budding division impacts subcolony connectivity by ensuring cells in the same subcolony remain physically closer together for a longer period of time. Moreover, we hypothesized that the absence of budding division makes it easier for individual cells to become separated from the rest of their subcolony. To test this hypothesis, we computed the average number of cells in small connected components (i.e. less than 10 cells) for budding and non-budding colonies. We find that the average number of cells in a small connected component was significantly higher in non-budding colonies (Figure 2.12 C).

In addition, we generated Kaplan-Meier survival curves for the time elapsed from creation of a subcolony to when it splits into more than 15 components. We considered this a separate event for each of the first 5 subcolonies and each of our 50 simulations. We found that the survival curves are different between the two groups (Figure 2.12 D) indicating that the probability that a given subcolony remains connected (i.e. less than 15 connected components) for longer than t hours is greater for budding colonies.

2.3.2 Nutrient-Limited Growth: Differential Growth Rates Impact Global Organization of Yeast Colonies

Prior studies have shown that nutrient limitation impacts patterns of growth and spatial organization in microbial colonies [57, 109, 112, 169]. To investigate the role of nutrient limitation and budding division together on patterns of growth and organization in our simulated yeast colonies, we revised our ABM to include nutrient-limited growth (Section 2.2.1) and used the same set of metrics as before to compare overall shape, size and spatial organization of cells. Rather than directly model the concentration of a nutrient in time as has been done previously [44–47, 72, 109, 112, 169, 170], we consider regions of the growth media to have a maximal possible biomass (i.e. carrying capacity) and decrease the cell growth progression of cells in each region accordingly. In particular, we provide a simplified model of nutrient dynamics that

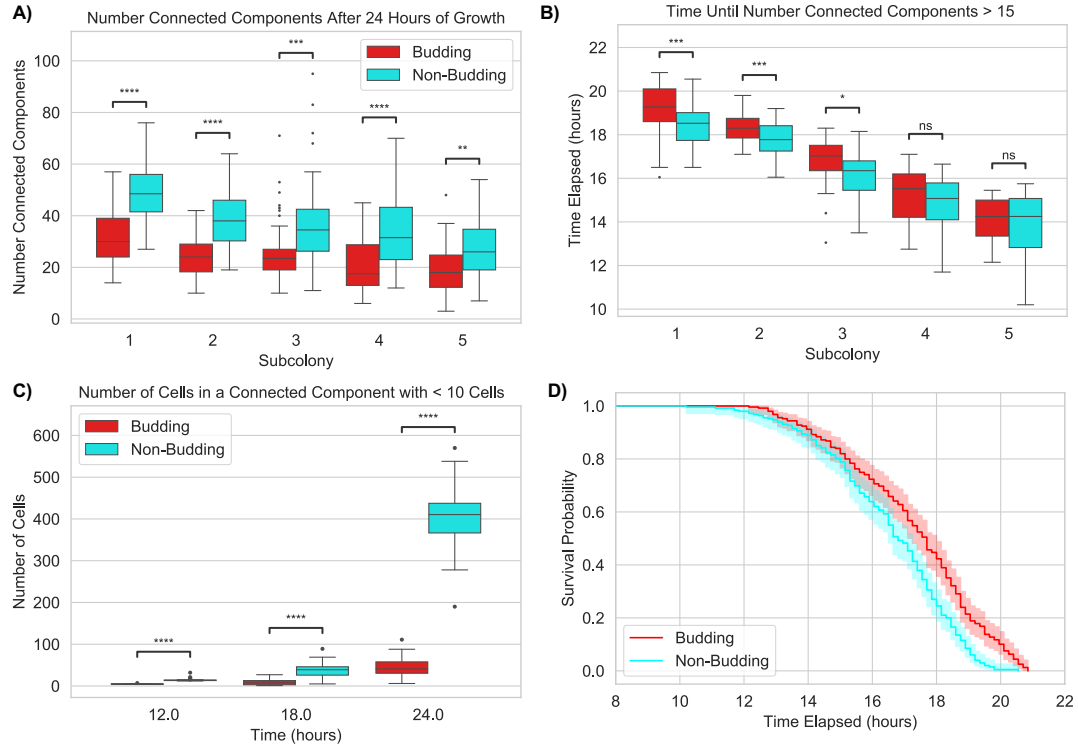


Figure 2.12: Subcolony Structure and Organization in Nutrient-Rich Colonies. (A) Comparison of the number of connected components for the first five subcolonies between budding (red) and non-budding (blue) colonies at the 24 h time point. We observe a statistically significant difference in the number of connected components for subcolony 1 ($p = 2.885e-10$), subcolony 2 ($p = 1.139e-10$), subcolony 3 ($p = 3.075e-04$), subcolony 4 ($p = 1.539e-06$) and subcolony 5 ($p = 1.959e-03$). (B) Comparison of time from creation until each of the first five subcolonies splits into 15 connected components. We observe a statistically significant difference between budding (red) and non-budding (blue) colonies for subcolony 1 ($p = 1.140e-04$), subcolony 2 ($p = 4.727e-04$), and subcolony 3 ($p = 1.581e-02$). (C) Comparison of the number of cells in a connected component with less than 10 cells. Note that overall very few cells are in small connected components; however, we observe a statistically significant difference between budding (red) and non-budding (blue) colonies at 12 h ($p = 7.408e-40$), 18 h ($p = 7.896e-19$) and 24 h ($p = 1.233e-59$). (p -values in (A), (B) and (C) were computed using independent t-tests). (D) Kaplan-Meier survival curves for the length of time a subcolony is made up of less than 15 connected components for budding (red) and non-budding (blue) colonies. The y-axis is the probability that a subcolony consists of less than 15 connected components for longer than t hours, where time is on the x-axis. We observe that the survival curves are different between the budding and non-budding colonies ($p = 1.165e-06$), indicating that budding promotes subcolony connectivity. (The p -value comparing survival curves was calculated using a log-rank test as described in Section 2.2.3. In addition, 95% confidence intervals for the survival function are shown. See Section 2.3.1 for details.)

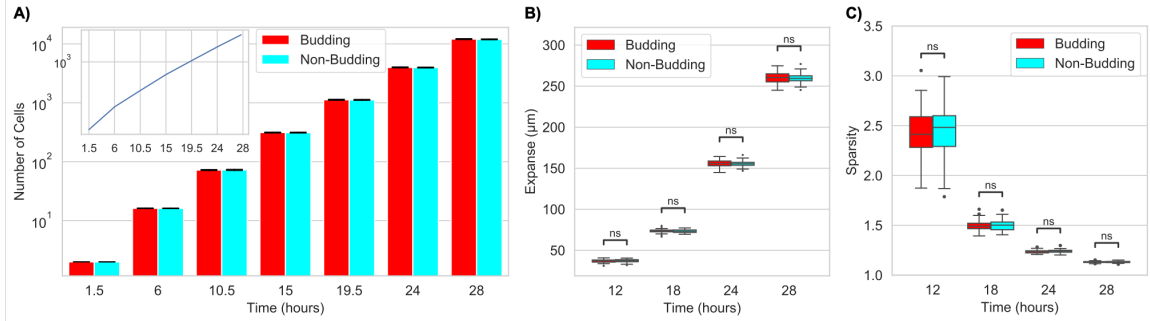


Figure 2.13: Population Growth, Expanse and Sparsity of Simulated Yeast Colonies in Nutrient-Limited Conditions. (A) Colony growth is exponential with doubling time ~ 123 min (inset). Bar plots represent average number of cells in each colony across all 50 simulations for both budding and non-budding colonies under nutrient-limited conditions calculated at 4.5 h intervals. (B) Colony expanse increases over time for both budding and non-budding colonies. (C) Colony sparsity decreases to 1 as the size of the colony increases over time for both budding and non-budding colonies. (See Section 2.3.2 for details.)

only considers the indirect effect of nutrients on cell cycle length and does not directly model nutrient concentration or a particular type of nutrient.

Nutrient Limitation Slows Colony Growth but Does Not Change Large-Scale Colony Structure

As in our nutrient rich condition, large-scale behavior (doubling time, sparsity and expanse) between budding and non-budding colonies is the same. (Figure 2.13 A, B and C). However, when nutrients are limited, our synthetic yeast colonies grow slower. Since colony connectivity and structure of subcolonies significantly changes between 1,000 and 10,000 cells, we computed each of our metrics for nutrient-limited colonies at an additional time point (28 hours) where the number of cells in the colonies is over 10,000 and more similar to the nutrient-rich case ($\sim 12,500$ cells after 28 hours of growth in nutrient-limited conditions compared to $\sim 15,000$ cells after 24 hours of growth in nutrient-rich conditions). The average doubling time of a colony increases from ~ 105 minutes under nutrient-rich conditions to ~ 123 in nutrient-limited conditions (Figure 2.13 A). The colony expanse at 24 hours decreases from $300 \mu\text{m}$ in nutrient-rich conditions to $150 \mu\text{m}$ in nutrient-limited conditions (Figure 2.13 B). Similarly to nutrient-rich growth, colony sparsity decreases toward 1, indicating the colony becomes more circular as it grows (Figure 2.13 C). Finally, as we assumed no difference in growth rate or cell cycle length between budding and non-budding division, these gross colony level metrics remain unchanged when comparing between division conditions.

Nutrient Limitation Creates Age-Structured Colonies by Promoting Birth at the Colony Boundary

First, as in the nutrient-rich condition, the asymmetric cell cycle means that the age distributions are shifted toward younger cells in nutrient-limited colonies. Because we grow nutrient-limited colonies for a longer period (28 hours vs. 24 hours for nutrient-rich colonies), we consider “young cells” as those less than 12 hours old (over 94% of cells in the colony). Similarly to the nutrient-rich condition, we found no difference in cell ages between budding and non-budding colonies (Figure 2.14 A). Next, and in contrast to nutrient-rich conditions, when the cell cycle is tied to locally available nutrients the age structure of the colony changes substantially. As shown in Figure 2.14 B, cells are far more likely to be born closer to the edge of the colony than in nutrient rich colonies. When we observe the age structure in greater detail, we observe that the distance from the colony center of mass at 28 hours is strongly correlated with age (Figure 2.14 C). However, this relationship between age-structure and distance is unaffected by budding division. As such, we conclude that within a given growth condition, budding division does not modify the age-structure within a colony.

Similarly to nutrient-rich growth, we observe that budding division influences the local neighborhood of a cell. More specifically, we observe a small, but statistically significant (non-overlapping 95% confidence intervals) difference, in the distance a cell less than 12 hours old is from its mother (Figure 2.14 D). While this difference remains small, it is interesting that the absolute difference between budding and non-budding mother-daughter cell pairs is larger for nutrient-limited colonies and may even be increasing with the age of the daughter cell. Because colonies grown in nutrient limited conditions are smaller than nutrient-rich colonies, this difference becomes even larger when it is considered relative to the size of the colony.

Nutrient-Limited Growth Promotes Colony Connectivity

As for nutrient-rich growth conditions, the colony connectivity decreases in time and is significantly higher in budding colonies (compare Figure 2.11 A with Figure 2.15 A). However, we observe that nutrient-limited growth promotes colony connectivity as both budding and non-budding colonies have higher connectivity in this growth condition at the 24 hour time point. In addition, the difference in colony connectivity is observable much earlier on in the life of the colony (12 hours versus 18 hours).

As above, we analyzed connectivity between mother and daughter pairs in two distinct phases: when they are attached for budding cell division (budded cells, Figure 2.15 B) and after separation (un-budded cells, Figure 2.15 C). We observe that, as in nutrient-rich growth, the connectivity between mother and daughter cells in both phases was higher for budding cell division, and that this difference in connectivity between division types increases in time. However, we note that the connectivity for un-budded cells was slightly higher at the 24 hour time point in the nutrient rich

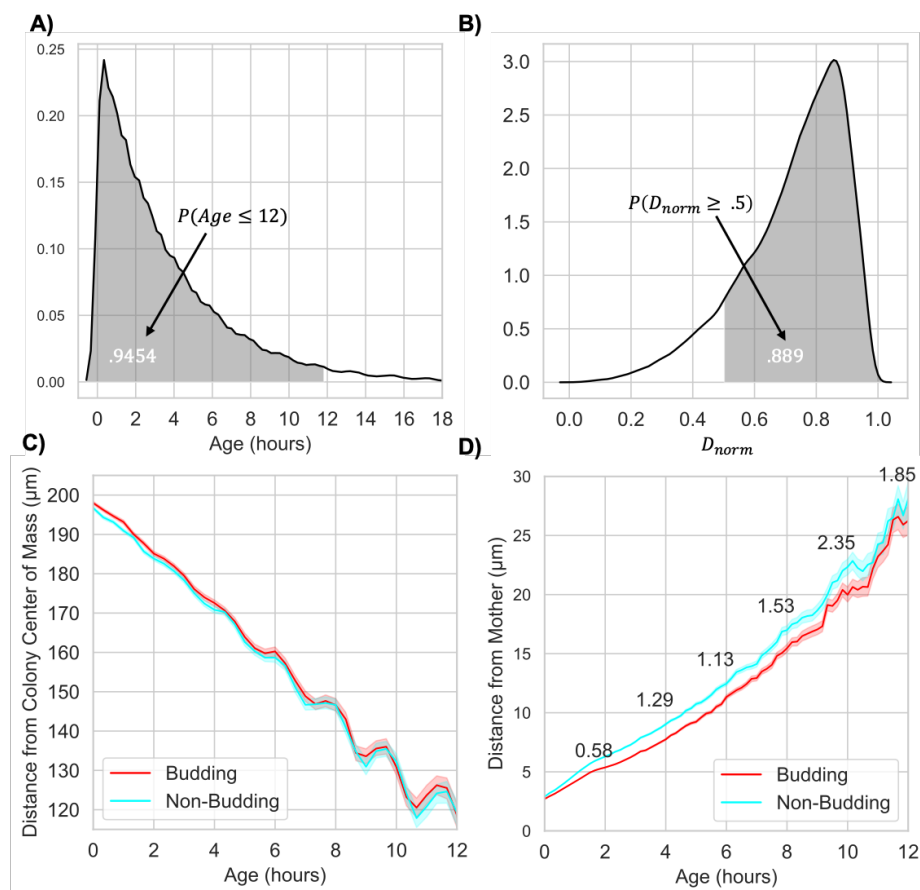


Figure 2.14: Age and Spatial Organization of Cells in Nutrient-Limited Colonies. (A) Empirical probability density function for cell ages in synthetic colonies after 28 h of colony growth. We observe that the probability that a given cell is ≤ 12 h old is .9454. (B) Empirical probability density function for the normalized distance from the colony center of mass to the birth location of cells born in the last hour of colony growth (D_{norm}). We observe that the probability that D_{norm} is $\geq .5$ is .889. (C) Cell age (h) versus cell distance from the colony center of mass for budding (red) and non-budding (blue) colonies at the 28 h time point. In both (C) and (D) average values were determined for different age groups using a sliding window with a fixed window size of 20 min. We conclude that distance from the colony center of mass at 28 h is strongly correlated with age. However, we see that cell distance from the colony center of mass is not impacted by budding division since 95% confidence intervals overlap. (D) Cell age (h) versus cell distance from its mother for budding (red) and non-budding (blue) colonies. The difference between budding and non-budding colonies is given for ages 2,4,6,8,10 and 12. We conclude that this difference is significant since 95% confidence interval do not overlap. Moreover, since nutrient-limited colonies are smaller than nutrient-rich colonies, this difference becomes even larger when it is considered relative to the size of the colony. (See Section 2.3.2 for details.)

growth condition (compare Figure 2.11 C with Figure 2.15 C).

We believe the explanation for both the overall higher colony connectivity in nutrient-limited growth (panel A) and the decreased connectivity for un-budded cells (panel C) is due to the impact of nutrient availability on the cell cycle and position of newly born cells. More specifically, the extended cell cycle induced by nutrient limitation (i.e cells stay in the budding phase longer) creates an overall higher colony connectivity because daughter cells stay attached longer. In addition, the age structure of the colony, where newly born cells are more likely to be at the colony perimeter, means that these newly born cells have more space to move away from their mother when they detach.

To better understand the impact of nutrient limitation and budding on the duration of the mother-daughter edges in G_S , we generated Kaplan-Meier survival curves (Figure 2.15 D). We observed that the probability the mother-daughter edge stays in G_S for longer than t hours after separation is higher in budding colonies. In addition, the restricted mean survival time for mother-daughter edges in budding colonies was 67 minutes compared to 52 minutes for non-budding colonies. However, as expected by our explanation for Figure 2.15 B and C, we observe the median duration of a mother-daughter edge in G_S is shorter when compared to the nutrient-rich condition.

Nutrient Limitation Further Promotes Subcolony Connectivity

Next we investigated the impact of nutrient limitation together with budding division on subcolony organization and connectivity. To do this, we first analyzed the number of connected components between budding and non-budding colonies in nutrient-limited conditions. We find that nutrient limited growth results in a significant decrease in the number of connected components for both budding and non-budding colonies compared to nutrient rich growth. (Compare Figure 2.12 A to Figure 2.16 A.) However, we also observe that the number of connected components for each of the first five subcolonies is significantly lower in budding colonies (Figure 2.16 A). In addition, we see that the average time elapsed from creation of a given subcolony until it splits into more than 5 connected components is significantly lower in non-budding colonies (Figure 2.16 B). These results not only reveal that nutrient limitation has a large impact on subcolony structure and connectivity, but also confirm that budding division maintains its role in promoting subcolony connectivity under nutrient limitation.

To further investigate the impact of nutrient limitation and budding division together on subcolony structure and connectivity we considered the number of cells contained in small connected components (i.e. less than 10 cells). We find that nutrient limitation results in a large decrease in the total number of cells belonging to a small connected component (i.e. less than 10 cells) for both budding and non-budding colonies at the 24 hour time point (compare Figure 2.12 C with Figure 2.16 C). This further highlights the strong impact of nutrient limitation on subcolony structure. In addition, we observe a significant difference in the number of cells in a small connected component between budding and non-budding colonies at the 18, 24, and 28

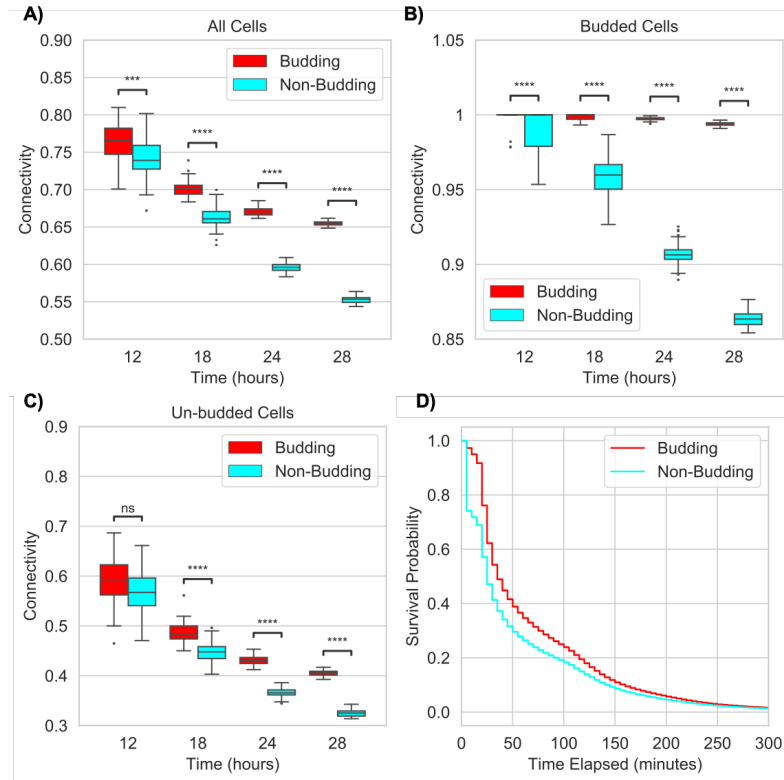


Figure 2.15: Colony Connectivity in Nutrient-Limited Colonies. Comparison of colony connectivity between budding and non-budding colonies in nutrient-limited conditions at 12 h, 18 h, 24 h and 28 h. We compared connectivity for all cells in (A), as well as cells in two distinct phases: during the time when cells are attached for budding colonies in (B) and after separation in (C). (See Section 2.2.2 for how we define connectivity.) (A) We observe a statistically significant difference in colony connectivity at 12 h ($p = 3.602e-04$), 18 h ($p = 8.921e-29$), 24 h ($p = 1.094e-79$) and 28 h ($p = 3.389e-110$). (B) We observe a rapid decrease in connectivity for non-budding colonies between 12-18, 18-24, and 24-28 h. This decrease leads to a statistically significant difference between budding and non-budding colonies at 12 h ($p = 6.834e-05$), 18 h ($p = 4.670e-40$), 24 h ($p = 1.050e-92$) and 28 h ($p = 2.658e-122$). (C) We observe a statistically significant difference in connectivity at 18 h ($p = 1.144e-15$), 24 h ($p = 1.888e-58$) and 28 h ($p = 2.140e-83$). (p -values for connectivity were computed using independent t-tests.) (D) Kaplan-Meier survival curves for the edge connecting mother-daughter cell pairs in G_S for budding (red) and non-budding (blue) colonies. The y-axis is the probability that a given mother-daughter edge will remain in G_S for longer than t hours after separation, where time is on the x-axis. We observe that the survival curves are different between the two groups ($p = 1.314e-75$) indicating that the probability that a mother-daughter edge remains in the spatial graph for longer than t hours is greater for budding colonies. (The p -value comparing survival curves was calculated using a log-rank test as described in Section 2.2.3. See Section 2.3.2 for details.)

hour time points. Moreover, we find that the survival curves for the time elapsed from creation of a subcolony to when it splits into more than 5 components are different between budding and non-budding colonies (Figure 2.12 D), indicating that the probability that a given subcolony remains connected (i.e. less than 5 connected components) for longer than t hours is greater for budding colonies. Similarly as before, we considered this a separate event for each of the first 5 subcolonies and each of our 50 simulations. These results further support our observation that budding division acts as a mechanism increasing spatial adjacency within subcolonies in both nutrient-rich and nutrient-limited conditions.

Nutrient Limitation Changes Global Colony Organization by Driving Variation in Subcolony Sizes

Results from our previous metrics provide compelling evidence that nutrient limitation has a large impact on spatial organization of growing yeast colonies. However, the strongest demonstration of the impact of nutrient-limited growth on morphological properties of growing yeast colonies is its effect on emergent patterns of subcolony structure and organization for which we do not yet have an explicit quantitative metric. Specifically, simulations of nutrient-limited colonies results in the appearance of subcolonies that grow in a “sector-like” formation (compare Figure 2.17 (top and middle) with Figure 2.8 (middle and bottom)). Namely, subcolony boundaries in nutrient-limited simulations are more linear. This is especially visible in the last time point at 28 hours.

In addition, unlike colonies grown in nutrient-rich conditions, colonies grown in nutrient-limited conditions have highly variable numbers of cells as a percentage of the colony. For example, consider the different sizes of the first subcolony (dark green) in both the top and middle row of Figure 2.17. At 28 hours, the first subcolony in the top row appears to consist of close to half of the total population. At the same time point, the first subcolony in the middle row makes up only a third of the total population. Moreover, in the top row, the first (dark green) and second (dark blue) subcolonies are noticeably different sizes, while in the middle row they are almost exactly the same size. Finally, we note that the colony in the top row (budding) has 13 subcolonies whereas the colony in the middle row (non-budding) has only 7. This is due to variation in the number of daughters produced by the founder cell. Interestingly, we found the variation observed in subcolony sizes did not differ between budding and non-budding conditions but was a property purely driven by nutrient limitation.

To investigate the variation in subcolony sizes between nutrient-rich and nutrient-limited growth, we computed the final percentage of the entire colony contained in each subcolony for nutrient-rich (24 hours) and nutrient-limited (28 hours) growth. As shown in Figure 2.17 C, the nutrient limited colonies have significantly higher variation. We conjecture that this difference in subcolony final composition is due to biases induced by global changes in age and birth structure. To test this conjecture we compared the empirical probability density functions for normalized birth loca-

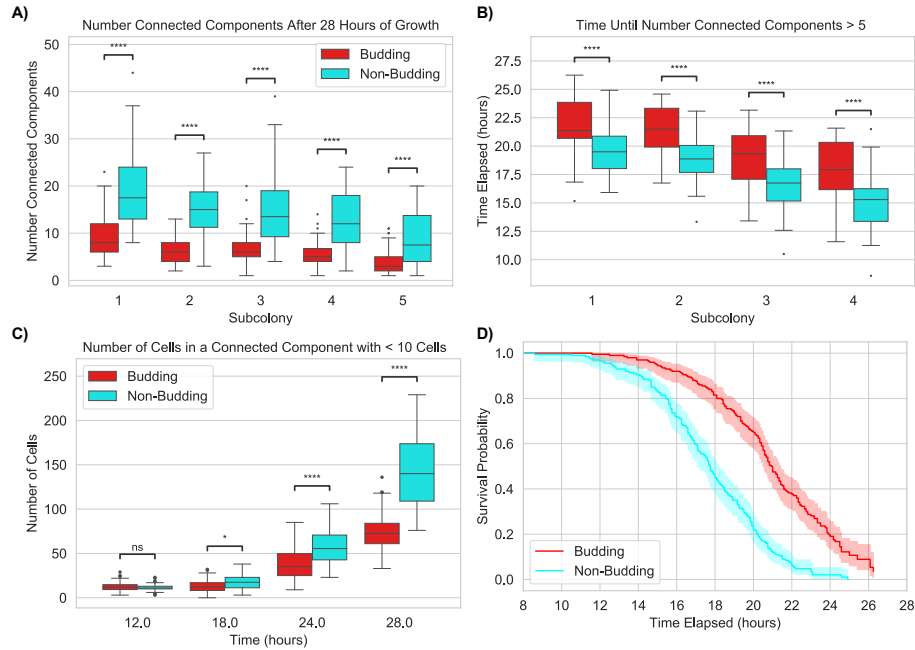


Figure 2.16: Subcolony Structure and Organization in Nutrient-Limited Colonies. (A) Comparison of the number of connected components for the first five subcolonies between budding (red) and non-budding (blue) colonies at the 28 h time point. We observe a statistically significant difference in the number of connected components for subcolony 1 ($p = 3.39e-10$), subcolony 2 ($p = 4.454e-17$), subcolony 3 ($p = 8.537e-11$), subcolony 4 ($p = 2.884e-11$) and subcolony 5 ($p = 3.783e-06$). (B) Comparison of time until each of the first five subcolonies splits into 5 connected components. We observe a statistically significant difference between budding (red) and non-budding (blue) colonies for subcolony 1 ($p = 8.967e-05$), subcolony 2 ($p = 6.246e-08$), subcolony 3 ($p = 8.636e-05$), and subcolony 4 ($p = 9.154e-06$). (C) Comparison of the number of cells in a connected component with less than 10 cells. Note that nutrient-limited growth results in a large decrease in the total number of cells in small components at the 24 h time point compared to nutrient-rich growth. However, we still observe a statistically significant difference between budding (red) and non-budding (blue) colonies at 18 h ($p = 3.537e-02$), 24 h ($p = 9.040e-07$), and 28 h ($p = 5.410e-18$). (D) Kaplan-Meier survival curves for the length of time a subcolony is made up of less than 5 connected components for budding (red) and non-budding (blue) colonies. The y-axis is the probability that a subcolony consists of less than 5 connected components for longer than t hours, where time is on the x-axis. We observed that the survival curves are different between the budding and non-budding colonies ($p = 2.428e-23$) indicating that the probability that a given subcolony remains connected (i.e. less than 5 connected components) for longer than t hours is greater for budding colonies. (The p -value comparing survival curves was calculated using a log-rank test as described in Section 2.2.3. In addition, 95% confidence intervals for the survival function are shown. See Section 2.3.2 for details.)

tion of cells born in the last hour of growth between nutrient-rich and nutrient-limited colonies (Figure 2.17 D). We found that the two empirical distributions are significantly different, indicating that cells in nutrient-limited colonies are more likely to be born at the edge of the colony. These results support our observation that budding division impacts local organization of growing yeast colonies while nutrient-limited growth changes global patterns of colony organization. To our knowledge, this difference in final subcolony variation in nutrient-limited growth has not been explicitly explored for budding division or yeast. Moreover, we note that since this difference is due to nutrient limited conditions, this variation would not hold for yeast colonies grown in liquid culture.

2.4 Discussion

As described in the introduction, morphological patterns in microbial colonies arise due to processes at different scales (Figure 2.1). In this Chapter, we developed a 2D ABM and used it to quantify the biophysical impact of budding cell division on the overall shape, size and spatial organization of growing yeast colonies. The novelty of our approach is that we explicitly model the mechanical interactions that arise due to budding cell division. An additional novelty lies in the metrics we developed to quantify spatial organization of cells within the colony. Moreover, results from these metrics reveal the impact of budding division and nutrient limitation on patterns of displacement and spatial rearrangement of cells leading to emergent local and global morphological properties of growing yeast colonies.

In Section 2.3.1, our findings reveal that budding division does not impact global, large-scale structures of growing yeast colonies (Figure 2.9 B, C), or the birth location of cells within the colony (Figure 2.10 C, D). However, we find that budding division substantially impacts local organization of cells including enforcing a smaller physical distance between mother and daughter cells even after separation. We believe this physical closeness is the consequence of two forces. First, the budding division process means mother and daughter cells are forced to be connected for a longer period of time than in non-budding colonies. Second, this prolonged connection means that when the mother and daughter cells do separate, the local environment is more likely to be “crowded” and thus mother-daughter cell pairs are more likely to remain close. This overall closeness between mother and daughter cells results in greater connectivity in the colony in terms of a larger proportion of mother-daughter edges in G_S (i.e., the colony connectivity metric, Figure 2.11) as well as subcolonies that consist of smaller numbers of connected components.

In Section 2.3.2, our findings reveal that nutrient limitation plays a significant role in directing global, large-scale colony organization. Our simple model of nutrient limitation produced colonies that grew more slowly (average doubling time of ~ 123 minutes compared to ~ 105 minutes for nutrient-rich colonies). (Figure 2.17 D) In addition, cells in nutrient-limited colonies were much more likely to be born near the edge of the colony, where nutrients are more readily available, than near the middle,

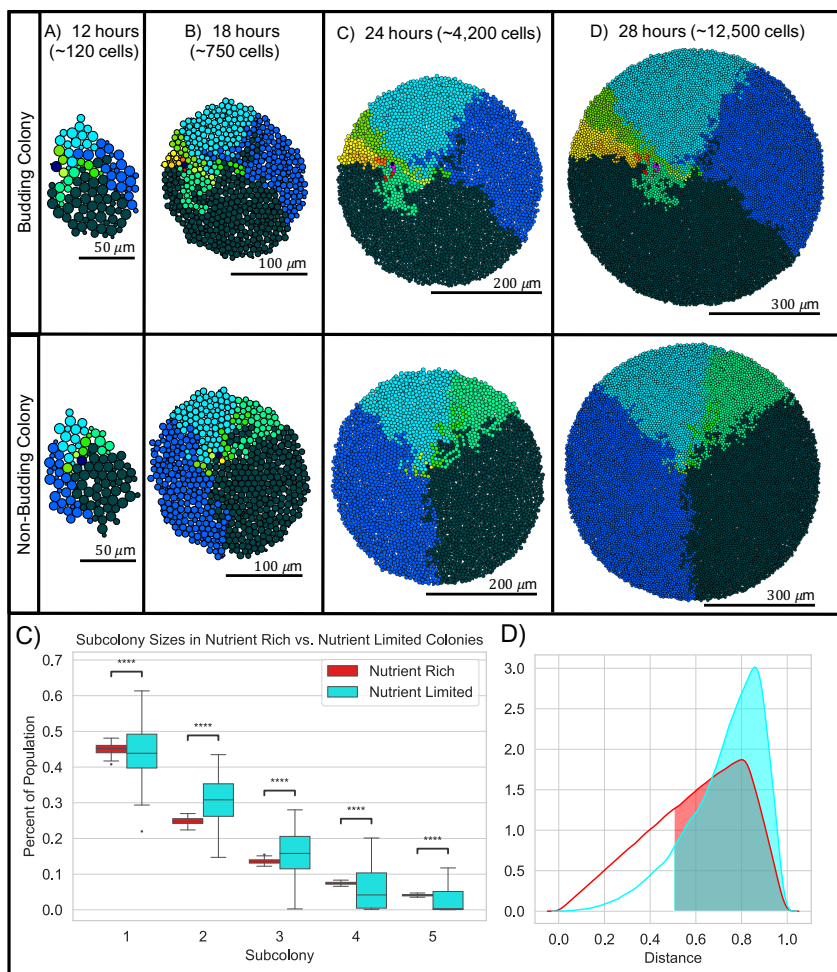


Figure 2.17: Nutrient Limitation Drives Spatial Organization of Cells. We compared growth and overall spatial organization between four different types of colonies (budding/non-budding, nutrient rich/nutrient limited). Cells in colonies are colored according to the unique subcolony each cell belongs to: Subcolony 1 (dark green), Subcolony 2 (blue), Subcolony 3 (cyan), Subcolony 4 (teal), Subcolony 5 (light green), Subcolony 6 (yellow-green), Subcolony 7 (yellow), Subcolony 8 (gold), Subcolony 9 (orange), Subcolony 10 (red), Subcolony 11 (magenta), Subcolony 12 (purple) and Subcolony 13 (pink). Typical simulation output from budding (A) and non-budding (B) colonies grown in nutrient limited conditions. (C) Nutrient limitation results in more variation in the percentage of the population contained in each subcolony (Subcolony 1 ($p = 2.779e-20$), Subcolony 2 ($p = 6.143e-25$), Subcolony 3 ($p = 4.636e-23$), Subcolony 4 ($p = 6.574e-32$), and Subcolony 5 ($p = 2.985e-09$)). (p -values were calculated using the Levene test for equal variances.) (D) Birth Location of cells born within the last hour of colony growth changes significantly between nutrient rich (red) and nutrient poor (cyan) conditions ($p < 1.0e-32$). (p -value was calculated using the Kolmogorov-Smirnov statistic in python. See Section 2.3.2 for details.)

where nutrient are highly depleted (Figure 2.17 D). Moreover, since lack of nutrients slows down cell division, most actively dividing cells are near the boundary of the colony. As such, our model captures emergent patterns of colony organization due to changes in local cell-cell interaction dynamics caused by rapid expansion and spatial rearrangement of newly born cells at the colony boundary. An interesting extension of our model would be to compare simulation results with our simple model of nutrient limitation to simulation results representing a nutrient field that changes in time due to diffusion and consumption by live cells.

Our results also reveal that together, budding and nutrient-limited growth facilitate subcolony connectivity. We observe that nutrient-limited growth in our simulations results in better defined and more contiguous subcolonies than in the nutrient-rich case (Figure 2.17 A, B). This is likely due to the effects of nutrient limitation on the cell cycle for densely packed regions of the colony. Furthermore, the nutrient limitation condition creates high variance in the final number of cells that will be in each subcolony at a particular time (Figure 2.17 C). We believe this variation is caused by different patterns of movement of the founder cell. Namely, if the founder cell is moved to areas of high nutrient availability due to interactions with neighboring cells, it can produce more offspring. Otherwise, if cell-cell interactions cause it to remain stationary in low-nutrient regions, its cell cycle progression will be slowed resulting in fewer total offspring. These observations further support the idea that nutrient-limited growth together with budding division result in colonies with the most well-defined subcolony shapes. We hypothesize that these contiguous subcolonies are precisely the discernible sectors observed in experimental yeast colonies. As such, our mathematical framework will prove valuable insight for generating hypotheses on sectoring behavior that can be compared to experimental studies. This variation in subcolony size from nutrient limitation will have a different impact for cells grown on a plate (i.e., physical media) compared to those in liquid culture (where cells are in a well-mixed liquid environment) and our work identifies population structure as a novel and unappreciated difference in these two common experimental conditions. As such, changes in population structure are a new lens through which experiments in these conditions can be compared. In addition to its impact on local spatial organization and subcolony structure, resource-driven structuring in microbial groups has been linked to colony fitness and survival [109, 112]. Our results suggest that nutrient-rich colonies have less structure and therefore allow a greater number of lineages to be maintained as the colony expands. This is in contrast to our simulations of nutrient-limited colonies where the number and sizes of subcolonies undergoes a lot more variation. An interesting avenue of further study would be to run simulations for a longer amount of time and assess how nutrient limitation impacts overall colony fitness and survival other than the overall slower growth we describe in our results.

We note that there are many cellular processes that our model did not explicitly consider. First, we did not include cell rotational forces in our model. Based on results in this chapter, biophysical properties of attached mother-daughter cell pairs play a role in local organizational properties of colonies. Thus, we believe extend-

ing our model to include rotational forces and capture more detailed geometry of attached mother-daughter cell pairs may further impact local organization dynamics. In addition, in nutrient-limited conditions, cells may enter a state of quiescence where they choose not to enter the $G1$ phase of the cell cycle [36]. Although we did not directly include quiescence as a state in our model, cells in our simulations do attain an effective “quiescence” state as they stop growing and creating new buds due to limits on their cell cycle progression. Third, our model does not include cell death. Although cell death is definitely a contributor to colony structure, specifically the 3D structure of colonies where the middle section is nutrient-starved, because the average lifetime of yeast cells exceeds the time-scale of our simulations [68, 108], the impact of cell death on the organization phenomena we observe is negligible.

Finally, we have included only one type of budding behavior. Yeast are capable of multiple budding strategies. For example, we do not consider multiple cell mating types and only consider diploid cells. While diploid yeast cells primarily divide in a bipolar budding pattern [145], haploid cells can bud in an axial pattern [27] and have been shown to divide faster than diploid cells [100]. If we only considered diploid cells with no variance in the location of the new birth scar, all cells would appear co-linear to each other. However this extreme case is unlikely to occur within large diploid colonies [27]. We model new budding sites in a manner similar to the implementation by Wang et al. [169], but we consider only daughter cells to be limited to bipolar cell division. In addition, we assume that a mother cell has at most one daughter cell attached, but this is not guaranteed for all strains [143]. Furthermore, we also neglect environmental factors in influencing budding strategies. While cells in larger yeast colonies have been shown to exhibit pseudohyphal growth through switches in budding strategies under nutrient-limited conditions [14, 15, 161], it was reported by Binder et al. [15] that colony expansion appears to evolve non-uniformly after approximately 100 hours. Because this far exceeds the time scales of the colonies we produced, we believe this effect to be negligible. However, we conjecture that for large colonies, pseudohyphal growth under nutrient-limited conditions will induce rapid changes in colony organization near the colony periphery.

Indeed several agent-based modeling frameworks have been developed to study the factors impacting yeast colony growth. Jönsson et al. developed a similar center-based model to study the impact of biophysical factors and growth inhibition by neighboring cells on colony expanse and sparsity [79]. Their results show that cell-growth inhibition by neighboring cells and bipolar division patterns are the most significant factors impacting colony sparsity where they used colony sparsity as a measure of exploratory behavior of the colony. However, their results focused on global shape and size of colonies instead of spatial organization of cells, and they did not consider colonies larger than several hundred cells. Wang et al. developed a center-based model to study the impact of budding patterns, mating type switches, cell death and nutrient limitation on yeast colony growth [169]. Consistent with Jönsson et al, they found that bipolar budding patterns improve colony development under nutrient limitation. In addition, they found that axial budding patterns enhance mating

probability during early stages of colony growth and suggest that the frequency of mating type switch might control the trade-off between diploidization and inbreeding. Interestingly, they showed that colony expansion does not depend on the overall age of the colony. They hypothesize this is due to the fact that young cells contribute most to colony expansion which is consistent with our results. One important difference between our model and the model presented by Wang et al., is that our model explicitly includes the process of a bud growing on the mother cell. The model by Wang et al. skips the growing process and introduces new daughter cells after they have detached from the mother cell and are a more substantial size. In addition, their study focused on characterizing heterogeneity of cell types in the colony and overall shape and size of the colony and they did not analyze the impact of spatial organization of cells. Since their model does incorporate many of the important factors we discussed above as possibly having an impact on spatial organization and structure of the colony, we believe an interesting avenue for future work would be to extend our model to include the same factors they considered and investigate how the addition of the budding process changes the resulting outcomes discussed in their paper.

Many questions remain in investigating the structure of growing yeast colonies, and our model may be easily generalized to include them. In particular, we plan to extend our model by including intracellular protein dynamics to more directly study prion sectors in yeast. Despite many years of biomedical research and our detailed understanding of protein aggregation dynamics on the molecular scale, our ability to mechanistically link protein aggregation processes to their disease phenotypes at the colony level is severely lacking, especially during transitions between prion states [50, 62, 137, 174]. For example, recent studies have demonstrated that a single colony can exhibit multiple phenotypes resulting from a change in aggregation dynamics between neighboring cells [83]. Our results in Section 2.3.2 demonstrate that nutrient limitation and budding together have a large impact on emergent patterns of sector-like regions in growing yeast colonies. Moreover, an important insight of our work is the large variation in subcolony sizes that arises due to nutrient limited growth. As a consequence of this result, we can better understand that the size of a subcolony sector is not directly correlated to its birth order. Thus, quantitative information connecting spatial information of sectoring patterns with molecular level dynamics is required. What is missing is a detailed and mechanistic computational model of sectoring phenomena which can then be coupled with an informatics look at the sectors in experiments. Methods and tools for counting different colonies have been developed which may aid in detecting phenotype sectors through finding connected groups of cells [12, 20, 52]. While there are resources that quantify sectoring in microbial colonies [57, 109], to our knowledge no other studies specifically quantify the causes of phenotype sectoring in yeast colonies. Our modeling framework can be readily extended to a multi-scale system to address phenotype sectoring in yeast through the incorporation of prion dynamics as a subcellular process (Figure 2.1). This offers the opportunity to make more meaningful comparisons between data and models and to infer, predict, and eliminate hypotheses on the characteristics of the

sectoring patterns.

In this study we proposed a 2D model of yeast colony growth, where directional forces along the z-axis are neglected. In reality, yeast colonies are three-dimensional (3D). Thus, there may be some important factors our model does not consider. For example, in 3D, high agar concentrations have been shown to influence formation of complex structures such as the vertical growth of stalk-like structures in yeast colonies [142]. In this scenario, complete contact with the substrate is not possible for all cells further impacting nutrient availability. Experimental evidence also shows that microbial colonies do not grow outward in a strictly radial direction due to varying agar concentrations [116] and de-activation of the FLO11 gene which reduces adhesion of the cells to the plate surface [134]. Thus, while cell dependency on nutrients is more complicated to model in 3D, these results suggest it will remain an important factor driving variances in colony morphology. Moreover, we hypothesize that 3D configuration may further promote local spatial organization, sectoring patterns and ordered structure of the colony. In our model, we study the formation of sector-like regions in the context of subcolonies. Under nutrient limitation, our model shows that subcolony sectors become more well-defined and begin near the location of the founder cell. However in 3D, different technologies used to capture and analyze sectoring phenomena may be limited to the colony surface. The capability of *in silico* experiments creates the advantage of allowing for more detailed analyses of sectoring phenomena where existing scientific tools and technologies may not provide sufficient information.

2.5 Conclusions

In this study, we introduced a novel two-dimensional, cell-based model describing the growth and movement, structure, and spatial organization of a colony of yeast cells to emphasize the importance of capturing budding behavior in these models. Our findings show that budding greatly impacts the local connectivity of a cell and that, together with nutrient limitation, acts to promote connected sectors with respect to the subcolony structure and produce highly variable subcolony sizes. Together, these findings offer novel interpretations and insights to observed sectoring phenotypes in yeast. We aim to investigate the multi-scale nature of these phenomena in future studies by extending our approach to include intracellular dynamics.

Chapter 3

[PSI]-CIC: A Deep-Learning Pipeline for the Annotation of Sectored *Saccharomyces cerevisiae* Colonies

This chapter is the paper I submitted in February 2024 which was co-authored by Dr. Wesley Naeimi, Dr. Tricia R Serio, and Dr. Suzanne Sindi. I led the development of formalizing and testing the methodology as well as writing each of the sections in this chapter. Dr. Wesley Naeimi curated the image data that was used to test the methodology in this work (see Appendix A for details).

3.1 Introduction

Prion diseases are a class of fatal and incurable neurodegenerative diseases in mammals that include Creutzfeldt-Jacob disease, fatal familial insomnia, Gerstmann-Straussler-Scheinker syndrome, and Kuru [127]. Early research by Prusiner [126, 127] suggested that a protein—not a virus—coined as a proteinacious infectious particle—or prion—was the key infectious agent in all types of prion diseases regardless of the mammalian host, thus establishing what we know today as the prion hypothesis [120, 171]. These alternatively folded proteins act as templates capable of inducing normally folded proteins of the same type to misfold [74, 75, 128, 154] (see Figure 3.1 A). Furthermore, these alternatively folded proteins are capable of undergoing templated conversion to form aggregates [120] which are capable of growing in size or fragmenting into smaller aggregates that induce further alternative folding, thus leading to a self-replicating aggregation process [31, 74]. Since the formalization of the prion protein [128], the study of biological processes behind prion disease and the search for appropriate solutions to eradicate them remains an active area of research.

3.1.1 Yeast as a Model System

Prion proteins are not exclusive to mammals. The yeast *Saccharomyces cerevisiae* has served as a model system to understand the mechanisms underlying the appearance and progression of human diseases, including “prion-like” diseases such as Alzheimer’s [7, 13, 150]. There are at least eight naturally occurring yeast prion proteins [26, 93, 175] setting the stage for yeast-based platforms to help screen potential anti-prion drug candidates [76]. One of the most widely studied prion protein in yeast is Sup35 which is an essential release factor in translation-termination [99, 164]. Sup35 aggregates have the ability to self-propagate within yeast populations [120]. When grown on solid media single yeast cells grow into circular colonies containing approximately 1×10^6 cells and exhibit a white unpigmented [*PSI*⁺] phenotype when the prion is present. In contrast, colonies that only contain the non-prion form of Sup35 exhibit the red pigmented [*psi*⁻] phenotype [83]. Spontaneous appearance of the [*PSI*⁺] phenotype is rare, occurring in approximately one in every 10^6 cell divisions [62, 88]. Remarkably, unlike their human counterparts, the [*PSI*⁺] phenotype in yeast is reversible [62, 83, 92, 141]. Heat shock destabilizes the prion phenotype in yeast which in time gives rise to colonies exhibiting both red and white phenotypes described as sectors [32, 83]. Figure 3.1 B and C summarizes the possible events determining the phenotype of newly born cells, and how the collective prion state of cells in a colony give rise to sectorized phenotypes at the colony level. This type of data provides information on the prion state of a cell population and insight into changes to the prion phenotype in response to experimental treatments.

To understand how sectoring occurs in yeast, we need to consider the underlying dynamics (conversion, aggregation, and fragmentation) at the intracellular scale (Figure 3.1 C). Mathematical models have been proposed that explore these dynamics [35, 102, 148] with one recently proposed to explore how multiple prion strains interact [91]. However, such dynamics take place inside individual yeast cells that have their own biological properties (such as division which allows for transmission of proteins between attached mother-daughter cell pairs). The second phenotype occurs when a cell loses prions due to a transmission defect or destruction of existing prions within a cell [115]. As cells continue to divide over time and form a colony of thousands to millions of individual cells, phenotypic sectoring becomes observable [83] (see Figure 3.1 C) indicating where subsequent daughter cells did not inherit the [*PSI*⁺] prion.

Thorough understanding of these multiscale processes may require large samples of yeast colonies under different experimental settings. This however leads to two potential problems. First, experimental settings do not always result in deterministic observable experimental output. Second, analyzing each individual colony is an extremely tedious process; colonies are often scored as sectorized or pure, but there is additional information based on the size and number of sectors to help better our understanding of prion curing. For these reasons, large-scale screening involving detailed colony phenotypic analysis is infeasible without the use of suitable instruments and algorithms capable of utilizing this information.

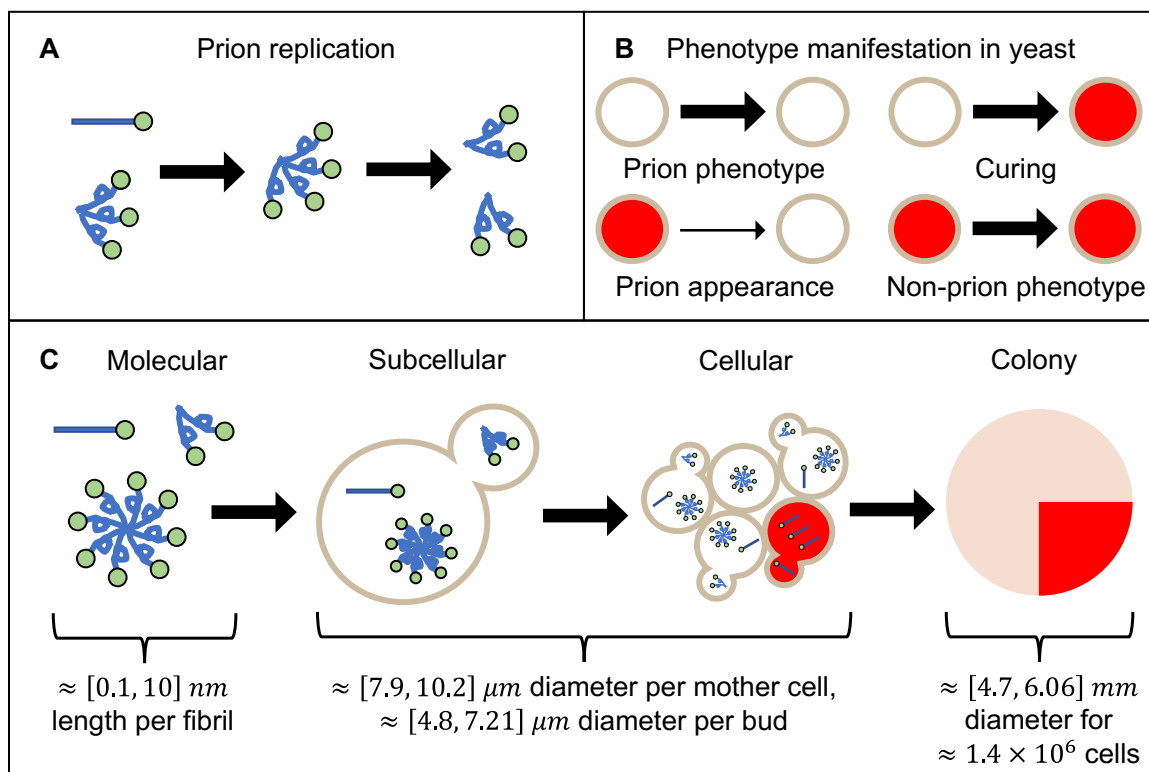


Figure 3.1: **Yeast prion phenotypes are the result of multiscale processes.**

A: At the molecular scale, alternatively folded proteins (twisted) act as templates that convert normally folded proteins (straight) into the alternatively folded form and assemble into aggregates. The aggregates then split into smaller segments (fragmentation) which increases the number of aggregates.

B: At the cellular scale, the presence of prion aggregates inside individual cells (represented as circles) are responsible for their white color, while the absence of prions allows pigment generation and gives them their red color. The prion phenotype could be lost sporadically, resulting in cured cells, while in rare instances—1 in 10^6 —(indicated by a thinner arrow) the prion phenotype appears spontaneously.

C: Phenotype expression in yeast involves multiscale processes. The dynamics inherent in protein misfolding are found at the molecular level (A). At the subcellular level, since prions are also found in yeast which undergo their own process of reproduction, allowing transmission of prions between attached cells. At the cellular level, the presence of prions within a cell in turn determines their phenotype (B). At the colony level, the collection of intercellular interactions that occur on the scale of a cell results in structured regions of one phenotype within the colony. Molecular scale was visually estimated from image data in [81]. Subcellular and cellular scales were estimated using data from [179]. A rough estimate for the colony scale was obtained using the minimum and maximum averaged surface area measurements of a mother cell in [179], multiplied by the approximate number of cells in colonies from data in [80].

3.1.2 The Role of Image Analysis

Technological advances have made it possible to use computational approaches to handle variation in experimental data and efficiently quantify large, complex biological datasets with the added benefit of reducing manual laboratory labor while producing outcomes comparable to manual labor. Such methods include software and image-based methods to automate microbial colony counting [29, 87, 162], edge detection [24] and for circular objects, the circle Hough transform [6, 73]. With the availability of greater processing power, deep neural networks or more specifically, convolutional neural networks (CNNs) have made it possible to quickly identify objects of interest in general datasets when conventional methods are inadequate. Deep learning methods applied to images typically have one or two objectives. One class of methods seek to classify entire images by associating them with a set of user-defined classes; a couple examples of well-known models include ResNet [156] and VGG [107, 147]. Another class of methods use semantic segmentation to assign user-defined classes to each pixel in an image, rather than assigning classes to the image as a whole; such models include U-Net [136] and Mask R-CNN [65]. Methods in both cases are usually either trained from scratch or build off of a pre-trained model—such as ImageNet [37]—then re-trained on a new dataset to be applicable to specific settings. It is also possible to construct computational pipelines using both classes of deep learning methods to obtain ensemble data from colony-level images of yeast. For example, the model proposed by Carl et al. [25] segments and classifies individual yeast colonies from images of plates using both semantic segmentation and image classification and demonstrates performance superior to the tool CellProfiler [87] for their scenario.

The majority of image-based models applicable to yeast however are designed for micro-colony data where individual cells are clearly visible using cell microscopy techniques, while efficient and similar models for large-scale colonies visible at eye level are lacking. While deep learning methods for semantic segmentation have been developed for microscopy images of yeast such as YeastSpotter [98], YeaZ [40], and YeastNet [139], each method is primarily optimized for cellular-level images of yeast. Carl et al. [25] has a method grouping colonies into broad classes, but the manual annotations in the images used in this method are limited to are used for classifying colonies into these broad classes and do not account for size and frequency of individual sectors. Furthermore, we do not yet have a related analysis performed on sectored *S. cerevisiae* colonies, nor do we have a dedicated toolset geared for quantifying individual sectored colonies from colony-level image data with human-comparable output. The work described in this Chapter aims to use a blend of computational tools to analyze and quantify colony level image data to aid in the analysis of *S. cerevisiae* prion experiments.

The goal of this Chapter is to create a toolset for learning more about the mechanisms behind prion protein dynamics that drive observable changes at the colony level. To that end, we introduce [*PSI*]-CIC ([*PSI*] Colony Image Classifier) a computational pipeline to segment and quantify individual colonies of *S. cerevisiae* found

in image data using both deep learning and conventional tools. In Section 3.2 we detail the $[PSI]$ -CIC algorithm from segmentation of plates to classification of colonies. Section 3.3 shows results of $[PSI]$ -CIC’s performance on a set of images where prion curing is induced by heat shock. Section 3.4 details a discussion of the $[PSI]$ -CIC and how this work has an impact on the use of image segmentation in the context of prion dynamics in yeast.

3.2 Methods

In Section 3.2.1, we describe the components of $[PSI]$ -CIC for analyzing sectored yeast colony phenotypes (see Figure 3.2). In the first component, we construct a neural network to perform image segmentation on plates containing hundreds of yeast colonies, then use the output of the network to locate and extract individual colonies. In the second component, we take each colony extracted previously and use image processing tools to classify colonies as $[PSI^+]$, $[psi^-]$, or sectored and estimate the frequency and shape of sectors present in each colony. Section 3.2.2 discusses how we train the network used in this component to recognize colonies. For this process, we detail how to incorporate synthetic training data of yeast colonies (see Appendix A.2) into the training process to both address the issue of limited annotated data available and to show its effectiveness in aiding segmentation of real colonies. Section 3.2.3 details how we evaluate the performance of the $[PSI]$ -CIC algorithm on the annotated experimental images.

3.2.1 $[PSI]$ -CIC Algorithm

We follow the approach by Carl et al. [25] and use the U-Net architecture for performing semantic segmentation on images of plates to assign a label to every pixel in the images (see Figure 3.2). U-Net is a type of supervised CNN originally designed for biomedical image segmentation [25, 122, 136], but is widely generalized to other segmentation tasks. For the implementation of U-Net in this Chapter, we modify the original architecture [136] in the following way: First, we use images of size 1024×1024 as input instead of size 572×572 . Next, we apply padding to the image before each convolutional layer to preserve the spatial resolution, which we believe is reasonable since each image almost exclusively contains background pixels on their borders. Finally, we modify the output layer such that the final segmentation is of the same spatial resolution as the input image and has three feature channels corresponding to one of three classes: background, white colony, or red colony. A softmax activation function is applied to the output of the last layer to obtain the probability of each class per pixel, then the label assigned to each pixel is the maximum probability across the three classes.

After U-Net is sufficiently trained and segmentations of the images are obtained, we use the resulting segmentation as input for an object detection method. Since colonies in each image appear circular by eye, we use the circle Hough transform

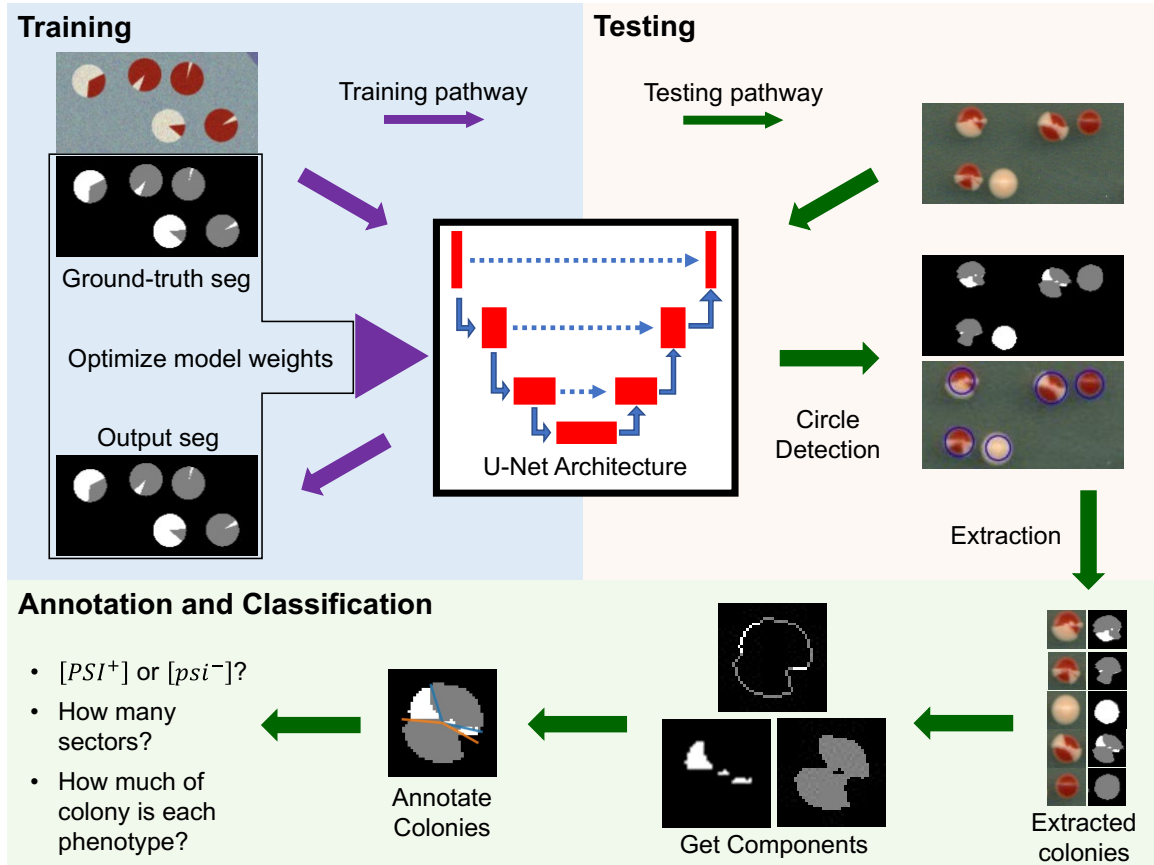


Figure 3.2: **Illustration of [PSI]-CIC.** Our proposed pipeline consists of a segmentation-classification framework, where we semantically segment images of plates containing hundreds of yeast colonies (see Section A.1) for the purpose of locating and classifying individual colonies. We create synthetic training images with corresponding ground-truth masks (details in Appendix A.2) used to fine-tune a modified U-Net architecture (purple arrows) (see Section 3.2.1) for performing segmentation on images of full plates. We then apply the sufficiently trained U-Net (green arrows) to segment the test images where colonies are detected (see Appendix A.3) and cropped for classification. The classification step leverages the spatial information in the segmentation to propose an annotation of the regions in the colony which is used to classify a colony as $[PSI^+]$, $[psi^-]$, or sectored.

as our method to detect colonies captured within the segmentation. Each colony detected with this method is recorded and cropped out of both the image and its segmentation for use in the classification step of $[PSI]$ -CIC (See Figure 3.2). Details for the implementation of the circle Hough transform is explained in Appendix A.3.

Once individual colonies and their segmentations are extracted from the full-size images, the goal is to classify each colony as $[PSI^+]$, $[psi^-]$ or sectored. Figure 3.3 A shows the annotation procedure for counting and quantifying sectors in each detected colony. The procedure here uses the colony segmentation as input, constructs and analyzes a proposed annotation or “idealized” sectoring using the colors of the colony regions, then uses the properties of the annotation to classify the colony.

We make a few assumptions about the colony segmentations in order to classify colonies in our experimental images. Since colonies appear circular, we first assume that the colony segmentations are sufficiently circular and that the center of the colony is also the center of the image. Since the red and white regions of colonies in the experimental images appear sector-like by visual inspection, we also assume that each red and white region of the colony originate from the center and expand outward with linear edges, forming the edges of a geometric sector. Finally, we assume that the colony boundary forms the arc of each sector-like region, which closes and bounds each region.

The following process uses these assumptions to propose idealized regions (see Figure 3.3 A) for each colony. Given a colony segmentation, we first decompose it into its interior and boundary components. A pixel in the colony segmentation is considered a boundary pixel if it is a colony pixel that is also adjacent to a background pixel. Otherwise, that pixel is considered to be an interior pixel. For simplicity, we skeletonize the boundary of the colony so that it has pixel width 1. Next, we further decompose both the interior and boundary components of the colony respectively into their red and white regions, and then find the connected components of red and white pixels separately on the boundary. For each component, we construct an “idealized” sector (see Figure 3.3 B) whose boundaries are represented using the component itself as the arc and two lines connecting the endpoints of the arc to the colony center.

To approximate where to draw the lines representing the other two boundaries of an idealized sector, we proceed to find the endpoints on the arc using two methods. This relies on there being no more than 2 endpoints for each skeletonized boundary. We first use the hit-miss algorithm within the SciPy package [168] to find the endpoints of the skeletonized boundary. For the second case, we scan each pixel on the skeletonized boundary and label a pixel as an endpoint if there is exactly one other boundary pixel adjacent to it. Note that this brute force method is capable of finding endpoints on a corner of a skeleton, while the hit-miss algorithm is capable of finding endpoints near a corner. We then take the union of endpoints located from both methods, because initial observations suggest they correct each other’s shortcomings.

The remaining two boundaries of the idealized sector are then drawn using Bresenham’s line algorithm [16] to connect the endpoints with the colony center via lines in pixel space, resulting in a closed shape representing the entirety of the idealized

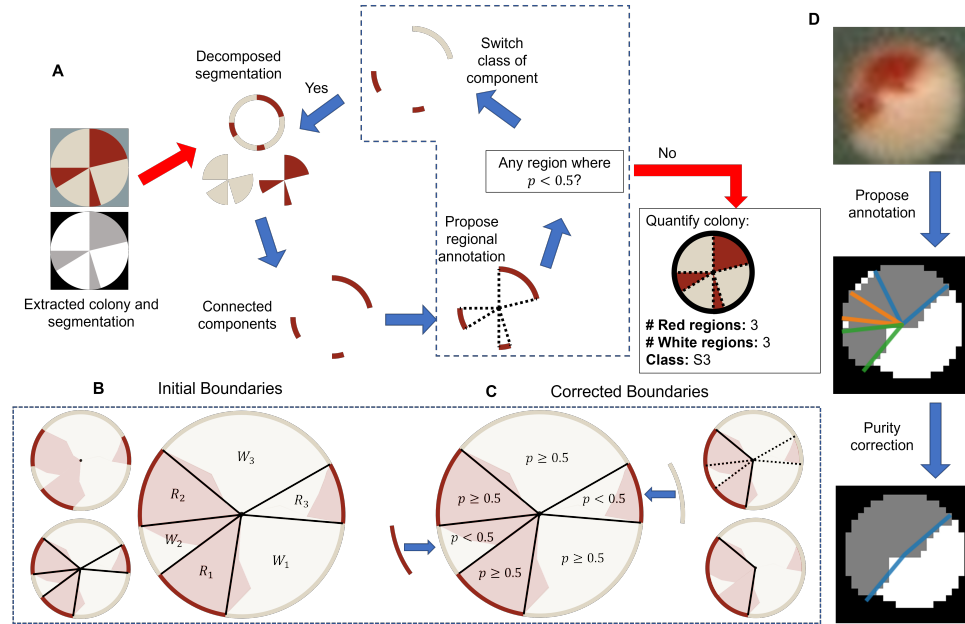


Figure 3.3: Novel annotation and sector counting procedure. (A): Flowchart of our proposed scheme to estimate and quantify red and white regions. We decompose the classes from the output segmentation into the interior and boundary components, decompose the boundaries by color, then count their connected components. We then propose an annotation for the region each sector occupies and check whether its interior also contains a significant number of pixels of the same color, defined as purity, p (see Section 3.2.1). Should a majority of pixels in the interior of the region contain pixels of a different color, the color of the boundary component will be switched. The process repeats until all boundary components are consistent with their corresponding interior regions. The number of consistent red regions remaining is used as the prediction for the number of sectors present in the image. (B): Assuming that the colony segmentation is split into red and white pixels (left top), we take the boundary of the colony and find the connected components of the red and white colony pixels respectively. We locate the endpoints of each component corresponding to the interfaces between red and white components, and for each point construct a line segment from that point to the center of the colony (left bottom). The line segments partition the entire colony into idealized regions whose color is defined by the boundary in each region (i.e. R_1 for red and W_1 for white) (right). (C): To ensure regions are consistent with their color, we use the purity metric as defined in Section 3.2.1 to find the proportion of pixels inside each region that have the same color as the pixels on the boundary (left). Any regions whose purity metric is less than 0.5 will have the outer boundary change color (right top). After the change, adjacent components that have the same color will be merged (right bottom). (D): Example of B and C applied to a segmentation of an experimental colony. An annotation of the red and white regions is proposed from the colony segmentation and its regions corrected in order to satisfy the purity constraint (see Section 3.2.1).

sector boundary, while the collection of pixels within represents the interior of the sector. This process is repeated for all red and white regions in the colony, resulting in the full initial annotation representing the regional breakdown of the colony.

Purity Metric

Since we assume each region in a colony appears sector-like, we attempt to quantify how well colony segmentations meet this assumption, then perform an additional step for regions which are inconsistent with this assumption. To quantify a region of a colony, we need to analyze the physical structure of the region itself and use simple methods to address inconsistent structure present in the segmentation. To that end, we define a metric we call “purity” to denote the proportion of pixels in each red/white region that are of the same class to measure how well each proposed region and the aggregation of the regions in a colony resemble well-defined sectors.

We first define purity in terms of a single region of a colony. After creating the regions as described in Section 3.2.1, the color of the region (red or white) is assigned to be the same color as the pixels in the region along the boundary of the colony. If we have a sectored colony with a red regions and b white regions (see Figure 3.3 B), we denote the red regions as R_1, \dots, R_a and the white regions as W_1, \dots, W_b . Next, we denote the function N to be the number of pixels in a region that have a given color. For instance, we define $N(R_i, red)$ as the number of red pixels in region R_i , and $N(R_i, white)$ as the number of white pixels in region R_i . Since these are the only two colors for colony pixels in our segmentations, the total number of colony pixels in the region is therefore the sum: $N(R_i, red) + N(R_i, white)$. We then define the purity, p of region R_i with respect to the red pixels as

$$p(R_i, red) = \frac{N(R_i, red)}{N(R_i, red) + N(R_i, white)}. \quad (3.1)$$

Similarly, we define the purity of region W_j with respect to the white pixels as

$$p(W_j, white) = \frac{N(W_j, white)}{N(W_j, red) + N(W_j, white)}. \quad (3.2)$$

Equations (3.1) and (3.2) are also described as the proportion of colony pixels within the region that are red or white respectively, and thus give values between 0 and 1, where values closer to 1 indicate the estimated region in the colony is more sector-like with respect to the color of the region, and values far away from 1 indicate the region is far from an idealized sector based on our assumptions.

To define purity for an entire colony, we apply weights to each region to account for size differences between the regions. We first define N_R and N_W to be the number of pixels across all red regions and all white regions respectively, i.e.

$$\begin{aligned}
N_R &= \sum_{i=1}^a [N(R_i, red) + N(R_i, white)], \\
N_W &= \sum_{j=1}^b [N(W_j, red) + N(W_j, white)].
\end{aligned} \tag{3.3}$$

Without loss of generality, for each region R_i and W_j , we assign weights, $\mu(R_i)$ and $\mu(W_j)$, where

$$\begin{aligned}
\mu(R_i) &= \frac{N(R_i, red) + N(R_i, white)}{N_R + N_W}, \\
\mu(W_j) &= \frac{N(W_j, red) + N(W_j, white)}{N_R + N_W}.
\end{aligned} \tag{3.4}$$

We then define colony purity, p_w , as the weighted average over all regional purities, i.e.

$$p_w = \sum_{i=1}^a p(R_i, red)\mu(R_i) + \sum_{j=1}^b p(W_j, white)\mu(W_j) \tag{3.5}$$

or equivalently,

$$p_w = \frac{\sum_{i=1}^a N(R_i, red) + \sum_{j=1}^b N(W_j, white)}{N_R + N_W}. \tag{3.6}$$

Just like in Equations (3.1) and (3.2), Equation (3.6) above takes a value between 0 and 1, where values closer to 1 indicate the estimated regions in the colony are collectively more sector-like with respect to the output segmentation. On the contrary, if p_w is a number strictly between 0 and 1, this indicates that the estimated regions do not completely capture idealized sectors. Values of p_w closer to 0 indicate greater disagreement between the proposed region and the corresponding region in the output segmentation of the colony.

Purity Correction

Depending on the shape of the colony segmentation, regions may not sufficiently resemble idealized sectors. Here, we include a procedure to identify inadequate regions by using the value of the purity metric to perform a ‘‘correction’’ of those region with respect to the colony segmentation. This results in a proposed regional annotation capturing standout regions in the colony segmentation (see Figure 3.3 C).

We assume that the red and white regions have been estimated and the purity for each has been obtained using Equations (3.1) and (3.2). We then impose a constraint on the purity of each region such that we require at least 50% of a region’s pixels to

be of the same color as the region itself to satisfy our assumption that the region is adequately sector-like. If this constraint is not met for a region, then we swap the labels of the pixels on the region's boundary which in turn changes the color assigned to the region. Mathematically, without loss of generality, if region R_i has a purity of less than 0.5 (i.e. $p(R_i, red) < 0.5$), then region R_i has more white pixels than red pixels. For such regions, we change the labels of the pixels along the arc of region R_i corresponding to the colony boundary from red to white. As a consequence, modifying the color of the boundary leads to changing the assigned color of the region from red to white. This process is repeated for all red and white regions independently. Following this procedure, regions are merged if their corresponding boundary pixels are of the same color (see Figure 3.3 C). By using the median inequality, it could be shown that if the purity of each of these regions is at least 0.5, then the resulting merged region will also have purity greater than 0.5. For example, if there are n red regions adjacent to each other following the correction, then

$$\begin{aligned}
0.5 &\leq \min_{1 \leq i \leq n} p(R_i, red) = \min_{1 \leq i \leq n} \frac{N(R_i, red)}{N(R_i, red) + N(R_i, white)} \\
&\leq \frac{\sum_{i=1}^n N(R_i, red)}{\sum_{i=1}^n [N(R_i, red) + N(R_i, white)]} \leq \max_{1 \leq i \leq n} \frac{N(R_i, red)}{N(R_i, red) + N(R_i, white)} \\
&\leq \max_{1 \leq i \leq n} p(R_i, red). \tag{3.7}
\end{aligned}$$

If there were any changes made to regions that did not satisfy our constraint, we then repeat the procedure as described in Section 3.2.1 to propose a regional annotation of the colony accounting for the swapped boundary pixels, and recompute the purity for all regions in the colony segmentation. This procedure is repeated until we obtain a proposed regional annotation of the colony where all regions satisfy our constraint.

At the conclusion of purity correction, the color of the pixels on the outer boundary for each independent region will be the same color as the majority of pixels in those regions. We then use Equation (3.6) as described in Section 3.2.1 to score how well the proposed regional annotation collectively captures sectoring behavior in the colony.

Class Assignment

Upon obtaining annotations of colonies whose regions all meet the condition described in Section 3.2.1, the number of red and white regions remaining are used to assign a qualitative class on each colony. Colonies with no red regions and at least 1 white region are labeled as $[PSI^+]$. Colonies with at least 1 red region but have no white regions are labeled as $[psi^-]$. Colonies that have at least 1 red and white region are labeled as sectored. In addition, sectored colonies are given a secondary label indicating frequency of sectors. A sectored colony is labeled as S1 if it has one sector, S2 if it has two sectors, and so on.

3.2.2 Training (Image Segmentation)

Due to the lack of hand annotated colony images, we turn to training a neural network with synthetic images where it is possible to efficiently create ground-truth masks labeling each pixel. An example of a synthetic image generated with its corresponding ground-truth mask is shown in Figure A.2. The objective of this approach is to generate sets of synthetic images of yeast colonies which exhibit key features of the yeast colonies found in the experimental images.

The key features in the images we consider for this work apply to the colonies and the background information. For the colonies, these features consist of circular colony shapes where each colony exhibits sectorized red and white regions with slight color variations. We use two representative colors (1 red and 1 white) to fill each circle representing the colony, where the circle is filled with the white color and the red sectors are overlaid. For the background, these features include the colors of the plate, the table on which the plate rests and the border of the plate where aberrations are present, each of which exhibit slight color variations. We choose a representative color independently for each of these three features. All of these features are subject to Poisson noise to introduce slight color variations that are observed in the experimental images (see Section A.1).

Two corresponding ground-truth masks are generated alongside each synthetic image representing a pixel-by-pixel segmentation of the synthetic image and frequency of sectors per colony respectively. The first mask is created by thresholding the synthetic image, with each pixel in the mask depicting the true label of every pixel (red, white, background). The second mask is generated by placing a small non-zero region at the center of colony, whose intensity is greater when more sectors are present. For simplicity, all the synthetic training images used here have exactly one sector in each colony. More details pertaining to the process for generating the synthetic images with corresponding ground-truth masks is described in Appendix A.2.

After the masks are created, the synthetic image is subject to Poisson noise to introduce slight color variations that are observed in the experimental images (see Section A.1). Both the synthetic images and the masks are each saved with size 1024×1024 . A total of 200 images with their two corresponding masks were generated for this study using the process described in this Section. Out of these images, 150 were used directly for training U-net, while the remaining 50 were set aside for validation. We use Google Colaboratory to train our U-Net architecture on the 150 images using the configuration described in Appendix A.4. Once U-Net is sufficiently trained, we use the experimental images as input to U-Net to obtain an output segmentation for the classification step of the $[PSI]$ -CIC algorithm. Since our image set does not include pixel-by-pixel annotations of the experimental images, the quality of the segmentations were judged by eye before a usable set of parameters for U-Net was used for the final version of our $[PSI]$ -CIC algorithm.

3.2.3 Evaluation

Labels are assigned to each colony at the end of the $[PSI]$ -CIC algorithm (Section 3.2.1) such that the conditions described in Section 3.2.1 were met. $[PSI]$ -CIC predicts colonies with only one colored region as either $[PSI^+]$ if they were purely white or $[psi^-]$ if they were purely red. Colonies that have at least one red and one white region are first labeled as sectored, then are assigned a secondary label indicating the frequency of sectors. In the experimental images, sectored colonies have at most five sectors, so the possible classes assigned to sectored colonies are S1, S2, S3, S4, and S5, denoting both a sectored colony with its frequency of sectors. We evaluate the performance of $[PSI]$ -CIC by comparing the proportion of extracted quantifiable colonies whose true labels match those predicted by $[PSI]$ -CIC, both with and without the secondary label for sectored colonies.

3.3 Results

Here we present results on the performance of $[PSI]$ -CIC on segmenting and classifying colonies from the images used in this work. Section 3.3.1 provides results on the segmentation and classification of colonies found within the training images. Section 3.3.2 presents results on the segmentation and extraction of quantifiable colonies, indicating how much of the annotated colony data $[PSI]$ -CIC was able to isolate. This section also provides results on the classification performance of $[PSI]$ -CIC. We show that our method is sufficiently accurate at classifying colonies as $[PSI^+]$, $[psi^-]$ or sectored.

3.3.1 Training Images

Figure 3.4 A shows an example of one synthetic image and its corresponding segmentation with distinguishable colonies. From the 150 images used to train U-Net, we obtained a cross-entropy loss of 0.0022 and achieved a segmentation accuracy of 99.96% for the training and validation images after 24 epochs. Approximately 12,786 colonies from the synthetic images were extracted for classification. The remaining 7,214 colonies were excluded since their centers were predicted to be within 150 pixels from the border of the image.

When only the number of connected components of red and white boundary regions were considered, approximately 98.2% of those colonies (12,250 colonies) were correctly classified as having exactly one sector, while the other 236 colonies were incorrectly classified as either $[PSI^+]$ or $[psi^-]$. When our purity correction scheme is applied, the prediction accuracy drops to 95.8%, with 547 colonies incorrectly classified as either $[PSI^+]$ or $[psi^-]$. Upon closer inspection of the incorrectly classified colonies, we found that colonies predicted as $[PSI^+]$ had no more than 4% of their composition as red and colonies predicted as $[psi^-]$ had no more than 4% of their composition as white, regardless of whether purity correction was applied. This sug-

gests that the classification accuracy of our proposed model requires a size threshold on each sector in order to be detectable.

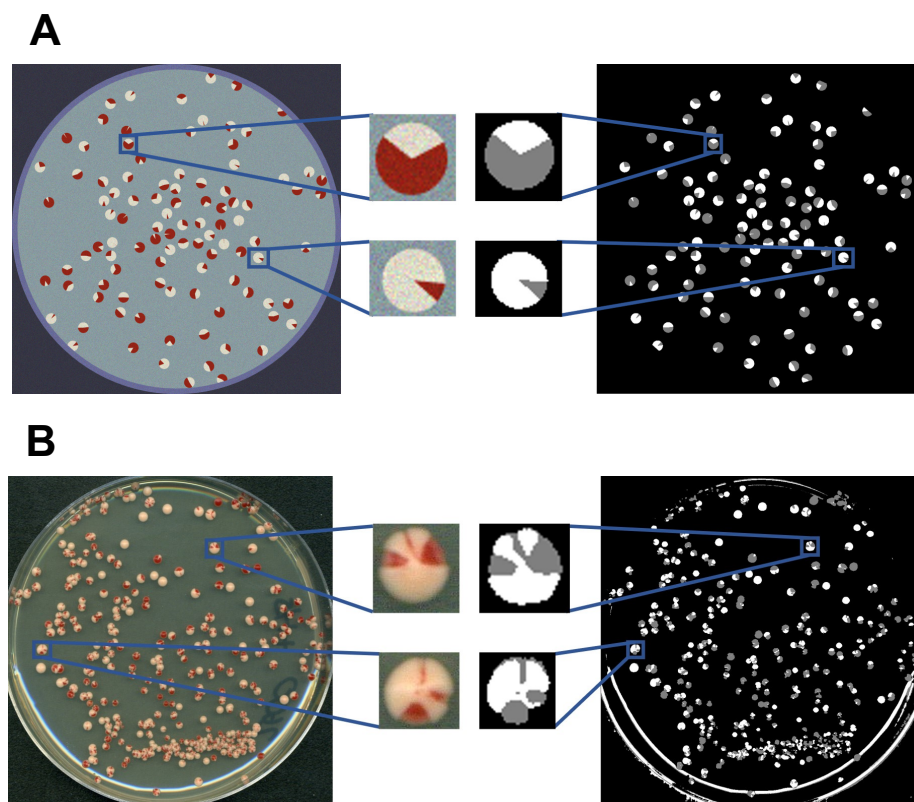


Figure 3.4: **Plate level segmentations.** (A): Example of a synthetic image (top left) and its corresponding output segmentation (top right) from the trained U-Net, with two isolated colonies shown up close. The U-Net segmentations have the following color code: Background pixels are black, red colony pixels are gray, and white colony pixels are white. (B): Output for U-Net using one of the experimental images as input. In the middle are the original representations and corresponding output segmentations from U-Net for two colonies from the image.

3.3.2 Testing Images

Using image set 1 (plates 1-5 as described in Appendix A.1), we obtained segmentations suitable enough for colony detection without pre-processing. An example of the output segmentation on one of the images in this set is shown in Figure 3.4 B. Following the execution of the circle Hough transform on these images, we obtained a total of 1,266 objects with corresponding segmentations which were extracted for classification. We note that almost all of the colonies near the edge of each plate were ignored as they were difficult to discern structurally.

Using image set 2 (plates 6-11 as described in Appendix A.1) and the color transfer methods as pre-processing (see Appendix A.1), the quality of the output segmenta-

Table 3.1: Isolating Quantifiable Colonies. Breakdown of the number of colonies found on each plate from the images used in this study. For each image, “True” is the total number of colonies in the image that a biologist performing manual counting would be considered quantifiable, or colonies that could be analyzed with simplicity. “Detections” is the number of objects considered for classification, regardless of whether or not they were of quantifiable colonies. Following image segmentation, “TP” (True Positives) is the number of quantifiable colonies extracted, “FN” (False Negatives) is the number of quantifiable colonies that were not detected, and “FP” (False Positives) is the number of non-quantifiable colonies detected. Precision is defined as $TP/(TP+FP)$, which is the proportion of all detections consisting of quantifiable colonies. Recall is defined as $TP/(TP+FN)$, which is the proportion of all quantifiable colonies detected.

Plate	True	Detections	TP	FN	FP	Precision	Recall
1	355	322	243	112	79	0.755	0.685
2	190	156	122	68	34	0.782	0.642
3	269	318	186	83	32	0.853	0.691
4	236	283	192	44	91	0.678	0.814
5	177	187	151	26	36	0.807	0.853
6	127	139	121	6	17	0.877	0.953
7	106	122	106	0	16	0.869	1
8	92	98	81	11	17	0.827	0.88
9	127	101	91	36	10	0.901	0.717
10	112	119	105	7	14	0.882	0.938
11	131	136	120	11	16	0.75	0.779

tions has significantly improved to the point where clearly distinguishable colonies are extractible. We obtained a total of 715 objects with corresponding segmentations which were extracted for classification. Similarly, nearly all colonies near the edge of each plate were ignored.

Across both image sets, we detected approximately 1,981 circular objects (see Table 3.1). From these objects, 1,585 were inspected to be of quantifiable colonies. Approximately 38 circular objects (which included 30 quantifiable colonies) had ill-defined estimated regions and thus were excluded from further analysis. After this, we had 1,555 quantifiable colonies which we classified and compared against manual annotations.

From the quantifiable colonies, 415 colonies were predicted to be sectoried, with the number of sectors predicted ranging from 1 to 3. Approximately 89.5% of the quantifiable colonies across all image sets used in this work were classified the same as those manually annotated (Figure 3.5 A). For colonies labeled as homogeneous, 691 were labeled as $[PSI^+]$ and 374 as $[psi^-]$ (Figure 3.5 B). In contrast, if we only count the number of connected components on the boundary without performing purity correction, we obtain only a 50.4% accuracy in predicting colony states, demonstrating that our purity correction scheme in $[PSI]$ -CIC performs better for estimating regions in the colony segmentations.

Table 3.2: Classification performance. Table of Precision, recall, and F1 score for each class on quantifiable colonies predicted with $[PSI]$ -CIC. For each class, the following definitions apply independently: True positives (TP) are colonies whose predicted class and ground-truth class match. False positives (FP) are the set of colonies with one predicted class, but whose manually annotated class is different. False negatives (FN) are colonies not predicted to be of a given class, but were manually annotated with that class. Precision is defined as $TP/(TP+FP)$, representing the number of colonies correctly predicted to be of the given class, divided by the number of colonies assigned this class. Recall is $TP/(TP+FN)$, representing the number of colonies correctly predicted to be of the given class, divided by the number of colonies manually annotated with the given class. The F_1 score is the harmonic mean of both precision and recall. The bottom three rows present the same measures but additionally include frequency of sectors predicted for colonies as a condition for being counted as TP.

Class	Precision	Recall	F_1 Score
$[PSI^+]$	0.969	1	0.984
$[psi^-]$	0.908	0.979	0.942
Sectoried	0.981	0.876	0.925
S1	0.788	0.810	0.799
S2	0.694	0.621	0.655
S3+	0.5	0.5	0.5

We use confusion matrices to see how both sector counting schemes place colonies

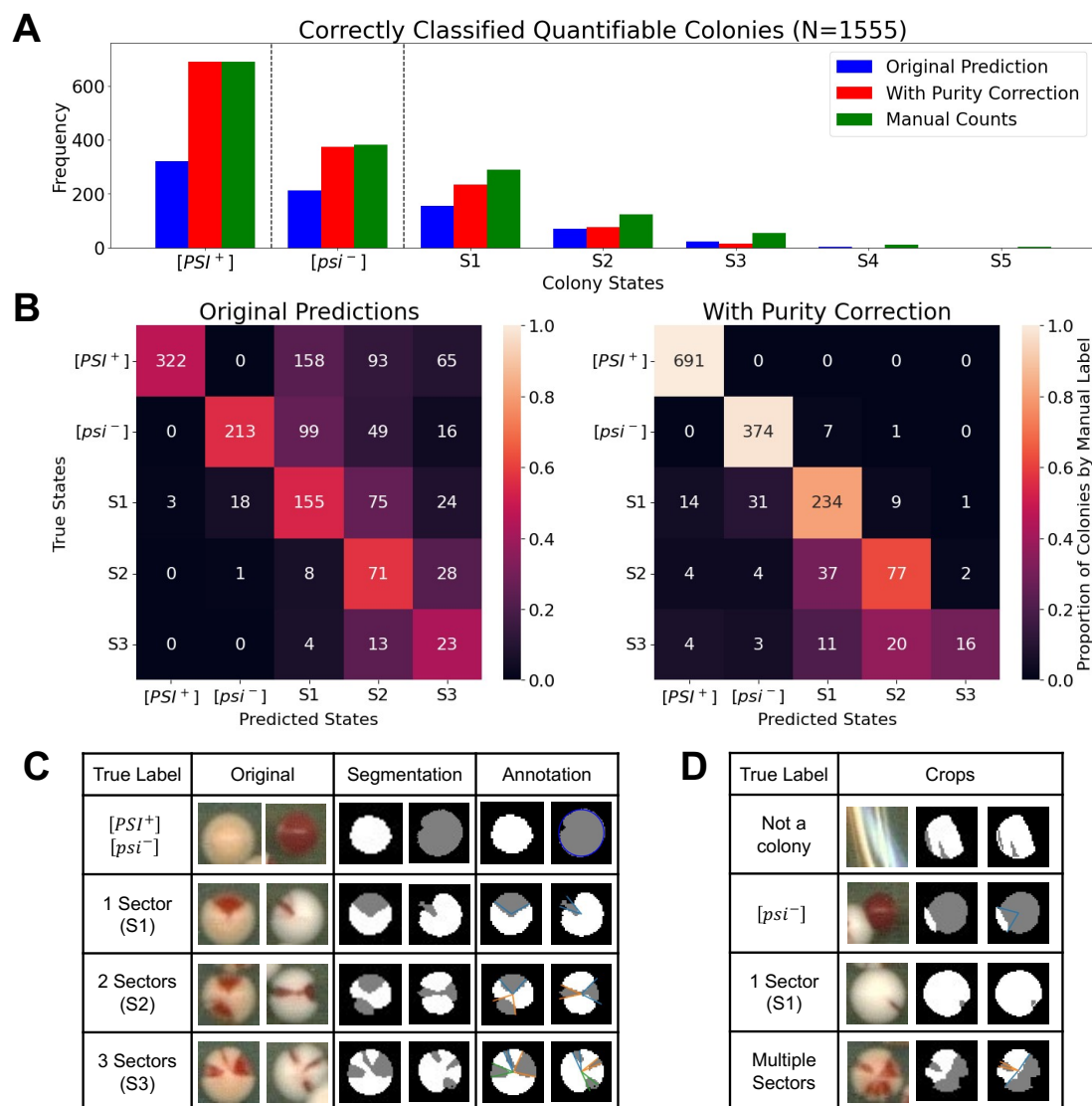


Figure 3.5: **Accuracy of colony-level predictions on quantifiable colony data.** (A): Total number of colonies of each class correctly classified. Blue and red bars indicate the number of colonies correctly classified without and with purity correction respectively. The height of the bars represent the number of colonies correctly classified, with the maximum number of colonies possible for each class indicated by the green bars. (B): Confusion matrices showing the frequency of correct and incorrect predictions with our pipeline without (left) and with (right) purity correction applied. The color of each cell indicates the percentage of colonies with the same ground-truth class that were assigned a predicted class through our pipeline. (C): Some examples of colony segmentations and annotations for [PSI^+], [psi^-] and sectored colonies along with the frequency of sectors (S1, S2, S3). (D): Some examples of segmentations and annotations for detected objects whose predictions were incorrect or which have no ground-truth class (either not quantifiable or not a colony).

into the correct groups in more detail across both image sets (Figure 3.5 B). We clearly see that including our purity correction scheme places more colonies on the main diagonal of the matrix. Surprisingly, all quantifiable colonies detected which were manually annotated as $[PSI^+]$ were correctly predicted to be $[PSI^+]$. This was not the case when purity correction was excluded. All but nine of the colonies manually annotated as $[psi^-]$ were correctly classified, with the incorrectly classified ones labeled as S1 or S2. When purity correction is not applied, this method significantly overestimates the number of sectored colonies whose manually annotated class is $[psi^-]$. For sectored colonies, our purity correction scheme shows improved accuracy in classifying colonies with one or two sectors, but slightly less accuracy in classifying colonies with three or more sectors. Fifteen colonies from those extracted were manually annotated as 4-sectored (S4) or 5-sectored (S5), but were not predicted with these classes.

Sectored colonies whose predicted class differs between those predicted without and with purity correction have their predicted frequency of sectors reduced as part of the correction scheme. This suggests our purity correction scheme is sufficiently preventing overcounting of the number of regions per colony in our dataset. Figure 3.5 C shows the segmentations and regional annotations of a few colonies which were correctly classified. Examples of colonies which were either non-quantifiable or incorrectly classified are shown in Figure 3.5 D.

The accuracy of colony classifications within each class is shown in Table 3.2. From the images we used in this study, we found that all quantifiable colonies extracted which were manually annotated as $[PSI^+]$ were correctly predicted as $[PSI^+]$, hence recall for this class was 1. The source of precision being less than 1 is due to a small number of 1-sector colonies being classified as $[PSI^+]$. Similarly, recall for $[psi^-]$ colonies was close to 1 due to some being incorrectly classified as sectored, and precision being less than 1 due to a subset of manually annotated sectored colonies being incorrectly classified as $[psi^-]$. Interestingly, while the accuracy in correctly predicting sectored quantifiable colonies is not as impressive, this category has the highest precision, indicating that the highest proportion of colonies predicted to be sectored were also manually annotated as sectored. However, when considering the frequency of sectors in these colonies, performance degrades with higher frequency of sectors as shown in the bottom half of Table 3.2. One possibility for lower prediction accuracy of multi-sector colonies may be due to both the smaller sizes of sectors as well as the inclusion of spacing between sectors (equivalently, the smaller sizes of white regions between sectors) negatively affecting the quality of the output segmentation of the colony, thus leading to unfavorable regions proposed in the classification step that do not accurately capture the true regions of the colony.

Examples of regional colony annotations before and after purity correction are shown in Figure 3.6 A. Many colony segmentations which had relatively small red or white regions did not meet the threshold for the purity metric and were thus not counted as separate regions. While the use of our purity correction scheme does alter the classifications of approximately half of colonies classified, our results show ap-

proximately 42% of all previously misclassified colonies became correct when purity correction was applied (Figure 3.6 B). In contrast, a subset of 55 colonies were classified incorrectly with purity correction when the original predictions were previously correct, yet the performance of our pipeline outweighs this disadvantage. In nearly all the colonies classified, the proposed regional annotations of the purity corrected colonies exhibit a higher weighted purity (Figure 3.6 C), as this was one of the objectives of our purity correction scheme as described in Section 3.2.1. Based on this information, our method is able to better capture sector-like regions in the colony segmentations which in turn improves accuracy in colony classification.

3.4 Discussion

Two aims of our pipeline for localizing colonies are to find all manually annotated colonies, and to suggest a way to classify colonies where manual annotations are not reliable. One objective [*PSI*]-CIC achieves is ensuring high recall, where as many quantifiable colonies from manually annotated data as possible are extracted, thus satisfying the first aim. Furthermore, we note that the precision for detecting quantifiable colonies is not very high (see Table 3.1). This is expected because nearly all colonies in the images have a sufficient degree of circularity, not just quantifiable ones. As a result, our method extracted and provided reasonable predictions for approximately 400 additional colonies from the images which were not considered quantifiable. As such, [*PSI*]-CIC could be used as an additional aid in quantifying colonies that are not considered quantifiable to experimentalists.

We observed a few major factors present in the colony images which had an influence on classification accuracy. First, we noticed that many colonies had at least one red or one white region comprising less than 5% of the colony area. As a result, our purity correction method in [*PSI*]-CIC did not accurately isolate these small regions. We believe this is likely a consequence of low image resolution. Near the center of the colony, it is possible for multiple sectors to occupy the same pixel, making it appear in the output segmentation that a sector may not originate at the colony center. Smaller regions in the segmentation may also not satisfy the threshold of the purity metric as defined in Section 3.2.1 and as such our method suggests such regions to be part of an adjacent region.

Second, there were also a subset of non-isolated quantifiable colonies whose individual colonies were classified. Due to our assumption that colonies are circular, adjacent clustered colonies may have overlapping regions present in each colony segmentation. Furthermore, clustered colonies were more likely to be excluded from classification since the circle detection step may have had insufficient information in these regions to detect circles there. Visual inspection also suggests that [*psi*⁻] regions in these colonies have a lower growth rate than [*PSI*⁺] regions, reducing circularity of the colony as a whole. However, this difference did not appear to have a significant effect on the number of colonies detected.

Third, individual colony sizes—or similarly, image sizes—may affect both segmen-

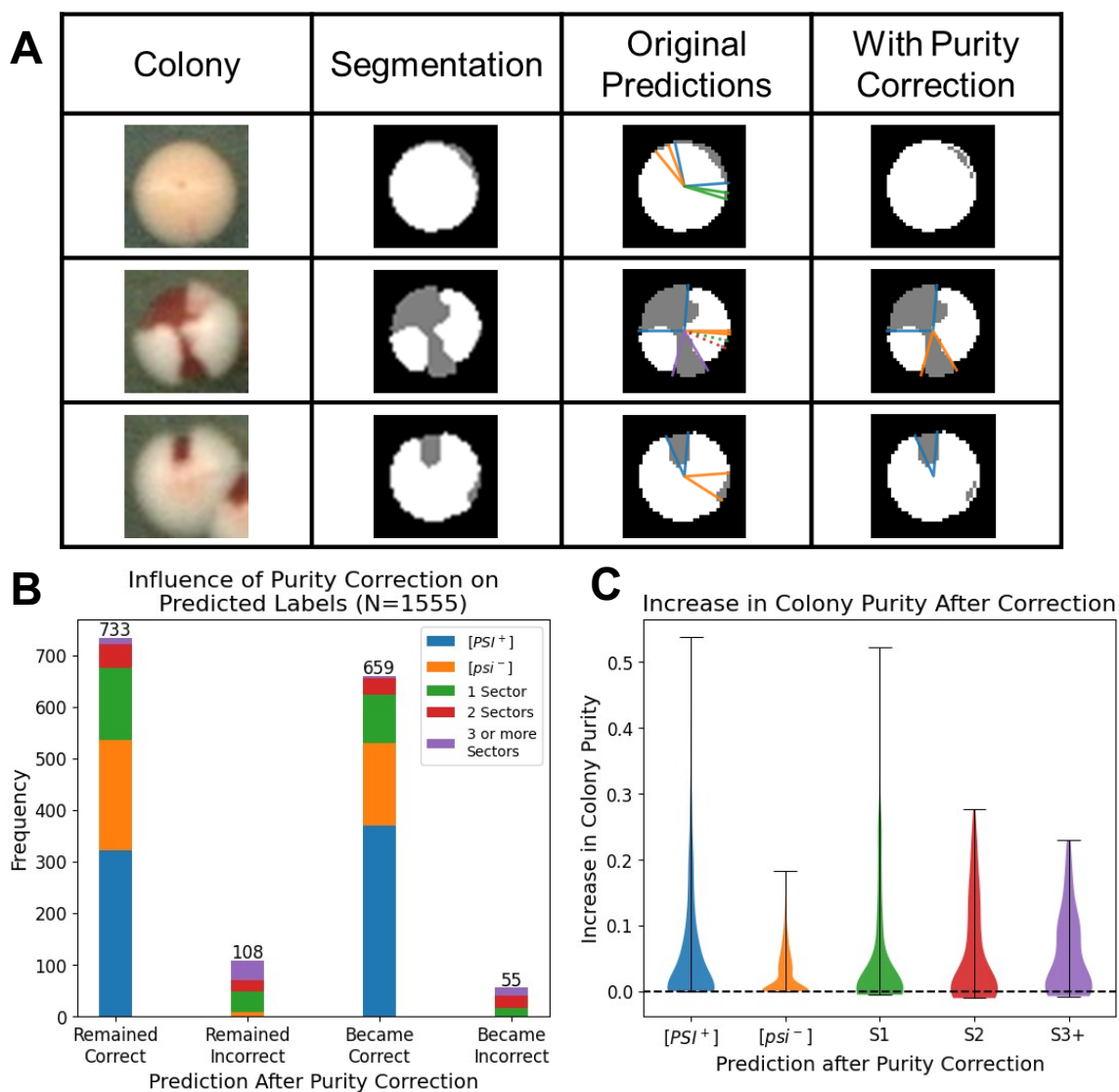


Figure 3.6: **Purity correction improves classification.** (A): Example colony segmentations with annotated regions before and after purity correction. (B): The predictions on the quantifiable colonies before and after purity correction, partitioned by their manually labeled states. “Remained Correct” is the set of colonies whose classifications both before and after purity correction matched their manually annotated classes. “Became Correct” is the set of colonies which were incorrectly classified before purity correction, but were correctly classified following purity correction. (C): Violin plots representing the distributions of differences between purities of colonies with and without purity correction, with positive differences indicating higher purity when correction is applied. Horizontal bars indicate the minimum and maximum differences for each subset of classified colonies based on their predicted state and sector frequencies.

tation and classification accuracy. Previous deep learning based segmentation tasks involving microbial colonies on plates used images with spatial dimensions in the couple thousands for each individual plate to adequately capture colonies within the full range of sizes present [25,52]. Capturing colony-level information from images of full plates would typically necessitate having high resolution images in order to ensure the individual colonies have a sufficient amount of resolution needed to exhibit clear sectors. However, the purity correction scheme in $[PSI]$ -CIC suggests that this size limitation need not be as strict if the objective is to partially capture estimate sectored regions rather than to fully segment them. Furthermore, *a priori* knowledge of colony sizes relative to the plate should still be used to impose a minimum size limitation for colony images to ensure a sufficient amount of detail is captured in the output of the model.

In contrast with the model proposed by Carl et al [25], $[PSI]$ -CIC relies on the use of synthetic images for training U-Net to segment real colonies rather than using real images directly in the training process. This is a convenient and reasonable strategy for simplifying ground-truth mask generation because colonies in our images appear circular and exhibit mostly geometric sector shapes by visual inspection. However, such a strategy makes it more difficult for semantic segmentation models to generalize to more complex image data. Despite this simplification, $[PSI]$ -CIC was still able to sufficiently locate, partition, and classify colonies in the experimental dataset. Further accuracy for classifying colonies using $[PSI]$ -CIC may be possible with using images of different colony sizes that still exhibit clear interfaces between the colony boundary as well as the interfaces between its red and white regions. One constraint to consider with using images of different sizes is that the colonies within the images must be large enough for detection. Rescaling larger images to a standard size will result in colonies being smaller; too small of a rescaling will result in colonies being too small to be detected computationally. Since the images used by Carl et al. [25] are more than 3000 pixels in both height and width dimensions—nearly three times higher than the images used in our study—and colony sizes much smaller in proportion to the plate sizes, such images may be too small to reliably segment and annotate if the plate images are resized to 1024×1024 . As such, a direct comparison of classification results between $[PSI]$ -CIC and that of Carl et al. [25] using their dataset and ours will not be feasible unless both models are capable of sufficiently classifying images of the same dimensions. Future work should address consistency of results with respect to image size and resolution to ensure a direct performance comparison could be made between $[PSI]$ -CIC and Carl et al. [25] as well as other similar image classification models which could be adapted for automated colony quantification.

The use of synthetic images for training CNNs is useful for improving image segmentation and classification when the quantity of annotated data is insufficient and the synthetic images capture sufficient variation present in the desired images to be segmented. While our synthetic images primarily capture the geometric features present in the experimental images, these features vary quantitatively across the experimental dataset. We point out three sources of variation which could be addressed

to boost complexity of the synthetic images. First, while the synthetic images account for most of the color variation present in image set 1 as described in Section A.1, they do not account for the color variation in image set 2 because the images in this set needed to be pre-processed before they were passed to U-Net for segmentation. An ideal sample of synthetic images should have similar color distributions as in the experimental images. Otherwise, U-Net would need to be independently retrained for each distinguishable set of images. The second source of variation consists of different colony sizes among the synthetic and experimental images. The colonies in our synthetic images have equal sizes, whereas the experimental images have a range of sizes. The third source of variation is the frequency of sectors present in each colony. While each of the synthetic images all contain colonies with exactly one sector each, sector sizes were allowed to vary between colonies. Even though our synthetic images do not fully capture all these sources of variation, we believe our approach is sufficient enough for using these images to train U-Net to segment experimental images. Our results (Section 3.3.2) show $[PSI]$ -CIC is adequately capturing sector-like regions in colonies and further classifying them as described in Section 3.2.1. We further note that while color and size are the two primary sources of variation in our images, other sources are possible. More structured and more diverse training data is needed to incorporate additional sources of variation present with experimental images and to ensure robust performance of $[PSI]$ -CIC across multiple experimental conditions.

We note that the primary features we considered when creating synthetic images for training U-Net involve circles and known colors from experimental images. As such, any other organism which exhibits these physical properties are prime candidates for automating colony classification. A natural extension of our work would be to adapt $[PSI]$ -CIC to classify colonies of *Candida albicans* which exhibit a white to opaque color switch [97, 140] as well as different colony size phenotypes under the same growth timeline [111]. Additional types of sectored image data at the colony level such as gene expression data obtained through fluorescent-based assays [63, 96] could be incorporated to develop methods for spatial structural analysis of such data. These considerations warrant a further generalizability study on the usefulness of $[PSI]$ -CIC in segmenting images containing other species of yeast or other circular shaped colonies as a future research direction.

3.5 Conclusion

In this study, we constructed a new computational pipeline we call $[PSI]$ -CIC designed for high-throughput segmentation and quantification of sectored yeast colonies found in images of experimental plates. We show that synthetic images could be used for training U-Net to segment colonies from experimental images based on their color and simple shape. Results show that we are able to obtain acceptable colony counts from plated colony images, given that the segmentation adequately captures the circularity and regions of the colony. We demonstrate that $[PSI]$ -CIC obtains colony states and sector frequencies comparable to manual annotations from experimen-

talists. This is the first model designed specifically for quantifying sectors in yeast colonies indicative of changes in prion dynamics within individual cells. The work discussed here is a big step forward for providing researchers a computational framework to gain novel insights into the mechanisms driving prion loss in yeast colonies.

Chapter 4

A Deep Learning Framework for *Candida albicans* Colony Classification

This chapter covers work in progress in collaboration with Dr. Clarissa Nobile and Austin Perry at UC Merced. I led the work on developing the computational pipeline, analyzing its results within this chapter. Austin Perry led the curation of the image data that was used in this work as well as provided a portion of the final version of the Introduction (Section 4.1). More details about the data itself is provided in Appendix B.

4.1 Introduction

Candida albicans is one of the most frequently encountered and studied human fungal pathogens [135], however it generally resides as a commensal organism in humans [69]. As a commensal organism it can be isolated from regions all over the body, including the skin and gastrointestinal tract [113, 118]. However, a delicate balance with the rest of the microbiota is needed to remain commensal [61]; if this balance is disturbed, *C. albicans* is capable of rapid proliferation which leads to infections [118]. *C. albicans* is responsible for at least 70% of fungal infections worldwide with a mortality rate of nearly 40% for especially serious cases [118, 157].

The ability of *C. albicans* to colonize such distinct niches is due in part to its propensity to exist as various cell types. Two such cell types are termed “white” and “opaque”, each with distinct cell shapes, colony morphologies and environmental responses [97, 160, 181]. Each cell type is generally stable under standard *in vitro* conditions, however cells will stochastically switch from one cell type to the other approximately once every 10^4 cell divisions [153]. To investigate this system, researchers rely on large-scale, low-throughput “switch assays”. These usually involve a kind of solid chromogenic media that differentially colors the white and opaque colonies. CHROMagar [121] is considered the first commercially available type of chromogenic

media aimed at isolating and identifying several different types of *Candida* species. Researchers will dilute cultures and spread plate their cell solutions on petri dishes of this chromatic media and then manually count the numbers of white, opaque and sectoried colonies, including colonies that experienced a switch event after the ancestral cell landed on the plate. This approach is very labor intensive and requires a great amount of time to accurately quantify. To decrease the amount of time it takes to count each of these colony types, I have developed a deep learning framework to efficiently quantify colonies of *C. albicans* with distinct colors in the CHROMagar media for each colony type.

In this Chapter I will demonstrate that the proposed deep learning framework results in efficient quantification of white and opaque colony images with high accuracy. I then discuss future directions for this framework in terms of classifying other colonies that show different features than the ones used in this chapter and how we can include these features in the training process.

4.2 Methods

In this section, I describe the components of the proposed algorithm for quantifying images of plates containing hundred of colonies (Section 4.2.1). The procedure for the cultivation of colonies, acquisition of the image data, and augmentation strategy applied to the image data, are provided in Appendix B.1. In Section 4.2.5 we discuss how we annotate the image data available so that a supervised neural network is applicable for this data. Due to a limited amount of annotated data available, we employ data augmentation to increase the size of our dataset (see Section B.2). We discuss how the neural networks used in our algorithm are configured and how they are evaluated in Section 4.2.6.

4.2.1 Candida Classification Pipeline

The proposed pipeline in this section aims to classify colony types from a set of images depicting hundreds of colonies on individual plates. Our algorithm is configured using 15 images containing individual plates that each have anywhere between 50 and 300 colonies each (see Appendix B.1 for details). To get images of colonies from the images of plates, we partition the algorithm into the extraction phase, followed by the classification phase. In the extraction phase, we attempt to individually extract and gather the colonies from each plate and save them as individual images. We employ a circular feature detection algorithm as our method of choice in obtaining the locations and sizes of colonies as described in Section 4.2.2. In the classification phase, we employ deep learning architectures trained on the extracted images in order to automatically and efficiently annotate and classify these colonies. Annotated images as described in Section 4.2.5 are used as part of preparing the architectures for the training process. Configuration of the training process, including how the images are partitioned into training and testing, is described in Section 4.2.6. Due to the lack

of data available we also employ data augmentation to balance the dataset whose process is described in Section B.2. After training, our proposed pipeline classifying colonies from plate level images is fully automated so that any similar image of plates will have their colonies extracted and classified efficiently.

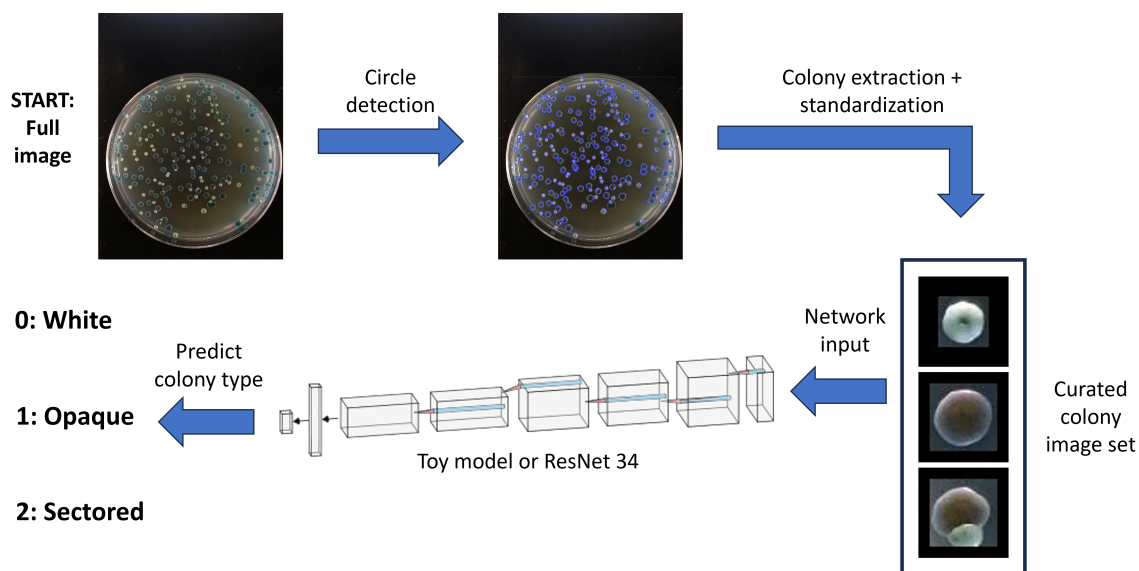


Figure 4.1: **Candida Detection Pipeline.** The pipeline is divided into two phases. In the first phase, the objective is to separate the colonies from the image. To do this, we employ circle detection to find the plate, then the colonies respectively. Each colony extracted is then standardized and aggregated into a data set for the second phase. The objective of the second phase is to predict a qualitative label for each colony extracted. We do this using deep neural network architectures with the objective of performing image classification. The output of the deep neural network allows for a prediction of a colony’s type.

4.2.2 Colony Detection

Plate Detection and Extraction

To obtain information on the colonies from the experimental images in our dataset, we utilize the circle Hough transform implementation in Octave via the function `imfindcircles` found within the `image` package. Throughout this process of extracting colonies suitable for classification in this work, `imfindcircles` is used independently for the purposes of plate extraction and subsequently, colony extraction.

The first use of `imfindcircles` is done to isolate the plate from the raw image. We begin by proportionately rescaling each image so that the longest dimension of the image is 1200 pixels. We use an input radius range of $[450, 650]$ pixels and a sensitivity of 0.97 to allow for imperfect circles to be detected. In each image, exactly one circle is detected corresponding to the border of the plate as desired. The region

of the plate to be extracted from the image is estimated by encompassing the detected circular region with the smallest square bounding box around it. The region inside the bounding box is then resized with dimensions 1024x1024 and saved as a separate image. This process is repeated for all 15 images in our dataset.

Colony Detection and Extraction

Once the 15 images have been rescaled as described above, the second use of `imfindcircles` is applied to the rescaled images for the purpose of detecting the circular objects within each image. Here we use an input radius range of [10, 50] pixels to allow for the small white and large opaque colonies to be detected, and set the sensitivity parameter to 0.9. Due to the varying colony sizes in the images, the radius range and sensitivity parameter were chosen based on trial and error.

Furthermore, since white and opaque colonies have opposite polarities with the intensity of the region inside the plate, we configure `imfindcircles` to locate colonies whose intensity is either less than or greater than the intensity of the background, ensuring circular objects that are either brighter or darker than the background are detectable. To do this, we first use `imfindcircles` with the object polarity parameter set to ‘bright’, then use `imfindcircles` with this parameter set to ‘dark’. To remove any duplicated detections, we compute the minimum distance of the center of a given dark polarity detection with the centers of all bright polarity detections, then filter out any dark polarity detections that has a minimum distance of less than 10 pixels from the center of any bright polarity detection.

A bounding box is independently applied to all detected regions in a manner similar to how the plate was extracted. The radius of the enclosed circle is recorded before extraction and saved as a CSV file, and all detected regions in the plate are cropped and saved as separate images.

To preserve spatial information on the size of each colony, instead of proportionately rescaling the image to size 64x64, we instead pad the image with black pixels until the image size of 64x64 is attained. This allows us to utilize size information as a component in the dual-input neural networks described in the next section.

Size Normalization

Using the size information of each plate and colony extracted from each image, we define a proportional colony size $c_{i,j}$ relative to a plate:

$$c_{i,j} = \frac{r_{i,j}}{R_j} \quad (4.1)$$

where $r_{i,j}$ is the estimated radius of colony i on plate j , and R_j is the radius of plate j . The value of $c_{i,j}$ represents the proportion of the colony radius to the plate radius, defining a non-dimensional measure describing the size of a colony. This measurement however is local with respect to the plate at which the colony resides.

To obtain a global measure of colony sizes across all plates, we attempt to normalize this measure. Specifically, we normalize this metric by defining another metric $d_{i,j}$:

$$d_{i,j} = \frac{c_{i,j}}{\max_{i,j} c_{i,j}}. \quad (4.2)$$

where $d_{i,j}$ is a non-dimensional measure between a given colony i from plate j and the largest colony in the dataset. $d_{i,j}$ takes values between 0 and 1, where 1 corresponds to the largest colony in the dataset. Figure 4.2 shows the distributions of the normalized colony sizes for each of the colonies extracted from our image set. These normalized size measures are used as a secondary input for the second phase of our proposed pipeline (see Figure 4.1).

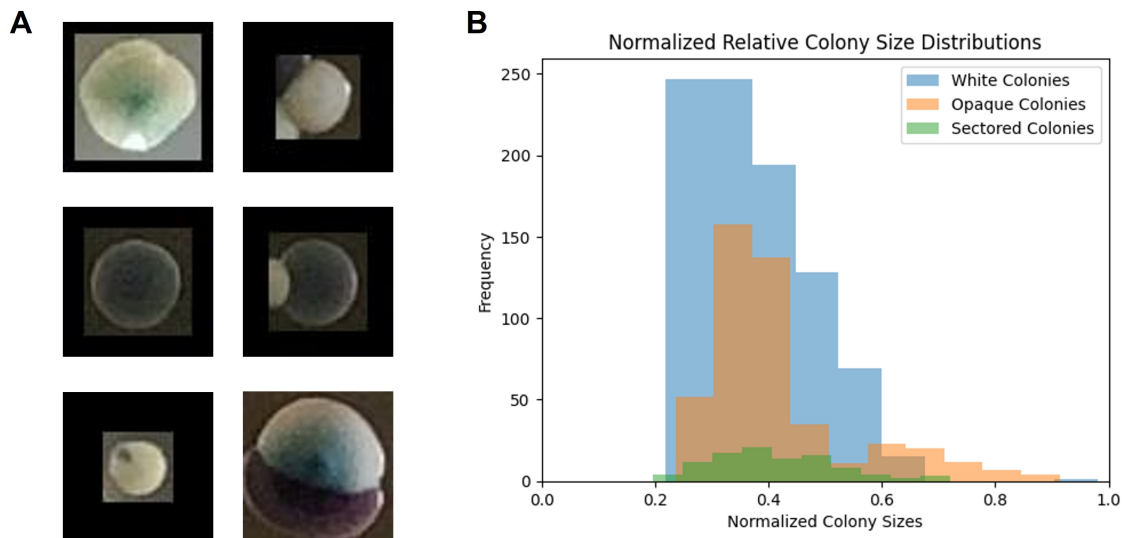


Figure 4.2: **Distributions of colony sizes.** Left: Examples of each colony type extracted from the image set each resized to 64x64 through padding. From top to bottom, two examples of white colonies, opaque colonies, and sectored colonies. Colony sizes are estimated by taking the proportion of the colony and plate radii, and normalized by dividing by the maximum proportion across the entire image set (see Section 4.2.2). Right: Histograms of the normalized size measures for the 1,561 colony objects in our dataset. Each color represents one histogram of normalized colony sizes for a specific colony type (blue: white colonies, orange, opaque colonies, green: sectored colonies). White colonies are the most abundant colony type in our dataset and have relative sizes between 0.2 and 0.75 with an outlier at 1 serving as the limiting factor in the normalization. Opaque colonies are the largest colonies in the image set on average, but have the highest amount of variation in normalized sizes. Sectored colonies are not as numerous—hence the smaller histogram—but have a normalized size range from approximately 0.2 to 0.75, a range of relative sizes similar to white colonies.

4.2.3 Neural Network Architectures

Here we detail the construction of two deep neural network architectures we use to predict qualitative labels for each colony image extracted. The first architecture contains 3 Visual Geometry Group (VGG) blocks with fully connected layers at the end. The second architecture uses a modified ResNet34 [66]. To integrate information about the size of each colony into the architecture, we also propose modifications of both architectures to allow a secondary input. We call these modified architectures dual-input architectures throughout this chapter.

Single Input Architectures

The first architecture is a simpler model composed of three VGG blocks in sequence which we call the “Toy Model” (Figure 4.3 A (top)). Each VGG block contains two convolutional layers with kernel size 3x3 and a ReLU activation function, a max-pooling layer with kernel size 2x2, and a dropout layer with probability 0.2. The three sets of convolutional layers contain 32, 64, and 128 output channels respectively. After the third VGG block, the output from the last VGG block is vectorized and used as input for a Dense nonlinear layer with 128 output units. Finally, the result is passed through one more Dense nonlinear layer where the number of output channels is equal to the number of desired classes, which for our dataset is 4. The full architecture for this model is shown in Figure 4.3 B (top).

The second architecture is a modification of the original ResNet34 architecture [66]. In a similar manner, we define a residual block (see Figure 4.3 A (bottom)) as a sequence of layers in the following order: One convolutional layer with kernel size 3x3 and stride 2x2, one batch normalization layer, one ReLU activation function, a second convolutional layer with kernel size 3x3 and stride 1x1, one batch normalization layer, one addition layer where the result of the batch normalization layer is added to the input of the residual block, and one ReLU activation function. If the input and output sizes of the residual block are different, the input of the residual block in parallel is used as input to an additional convolutional layer with kernel size 1x1 which projects the input down to the desired dimension, followed by a batch normalization layer before being used as input to the addition layer. The output of the last residual block is then vectorized and used as input to one last Dense layer with 4 output classes. The full architecture for this model is shown in Figure 4.3 C (top).

Dual Input Architectures

To account for size variability within each colony type, we modified both of the architectures described previously to include our global non-dimensional size measure as a secondary input (see Section 4.2.2). To accommodate this additional input, we modify the single input architectures in the following way. First, we modify the corresponding single input architectures by concatenating four Dense nonlinear layers after the vectorization step (the Flatten layers), each subsequent one having fewer and

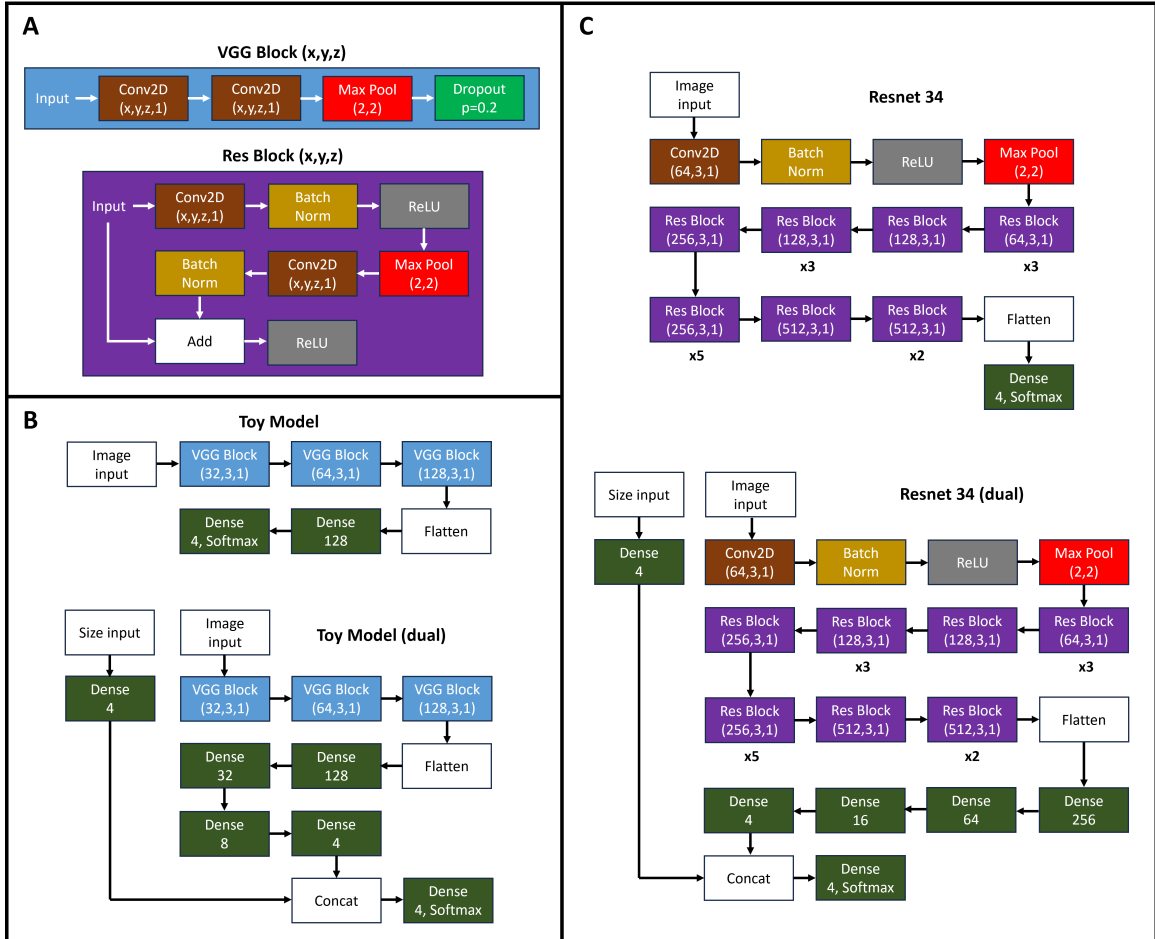


Figure 4.3: **Neural Network Architectures.** The architectures used in this work are singular and dual input modifications of deep neural networks. A: Shorthand definitions of a VGG Block and a Res Block (residual block) (see Section 4.2.3), where x is the number of output channels, y is the kernel size, and z is the stride length. B: The singular and dual input architectures of the toy model. C: The singular and dual input architectures of the modified Resnet 34. The bottom portions of B and C contain additional Dense layers for the images to pass through and a secondary input for size which passes through a Dense layer before merging with the output from the image layers near the end of the network.

fewer output layers until the last one has four output layers. Second, we introduce a parallel pathway exclusively for the size input which contains one Dense nonlinear layer with four output channels. Third, the output of this Dense nonlinear layer is concatenated to the result of the last Dense nonlinear layer from the image pathway. Finally, this output is passed through one Dense nonlinear layer with the number of desired classes as the number of output channels, which is 4 for our dataset. This construction aims to ensure image information passed through the network does not significantly overshadow size information.

We modify the architecture of our Toy Model in the following way: First, we introduce the size pathway by applying a Dense nonlinear layer with 4 output units to the size input. Second, additional Dense nonlinear layers with output units of 32, 8, and 4 respectively, are imposed on the output of the 128 unit Dense layer within the image pathway. Third, the outputs of the two Dense layers with four output units corresponding to the image and size respectively are concatenated to form a vector of length 8. Finally, this is used as input to a Dense nonlinear layer with four output units (corresponding to the number of desired classes) and a softmax activation function. The full architecture for this model is shown in Figure 4.3 B (bottom).

We modify the ResNet 34 architecture in the following way: First, we introduce the size pathway by applying a Dense nonlinear layer with 4 output units to the size input. Second, additional Dense nonlinear layers with output units of 256, 64, 16, and 4 respectively, are imposed after the output of the flatten operation within the image pathway. Third, the outputs of the two Dense layers with four output units corresponding to the image and size respectively are concatenated to form a vector of length 8. Finally, this is used as input to a Dense nonlinear layer with four output units (corresponding to the number of desired classes) and a softmax activation function. The full architecture for this model is shown in Figure 4.3 C (bottom).

4.2.4 Label Prediction

The output of all four neural networks when applied to a colony image is a probability vector of length 4. Each element indicates the probability that the image is of a particular class given the data that the model was trained on. The classes we consider are for white colonies, opaque colonies, sectored colonies, and background. The length corresponds to the number of desired classes, and each element in the vector indicates the probability that the image is of a particular class given the data, with the sum of all probabilities adding to 1. The predicted label of the colony is chosen based on the position of the maximal probability in the vector. When applied to a collection of colonies, the output for the collection of colonies is a vector of labels corresponding independently to the maximal probabilities of each type associated with a colony image.

4.2.5 Ground Truth Annotation

Each of the 15 plate images are manually annotated by superimposing colored dots on top of each colony present. The dots were placed on the approximate centers of the colonies. The color of the dot is chosen based on the colony type: cyan for white colonies, magenta for opaque colonies, and yellow for sectored colonies (see Figure 4.4 for an example). Images extracted from the plates that do not contain colonies are left un-annotated and are in turn considered as part of the background class.

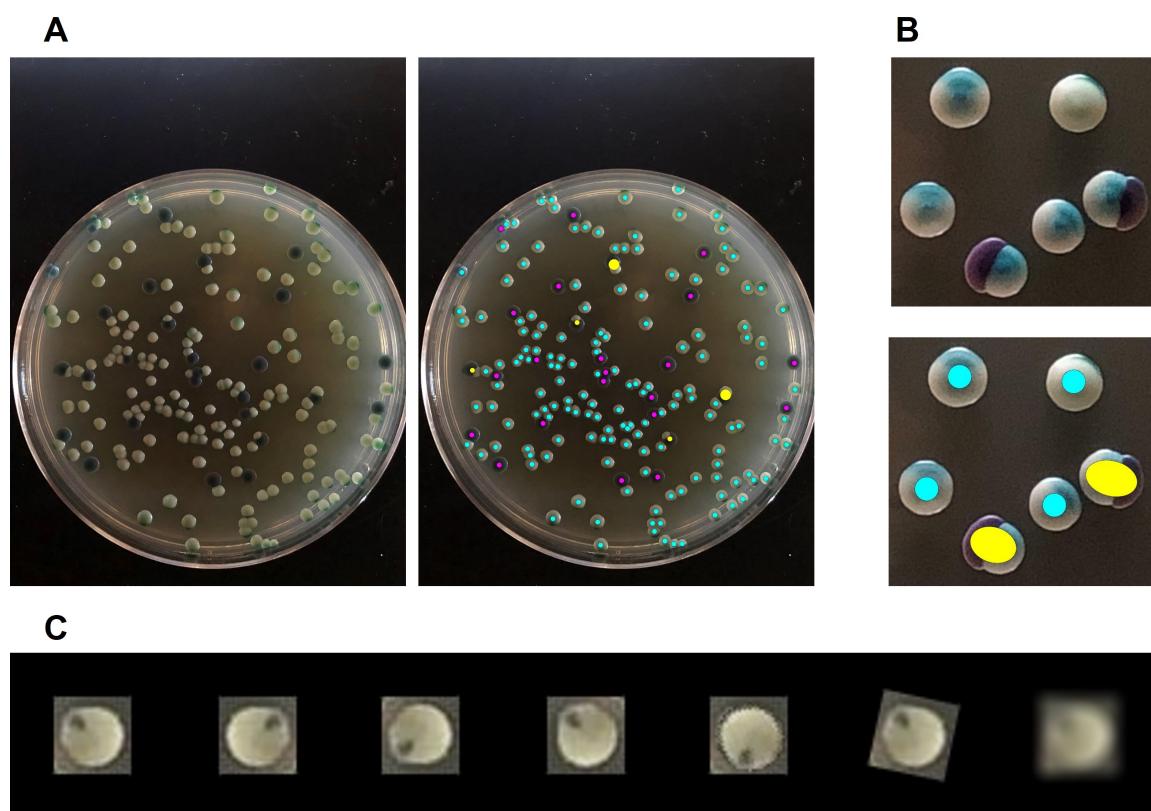


Figure 4.4: **Preprocessing images for training.** A: An image of a full plate with its corresponding superimposed annotations. The color of the dots represent the corresponding colony type (cyan: white colony, magenta: opaque colony, yellow: sectored colony). B: A closeup image of several colonies along with its corresponding annotated version. C: Augmented versions of an example colony extracted. Transformations applied to the colony, from left to right, are the following: original, horizontal flip, vertical flip, transposition, rotation of colony region, random rotation of the whole image by up to 90 degrees from the original orientation, and blurring.

To preserve the location of the dots in each image, we saved three images of the same size as the image being annotated, where each of the three images contain only the dots corresponding to one independent colony type. An annotated copy for the image is created by directly superimposing the three images on top of the original

Table 4.1: **Counts of extracted images from each plate.** Table of the number of objects detected and objects separated by type. The first two images corresponding to each strain are used for training purposes, while the third images of each are set aside for testing. The number of objects extracted from each image are shown in the left table. The total number of objects of each type extracted from all the training and testing images respectively are shown on the right table.

Plate Data	# of Detected Objects Extracted			# of Detected Objects by Type		
	Training Set		Testing Set			
	Rep. 1	Rep. 2	Rep. 3	Type	Training	Testing
AHY_683 Set 1	69	60	79	White	583	319
AHY_683 Set 2	213	224	220	Opaque	321	138
AHY_838	62	34	55	Sectored	58	43
APY_001	63	78	88	Background	144	88
SN_250	160	143	146	Total	1106	588
Total Objects:	1106		588			

plate image of the same size and orientation.

4.2.6 Training

From the 15 images available, we use 10 for the purpose of training each of our architectures. The remaining 5 images are set aside for testing the performance of each architecture post-training (see Figure 4.1). We use the first two images corresponding to each of the five strains to be used in the training process and set aside the third image of each strain for testing each trained model. The training images are subject to colony detection, extraction, and image augmentation (see Section B.2), while the testing images are subject to only colony detection and extraction. Examples of colony augmentation are shown in Figure 4.4 C.

Approximately 20% of the colonies extracted from the training images are set aside for model validation. For each architecture, we use the Adam optimization function with momentum 0.9 and categorical focal cross-entropy [95] as our loss function with the weight-balancing factor (α) set to 0.25 and the focusing parameter (γ) set to 2. The learning rate was fixed to 0.001 for the entire training process. Each architecture was trained for 100 epochs and intermittently evaluated on the testing images to measure performance.

The true labels of the colonies are compared with the same predicted labels on these colonies. We measure performance by taking the proportion of colonies whose predicted labels match their annotated true labels, divided by the total number of colonies considered. We further quantify the proficiency and deficiency of each architecture on images of each colony type independently.

4.3 Results

Approximately 587 objects were extracted from the testing set of 5 plates. We evaluate model performance for eight different model scenarios defined by considering all combinations of the two types of architectures (toy model or Resnet 34), two input types (single and dual), and two augmentation scenarios (no augmentation or with augmentation).

We find that in all eight cases, the accuracy in classifying white and opaque colonies is between 80-98%. Sectored colonies however are the colony type that each model struggled with, but has the greatest variability in accuracy. All eight models correctly predict between 0-56% of sectored colonies, with those incorrectly classified being predicted as either white or opaque colonies. Figure 4.5 shows an example of how colonies were classified for a single instance of a trained single input Resnet 34 model.

Precision-Recall and Receiver Operating Characteristic (ROC) curves are plotted for each colony class and the area under the curve (AUC) metric is computed for both types of curves in order to evaluate model performance for each model on the colony classes. In each case, precision-recall scores for each colony type followed a similar pattern with that of the confusion matrices. Interestingly, ROC curves depict an opposite effect for sectored colonies, showing ROC-AUC scores of at least 0.69 across all eight models. While each model may be inaccurately predicting sectored colonies, each model still achieves better performance than a truly random classifier.

Finally, we show that similar behavior is present when multiple instances of the same model are trained on the same data with only the exception of the dual input toy model.

4.3.1 Single Input Architectures: Quantitative Performance

Toy Model

No augmentation. An accuracy of 85% across the entire test image set is obtained. The per-class accuracies are 95% for white colonies, 97% for opaque colonies, 9% for sectored colonies, and 67% for background respectively (Figure 4.6 (top)). (Precision, Recall) scores for each class are (0.95, 0.90) for white colonies, (0.97, 0.77) for opaque colonies, (0.09, 0.31) for sectored colonies, and (0.67, 0.95) for background respectively. F1 scores for each class are 0.92 for white colonies, 0.86 for opaque colonies, 0.14 for sectored colonies, and 0.79 for background respectively.

With augmentation. An accuracy of 83% across the entire test image set is obtained. The per-class accuracies are 85% for white colonies, 99% for opaque colonies, 26% for sectored colonies, and 76% for background respectively (Figure 4.6 (bottom)). (Precision, Recall) scores for each class are (0.85, 0.94) for white colonies, (0.99, 0.76) for opaque colonies, (0.26, 0.24) for sectored colonies, and (0.76, 0.88) for background respectively. F1 scores for each class are 0.89 for white colonies, 0.86 for opaque colonies, 0.25 for sectored colonies, and 0.82 for background respectively.

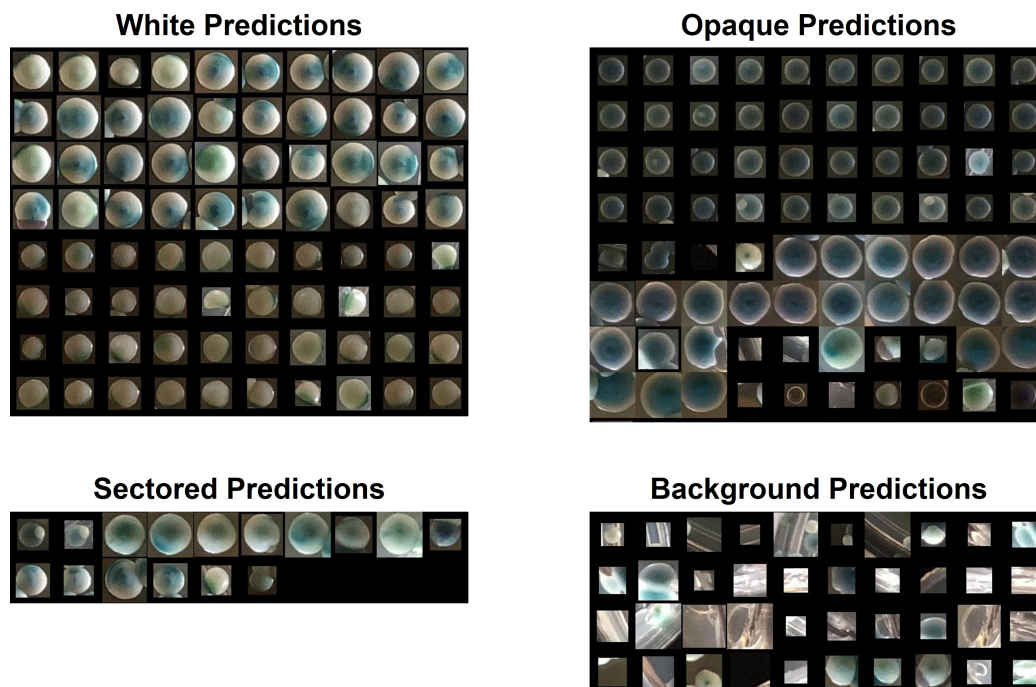


Figure 4.5: **Example colony predictions.** Concatenations of colonies from the test image set predicted by the single input Resnet 34 model to be white, opaque, sectored, or background noise.

Resnet 34

No augmentation. An accuracy of 84% across the entire test image set is obtained. The per-class accuracies are 93% for white colonies, 96% for opaque colonies, 14% for sectored colonies, and 67% for background respectively (Figure 4.7 (top)). (Precision, Recall) scores for each class are (0.93, 0.91) for white colonies, (0.96, 0.80) for opaque colonies, (0.14, 0.30) for sectored colonies, and (0.67, 0.77) for background respectively. F1 scores for each class are 0.92 for white colonies, 0.87 for opaque colonies, 0.19 for sectored colonies, and 0.72 for background respectively.

With augmentation. An accuracy of 81% across the entire test image set is obtained. The per-class accuracies are 80% for white colonies, 96% for opaque colonies, 42% for sectored colonies, and 81% for background respectively (Figure 4.7 (bottom)). (Precision, Recall) scores for each class are (0.80, 0.95) for white colonies, (0.96, 0.86) for opaque colonies, (0.42, 0.26) for sectored colonies, and (0.81, 0.74) for background respectively. F1 scores for each class are 0.87 for white colonies, 0.90 for opaque colonies, 0.32 for sectored colonies, and 0.77 for background respectively.

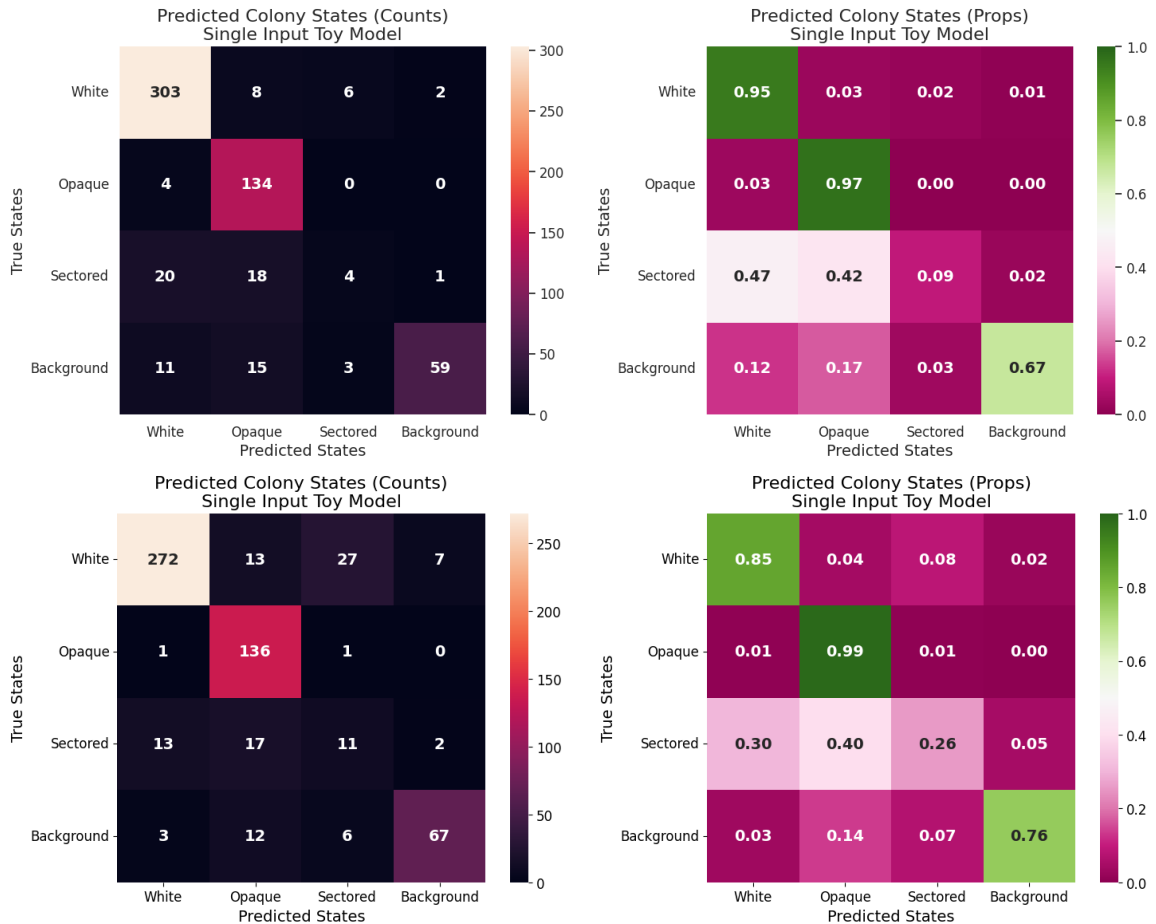


Figure 4.6: **Confusion Matrices for the Single Input Toy Model.** Confusion matrices showing the proportion of colonies classified normalized by row, with true labels on the vertical axis, and predicted labels on the horizontal axis. Each cell is assigned a proportion of colonies predicted to have a certain label, given their assigned true label. Confusion matrices above are for the performance on the single-input toy model with no augmentation applied (top) and with augmentation applied (bottom) to the training images.

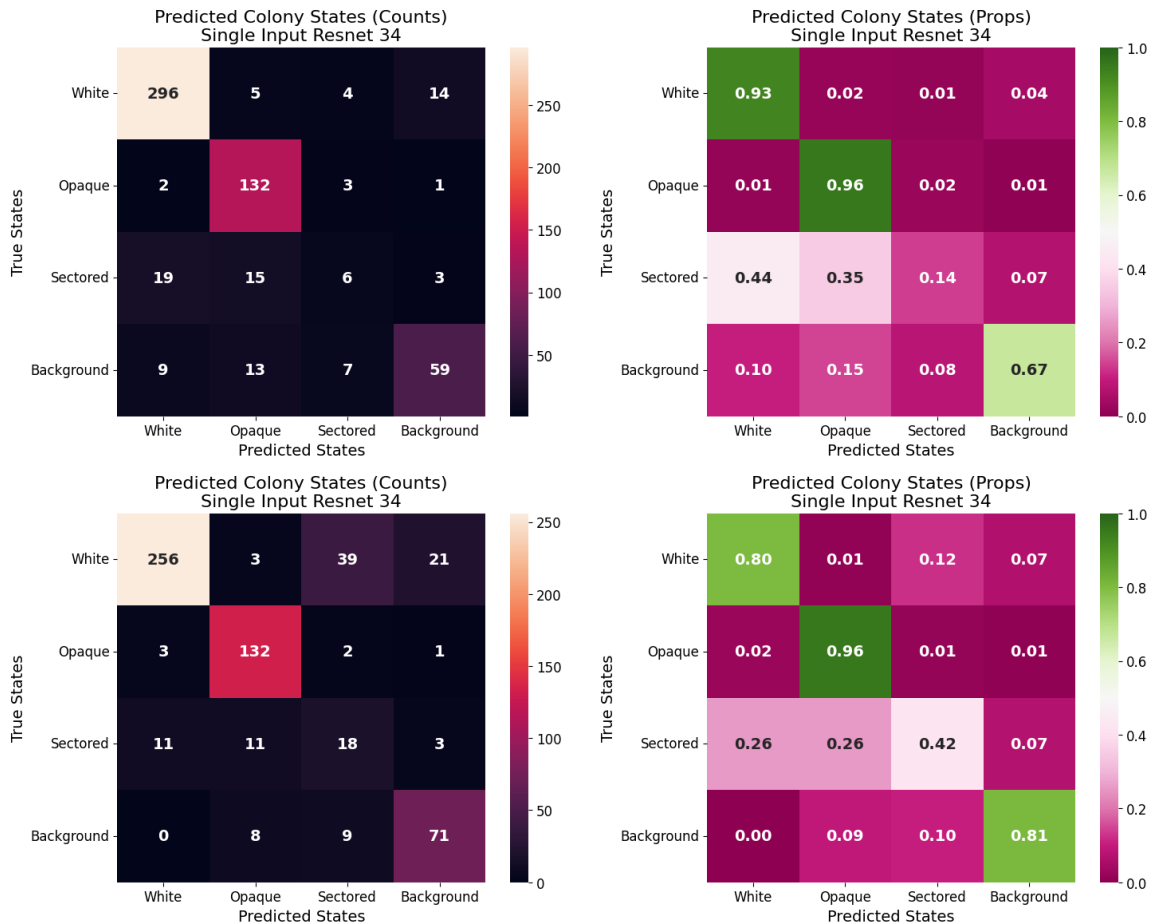


Figure 4.7: **Confusion Matrices for the Single Input Resnet 34 Model.** Confusion matrices showing the proportion of colonies classified normalized by row, with true labels on the vertical axis, and predicted labels on the horizontal axis. Each cell is assigned a proportion of colonies predicted to have a certain label, given their assigned true label. Confusion matrices above are for the performance on the single-input Resnet 34 model with no augmentation applied (top) and with augmentation applied (bottom) to the training images.

4.3.2 Single Input Architectures: Qualitative Performance

Toy Model

No augmentation. Precision-Recall AUC scores for each class are 0.94 for white colonies, 0.90 for opaque colonies, 0.15 for sectored colonies, and 0.77 for background respectively. ROC-AUC scores for each class are 0.94 for white colonies, 0.97 for opaque colonies, 0.69 for sectored colonies, and 0.91 for background respectively. (Figure 4.8 (top))

With augmentation. Precision-Recall AUC scores for each class are 0.96 for white colonies, 0.95 for opaque colonies, 0.19 for sectored colonies, and 0.80 for background respectively. ROC-AUC scores for each class are 0.95 for white colonies, 0.98 for opaque colonies, 0.82 for sectored colonies, and 0.91 for background respectively. (Figure 4.8 (bottom))

Resnet 34

No augmentation. Precision-Recall AUC scores for each class are 0.95 for white colonies, 0.94 for opaque colonies, 0.18 for sectored colonies, and 0.77 for background respectively. ROC-AUC scores for each class are 0.96 for white colonies, 0.98 for opaque colonies, 0.72 for sectored colonies, and 0.94 for background respectively. (Figure 4.9 (top))

With augmentation. Precision-Recall AUC scores for each class are 0.94 for white colonies, 0.93 for opaque colonies, 0.21 for sectored colonies, and 0.79 for background respectively. ROC-AUC scores for each class are 0.93 for white colonies, 0.98 for opaque colonies, 0.77 for sectored colonies, and 0.93 for background respectively. (Figure 4.9 (bottom))

4.3.3 Dual Input Architectures: Quantitative Performance

Toy Model

No augmentation. An accuracy of 84% across the entire test image set is obtained. The per-class accuracies are 94% for white colonies, 99% for opaque colonies, 0% for sectored colonies, and 67% for background respectively (Figure 4.10 (top)). (Precision, Recall) scores for each class are (0.94, 0.91) for white colonies, (0.99, 0.72) for opaque colonies, (0.00, 0.00) for sectored colonies, and (0.67, 0.87) for background respectively. F1 scores for each class are 0.92 for white colonies, 0.83 for opaque colonies, 0.00 for sectored colonies, and 0.76 for background respectively.

With augmentation. An accuracy of 73% across the entire test image set is obtained. The per-class accuracies are 66% for white colonies, 96% for opaque colonies, 56% for sectored colonies, and 69% for background respectively (Figure 4.10 (bottom)). (Precision, Recall) scores for each class are (0.66, 0.97) for white colonies, (0.96, 0.83) for opaque colonies, (0.56, 0.16) for sectored colonies, and (0.69, 0.94) for background respectively. F1 scores for each class are 0.78 for white colonies, 0.89

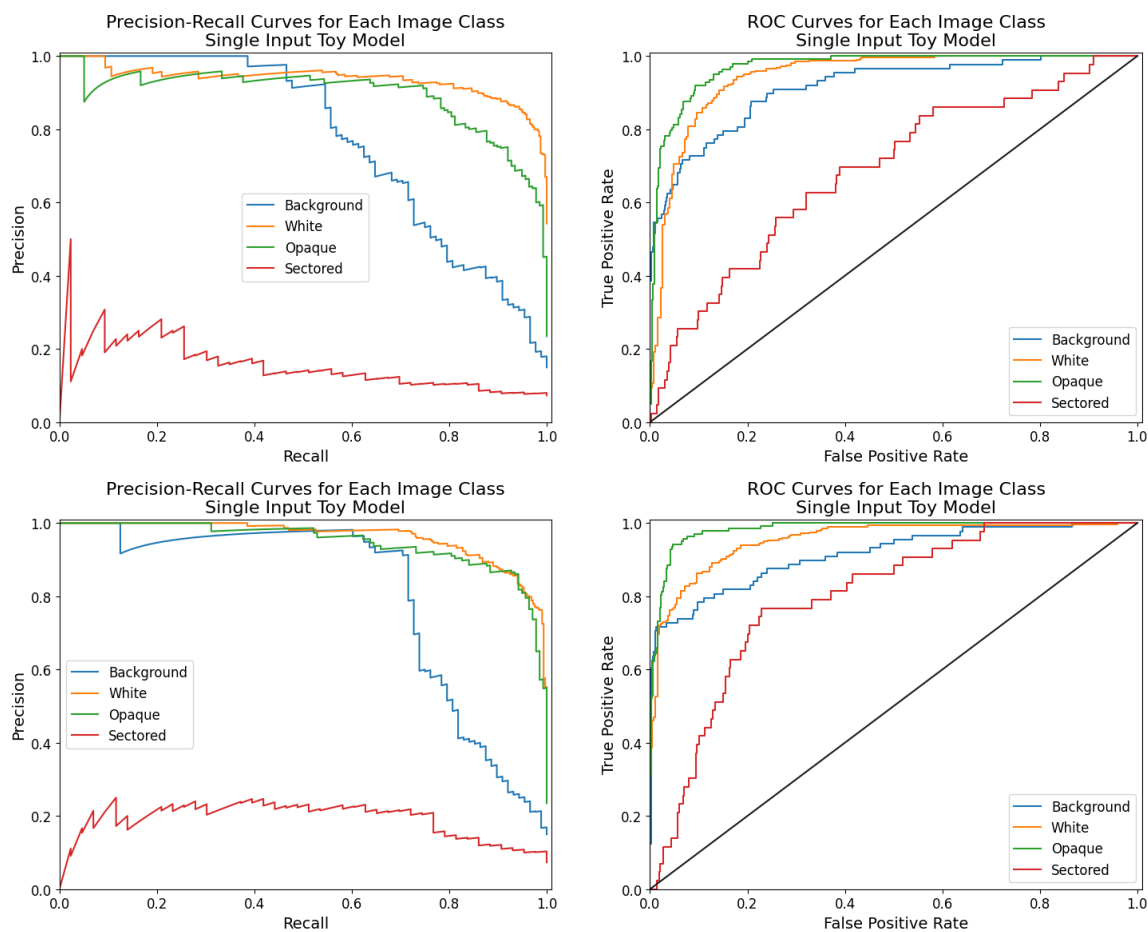


Figure 4.8: **Precision-Recall and ROC curves for the Single Input toy model.** Left: Precision-Recall curves showing performance of the single input toy model on each image type. Right: ROC Curves showing qualitative performance on each colony type compared to a random classifier (black line). The top row indicates performance curves for the single input toy model applied to the original dataset, and the bottom row indicates performance curves for the single input toy model applied to the augmented dataset.

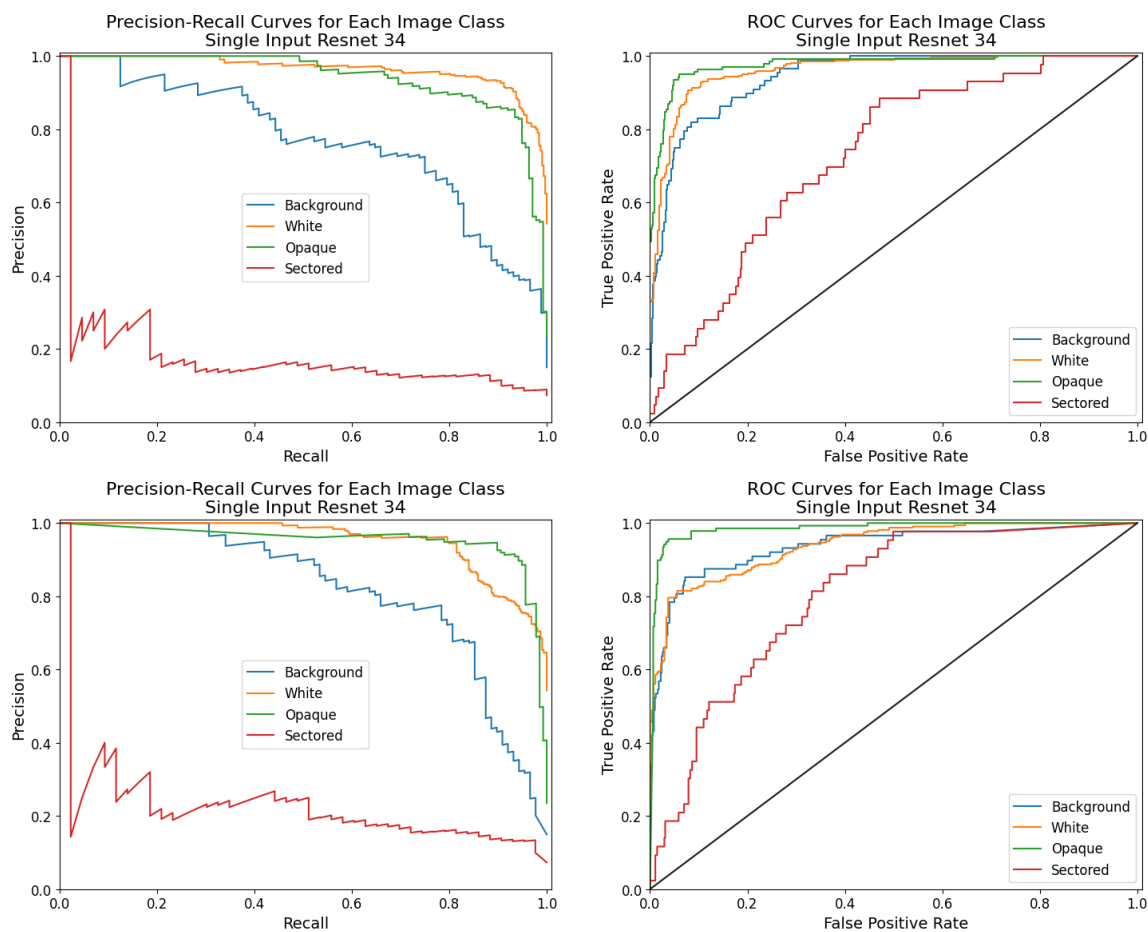


Figure 4.9: **Precision-Recall and ROC curves for the Single input Resnet 34 model.** Left: Precision-Recall curves showing performance of the single input Resnet 34 model on each image type. Right: ROC Curves showing qualitative performance of this model on each colony type compared to a random classifier (black line). The top row indicates performance curves for the single input Resnet 34 model applied to the original dataset, and the bottom row indicates performance curves for the single input Resnet 34 model applied to the augmented dataset.

for opaque colonies, 0.25 for sectored colonies, and 0.80 for background respectively. Across all eight model instances, this model had the highest accuracy for correctly predicting sectored colonies.

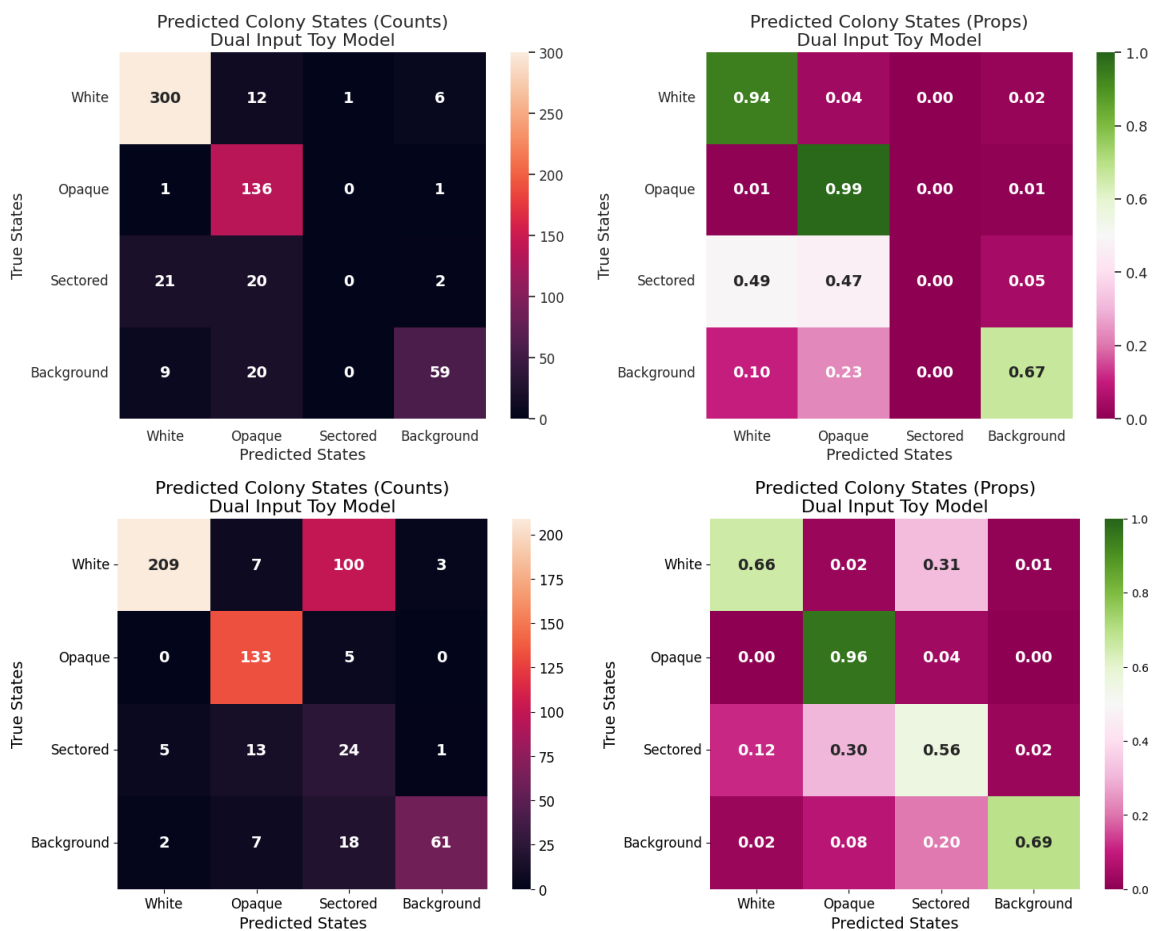


Figure 4.10: **Confusion Matrices for the Dual Input Toy Model.** Confusion matrices showing the proportion of colonies classified normalized by row, with true labels on the vertical axis, and predicted labels on the horizontal axis. Each cell is assigned a proportion of colonies predicted to have a certain label, given their assigned true label. Confusion matrices above are for the performance on the dual-input toy model with no augmentation applied (top) and with augmentation applied (bottom) to the training images.

Resnet 34

No augmentation. An accuracy of 84% across the entire test image set is obtained. The per-class accuracies are 91% for white colonies, 96% for opaque colonies, 14% for sectored colonies, and 75% for background respectively (Figure 4.11 (top)). (Precision, Recall) scores for each class are (0.91, 0.90) for white colonies, (0.96, 0.81)

for opaque colonies, (0.14, 0.30) for sectored colonies, and (0.75, 0.81) for background respectively. F1 scores for each class are 0.91 for white colonies, 0.88 for opaque colonies, 0.19 for sectored colonies, and 0.78 for background respectively.

With augmentation. An accuracy of 80% across the entire test image set is obtained. The per-class accuracies are 77% for white colonies, 96% for opaque colonies, 40% for sectored colonies, and 83% for background respectively (Figure 4.11 (bottom)). (Precision, Recall) scores for each class are (0.77, 0.96) for white colonies, (0.96, 0.90) for opaque colonies, (0.40, 0.30) for sectored colonies, and (0.83, 0.57) for background respectively. F1 scores for each class are 0.85 for white colonies, 0.93 for opaque colonies, 0.34 for sectored colonies, and 0.68 for background respectively.

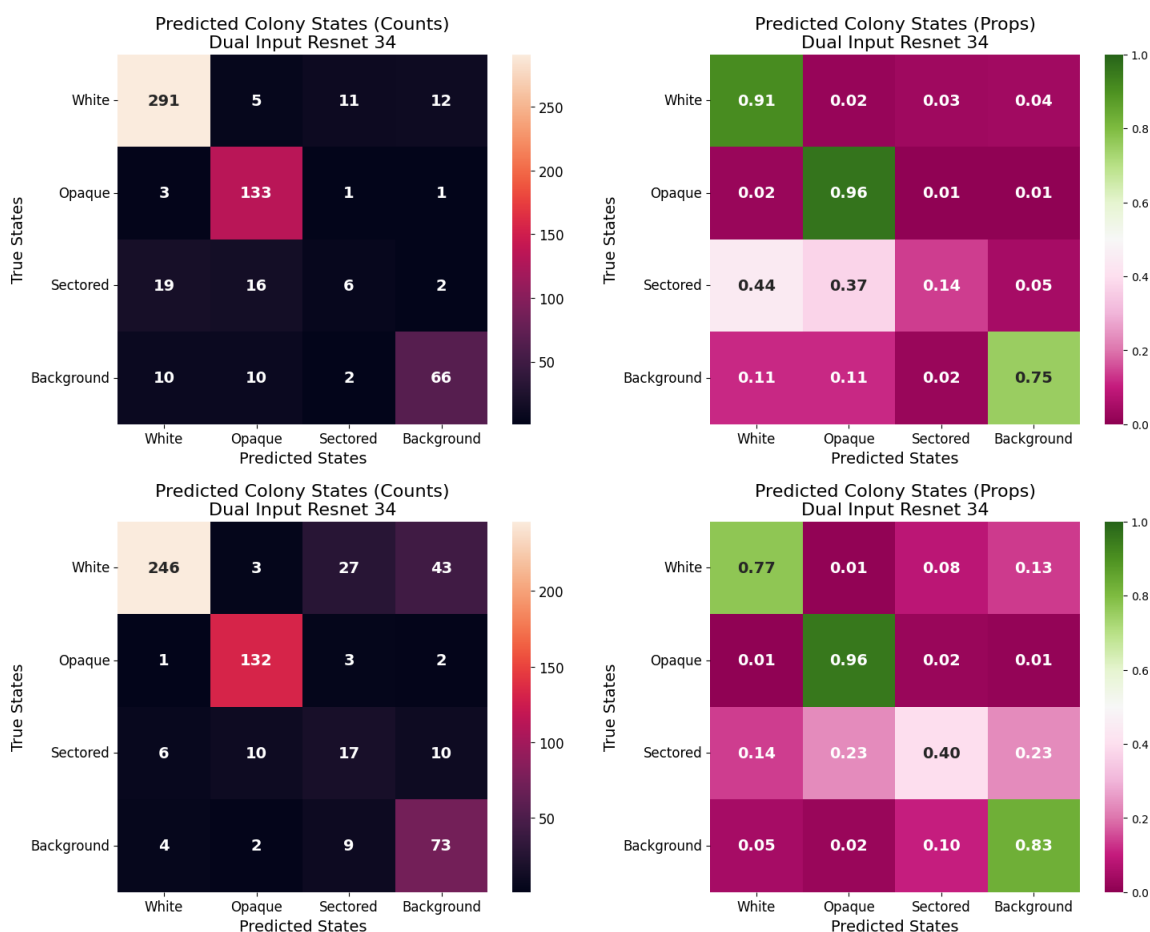


Figure 4.11: **Confusion Matrices for the Dual Input Resnet 34 Model.** Confusion matrices showing the proportion of colonies classified normalized by row, with true labels on the vertical axis, and predicted labels on the horizontal axis. Each cell is assigned a proportion of colonies predicted to have a certain label, given their assigned true label. Confusion matrices above are for the performance on the dual-input Resnet 34 model with no augmentation applied (top) and with augmentation applied (bottom) to the training images.

4.3.4 Dual Input Architectures: Qualitative Performance

Toy Model

No augmentation. Precision-Recall AUC scores for each class are 0.94 for white colonies, 0.88 for opaque colonies, 0.18 for sectored colonies, and 0.72 for background respectively. ROC-AUC scores for each class are 0.93 for white colonies, 0.96 for opaque colonies, 0.72 for sectored colonies, and 0.89 for background respectively. (Figure 4.12 (top))

With augmentation. Precision-Recall AUC scores for each class are 0.93 for white colonies, 0.94 for opaque colonies, 0.22 for sectored colonies, and 0.81 for background respectively. ROC-AUC scores for each class are 0.93 for white colonies, 0.98 for opaque colonies, 0.78 for sectored colonies, and 0.90 for background respectively. (Figure 4.12 (bottom))

Resnet 34

No augmentation. Precision-Recall AUC scores for each class are 0.92 for white colonies, 0.94 for opaque colonies, 0.22 for sectored colonies, and 0.79 for background respectively. ROC-AUC scores for each class are 0.94 for white colonies, 0.98 for opaque colonies, 0.77 for sectored colonies, and 0.92 for background respectively. (Figure 4.13 (top))

With augmentation. Precision-Recall AUC scores for each class are 0.93 for white colonies, 0.96 for opaque colonies, 0.20 for sectored colonies, and 0.73 for background respectively. ROC-AUC scores for each class are 0.89 for white colonies, 0.99 for opaque colonies, 0.81 for sectored colonies, and 0.93 for background respectively. (Figure 4.13 (bottom))

4.3.5 Robustness of Model Performance

Since machine learning models naturally have stochastic components which in turn lead to varying results, we opted to train and test 10 instances of each model to analyze independently how the performance of each model varies.

Breakdowns of the accuracy scores on the validation and testing images across each of the 10 instances of all models are shown in Table 4.2. In most cases, we find that the simplest model (the single input toy model) has better performance overall on the same images. For the dual input Resnet 34 model, training with augmented images results in the best performance overall for this type of model. In each instance, the dual input toy model performs the worst on both scenarios where data augmentation is or is not considered in the training process. The significantly varying accuracy of the dual input toy model suggests that this model is highly unstable and will likely perform better through a refined training procedure than those used for the other models. This quantitative analysis suggests that the performance of each of the model types, with the exception of the dual input toy model, are very similar.

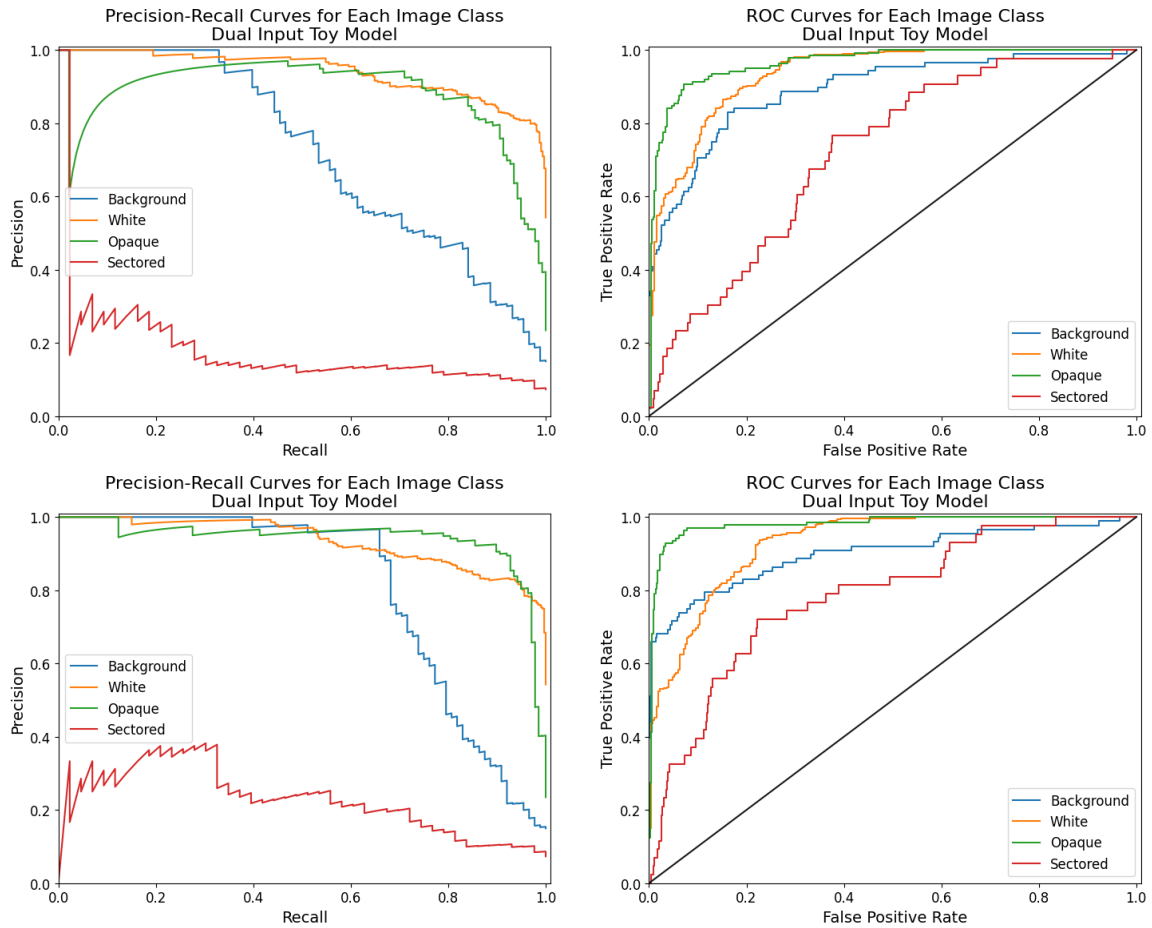


Figure 4.12: **Precision-Recall and ROC curves for the dual input toy model.** Left: Precision-Recall curves showing performance of the dual input toy model on each image type. Right: ROC Curves showing qualitative performance of this model on each colony type compared to a random classifier (black line). The top row indicates performance curves for the dual input toy model applied to the original dataset, and the bottom row indicates performance curves for the dual input toy model applied to the augmented dataset.

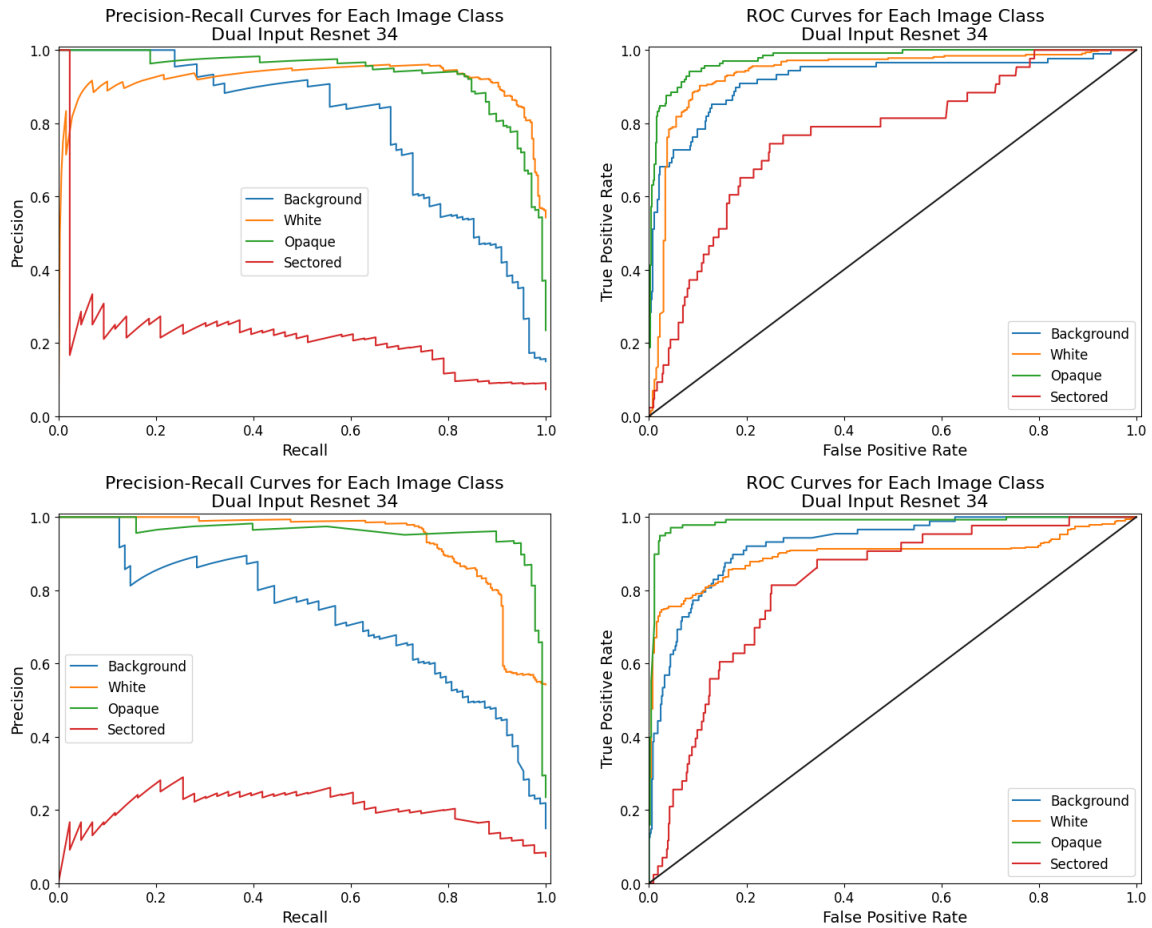


Figure 4.13: **Precision-Recall and ROC curves for the Dual input Resnet 34 model.** Left: Precision-Recall curves showing performance of the dual input Resnet 34 model on each image type. Right: ROC Curves showing qualitative performance of this model on each colony type compared to a random classifier (black line). The top row indicates performance curves for the dual input Resnet 34 model applied to the original dataset, and the bottom row indicates performance curves for the dual input Resnet 34 model applied to the augmented dataset.

Table 4.2: **Prediction accuracy of each model over multiple runs.** Average proportion of the test images correctly classified in both the validation and testing sets when models are trained with or without augmented images. Each row represents the model type (both the toy model and modified Resnet 34), grouped by the number of inputs. Each column indicates the image set (validation or testing images), grouped by whether augmented images were included in the training set. Each cell shows the average accuracy score obtained across the 10 models within one standard deviation. Green shaded cells indicate best performing model given an image set, while orange indicates the worst performing model.

		No Aug		With Aug	
		Val Acc %	Test Acc %	Val Acc %	Test Acc %
Image only	Toy Model	86.70 \pm 0.54	83.56 \pm 1.26	96.12 \pm 0.40	82.47 \pm 0.61
	ResNet 34	83.35 \pm 3.44	81.55 \pm 2.99	96.01 \pm 0.97	82.44 \pm 1.23
Image + Size	Toy Model	80.63 \pm 8.94	77.29 \pm 7.70	72.75 \pm 34.66	61.19 \pm 29.25
	ResNet 34	85.70 \pm 1.98	83.15 \pm 2.57	95.92 \pm 0.81	83.02 \pm 1.20

Distributions of prediction scores across 10 instances of the single input toy model are shown in Figure 4.14 to visualize stochastic changes in the accuracy of the model applied to the testing images. From this visualization we see that each model performs very well at classifying white and opaque colonies, and that each model incorrectly classifies most sectoried colonies as either white or opaque.

4.4 Discussion

Controlling for the formation of white and opaque colonies is easier when the experiment incorporates known knowledge about what “locks” colonies into one phenotype over the other. However, this is not necessarily the case for colonies exhibiting both white and opaque phenotypes simultaneously. For sectoried colonies, it is noticeable that a white-to-opaque switching event (or vice-versa) had occurred. For homogeneous colonies it is much harder to determine whether or not a switching event took place at all.

Since most colonies in our dataset are homogeneous (either fully white or fully opaque), each model has a sufficient amount of training data to adequately classify these types of colonies. In contrast, since we have a lack of sectoried colony data in the images, models have insufficient performance on correctly classifying sectoried colonies. When we included augmented data in the training process, we show a slight improvement in the qualitative performance of each model toward accurately classifying all colony types. In particular, the ROC plots indicate that sectoried colony classification had the most improvement overall across the four models considered in this work.

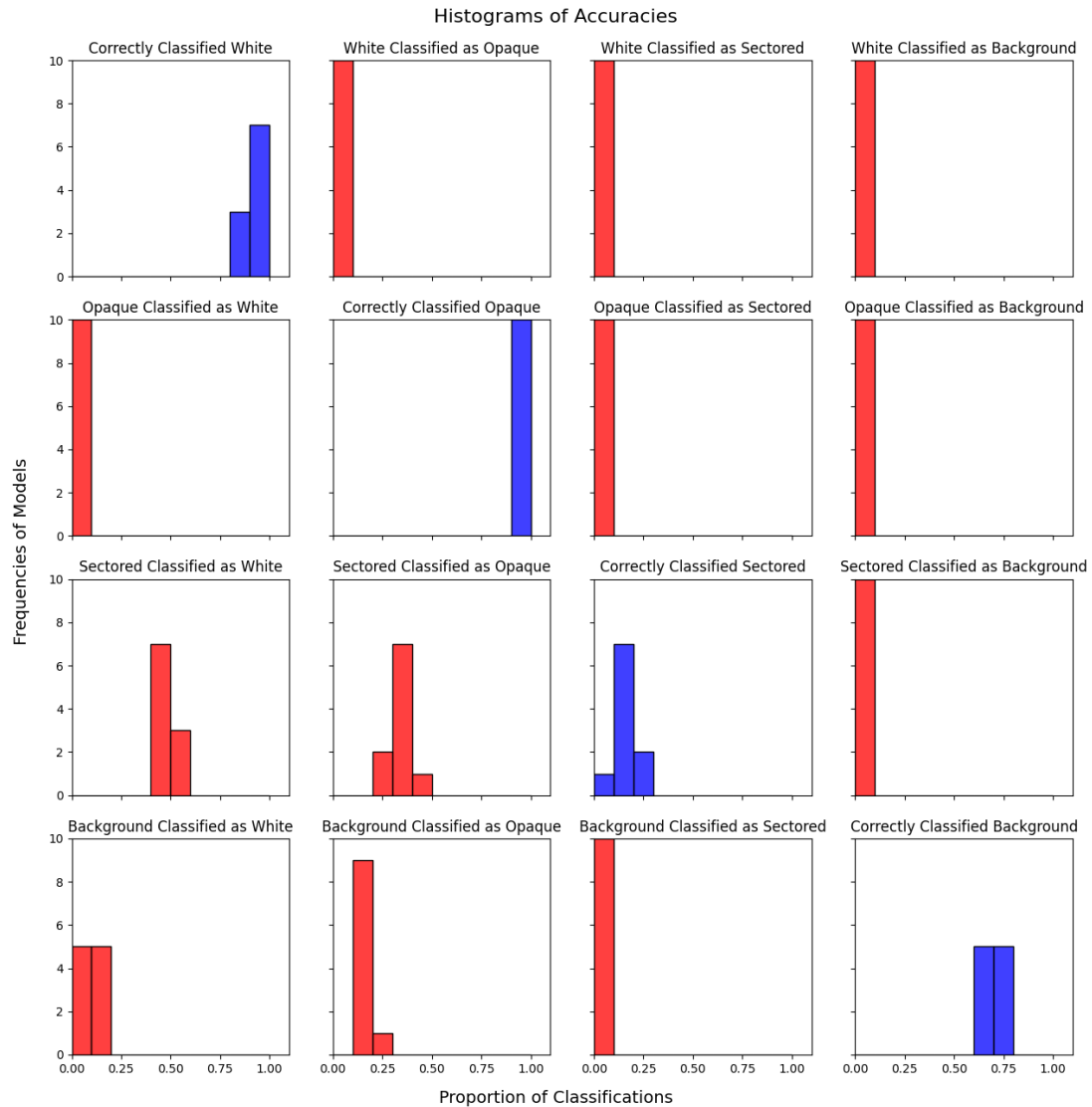


Figure 4.14: **Distribution of prediction scores for the single input toy model without data augmentation.** Histograms showing the distributions of the accuracy scores for true and predicted class pairs for all 10 instances of the same model. The main diagonal corresponds to the 10 model accuracies with the heights of the bars indicating how many models achieves this accuracy score, while off diagonal areas indicate proportions of colonies misclassified for each instance.

While we have shown that inclusion of augmented images leads to a positive improvement on model performance, it is possible to bolster performance even further by under-sampling from the data available in addition to using synthetic images. At present, no under-sampling has been performed in this work; only synthetic colony images were injected into the training process to create a balanced dataset. However, because the difference between the quantity of white and sectoried colonies is very high, a large quantity of synthetic images are needed to obtain a balanced dataset. Since the synthetic images all exhibit the same features, redundancy may become an issue. In contrast, under-sampling may be used to reduce the amount of images from over-represented classes, thus helping to achieve an equitable balance between image classes while potentially sacrificing some feature information from the over-represented class. Training a neural network that captures all primary features found in the real and synthetic images independently should lead to a higher increase in model performance over simply adding synthetic data to a large real image set. I suggest that an equitable approach be considered for designing the set of training images that both incorporates a balance of real and synthetic data across all image classes in the training process.

While cultivating *C. albicans* on CHROMagar allows for visual differentiation of white and opaque colonies after growth, Sabouraud’s Dextrose Agar (SDA) is another media commonly involved with cultivation of experimental *C. albicans*. One reason for this is that the media has been commercially available for significantly longer than CHROMagar which was first reported in the 1990s [121, 124]. While CHROMagar and SDA are both types of chromogenic media, CHROMagar allows for a much more clear differentiation of white and opaque colonies than SDA media, whereas in SDA media differentiation between the colony types is very difficult to immediately notice. Chapter 5 will explore the performance of the the models used here on this different media to test the limits on their predictive power.

4.5 Conclusion and Future Work

This Chapter discussed the construction of a computational framework for counting and quantifying *C. albicans* colonies undergoing white-opaque phenotypic switches. To accomplish this task, we extract the colonies from the images using circle detection and train deep learning models to classify colonies as white, opaque, or sectoried. In addition, we introduced a framework for training deep learning models that include additional metadata such as colony size as a secondary input. In our results, we showed that using circular feature extraction coupled with deep learning for image classification efficiently and sufficiently produces human comparable colony counts. This pipeline will serve to significantly reduce manual labor for future chromogenic colony quantification.

We have demonstrated the usefulness of our pipeline in quantifying colonies growing on CHROMagar media. However, there are other useful media involved in cultivation of *C. albicans* that will serve as additional training data to improve the

generalizability of the pipeline discussed in this Chapter. In the next Chapter, we will extend the applicability of this pipeline by introducing additional datasets for training our models that include images of colonies grown under SDA media, aiming to achieve a goal of applying this framework toward large quantities of colony image data where manual annotation becomes intractable.

Chapter 5

Counting Microbial Colonies

This chapter covers prior work with Dr. Teal Brechtel from the University of Massachusetts, Amherst as well as work in progress in collaboration with Dr. Clarissa Nobile, Austin Perry, Daravuth Cheam, and Dr. Ruihao Li at UC Merced. I led the development of the methodology used in this chapter and wrote the text for each section. Austin Perry, Dr. Ruihao Li, Dr. Teal Brechtel, and Daravuth Cheam curated the image data that was used to test the methodologies in this work. Examples of the images used in this work are provided in Appendix C.

5.1 Introduction

This chapter will demonstrate how coupling deep learning with traditional feature detection is capable of resolving detection problems where traditional feature extraction alone is inadequate for a subset of images. We will demonstrate the effectiveness of this strategy on two datasets, one where traditional detection is okay, and one where it is not. We will discuss the pitfalls that may arise when using traditional detection methods alone and how the integration of deep learning is able to aid in addressing some of these issues.

In Chapter 4, I have shown that we can classify white and opaque colonies efficiently under my proposed computational approach. However, the dataset that was used in this chapter is very small and contains only one type of growth media. When the model is tested on images of colonies grown on a different media, the performance of the model suffers. This behavior is because the model was trained to recognize colonies on CHROMagar media but was never trained to recognize colonies on a different media. This chapter will extend upon this work by training and applying these models on additional image data and multiple media types and will demonstrate high accuracy in segmenting colonies from both media types.

In this Chapter, we apply the techniques discussed in this dissertation toward four additional large datasets of images containing plated yeast colonies. We will use two of these image sets to demonstrate that for the colony counting problem, traditional circle detection with occasional image preprocessing is not always a simple solution.

We will use the other two image sets to extend upon the work of Chapter 4 by demonstrating its generalizability toward images of different growth media.

The first dataset considered is a collection of images of plated *S. cerevisiae* colonies with the [*PSI*⁺] prion phenotype on agar media, where the images show the bottom of each plate. The second dataset is a collection of five images of plated bacterial colony biofilms. We will show in this Chapter that edge detection methods alone are inadequate in segmenting the colonies from the second set of images and thus attempt to integrate deep learning into the colony counting process.

The third dataset is an extension of the dataset used in Chapter 4 with additional images of CHROMagar plates containing between 4 and 250 colonies. The fourth dataset contains 60 images of *C. albicans* colonies plated on Sabouraud Dextrose Agar (SDA) media. The primary difference in this dataset compared to the former one is that all colonies are visually similar from above, making it difficult and tedious to distinguish between white and opaque colonies by eye.

Before discussing further applications of the deep learning frameworks throughout this dissertation in more detail, we must first discuss the necessity of using deep learning in terms of whether a traditional approach will obtain similar results to a deep learning model while also saving time. In the first half of this Chapter, we will address the colony counting problem by showing when traditional circle detection methods fail, integrating deep learning helps provide a window to finding a solution to the colony counting problem. For the second half of this Chapter, I will extend upon the work in Chapter 4 for improving colony detection and classification on *C. albicans* colony images. More specifically, I will demonstrate the generalizability of our deep learning models toward classifying colonies from different growth media. This section aims to demonstrate the usefulness of deep learning to quantify colonies where manual annotation is more prone to errors depending on the type of media used to cultivate *C. albicans* colonies.

5.2 Analyzing Performance of Circular Object Detection for Colony Images

In this section we will talk about situations where the use of deep learning may or may not be necessary for general colony counting. For this section, the assumed problem we are aiming to find an efficient solution for is a ballpark estimate of total colony counts across large numbers of plates.

The image sets we use for this include [*PSI*⁺] colonies taken from the bottom of the plate. Here we assume that the composition of the colonies in the images are already known. What we do not have are the numbers of colonies per plate. To aid in our approximation for the number of colonies in each image, we will use four important properties found within the [*PSI*⁺] colony images. First, at a local level colonies appear circular to the naked eye. Second, there is a fair amount of contrast between each colony and the plate. Third, the variation in the color of each colony as

depicted in this images is relatively small, suggesting colonies appear homogeneous at the naked eye. Fourth, other circular objects of similar sizes to the colonies in each image are not present; the only other circular objects in each image are the plates. All of these conditions motivate the application of a circle detection algorithm toward each image in order to count the number of colonies present in each image and segment the regions of interest.

The second dataset we use in this section contains five images of plates containing 20-200 bacterial colonies. The colonies in this dataset each have the same properties as the colonies from the $[PSI^+]$ colony image dataset above. However, the weakest property in this set is the visual distinction between colonies and the plate. We will show that circle detection alone is not enough to segment colonies from these images, and as such, additional processing of the images is required.

For the work in this Chapter, we extensively utilize the circle Hough transform (CHT) for circular object detection. In the next section we provide details about how this method works when considering candidates for circular objects in an image.

5.2.1 More about the CHT Method

Originally a method used to detect lines in an image, CHT has been adapted to detect imperfectly round objects of a given radius. This method is typically applied on single-channel or grayscale images, since its performance relies on the result of an edge detection algorithm such as the Canny edge detector [24]. As stated earlier, the images we wish to analyze need to have a high enough contrast between the colonies and the plate; this is because edge detection seeks to find large changes in local intensity. The locations of detected edges are important; the arrangement of edge pixels in a circle will be used for circle detection, while the range of radii of the arrangement in pixels approximate the size of the circle.

The equation for a circle of radius r in the x-y plane centered at the point (a, b) is given by

$$(x - a)^2 + (y - b)^2 = r^2. \quad (5.1)$$

To detect all possible circles of radius r in the original image, one has to find all possible parameter pairs (a, b) that are potential candidates for circle centers. What CHT does is map a collection of points in Cartesian space to a circle in Hough space which represents the space of parameters a-b. In our case, each pixel represents a single point in Cartesian space, and each pixel is mapped to a circle of pixels of radius r in the Hough space. Next, a corresponding matrix called the accumulator is created to represent the number of circles that cover each given pixel in the Hough space. This is considered an intensity representation for the number of circles that pass through each pixel in the Hough space. The main idea of CHT is to find the regions in the accumulator matrix that achieve local maxima relative to all other entries in the accumulator matrix. The location of these local maxima indicate locations of the

centers of circles of a pre-determined radius in the original image. Each step of this process is visualized in Figure 5.1.

At the start of the algorithm, the original image goes through a separate edge detection algorithm (Canny [24] or Sobel [30,110,152]) to reveal the points of contrast to be used in the next step. The result is then stored as a separate image. Next, the accumulator matrix corresponding to the locations of the centers of circles of radius r is initialized to zero. For each point that appears in the edge detection step, the next step is to map this point to a circle in the Hough space whose center is the same as the location in Cartesian space. For each section in the Hough space that circle covers, the corresponding location in the accumulator matrix is incremented by one.

The next step is to find the local maxima in the accumulator matrix arising from the intersection of multiple circles in the Hough space. These maxima are found through the voting scheme used when updating the accumulator matrix in the previous step. The points that received a high number of votes are marked as potential candidates for the centers of circles in Cartesian space. Finally candidates are filtered out based on the intensity of each maximum relative to the radius of the circle and detection sensitivity, then the candidates for circle centers are chosen. Further work is done for optimizing the radius, but we do not discuss this here.

We use the Matlab function `imfindcircles` to implement the circle Hough transform for finding circular objects in images. This function uses the Atherton-Kerbyson method [6] by default in order to quickly construct the accumulator matrix. This function also requires either a single value or a 2-vector containing the minimum and maximum radius respectively for detecting circles in the input image. The sensitivity parameter sets a threshold for the value of the local maxima in the accumulator matrix for each radius, allowing us to detect imperfect circular objects present in the image. Depending on the composition of colonies in the images, different configurations of `imfindcircles` are necessary. In the subsections below, we configure and apply `imfindcircles` independently for additional image sets.

5.2.2 CHT Implementation: $[PSI^+]$ Colony Images

To detect the plates in the images, we set the sensitivity to 0.98 and 0.99 and the radius range to 450-550 pixels. A high sensitivity is used since the circular objects—the plates—we wish to detect are large with potentially imperfect shapes. To speed up computational time for detecting the plates, we re-scale the dimension of all the original images by one third before implementing `imfindcircles`. Once obtaining the center and radius, the location of the center is reciprocally re-scaled so that the center corresponds to the same location in the original, un-scaled image.

For each circle that was detected in the original image, we crop a square region around the detected circle plus up to 20 pixels on each side to ensure that the entire plate is contained in the image. The result is then saved as a separate image for further analysis. Due to the position of the camera relative to the plate, small variances in the image sizes are apparent, with dimensions ranging from 525 to 540 pixels per

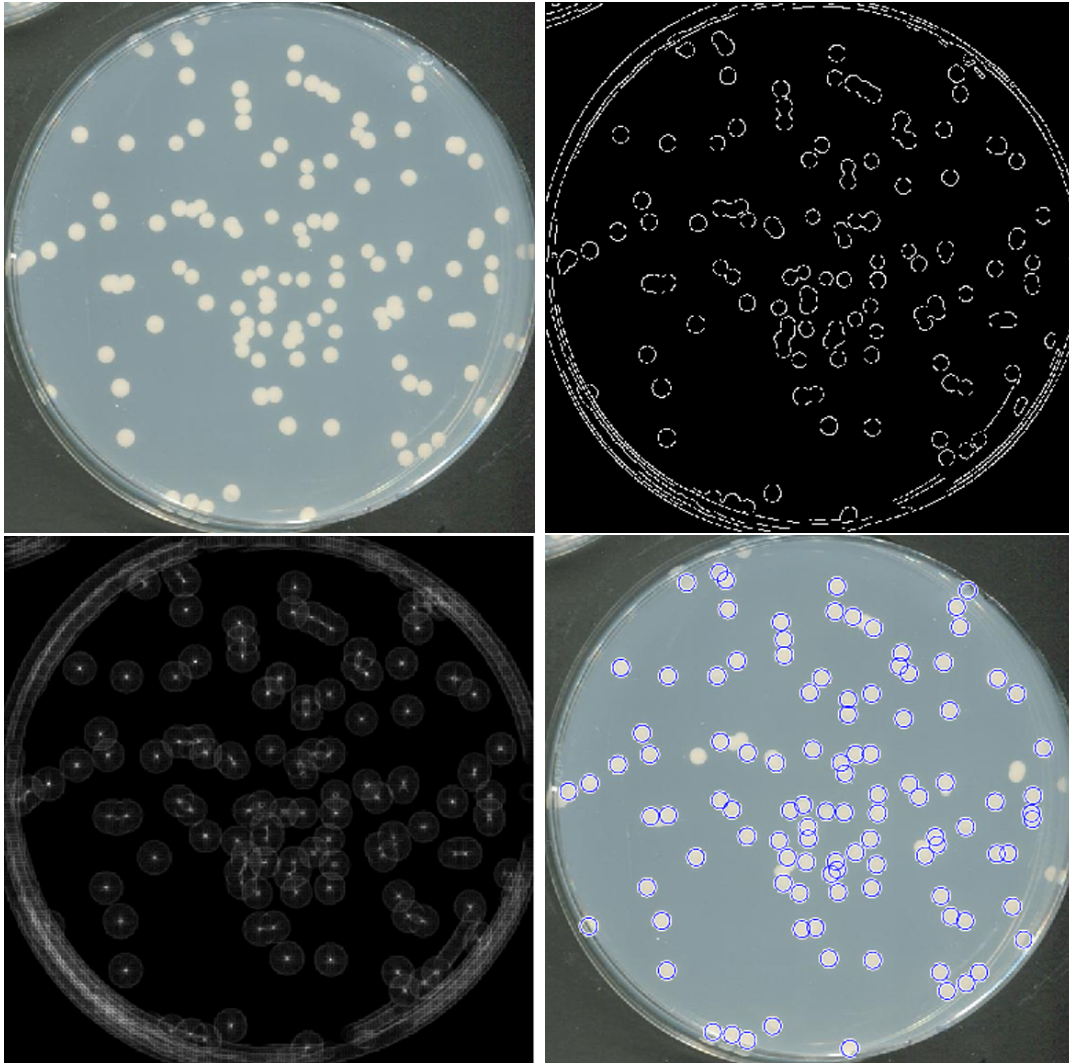


Figure 5.1: **Visualizing steps in the circle Hough transform.** Application of the circle Hough transform to a sample image of yeast colonies on a petri dish. (Top left): The original image. (Top right): The image was converted to grayscale followed by a Canny filter to locate the edges between pixels. The thresholding value was chosen based on Otsu's method. (Bottom left): The accumulator matrix corresponding to the edges found in the original image. A circle of radius 9 pixels centered at each edge pixel was constructed, and the number of circles that cross each pixel is recorded. Darker pixels indicate less circles passing through them, while lighter pixels have more circles passing through them. The local maxima in the accumulator matrix (the isolated white regions) are potential candidates for the centers of circles of radius 9 pixels corresponding to the locations of the circles in the original image. (Bottom right): The original image with the circle Hough transform performed using the Matlab function `imfindcircles`. The centers of the detected circles correspond to the peaks of the accumulator matrix on the bottom left.

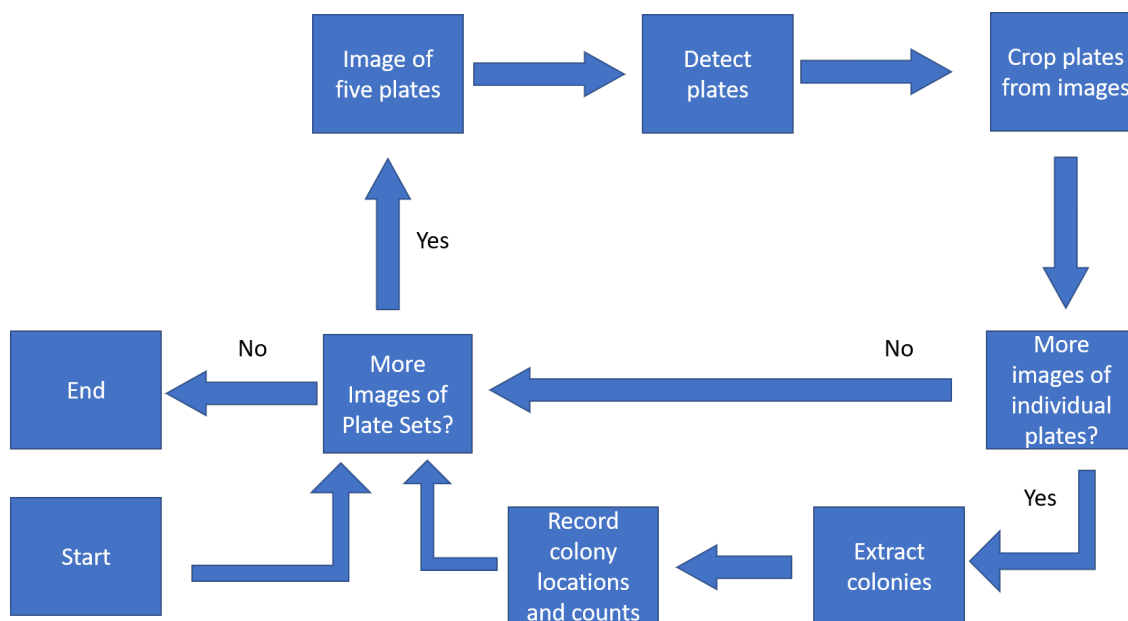


Figure 5.2: **Flowchart of the colony extraction process on the $[PSI^+]$ yeast colony image set.** At the start, we have a collection of images showing five plates each. A circle Hough transform algorithm is implemented in Matlab to extract the center and size of the plates. Each plates is then cropped and saved as a separate image for further analysis. For each plate, we run a second circle Hough transform in Matlab set to detect individual colonies and record the total number and locations of all the colonies in each image. The process is repeated for each of the images of five plates in the collection before termination.

cropped image.

We then implement `imfindcircles` a second time on the cropped images in order to detect the colonies. Here, we set the sensitivity to 0.95 and the radius range to 10-40 pixels. The radius range was chosen based on the colony size relative to the dimensions of the image and is left wide enough for detecting feasibly sized colonies from the images. The location of each colony and the total number of colonies in each separate image is recorded. The process is repeated for all the individual plates detected. The entire image processing workflow is shown in Figure 5.2.

5.2.3 Measuring Accuracy of CHT

The circle Hough transform is an excellent method for detecting circular objects in images. However, the method itself has computational drawbacks. First, the circle Hough transform relies on the result of an edge detection algorithm, such as Canny [24] or Sobel [30,110,152], to build the accumulator matrix. As such, the objects of interest in the image must have enough contrast between the object and the background in order for the edge detector to correctly segment the image. Technical factors such as

image compression can also introduce regions with gradients smaller or larger than an uncompressed image. As such, these factors can affect the accuracy of the circle Hough transform by making it more difficult for the edge detection step to locate the object of interest. Second, the algorithm can capture other circular objects that you do not intend to be detected. The image data must be chosen carefully as to ensure that only the circular objects you wish to examine are detectable. The most convenient approach is to ensure that the objects you want to analyze have a shape unique to every other object in the image. Any circular objects that you do not want to have detected should either be absent from the image or of a radius size outside the desired range.

The third issue is that `imfindcircles` explicitly requires a range of possible radii for objects to detect. In the case of image data, the units for radii are usually in pixels. Therefore, image acquisition methods would need to ensure that all images to be analyzed are of similar dimensions. A common tactic would be to use a large range of radii to account for a wider range of sizes for each object present in an image. However, this leads to another major drawback of the method which is high computational complexity. To find circles of different radii in the image, the algorithm requires a 3D accumulator space to track all the possible candidates for circles in the original image. Each accumulator matrix is used to find circles of a single radius, and so the algorithm would have to store a separate accumulator matrix for each radius in the range you want to search. Therefore, circle detection for objects with larger ranges of radii not only becomes more expensive as the range increases, but in the extreme case becomes computationally intractable.

Even if the range of radii for desired circular objects is known, another problem to resolve is how many circular objects detected are of the desired objects and similarly, how many are overlooked. The sensitivity parameter in `imfindcircles` allows for a tolerance of imperfection when detecting circular objects, with higher sensitivity increasing such tolerance. Finding the optimal value for sensitivity in practice will depend on the data itself, as we will demonstrate in this chapter.

To analyze the accuracy of the circle Hough transform implementation, we apply the method to 50 images of plates containing [*PSI*⁺] colonies. Each of the 50 plates have corresponding colony counts, while corresponding manual annotations are available for 10 out of these 50 images indicating the presence of a colony. The comparison between the total number of detected circles and the number of colonies counted by hand for each plate using both counting methods is shown in Figure 5.3. We use Adobe Photoshop Elements 11 to overlay each image containing the detected circles on top of their corresponding hand-counted image. The top layer is set to 50% opacity so that the hand counts and the algorithm counts are both visible before saving the result as a separate image. Next, we use Inkscape to mark every detection with color coded circles. We use the hand-counted plates as the true colony marker, and we use the detections from the circle Hough transform to test whether a colony was successfully detected. We mark “True positives” as colonies that were successfully detected, “False positives” as detections of non-colonies, and “False negatives” as

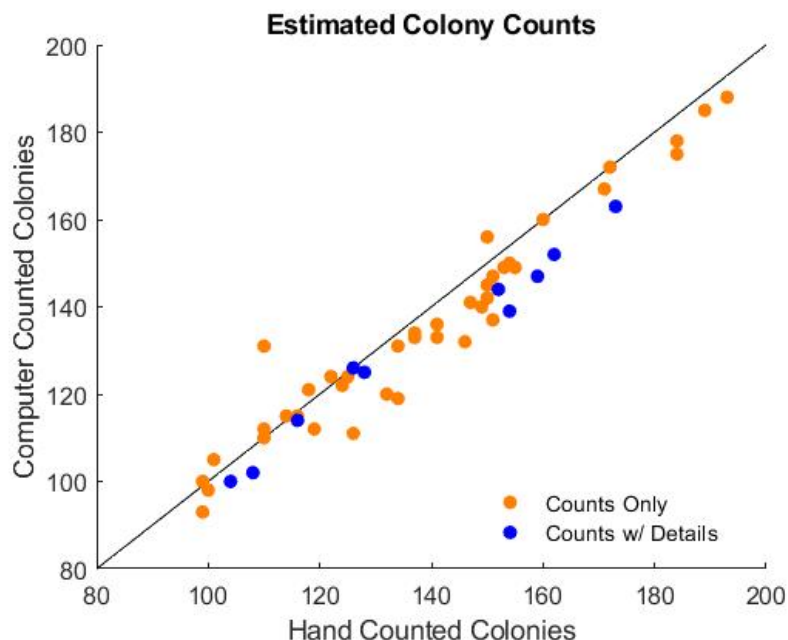


Figure 5.3: $[PSI^+]$ **colony counts**. Scatter plot showing the number of colonies counted using both methods. Each point represents a plate with a given colony count obtained by hand (x-axis) and using the circle Hough transform (y-axis). Points that lie on the black line indicate matching counts from both methods. Since more points appear below the line, this suggests that the circle Hough transform often undercounts the number of colonies actually present in the images.

colonies that failed to be detected. The total marks for these groups are recorded. The process above is repeated for a total of 10 plates (5 from "high 10 1" and 5 from "minus 10 1"). The counts are shown in Table 5.1.

We note that a majority of the 50 hand-counted plates had a higher total colony count than from the use of the circle Hough transform, with the error between the counts of both methods differing by at most 15 colonies. Under the more detailed analysis, we see that the number of true positives is always less than the actual number of colonies, indicating the presence of colonies that were undetected (false negatives).

To test the effects of local colony density and environmental effects on the accuracy of the circle Hough transform, we develop a pipeline extracting the location of the annotations of the colony. Before this operation can be performed, we must first complete the annotation of the counted images (Figure 5.4). The portion of the image containing the annotations are extracted, with each dot made small enough so that they will not overlap. The region of the annotations is made square and resized to 1000x1000, such that the center of the image corresponds directly to the center of the plate. Each dot is found by finding all the connected components in the image (i.e. the groups of pixels comprising each dot) and individually estimating and recording

Table 5.1: **Error analysis for colony detection.** Table of true positives (TP), false positives (FP), and false negatives (FN) for the 10 detailed annotated plate images.

Plate	True Counts	TP	FP	FN
1	162	152	0	10
2	108	101	1	8
3	128	123	1	7
4	116	113	1	4
5	154	136	1	16
6	126	146	0	13
7	126	123	3	3
8	173	160	3	13
9	152	140	3	3
10	104	98	2	2

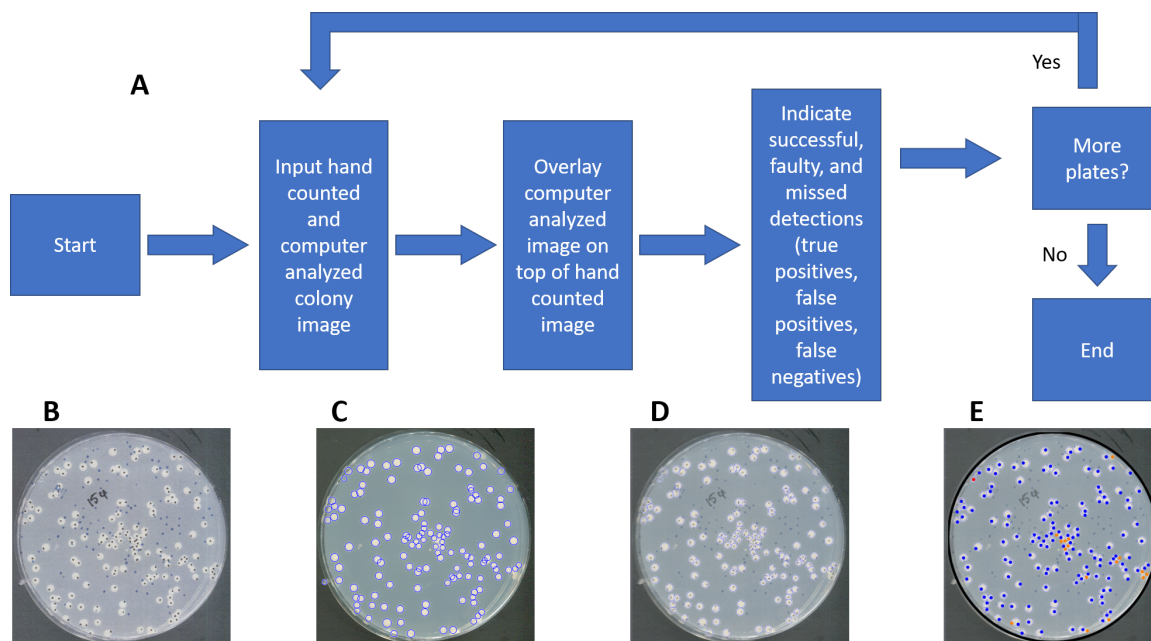


Figure 5.4: **Accuracy analysis pipeline for $[PSI^+]$ colony images.** Process for checking the accuracy of the circle Hough transform on the yeast colony image set. The entire process is shown in the flowchart in (A). Each plate used for further analysis has an image showing the hand-marked colonies with total counts (B) and a second image of the same plate that underwent detection using the circle Hough transform (C). Image (C) is set to 50% opacity and laid on top of image (B) in order to produce image (D) showing which colonies are marked and detected. The location of any true positives, false positives, and false negatives are marked by hand as shown in image (E).

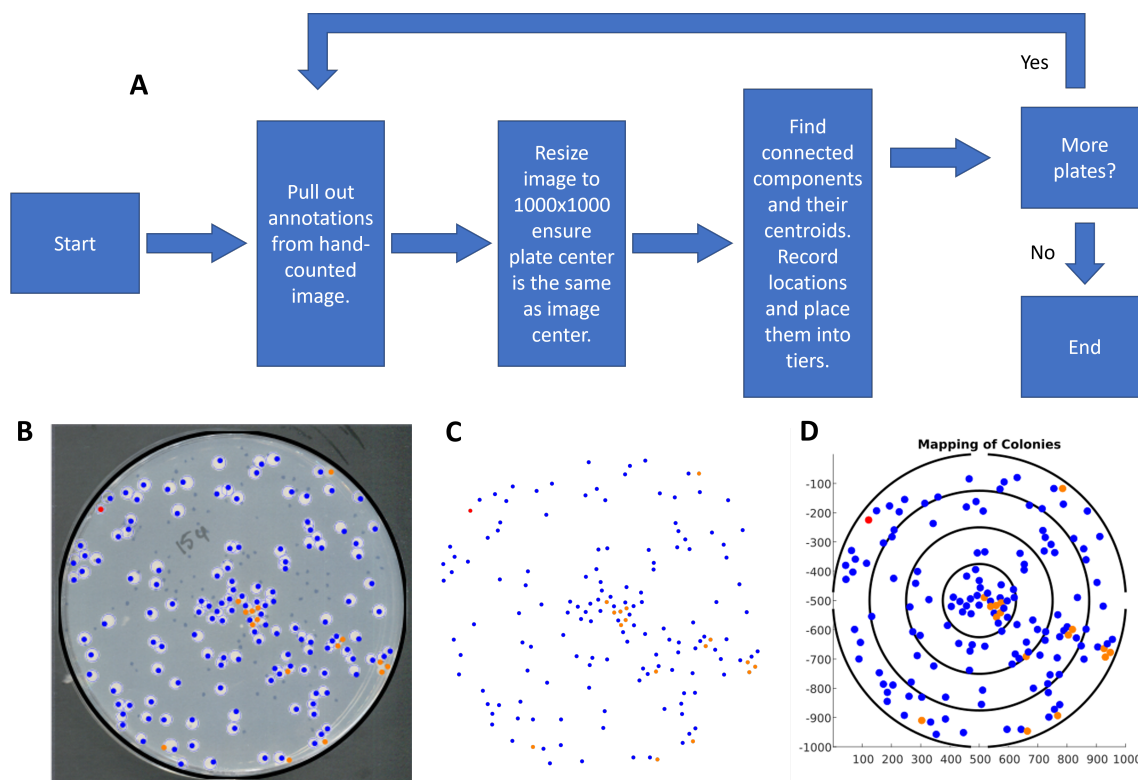


Figure 5.5: **Pipeline for extracting annotations from the images.** (A) Process of extracting the approximate location of each colony in the images using the annotations. Using the annotated images (B), we extract the annotations (C) and resize them to 1000x1000. The colors of the annotations are chosen so that they are clearly distinct from the background, while all annotations are made small enough so that they are non-overlapping. The number of dots present in the annotations is estimated by finding the number of connected components of (C). The centroid of each component is computed and plotted in (D) based on the color corresponding to each dot.

their centroids. Each dot is placed into one of four tiers based on their distance from the center of the plate (see Figure 5.5). Dots within the first 25% of the plate radius are in the first tier. Dots between the 25% and 50% radius are in the second tier. Dots between the 50% and 75% radius are in the third tier. Dots between the 75% radius and the plate border are placed in the fourth tier. The process is shown in Figure 5.5.

We find that nearly all false positives are in a close vicinity to the plate border (Figure 5.6, left), suggesting that the border itself inhibits the accuracy of the circle Hough transform. Analogously, border-adjacent colonies are what previous studies tend to ignore in their analyses, and so this result provides additional insight into the trend of why the inclusion of border colonies complicates colony counting.

We find that most of the false negatives appear in the first and fourth tiers of the

plates (Figure 5.6, right). Furthermore, most false negatives appear to be adjacent to other colonies where the distance between their approximated colony centers is less than one colony diameter, suggesting non-isolated colonies have a higher likelihood of being undetected. The latter is expected since the edges of the colonies would be rendered hidden when two or more colonies are in contact.

We then have our own measure for analysing the algorithm accuracy and efficacy of the circle Hough transform. This allows us through trial-and-error to estimate an optimal sensitivity parameter for detecting the $[PSI^+]$ colonies in this dataset (Figure 5.7).

5.2.4 U-Net + CHT: Addressing a Detection Problem in Bacteria Colony Images

This section explores the application of CHT and the deep learning framework developed in Chapter 3 to a dataset of five bacterial colony images. The details of the dataset are provided in Appendix C.1. Here we demonstrate the direct application of CHT to counting these bacterial colonies is insufficient, and thus we turn to deep learning to resolve issues with counting the bacterial colonies in these images.

Plate images were first extracted from the original images with `imfindcircles` using a radius range of 450-550 pixels and sensitivity of 0.985. Extracted plates were then resized to 1024x1024. Next `imfindcircles` was applied to each extracted plate using a radius range of 10-40 pixels and sensitivity of 0.95.

Figure 5.8 (top) shows an example of `imfindcircles` applied to one of the bacterial colony images and resulting circular objects detected. In each of the five cases we find that all detected objects lie on the border of the plate. Closer visual inspection showed that no detection was of a colony in the images, rendering `imfindcircles` insufficient for our problem on the original images. We then turn to the U-Net architecture described in Chapter 3 as an alternative solution to the bacterial colony counting problem in this section.

The synthetic images we use to train the U-Net architecture are generated using the process described in Appendix C.2. A total of 200 synthetic images are generated with corresponding binary ground-truth masks depicting the locations of colony pixels. From these images, 150 are used directly in the training process, and 50 are set aside for validation. The five bacterial colony images we started with are used as the testing set following the process of training U-Net. We utilize the same configurations and implementation for training U-Net as described in Chapter 3 and Appendix A.4 respectively, with the exception of the number of output classes which is set to 2 since the objective is to obtain binary segmentations of colony versus background.

After U-Net has been trained, the five testing images were used as input to U-Net and corresponding binary segmentations of the images were obtained as output. The resulting output ideally has the highest possible contrast between colony and background pixels. Figure 5.8 (bottom) shows an example of a binary segmentation of one of the bacterial colony images as output of the trained U-Net. We then use

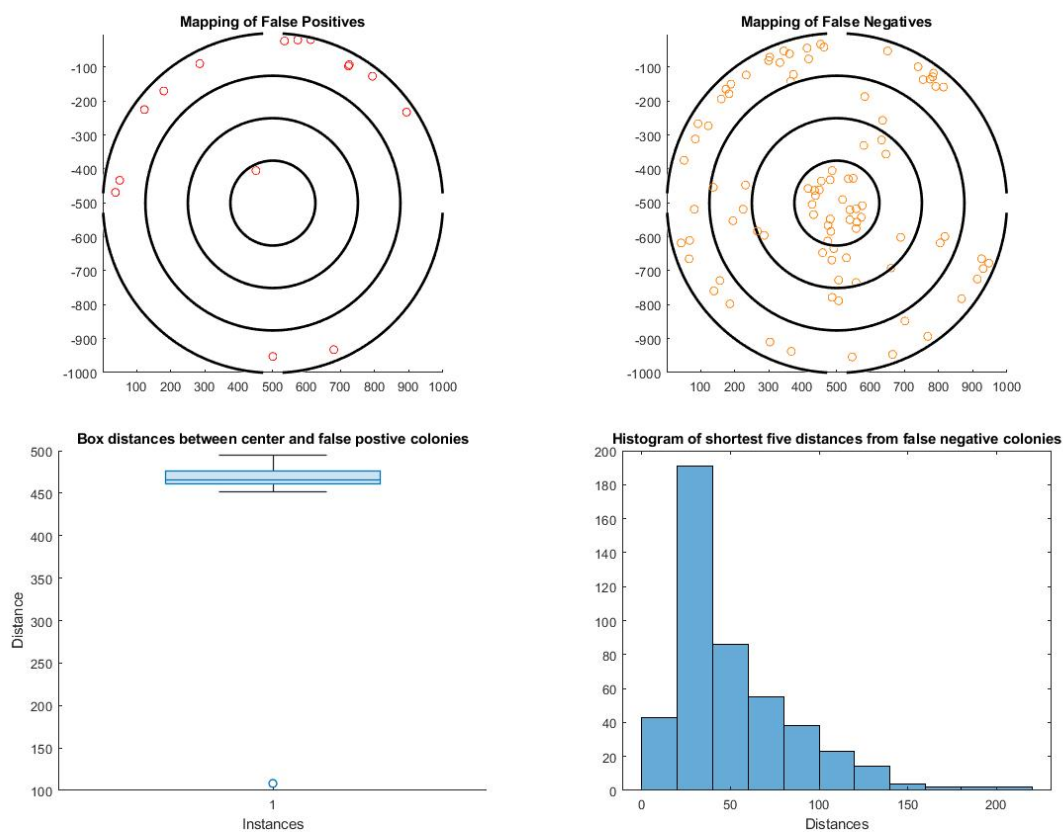


Figure 5.6: **Analyzing false positive and false negative colony detections.** The top row shows plots of the relative location of any false positive detection (left) and undetected colony (right) with respect to the center of the plate. For each colony, the distance between its center and the plate's center is recorded and the colony is placed into one of four tiers based on its distance from the plate's center. The boxplot showing the distribution of distances from the plate center for all false positives is shown on the bottom left. From the 10 plates analyzed further, 16 false positive detections were recorded, with 15 of them found in the tier closest to the plate's border. The histogram showing the distribution of distances from the plate center for all false negatives is shown on the bottom right. The distance between each false negative and its five nearest neighbors is recorded in the histogram. Neighbors of false negatives tend to be within 100 pixels away under our standardization. This suggests that colonies that form tight clusters of two or more colonies are more likely to be undetected by the circle Hough transform.

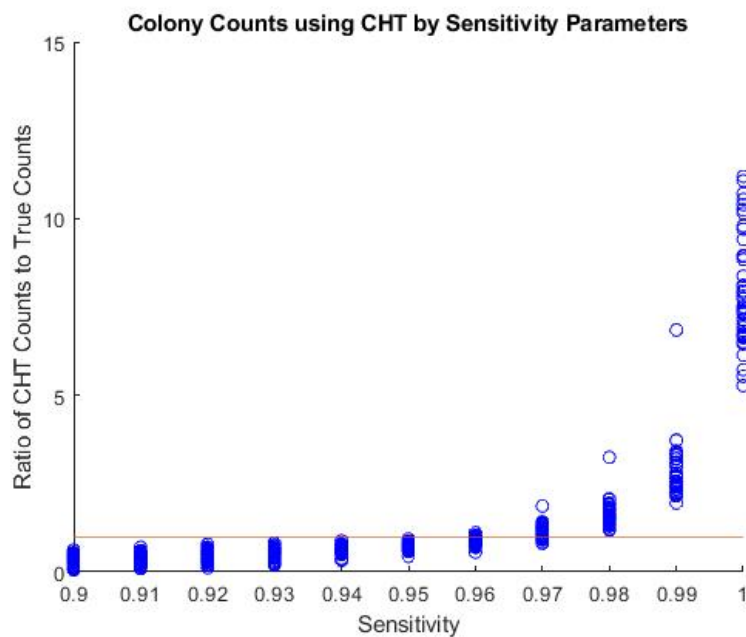


Figure 5.7: **Finding the optimal sensitivity parameter.** Ratio of detections as a function of the sensitivity parameter. For each value of the sensitivity parameter between 0.9 and 1, the circle Hough transform was performed on the same 50 images and the number of detections was recorded and divided by the true number of colonies. The red line is where the number of detections is equal to the true number of colonies. The plot suggests that a sensitivity between 0.96 and 0.97 may be suitable for our image set, but more analysis is needed to gauge the accuracy under these parameter values.

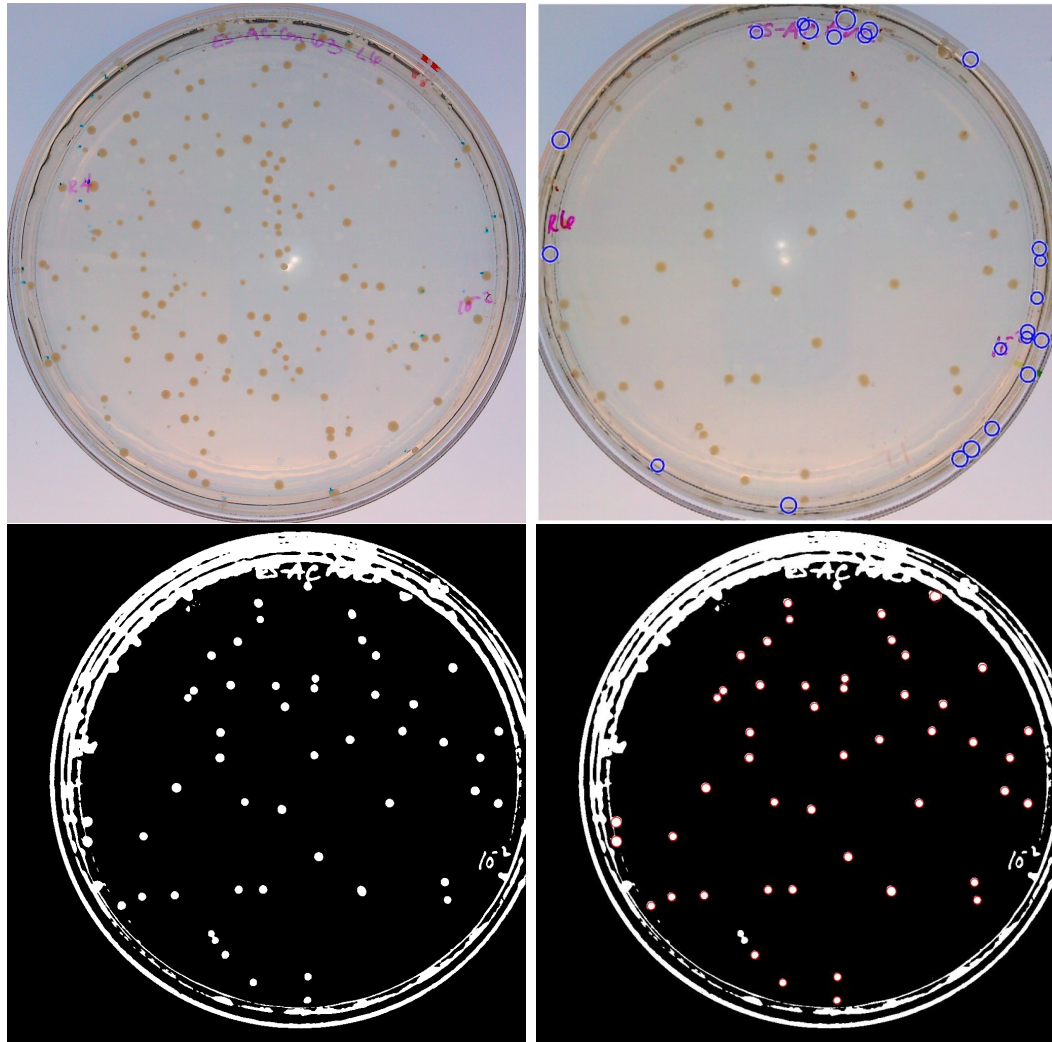


Figure 5.8: **Using `imfindcircles` may be ineffective without preprocessing.** Top Left: An example of one of the five bacterial colony plate images used in this analysis. Top Right: Applying `imfindcircles` results in circular object detections only at the edge of the plate. All detected objects are false positives, with no internal colonies detected. We then apply U-Net to the original image to obtain a binary segmentation (Bottom Left) of the original image. We then apply `imfindcircles` to the segmentation (Bottom Right) which results in colonies being detected inside the plate.

colony extraction methods described in Appendix A.3 to locate the colonies in the binary images. As a result, many detections were found, nearly all of them were bacterial colonies, with a few within approximately one colony diameter distance of the border of the plate.

Our results further demonstrate an issue with using circle detection methods for locating objects that have insufficient contrast between the foreground and background. Our results provide more information on the limitations of using circle detection methods directly on images without any preprocessing. The biggest limitation is insufficient contrast between foreground objects and background. In the images, $[PSI^+]$ colonies have high contrast with the plate surface, whereas the contrast between the bacterial colonies and the plates is less pronounced. However, such changes in contrast lead to fundamentally different results when detecting circular objects using edge detection algorithms. Here we have shown the potential for how deep learning aids in developing a solution to the colony counting problem by providing a framework to preprocess image data for the application of circle detection methods suited for colony counting.

5.3 Generalizing *Candida albicans* Colony Classification using Additional Datasets

In this section, we train and apply the deep learning architectures developed in Chapter 4 on a larger and more diverse dataset of *C. albicans* images. More specifically, we include two additional datasets of *C. albicans* colonies on top of the dataset introduced in Chapter 4. Examples of images from these datasets are shown in Figure C.1. The first additional dataset contains 69 images of plated *C. albicans* colonies growing on CHROMagar media. The second dataset contains 39 images of plated *C. albicans* colonies growing on Sabouraud Dextrose Agar (SDA) media.

For this section, we train the instances of the four neural networks in Chapter 4 without data augmentation in three scenarios: 1) the 15 CHROMagar images with the additional 69 CHROMagar images introduced here (henceforth known as the extended CHROMagar dataset), 2) the 39 SDA images, and 3) the combination of the datasets for the first two scenarios. For each scenario, the training setup, learning rate, loss function and accuracy metrics considered are the same as described in Section 4.2.6.

5.3.1 Performance on the Extended CHROMagar Dataset

A total of 84 images of plates with *C. albicans* colonies growing on CHROMagar media were used here (Example shown in Figure C.1). From these, 60 images are used for training our networks, 16 are set aside for validation, and 8 are for testing the performance of each network following the training process. For the testing set, we extracted 175 white colony images, 20 opaque colony images, 2 sectorized colony images, and 64 background images respectively.

Summary of Model Performance

We plotted precision-recall and ROC curves for each of the colony classes and estimated the area under each curve to obtain qualitative performance metrics for each model (see Table 5.2). All four models show comparative qualitative performance on predicting colonies in the CHROMagar dataset, with each ROC metric at least 0.98 on white, opaque, and background images.

The quantitative performance metrics for each model on each of the image classes are summarized in Table 5.3. All models attained an F1 score of at least 0.97 on their performance for predicting white colonies, with each model attaining a precision of at least 0.98 and a recall of at least 0.96.

Out of the four models trained and tested on the CHROMagar images, the single input Resnet 34 model achieved the highest performance, attaining an accuracy of 97% across the entire test set. The single input Resnet 34 model is the only one of the four that correctly predicted at least one colony to be sectored in the CHROMagar images and has also attained the highest F1 score for correctly predicting opaque colonies.

The dual input toy model has the lowest performance with an accuracy of 0.91 on the entire image set, where the primary penalty to the accuracy is due to the lower prediction accuracy on background images. This model also achieved the lowest recall score (0.57) for opaque colonies, the main contributor to its 0.73 F1 score on opaque colonies. A likely reason for the drop in performance is due to many background images being misclassified as opaque colonies (see Figure C.5).

Table 5.2: **Area under the curve (AUC) metrics for model performance on each class of images in the CHROMagar dataset.** Table of AUC values under the precision-recall (PR) and receiver operating characteristic (ROC) curves for each image class respectively on each of the four models applied to the CHROMagar images.

Model	(PR, ROC) White	(PR, ROC) Opaque	(PR, ROC) Sectored	(PR, ROC) Background
Single Input Toy Model	(0.997, 0.994)	(0.918, 0.991)	(0.026, 0.858)	(0.981, 0.990)
Dual Input Toy Model	(0.984, 0.970)	(0.942, 0.990)	(0.019, 0.811)	(0.878, 0.943)
Single Input Resnet 34	(0.997, 0.994)	(0.927, 0.985)	(0.168, 0.904)	(0.971, 0.989)
Dual Input Resnet 34	(0.997, 0.994)	(0.907, 0.991)	(0.035, 0.831)	(0.977, 0.981)

Table 5.3: **Accuracy analysis of the performance of the four deep learning models on the testing images in the extended CHROMagar dataset.** Table of overall accuracy (Acc), precision (Pr), recall (Re), and F1 scores computed on each colony class across four deep learning models utilized. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Model	Acc	(Pr, Re, F1) White	(Pr, Re, F1) Opaque	(Pr, Re, F1) Sectored	(Pr, Re, F1) Background
Single Input Toy Model	0.96	(0.99, 0.96, 0.98)	(0.95, 0.76, 0.84)	(0.00, 0.00, 0.00)	(0.86, 1.00, 0.92)
Dual Input Toy Model	0.91	(0.99, 0.96, 0.97)	(1.00, 0.57, 0.73)	(0.00, 0.00, 0.00)	(0.70, 1.00, 0.83)
Single Input Resnet 34	0.97	(0.98, 0.98, 0.98)	(0.95, 0.90, 0.93)	(0.50, 0.33, 0.40)	(0.94, 0.97, 0.95)
Dual Input Resnet 34	0.96	(0.99, 0.98, 0.99)	(0.85, 0.89, 0.87)	(0.00, 0.00, 0.00)	(0.95, 0.94, 0.95)

5.3.2 Performance on the SDA Dataset

A total of 39 images of plates with *C. albicans* colonies growing on SDA media were used here (Example shown in Figure C.1). From these, 27 images are used for training our networks, 8 are set aside for validation, and 4 are for testing the performance of each network following the training process. For the testing set, we extracted 335 white colony images, 64 opaque colony images, 12 sectored colony images, and 78 background images respectively.

Summary of Model Performance

We plotted precision-recall and ROC curves for each of the colony classes and estimated the area under each curve to obtain qualitative performance metrics for each model (see Table 5.4). All four models show distinctions in performance across each classes of images in the SDA dataset. The dual input toy model exhibits the best performance on background prediction, but also exhibits the worst performance on predicting images in all other classes. The single input toy model ranks high in terms of qualitative performance between each of the image classes.

The quantitative performance metrics for each model on each of the image classes are summarized in Table 5.5. All models attained an F1 score of at least 0.94 on their performance for predicting white colonies, with each model attaining a precision of at least 0.95 and a recall of at least 0.91.

Out of the four models trained and tested on the SDA images, the dual input Resnet 34 model achieved the highest performance, attaining an accuracy of 91%

across the entire test set. While multiple models achieved relatively similar performance on predicting background images, the dual input Resnet 34 model also achieved the highest F1 score on correctly predicting white and opaque colonies.

The dual input toy model has the lowest performance with an accuracy of 0.87 on the entire image set. Its low performance on the dataset as a whole is likely due to its low F1 scores for opaque and background image prediction, which is the lowest across the four models (see Figure C.9). Another reason for the low accuracy is due to the class imbalance affected by the quantity of white colonies; this model also attained the lowest precision for white colony prediction, with many white colonies misclassified as background. In addition, most of the sectored colonies were misclassified as opaque colonies, thus penalizing the recall of the model on opaque colony prediction (see Figure C.9). The dual input Resnet 34 model had the opposite effect with most sectored colonies being misclassified as opaque colonies (see Figure C.11).

Table 5.4: **Area under the curve (AUC) metrics for model performance on each class of images in the SDA dataset.** Table of AUC values under the precision-recall (PR) and receiver operating characteristic (ROC) curves for each image class respectively on each of the four models applied to the SDA images.

Model	(PR, ROC) White	(PR, ROC) Opaque	(PR, ROC) Sectored	(PR, ROC) Background
Single Input Toy Model	(0.954, 0.943)	(0.928, 0.979)	(0.086, 0.864)	(0.832, 0.908)
Dual Input Toy Model	(0.920, 0.911)	(0.868, 0.956)	(0.055, 0.760)	(0.781, 0.869)
Single Input Resnet 34	(0.960, 0.950)	(0.938, 0.975)	(0.143, 0.833)	(0.863, 0.929)
Dual Input Resnet 34	(0.955, 0.944)	(0.940, 0.979)	(0.059, 0.619)	(0.833, 0.882)

5.3.3 Performance on the Combined CHROMagar + SDA Datasets

Here, we retrain and apply the deep learning architectures developed in Chapter 4 on both the CHROMagar and SDA media images to test whether our networks are able to accurately classify colonies from both media. The partitioning of the images from this data is the same as described in this section. As a result, our dataset is partitioned into 87 images for training, 24 for validation, and 12 for testing. For the testing set, we extracted 510 white colony images, 84 opaque colony images, 14 sectored colony images, and 142 background images respectively.

Table 5.5: **Accuracy analysis of the performance of the four deep learning models on the testing images in the extended SDA dataset.** Table of overall accuracy (Acc), precision (Pr), recall (Re), and F1 scores computed on each colony class across four deep learning models utilized. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Model	Acc	(Pr, Re, F1) White	(Pr, Re, F1) Opaque	(Pr, Re, F1) Sectored	(Pr, Re, F1) Background
Single Input Toy Model	0.91	(0.99, 0.92, 0.95)	(0.92, 0.80, 0.86)	(0.17, 1.00, 0.29)	(0.65, 0.96, 0.78)
Dual Input Toy Model	0.87	(0.95, 0.94, 0.94)	(0.94, 0.59, 0.73)	(0.00, 0.00, 0.00)	(0.60, 1.00, 0.75)
Single Input Resnet 34	0.90	(0.98, 0.91, 0.95)	(0.92, 0.81, 0.86)	(0.08, 0.33, 0.13)	(0.65, 0.96, 0.78)
Dual Input Resnet 34	0.91	(0.99, 0.92, 0.96)	(0.94, 0.83, 0.88)	(0.00, 0.00, 0.00)	(0.67, 0.95, 0.78)

Summary of Model Performance

We plotted precision-recall and ROC curves for each of the colony classes and estimated the area under each curve to obtain qualitative performance metrics for each model (see Table 5.6). With images from both media present, any significant differences in performance overall are less pronounced compared to the models trained on images under one media. Overall, the single input toy model has the highest qualitative performance between each class.

The quantitative performance metrics for each model on each of the image classes are summarized in Table 5.7. All models attained an F1 score of at least 0.95 on their performance for predicting white colonies, with each model attaining a precision of at least 0.95 and a recall of at least 0.94.

Out of the four models trained and tested on the SDA images, the single input Resnet 34 model achieved the highest performance, attaining an accuracy of 92% across the entire test set. While all four achieved relatively similar performance on predicting background images, the dual single Resnet 34 model achieved the highest F1 scores for performance on predicting colonies from all four classes, including sectored colonies.

The dual input Resnet 34 model however has the lowest performance overall. This model achieved the lowest accuracy on white and opaque colonies and thus likely has an accuracy score affected by class imbalance due to the higher quantity of images in both classes. Furthermore, while the single input toy model achieved the lowest recall score for opaque colonies, the precision of the dual input Resnet 34 model for opaque colonies suffered, with approximately 18% of opaque colonies being misclassified as

background (see Figure C.15).

Table 5.6: Area under the curve (AUC) metrics for model performance on each class of images in the combined CHROMagar + SDA dataset. Table of AUC values under the precision-recall (PR) and receiver operating characteristic (ROC) curves for each image class respectively on each of the four models applied to both the CHROMagar and SDA images.

Model	(PR, ROC) White	(PR, ROC) Opaque	(PR, ROC) Sectored	(PR, ROC) Background
Single Input Toy Model	(0.975, 0.967)	(0.911, 0.980)	(0.109, 0.869)	(0.899, 0.942)
Dual Input Toy Model	(0.984, 0.972)	(0.900, 0.975)	(0.094, 0.868)	(0.909, 0.956)
Single Input Resnet 34	(0.971, 0.964)	(0.930, 0.977)	(0.132, 0.822)	(0.919, 0.954)
Dual Input Resnet 34	(0.978, 0.960)	(0.799, 0.966)	(0.171, 0.844)	(0.877, 0.933)

Partitioned Accuracy

Since each of the four models here were trained on both the CHROMagar and SDA images, it then leaves us to analyze the performance of the model on both test sets independently. All four models demonstrate high precision on white and opaque colonies across the board with the exception of the dual input Resnet 34 model (see Table 5.11). Precision and recall scores on background images vary significantly, with CHROMagar background images being classified correctly more often than SDA background images, despite the quantity of CHROMagar background images being less than in SDA. Furthermore, sectored colony predictions appear to be in small number, with undercounting of sectored colonies and no more than two sectored colonies being classified correctly in each model.

5.3.4 Discussion on Extended Data

The results of our pipeline demonstrate the generalizability and robustness of deep learning models applied to larger training sets with additional media. In the case of colony quantification, deep learning integration becomes a very useful tool where human annotation becomes expensive. We have further demonstrated that for SDA images, where colony types are much more difficult to distinguish by eye, that deep learning provides an avenue for faster and comparable colony quantification to that of human annotation.

The issues I encountered in Chapter 4 for classifying sectored CHROMagar colonies are also apparent in the classification of sectored SDA colonies. Furthermore, we see

Table 5.7: **Accuracy analysis of the performance of the four deep learning models on the testing images in the combined CHROMagar + SDA dataset.** Table of overall accuracy (Acc), precision (Pr), recall (Re), and F1 scores computed on each colony class across four deep learning models utilized. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Model	Acc	(Pr, Re, F1) White	(Pr, Re, F1) Opaque	(Pr, Re, F1) Sectoried	(Pr, Re, F1) Background
Single Input Toy Model	0.89	(0.97, 0.93, 0.95)	(0.92, 0.71, 0.80)	(0.14, 0.25, 0.18)	(0.67, 0.99, 0.80)
Dual Input Toy Model	0.91	(0.99, 0.93, 0.96)	(0.88, 0.76, 0.82)	(0.00, 0.00, 0.00)	(0.73, 0.98, 0.83)
Single Input Resnet 34	0.92	(0.99, 0.93, 0.96)	(0.92, 0.86, 0.89)	(0.21, 0.50, 0.30)	(0.76, 0.97, 0.85)
Dual Input Resnet 34	0.88	(0.95, 0.94, 0.95)	(0.73, 0.75, 0.74)	(0.14, 0.18, 0.16)	(0.80, 0.81, 0.81)

a disproportional drop in accuracy for sectoried colony classification across all models despite using additional sectoried colony data. One of the primary reasons for this discrepancy is that the quantity of available sectoried colony images is still insufficient for a deep learning model to recognize the features embedded within these images. Many of these colonies are incorrectly classified as white, suggesting that each model is likely overfitting to the white colony class due to the over-representation of white colony images in the training process. One way to address overfitting when training a neural network is to find a balance between the number of images of each class when designing the training set. However, depending on the abundance of sectoried colonies in the experimental data, this might not be achieved unless the direction is to introduce augmented or synthetic sectoried colony images in the training set or reducing the quantity of images in all other classes. This will further have to be done based on the amount of data available across both media types to also ensure model colony classification performance is similar for images of both media. This idea extends to larger datasets with additional media types not used here; training a deep neural network to recognize colonies across multiple media types is ideal when the same media types are used in both training and testing deep neural networks. However, applying the same neural network more generally will determine if fine-tuning is necessary for improving classification performance on data containing features not present in the original training set.

While in this chapter as well as in Chapter 4 we only considered two types of deep learning models with dual input extensions, they are not in any way the state

Table 5.8: **Accuracy analysis of the performance of the single input toy model on each testing set.** Table of precision, recall, and F1 scores computed on each colony class across all three image sets. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Metric/Images	CHROMagar	SDA	CHROMagar + SDA
Precision			
White	0.97 (170 / 175)	0.98 (327 / 335)	0.97 (497 / 510)
Opaque	0.90 (18 / 20)	0.92 (59 / 64)	0.92 (77 / 84)
Sectored	0.50 (1 / 2)	0.08 (1 / 12)	0.14 (2 / 14)
Background	0.80 (51 / 64)	0.56 (44 / 78)	0.67 (95 / 142)
Recall			
White	0.94 (170 / 180)	0.92 (327 / 357)	0.93 (497 / 537)
Opaque	0.75 (18 / 24)	0.69 (59 / 85)	0.71 (77 / 109)
Sectored	0.17 (1 / 6)	0.50 (1 / 2)	0.25 (2 / 8)
Background	1.00 (51 / 51)	0.98 (44 / 45)	0.99 (95 / 96)
F1 Score			
White	0.96	0.95	0.95
Opaque	0.82	0.79	0.80
Sectored	0.25	0.14	0.18
Background	0.89	0.72	0.80

Table 5.9: **Accuracy analysis of the performance of the dual input toy model on each testing set.** Table of precision, recall, and F1 scores computed on each colony class across all three image sets. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Metric/Images	CHROMagar	SDA	CHROMagar + SDA
Precision			
White	0.99 (173 / 175)	0.99 (333 / 335)	0.99 (506 / 510)
Opaque	0.95 (19 / 20)	0.86 (55 / 64)	0.88 (74 / 84)
Sectored	0.00 (0 / 2)	0.00 (0 / 12)	0.00 (0 / 14)
Background	0.91 (58 / 64)	0.58 (45 / 78)	0.73 (103 / 142)
Recall			
White	0.98 (173 / 177)	0.90 (333 / 369)	0.93 (506 / 546)
Opaque	0.83 (19 / 23)	0.74 (55 / 74)	0.76 (74 / 97)
Sectored	0.17 (0 / 1)	0.00 (0 / 1)	0.00 (0 / 2)
Background	0.97 (58 / 60)	1.00 (45 / 45)	0.98 (103 / 105)
F1 Score			
White	0.98	0.95	0.96
Opaque	0.88	0.80	0.82
Sectored	0.00	0.00	0.00
Background	0.94	0.73	0.83

Table 5.10: **Accuracy analysis of the performance of the single input Resnet 34 on each testing set.** Table of precision, recall, and F1 scores computed on each colony class across all three image sets. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Metric/Images	CHROMagar	SDA	CHROMagar + SDA
Precision			
White	0.98 (172 / 175)	0.99 (331 / 335)	0.99 (503 / 510)
Opaque	0.90 (18 / 20)	0.92 (59 / 64)	0.92 (77 / 84)
Sectored	0.50 (1 / 2)	0.17 (2 / 12)	0.21 (3 / 14)
Background	0.91 (58 / 64)	0.64 (50 / 78)	0.76 (108 / 142)
Recall			
White	0.96 (172 / 179)	0.91 (331 / 364)	0.93 (503 / 543)
Opaque	0.90 (18 / 20)	0.84 (59 / 70)	0.86 (77 / 90)
Sectored	0.33 (1 / 3)	0.67 (2 / 3)	0.50 (3 / 6)
Background	0.98 (58 / 59)	0.96 (50 / 52)	0.97 (108 / 111)
F1 Score			
White	0.97	0.95	0.96
Opaque	0.90	0.88	0.89
Sectored	0.40	0.27	0.30
Background	0.94	0.77	0.85

Table 5.11: **Accuracy analysis of the performance of the dual input Resnet 34 on each testing set.** Table of precision, recall, and F1 scores computed on each colony class across all three image sets. Precision is computed by counting, for all images with a given true label, the number of those colonies correctly predicted to have that label. Recall is computed by counting, for all images predicted to have a given label, the number of those colonies correctly predicted to have that label. The F1 score is the harmonic mean of precision and recall.

Metric/Images	CHROMagar	SDA	CHROMagar + SDA
Precision			
White	0.98 (172 / 175)	0.94 (314 / 335)	0.95 (486 / 510)
Opaque	0.90 (18 / 20)	0.67 (43 / 64)	0.73 (61 / 84)
Sectored	0.50 (1 / 2)	0.11 (1 / 12)	0.14 (2 / 14)
Background	0.95 (61 / 64)	0.69 (53 / 78)	0.80 (114 / 142)
Recall			
White	0.98 (172 / 175)	0.92 (314 / 343)	0.94 (486 / 518)
Opaque	0.86 (18 / 21)	0.72 (43 / 60)	0.75 (61 / 81)
Sectored	0.50 (1 / 2)	0.11 (1 / 9)	0.18 (2 / 11)
Background	0.97 (61 / 63)	0.69 (53 / 77)	0.81 (114 / 140)
F1 Score			
White	0.98	0.93	0.95
Opaque	0.88	0.69	0.74
Sectored	0.50	0.10	0.16
Background	0.96	0.68	0.81

of the art. However, current literature on the use of metadata aimed at supporting image classification model is scarce. As such, our framework opens the possibility of integrating data such as colony size into deep learning models to improve the predictive power of current models.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this dissertation, I presented my research on the detection and analysis of complex microbial colonies using traditional circular detection methods and integrating deep learning for the analysis of lower level features in yeast to study prion protein dynamics in *Saccharomyces cerevisiae*. Using this study as a base, I constructed a framework where modeling and simulation with data-driven methods are able to communicate with one another to drive our understanding of the formation of multiple colony level phenotypes. In addition, I also partially developed a similar framework for the study of the formation of white and opaque colony regions in *Candida albicans*, thus providing a tool for further study driving white-opaque switch events that can be coupled with model-based approaches.

In Chapter 2, my collaborators and I constructed an agent-based model of budding yeast colony growth to study the size and shape of sectors due to budding and nutrient limitation. Our results show that the process of budding has a significant impact on local cell connectivity. Furthermore, when budding is coupled with nutrient limitation, the two biophysical features act to promote the formation of well-defined sector-like structures with highly variable sizes. Such features provide novel insights into the formation of sectorized phenotypes in yeast colonies and offer new windows of interpretation of colony formation through a modeling lens.

In Chapter 3, I constructed a deep learning pipeline called *[PSI]-CIC* for automated quantification of sectorized yeast colonies found in image data. One feature we demonstrated was that synthetic training data resembling the experimental data is integrable to the model training process when insufficient quantities of experimental data are available. Through this approach, we showed that our framework is able to produce accurate colony counting results comparable to human annotation, provided that the output of the deep learning model produces adequate colony segmentations. While other models aimed at plated colonies focus on direct image classification, our framework is the first to consider a deeper level quantification of sector formation in yeast colonies by integrating a traditional edge-based approach. This work opens a

window of opportunity to study the formation of sectored phenotypes from colony level data and together with the agent-based model of growing yeast colonies, offers a framework for making meaningful model-driven inferences about what drives sector formation in yeast colonies.

In Chapter 4, I constructed a second deep learning pipeline aimed at quantifying colonies of *C. albicans* to efficiently identify white and opaque phenotypes from image data. We also demonstrate the performance of deep learning models coupled with additional metadata on colonies extracted from images as an aid for such models to improve colony classification. While our approach on accurately identifying heterogeneous colonies was challenging, our method is able to accurately identify homogeneous white or opaque colonies with considerable accuracy.

In Chapter 5, I also presented progress on using traditional circle detection on colony images of different species and justified when it is feasible to couple deep learning with traditional circle detection for the purpose of colony identification. We demonstrate this idea using a bacterial colony dataset where traditional circle detection fails to capture colonies, but when preprocessed using deep learning methods for image segmentation, significantly improves its applicability. Furthermore, we extended on our work for classifying *C. albicans* colonies using a larger and more diverse dataset that included a second type of media where visual differentiation is extremely difficult through manual approaches. We demonstrate that the deep learning models trained on images of both media are able to accurately quantify white and opaque colonies from both media. This data-driven framework of colony quantification can then be coupled with a model driven approach so that similar meaningful interpretations and insights can be made on what drives phenotype formation in *C. albicans*.

6.2 Future Directions

While this framework for coupling model-driven and data-driven approaches can help provide additional insights into the phenotype formation in yeast colonies, there are still ways to refine this framework to provide additional insights into other features within yeast colonies. While the study of such features lie beyond the scope of this dissertation as it stands, I provide a few directions on how to extend the scope of the work discussed.

6.2.1 Embedding Prion Aggregation into the Center-based Model

In the last couple decades, many models have been proposed for studying processes of prion aggregation in mammals [23, 35, 102, 129]. From these models, the Nucleated Polymerization Model is one of the most widely accepted and used for studying similar aggregation processes in yeast colonies [35]. It is important to note that these models do not make additional assumptions about the domain at which these processes occur, so it is unclear how space plays a role in the production of

aggregates in either mammals or yeast. More recently, prion aggregation models were extended to networks in order to study the spread of Alzheimer’s disease in human brains [54]. While Alzheimer’s disease is not classified as a prion disease due to non-transmissibility, an argument is that since proteins involved with the disease act in a prion-like manner [77], the same models can be adapted to study prion propagation in yeast colonies. Furthermore, recent work by Lemarre et al. [92] proposed a model of prion aggregation in yeast with aggregate transmission bias between mother-daughter cell pairs and hypothesized that this bias contributes heavily to $[PSI^+]$ curing. However, the cumulative effect of this bias toward sector formation in yeast colonies on a larger scale has not been investigated in detail.

One avenue of research would be to extend the cell-based model detailed in Chapter 2 using a method that simultaneously addresses the lack of quantitative colony-level information in the literature on the formation of sectorized phenotypes in yeast colonies and the impact of budding on prion aggregation between mother-daughter cell pairs. Using the framework of Fornari et al. [54], it is possible to represent colonies as a dynamic network of cells where individual cells would be represented as nodes, and physically attached mother-daughter cell pairs are joined by edges (see Figure 6.1). This network representation of a colony is independent of prion aggregation, which allows us to embed a model of prion aggregation per cell without affecting the structure of the network. For mother-daughter pairs joined by edges, existing prion aggregation models can be adapted to include transmission of aggregates between cells. Lemarre et al. [92] modeled aggregate concentrations using impulsive differential equations where the concentrations in the mother and daughter cells change discontinuously at the time of detachment. While models of prion aggregation exist for single and budding cells [38, 70], they have not been applied to large-scale colony models, but since budding is explicitly included in our center-based model from Chapter 2, it is necessary to embed a model of prion aggregation where transmission is possible throughout the entire period of budding division. Furthermore, a direct data-driven application of this research would be to develop a method that takes this center-based model with embedded prion aggregation and tests how emergent sectoring patterns in this model closely resemble sectoring patterns in experimental yeast colonies. Such work will allow researchers to create meaningful comparisons between the model and entire yeast colonies from experimental data that will give us valuable insight into the mechanisms driving the loss of the prion phenotype in yeast.

Possible Challenges

Since our cell-based model explicitly includes budding as a mechanism, one issue that will arise is how to adapt a model of prion aggregation specifically to mother-daughter pairs. The difficulty is due to changes in aggregate concentrations as a result of transmission between two cells of varying size, one of which is growing rapidly while attached. One way we can address this is by modeling aggregate concentration in both cells simultaneously. Fornari’s network approach [54] allows for aggregates to diffuse across connected nodes, and Heydari’s method allows for aggregate diffu-

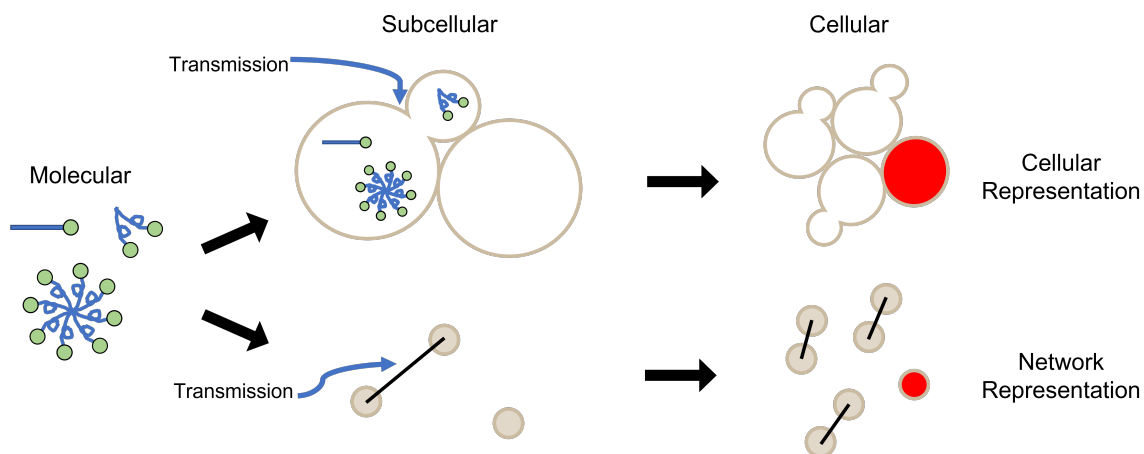


Figure 6.1: **Representing mechanisms of prion aggregation at multiple scales.** A model for prion aggregation at the molecular level is chosen first. At the subcellular level, this model must be modified to account for transmission between two attached cells. Using the cellular representation of a yeast colony (top), we can represent the colony as a network where nodes represent cells and attached mother-daughter pairs are connected by edges. The same model of prion aggregation is applied to each node, but each mother-daughter pair connected by an edge will have an additional transmission factor accounting for aggregates moving between cells.

sion between mother-daughter pairs in three dimensions [70]. By modifying these approaches for modeling aggregate transmission between attached mother-daughter pairs in our center-based model, it is possible to establish a bridge between single- and multi-cell aggregate dynamics.

Another issue that will arise is predicting when the first prion loss event will occur between dividing cells. Assuming that the initial cell has a white phenotype, even if all the parameters of a prion aggregation model remain constant and are equal for all cells, varying division times add a layer of complexity to predicting the number of divisions needed to produce the first cell with an opposing phenotype. If the number of divisions required to produce this first loss event is significantly large, this will also prove to be a computational challenge since the total number of cells will also be significantly large. Before analyzing sector-like structures at a colony level-scale, the issue of predicting when the first loss event will occur by modeling protein concentrations of cells within a single lineage should be addressed. Lemarre et al. [92] has done this using their proposed model, but it is unknown if for other models the number of generations to the first loss event is significantly different.

6.2.2 Deep-Learning for Sectorized Colony Image Classification

In Chapter 3, I implemented a pipeline designed to quantify colonies and their sectors respectively. Through trial and error, I found that the application of this pipeline had disadvantages due to the features inherent in the image set and the dependency of diverse training data needed for a deep neural network to learn these features. Here, I propose a couple ways where deep-learning methods can be used to resolve these disadvantages and provide a basis for quantifying colony structure directly from experimental image data.

In Chapters 3, 4, and 5, I applied a traditional circle detection method for counting yeast colonies that can disambiguate clusters of multiple colonies. Unfortunately, I found that this method is not effective at counting sectorized colonies within our image set. As such, I turned to deep learning to overcome this obstacle. However, one realm of complexity in the experimental images is the density of colonies present specifically clusters of multiple colonies. As clusters become more dense, it becomes more difficult to count colonies manually within the cluster. Similarly, traditional circle detection becomes an issue when there is a lack of information about the shape of individual colonies within a cluster, leading to under-counting of colonies present. As such, cluster disambiguation is a problem where deep learning is capable of providing a solution (see Figure 6.2 A). A couple methods have already been developed for the application of counting clustered objects. First, Ferrari et al [52] proposed a deep learning method which located all bacterial colony clusters on a plate and attempted to count the number of bacteria colonies in each cluster. Overton [122] proposed a deep learning architecture called DO-U-Net to count tents from satellite imagery and blood smears, both of which have complex morphologies. Both of these methods however have not been tested for counting yeast colonies with multiple phenotypes, but I argue this method is still applicable to our image set because clusters of multiple colonies can adopt complex morphologies just like the blood smear images in the Overton study. Therefore, it is possible to re-purpose the approaches of Ferrari et al [52] and Overton [122] simultaneously to improve image segmentation of sectorized colony images and deep-learning enabled colony counting (see Figure 6.2 B).

The pipeline I proposed in Chapter 3 relies on the output of U-Net in order to predict the number of sectors present in a colony, so at present no deep-learning step is currently implemented for specifically counting sectors. Therefore, a future direction is to modify *[PSI]-CIC* by replacing the sector quantification step with a machine learning image classifier in order to eliminate the dependency on very specific colony segmentation data. Two ways to implement this are the following: One way is to adapt Ferrari's approach [52] of disambiguating bacterial clusters to the problem of yeast sector counting, as both are analogous to one another. Another way is to adapt the approach of Carl et al. [25] where the classification step is re-purposed for sector counting over color labeling. By using a deep-learning based method for colony classification, every major component of the pipeline will be consistently data-driven, enabling analysis of sectorized yeast colonies that relies solely on training data.

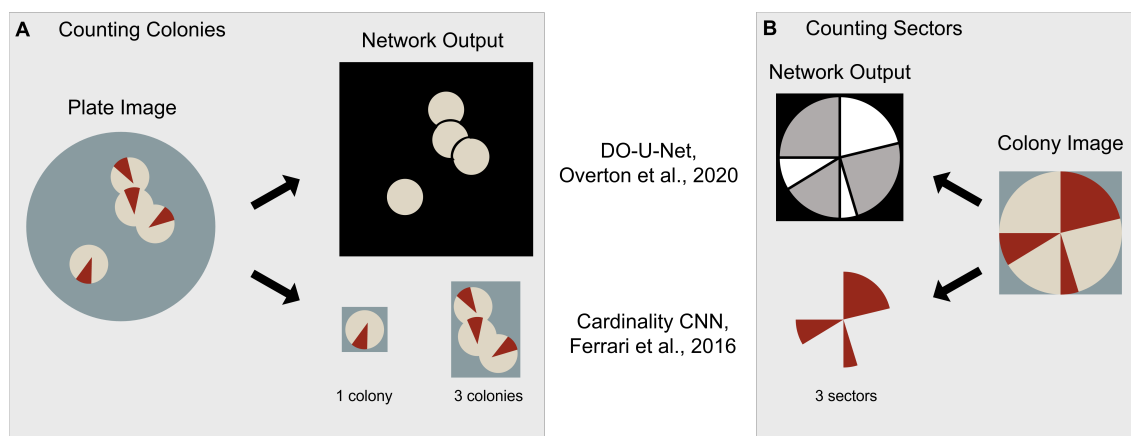


Figure 6.2: **Expected output of using deep-learning methods for colony analysis.** For images of plates and individual colonies, the expected output of Overton’s DO-U-Net [122] and Ferrari’s CNN [52] are shown.

A: Expected network output for counting colonies when a plated image of multiple colonies is provided as input. With Overton’s network, colony pixels and non-overlapping colony regions are assigned one label while others are labeled as background pixels. This separates clusters, allowing for more accurate colony counting. With Ferrari’s network, colony clusters are disambiguated by estimating the number of colonies they comprise.

B: Expected network output for counting sectors when an image of an isolated colony is provided as input. Overton’s network can be modified to place the interface between regions of different phenotypes into the background. Ferrari’s network can be modified to count sectors directly from the image.

Finally, I propose to combine this pipeline with our center-based model with prion aggregation for the purpose of estimating parameters of the embedded prion aggregation model. Each plated image utilized in our pipeline contains anywhere between 80 and 200 colonies; these individual colonies offer us a rich dataset for training neural-networks to perform model parameter estimation. This data will assist in determining whether the appropriate prion aggregation models can explain colony-level sectoring behavior due to prion loss in yeast.

Possible Challenges

One issue that will arise with respect to colony counting is the adaptability of deep-learning approaches to sectored yeast colonies. The performance of the methods by Ferrari and Overton [52, 122] have not been tested on yeast colony images, so it is not known how well these methods will perform on the images we have used throughout this dissertation. However, because these methods work with more complex morphologies than the colonies in our images, we do not expect a significant degradation in performance.

Fitting a model of prion aggregation to study the emergence of sectored phenotypes in general will be difficult due to the size of the parameter space, the variability of parameters, and limited studies to validate a subset of parameters. For example, the Nucleated Polymerization Model has four parameters that describe the rates of protein synthesis, conversion, aggregation, and fragmentation respectively. It will be necessary to fix a subset of the parameters using values from prior literature [159] while allowing others to vary. Furthermore, the parameter values of the model pertaining to one set of colony images may not necessarily apply to another set of colony images, especially when colonies in both image sets are grown under different experimental conditions. To simplify this, it will be best to focus efforts toward fitting a model for colonies of one type of strain. Once the model is appropriately capturing sectoring behavior in these colonies, we can make small modifications to the model to adapt it for other yeast strains.

6.2.3 Developing a Center-Based Model of *Candida albicans* Colony Growth

To ensure a complete construction of the framework to compare simulated colony growth with experimental output, it is necessary to construct a model that accurately simulates the growth of *C. albicans* colonies. Simply put, such a model has not been developed at a large scale. In Chapter 4 our experimental data shows that the sizes of opaque colonies tend to be larger than white colonies. We can link this observation to the cell-level experimental data and hypothesize that opaque cells occupy more space than white cells because of their elongated shape. A model to capture this behavior while also preserving the intercellular dynamics within the colony would aim to test this hypothesis and also help researchers compare its results to current theories on the growth of white and opaque regions of *C. albicans*.

Possible Challenges

Similar to the *S. cerevisiae* center-based model, one issue that will arise is the computational complexity with simulating large colonies. Another issue is utilizing realistic biophysical behaviors to model white and opaque cells, including the switching events between the two phenotypes. As indicated in previous studies, while white cells appear mostly round, opaque cells appear elongated with varying aspect ratios. As such, simulating the shape of a cell will be a critical component to modeling a growing colony structure. The most complex issue to resolve in a simulation of *C. albicans* colonies will be how to model the reproductive process. In *C. albicans* the rate of white-opaque phenotypic switching is closely connected to cell mating [10]. White cells must switch to the opaque state in order to facilitate mating with another cell. The process is mostly heterothallic where two diploid cells fuse to form a white tetraploid cell which subsequently sporulates into multiple diploid cells to increase the population of the colony. As more is revealed about the biophysical behavior of cell mating, constructing a model which integrates cell mating spatially will be an interesting avenue of research for uncovering how such mechanisms govern colony structure over time.

6.2.4 Final Thoughts

While these are only a few examples of extensions to this work that can be explored, many of these are based off the scope of this current work. The integration of approaches combining modeling and simulation with data focused tools opens many avenues of interdisciplinary collaboration to solve complex problems behind the spread of prion disease using yeast colonies as a model system. However, we stress the use of modeling frameworks where data and simulation approaches complement one another, and with the technology available such frameworks are much more feasible today. This dissertation provides many substantial contributions to the fields of mathematical biology and machine learning through building methods and tools for gaining insight into multiscale processes occurring in microbial colonies.

Appendix A

Data Curation and $[PSI]$ -CIC Implementation

A.1 Image Acquisition and Pre-processing

Exponentially growing cultures of the yeast *Saccharomyces cerevisiae* strain 74D-694 *MATa*: *ade1-14*, *trp1-289*, *his3Δ-200*, *ura3-52*, *leu2-3*, *112*, $[PSI^+][PIN^+]$ were subjected to heat shock at 40°C for 30 minutes by water bath to induce curing. Approximately 500 cells were then plated onto YPD-Cox media (0.25% yeast extract, 1% bactopectone, 2% agar, 4% glucose) and grown for 3 days at 30°C followed by 5 days at room temperature to allow colony pigmentation to develop. Images of plates were acquired using an Epson V370 scanner.

A total of 11 images of different plates were acquired and used to test $[PSI]$ -CIC. Image set 1 (plates 1-5) contains five images with one plate per image each containing up to approximately 500 colonies (example in Figure A.1 (left)). All the colonies in these images are either white $[PSI^+]$, red $[psi^-]$, or sectored phenotype (a mix of both $[PSI^+]$ and $[psi^-]$). One of these five plates contains a large number of colonies with sectored phenotypes. Image set 2 (plates 6-11) contains six images which are similar to those in image set 1 (example in Figure A.1 (right)), but these images are less saturated overall and four of these plates contain a significant number of sectored colonies present. These images were pre-processed before testing.

Colonies in each image were hand annotated and sectoring of each quantified by a yeast biologist. Colonies appearing entirely white or red were scored $[PSI^+]$ or $[psi^-]$ respectively. If a mix of red and white pigment was present in a colony it was scored as sectored. Colonies too small to reasonably determine the presence or absence of sectoring were deemed unquantifiable and not considered in our results. If the boundaries of multiple colonies intersect each other in a cluster extensively enough such that half the colony volume is shared, the entire cluster is deemed unquantifiable.

Both image sets 1 and 2 were used for different experiments at different times. One important feature to note across image sets 1 and 2 is variation of color and lighting conditions. Since U-Net is trained on synthetic images whose color is based off the

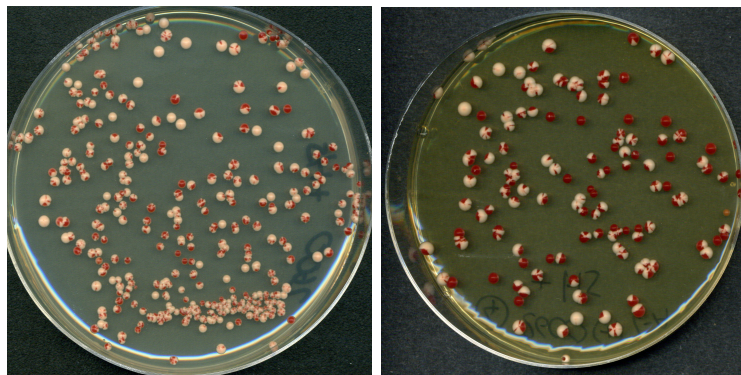


Figure A.1: **Yeast colony images.** Example images of yeast colony plate 2 from image set 1 (left) and plate 8 from image set 2 (right).

images in set 1, U-Net may not accurately segment colonies from image set 2 because by eye the color profiles are different from what U-Net was trained with. Instead of retraining U-Net to address this issue, we opt toward pre-processing the real images until they appear close to a “standardized” image. We use an implementation of a color profile transfer scheme written by [28] and adapt it for execution on Google Colaboratory. This code is an implementation of the work by Reinhard et al [133] which transforms a source image by applying onto it the color characteristics of a desired “target” image. The objective for this pre-processing step is to ensure that the images in set 2 have similar color features as the images in set 1 so that U-Net will produce similar quality output segmentations.

For the purpose of this work, we chose the target image to be the image of plate 2 in image set 1 (also shown in Figure A.1 (left)). All six images in set 2 (plates 6-11) were used as source images for the color transfer scheme before input to U-Net. No pre-processing was done on image set 1 because these images have the color profiles that U-Net was originally trained on. No adjustments in brightness and contrast were applied to these images before or after the color transfer scheme was applied. While there are subtle differences between the color profiles in the original and pre-processed images in set 2, their output segmentations are significantly different. In particular, the segmentation of the preprocessed images display obvious quality improvements such that many more colonies could be sufficiently discerned for classification. Most of the colonies present in the output segmentations were sufficient enough for the classification scheme as described in Section 3.2.1.

A.2 Synthetic Image Generation

Due to the lack of hand annotated colony images, we turn to training a neural network with synthetic images where it is possible to efficiently create ground-truth masks labeling each pixel. An example of a synthetic image generated with its corresponding ground-truth mask is shown in Figure A.2. This approach involves generat-

ing sets of synthetic images of yeast colonies which exhibit key features of the yeast colonies found in the experimental images, which comprise of colonies with sectored red and white regions where the color of each slightly vary. We use two representative colors for the colonies—1 red and 1 white color—to fill each circle representing the colony and the overlying sector. Similarly, we use three representative colors for the background corresponding to the interior of the plate, the border of the plate, and the table on which the plate rests respectively and fill each of these regions with those colors. Each color selected corresponds to an RGB vector $[R, G, B]$ such that $R, G, B \in [0, 255]$.

For each synthetic image, two representations as well as five masks are generated, each with size 1024×1024 . The two representations of each image include one with Poisson noise and one without. The images containing Poisson noise are used for training U-Net in Section 3.2.2, while the images without Poisson noise are to simplify the process for creating the associated ground-truth masks. The five masks created label 1) the colony pixels, 2) the white colony pixels, 3) the red colony pixels, 4) the red and white colony pixels merged, and 5) the number of sectors in each colony. The first three masks are created through a series of grayscale conversions and binary thresholding operations on the image at intermediate steps of the process. The fourth mask is used as the ground-truth mask for training U-Net, while the fifth mask is used to assess the accuracy of $[PSI]$ -CIC in quantifying the frequency of sectors in each colony (see Section 3.2.2).

For each synthetic image, the process for creating the noisy/noiseless representations and ground-truth masks is as follows:

1. We first initialize the image by changing the color of the background represented by the RGB vector $[54, 54, 68]$. This element represents the tabletop at which the plate rests.
2. A circle of radius 30 whose center coincides with the image center is generated above the background and filled with the color represented by the RGB vector $[137, 155, 160]$. This element represents the body of the plate.
3. 100 points are sampled inside the circle generated in step 2 such that the minimum distance between any two points is at least 2. Then, circles of radius 1 are generated whose centers coincide with the sampled points. Each circle is then filled with the color represented by the RGB vector $[221, 217, 199]$. These elements represent the colonies on the plate.
4. Two circles of radius 29 and 31, each with the same center as the circle generated in step 2 are generated. The space in between the circles is filled with the color represented by the RGB vector $[105, 107, 152]$. This element represents the part of the background corresponding to the border of the plate.
5. An image of size 1024×1024 is saved temporarily. Then, binary thresholding is performed on the resulting image following a grayscale transformation. The result is the final ground-truth mask representing colony pixels.

6. For each circle generated in step 3, two points are uniformly selected on the circle, and lines connect those two points independently with the center of the circle. The space in between is filled with the color represented by the RGB vector [148, 36, 23]. This element represents the red region of a colony. For circles where n sectors will be generated, $2n$ points are uniformly selected, and the process described here is performed for each pair of points along the length of the circle.
7. Step 4 is repeated to regenerate the border above the colonies.
8. An image of size 1024×1024 is saved. The result is the noiseless representation of the synthetic image.
9. Binary thresholding is performed following a grayscale transformation on the image from step 8. The result is the final ground-truth mask representing white colony pixels only.
10. The white colony mask from step 9 is subtracted from the full colony mask in step 5. The result is the final ground-truth mask representing red colony pixels only.
11. Since the red colony pixel and white colony pixel masks are fully disjoint, we merge the two masks, assigning one label to white colony pixels and a different label to red colony pixels while all background pixels are labeled 0. The result is the final ground-truth mask showing the locations of red and white colony pixels and is used for training U-Net.
12. An additional mask is created at the center of each colony which shows a small square whose label is the number of sectors generated plus 1. The result is saved as a mask of size 1024×1024 . This represents the true labels for the frequency of sectors in each colony within a synthetic image and is used to assess the performance of *[PSI]-CIC* (Section 3.2.3).
13. Finally, the noiseless image saved from step 8 is given Poisson noise, then saved with size 1024×1024 . The result is the noisy representation of the synthetic images that is used for training U-Net (Section 3.2.2).

A.3 Colony Extraction

Implementation of the steps to locate colonies as described in Appendix A.4 is done using the Python package `oct2py` [146] to allow Octave to run within the environment. Octave’s function `imfindcircles` is used to implement the circle Hough transform [73] for locating circular objects in the output segmentations. This function requires two additional parameters: a range of radii of circular objects to detect, and

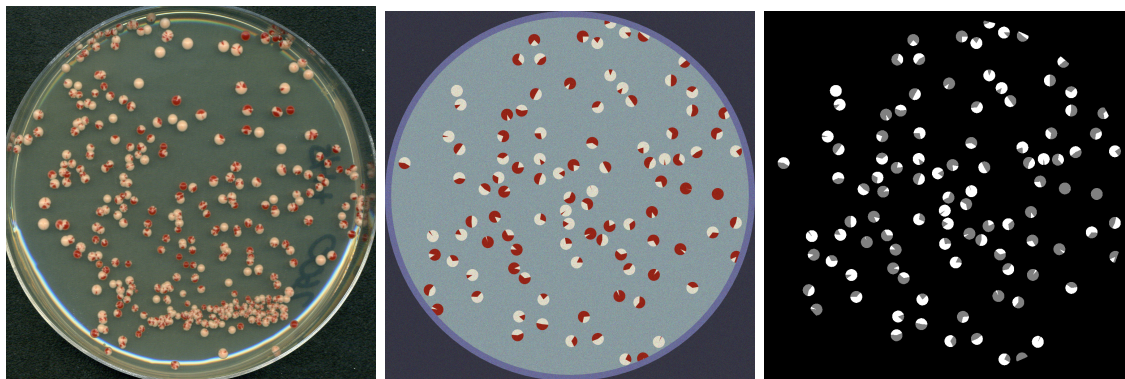


Figure A.2: **Synthetic Representation of Experimental Images.** (Left): Experimental image of yeast colonies with both red and white phenotype obtained with permission from the Serio lab. (Middle): Synthetic image of yeast colonies with both red and white phenotypes generated using Matlab. (Right): The ground-truth mask indicating the label of each pixel in the synthetic image. Background pixels are black, red colony pixels are gray, and white colony pixels are white.

a sensitivity to allow for the detection of objects with slight circular imperfections. To explore the variability in colony sizes within the experimental images, we first find all connected components of colony pixels in their output segmentations, then for each connected component, we locate clusters corresponding to isolated colonies and estimate their radii individually. To do this, we place a bounding box around each connected component separately. Here, we make the assumption that a connected component in the segmentation corresponds to an isolated colony if it meets the following conditions:

1. The connected component must have a number of pixels between a minimum and maximum value. In our case, we require all connected components to have between 100 and 2000 pixels. This is a way to filter colonies that are too big or too small.
2. The bounding box of the connected component must have an aspect ratio close to 1. In our case, we required the greater ratio between length and width of the bounding box to be less than 1.2 to account for image compression and imperfections in the circularity of colonies in the output segmentation. This is also a filter for removing most clusters of colonies from consideration, especially those whose colonies appear to be co-linear.
3. The proportion of pixels within the bounding box consisting of either red or white colony pixels must be between a minimum and maximum value. In our case, we required that the proportion of pixels inside the bounding box to be between 0.7 and 0.9 which contains $\pi/4$, the ratio between the area of a circle and smallest enclosing square respectively. This helps remove colonies whose circularity is insignificant or are too close to the border of the plate.

The collection of radii is used to estimate a range of radii to use for detecting circular objects in the entire segmented image. Since `imfindcircles` strongly recommends that circular objects have a radius of at least 5 pixels, we also set an arbitrary minimum of 7 pixels for the radius of circular objects. If the minimum dimension of any bounding box is less than 7, we temporarily rescale the entire segmentation so that the smallest dimension of any bounding box is 7, before using `imfindcircles`, using the range of radii for the objects in the scaled image. The sensitivity parameter of `imfindcircles` is set to 0.9 to allow adequately imperfect circular objects in the output segmentations to be detected. Following the implementation of `imfindcircles`, the radii, center coordinates, and the coordinates of the bounding boxes are recorded and saved in CSV files. If rescaling was done prior to recording this data, the data is rescaled so that it corresponds to the original sized segmentation. Finally, the region within each bounding box is cropped from the image and saved as a separate image for classification.

A.4 Implementation

A total of 200 synthetic images and corresponding ground-truth masks were created in MATLAB for use as training data, where 150 of these images are used directly in training and 50 were set aside for validation. For each image, 100 non-overlapping white circles representing colonies were placed within the region representing the plate, then one red sector was placed above every colony in the image. Specific details about the placement of colonies and sectors and generation of ground-truth masks are described in Appendix A.2. For the purpose of classification, each of the 20,000 colonies across all synthetic images were labeled to have exactly one sector. All images and ground-truth masks were saved as PNG files.

The remainder of [PSI]-CIC is implemented in an interactive Python notebook with GPU access using Google Colaboratory. Construction of the U-Net architecture was implemented and compiled using the Keras packages in Tensorflow. The network is trained using the synthetic images of size 1024×1024 , with a batch size of 1 due to the size of the images used and the amount of computational memory available. We use Tensorflow's categorical cross-entropy loss function and Adam optimizer. The number of epochs was not predetermined; instead, training stopped when the validation loss decreased by at most 0.001 over a period of 5 epochs. This is a helpful check to prevent the model from overfitting to the image set. The learning rate is initially set to 10^{-4} , but as the validation loss decreases and reaches a local minimum, the learning rate decreases by a factor of 10, with the minimum learning rate possible being 10^{-6} . Segmentation accuracy for each image is computed to be the number of pixels whose labels match their corresponding ground-truth labels divided by the total number of pixels in the image, while accuracy over the entire image set is the mean of the individual accuracies.

After each epoch, a check is performed on the validation images to determine if the segmentation accuracy is higher than in the previous epoch; if the accuracy is higher,

the new parameters are saved, which could be used as a checkpoint for future training of U-Net. When U-Net is sufficiently trained to segment red and white colony pixels in the synthetic images, we apply it to produce output segmentations of the colonies for each experimental image whose pixels are assigned one of three labels (red, white, or background). The segmentations of each plate are saved as individual PNG files.

Following segmentation of the images, steps for locating colonies are implemented using `oct2py` [146] in Python which enables the use of Octave functions. The circle Hough transform is done using the Octave function `imfindcircles` found within the `image` package and is used to detect circular objects in the resulting output segmentation consisting of clusters of colony pixels (both red and white). The specific use of `imfindcircles` for locating colonies is described in detail in Appendix A.3. Objects detected close to the edges of the image are filtered out. Information about the size and location of the extracted objects are saved as CSV files, with one file per image. Proposed regional annotations for each colony extracted are constructed and qualitative classes are assigned to each colony as described in Section 3.2.1.

Appendix B

Image Acquisition for Candida Pipeline

B.1 Experimental Images

Strains were streaked out on YPD agar and incubated at 25°C for 4 days. Single white colonies were picked and inoculated into YPD broth and grown at 25°C overnight. The optical density of each overnight culture was measured using 600nm light. Each culture was serially diluted using 1xPBS. Approximately 100-200 cfus were spread onto CHROMagar plates and incubated at 25°C for at least 3 days before viewing.

Images of the plates were taken at least 3 days after plating. A total of 15 images were acquired. Each image has one plate containing up to 300 *C. albicans* colonies.

B.2 Data Augmentation

Due to the low quantity of non-white colony images, we opt to use data augmentation applied to the colony images we originally extracted from the images to both increase the size of the colony dataset and have each architecture learn from slightly modified versions of the same colonies. We opt to perform data augmentation for balancing the quantity of images corresponding to each colony type. More specifically, we wanted to ensure that the image set contains an equal number of images from each colony type, including background images.

We apply data augmentation by applying a set of transformations to randomly selected colony images in the training dataset as constructed in Section 4.2.6. For each image added, we randomly select with replacement an image from the original pool of the same class, apply a series of transformations, and save the transformed image of the same size into the enhanced training set. Each image is applied the following transformations each with probability 0.5 of occurring: Horizontal flip of the whole image, vertical flip of the whole image, transpose of the whole image, rotation of the colony region, and random rotation of the whole image by up to 90 degrees from the

original orientation. The labels and sizes corresponding to each of the original images remained unchanged throughout all transformations. All image transformations were performed using the `albumentations` package in Python.

Since white colonies are the most abundant, no white colonies underwent augmentation. Images are augmented to create a total of 3620 images, with 905 images for each of the four classes.

B.3 Implementation

The framework for the neural network models was written in Google Colaboratory which is a remote version of Jupyter operated by Google. Each of the models discussed in Chapter 4 are built from the Keras modules within Tensorflow. The implementation of the circle Hough transform was done using the `oct2py` package which interfaces between Octave and Python to run Matlab functions. All post-analysis is done in Google Colaboratory using standard Python packages.

Each of the models run for 100 epochs with the same learning rate (10^{-3}), same Adam optimization method, and the same categorical focal cross-entropy loss function. Final validation accuracy is obtained by evaluating the model with the images as input.

Appendix C

Image Acquisition for Other Colony Images

C.1 Additional Experimental Data

Approximately 50 images of plated *S. cerevisiae* colonies growing on agar were acquired from the University of Massachusetts, Amherst. An additional five images of bacterial colonies were acquired from the University of California, Merced. Examples of these images are shown in Figure C.1 (top).

An additional 69 images of plated *C. albicans* colonies growing on CHROMagar and 39 images of plated *C. albicans* colonies growing on SDA were acquired from the Nobile and Hernday Labs respectively at the University of California, Merced. Examples of these images are shown in Figure C.1 (bottom).

C.2 Synthetic Bacterial Colony Image Generation

To train neural network architectures to segment the bacterial colonies as shown in Figure C.1 (top right), we modify our approach from Appendix A.2 to create synthetic images and corresponding ground-truth masks depicting the locations of colony pixels. An example of a synthetic image generated with its corresponding ground-truth mask is shown in Figure C.2. To remove the need to apply binary thresholding on a synthetic image to generate a ground-truth mask, we reverse the process described in Appendix A.2 and instead create the ground-truth mask before the image is generated. We detail the generation of the features of the synthetic images in the following subsections.

C.2.1 Colony Location and Size

A collection of 200 synthetic images of plates were created exhibiting similar features observed in the experimental images. The process for generating the information on the location and size of each colony in the images is as follows.

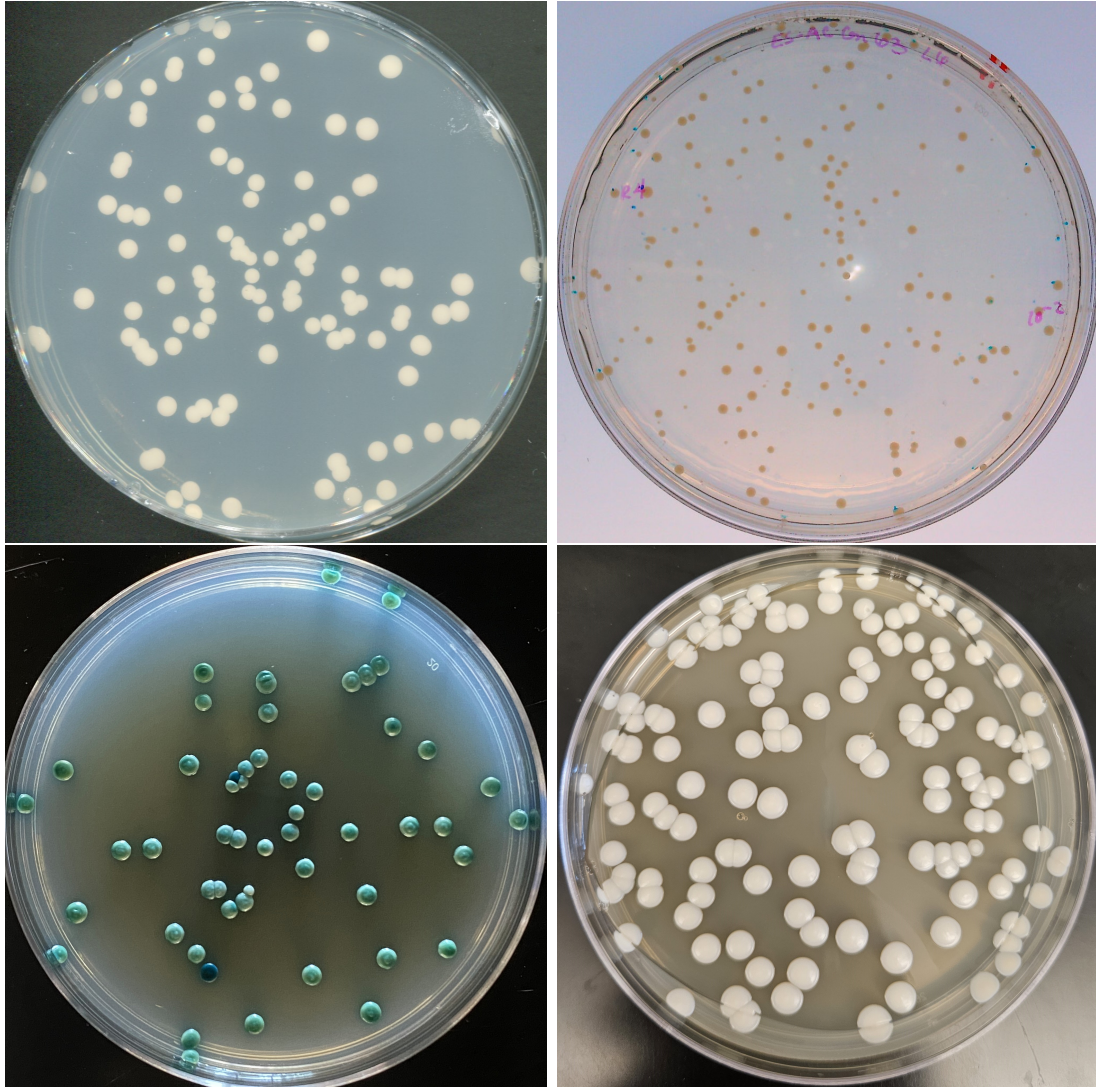


Figure C.1: **Additional Image Datasets.** One example from each of the microbial colony datasets being used in Chapter 5. Top: The efficacy of the circle Hough transform and inclusion of U-Net into colony counting for $[PSI^+]$ colonies (left) and bacterial colonies (right). Bottom: The generalizability of our CHT + image classification network detailed in Chapter 4 which looks at additional images of colonies on CHROMagar (left) and additional colonies on SDA media (right).

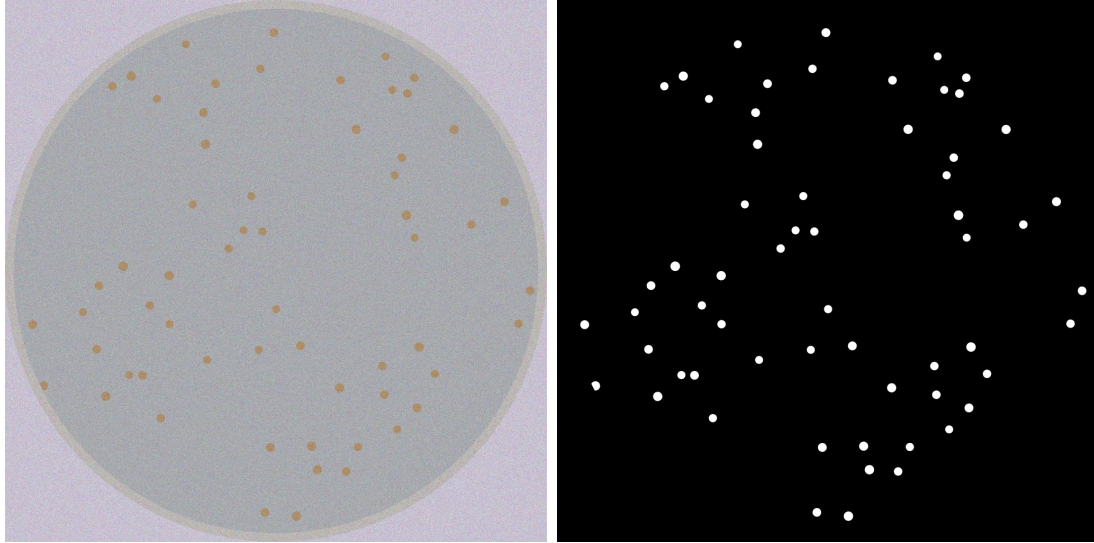


Figure C.2: **Synthetic bacterial colony image with binary ground-truth mask.** (Left): Example image generated in Matlab containing synthetic bacterial colonies of varying sizes on a single plate. (Right): Corresponding binary ground-truth mask indicating the locations of colony pixels in white and background pixels in black.

A circle of radius 30 is generated whose center corresponds to the center of the image. Each image contains a total of 60 synthetic colonies generated inside the circle of radius 30. For each colony generated, a circle of radius 0.5 ± 0.1 is generated, where the center of this circle is chosen uniformly inside the circle of radius 30. Rejection sampling is performed such that the colony location and radius is accepted if the colony is contained in the circle of radius 30. For each subsequent colony generated, we enforce colonies to be non-overlapping and be at least a certain tolerable distance apart from any other colony in the image. Rejection sampling is performed such that the center and radius of the colony being generated are accepted if the minimum distance between any other colony center with corresponding radius is greater than the tolerable distance.

We formalize the two global conditions for rejection sampling of a colony as follows. The first condition is that the colony must be contained within the circle of radius 30, i.e.,

$$\|\vec{c}_i\| + r_i < 30 \quad \forall i, 1 \leq i \leq 60. \quad (\text{C.1})$$

The second condition is that the distance between any pair of colonies on the plate must be at least a certain distance apart, i.e.

$$\|\vec{c}_i - \vec{c}_j\| - (r_i + r_j) > \delta \quad \forall 1 \leq i \leq 60, \quad 1 \leq j \leq 60, \quad i \neq j, \quad (\text{C.2})$$

where δ is the minimum distance tolerance between colony edges.

Colony locations and radii are sampled one at a time to ensure they meet these two conditions. We can iteratively generate a sequence of accepted centers and radii through the following process. Assume that k colonies with centers and radii have been accepted. Therefore, the center and radii of colony $k + 1$ will be accepted if and only if the following two conditions hold. First, the colony must be contained inside the circle of radius 30, i.e.,

$$\|\vec{c}_{k+1}\| + r_{k+1} < 30. \quad (\text{C.3})$$

Second, colony $k + 1$ must be at least δ Euclidean distance units away from the other k colonies accepted, i.e.

$$\delta < \min_{1 \leq i \leq k} (\|\vec{c}_i - \vec{c}_{k+1}\| - |r_i - r_{k+1}|) \quad (\text{C.4})$$

Colony center \vec{c}_{k+1} and corresponding radius r_{k+1} will be accepted if and only if the two conditions are satisfied, and will be rejected if either condition is not met. If a colony location and radii is rejected, a new location and radii is sampled to replace them and the same checks are performed once more. The process will continue until 60 colony centers and radii have been accepted.

C.2.2 Mask Generation

We first initialize the ground-truth mask by changing the color of the background to black. Next, the 60 circles with corresponding centers and radii accepted in the previous subsection are generated and filled white. To account for colonies obscured in the border region, we generate a region in the mask that hides colonies near the border of the plate. To do this, we create a filled annulus with inner radius 29 and outer radius 31 with the same center as the circle with radius 30 and force all pixels inside the annulus to be black. As a result, any given pixel in the ground-truth mask is white if it is both inside a colony and outside the annulus, and black otherwise. The final result is saved as a binary image.

C.2.3 Color Selection

Regions within a colony, within the plate, within the border and within the table underneath the plate are sampled and cropped from the full images (see Figure C.3). The color information from the crops is concatenated into a series of $[R, G, B]$ values, one vector for each pixel in the sample crops for each type of region respectively. For each of the four components, a matrix is created with three columns representing the three color channels and the number of rows equal to the number of images we wish to generate.

Three-dimensional Gaussian distributions were fitted independently to each of the four sets of RGB values obtained from the crops. For each image, we sample one color independently from each of the four distributions and round the values in each

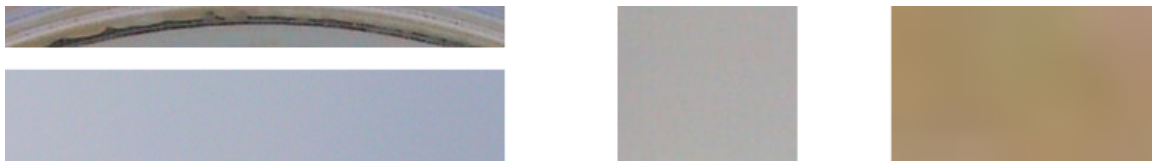


Figure C.3: **Sampled Regions in the Bacterial Colony Images.** Examples of regions extracted from the testing images to be used. (Left): Samples of the border and table regions respectively. (Middle): Sample of a plate region without any colonies present. (Right): Sample of a bacterial colony without the plate visible.

component so that all numerical values are integers between 0 and 255 for each color channel.

For each image, we wish to use sets of colors sampled from the Gaussian distribution such that each color sampled has a Mahalanobis distance of less than 1.38629. Assume we have a multivariate Gaussian distribution $N(\vec{\mu}, \Sigma)$ with mean vector $\vec{\mu}$ and covariance matrix Σ . The Mahalanobis distance d_M between a point \vec{x} from a distribution $N(\vec{\mu}, \Sigma)$ is defined by

$$d_M(\vec{x}, N(\vec{\mu}, \Sigma)) = \sqrt{(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})}. \quad (\text{C.5})$$

For each distribution, we sample one color and test whether the Mahalanobis distance from that sampled point to the distribution it was sampled from is less than 1.38629. We accept the sample if this condition is met, and reject if the condition is not met. Any sample that is rejected is replaced until we obtain a sample that has a Mahalanobis distance less than this threshold. The process is repeated until the number of accepted colors from each of the four distributions is equal to the number of images we wish to generate.

C.2.4 Image Generation

We create the images in a manner similar to how the masks were created. For each image, we have a set of four colors each corresponding to the table, plate, border, and colony regions. The background of the image is generated by assigning to each pixel the color for the table. Next, a circle of radius 30 is generated at the center of the images and is filled with the color corresponding to the plate. Then, the colonies with corresponding centers and radii are plotted and independently filled with the color corresponding to the colony. Then, the annulus with inner radius 29 and outer radius 31 is generated and filled with the color representing the border of the plate. The resulting image is given Poisson noise to introduce color variation. The final image is then saved as a JPEG file. Paired with the binary image generated previously, this gives us image data for training a supervised neural network to perform a semantic binary segmentation on plated colony images as described in Section 5.2.4.

C.3 Performance Metrics of *C. albicans* Classification Pipeline on Additional Experimental Data

C.3.1 Extended CHROMagar Dataset

Single Input Toy Model

An accuracy of 95% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 95% for opaque colonies, 0% for sectoried colonies, and 86% for background respectively (Figure C.4 (top)). (Precision, Recall) scores for each class are (0.99, 0.96) for white colonies, (0.95, 0.76) for opaque colonies, (0.00, 0.00) for sectoried colonies, and (0.86, 1.00) for background respectively. F1 scores for each class are 0.98 for white colonies, 0.84 for opaque colonies, 0.00 for sectoried colonies, and 0.92 for background respectively (Table 5.3).

Precision-Recall AUC scores for each class are 0.997 for white colonies, 0.918 for opaque colonies, 0.026 for sectoried colonies, and 0.981 for background respectively. ROC-AUC scores for each class are 0.994 for white colonies, 0.991 for opaque colonies, 0.858 for sectoried colonies, and 0.990 for background respectively. (Figure C.4 (bottom))

Dual Input Toy Model

An accuracy of 91% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 100% for opaque colonies, 0% for sectoried colonies, and 70% for background respectively (Figure C.5 (top)). (Precision, Recall) scores for each class are (0.99, 0.96) for white colonies, (1.00, 0.57) for opaque colonies, (0.00, 0.00) for sectoried colonies, and (0.70, 1.00) for background respectively. F1 scores for each class are 0.97 for white colonies, 0.73 for opaque colonies, 0.00 for sectoried colonies, and 0.83 for background respectively (Table 5.3).

Precision-Recall AUC scores for each class are 0.984 for white colonies, 0.942 for opaque colonies, 0.019 for sectoried colonies, and 0.878 for background respectively. ROC-AUC scores for each class are 0.970 for white colonies, 0.990 for opaque colonies, 0.811 for sectoried colonies, and 0.943 for background respectively. (Figure C.5 (bottom))

Single Input Resnet 34

An accuracy of 97% across the entire test image set is obtained. The per-class accuracies are 98% for white colonies, 95% for opaque colonies, 50% for sectoried colonies, and 94% for background respectively (Figure C.6 (top)). (Precision, Recall) scores for each class are (0.98, 0.98) for white colonies, (0.95, 0.90) for opaque colonies, (0.50, 0.33) for sectoried colonies, and (0.94, 0.97) for background respectively. F1

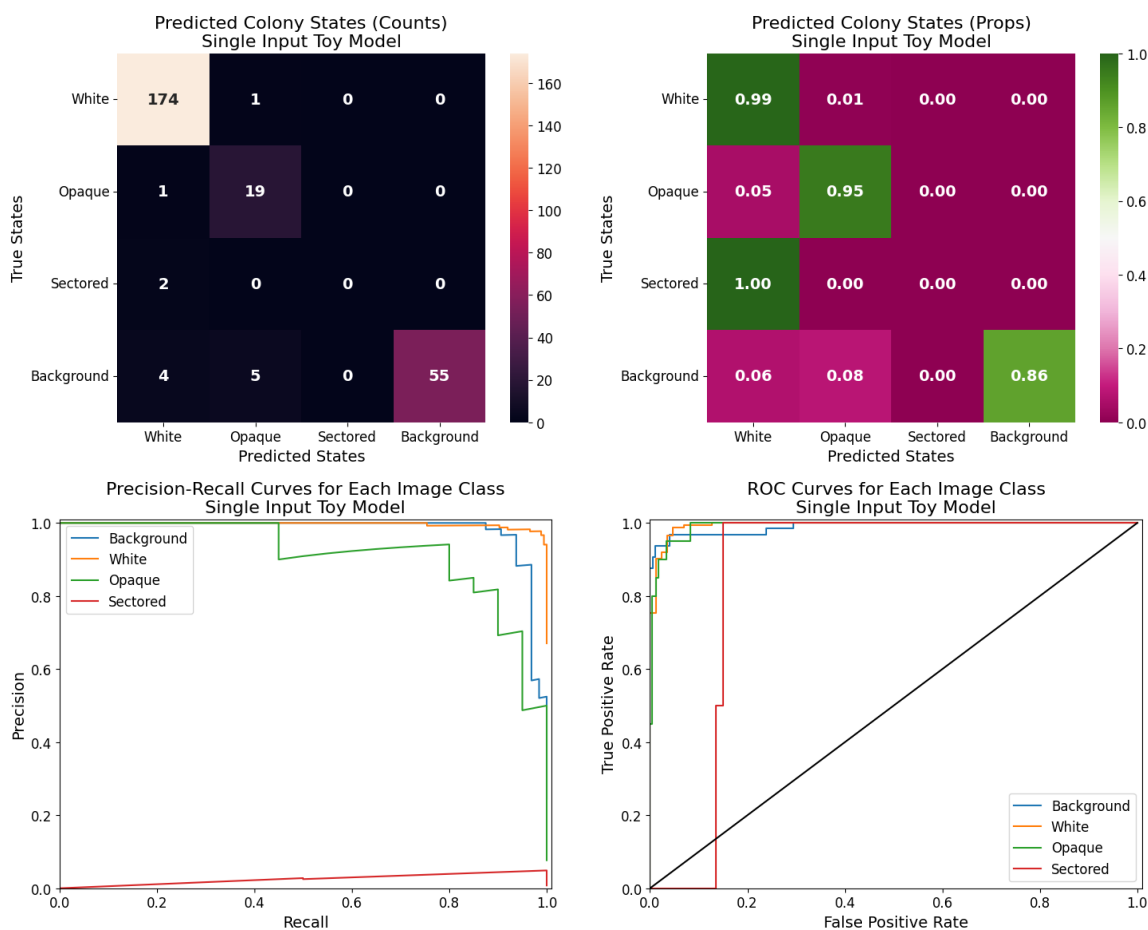


Figure C.4: **Accuracy and performance of the single input toy model on the extended CHROMagar image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input toy model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

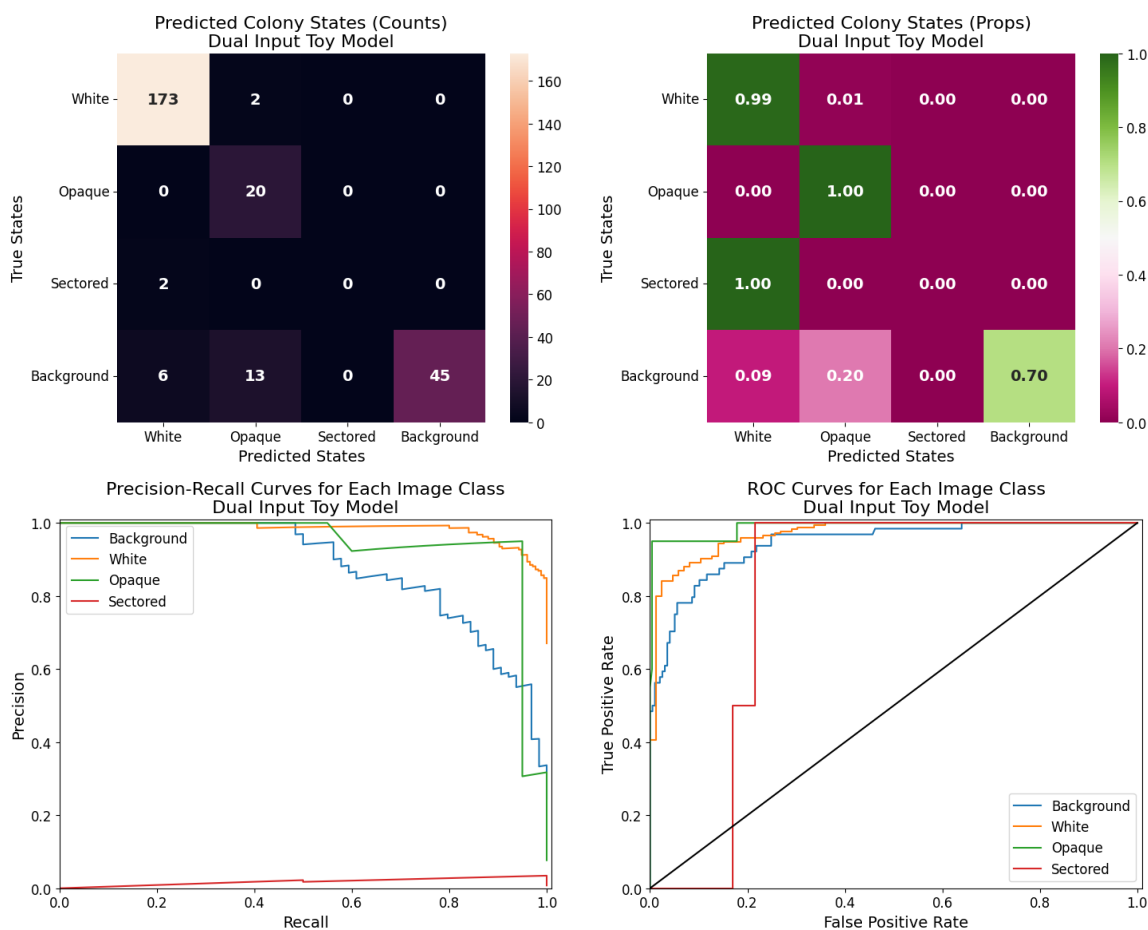


Figure C.5: **Accuracy and performance of the dual input toy model on the extended CHROMagar image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input toy model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

scores for each class are 0.98 for white colonies, 0.93 for opaque colonies, 0.40 for sectored colonies, and 0.95 for background respectively (Table 5.3).

Precision-Recall AUC scores for each class are 0.997 for white colonies, 0.927 for opaque colonies, 0.168 for sectored colonies, and 0.971 for background respectively. ROC-AUC scores for each class are 0.994 for white colonies, 0.985 for opaque colonies, 0.904 for sectored colonies, and 0.989 for background respectively. (Figure C.6 (bottom))

Dual Input Resnet 34

An accuracy of 96% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 85% for opaque colonies, 0% for sectored colonies, and 95% for background respectively (Figure C.7 (top)). (Precision, Recall) scores for each class are (0.99, 0.98) for white colonies, (0.85, 0.89) for opaque colonies, (0.00, 0.00) for sectored colonies, and (0.95, 0.94) for background respectively. F1 scores for each class are 0.99 for white colonies, 0.87 for opaque colonies, 0.00 for sectored colonies, and 0.95 for background respectively (Table 5.3).

Precision-Recall AUC scores for each class are 0.997 for white colonies, 0.907 for opaque colonies, 0.035 for sectored colonies, and 0.977 for background respectively. ROC-AUC scores for each class are 0.994 for white colonies, 0.991 for opaque colonies, 0.831 for sectored colonies, and 0.981 for background respectively. (Figure C.7 (bottom))

C.3.2 SDA Dataset

Single Input Toy Model

An accuracy of 91% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 92% for opaque colonies, 17% for sectored colonies, and 65% for background respectively (Figure C.8 (top)). (Precision, Recall) scores for each class are (0.99, 0.92) for white colonies, (0.92, 0.80) for opaque colonies, (0.17, 1.00) for sectored colonies, and (0.65, 0.96) for background respectively. F1 scores for each class are 0.95 for white colonies, 0.86 for opaque colonies, 0.29 for sectored colonies, and 0.78 for background respectively (Table 5.5).

Precision-Recall AUC scores for each class are 0.954 for white colonies, 0.928 for opaque colonies, 0.086 for sectored colonies, and 0.832 for background respectively. ROC-AUC scores for each class are 0.943 for white colonies, 0.979 for opaque colonies, 0.864 for sectored colonies, and 0.908 for background respectively. (Figure C.8 (bottom))

Dual Input Toy Model

An accuracy of 87% across the entire test image set is obtained. The per-class accuracies are 95% for white colonies, 94% for opaque colonies, 0% for sectored colonies, and 60% for background respectively (Figure C.9 (top)). (Precision, Recall) scores for

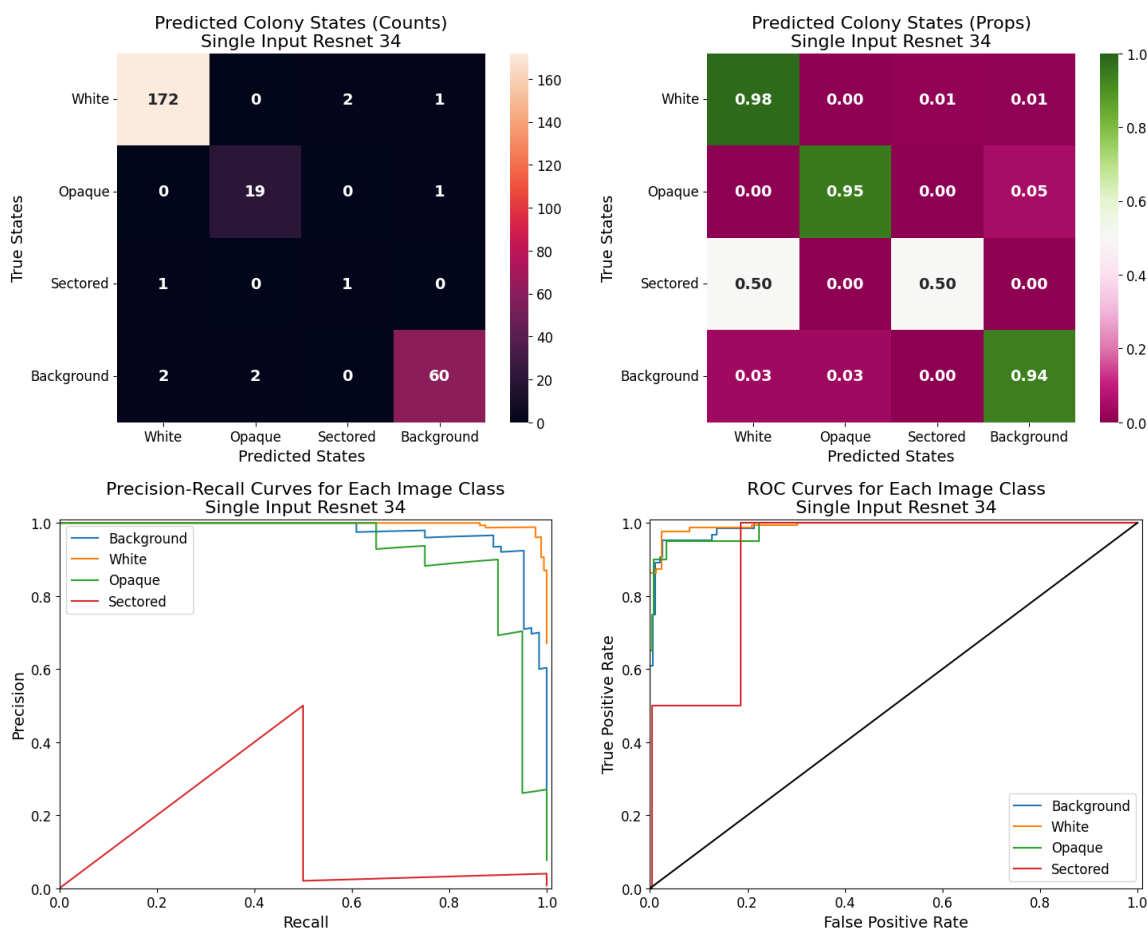


Figure C.6: **Accuracy and performance of the single input Resnet 34 on the extended CHROMagar image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

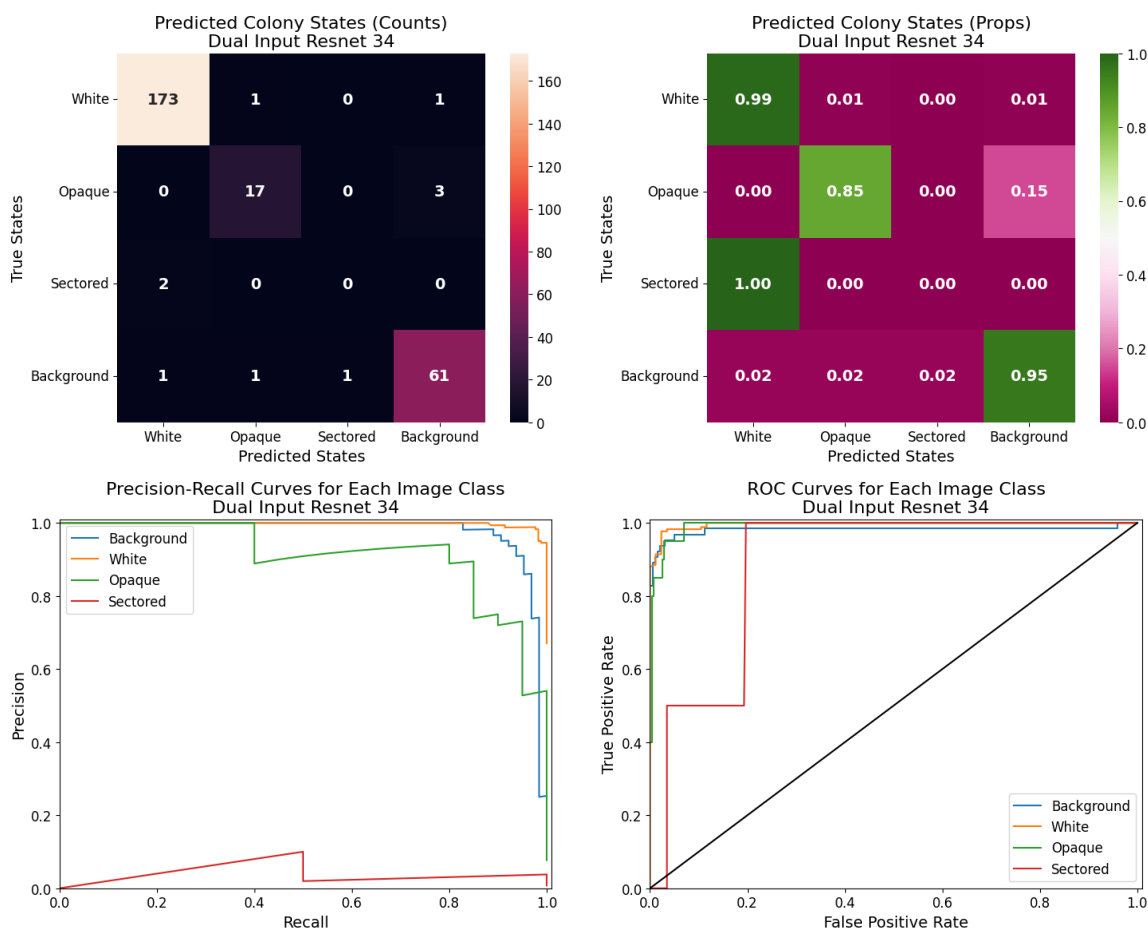


Figure C.7: Accuracy and performance of the dual input Resnet 34 on the extended CHROMagar image set. Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

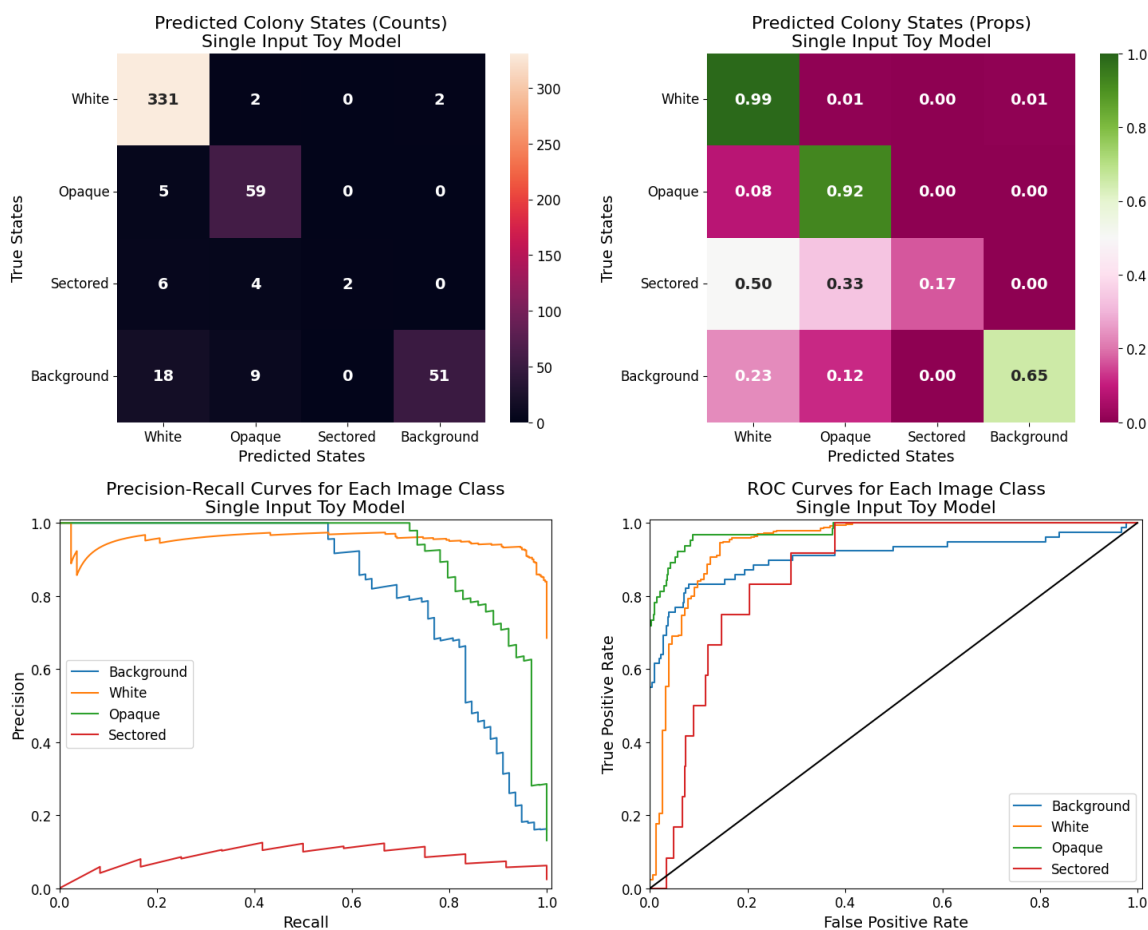


Figure C.8: **Accuracy and performance of the single input toy model on the SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input toy model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

each class are (0.95, 0.94) for white colonies, (0.94, 0.59) for opaque colonies, (0.00, 0.00) for sectored colonies, and (0.60, 1.00) for background respectively. F1 scores for each class are 0.94 for white colonies, 0.73 for opaque colonies, 0.00 for sectored colonies, and 0.75 for background respectively (Table 5.5).

Precision-Recall AUC scores for each class are 0.920 for white colonies, 0.868 for opaque colonies, 0.055 for sectored colonies, and 0.781 for background respectively. ROC-AUC scores for each class are 0.911 for white colonies, 0.956 for opaque colonies, 0.760 for sectored colonies, and 0.869 for background respectively. (Figure C.9 (bottom))

Single Input Resnet 34

An accuracy of 90% across the entire test image set is obtained. The per-class accuracies are 98% for white colonies, 92% for opaque colonies, 8% for sectored colonies, and 65% for background respectively (Figure C.10 (top)). (Precision, Recall) scores for each class are (0.98, 0.91) for white colonies, (0.92, 0.81) for opaque colonies, (0.08, 0.33) for sectored colonies, and (0.65, 0.96) for background respectively. F1 scores for each class are 0.95 for white colonies, 0.86 for opaque colonies, 0.13 for sectored colonies, and 0.78 for background respectively (Table 5.5).

Precision-Recall AUC scores for each class are 0.960 for white colonies, 0.938 for opaque colonies, 0.143 for sectored colonies, and 0.863 for background respectively. ROC-AUC scores for each class are 0.950 for white colonies, 0.975 for opaque colonies, 0.833 for sectored colonies, and 0.929 for background respectively. (Figure C.10 (bottom))

Dual Input Resnet 34

An accuracy of 91% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 94% for opaque colonies, 0% for sectored colonies, and 67% for background respectively (Figure C.11 (top)). (Precision, Recall) scores for each class are (0.99, 0.92) for white colonies, (0.94, 0.83) for opaque colonies, (0.00, 0.00) for sectored colonies, and (0.67, 0.95) for background respectively. F1 scores for each class are 0.96 for white colonies, 0.88 for opaque colonies, 0.00 for sectored colonies, and 0.78 for background respectively (Table 5.5).

Precision-Recall AUC scores for each class are 0.955 for white colonies, 0.940 for opaque colonies, 0.059 for sectored colonies, and 0.833 for background respectively. ROC-AUC scores for each class are 0.944 for white colonies, 0.979 for opaque colonies, 0.619 for sectored colonies, and 0.882 for background respectively. (Figure C.11 (bottom))

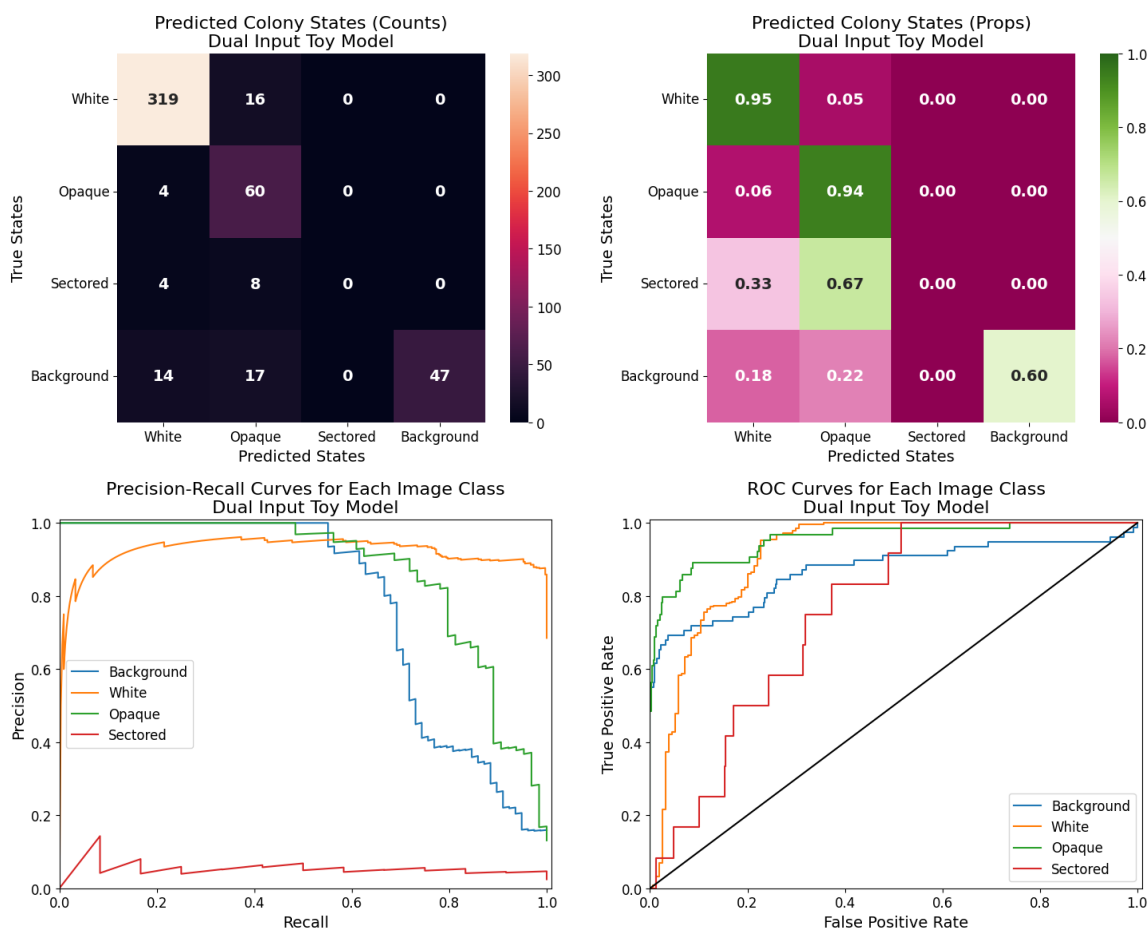


Figure C.9: **Accuracy and performance of the dual input toy model on the SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input toy model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

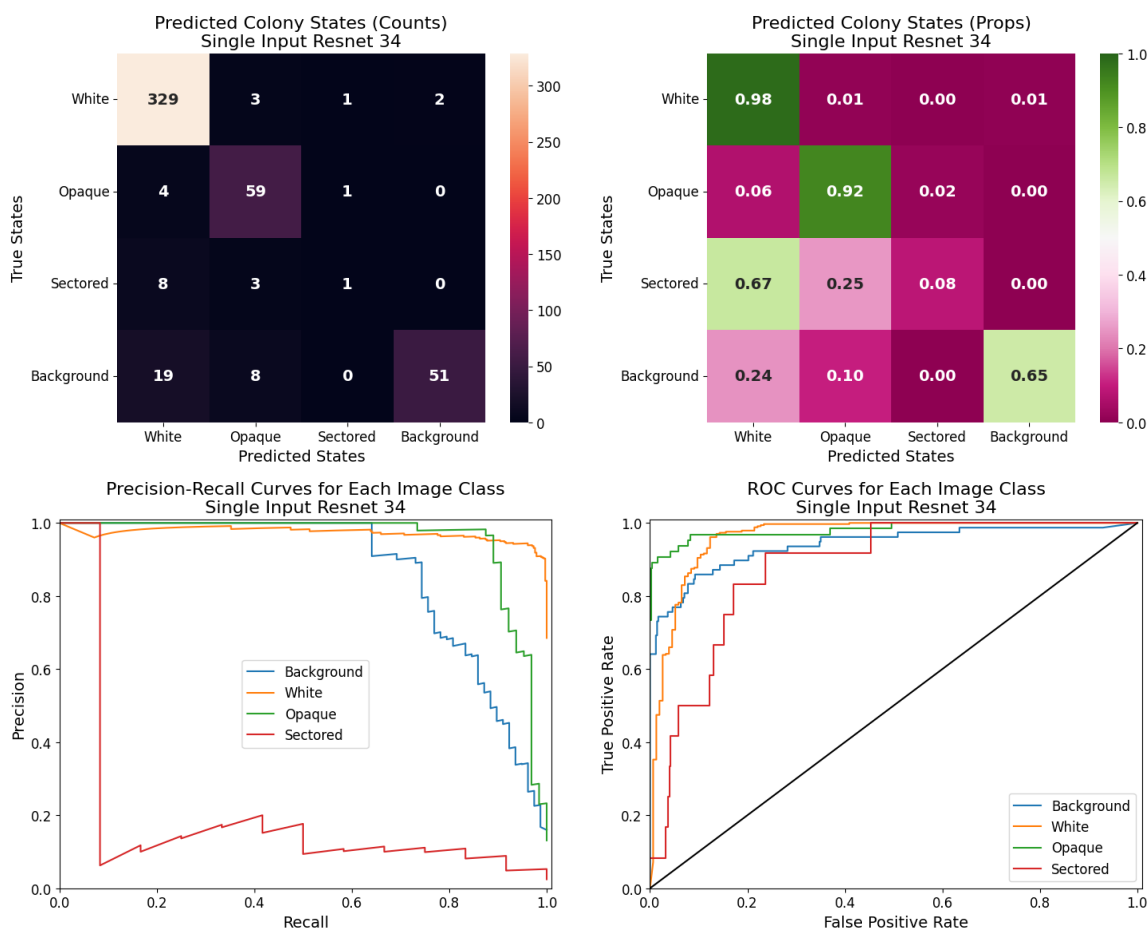


Figure C.10: **Accuracy and performance of the single input Resnet 34 on the SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

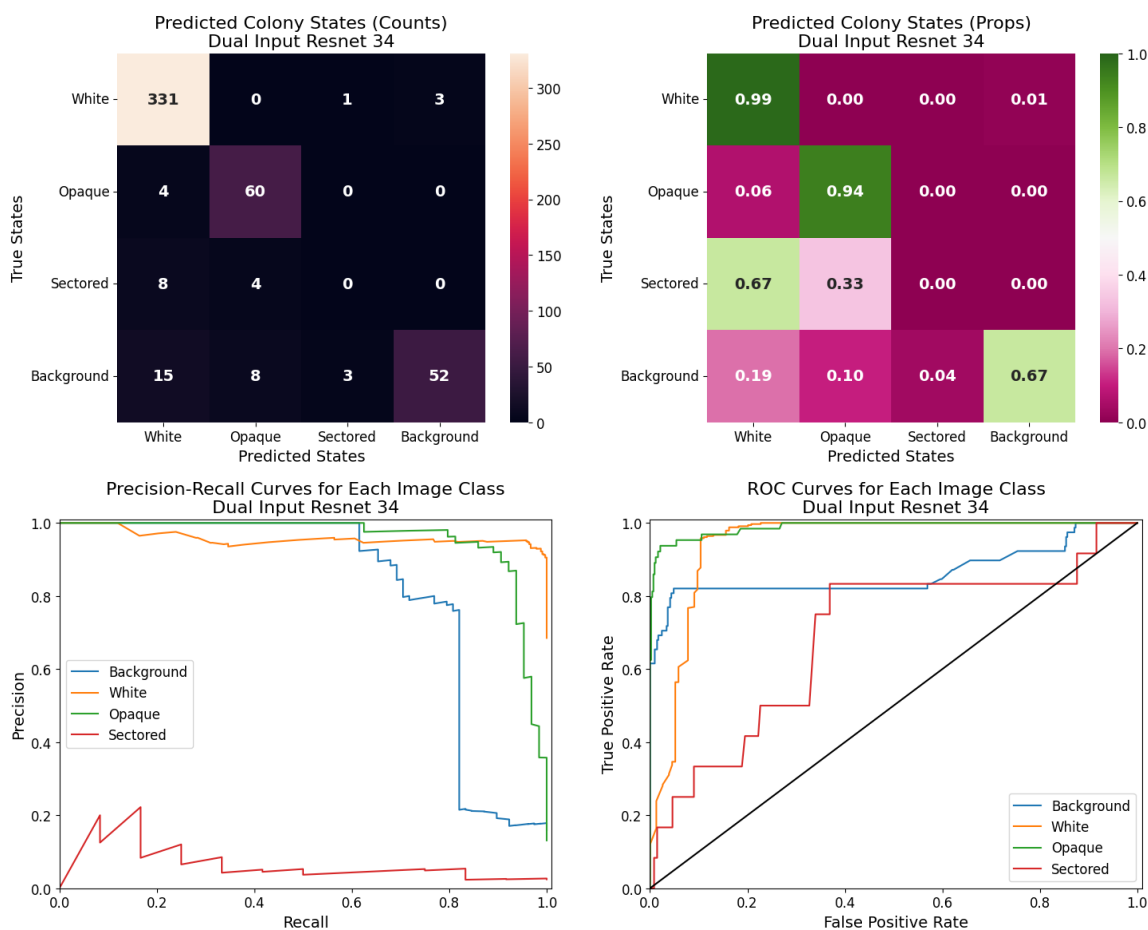


Figure C.11: **Accuracy and performance of the dual input Resnet 34 on the SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

C.3.3 Combined CHROMagar + SDA Dataset

Single Input Toy Model

An accuracy of 89% across the entire test image set is obtained. The per-class accuracies are 97% for white colonies, 92% for opaque colonies, 14% for sectoried colonies, and 67% for background respectively (Figure C.12 (top)). (Precision, Recall) scores for each class are (0.97, 0.93) for white colonies, (0.92, 0.71) for opaque colonies, (0.14, 0.25) for sectoried colonies, and (0.67, 0.99) for background respectively. F1 scores for each class are 0.95 for white colonies, 0.80 for opaque colonies, 0.18 for sectoried colonies, and 0.80 for background respectively.

Precision-Recall AUC scores for each class are 0.975 for white colonies, 0.911 for opaque colonies, 0.109 for sectoried colonies, and 0.899 for background respectively. ROC-AUC scores for each class are 0.967 for white colonies, 0.980 for opaque colonies, 0.869 for sectoried colonies, and 0.942 for background respectively. (Figure C.12 (bottom))

Dual Input Toy Model

An accuracy of 91% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 88% for opaque colonies, 0% for sectoried colonies, and 73% for background respectively (Figure C.13 (top)). (Precision, Recall) scores for each class are (0.99, 0.93) for white colonies, (0.88, 0.76) for opaque colonies, (0.00, 0.00) for sectoried colonies, and (0.73, 0.98) for background respectively. F1 scores for each class are 0.96 for white colonies, 0.82 for opaque colonies, 0.00 for sectoried colonies, and 0.83 for background respectively.

Precision-Recall AUC scores for each class are 0.984 for white colonies, 0.900 for opaque colonies, 0.094 for sectoried colonies, and 0.909 for background respectively. ROC-AUC scores for each class are 0.972 for white colonies, 0.975 for opaque colonies, 0.868 for sectoried colonies, and 0.956 for background respectively. (Figure C.13 (bottom))

Single Input Resnet 34

An accuracy of 92% across the entire test image set is obtained. The per-class accuracies are 99% for white colonies, 92% for opaque colonies, 21% for sectoried colonies, and 76% for background respectively (Figure C.14 (top)). (Precision, Recall) scores for each class are (0.99, 0.93) for white colonies, (0.92, 0.86) for opaque colonies, (0.21, 0.50) for sectoried colonies, and (0.76, 0.97) for background respectively. F1 scores for each class are 0.96 for white colonies, 0.89 for opaque colonies, 0.30 for sectoried colonies, and 0.85 for background respectively.

Precision-Recall AUC scores for each class are 0.971 for white colonies, 0.930 for opaque colonies, 0.132 for sectoried colonies, and 0.919 for background respectively. ROC-AUC scores for each class are 0.964 for white colonies, 0.977 for opaque colonies,

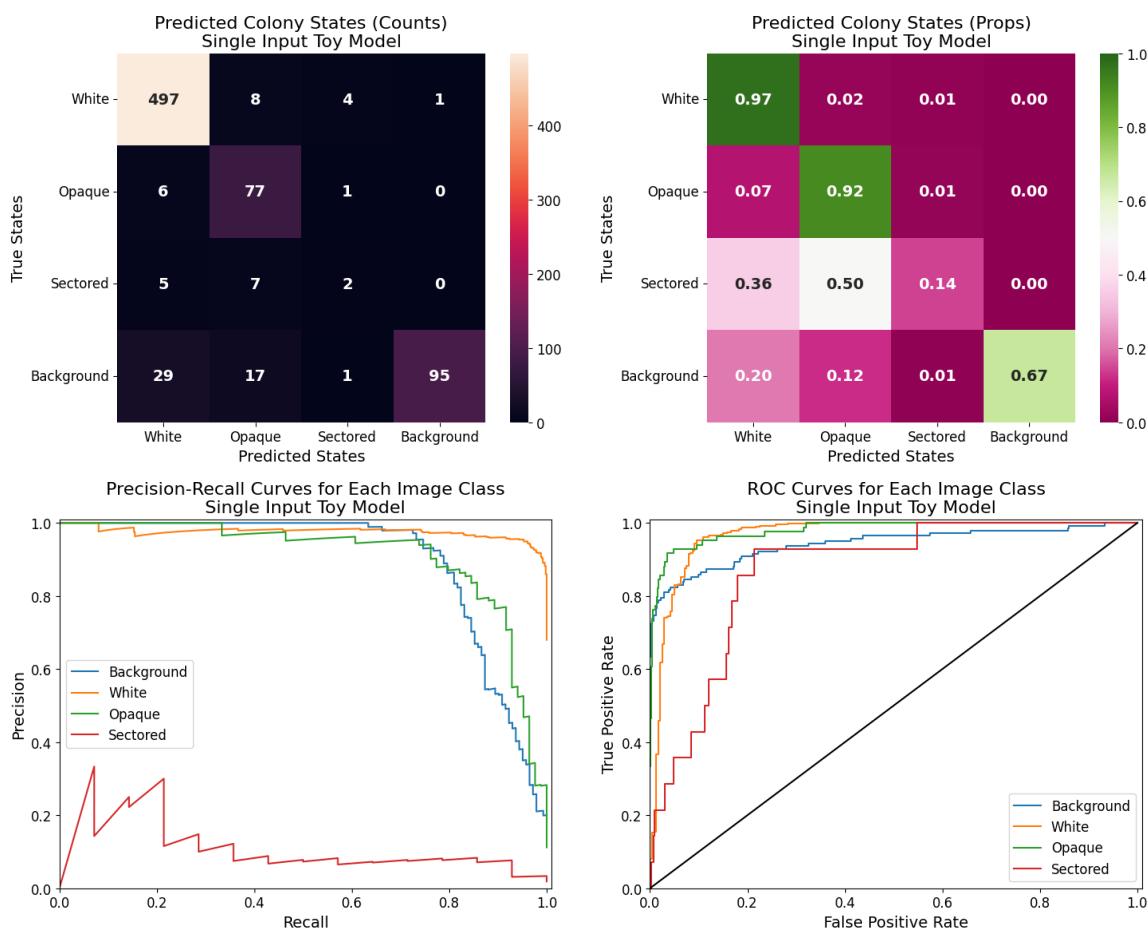


Figure C.12: **Accuracy and performance of the single input toy model on the combined CHROM and SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input toy 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

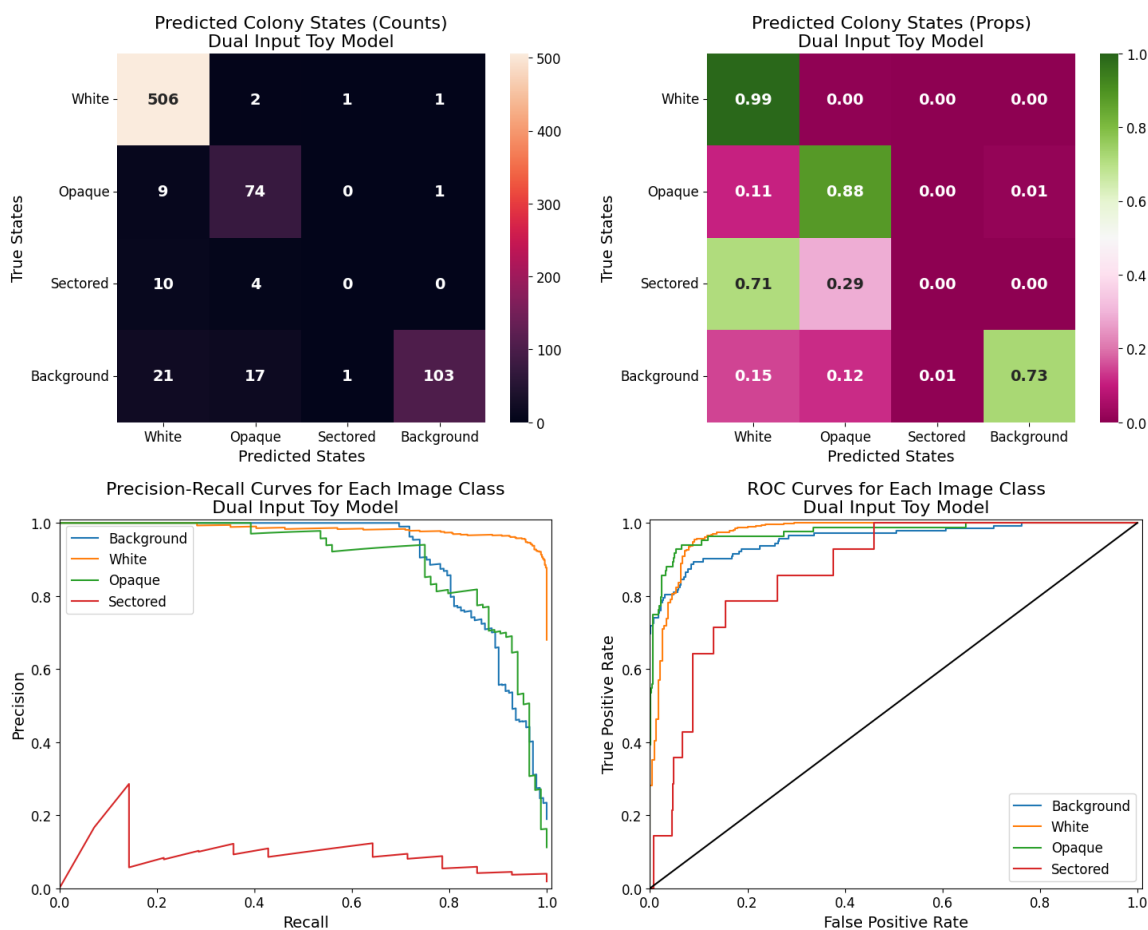


Figure C.13: **Accuracy and performance of the dual input toy model on the combined CHROM and SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input toy 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

0.822 for sectored colonies, and 0.954 for background respectively. (Figure C.14 (bottom))

Dual Input Resnet 34

An accuracy of 88% across the entire test image set is obtained. The per-class accuracies are 95% for white colonies, 73% for opaque colonies, 14% for sectored colonies, and 80% for background respectively (Figure C.15 (top)). (Precision, Recall) scores for each class are (0.95, 0.94) for white colonies, (0.73, 0.75) for opaque colonies, (0.14, 0.18) for sectored colonies, and (0.80, 0.81) for background respectively. F1 scores for each class are 0.95 for white colonies, 0.74 for opaque colonies, 0.16 for sectored colonies, and 0.81 for background respectively.

Precision-Recall AUC scores for each class are 0.978 for white colonies, 0.799 for opaque colonies, 0.171 for sectored colonies, and 0.877 for background respectively. ROC-AUC scores for each class are 0.960 for white colonies, 0.966 for opaque colonies, 0.844 for sectored colonies, and 0.933 for background respectively. (Figure C.15 (bottom))

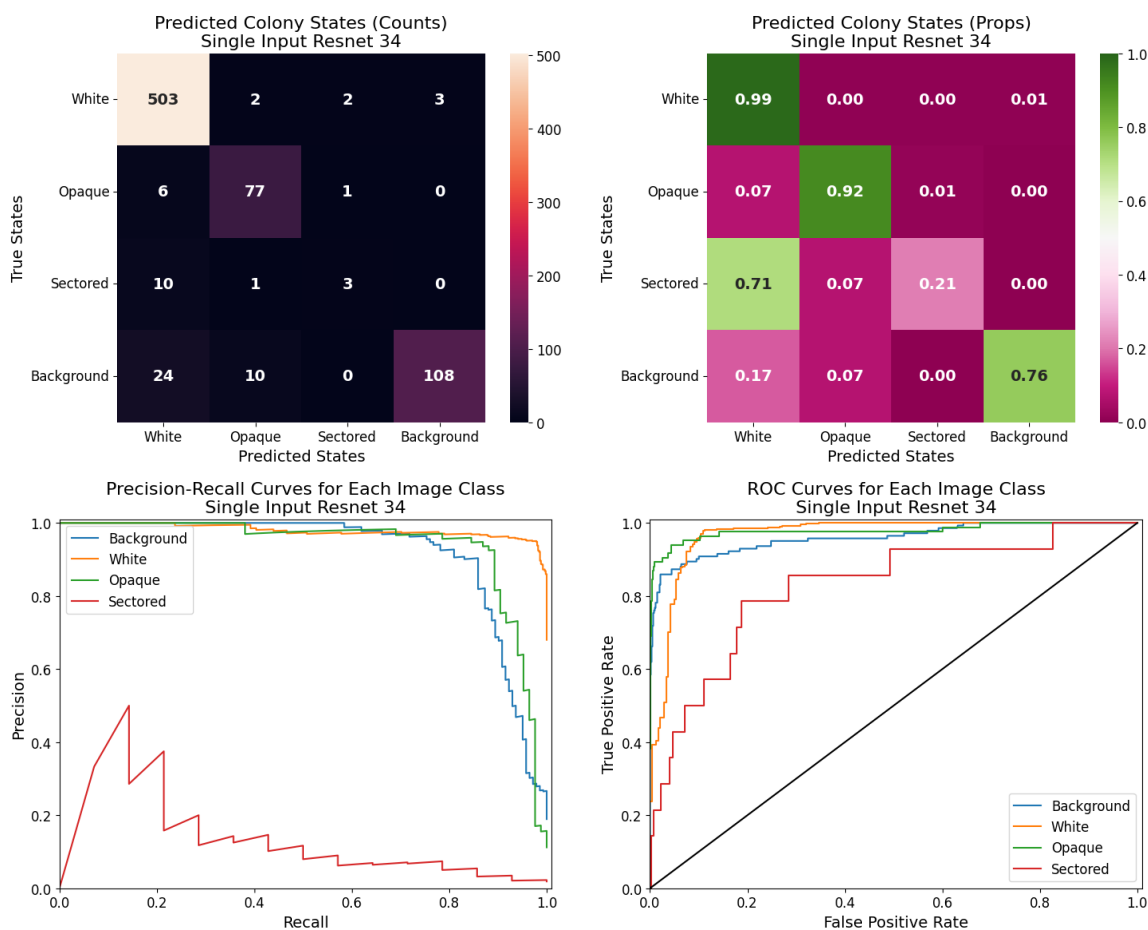


Figure C.14: **Accuracy and performance of the single input Resnet 34 on the combined CHROM and SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the single input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

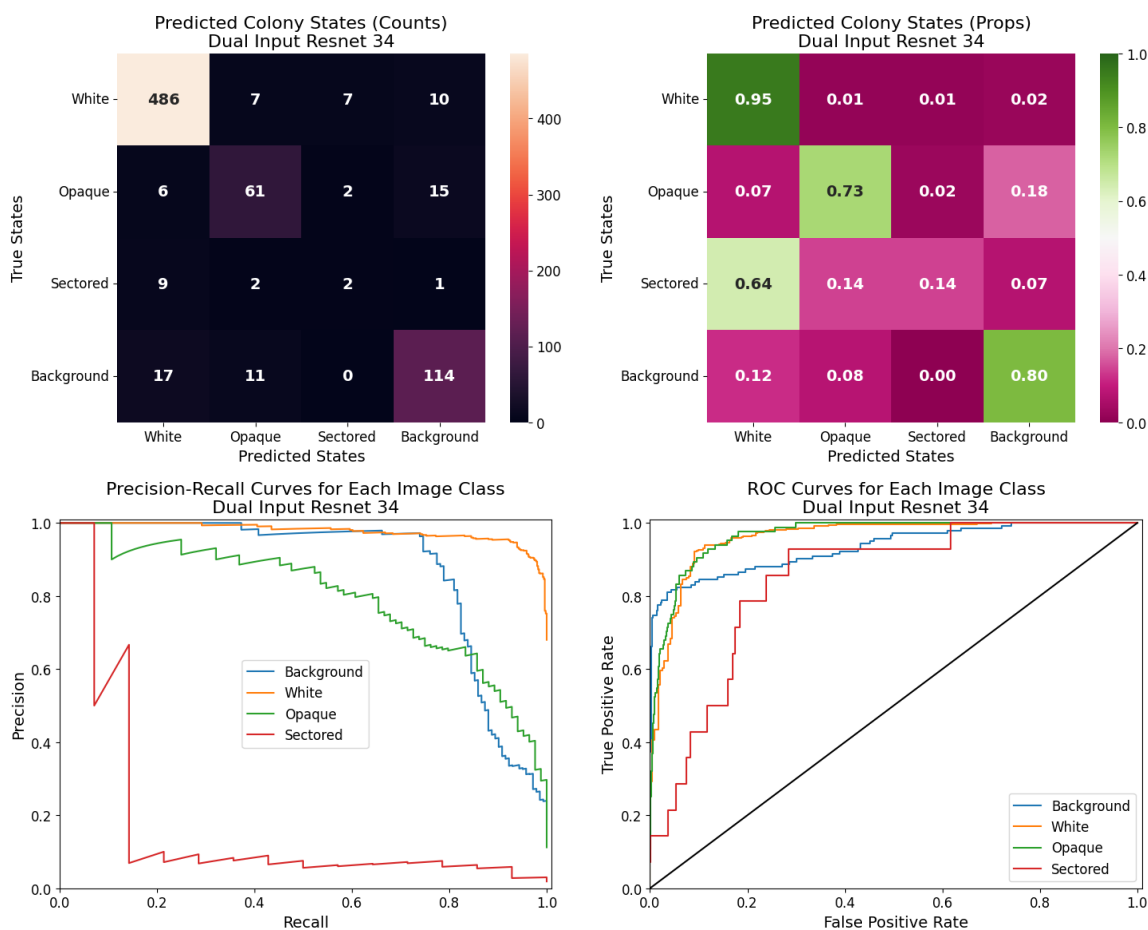


Figure C.15: **Accuracy and performance of the dual input Resnet 34 on the combined CHROM and SDA image set.** Top: Confusion matrices showing the number of correctly and incorrectly classified colonies, with proportions normalized by row. Bottom: Precision-recall and ROC curves showing qualitative performance of the dual input Resnet 34 model on classifying images for each of the four image types. The black line is a reference to the performance of a truly random classifier.

References

- [1] Dimas Praja Purwa Aji, Devi Aprianti, and Sparisoma Viridi. Stochastic simulation of yeast cells and its colony growth by using circular granular model for cases of growth and birth probabilities depends on position. In Journal of Physics: Conference Series, volume 1245, page 012010. IOP Publishing, 2019.
- [2] Mark S Alber, Yi Jiang, and Maria A Kiskowski. Lattice gas cellular automation model for rippling and aggregation in myxobacteria. Physica D: Nonlinear Phenomena, 191(3-4):343–358, 2004.
- [3] Aboutaleb Amiri, Cameron Harvey, Amy Buchmann, Scott Christley, Joshua D Shrout, Igor S Aranson, and Mark Alber. Reversals and collisions optimize protein exchange in bacterial swarms. Physical Review E, 95(3):032408, 2017.
- [4] D Aprianti, F Haryanto, A Purqon, SN Khotimah, and S Viridi. Study of budding yeast colony formation and its characterizations by using circular granular cell. In Journal of Physics: Conference Series, volume 694, page 012079, 2016.
- [5] Devi Aprianti, SN Khotimah, and S Viridi. Budding yeast colony growth study based on circular granular cell. In Journal of Physics: Conference Series, volume 739, page 012026. IOP Publishing, 2016.
- [6] Tim J Atherton and Darren J Kerbyson. Size invariant circle detection. Image and Vision computing, 17(11):795–803, 1999.
- [7] Sviatoslav Bagriantsev and Susan Liebman. Modulation of $\alpha\beta$ 42 low-n oligomerization using a novel yeast reporter system. BMC biology, 4(1):1–12, 2006.
- [8] Mikahl Banwarth-Kuhn, Jordan Collignon, and Suzanne Sindi. Quantifying the biophysical impact of budding cell division on the spatial organization of growing yeast colonies. Applied Sciences, 10(17):5780, 2020.
- [9] Eshel Ben-Jacob, Ofer Schochet, Adam Tenenbaum, Inon Cohen, Andras Czirok, and Tamas Vicsek. Generic modelling of cooperative growth patterns in bacterial colonies. Nature, 368(6466):46–49, 1994.

- [10] Richard J Bennett. Coming of age—sexual reproduction in candida species. PLoS Pathogens, 6(12):e1001155, 2010.
- [11] Van Bettauer, Anna Carolina Borges Pereira Costa, Raha Parvizi Omran, Samira Massahi, Eftyhios Kirbizakis, Shawn Simpson, Vanessa Dumeaux, Chris Law, Malcolm Whiteway, and Michael T Hallett. A deep learning approach to capture the essence of candida albicans morphologies. Microbiology Spectrum, 10(5):e01472–22, 2022.
- [12] JM Bewes, N Suchowerska, and DR McKenzie. Automated cell colony counting and analysis using the circular hough image transform algorithm (chita). Physics in Medicine & Biology, 53(21):5991, 2008.
- [13] Prashant Bharadwaj, Ralph Martins, and Ian Macreadie. Yeast as a model for studying alzheimer’s disease. FEMS yeast research, 10(8):961–969, 2010.
- [14] Benjamin J Binder and Matthew J Simpson. Cell density and cell size dynamics during in vitro tissue growth experiments: Implications for mathematical models of collective cell behaviour. Applied Mathematical Modelling, 40(4):3438–3446, 2016.
- [15] Benjamin J Binder, Joanna F Sundstrom, Jennifer M Gardner, Vladimir Jiranek, and Stephen G Oliver. Quantifying two-dimensional filamentous and invasive growth spatial patterns in yeast colonies. PLoS computational biology, 11(2), 2015.
- [16] J. E. Bresenham. Algorithm for computer control of a digital plotter. IBM Systems Journal, 4(1):25–30, 1965.
- [17] Bonita J Brewer, Ewa Chlebowicz-Sledziewska, and Walton L Fangman. Cell cycle phases in the unequal mother/daughter cell cycles of saccharomyces cerevisiae. Molecular and cellular biology, 4(11):2529–2531, 1984.
- [18] James R Broach. Nutritional control of growth and development in yeast. Genetics, 192(1):73–105, 2012.
- [19] Stefan Brückner and Hans-Ulrich Mösche. Choosing the right lifestyle: adhesion and development in saccharomyces cerevisiae. FEMS microbiology reviews, 36(1):25–58, 2012.
- [20] Silvio D Brugger, Christian Baumberger, Marcel Jost, Werner Jenni, Urs Brugger, and Kathrin Mühlemann. Automated counting of bacterial colony forming units on agar plates. PloS one, 7(3):e33695, 2012.
- [21] Breck Byers. Cytology of the yeast life cycle. The molecular biology of the yeast Saccharomyces: life cycle and inheritance., pages 59–96, 1981.

- [22] Helen Byrne and Dirk Drasdo. Individual-based and continuum models of growing cell populations: a comparison. Journal of mathematical biology, 58(4-5):657, 2009.
- [23] Vincent Calvez, Natacha Lenuzza, Marie Doumic, Jean-Philippe Deslys, Franck Mouthon, and Benoit Perthame. Prion dynamics with size dependency–strain phenomena. Journal of Biological Dynamics, 4(1):28–42, 2010.
- [24] John Canny. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, 8(6):679–698, 1986.
- [25] Sarah H Carl, Lea Duempelmann, Yukiko Shimada, and Marc Bühler. A fully automated deep learning pipeline for high-throughput colony segmentation and classification. Biology Open, 9(6), 2020.
- [26] Sean M Cascarina and Eric D Ross. Yeast prions and human prion-like proteins: sequence features and prediction methods. Cellular and Molecular Life Sciences, 71(11):2047–2063, 2014.
- [27] John Chant, Michelle Mischke, Elizabeth Mitchell, Ira Herskowitz, and John R Pringle. Role of bud3p in producing the axial budding pattern of yeast. The Journal of cell biology, 129(3):767–778, 1995.
- [28] Ching-Yu Chia. Color-transfer-between-images. <https://github.com/chia56028/Color-Transfer-between-Images>, 2019.
- [29] Priya Choudhry. High-throughput method for automated colony and cell counting by digital image analysis based on edge detection. PloS one, 11(2):e0148469, 2016.
- [30] James Coady, Andrew O’Riordan, Gerard Dooly, Thomas Newe, and Daniel Toal. An overview of popular digital image processing filtering operations. In 2019 13th International conference on sensing technology (ICST), pages 1–5. IEEE, 2019.
- [31] John Collinge. Molecular neurology of prion disease. Journal of Neurology, Neurosurgery & Psychiatry, 76(7):906–919, 2005.
- [32] BS Cox. [psi. sup.+]. a cytoplasmic suppressor of super-suppressor in yeast. Heredity, 20:505–521, 1965.
- [33] Paul J Cullen and George F Sprague. The regulation of filamentous growth in yeast. Genetics, 190(1):23–49, 2012.
- [34] Cameron Davidson-Pilon, Jonas Kalderstam, Noah Jacobson, sean reed, Ben Kuhn, Paul Zivich, Mike Williamson, AbdealiJK, Deepyaman Datta, Andrew

- Fiore-Gartland, Alex Parij, Daniel Wilson, Gabriel, Luis Moneda, Kyle Stark, Arturo Moncada-Torres, Harsh Gadgil, Jona, Karthikeyan Singaravelan, Lilian Besson, Miguel Sancho Peña, Steven Anton, Andreas Klintberg, Javad Noorbakhsh, Matthew Begun, Ravin Kumar, Sean Hussey, Dave Golland, jlim13, and Abraham Flaxman. Camdavidsonpilon/lifelines: v0.24.16, jul 2020.
- [35] Jason K Davis and Suzanne S Sindi. A mathematical model of the dynamics of prion aggregates with chaperone-mediated fragmentation. Journal of mathematical biology, 72(6):1555–1578, 2016.
- [36] Claudio De Virgilio. The essence of yeast quiescence. FEMS microbiology reviews, 36(2):306–339, 2012.
- [37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.
- [38] Aaron Derdowski, Suzanne S Sindi, Courtney L Klaips, Susanne DiSalvo, and Tricia R Serio. A size threshold limits prion transmission and establishes phenotypic diversity. Science, 330(6004):680–683, 2010.
- [39] Stefano Di Talia, Jan M Skotheim, James M Bean, Eric D Siggia, and Frederick R Cross. The effects of molecular noise and size control on variability in the budding yeast cell cycle. Nature, 448(7156):947–951, 2007.
- [40] Nicola Dietler, Matthias Minder, Vojislav Gligorovski, Augoustina Maria Economou, Denis Alain Henri Lucien Joly, Ahmad Sadeghi, Chun Hei Michael Chan, Mateusz Koziński, Martin Weigert, Anne-Florence Bitbol, et al. A convolutional neural network segments yeast microscopy images with high accuracy. Nature communications, 11(1):5723, 2020.
- [41] S Dini, BJ Binder, SC Fischer, C Mattheyer, A Schmitz, EHK Stelzer, NG Bean, and JEF Green. Identifying the necrotic zone boundary in tumour spheroids with pair-correlation functions. Journal of The Royal Society Interface, 13(123):20160649, 2016.
- [42] Susanne DiSalvo and Tricia R Serio. Insights into prion biology: integrating a protein misfolding pathway with its cellular environment. Prion, 5(2):76–83, 2011.
- [43] Anne M Dranginis, Jason M Rauceo, Juan E Coronado, and Peter N Lipke. A biochemical guide to yeast adhesins: glycoproteins for social and antisocial occasions. Microbiology and molecular biology reviews, 71(2):282–294, 2007.
- [44] Dirk Drasdo and Gabor Forgacs. Modeling the interplay of generic and genetic mechanisms in cleavage, blastulation, and gastrulation. Developmental

- dynamics: an official publication of the American Association of Anatomists, 219(2):182–191, 2000.
- [45] Dirk Drasdo, Stefan Hoehme, and Michael Block. On the role of physics in the growth and pattern formation of multi-cellular systems: What can we learn from individual-cell based models? Journal of Statistical Physics, 128(1-2):287, 2007.
- [46] Dirk Drasdo and Stefan Höhme. A single-cell-based model of tumor growth in vitro: monolayers and spheroids. Physical biology, 2(3):133, 2005.
- [47] Dirk Drasdo and Markus Loeffler. Individual-based models to growth and folding in one-layered tissues: intestinal crypts and early development. Nonlinear Analysis-Theory Methods and Applications, 47(1):245–256, 2001.
- [48] David G Drubin and W James Nelson. Origins of cell polarity. Cell, 84(3):335–344, 1996.
- [49] Bradley Efron. Logistic regression, survival analysis, and the kaplan-meier curve. Journal of the American statistical Association, 83(402):414–425, 1988.
- [50] William P Esler, Evelyn R Stimson, Joan M Jennings, Harry V Vinters, Joseph R Ghilardi, Jonathan P Lee, Patrick W Mantyh, and John E Maggio. Alzheimer’s disease amyloid propagation by a template-dependent dock-lock mechanism. Biochemistry, 39(21):6288–6295, 2000.
- [51] Reza Farhadifar, Jens-Christian Röper, Benoit Aigouy, Suzanne Eaton, and Frank Jülicher. The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing. Current Biology, 17(24):2095–2104, 2007.
- [52] Alessandro Ferrari, Stefano Lombardi, and Alberto Signoroni. Bacterial colony counting with convolutional neural networks in digital microbiology imaging. Pattern Recognition, 61:629–640, 2017.
- [53] Andrew M Finch, Richard C Wilson, and Edwin R Hancock. Matching delaunay graphs. Pattern Recognition, 30(1):123–140, 1997.
- [54] Sveva Fornari, Amelie Schäfer, Mathias Jucker, Alain Goriely, and Ellen Kuhl. Prion-like spreading of alzheimer’s disease within the brain’s connectome. Journal of the Royal Society Interface, 16(159):20190356, 2019.
- [55] Corey Frazer, Aaron D Hernday, and Richard J Bennett. Monitoring phenotypic switching in candida albicans and the use of next-gen fluorescence reporters. Current protocols in microbiology, 53(1):e76, 2019.
- [56] Neele J Froböse, Franziska Schuler, Alexander Mellmann, Marc T Hennies, Evgeny A Idelevich, and Frieder Schaumburg. Phenotypic variants of bacterial

colonies in microbiological diagnostics: how often are they indicative of differing antimicrobial susceptibility patterns? Microbiology Spectrum, 9(2):e00555–21, 2021.

- [57] Andrea Giometto, David R Nelson, and Andrew W Murray. Physical interactions reduce the power of natural selection in growing yeast colonies. Proceedings of the National Academy of Sciences, 115(45):11448–11453, 2018.
- [58] Chad M Glen, Melissa L Kemp, and Eberhard O Voit. Agent-based modeling of morphogenetic systems: Advantages and challenges. PLOS Computational Biology, 15(3):e1006577, 2019.
- [59] Amelia Gontar, Murk J Bottema, Benjamin J Binder, and Hayden Tronnolone. Characterizing the shape patterns of dimorphic yeast pseudohyphae. Royal Society open science, 5(10):180820, 2018.
- [60] Thomas E Gorochowski. Agent-based modelling in synthetic biology. Essays in biochemistry, 60(4):325–336, 2016.
- [61] Megha Gulati and Clarissa J Nobile. *Candida albicans* biofilms: development, regulation, and molecular mechanisms. Microbes and infection, 18(5):310–321, 2016.
- [62] Randal Halfmann, Daniel F Jarosz, Sandra K Jones, Amelia Chang, Alex K Lancaster, and Susan Lindquist. Prions are a common mechanism for phenotypic inheritance in wild yeasts. Nature, 482(7385):363–368, 2012.
- [63] Oskar Hallatschek, Pascal Hersen, Sharad Ramanathan, and David R Nelson. Genetic drift at expanding frontiers promotes gene segregation. Proceedings of the National Academy of Sciences, 104(50):19926–19930, 2007.
- [64] Oskar Hallatschek and David R Nelson. Life at the front of an expanding population. Evolution: International Journal of Organic Evolution, 64(1):193–206, 2010.
- [65] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.
- [66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [67] Aaron D Hernday, Matthew B Lohse, Polly M Fordyce, Clarissa J Nobile, Joseph L DeRisi, and Alexander D Johnson. Structure of the transcriptional network controlling white-opaque switching in *Candida albicans*. Molecular microbiology, 90(1):22–35, 2013.

- [68] Ira Herskowitz. Life cycle of the budding yeast *saccharomyces cerevisiae*. Microbiological reviews, 52(4):536, 1988.
- [69] SE Herwald and CA Kumamoto. *Candida albicans* niche specialization: features that distinguish biofilm cells from commensal cells. curr fungal infect rep 8: 179–184, 2014.
- [70] A Ali Heydari, Suzanne S Sindi, and Maxime Theillard. Conservative finite volume method on deforming geometries: the case of protein aggregation in dividing yeast cells. Journal of Computational Physics, page 110755, 2021.
- [71] Stefan Hoehme and Dirk Drasdo. A cell-based simulation software for multi-cellular systems. Bioinformatics, 26(20):2641–2642, 2010.
- [72] Raphael Hornung, Alexander Grünberger, Christoph Westerwalbesloh, Dietrich Kohlheyer, Gerhard Gompper, and Jens Elgeti. Quantitative modelling of nutrient-limited growth of bacterial colonies in microfluidic cultivation. Journal of the Royal Society Interface, 15(139):20170713, 2018.
- [73] Paul Hough. Method and means for recognizing complex patterns, December 1962. US Patent 3,069,654.
- [74] Charles R Hutti, Kevin A Welle, Jennifer R Hryhorenko, and Sina Ghaemmaghami. Global analysis of protein degradation in prion infected cells. Scientific reports, 10(1):1–13, 2020.
- [75] Daehee Hwang, Inyoul Y Lee, Hyuntae Yoo, Nils Gehlenborg, Ji-Hoon Cho, Brianne Petritis, David Baxter, Rose Pitstick, Rebecca Young, Doug Spicer, et al. A systems approach to prion disease. Molecular systems biology, 5(1):252, 2009.
- [76] Takao Ishikawa et al. *Saccharomyces cerevisiae* in neuroscience: how unicellular organism helps to better understand prion protein? Neural Regeneration Research, 16(3):489, 2021.
- [77] Lars M Ittner and Jürgen Götz. Amyloid- β and tau—a toxic pas de deux in alzheimer’s disease. Nature Reviews Neuroscience, 12(2):67–72, 2011.
- [78] Craig R Johnson and Maarten C Boerlijst. Selection at the level of the community: the importance of spatial structure. Trends in Ecology & Evolution, 17(2):83–90, 2002.
- [79] Henrik Jönsson and Andre Levchenko. An explicit spatial model of yeast microcolony growth. Multiscale Modeling & Simulation, 3(2):346–361, 2005.
- [80] Sarah B Joseph and David W Hall. Spontaneous mutations in diploid *saccharomyces cerevisiae*: more beneficial than expected. Genetics, 168(4):1817–1825, 2004.

- [81] Mehdi Kabani and Ronald Melki. Yeast prions assembly and propagation: contributions of the prion and non-prion moieties and the nature of assemblies. Prion, 5(4):277–284, 2011.
- [82] Jona Kayser, Carl F Schreck, QinQin Yu, Matti Gralka, and Oskar Hallatschek. Emergence of evolutionary driving forces in pattern-forming microbial populations. Philosophical Transactions of the Royal Society B: Biological Sciences, 373(1747):20170106, 2018.
- [83] Courtney L Klaips, Megan L Hochstrasser, Christine R Langlois, and Tricia R Serio. Spatial quality control bypasses cell-based limitations on proteostasis to promote prion curing. eLife, 3:e04288, 2014.
- [84] Dirk Krafzig, Frank Klawonn, and Herbert Gutz. Theoretical analysis of the effects of mitotic crossover in large yeast populations. Yeast, 9(10):1093–1098, 1993.
- [85] Stephen J Kron, Cora A Styles, and Gerald R Fink. Symmetric cell division in pseudohyphae of the yeast *saccharomyces cerevisiae*. Molecular biology of the cell, 5(9):1003–1022, 1994.
- [86] Jochen Kursawe, Pavel A Brodskiy, Jeremiah J Zartman, Ruth E Baker, and Alexander G Fletcher. Capabilities and limitations of tissue size control through passive mechanical forces. PLoS Comput. Biol., 11(12):e1004679, dec 2015.
- [87] Michael R Lamprecht, David M Sabatini, and Anne E Carpenter. Cellprofiler™: free, versatile software for automated biological image analysis. Biotechniques, 42(1):71–75, 2007.
- [88] Alex K Lancaster, J Patrick Bardill, Heather L True, and Joanna Masel. The spontaneous appearance rate of the yeast prion [psi+] and its implications for the evolution of the evolvability properties of the [psi+] system. Genetics, 184(2):393–400, 2010.
- [89] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. International Journal of Computer & Information Sciences, 9(3):219–242, 1980.
- [90] Phoebe S Lee, Patricia W Greenwell, Margaret Dominska, Malgorzata Gawel, Monica Hamilton, and Thomas D Petes. A fine-structure map of spontaneous mitotic crossovers in the yeast *saccharomyces cerevisiae*. PLoS Genet, 5(3):e1000410, 2009.
- [91] Paul Lemarre, Laurent Pujo-Menjouet, and Suzanne S Sindi. Generalizing a mathematical model of prion aggregation allows strain coexistence and co-stability by including a novel misfolded species. Journal of mathematical biology, 78(1):465–495, 2019.

- [92] Paul Lemarre, Laurent Pujo-Menjouet, and Suzanne S Sindi. A unifying model for the propagation of prion proteins in yeast brings insight into the [psi+] prion. PLoS computational biology, 16(5):e1007647, 2020.
- [93] Liming Li and Anthony S Kowal. Environmental regulation of prions in yeast. PLoS pathogens, 8(11):e1002973, 2012.
- [94] Susan W Liebman and Yury O Chernoff. Prions in yeast. Genetics, 191(4):1041–1072, 2012.
- [95] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.
- [96] Tzu-Yu Liu, Anne E Dodson, Jonathan Terhorst, Yun S Song, and Jasper Rine. Riches of phenotype computationally extracted from microbial colonies. Proceedings of the National Academy of Sciences, 113(20):E2822–E2831, 2016.
- [97] Matthew B Lohse and Alexander D Johnson. White–opaque switching in candida albicans. Current opinion in microbiology, 12(6):650–654, 2009.
- [98] Alex X Lu, Taraneh Zarin, Ian S Hsu, and Alan M Moses. Yeastspotter: accurate and parameter-free web segmentation for microscopy images of yeast cells. Bioinformatics, 35(21):4525–4527, 2019.
- [99] Douglas R Lyke, Jane E Dorweiler, and Anita L Manogaran. The three faces of sup35. Yeast, 36(8):465–472, 2019.
- [100] BK Mable. Ploidy evolution in the yeast *saccharomyces cerevisiae*: a test of the nutrient limitation hypothesis. Journal of Evolutionary Biology, 14(1):157–170, 2001.
- [101] Ramiro Magno, Verônica A Grieneisen, and Athanasius FM Marée. The biophysical nature of cells: potential cell behaviours revealed by analytical and computational studies of cell surface mechanics. BMC biophysics, 8(1):8, 2015.
- [102] Joanna Masel, Vincent AA Jansen, and Martin A Nowak. Quantifying the kinetic parameters of prion replication. Biophysical chemistry, 77(2):139–152, 1999.
- [103] Widya Meiriska, FA Purnama, DPP Aji, D Aprianti, and S Viridi. Network analysis of *saccharomyces cerevisiae* colony: Relation between spatial position and generation. In Journal of Physics: Conference Series, volume 1245, page 012006. IOP Publishing, 2019.
- [104] Sabeeha S Merchant and John D Helmann. Elemental economy: microbial strategies for optimizing growth in the face of nutrient limitation. In Advances in microbial physiology, volume 60, pages 91–210. Elsevier, 2012.

- [105] M Milani, D Batani, F Bortolotto, C Botto, G Baroni, S Cozzi, A Masini, L Ferraro, F Previdi, M Ballerini, et al. Differential two colour x-ray radiobiology of membrane/cytoplasm yeast cells: Tmr large-scale facilities access programme. NASA, 1998.
- [106] Mathew G Miller and Alexander D Johnson. White-opaque switching in candida albicans is controlled by mating-type locus homeodomain proteins and allows efficient mating. Cell, 110(3):293–302, 2002.
- [107] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 2021.
- [108] Nadege Minois, Magdalena Frajnt, Chris Wilson, and James W Vaupel. Advances in measuring lifespan in the yeast saccharomyces cerevisiae. Proceedings of the National Academy of Sciences, 102(2):402–406, 2005.
- [109] Sara Mitri, Ellen Clarke, and Kevin R Foster. Resource limitation drives spatial organization in microbial groups. The ISME journal, 10(6):1471–1482, 2016.
- [110] Elham Jasim Mohammad, Rawaa Yaseen Taha, and Howraa Ateea Mazher. Design and fundamentals of sobel edge detection of an image. J Multidiscip Eng Sci Technol (JMEST) ISSN, 9(3):2458–9403, 2022.
- [111] Joachim Morschhäuser. Regulation of white-opaque switching in candida albicans. Medical microbiology and immunology, 199:165–172, 2010.
- [112] Carey D Nadell, Kevin R Foster, and Joao B Xavier. Emergence of spatial structure in cell groups and the evolution of cooperation. PLoS Comput Biol, 6(3):e1000716, 2010.
- [113] B Anne Neville, Christophe d’Enfert, and Marie-Elisabeth Bougnoux. Candida albicans commensalism in the gastrointestinal tract. FEMS yeast research, 15(7):fov081, 2015.
- [114] T J Newman. Modeling multicellular systems using subcellular elements. Math. Biosci. Eng., 2(3):613–624, jul 2005.
- [115] Gary P Newnam, Jennifer L Birchmore, and Yury O Chernoff. Destabilization and recovery of a yeast prion after mild heat shock. Journal of molecular biology, 408(3):432–448, 2011.
- [116] Baochi Nguyen, Arpita Upadhyaya, Alexander van Oudenaarden, and Michael P Brenner. Elastic instability in growing yeast colonies. Biophysical journal, 86(5):2740–2747, 2004.

- [117] Li Ni and Michael Snyder. A genomic study of the bipolar bud site selection pattern in *saccharomyces cerevisiae*. Molecular biology of the cell, 12(7):2147–2170, 2001.
- [118] Clarissa J Nobile and Alexander D Johnson. *Candida albicans* biofilms and human disease. Annual review of microbiology, 69:71–92, 2015.
- [119] Suzanne M Noble, Brittany A Gianetti, and Jessica N Witchley. *Candida albicans* cell-type switching and functional plasticity in the mammalian host. Nature Reviews Microbiology, 15(2):96, 2017.
- [120] Martin A Nowak, David C Krakauer, Aron Klug, and Robert M May. Prion infection dynamics. Integrative Biology: Issues, News, and Reviews: Published in Association with The Society for Integrative and Comparative Biology, 1(1):3–15, 1998.
- [121] Frank C Odds and RIA Bernaerts. Chromagar candida, a new differential isolation medium for presumptive identification of clinically important candida species. Journal of clinical microbiology, 32(8):1923–1929, 1994.
- [122] Toyah Overton and Allan Tucker. Do-u-net for segmentation and counting. In International Symposium on Intelligent Data Analysis, pages 391–403. Springer, 2020.
- [123] Jarosław Pawłowski, Sylwia Majchrowska, and Tomasz Golan. Generation of microbial colonies dataset with deep learning style transfer. Scientific Reports, 12(1):5212, 2022.
- [124] John D Perry. A decade of development of chromogenic culture media for clinical microbiology in an era of molecular diagnostics. Clinical microbiology reviews, 30(2):449–479, 2017.
- [125] Vítězslav Plocek, Libuše Váchová, Vratislav Št’ovíček, and Zdena Palková. Cell distribution within yeast colonies and colony biofilms: How structure develops. International Journal of Molecular Sciences, 21(11):3873, 2020.
- [126] Stanley B Prusiner. Novel proteinaceous infectious particles cause scrapie. Science, 216(4542):136–144, 1982.
- [127] Stanley B Prusiner. Molecular biology and pathogenesis of prion diseases. Trends in biochemical sciences, 21(12):482–487, 1996.
- [128] Stanley B Prusiner. Prions. Proceedings of the National Academy of Sciences, 95(23):13363–13383, 1998.
- [129] Jan Prüss, Laurent Pujon-Menjouet, Glenn F Webb, and Rico Zacher. Analysis of a model for the dynamics of prions. Discrete & Continuous Dynamical Systems-B, 6(1):225, 2006.

- [130] Florentin Anggraini Purnama, Widya Meiriska, Dimas Praja Purwa Aji, Devi Aprianti, and Sparisoma Viridi. Network analysis of *saccharomyces cerevisiae*. In Journal of Physics: Conference Series, volume 1245, page 012081. IOP Publishing, 2019.
- [131] Boyang Qin, Chenyi Fei, Andrew A Bridges, Ameya A Mashruwala, Howard A Stone, Ned S Wingreen, and Bonnie L Bassler. Cell position fates and collective fountain flow in bacterial biofilms revealed by light-sheet microscopy. Science, 2020.
- [132] Bernardo Ramírez-Zavala, Oliver Reuß, Yang-Nim Park, Knut Ohlsen, and Joachim Morschhäuser. Environmental induction of white–opaque switching in *candida albicans*. PLoS Pathog, 4(6):e1000089, 2008.
- [133] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. IEEE Computer graphics and applications, 21(5):34–41, 2001.
- [134] Todd B Reynolds and Gerald R Fink. Bakers’ yeast, a model for fungal biofilm formation. Science, 291(5505):878–881, 2001.
- [135] Jesus A Romo and Carol A Kumamoto. On commensalism of *candida*. Journal of Fungi, 6(1):16, 2020.
- [136] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.
- [137] Alejandro Ruiz-Riquelme, Heather HC Lau, Erica Stuart, Adrienn N Goczi, Zhilan Wang, Gerold Schmitt-Ulms, and Joel C Watts. Prion-like propagation of β -amyloid aggregates in the absence of app overexpression. Acta neuropathologica communications, 6:26, 2018.
- [138] Pekka Ruusuvuori, Jake Lin, Adrian C Scott, Zhihao Tan, Saija Sorsa, Aleks Kallio, Matti Nykter, Olli Yli-Harja, Ilya Shmulevich, and Aimée M Dudley. Quantitative analysis of colony morphology in yeast. BioTechniques, 56(1):18–27, 2014.
- [139] Danny Salem, Yifeng Li, Pengcheng Xi, Hilary Phenix, Miroslava Cuperlovic-Culf, and Mads Kaern. Yeastnet: Deep-learning-enabled accurate segmentation of budding yeast cells in bright-field microscopy. Applied Sciences, 11(6):2692, 2021.
- [140] Christoph Sasse, Mike Hasenberg, Michael Weyler, Matthias Gunzer, and Joachim Morschhäuser. White-opaque switching of *candida albicans* allows immune evasion in an environment-dependent fashion. Eukaryotic cell, 12(1):50–58, 2013.

- [141] Prasanna Satpute-Krishnan and Tricia R Serio. Prion protein remodelling confers an immediate phenotypic switch. Nature, 437(7056):262–265, 2005.
- [142] Ruth Scherz, Vera Shinder, and David Engelberg. Anatomical analysis of *saccharomyces cerevisiae* stalk-like structures reveals spatial organization and cell specialization. Journal of bacteriology, 183(18):5402–5413, 2001.
- [143] Tricia R. Serio. Personal Communication, 2020.
- [144] James A Shapiro. The significances of bacterial colony patterns. Bioessays, 17(7):597–607, 1995.
- [145] Yi-Jun Sheu, Yves Barral, and Michael Snyder. Polarized growth controls cell shape and bipolar bud site selection in *saccharomyces cerevisiae*. Molecular and cellular biology, 20(14):5235–5247, 2000.
- [146] S Silvester et al. Oct2py. <https://github.com/blink1073/oct2py>, 2011.
- [147] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [148] Suzanne S Sindi. Mathematical modeling of prion disease. Prion-an overview, InTech, pages 207–227, 2017.
- [149] Alexander E Smith, Zhibing Zhang, Colin R Thomas, Kenneth E Moxham, and Anton PJ Middelberg. The mechanical properties of *saccharomyces cerevisiae*. Proceedings of the National Academy of Sciences, 97(18):9871–9874, 2000.
- [150] Michael G Smith and Michael Snyder. Yeast as a model for human disease. Current protocols in human genetics, 48(1), 2006.
- [151] William PJ Smith, Yohan Davit, James M Osborne, Wook Kim, Kevin R Foster, and Joe M Pitt-Francis. Cell morphology drives spatial patterning in microbial communities. Proceedings of the National Academy of Sciences, 114(3):E280–E286, 2017.
- [152] Irwin Sobel, R Duda, P Hart, and John Wiley. Sobel-feldman operator, 2022.
- [153] David R Soll. High-frequency switching in *candida albicans*. Clinical Microbiology Reviews, 5(2):183–203, 1992.
- [154] Kinshuk Raj Srivastava and Lisa J Lapidus. Prion protein dynamics before aggregation. Proceedings of the National Academy of Sciences, 114(14):3572–3577, 2017.
- [155] John D Stenson, Peter Hartley, Changxiang Wang, and Colin R Thomas. Determining the mechanical properties of yeast cell walls. Biotechnology progress, 27(2):505–512, 2011.

- [156] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3147–3155, 2017.
- [157] Jasminka Talapko, Martina Juzbašić, Tatjana Matijević, Emina Pustijanac, Sanja Bekić, Ivan Kotris, and Ivana Škrlec. Candida albicans—the virulence factors and clinical manifestations of infection. Journal of Fungi, 7(2):79, 2021.
- [158] Alexander Tam, J Edward F Green, Sanjeeva Balasuriya, Ee Lin Tek, Jennifer M Gardner, Joanna F Sundstrom, Vladimir Jiranek, and Benjamin J Binder. Nutrient-limited growth with non-linear cell diffusion as a mechanism for floral pattern formation in yeast biofilms. Journal of Theoretical Biology, 448:122–141, 2018.
- [159] Motomasa Tanaka, Sean R Collins, Brandon H Toyama, and Jonathan S Weissman. The physical basis of how prion conformations determine strain phenotypes. Nature, 442(7102):585–589, 2006.
- [160] Li Tao, Han Du, Guobo Guan, Yu Dai, Clarissa J Nobile, Weihong Liang, Chengjun Cao, Qiuyu Zhang, Jin Zhong, and Guanghua Huang. Discovery of a “white-gray-opaque” tristable phenotypic switching system in candida albicans: roles of non-genetic diversity in host adaptation. PLoS biology, 12(4):e1001830, 2014.
- [161] Hayden Tronolone, Jennifer M Gardner, Joanna F Sundstrom, Vladimir Jiranek, Stephen G Oliver, and Benjamin J Binder. Quantifying the dominant growth mechanisms of dimorphic yeast using a lattice-based model. Journal of The Royal Society Interface, 14(134):20170314, 2017.
- [162] Hayden Tronolone, Jennifer M Gardner, Joanna F Sundstrom, Vladimir Jiranek, Stephen G Oliver, and Benjamin J Binder. Tammicol: Tool for analysis of the morphology of microbial colonies. PLoS computational biology, 14(12):e1006629, 2018.
- [163] Hayden Tronolone, Alexander Tam, Zoltán Szenczi, JEF Green, Sanjeeva Balasuriya, Ee Lin Tek, Jennifer M Gardner, Joanna F Sundstrom, Vladimir Jiranek, Stephen G Oliver, et al. Diffusion-limited growth of microbial colonies. Scientific Reports, 8(1):5992, 2018.
- [164] Mick F Tuite and Brian S Cox. The genetic control of the formation and propagation of the [psi+] prion of yeast. Prion, 1(2):101–109, 2007.
- [165] Mick F Tuite and Tricia R Serio. The prion hypothesis: from biological anomaly to basic regulatory mechanism. Nature reviews Molecular cell biology, 11(12):823–833, 2010.

- [166] Libuše Váchová and Zdena Palková. How structured yeast multicellular communities live, age and die? FEMS yeast research, 18(4):foy033, 2018.
- [167] Paul Van Liedekerke, MM Palm, N Jagiella, and Dirk Drasdo. Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. Computational particle mechanics, 2(4):401–444, 2015.
- [168] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272, 2020.
- [169] Yanli Wang, Wing-Cheong Lo, and Ching-Shan Chou. A modeling study of budding yeast colony formation and its relationship to budding pattern and aging. PLoS computational biology, 13(11):e1005843, 2017.
- [170] Mya R Warren, Hui Sun, Yue Yan, Jonas Cremer, Bo Li, and Terence Hwa. Spatiotemporal establishment of dense bacterial colonies growing on hard agar. Elife, 8:e41093, 2019.
- [171] Joel C Watts, Aru Balachandran, and David Westaway. The expanding universe of prion diseases. PLoS pathogens, 2(3):e26, 2006.
- [172] Nigel P Weatherill and Oubay Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. International Journal for Numerical Methods in Engineering, 37(12):2005–2039, 1994.
- [173] Marc Weber. statannot, 2019.
- [174] Robert P Weinberg, Vera V Koledova, Hyeeri Shin, Jennifer H Park, Yew Ai Tan, Anthony J Sinskey, Ravigadevi Sambanthamurthi, and ChoKyun Rha. Oil palm phenolics inhibit the in vitro aggregation of β -amyloid peptide into oligomeric complexes. International Journal of Alzheimer’s Disease, 2018, 2018.
- [175] Reed B Wickner and Amy C Kelly. Prions are affected by evolution at two levels. Cellular and molecular life sciences, 73(6):1131–1144, 2016.
- [176] David J Wooten and Vito Quaranta. Mathematical models of cell phenotype regulation and reprogramming: Make cancer cells sensitive again! Biochimica et Biophysica Acta (BBA)-Reviews on Cancer, 1867(2):167–175, 2017.

- [177] Jing Xie, Li Tao, Clarissa J Nobile, Yaojun Tong, Guobo Guan, Yuan Sun, Chengjun Cao, Aaron D Hernday, Alexander D Johnson, Lixin Zhang, et al. White-opaque switching in natural *mtl a/α* isolates of *Candida albicans*: evolutionary implications for roles in host adaptation, pathogenesis, and sex. *PLoS Biol*, 11(3):e1001525, 2013.
- [178] Tao Yu and Jonathan Scolnick. Complex biological questions being addressed using single cell sequencing technologies. *SLAS technology*, 27(2):143–149, 2022.
- [179] Maksim Zakhartsev and Matthias Reuss. Cell size and morphological properties of yeast *Saccharomyces cerevisiae* in relation to growth temperature. *FEMS yeast research*, 18(6):foy052, 2018.
- [180] Hong Zhang, Weibin Gong, Si Wu, and Sarah Perrett. Studying protein folding in health and disease using biophysical approaches. *Emerging Topics in Life Sciences*, 5(1):29–38, 2021.
- [181] Qiushi Zheng, Qiuyu Zhang, Jian Bing, Xuefen Ding, and Guanghua Huang. Environmental and genetic regulation of white-opaque switching in *Candida tropicalis*. *Molecular microbiology*, 106(6):999–1017, 2017.