國立台灣大學電機資訊學院生醫電子與資訊學研究所

碩士論文

Graduate Institute of Biomedical Electronics and Bioinformatics

College of Electrical Engineering and Computer Science

National Taiwan University

Master’s Thesis

基於 ReaDDy2 的粒線體裂變-融合動力學反應擴散模型

ReaDDy2-based Reaction-Diffusion Model for Mitochondrial Fission-Fusion Dynamics

廖子言

Tzu-Yen Liao

指導教授：魏安祺 博士

Advisor: An-Chi Wei, Ph.D.

中華民國 114 年 8 月

August, 2025

doi:10.6342/NTU202503815

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

基於 ReaDDy2 的粒線體裂變-融合動力學反應擴散模型

ReaDDy2-based Reaction-Diffusion Model for Mitochondrial Fission-Fusion Dynamics

本論文係 廖子言 R12945007 在國立臺灣大學電機資訊學院生醫電子與資訊學研究所完成之碩士學位論文，於民國 114 年 07 月 24 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Biomedical Electronics and Bioinformatics, College of Electrical Engineering and Computer Science, National Taiwan University 24/07/2025 have examined a Master's thesis entitled above presented by Tzu-Yen, Liao, R12945007 candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

（指導教授 Advisor）

所長 Director: 林風

Acknowledgment

在本篇碩士論文完成之際，內心充滿無限的感激與敬意。首先，誠摯感謝我的指導教授魏安祺教授，感謝魏老師在研究過程中的細心指導與耐心陪伴。無論是在理論上的深入解析，或是在研究方向上的指引，老師都始終給予我最堅定的支持與鼓勵，讓我能夠在探索的道路上持續前行。

特別感謝文緯學長與祐德學長在我研究遇到困難時，總能即時伸出援手，不僅解決技術上的問題，也在思考上給予我極大的啟發。能在求學過程中遇到這麼熱心的前輩，是我極為珍惜的幸運。

同時，也感謝于珩與鎧帆兩位同屆夥伴，在實驗室中的並肩作戰與彼此打氣，使這段研究旅程多了許多溫暖與回憶。每一段深夜討論、每一次共同克服挑戰的時刻，都是我生命中難能可貴的經驗。

最後，感謝所有在我碩士求學過程中曾經給予我幫助與鼓勵的人，無論是家人、朋友，或是在生活中默默支持我的每一位。正是有了這些陪伴與支持，讓我得以完成這份論文，邁向下一段人生旅程。

doi:10.6342/NTU202503815

摘要

粒線體形成動態的絲狀網絡，其結構由融合和裂變之間的持續相互作用所塑造。了解這些結構轉變如何隨時間和在不同細胞條件下演變仍然是粒線體生物學領域的關鍵挑戰。在本研究中，我們提出了一個計算框架，該框架透過將基於粒子的擴散與結構和空間反應機制相結合來模擬粒線體網絡重塑。該模型基於節點連通性和空間鄰近性，編碼了受生物學啟發的融合和裂變規則，使分支、伸長和碎片化等拓撲事件能夠隨著時間的推移自然發生。

此模擬系統由實驗影像產生的骨架圖初始化，並透過雙層反應方案演化：結構反應基於局部圖規則重建內部拓撲結構，而空間反應在滿足鄰近性標準時合併各個組件。一系列詳細的輸出包括:粒子軌跡、拓撲檔案、反應日誌和度分佈:支援可視化和定量分析。單次運行模擬揭示了網路複雜性的動態變化，例如端點頻率的增加和平均聚合物長度的減少。對100次重複實驗進行多重運行統計平均，證明了度機率的穩健收斂性，並允許與實驗數據直接比較。

透過時序性顯微鏡影像的定量驗證，在對照組條件下表現出高度一致性，但在其他藥物如 FCCP 和 Mdivi-1 等條件下，會有誤差產生。這些結果表明，該模型雖然能捕捉粒線體重塑，但也指出了需要納入其他可能潛在的生物學機制，例如局部降解或生化回饋，才能獲得完全的準確性。總體而言，此模擬平台為探索粒線體動力學提供了一個分析的工具，在實驗假設檢定、藥物反應建模和細胞能量學的系統級研究中具有應用價值。

關鍵字：粒線體動力學、融合-裂變、網路建模、基於粒子的模擬、反應擴散、結構拓撲、ReaDDy2、基於圖的生物學、基於影像的驗證、粒線體碎片化、基於代理的建模、系統生物物理學

ABSTRACT

Mitochondria form dynamic, filamentous networks whose architecture is shaped by a continuous interplay between fusion and fission. Understanding how these structural transformations evolve over time and under different cellular conditions remains a key challenge in mitochondrial biology. In this study, we present a computational framework that simulates mitochondrial network remodeling by integrating particle-based diffusion with both structural and spatial reaction mechanisms. The model encodes biologically inspired rules for fusion and fission based on node connectivity and spatial proximity, enabling topological events such as branching, elongation, and fragmentation to emerge naturally over time.

The simulation system is initialized from experimental image-derived skeleton graphs and evolves through a dual-layer reaction scheme: structural reactions restructure internal topology based on local graph rules, while spatial reactions merge separate components when proximity criteria are met. A series of detailed outputs—including particle trajectories, topology files, reaction logs, and degree distributions, both visualization and quantitative analysis. Single-run simulations reveal dynamic transitions in network complexity, such as increases in endpoint frequency and reductions in average polymer length. Multi-run statistical averaging across 100 replicates demonstrates robust convergence of degree probabilities and allows for direct comparison with experimental data.

Quantitative verification of timing microscopy images showed high consistency under the control group conditions, but errors occurred under other drugs such as FCCP and Mdivi-1. These results suggest that while capturing metaphysical remodeling, the model also points to the need to incorporate other possible potential biological mechanisms, such as local degradation or biochemical feedback, in order to achieve complete accuracy. Overall, this simulation platform provides an analytical tool for exploring metasoma dynamics, with application value in systematic research on experimental hypothesis assays, drug response modeling, and cell energy.

doi:10.6342/NTU202503815

Keywords: Mitochondrial dynamics, fusion–fission, network modeling, particle-based simulation, reaction-diffusion, structural topology, ReaDDy2, graph-based biology, image-based validation, mitochondrial fragmentation, agent-based modeling, systems biophysics

doi:10.6342/NTU202503815

TABLE OF CONTENTS

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

LIST OF FIGURES

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

LIST OF TABLES

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

1. Chapter 1: Introduction

1.1 Background and Motivation

Mitochondria are not static organelles; rather, they form highly dynamic, interconnected networks that constantly adapt to the physiological demands of the cell. These networks exhibit continuous remodeling through coordinated cycles of fusion and fission, processes that balance connectivity with modularity to maintain cellular health. The dynamic restructuring of mitochondria plays essential roles in regulating energy metabolism, calcium signaling, redox balance, and apoptosis. Under physiological conditions, a finely tuned fusion–fission balance ensures proper mitochondrial function. However, disruptions to this balance—such as those caused by genetic mutations, toxins, or metabolic stress—can lead to fragmented or hyperfused networks, ultimately contributing to the pathogenesis of numerous diseases.

Computational modeling offers a powerful means to study mitochondrial morphology and its changes under various perturbations. Yet, many existing models remain either too abstract to capture spatial and topological details, or too detailed to allow for large-scale analysis or integration with imaging data. This work addresses this gap by developing a particle-based, graph-aware simulation framework that combines structural logic, spatial dynamics, and image-driven initialization to explore mitochondrial network evolution under different conditions. Our goal is to bridge mechanistic modeling with experimental validation, providing a platform that is both biologically grounded and computationally tractable.

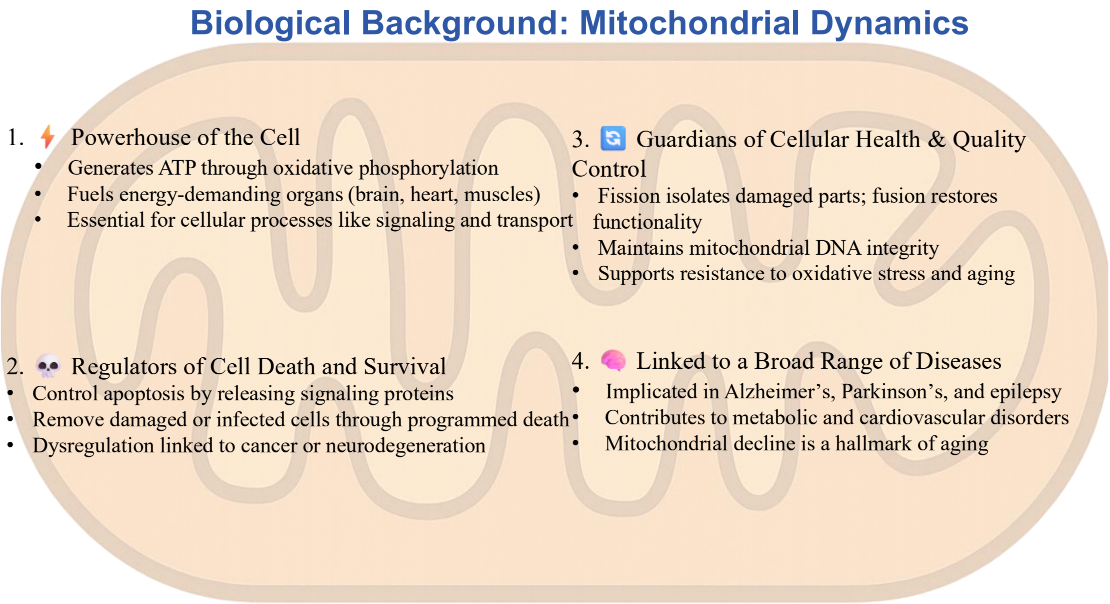
1.1.1 The Importance of Mitochondria in Human Health

Often described as the "powerhouses" of the cell (fig 1.1.), mitochondria are vital for generating ATP through oxidative phosphorylation [1]. Beyond energy production, mitochondria are central to a wide array of cellular processes including apoptosis [2], ROS signaling [3], and lipid metabolism [4]. Their proper functioning is critical for

doi:10.6342/NTU202503815

maintaining tissue homeostasis, particularly in energy-demanding systems such as the brain, heart, and skeletal muscles [5].

Importantly, dysfunction in mitochondrial dynamics has been implicated in a broad spectrum of diseases. Neurodegenerative disorders such as Parkinson’s and Alzheimer’s disease [6], metabolic syndromes [7], certain forms of cancer [8], and rare mitochondrial myopathies [9] all share characteristic changes in mitochondrial morphology and connectivity. Understanding the physical and regulatory principles underlying mitochondrial network behavior is therefore of critical importance—not only for basic cell biology, but also for developing diagnostic tools and therapeutic interventions [10].



Biological Background: Mitochondrial Dynamics

1. ⚡ Powerhouse of the Cell
   - Generates ATP through oxidative phosphorylation
   - Fuels energy-demanding organs (brain, heart, muscles)
   - Essential for cellular processes like signaling and transport

2. 💀 Regulators of Cell Death and Survival
   - Control apoptosis by releasing signaling proteins
   - Remove damaged or infected cells through programmed death
   - Dysregulation linked to cancer or neurodegeneration

3. 🔄 Guardians of Cellular Health & Quality Control
   - Fission isolates damaged parts; fusion restores functionality
   - Maintains mitochondrial DNA integrity
   - Supports resistance to oxidative stress and aging

4. 🧠 Linked to a Broad Range of Diseases
   - Implicated in Alzheimer’s, Parkinson’s, and epilepsy
   - Contributes to metabolic and cardiovascular disorders
   - Mitochondrial decline is a hallmark of aging

Figure 1.1. Biological roles of mitochondrial dynamics

This figure summarizes the multifaceted functions of mitochondria in cellular biology. Mitochondria act as the cell’s energy hub through ATP production (1), regulate apoptosis and survival mechanisms (2), preserve cellular integrity via quality control processes such as fission and fusion (3), and are implicated in a wide spectrum of diseases and aging-related decline (4).

doi:10.6342/NTU202503815

1.1.2 Mitochondrial Fusion and Fission

Mitochondrial morphology is controlled by a balance between two opposing processes: fusion, which promotes interconnectivity and mitochondrial complementation, and fission, which facilitates mitophagy and organelle distribution [11]. Fusion allows individual mitochondria to merge, sharing contents and diluting damage, while fission enables segregation of damaged segments and their subsequent degradation (fig 1. 2.) These processes are regulated by conserved GTPases—such as MFN1/2 and OPA1 for fusion, and DRP1 for fission—and are responsive to biochemical signals including changes in membrane potential, energy status, and reactive oxygen species [12].

At the structural level, fusion and fission events give rise to a range of topological motifs, from extended tubules and branching junctions to small, punctuate fragments [13]. Capturing this diversity requires a model that accounts not only for spatial proximity but also for the local connectivity of network nodes [14]. Additionally, many stress conditions or drug treatments, for instance, depolarizing agents like FCCP—can disrupt the balance and trigger widespread network fragmentation [15]. Mutations in MFN2 or altered DRP1 activity, as seen in some neuropathies and degenerative conditions, further exacerbate such disruptions [16].

Recent advances in computational modeling have enabled simulations of mitochondrial dynamics using network theory and time-resolved imaging, though incorporating biophysical parameters like membrane tension and curvature remains challenging [17],[18].

doi:10.6342/NTU202503815

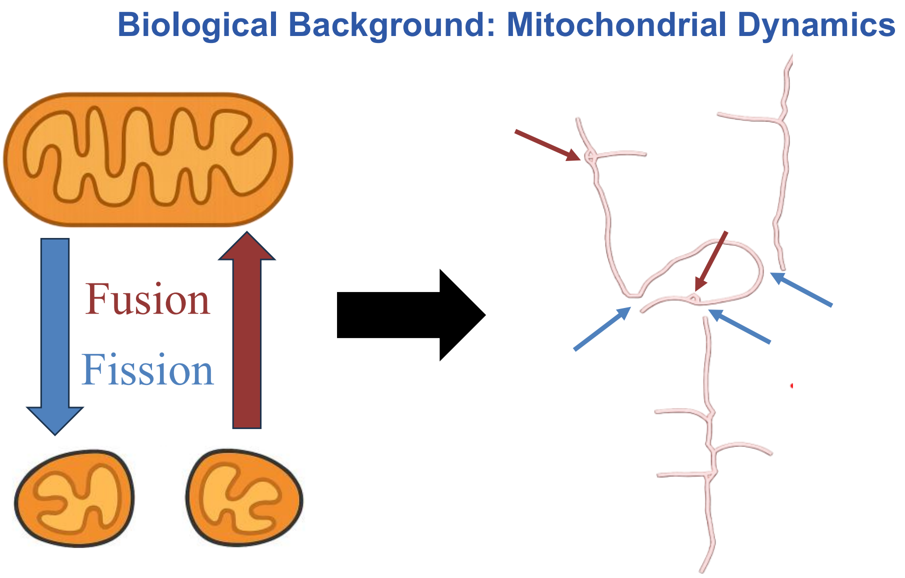


Figure 1.2. Schematic illustration of mitochondrial dynamics.

Mitochondrial morphology is shaped by the opposing processes of fission and fusion. Fusion combines fragmented mitochondria to form interconnected networks, while fission divides mitochondria to isolate damaged segments or facilitate distribution. These dynamic structural changes are visualized on the right, where red arrows indicate fusion events and blue arrows mark fission sites.

1.1.3 Physiological Determinants of the Fusion–Fission Equilibrium

The choreography of mitochondrial fusion and fission is inseparable from the cell’s physiological milieu. Bioenergetic cues—such as membrane potential ($\Delta\Psi$), ATP/ADP ratio, and local redox balance—feed directly into the activity of the core GTPases MFN1/2, OPA1, and DRP1 that govern membrane merger or scission [12]. High $\Delta\Psi$ and ample ATP bias MFN- and OPA1-mediated fusion, fostering tubular interconnectivity that maximizes metabolic complementation, whereas depolarization or oxidative stress recruits DRP1 to promote fission and facilitate mitophagy [15]. Calcium spikes, cytoskeletal tension, and cell-cycle checkpoints provide additional layers of regulation, ensuring that network remodeling is synchronized with metabolic demand and organelle inheritance [11].

Disruption of these physiological signals—through genetic lesions (e.g., MFN2 mutations) or pharmacological insults such as the uncoupler FCCP—shifts the balance toward fragmentation and can precipitate neuropathies or degenerative phenotypes [15, 16]. Capturing such state-dependent toggling between fusion and fission therefore requires models that integrate not only spatial proximity and topology [14] but also dynamic, physiology-linked reaction propensities. This mechanistic underpinning sets the stage for the computational simulations described in Section 1.1.4, where parameterized fusion and fission rates are varied systematically to reproduce—and ultimately predict—how physiological perturbations reshape the mitochondrial network [19–22].

1.1.4 The Importance of Computational Simulation for Mitochondria Network

While high-resolution fluorescence microscopy can capture striking snapshots of mitochondrial morphology, any single frame (Fig 1.3.) freezes only a moment of an intrinsically dynamic reticulum. Limited temporal resolution means that entire rounds of repositioning, fusion, or fission may pass unrecorded, so microscopy alone seldom reveals the full trajectory of network remodeling [19]. Even in time-lapse sequences, individual fusion and fission events often unfold at—or below—the diffraction limit and are therefore difficult to recognize by eye. Automated tracking algorithms are thus indispensable for detecting these events, but they, too, require validation.

Computational simulation provides that missing ground truth. By explicitly modeling the underlying particle interactions, simulations let us probe how local bond formation or breakage scales up to affect global properties such as network size, fragmentation, and complexity [20]. They also support systematic parameter sweeps—e.g., varying fusion rates or bond stiffness—to uncover critical thresholds that would be experimentally impractical to explore [21]. Because simulations deliver unrestricted temporal and spatial access to every particle and interaction, they furnish reference trajectories against which tracking algorithms can be benchmarked, thereby closing the experimental-computational loop. Advances in GPU acceleration and

doi:10.6342/NTU202503815

open-source libraries now make it feasible to construct multi-scale, data-initialized models that not only deepen theoretical understanding but also guide experimental design—for instance, by predicting mitochondrial responses to metabolic stress, drug treatment, or disease-associated mutations [22].

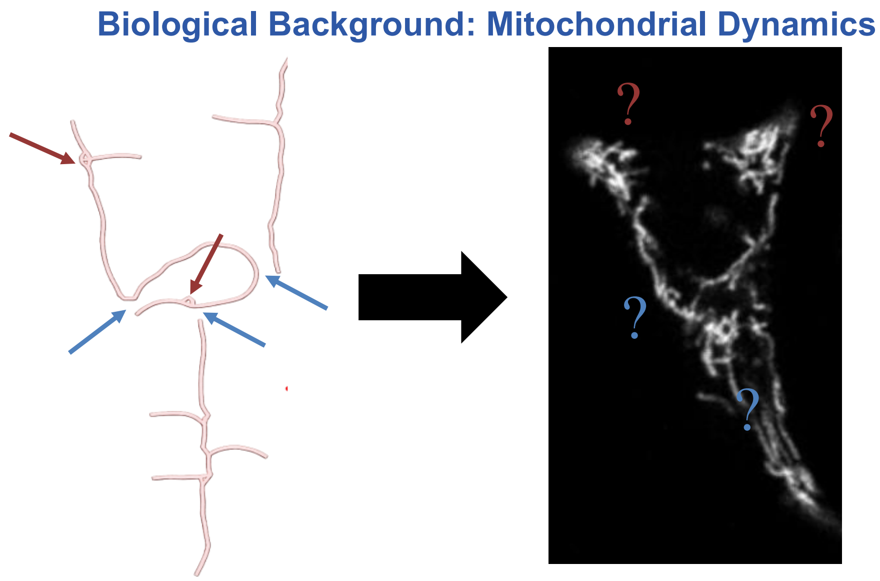


Figure 1.3. Mapping mitochondrial dynamics to experimental imaging.

The left panel illustrates the dynamic processes of mitochondrial fission (blue arrows) and fusion (red arrows). These structural transitions correspond to morphological features observed in live-cell fluorescence microscopy (right panel), where question marks highlight candidate regions undergoing fusion (red) or fission (blue). This visual link demonstrates the feasibility of identifying dynamic mitochondrial events from static images.

1.1.5 Development of Quantitative Network Analysis

In parallel with advances in computational modeling, new tools for image analysis have transformed our ability to quantify mitochondrial networks. Techniques such as skeletonization, graph extraction, and degree distribution analysis now allow researchers to convert microscopy images into structured, analyzable data. These tools make it possible to compare mitochondrial networks across conditions, track changes over time, and link morphological metrics to functional outcomes.

However, most experimental studies remain descriptive, and few integrate network analysis with predictive modeling. The ability to simulate networks initialized from real images, and to track their structural evolution using graph-theoretic measures creates a powerful synergy between computation and experimentation. This integration enables not only the generation of hypotheses but also the formal testing of how well a model reproduces biologically observed dynamics.

The present work leverages this integration by initializing simulations from image-derived graphs, simulating topological evolution with particle and reaction rules, and quantitatively comparing outputs to experimental benchmarks. In doing so, it provides a complete pipeline for understanding mitochondrial behavior from both a mechanistic and a system-level perspective.

1.2 Literature Review

The dynamic behavior of mitochondrial networks—driven by the fundamental processes of fusion, fission, biogenesis, and degradation—plays a crucial role in cellular homeostasis, signaling, and stress response. As direct experimental access to these processes remains limited by spatial and temporal resolution constraints, computational modeling has become an indispensable methodology for probing mitochondrial dynamics across scales. Over the past decade, the modeling paradigm has undergone a marked evolution, progressing from highly abstract, topology-based representations to sophisticated, data-informed simulations capable of capturing spatial heterogeneity, temporal continuity, and agent-level physiological variability. This literature review provides a structured examination of this progression, beginning with foundational graph-theoretical models and advancing through recent developments in mechanical, spatiotemporal, and agent-based frameworks. Through a comparative analysis of seminal contributions—including those of Sukhorukov et al. (2012), Holt et al. (2024), Wang et al. (2023), and—we delineate how successive

doi:10.6342/NTU202503815

generations of models have incrementally relaxed core assumptions, integrated empirical data, and expanded the explanatory and predictive scope of mitochondrial network simulations.

1.2.1 The Fundamental Logic of Computational Modeling

Computational modeling has become an essential methodology across the physical, life, and social sciences, providing a systematic framework for simulating, explaining, and predicting the behavior of complex systems [23][24]. A computational model is fundamentally an abstraction of a real-world system, constructed through mathematical or algorithmic representations of entities and their interactions. These models enable researchers to explore how system-level phenomena arise from localized rules, often under conditions that are analytically intractable or experimentally inaccessible.

At the core of computational modeling lies the principle of simplification and abstraction. Rather than attempting to replicate every molecular or environmental detail, models isolate and encode the essential rules that govern system behavior [25]. This formalization of hypotheses turns simulation into a laboratory for thought experiments, letting investigators probe the logical consequences of assumptions before turning to the bench.

Computational models typically serve three interrelated purposes: explanatory, predictive, and exploratory [26][27]. As explanatory tools, models help illuminate the mechanisms underlying observed phenomena—such as pattern formation, emergent organization, or non-linear feedback. Predictively, they can be used to forecast system behavior under new conditions, guiding experimental design or policy decisions. Exploratorily, models allow for the simulation of hypothetical scenarios that are impractical to study experimentally due to ethical, financial, or technical constraints.

doi:10.6342/NTU202503815

The utility of a computational model lies not in its capacity to replicate every detail of a system, but in its ability to generate insight through tractable approximations. Effective models often rely on assumptions such as discretizing time and space, lumping molecular species into classes, or limiting interactions to nearest neighbors. While these assumptions introduce simplifications, they are justified insofar as they preserve the dynamics of interest. As Ellis and Kopel (2015) argue, the value of a model is ultimately measured by its ability to connect micro-level mechanisms with macro-level observations in a coherent and testable framework.

In the context of biological systems—such as mitochondrial networks, computational models are particularly powerful because they can encode both spatial and topological features. Agent-based models and reaction-diffusion systems, for instance, allow the simulation of dynamic morphologies driven by local interactions such as fusion and fission. When initialized from empirical data, such as microscopy-derived skeletons, and validated through statistical comparison, these models offer a rigorous yet flexible approach for hypothesis testing and system-level inference.

The framework developed in this study reflects these modeling principles. It combines a particle-based reaction-diffusion environment with graph-based structural logic, enabling the simulation of mitochondrial network evolution under user-defined conditions. Through repeated simulation, parameter sensitivity analysis, and comparison to experimental data, the model exemplifies how computational abstraction can yield biologically meaningful insights—without requiring an exhaustive description of every molecular event.

1.2.2 Assumptions Underlying Mitochondrial Network Models

Modeling the mitochondrial network at the cellular scale involves several simplifying assumptions that are critical for tractability and simulation efficiency [20]. These assumptions are not arbitrary; rather, they are motivated by experimental

doi:10.6342/NTU202503815

observations and carefully tuned to reflect biologically plausible constraints. The aim is not to simulate mitochondria in exhaustive molecular detail, but to capture the essential mechanisms—namely fusion and fission—that govern network formation and transformation.

A foundational assumption in mitochondrial network modeling is that the structural state of the reticulum can be adequately represented as a discrete graph, where nodes correspond to specific functional or topological features (e.g., free ends, internal segments, branching points), and edges represent mitochondrial tubules. This abstraction enables the application of graph-theoretical and kinetic modeling tools while sidestepping the complexities of membrane dynamics or continuous mechanics.

One key simplification is the restriction of node degrees to three fundamental types—k = 1 (end points), k = 2 (linear connections), and k = 3 (branching nodes)(fig.1 4.). Higher-degree nodes (e.g., k > 3) are rare in fluorescence microscopy images and often result from optical artifacts or accidental projections of unrelated segments along the z-axis. Studies have shown that >96% of branching nodes in well-segmented mitochondrial skeletons are of degree 3[28], making this a robust constraint for modeling efforts.

The reaction mechanisms themselves are also idealized into four archetypal transformations: tip-to-tip fusion/fission and tip-to-side fusion/fission(fig 1.4.) . These are mapped onto changes in node degree within the network and represented through simple kinetic rules. For example, two free ends may merge to form a linear segment ($2X_1 \rightarrow X_2$), or a free end may attach to a linear segment to form a branch ($X_1 + X_2 \rightarrow X_3$). Corresponding fission events reverse these transformations. This minimalist reaction scheme effectively reproduces the observed network morphologies across multiple conditions.

doi:10.6342/NTU202503815

Spatial effects are treated implicitly rather than explicitly in many formulations. While real mitochondria move along cytoskeletal filaments and exhibit spatially heterogeneous behavior, the well-mixed approximation is often adopted, assuming a homogeneous intracellular environment [29]. This is justified by experimental simulations showing that frequent interactions with differently oriented cytoskeletal fibers result in rapid randomization of movement, making localized correlations negligible over longer timescales.

Another assumption is the time-scale separation between fusion/fission events and other mitochondrial processes, such as biogenesis, degradation (mitophagy), or signaling. These slower processes are excluded from the core dynamic equations, allowing the model to focus exclusively on topological evolution. However, the framework remains extensible and could incorporate these factors in future studies targeting long-term dynamics.

Finally, reaction rates are often assumed to be constant, independent of intracellular protein gradients or local biochemical conditions. Although more detailed models may include time- and concentration-dependent rate functions, constant-rate kinetics remain useful for evaluating general network behavior and for identifying phase transitions such as percolation, where small changes in rate ratios (e.g., fusion/fission) yield large structural reorganizations.

Together, these assumptions yield a robust, generalizable framework that enables the simulation of mitochondrial network evolution under various physiological and pathological conditions. By abstracting away unnecessary complexity, the model allows for focused investigation of the reticulum's structural dynamics, paving the way for hypothesis testing and eventual parameter calibration against experimental data.

doi:10.6342/NTU202503815

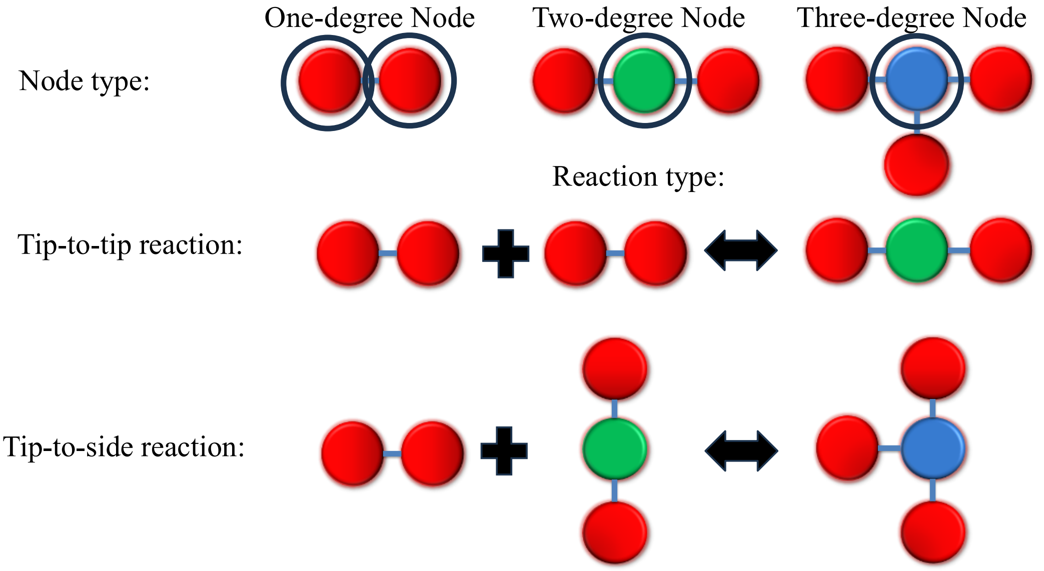


Figure 1.4. Topological classification of nodes and structural reaction types in mitochondrial networks.

This schematic defines node types based on connectivity: one-degree (red), two-degree (green), and three-degree nodes (blue). Two key structural reactions are illustrated—tip-to-tip fusion, which joins two linear segments to form a continuous path, and tip-to-side fusion, which creates a branched three-way junction. These reactions are reversible and underpin mitochondrial network remodeling.

1.2.3 Simulation focus on spatial constraints and mechanical forces

In this section, we critically assess the simulation strategy presented in Holt et al. (2024) [30] to evaluate how it aligns with or diverges from the modeling assumptions outlined in prior mitochondrial network frameworks, such as those in Sukhorukov et al. (2012) [20].

The model presented in Holt et al. represents a significant expansion of previous non-spatial and deterministic models by incorporating spatiotemporally resolved, stochastic simulations governed by a mechanistic Langevin dynamics framework (fig

doi:10.6342/NTU202503815

1.5.). In contrast to earlier formulations that assumed a homogeneous, well-mixed cellular environment with implicitly averaged fusion/fission interactions, the current simulation explicitly tracks individual mitochondrial segments in 3D or 2D space using mechanical rules for bending, stretching, steric interaction, and confinement.

Where earlier models encoded mitochondrial fission and fusion as phenomenological rules operating on a graph of node degrees 1–3, Holt et al. derive their interactions from a physically grounded scheme, modeling fusion as a probabilistic bond formation between well-aligned mitochondrial ends within a reaction radius and fission as a rate-dependent bond breakage determined by local mechanical constraints. Crucially, fusion is conditional not only on proximity but also on angular alignment—a departure from earlier topological rules that abstracted these mechanics away.

The assumptions about node degrees are preserved—both models limit network topology to nodes of degrees 1–3, excluding higher-order junctions due to their experimental rarity. However, Holt et al. challenges the assumption of uncorrelated reaction events. While previous models assumed that the stochastic independence of fusion and fission was adequate within a mean-field environment, the newer framework introduces time-delayed reactivation (“recharge”) mechanisms after fission events. This refinement acknowledges that newly fissioned mitochondria are not instantly fusion-competent—an insight aligned with empirical observations but absent from traditional models.

Another key divergence lies in how simulations relate to measurable quantities. Holt et al. directly calibrates their simulation using empirical fusion/fission rates and mitochondrial density data from mammalian and yeast cells [31], allowing dimensionless parameters like k_f to be interpreted in minutes. Furthermore, they map microscopic parameters to macroscopic observables (e.g., segment length, cluster size, connectivity) using a combination of agent-based simulation and mean-field reduction, thereby creating a bridge between local dynamics and emergent structure.

In sum, the simulation set up by Holt et al. does not violate the assumptions in earlier models but rather *extends and refines* them. It transforms a phenomenological system into a mechanically consistent, biophysically realistic model that remains compatible with the topological constraints established in prior work, while introducing time-dependent, geometry-sensitive rules that better reflect the physical and biochemical realities of mitochondrial remodeling.

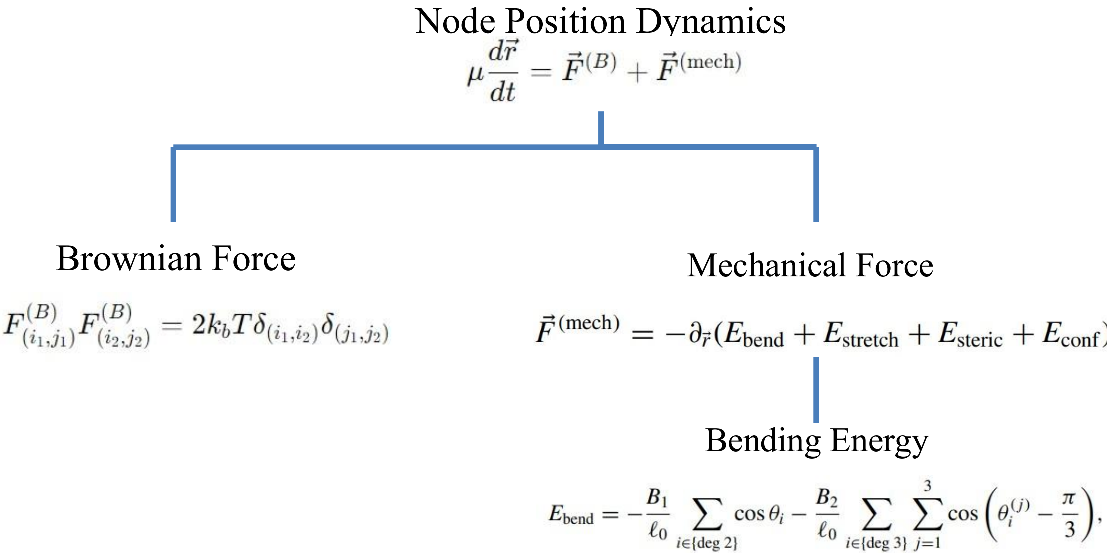

Node Position Dynamics
$\mu \frac{d\vec{r}}{dt} = \vec{F}^{(B)} + \vec{F}^{(\mathrm{mech})}$
Brownian Force
$F^{(B)}_{(i_1,j_1)} F^{(B)}_{(i_2,j_2)} = 2k_bT\delta_{(i_1,i_2)}\delta_{(j_1,j_2)}$
Mechanical Force
$\vec{F}^{(\mathrm{mech})} = -\partial_{\vec{r}}(E_{\mathrm{bend}} + E_{\mathrm{stretch}} + E_{\mathrm{steric}} + E_{\mathrm{conf}}$
Bending Energy
$E_{\mathrm{bend}} = -\frac{B_1}{\ell_0} \sum_{i \in \{\deg 2\}} \cos\theta_i - \frac{B_2}{\ell_0} \sum_{i \in \{\deg 3\}} \sum_{j=1}^{3} \cos\left(\theta_i^{(j)} - \frac{\pi}{3}\right),$

Figure 1.5. Forces governing node position dynamics in the mitochondrial network model.

The Node movement is governed by a combination of stochastic Brownian forces and deterministic mechanical forces. The total force includes contributions from bending energy, stretching, steric repulsion, and conformational constraints. Bending energy is specifically defined for two- and three-degree nodes to penalize deviations from linear or Y-shaped configurations, respectively.

1.2.4 Advances in Dynamic Simulation and Temporal Network Tracking

Building upon earlier graph-based representations of mitochondrial networks that were largely static or limited to topology without spatial-temporal integration, Wang et al. (2023) introduce MitoTNT, a temporal network tracking framework designed to extract and quantify four-dimensional (4D) mitochondrial dynamics from live-cell lattice light-sheet microscopy data [32][20]. Their work notably extends prior assumptions and modeling limitations described by Sukhorukov et al. (2012), who modeled mitochondrial networks through discrete tip-to-tip and tip-to-side fusion/fission events within a non-spatial, well-mixed graph-based formalism. The primary innovation in the MitoTNT framework is the explicit integration of spatial coordinates and time stamps to reconstruct dynamic mitochondrial behavior at sub-second resolution, enabled by lattice light-sheet imaging [34].

To validate the tracking algorithm, Wang et al. implement a reaction-diffusion simulation using ReaDDy22, a particle-based platform that captures motion, branching structure, and remodeling events at a mesoscopic scale [33][38]. Unlike earlier assumptions of spatial homogeneity or well-mixed conditions, their model includes spatial heterogeneity and explicitly simulates motility of skeleton nodes under realistic reaction constraints. Notably, fission and fusion are implemented as structural reactions—where bonds are created or deleted between connected mitochondrial particles—and these reactions are calibrated to match experimental observations of network morphology and dynamics.

This approach relaxes several of the simplifying assumptions from earlier models:

- Spatial tracking is no longer abstracted; each mitochondrial segment, including bulk (degree-2) nodes, is discretized and tracked frame-by-frame in 3D space.

doi:10.6342/NTU202503815

- Temporal correlations and motion patterns are explicitly measured, revealing patterns such as rotational or diffusive motility, and dynamic remodeling consistent with asymmetric fission and fusion events [36].

- The simulation incorporates topology-informed tracking via optimization (linear assignment problem with spatial and topological cost terms), ensuring continuity in temporal graphs even in dense or complex network regions.[37]

- It also captures non-trivial effects of pharmacological perturbations, such as reduced network resilience or increased reachability after drug treatment—phenomena not modeled in previous theoretical frameworks.

Hence, while the foundational principles from earlier models (e.g., importance of tip-mediated events, emphasis on node degree distributions, and reaction-based remodeling) are preserved, the MitoTNT framework extends them into a full spatiotemporal domain and enables high-throughput validation against experimental data. In doing so, it bridges the gap between conceptual graph models and empirical network behavior observed in live cells.

1.3 Research Significance and Impact

As outlined in the preceding literature review, the modeling of mitochondrial networks has progressed from simplified, topology-driven abstractions toward increasingly detailed frameworks that incorporate spatial, temporal, and dynamic remodeling processes. This evolution has enabled more biologically meaningful simulations of mitochondrial behavior; however, many existing models are constrained by fixed assumptions, limited parameter flexibility, or dependence on

doi:10.6342/NTU202503815

specific experimental datasets. These limitations can restrict their utility in exploratory studies or theoretical investigations of network behavior under diverse or hypothetical conditions.

This thesis addresses these challenges by developing a flexible computational framework for simulating mitochondrial network dynamics. The model is designed to incorporate key structural and behavioral features of mitochondria—such as fusion and fission processes, spatial topology, and network remodeling—while allowing users to assign parameter values freely based on theoretical considerations or intended experimental scenarios. This parameter flexibility facilitates a broad range of investigations, including sensitivity analysis, phase-space exploration, and hypothesis generation, without requiring direct integration of experimental datasets.

The significance of this work lies in its ability to support exploratory and theoretical studies of mitochondrial networks across a variety of physiological and pathological conditions. By simulating how different fusion/fission rates, node configurations, or spatial constraints influence global network properties—such as degree distribution, connectivity, or fragmentation, the model provides insights into the fundamental mechanisms governing mitochondrial morphology and stability. Furthermore, the framework enables researchers to test how shifts in model parameters may mimic stress responses or therapeutic interventions, thereby generating predictions that can inform future empirical studies.

In summary, this thesis contributes a generalizable and customizable simulation platform for studying mitochondrial network dynamics. By balancing structural realism with parameter flexibility, it offers a valuable tool for probing the emergent behavior of mitochondrial systems and for guiding experimental inquiry through computational experimentation.

doi:10.6342/NTU202503815

1.4 Specific Aim

The overarching goal of this research is to construct a Python-based computational pipeline that couples segmented fluorescence microscopy with particle-level reaction–diffusion simulation (ReaDDy2) to analyze and predict mitochondrial network dynamics. At its core, the framework links image-derived structure to a physics-grounded model so that hypotheses about fusion–fission regulation can be tested quantitatively across pharmacological conditions.

This aim is pursued through a six-stage methodology. First, high-resolution mitochondrial images in proprietary. czi format are converted into analysis-ready stacks, eliminating user bias by directly importing segmented skeletons rather than relying on manual tracing. Second, skeletonisation and graph reconstruction in Python (using scikit-image, skan, and NetworkX) translate pixel data into explicit node–edge lists, providing a precise topological description ready for simulation. Third, these graphs are instantiated in ReaDDy2 as three-dimensional networks of bonded particles, creating a flexible platform capable of simulating fusion and fission under diverse treatment regimes. Fourth, stochastic reaction rules govern bond formation and dissociation, while diffusion and bending forces drive spatial reconfiguration; throughout, the engine records heterogeneous data stream coordinates, bond states, reaction counts, and event positions—to maximise downstream analytical robustness. Fifth, the evolving topology is rendered as PDB/PSF sequences, enabling visualisation that can be fed into tracking algorithms for independent validation of reconstruction accuracy. Finally, statistical ensemble runs perform cross-validation between simulated ground truth and microscopy observations, allowing the identification of latent mitochondrial states that may underlie observed image variability.

By integrating unbiased image import, treatment-agnostic simulation, multi-modal data logging, high-fidelity visualisation, and systematic cross-validation,

doi:10.6342/NTU202503815

the pipeline delivers a reproducible and extensible environment for probing how molecular interventions reshape mitochondrial architecture over time.

2. Chapter 2: Methods and Materials

2.1 ReaDDy2 Package Overview

ReaDDy2[33] (Reaction-Diffusion Dynamics) is a particle-based simulation framework designed for modeling reaction-diffusion systems at a spatially minimal scale. It is particularly useful for simulating biochemical and biophysical processes where spatial diversity and stochastic effects play a significant role. Unlike traditional mean-field or continuum-based reaction-diffusion models, ReaDDy2 explicitly represents particles as discrete entities and simulates their movement and interactions within a defined environment.

2.1.1 Key Features and Capabilities

1. Particle-Based Modeling
   - ReaDDy2 using individual particles with distinct properties to simulate molecules, proteins, or other biological entities.
   - To simulate the realistic biological entities through space, Particles diffuse, or transport follow by the Brownian motion.
2. Reaction Pathways and Kinetics
   - Users can define complex reaction networks, naming reaction rules no matter if they are unimolecular or bimolecular reaction.
   - To show reaction event is stochastic, the framework includes reaction propensity calculations.
3. Diffusion and Spatial Constraints
   - Each particle species is assigned a diffusion coefficient, allowing particles to have different mobility properties in the simulation.
   - To reflect biological or physical constraints or to avoid edge effects, the environment can be constrained with boundary conditions, obstacles, and classified regions.
4. Stochastic Reaction-Diffusion Dynamics
   - To ensuring precise capturing reaction kinematic at microscopic scales, ReaDDy2 package implement stochastic simulation algorithms such as the Gillespie reaction handler.

doi:10.6342/NTU202503815

- Because of rare or noise-sensitive biochemical, this method generally more useful than a deterministic model when capturing reaction.

5. Topology and Coarse-Grained Modeling
    - ReaDDy2 package supports the representation of macromolecular structures, such as polymers, networks, or protein complexes, through topology-based modeling.
    - This feature allows users to model complex biomolecular modification and dynamic rearrangements.
6. High-Performance Computing and Parallelization
    - ReaDDy2 using CPU-based parallel computing to provide efficient

2.1.2 Simple System Setup

The simulation environment was defined as a 10 x 10 x 10 nm^3 cubic box with periodic boundary conditions to prevent boundary effect. The system initially contains N particles distributed randomly within the box size. The primary species include:

- A: represents as reactant molecule
- B: represents as second reaction molecule
- C: Product formed via fusion of A and B

Each species was assigned a diffusion constant

- A, B: $D_{A,B} = 0.1\, \mu m^2/s$
- C: $D_c = 0.05\, \mu m^2/s$

The primary reaction modeled was a fusion event:

$$A + B \rightarrow C$$

This reaction was implemented at a reaction rate constant $k_{fu} = 1.0$ and reaction radius of 1 nm. Reaction kinetics were simulated by the Gillespie reaction handler, ensuring precise stochastic reaction events happen.

2.1.3 Simulation Workflow

The simulation followed a structured workflow:

doi:10.6342/NTU202503815

1. System Initialization: defined the simulation area, particles or topologies species, potential, structural topology, reaction parameters, and placed initial particles randomly.
2. Execution Phase: the simulation was executed for 5000 steps with a timestep of 1e-3 ns.
3. Data Collection: Tracked species observes such as number of particles, reaction count, and spatial distribution over time.
4. Post-processing: Analyzed the system using particle count statistics and spatial visualization using VMD software

2.1.4 Implementation in Python

The simulation was implemented in Python using ReaDDy2, A simplified version of the implementation is shown below:

```python
import ReaDDy2

# Define the Reaction-Diffusion System
system = ReaDDy2.ReactionDiffusionSystem(box_size=[10, 10, 10])

# Add particle species with their respective diffusion constants
system.add_species("A", diffusion_constant=0.1)
system.add_species("B", diffusion_constant=0.1)
system.add_species("C", diffusion_constant=0.05)

# Define a binary fusion reaction: A + B -> C
# - rate: how likely the reaction is to occur when particles are close
# - radius: max distance within which A and B can react
system.reactions.add("fusion: A + B -> C", rate=1.0, radius=1.0)

# Set Up the Simulation
simulation = system.simulation(kernel="CPU")

# Use 16 CPU threads for better performance
simulation.kernel_configuration.n_threads = 16

```

doi:10.6342/NTU202503815

```python
# Add initial positions of A and B particles
simulation.add_particles("A", positions=[[1, 1, 1], [2, 2, 2]])
simulation.add_particles("B", positions=[[3, 3, 3], [4, 4, 4]])

# Define output file for the simulation trajectory
simulation.output_file = 'simple_system.h5'

# Use Euler–Brownian Dynamics integrator for particle motion
simulation.integrator = "EulerBDIntegrator"

# Use Gillespie algorithm to handle stochastic reactions
simulation.reaction_handler = "Gillespie"

# Set observables:
# - Save full particle trajectories every 300 steps
simulation.record_trajectory(stride=300)
# - Observe and log individual particle positions every step
simulation.observe.particles(stride=1)
# - Topology tracking not used here but included for completeness
simulation.observe.topologies(stride=300)

# Show simulation progress every 10 steps
simulation.progress_output_stride = 10
simulation.show_progress = True

# Run the Simulation
simulation.run(n_steps=5000, timestep=1e-3)

# Convert Output for Visualization in VMD
trajectory = ReaDDy2.Trajectory('simple_system.h5')
trajectory.convert_to_xyz()
```

2.1.5 ReaDDy2 Package Function Explanation

1. The box size

```
system = ReaDDy2.ReactionDiffusionSystem([X, Y, Z]
```

If we set box_size = (X, Y, Z), the center of box will be set at (0, 0, 0) and span

- x-axis: [-X/2, X/2)
- y-axis: [-Y/2, Y/2)
- z-axis: [-Z/2, Z/2)

2. Periodic boundary conditions

```
system = ReaDDy2.ReactionDiffusionSystem([1,1,1], periodic_boundary_conditions=[False, True, True]) #[X, Y, Z]
```

In most cases, if we the set box size also needs set the boundary conditions avoid boundary effect.

3. Temperature

```
Temperature = T * ReaDDy2.units.kelvin
```

- T: represents as the variable of the temperature in Kelvin
- ReaDDy2.units.kelvin: represents as the unit conversion to make sure the temperature is implemented correctly.

If you do not set the specified temperature, the default temperature will be 293K

4. Particle species

```
# default unit:  〚nm〛 ^2 s^(-1)
system.add_species("A", diffusion_constant=1.)
# set up the new physical unit: 〚km〛 ^2  〚hour〛 ^(-1)
system.add_species("B", diffusion_constant=2. * ReaDDy2.units.km**2 / ReaDDy2.units.hour)
```

This is one of the most important functions in ReaDDy2 package. To add particles in the simulation, we given particle "name" and its diffusion constant D with units of $length^2 time^{-1}$. The purpose of diffusion constant is deciding the magnitude of random displacement follow by the governing dynamics using overdamped Langevin equation:

$$\frac{dx(t)}{dt} = -D\frac{\nabla V\big(x(t)\big)}{k_b T} + \xi(t)$$

- x(t): represents as a vector $\in R^3$ related to the instantaneous position of a particle at time t.
- V: represents external potential field
- -∇V: represents the gradient of the potential, the negative sign means the force will push the particle toward lower energy area.
- k_b: represents the Boltzmann constants.
- T: represent the kelvin temperature.
- $\xi(t)$: represents random noise of velocity with formula and condition:

  $$\langle \xi(t) \rangle = 0, \qquad \langle \xi(t)\xi(t') \rangle = 2D\delta(t - t')$$

  - The first term means the noise has no bias, totally random.
  - The second term means the noise is time-uncorrelated, meaning that it changes randomly and does not depend on past value.

5. Reaction (this function only involved isolated particles interaction) (unused)

A. Conversion:

```
system.reactions.add_conversion(name="conv", type_from="A", type_to="B", rate=0.1)
```

An isolated particle of type A can be transferred into type B with fixed rate constant λ:

$$A \overset{\lambda}{\rightarrow} B$$

B. Decay:

```
system.reactions.add_decay(name="decay of A", particle_type="A", rate=0.1)
```

An isolated particle can be vanished with fixed rate constant λ:

$$A \overset{\lambda}{\rightarrow} \emptyset$$

C. Fusion:

```
system.reactions.add_fusion( name="fus", type_from1="A", type_from2="B", type_to="C", rate=0.1, educt_distance=2.)
```

An isolated particle of type A can be fused with another isolated particle of type B to produce a particle of type C with fixed rate constant λ and fixed react radius R.

$$A \overset{R}{+} B \overset{\lambda}{\rightarrow} C$$

D. Fission:

```
system.reactions.add_fission( name="fis", type_from="C", type_to1="A", type_to2="B", rate=0.1, product_distance=2.)
```

An isolated particle of type C can be dissociated into two particles of type A and B with a fixed constant rate λ and fixed distance R between two particles after the fission happens.

$$C \overset{\lambda}{\rightarrow} A \overset{R}{+} B$$

E. Enzymatic:

```
system.reactions.add_enzymatic( name="enz", type_catalyst="C", type_from="A", type_to="B", rate=0.1, educt_distance=2.)
```

An isolated particle of type A can be reacted with enzyme of type C to produce a particle of type B with fixed rate constant λ and fixed reaction radius R.

$$A \overset{R}{+} C \overset{\lambda}{\rightarrow} B + C$$

6. Potentials

A. External potential

a. box potentials

```
system.box_size=[3, 3, 3]
system.potentials.add_box( particle_type="A", force_constant=10.,
 origin= [-1, -1, -1], extent= [2, 2, 2]) # note that box size and
box potential size are totally independent.
```

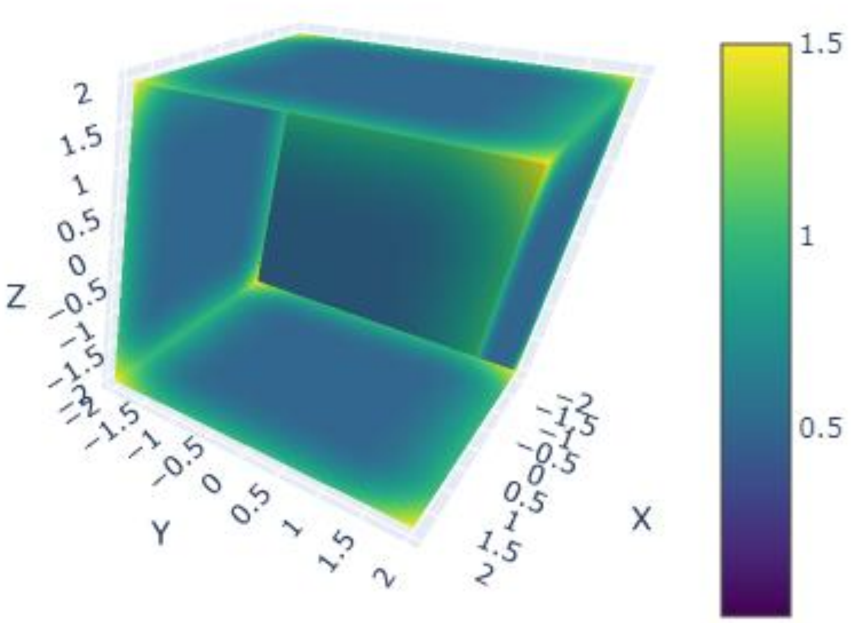


Figure 2.1. Visualization of an external box potential in 3D space.

This plot illustrates the spatial profile of a box potential applied to particles of type "A" in a simulation domain of size [3, 3, 3]. The box potential is defined independently by its own origin ([-1, -1, -1]) and extent ([2, 2, 2]), generating a harmonic confinement with force constant 10. The color scale represents the potential energy magnitude across the 3D space.

Since if we don’t want a periodic boundary and try avoiding boundary effect, we usually set box potential inside the box (fig 2.1.). The logic of the code is we can define every single type of particle’s box potential with different force constant and coverage. And the logic of setting coverage is you have origin locate at front lower left and extend to the back upper right. From the above code example, the coverage of the potential will be x: [-1, 1], y: [-1, 1], z: [-1, 1]. The potential energy term given by:

$$V(x) = \sum_{i=1}^{d} \begin{cases} 0, & if\ x_i \in C_i \\ \frac{1}{2}kd(x_i, C_i)^2, & otherwise \end{cases}$$

- C_i: represents three-dimension interval with $[origin_i,\ origin_i + extent_i]$
- k: represents the force constant
- $d(x_i,\ C_i)$: represent the shortest distance between the particle's position x_i and the nearest boundary of the box.

Since the box potential is soft potential, it means particles may go through the boundary of box potentials and drag back immediately. Beware setting the size and the force constant.

b. Spherical potential (unused)

This kind of potential can be divided into three types:

I. Spherical exclusion

```
system.box_size = [3, 3, 3]
system.potentials.add_sphere_out(particle_type="A", force_constant=10., origin=[0, 0, 0], radius=3.)
```

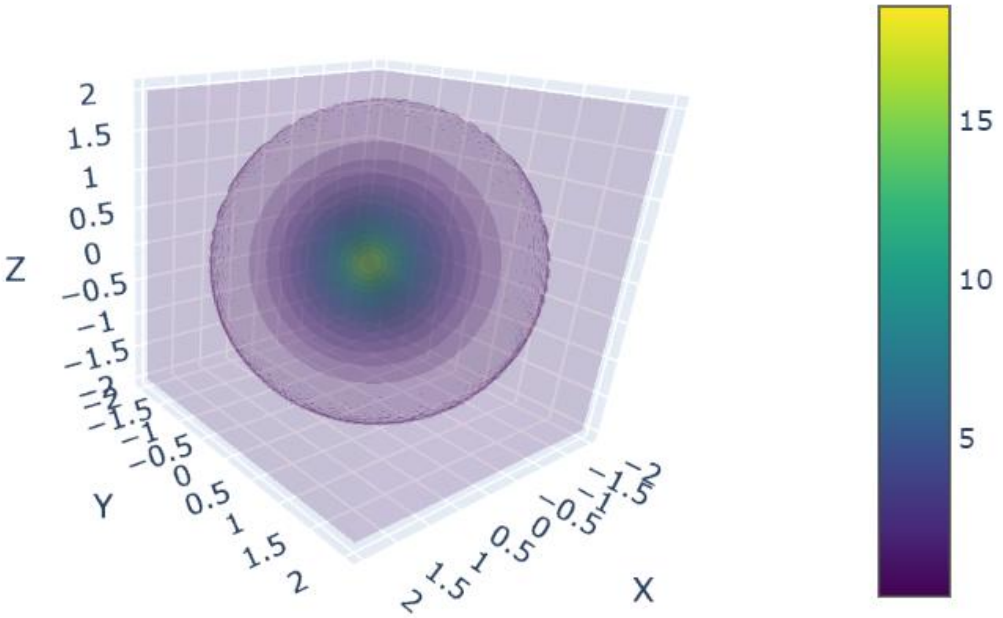


Figure 2.2. Visualization of a spherical exclusion potential in 3D space.

The figure visualizes a radial potential applied using add_sphere_out, where particles of type "A" experience a repulsive harmonic force outside a spherical boundary of radius 3.0 centered at the origin. The system box size is set to [3, 3, 3], and the force constant is 10. The color scale indicates the increasing potential energy as particles approach or exceed the boundary.

Add this kind of potential to prevent particles of a specific type from entering the interior of a defined sphere(fig 2.2.). The associated energy contribution is described by the following expression:

$$V(x) = \begin{cases} \frac{1}{2}k(||x - c\,||_2 - r)^2, & if\ ||x - c\,||_2 < r \\ 0, & otherwise \end{cases}$$

- k: represent the force constant.
- c: $\in\ R^3$, represent the center of the sphere.
- r: $\in\ R_{>0}$, represent the radius of sphere.

This potential is also soft potential.

II. Spherical inclusion

```
system.box_size = [3, 3, 3]
system.potentials.add_sphere_in(particle_type="A", force_constant=10., origin=[0, 0, 0], radius=1.)
```

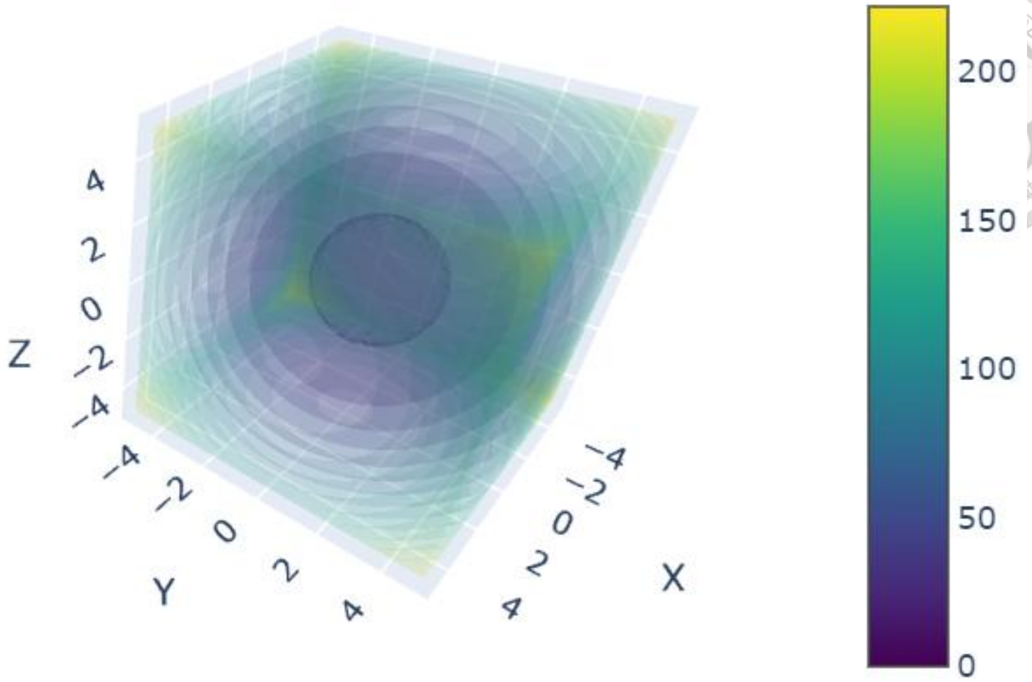


Figure 2.3. Visualization of a spherical inclusion potential in 3D space.

The figure visualizes a radial potential applied using add_sphere_in, where particles of type "A" experience a repulsive harmonic force inside a spherical boundary of radius 3.0 centered at the origin. The system box size is set to [3, 3, 3], and the force constant is 10. The color scale indicates the increasing potential energy as particles approach or exceed the boundary.

Applies a spherical potential that confines particles of a designated type within the boundary of a defined sphere (fig 2.3.). The corresponding energy expression is as follows:

$$V(x) = \begin{cases} \frac{1}{2}k(||x-c\,||_2 - r)^2, & if||x-c||_2 \geq r, \\ 0\,, otherwise \end{cases}$$

- k: represent the force constant.
- c: $\in R^3$, represent the center of the sphere.
- r: $\in R_{>0}$, represent the radius of sphere.

This potential is also soft potential.

doi:10.6342/NTU202503815

III. Spherical barrier

```python
system.box_size = [3, 3, 3]
# as a barrier
system.potentials.add_spherical_barrier(particle_type="A", height=1.0, width=0.1, origin=[0, 0, 0], radius=1.)
# sticky
system.potentials.add_spherical_barrier(particle_type="A", height=-1.0, width=0.1, origin=[0, 0, 0], radius=1.)
```

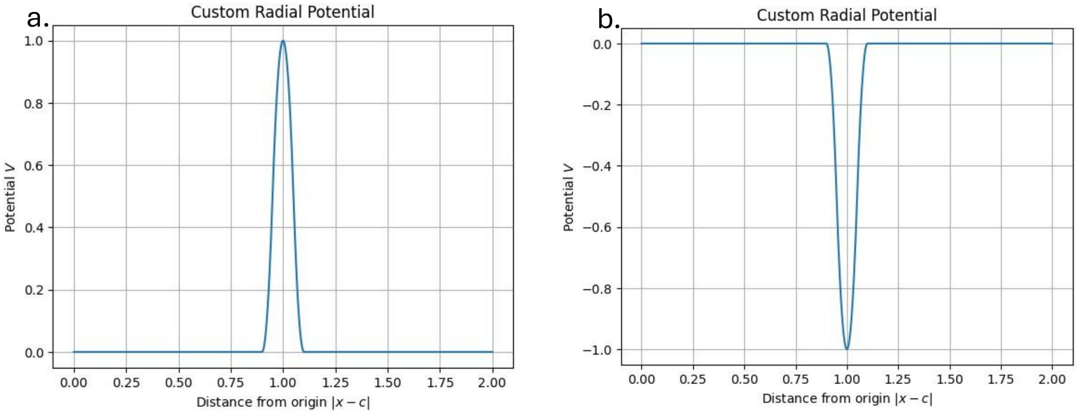


Figure 2.4. Custom spherical barrier potentials in ReaDDy2 simulations.

(a) A repulsive spherical barrier centered at radius 1.0 with height = +1.0 and width = 0.1 creates a sharp energy peak, preventing particles from crossing the defined shell.

(b) An attractive ("sticky") barrier with height = –1.0 forms a narrow energy well at the same radius, allowing localization of particles near the shell.

Both potentials are defined radially from the origin in a box of size [3, 3, 3] using add_spherical_barrier.

This potential creates a radial barrier centered at a specified origin, defined by a particular radius. It is characterized by an energy height and a finite width. If the height is negative, the potential functions as an attractive or "adhesive" spherical region. The potential is constructed using segments of harmonic functions, ensuring that the energy profile remains smooth and continuous, while the resulting force is continuous but not differentiable. The corresponding energy expression is:

$$\begin{cases} 0,\ \ if\ ||x-c||_2 < r-w \\ \frac{2h}{w^2}(\left||x-c|\right|_2 - r + w)^2,\ \ if\ r-w\ \le\ \left||x-c|\right|_2 < r-\frac{w}{2} \\ h-\frac{2h}{w^2}\left(\left||x-c|\right|_2 - r\right)^2, if\ r-\frac{w}{2}\ \le\ \left||x-c|\right|_2 < r+\frac{w}{2} \\ \frac{2h}{w^2}\left(\left||x-c|\right|_2 - r - w\right)^2,\ \ if\ r+\frac{w}{2}\ \le\ \left||x-c|\right|_2\ <\ r+\frac{w}{2} \\ 0,\ \ otherwise \end{cases}$$

This potential is also soft potential.

B. Pair potentials

a. Harmonic repulsion

```
# we can set the pair potential between same particle type with same radius
system.potentials.add_harmonic_repulsion("A", "A", force_constant=10., interaction_distance=5.)
system.potentials.add_harmonic_repulsion("B", "B", force_constant=10., interaction_distance=6.)
#or we can set the pair potential between different particle types with its own radius.
system.potentials.add_harmonic_repulsion("A", "B", force_constant=10., interaction_distance=2.5+3.)
```

To avoid particles overlapping or to simulate a radius of a particle type, we can add potential pairs between them. The potential formula given by:

$$V(x_1, x_2) = \begin{cases} \frac{1}{2}k\left(\left||x_1 - x_2|\right|_2 - r\right)^2,\ \ if\left||x_1 - x_2|\right|_2 <\ r \\ 0,\ \ otherwise \end{cases}$$

- k: represents the force constant.
- $\left||x_1 - x_2|\right|_2$: represents the shortest distance between particles
- r: represents the radius set in the code

The harmonic repulsion is also soft potential.

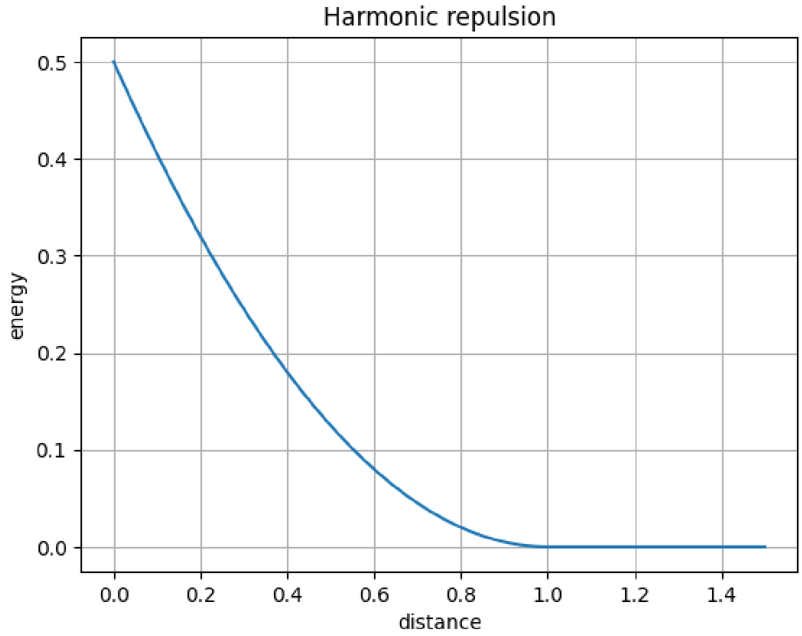


Figure 2.5. Harmonic repulsion pair potential.

This plot illustrates the energy profile of a harmonic repulsion potential, which penalizes close-range interactions between particles. The potential energy decreases smoothly with increasing interparticle distance, reaching zero beyond the cutoff range. This interaction is typically used to prevent particles overlapping in simulations.

b. Weak interaction piecewise harmonic (unused)

```
system.potentials.add_weak_interaction_piecewise_harmonic( "A", "B", force_constant=10., desired_distance=0.5, depth=1., cutoff=1.)
```

$$V\left(\|x_1 - x_2\|_2\right) = V(r) = \begin{cases} \frac{1}{2}k(r-d)^2 - h, \ \ if\ r < d \\ \frac{h}{2}\left(\frac{r_c - d}{2}\right)^{-2}(r-d)^2 - h, \ \ if\ d \le r\ < d + \frac{r_c - d}{2} \\ -\frac{h}{2}\left(\frac{r_c - d}{2}\right)^{-2}(r - r_c)^2, \ \ if\ d + \frac{r_c - d}{2} \le r < r_c \\ 0, \ \ otherwise \end{cases}$$

- k: represents the force constant.

- $||x_1 - x_2||_2$: denotes the Euclidean distance between two particles,
- r: represents the interparticle distance.
- d: represents the preferred equilibrium distance.
- h: represents the depth of the potential well.
- r_c: represents the cutoff radius beyond which the interaction is zero.

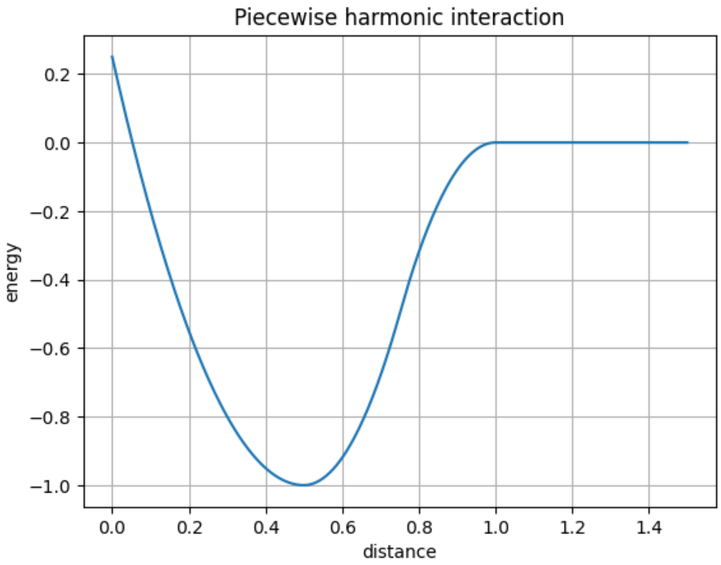


Figure 2.6. Piecewise harmonic interaction potential.

This potential defines a short-range attractive interaction that transitions smoothly to zero beyond a cutoff distance. The energy well promotes moderate attraction between particles, while ensuring bounded interaction strength and computational stability.

c. Lennard-Jones (unused)

```
system.potentials.add_lennard_jones("A", "B", m=12, n=6, cutoff=2.5, shift=True, epsilon=1.0, sigma=1.0)
```

$$V_{LJ_{trunc}}(r) = \begin{cases} k\left[\left(\frac{\sigma}{r}\right)^m - \left(\frac{\sigma}{r}\right)^n\right] - V_{LJ}(r_c), & if\ r\ \leq r_C \\ 0, & otherwise \end{cases}$$

$where$:

- $V_{LJ}(r) = k\ [\left(\frac{\sigma}{r}\right)^m - \left(\frac{\sigma}{r}\right)^n]$

- $k = \frac{-\varepsilon}{\left(\frac{\sigma}{r_{min}}\right)^m - \left(\frac{\sigma}{r_{min}}\right)^n}$

Parameter Descriptions:

- r = $||x_1 - x_2||_2$: denotes distance between two particles.
- σ: represents distance at which the potential is zero.
- ε: represents depth of the potential well, with $V_{LJ}(r_{min}) = -\varepsilon$.
- r_{min}: represents the distance at which the potential reaches its minimum.
- r_c: represents the cutoff radius.
- m, n: represents exponents controlling repulsive (m) and attractive (n) strength, usually m = 12, n = 6.
- k: represents the force constant derived to ensure the correct potential depth at $r = r_{min}$.

This potential is not soft potential.

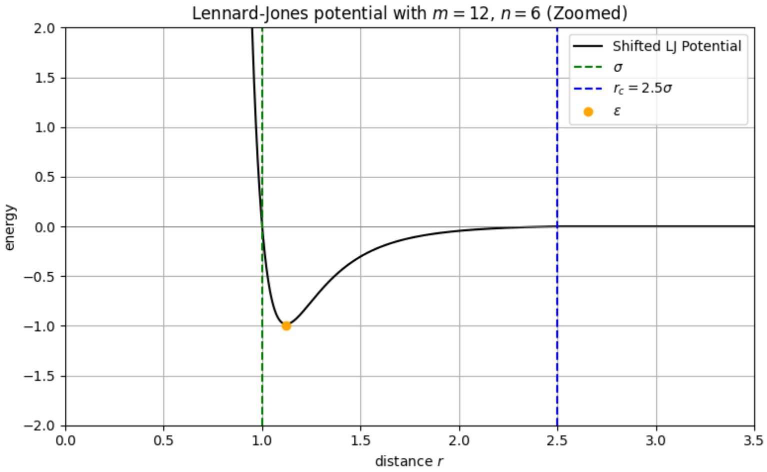


Figure 2.7. Zoomed-in view of a shifted Lennard-Jones potential (m = 12, n = 6)

This figure shows a shifted Lennard-Jones potential, highlighting its minimum energy (ε), interaction range (σ), and cutoff distance ($r_c = 2.5\sigma$). The potential captures both short-range repulsion and mid-range attraction but was ultimately not used in the final model.

d. Screened electrostatics (unused)

doi:10.6342/NTU202503815

```
system.potentials.add_screened_electrostatics( "A", "B", electrostatic_strength=-1., inverse_screening_depth=2., repulsion_strength=0.5, repulsion_distance=1., exponent=12, cutoff=2.5.)
```

$$V\left(||x_1 - x_2||_2\right) = V(r) = \begin{cases} C\dfrac{e^{-\kappa r}}{r} & + D\left(\dfrac{\sigma}{r}\right)^n, \ if\ r \le r_c \\ 0, \ otherwise \end{cases}$$

- $r = ||x_1 - x_2||_2$: represents the distance between two particles.
- $C \in R$: represents electrostatic interaction strength.
- $\kappa \in R$: represents inverse screening length, controls how quickly the electrostatic interaction decays.
- $D \ \in R$: represents strength of the core repulsion (units: energy).
- $\sigma \ \in R$: represents radius where the core repulsion term becomes significant.
- $n \ \in N$: represents exponent of the repulsive term(dimensionless), determines how sharply the repulsion increases.
- $r_c \in R$: represents cutoff radius beyond which the potential is zero.

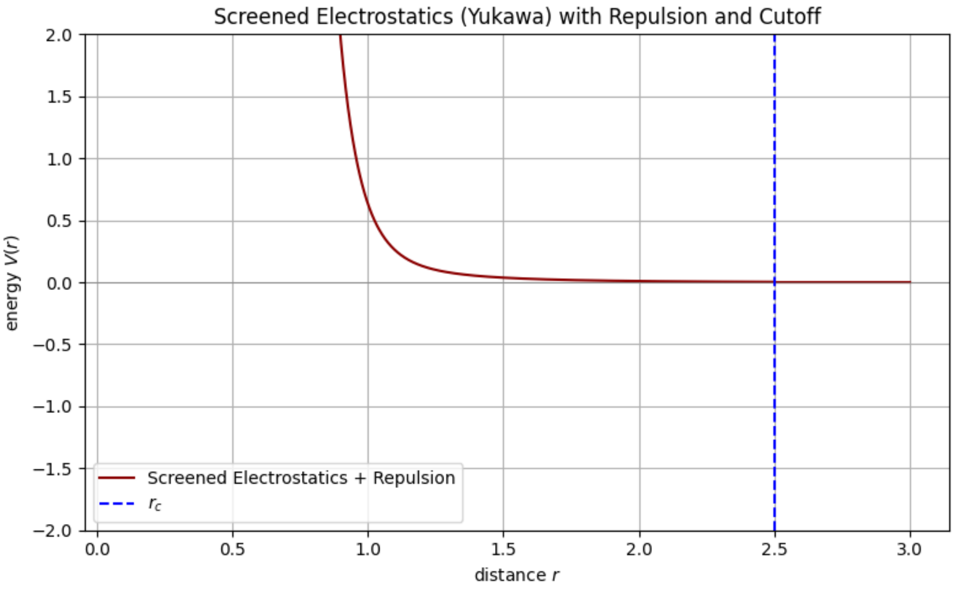


Figure 2.8. Screened electrostatic (Yukawa) potential with repulsion and cutoff = 2.5.

This plot shows a repulsive Yukawa potential truncated at a cutoff distance $r_c = 2.5$, representing short-range screened electrostatic interactions. Although tested, this potential was not incorporated into the final simulation due to its limited relevance to mitochondrial dynamics.

7. Topology

```python
system.topologies.add_type("My topology type")
system.add_topology_species("T", diffusion_constant=2.0)
```

This is also very import function of ReaDDy2 package, since when we try to simulate the molecular level or any bio system, we may encounter big and complex structures and networks. In code, we can define the structure or network type which relates to how interact when particles react to it, and after that we can add the particle type which allows us to react to the structure and network.

A. Topology Potential

Since ReaDDy2 is particle-base model, even the particles been connected and became the structure and network. The ReaDDy2 still allowing particles which in the structure have their own potential with others.

a. Harmonic bonds

```python
system.add_topology_species("T1", diffusion_constant=2.)
system.add_topology_species("T2", diffusion_constant=4.)
# define the bond with same particle type
system.topologies.configure_harmonic_bond("T1", "T1", force_constant=10., length=2.)
# define the bond with different particle type
system.topologies.configure_harmonic_bond( "T1", "T2", force_constant=10., length=2. )
```

This potential is very similar to the pair potential, to avoid the particle overlapping in the structure, since harmonic bond is also soft potential, the distance between the particles will slightly change from time to time. The formula given by:

$$V\left(\left|\left|x_1 - x_2\right|\right|_2\right) = V(r) = k\ (r - r_0)^2$$

- r_0: represents the preferred distance we set.
- k: represents the force constant

b. Harmonic angles

```
system.add_topology_species("T1", diffusion_constant=2.)
system.add_topology_species("T2", diffusion_constant=4.)
system.add_topology_species("T3", diffusion_constant=4.)
# This potential also be defined with same particle type
system.topologies.configure_harmonic_angle("T1", "T1", "T1", force_constant=1., equilibrium_angle=3.141)
# This potential also be defined with different particle type
system.topologies.configure_harmonic_angle("T1", "T2", "T3", force_constant=1., equilibrium_angle=3.141)
```

Harmonic angle is a potential which involves three particles instead of two. We can define a preferred angle degree and force between three connected particles in the structure and network; this function helps maintain structural stability and prevent unexpected distortions. This potential is also soft potential. The formula given by:

$$V(\theta_{i,j,k}) = k(\theta_{i,j,k} - \theta_0)^2$$

- θ_0: represent the preferred angle.
- k: represent the force constant.

c. Cosine dihedrals (unused)

```
system.add_topology_species("T1", diffusion_constant=2.)
system.add_topology_species("T2", diffusion_constant=4.)
system.add_topology_species("T3", diffusion_constant=4.)
system.add_topology_species("T4", diffusion_constant=4.)
system.topologies.configure_cosine_dihedral(
    "T1", "T2", "T3", "T4", force_constant=10, multiplicity=1., phi0=0.
)
```

proper dihedral angle ϕ is defined between four particles with positions x_i, x_j, x_k, and x_l. The associated potential energy is given by:

$$V(\phi) = k(1 + \cos(n\phi - \phi_0))$$

- ϕ: denotes dihedral angle spanned by force particles with positions x_i, x_j, x_k, x_l; measured between two planes formed by (x_i, x_j, x_k) and (x_j, x_k, x_l).
- $k \in R$: represents force constants.
- $n \in N_{>0}$: Multiplicity, the number of energy minima over a full 360° rotation of the dihedral.
- $\phi_0 \in [-\pi, \pi]$: represents phase offset angle (in radians), shifts the location of the minima.

The i_{th} minimum of potential occurs at:

$$\phi_i = \frac{1}{n}\left(\frac{\pi}{2} - \phi_0 + i\pi\right), \qquad i \epsilon Z$$

B. Topology Reaction

Since in realistic bio systems, any kind of network, structure, chain or larger complex is may not static, they can evolve reaction dynamically over time. In ReaDDy2, we can use topology reaction function to achieve:

1. Changing particle types within a topology, which can alter interaction forces.
2. Breaking and forming bonds, leading to topology separation or reorganization.
3. Attaching free particles to existing topologies.
4. Connecting different topologies by adding new edges.

To handle the interaction between the particle and the topologies, this function has two categories.

a. Structural reaction:

This function focuses on the adjustment of the structure and network when fission happens. For example, there are four particles that are connected, they should have three bonds and two angles. When fission happens on the right most of the particles, in ReaDDy2 package, it will automatically delete the bond and the angle of that

doi:10.6342/NTU202503815

particle to avoid unexpected error. It can be seen as a default function without extra coding. If you want to customize the structural reaction section, it can be divided into three steps:

1. The reaction function:

```
def no_op_reaction_function(topology):
    recipe = ReaDDy2.StructuralReactionRecipe(topology)
    return recipe
```

In this section, there are two important components that must be clarified which are topology and recipe. If you want to know information about the topology, the code will start from topology…etc. When we start collecting information from topology. There are two basic datasets we need to build:

- edges = topology.get_graph().get_edges(): get the all edge from the topology.
- vertices = topology.get_graph().get_vertices(): get the all vertex from the topology.

We can use these two datasets to get the specific particle and then type this line of code to preserve it: pix1 = e[0].get().particle_index. Once we have index of particle, we finally get more information such as:

- topology.position_of_vertex(vertices[pix1]): get the position of particle with specific index ([x, y, z]).
- topology.particle_type_of_vertex(vertice[pix1]): get the type of particle with specific index.
- topology.particle_id_of_vertex(vertice[pix1]): get the unique id of particle with specific index.

Once we have this information, we can combine it to get the particle or edge we want and then use recipe code function to change its property such as:

- Recipe.change_particle_type(vertices[pix1]), type): change the particle with specific index to the type you choose.
- Recipe.add_edge(vertice[pix1], vertice[pix2]): add edge between two different particles with specific index if they didn’t have edge previously.

doi:10.6342/NTU202503815

- Recipe.remove_edge(vertice[pix1], vertice[pix2]): remove edge between two different particles with specific index.
- Recipe.remove.edge(edge):same function as the last one but choose the specific edge instead of the particle.
- Recipe.separate_vertex(vertice[pix1]): isolate the particle from the topology i.e. remove all edge from the particle. It must be careful when using this function. It may let the topology fall apart if the particle is in the critical position.
- Recipe.change_topology_type(type): change the entire topology by the type you choose.
- Recipe.append_particle(list_of_neighbor_vertices, particle type, position): implant the particle to the topology by given position, particle type, the list of the particles which try to connect and become one node of the topology.

2. The rate function

```
def my_rate_function(topology):
    n = len(topology.get_graph().get_vertices())
    if n > 3:
      return .5 * n
    else:
      return 20.
```

This section decides how fast the speed of the reaction section happens following the probability formula:

$$p = 1 - e^{-\lambda\tau}$$

- $\lambda \in R_{\geq 0}$: represent the rection rate.
- τ: time step.

This section operation is very similar to the rection function section, we extract the information from the topology (edges or vertices) to get the rate we want. Example from above, we can get the total length of edges from the whole topology, and we set the rate base on that.

3. Adding a structural reaction

```
system.topologies.add_structural_reaction(
    name="my_structural_reaction",
    topology_type="TType",
    reaction_function=no_op_reaction_function,
    rate_function=my_rate_function,
    raise_if_invalid=True, expect_connected=False
)
```

Once we finish the reaction and rate section , finally we can import this structural reaction into simulation with extra information: what name is this reaction, what topology type will happen this reaction, when reaction is invalid will raise an error or just skip it , and after reaction happen will let the topology connected as the same or fall apart into two or more independent topology.

b. Spatial reaction

```
system.topologies.add_spatial_reaction('TT-Fusion: T1(p1)+T2(p2)
-> T3(p3--p4)', rate=1., radius=1.)
```

Spatial reactions occur when particles are near each other and depend on both particles and topology types. There are two main types:

- Fusion reactions: which form a bond between particles.
- Enzymatic reactions: which alter particles or topology types without bonding.

Each spatial reaction is defined by:

- A rate constant: indicating how often it occurs per time step.
- A radius constant: defining the search area for potential reactants.

To deal with complicate situation when spatial reaction happen, ReaDDy2 package have multiple reaction type, let T_i represent topology types and P_i represents particles:

- TT-Fusion: T1(p1)+T2(p2) -> T3(p3--p4)

doi:10.6342/NTU202503815

Merges a topology of type T1 with another of type T2 by bonding p1 and p2, resulting in a new topology T3 with p3 and p4 as the bonded particle types.

- TT-Fusion-self: T1(p1)+T1(p2) -> T3(p3--p4) [self=true]
  Similar to the previous case but allows particles within the same topology of type T1 to fuse.

- TP-Fusion: T1(p1) +(p2) -> T2(p3--p4)
  A free particle of type p2 bonds with a particle of type p1 in topology T1, forming topology T2 with the bonded particles becoming p3 and p4.
- TT-Enzymatic: T1(p1)+T2(p2) -> T3(p3)+T4(p4)
  Alters particle and topology types without changing graph structure, potentially modifying interaction dynamics.
- TP-Enzymatic: T1(p1)+(p2) -> T2(p3)+(p4)

  Similar to TT-Enzymatic, but involves one topology and a single free particle.

2.2 Mitochondrial Dynamic Network Simulation Framework

To model the dynamics of mitochondrial networks observed in single-cell microscopy experiments, we developed a reaction-diffusion simulation framework using ReaDDy2, a particle-based simulation package. This framework captures the essential behaviors of mitochondrial structures by simulating particles with both spatial and structural interactions—allowing for movement, diffusion, and biochemical reactions over time. It is specifically designed to reflect the topological evolution of mitochondria, including fission and fusion events that alter the network's connectivity.

Our simulation integrates empirical imaging data with computational modeling to enable direct comparison between experimental observations and silico dynamics. The following workflow outlines the complete process from image acquisition to simulation and validation (fig 2.9.).

doi:10.6342/NTU202503815

doi:10.6342/NTU202503815

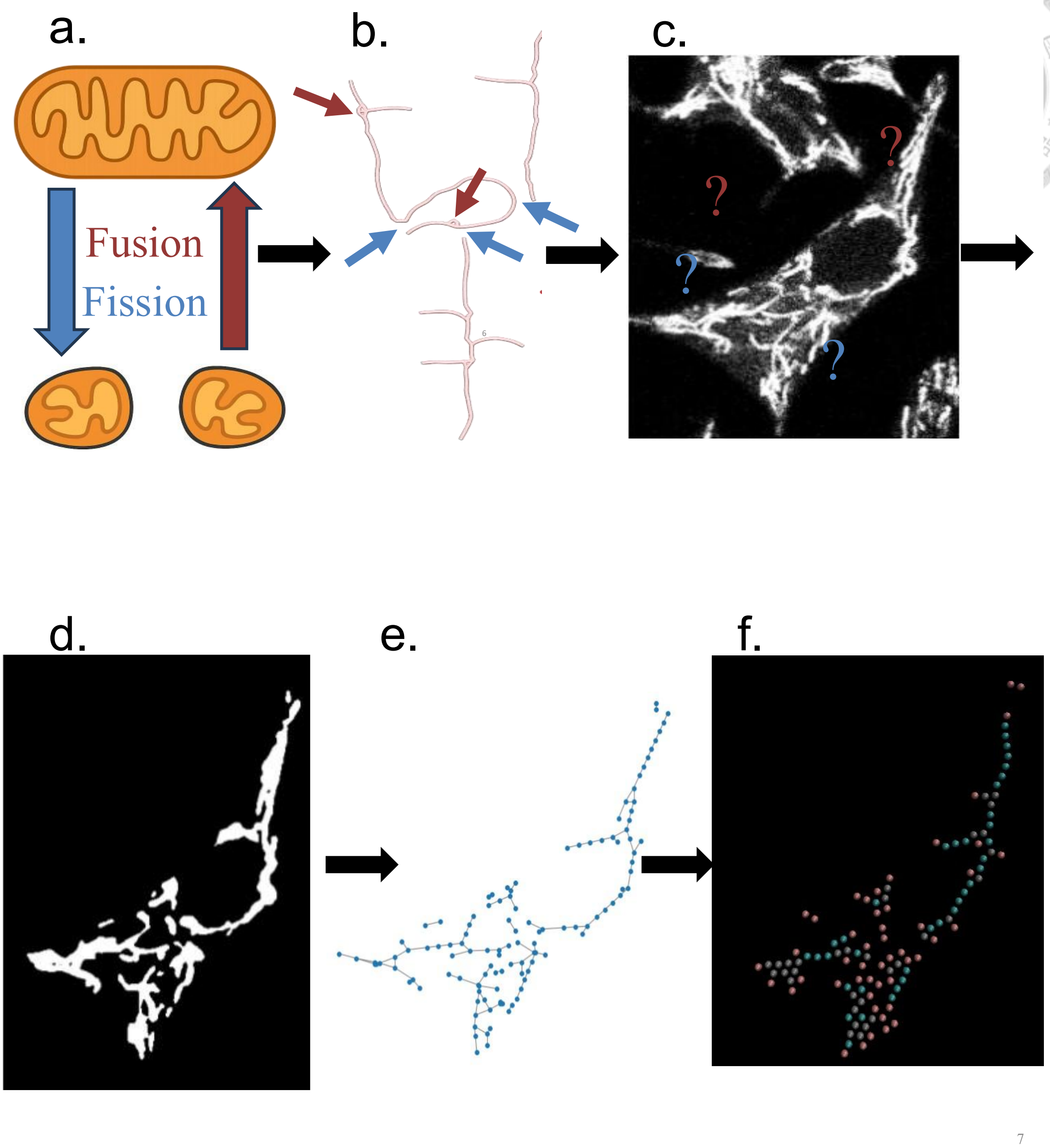


Figure 2.9. End-to-end workflow for translating experimental mitochondrial images into a particle-based ReaDDy simulation.

(a) Mitochondrial reaction schematic – Conceptual diagram summarizing the elementary structural reactions (tip-to-tip fusion, tip-to-side fusion, and fission) and their kinetic parameters. Arrows illustrate how the same tubular segment can merge, branch, or split, establishing the rule set later used in the simulation.

(b) Mitochondrial network-structure schematic – Abstract network view in which each tubule is represented as an edge and each junction/end-point as a node. Edge colors encode segment identity (e.g., parent branch vs. newly fused branch), clarifying how local reactions remodel global topology.

(c) Raw fluorescence image – A greyscale live-cell frame showing a dense mitochondrial reticulum. Individual fusion or fission events are difficult to discern because neighboring tubules overlap optically, yielding a visually continuous sheet of signal.

(d) Processed binary mask – The raw image is thresholder, denoised, and morphologically cleaned to isolate the mitochondrial foreground (white) from background (black). This representation preserves overall geometry while suppressing intensity fluctuations.

(e) Skeleton-derived graph – Using scikit-image's skeletonization and NetworkX, the binary mask is reduced to a one-pixel-wide skeleton and converted into a node-edge graph. Blue dots mark detected branch points and end-points; thin lines depict the extracted edges, providing precise coordinates for every segment.

(f) Simulation snapshot with particle typing – A ReaDDy simulation initialized from the graph in (e). Each skeleton node is replaced by a particle whose color encodes its current role (e.g., red = end-point, teal = internal tubule node). The scene illustrates how the experimental geometry is mapped onto a dynamic particle system where fusion and fission rules from (a) can operate quantitatively.

Together, panels (a–f) trace the pipeline from raw microscopy data to a fully specified, rule-based simulation ready for hypothesis testing and parameter sweeps.

2.3 Workflow Overview

1. Microscopy Image Extraction

High-resolution time-lapse fluorescence microscopy images of a single cell's mitochondrial network are processed using ImageJ. This step involves extracting the image stack representing sequential frames of mitochondrial morphology over time.

doi:10.6342/NTU202503815

2. Image Processing and Network Reconstruction

The image stack is processed using scikit-image, skan, and networkx to perform skeletonization and convert the mitochondrial structures into node-edge graphs.

- Nodes represent branching or terminal points of the mitochondrial network.
- Edges represent the mitochondrial tubes connecting these points.
- This graph-based representation is exported as CSV files storing node coordinates and edge connectivity.
- Additionally, the degree of each node (i.e., its connectivity) is recorded and logged across all frames, generating a reference dataset of real mitochondrial network dynamics.

3. Simulation Initialization and Relaxation with ReaDDy2

Using the exported node and edge CSV files, we initialize a static mitochondrial network in the ReaDDy2 simulation environment. Each node from the CSV is added as a particle with its corresponding spatial coordinate, while each edge defines a bonded interaction between particles, effectively reconstructing the network topology.

At this stage, no reactions (e.g., fusion or fission) are introduced. Instead, the system is allowed to relax dynamically under the influence of physical constraints such as bond lengths, repulsion forces, and diffusion. This ensures the network reaches a stable initial configuration—free of overlaps or unrealistic geometries—before enabling reactive events in later simulation phases.

4. Dynamic Reaction-Diffusion Simulation

Once the initial network is constructed and relaxed, we introduce reaction rules—including fission, fusion, fragmentation, and reassociation—into the ReaDDy2 simulation environment. These reactions allow the mitochondrial network to evolve over time through both spatial diffusion and topological changes.

doi:10.6342/NTU202503815

To analyze dynamic behavior, we employ two complementary simulation strategies:

a. Single-run Full Data Simulation:

In one script, the simulation is run once, but with a rich set of observables recorded over time, including:

- Node degree changes, capturing local structural evolution,
- Reaction event counts (e.g., number of fissions and fusions),
- Particle positions and trajectories for spatial analysis,
- Topology-level statistics, such as the number of connected components and network fragmentation.

This run generates a detailed time series that provides insights into how the network evolves and reorganizes during the simulation.

b. Multi-run Statistical Averaging

In the second script, the simulation is repeated 100 times, each starting from the same initial network.

Only the node degree distribution is recorded for each run, focusing on:

- Degree-1 (endpoints),
- Degree-2 (linear connections),
- Degree-3 (branch points).

By averaging the degree probabilities across all runs, we obtain a robust, statistically smoothed trajectory of network connectivity that helps mitigate random fluctuations and better reflect overall trends.

This dual approach ensures both deep temporal insight from a single run and quantitative reliability from multiple replicates, enabling cross-validation between simulated outcomes and experimentally observed network dynamics.

doi:10.6342/NTU202503815

5. Visualization and Post-Simulation Validation

Following the Single-run Full Data Simulation, an .xyz file is automatically generated and visualized using VMD (Visual Molecular Dynamics). This quick rendering allows for initial quality control, confirming that the simulation ran successfully and that the mitochondrial topology evolved without errors or crashes.

After verifying simulation integrity, we use the more detailed PDB and PSF files—also generated from this run—for high-resolution, frame-by-frame visualization in VMD. These snapshots provide an accurate visual timeline of the network's structural evolution.

The resulting image sequences can be:

- Compiled into videos for presentations or documentation,
- Used as synthetic benchmark data for testing and validating mitochondrial tracking algorithms in downstream image analysis pipelines.

This visualization step is critical for both qualitative inspection and quantitative image-processing validation, ensuring that the simulated mitochondrial dynamics align with experimental expectations.

6. Quantitative Comparison with Experimental Data

This analysis is based on the results of the Multi-run Statistical Averaging simulation, where the degree distribution of the mitochondrial network is tracked over repeated runs. From these simulations, we compute the average probabilities of node degrees (e.g., degree 1, 2, 3) at each time step.

These averaged simulation results are then directly compared to the degree distribution logs extracted from time-series microscopy images of real mitochondrial

doi:10.6342/NTU202503815

networks. This quantitative comparison serves as a validation step to evaluate how well the simulation captures the key features of biological mitochondrial dynamics.

By aligning trends between simulated and experimental degree distributions, we can:

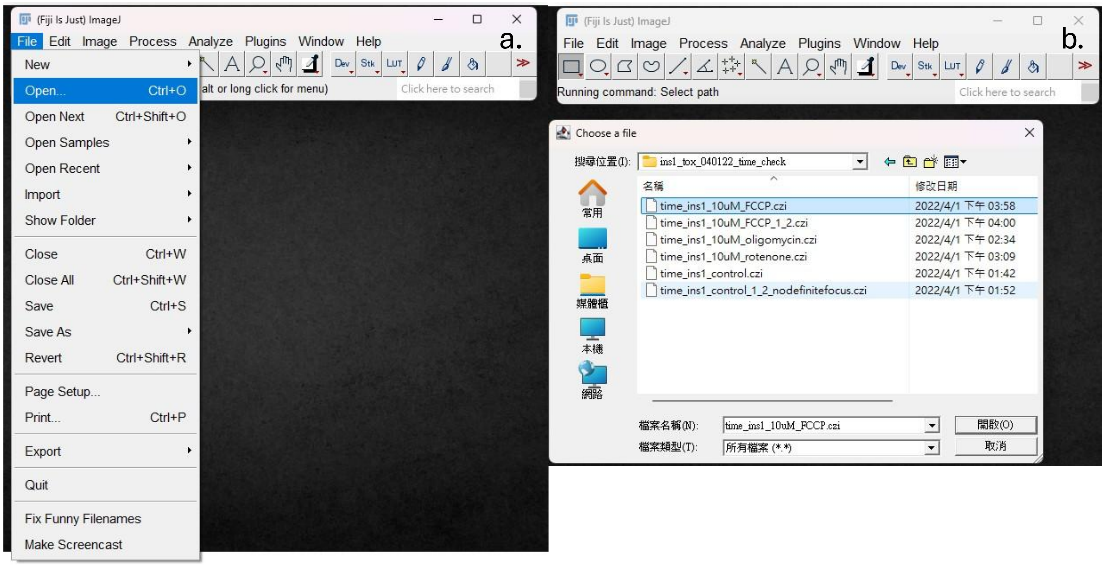
- Assess the realism of the modeled reactions and structural behavior,
- Identify discrepancies between model and biology,
- Tune simulation parameters (e.g., fusion/fission rates, diffusion constants) to better reflect observed cellular behavior.

This step bridges silico modeling and experimental observation, providing a data-driven foundation for iterative model refinement.

2.4 Microscopy Image Extraction

This section introduces how to extract image from real cell mitochondria network using ImageJ [39].

1. Open ImageJ and select czi file



a.
(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help
New
Open... Ctrl+O
Open Next Ctrl+Shift+O
Open Samples
Open Recent
Import
Show Folder
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As
Revert Ctrl+Shift+R
Page Setup...
Print... Ctrl+P
Export
Quit
Fix Funny Filenames
Make Screencast
alt or long click for menu)
Click here to search
b.
Running command: Select path
Choose a file
搜尋位置(I): ins1_tox_040122_time_check
常用
桌面
媒體櫃
本機
網路
名稱
修改日期
time_ins1_10uM_FCCP.czi 2022/4/1 下午 03:58
time_ins1_10uM_FCCP_1_2.czi 2022/4/1 下午 04:00
time_ins1_10uM_oligomycin.czi 2022/4/1 下午 02:34
time_ins1_10uM_rotenone.czi 2022/4/1 下午 03:09
time_ins1_control.czi 2022/4/1 下午 01:42
time_ins1_control_1_2_nodefinitefocus.czi 2022/4/1 下午 01:52
檔案名稱(N): time_ins1_10uM_FCCP.czi
檔案類型(T): 所有檔案 (*.*)
開啟(O)
取消

Figure 2.10. Loading microscopy data in ImageJ.

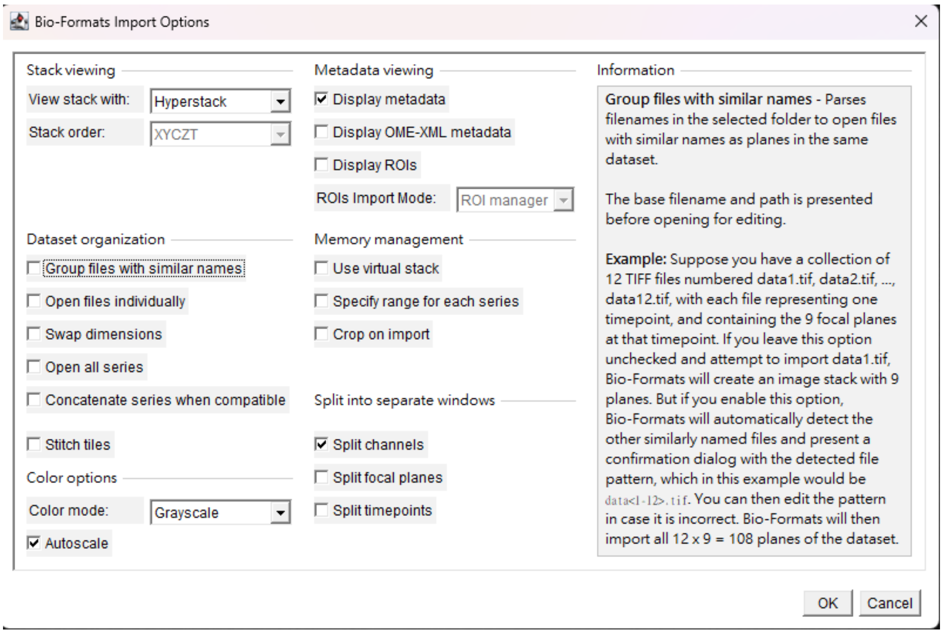
(a) The File > Open option is used to import raw microscopy images into FIJI.

(b) A .czi file is selected from the directory, representing time-series mitochondrial fluorescence data under various pharmacological treatments (e.g., FCCP, oligomycin, rotenone, control). These files serve as the input for downstream image processing and network extraction.

2. Check and adjust czi file properties

Once you select and open czi file there should come up with this window, there few functions that should be understand:

- Hyperstack: Enables multi-dimensional image viewing in a structured format.
  - Stack order XYCZT:
  - Specify the axis order of the data:
    - X = horizontal pixels, Y = vertical pixels, C = channel (e.g., red, green, blue, or different stains), Z = depth (z-stack), T = time (frames over time).
  - Color mode Grayscale: Opens the image in grayscale mode.
  - Display metadata (Check): his will open a separate window showing embedded metadata from the image file
  - Split channels(check): Separates each channel into individual windows.



Bio-Formats Import Options

Stack viewing
View stack with: Hyperstack
Stack order: XYCZT

Dataset organization
☐ Group files with similar names
☐ Open files individually
☐ Swap dimensions
☐ Open all series
☐ Concatenate series when compatible
☐ Stitch tiles

Color options
Color mode: Grayscale
☑ Autoscale

Metadata viewing
☑ Display metadata
☐ Display OME-XML metadata
☐ Display ROIs
ROIs Import Mode: ROI manager

Memory management
☐ Use virtual stack
☐ Specify range for each series
☐ Crop on import

Split into separate windows
☑ Split channels
☐ Split focal planes
☐ Split timepoints

Information
Group files with similar names - Parses filenames in the selected folder to open files with similar names as planes in the same dataset.

The base filename and path is presented before opening for editing.

Example: Suppose you have a collection of 12 TIFF files numbered data1.tif, data2.tif, ..., data12.tif, with each file representing one timepoint, and containing the 9 focal planes at that timepoint. If you leave this option unchecked and attempt to import data1.tif, Bio-Formats will create an image stack with 9 planes. But if you enable this option, Bio-Formats will automatically detect the other similarly named files and present a confirmation dialog with the detected file pattern, which in this example would be data<1-12>.tif. You can then edit the pattern in case it is incorrect. Bio-Formats will then import all 12 x 9 = 108 planes of the dataset.

Figure 2.11. Bio-Formats Import Options in ImageJ for .czi image loading.

The Bio-Formats importer provides flexible options for viewing and organizing multidimensional microscopy datasets. Key settings include View stack with: Hyperstack, Stack order: XYZCT, and enabling grayscale mode. Options such as Group files with similar names, Split channels, and Split timepoints ensure proper segmentation of image series for subsequent analysis. Metadata display and autoscaling are enabled to facilitate standardized preprocessing.

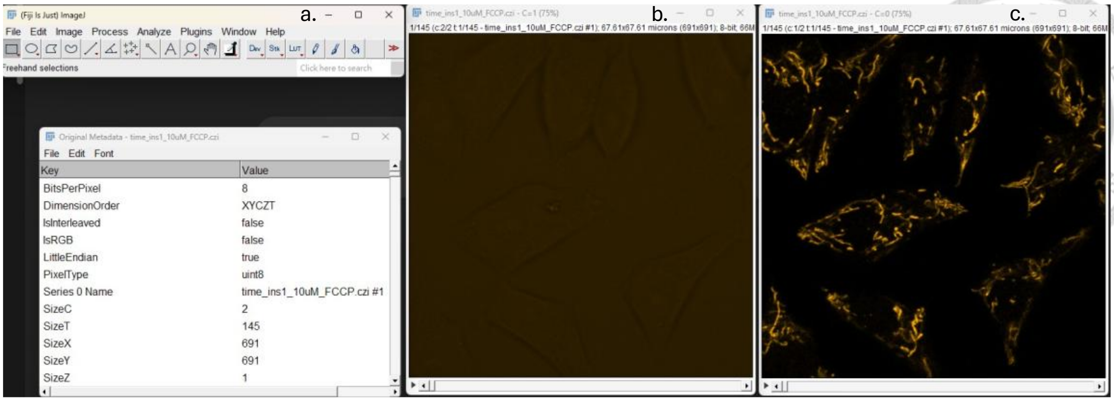
3. Resulting Windows After Import.

Once press the ok button. In this czi file, there should came up with 4 windows:

- ImageJ control panel: Main toolbar for navigating and analyzing images.
- Metadata panel: Displays detailed image metadata (e.g., dimensions, pixel size, number of channels, bit depth).
- Channel 1 (cell): first image window showing the cell structure (e.g., phase contrast or cytoplasm stain).
- Channel 2 (mitochondria network structure): second image window displaying mitochondria-specific fluorescence, revealing dynamic network structures.

Since the image windows don’t display useful details initially and need adjustment for better visibility.

doi:10.6342/NTU202503815



a.
(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help
Dev Stk LUT
Freehand selections
Click here to search
Original Metadata - time_ins1_10uM_FCCP.czi
File Edit Font
b.
time_ins1_10uM_FCCP.czi - C=1 (75%)
c.
time_ins1_10uM_FCCP.czi - C=0 (75%)

Key	Value
BitsPerPixel	8
DimensionOrder	XYCZT
IsInterleaved	false
IsRGB	false
LittleEndian	true
PixelType	uint8
Series 0 Name	time_ins1_10uM_FCCP.czi #1
SizeC	2
SizeT	145
SizeX	691
SizeY	691
SizeZ	1

1/145 (c:2/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 67.61x67.61 microns (691x691); 8-bit; 66M

1/145 (c:1/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 67.61x67.61 microns (691x691); 8-bit; 66M

Figure 2.12. Resulting windows after importing a .czi microscopy file in ImageJ.

(a) The ImageJ control panel and metadata viewer provide information of image (e.g., dimension order XYZCT, channel count, and bit depth).

(b) Channel 1 shows cell morphology.

(c) Channel 2 displays mitochondria network structures.

Brightness/contrast adjustment is typically required to enhance visibility upon initial loading.

4. image adjustment: Channel 1

a. please press to the channel 1 window first, to make sure to process correct target, then we adjust its type to 8 bits.

doi:10.6342/NTU202503815

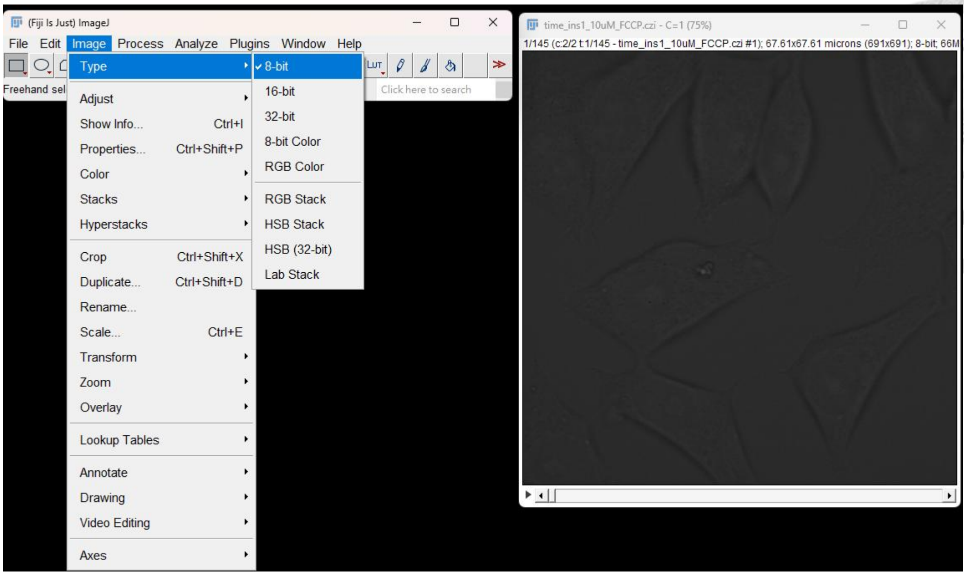


Figure 2.13. Bit-depth adjustment for Channel 1 in ImageJ.

To enhance image compatibility and visualization, the active window corresponding to Channel 1 (cell morphology) is selected. Under the Image > Type menu, the image is converted to 8-bit format. This standardization step facilitates consistent processing and downstream analysis.

b. And we find the contrast function; after press it, there should come up with another window, then press its auto button to do automatically adjust and press it apply button to finish the contrast adjustment.

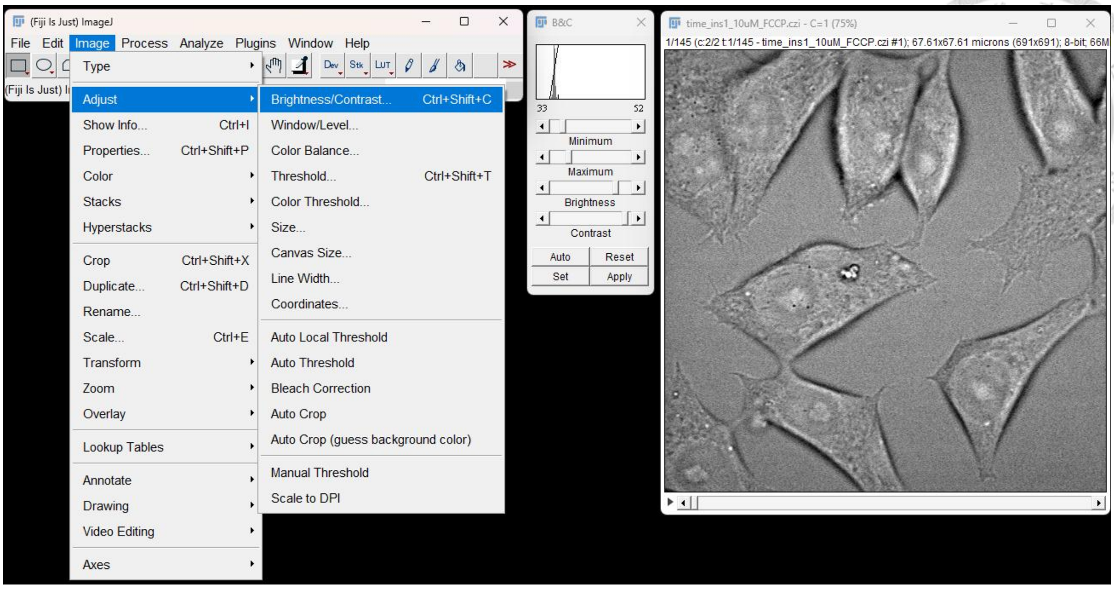


Figure 2.14. Automatic brightness and contrast adjustment for Channel 1 in ImageJ.

With Channel 1 selected, the Image > Adjust > Brightness/Contrast function is used to enhance image visibility. The Brightness/Contrast control window is opened, and the Auto button is applied to automatically optimize display settings. The Apply button finalizes the adjustment, improving contrast for clearer visualization of cellular features.

5. image adjustment channel 2

Like channel 1, we just need change its type to 8 bits

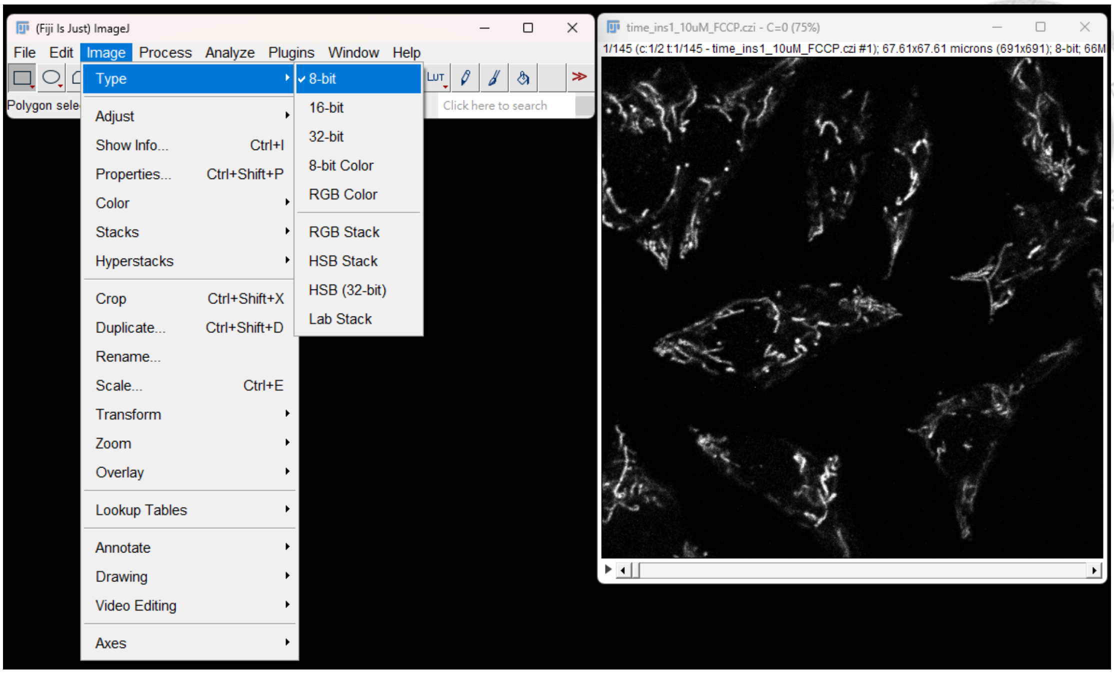


Figure 2.15. Bit-depth adjustment for Channel 2 in ImageJ.

Like Channel 1, the mitochondrial fluorescence image (Channel 2) is selected and converted to 8-bit format via Image > Type > 8-bit in ImageJ. This step ensures compatibility for subsequent processing and analysis, such as thresholding and segmentation.

6.observe windows to find mitochondria network of single cells.

According to the windows, we can cross-comparison to find the network we want.

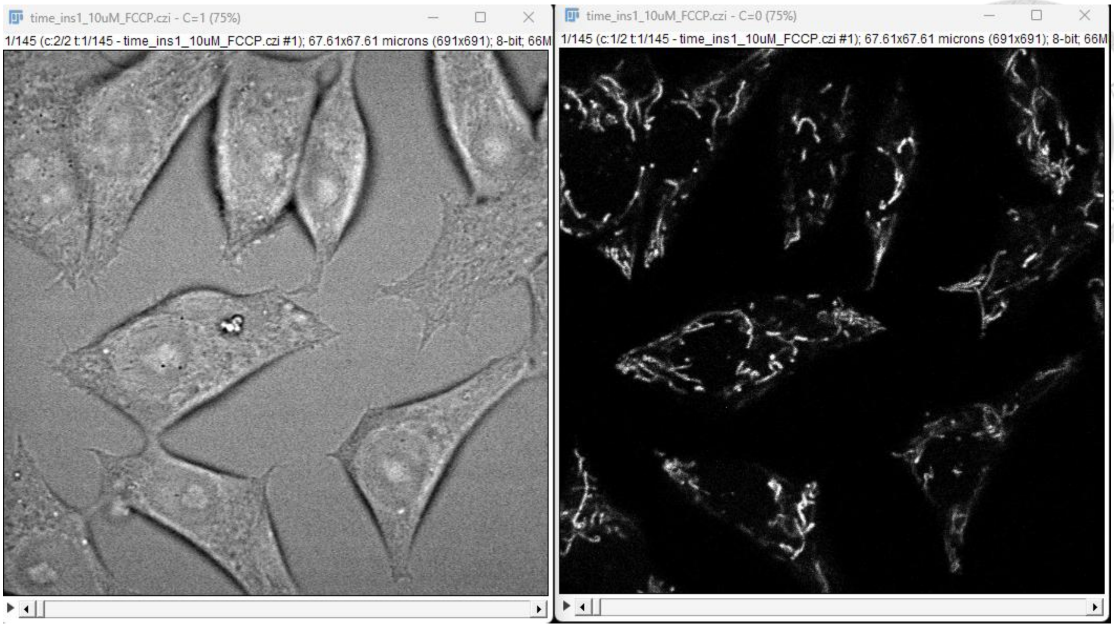

time_ins1_10uM_FCCP.czi - C=1 (75%)
1/145 (c:2/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 67.61x67.61 microns (691x691); 8-bit; 66M
time_ins1_10uM_FCCP.czi - C=0 (75%)
1/145 (c:1/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 67.61x67.61 microns (691x691); 8-bit; 66M

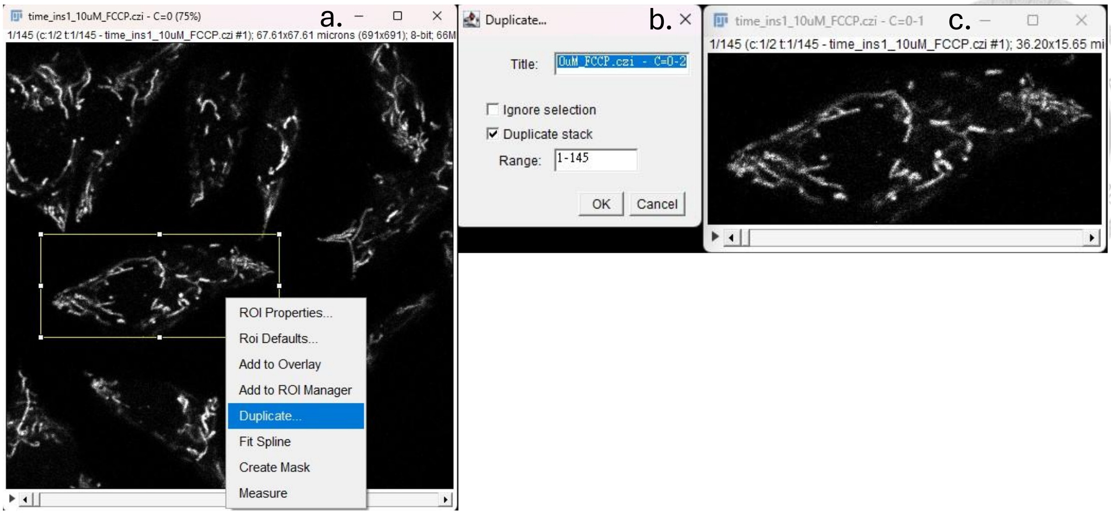
Figure 2.16. Cross-comparison of cell morphology and mitochondrial fluorescence in ImageJ.

Two synchronized windows display Channel 1 (left, cell morphology) and Channel 2 (right, mitochondrial network) from the same .czi image stack. By cross-referencing these views, single-cell mitochondrial networks can be accurately located and selected for further analysis.

7. Extract the network of single cells from window

Once we select the network, we can frame the network and click the mouse right button to find Duplicate function, and then there should be come up with another window, we check duplicate stack to create image with time series, finally we press ok button another window shows up to display the network of single cells.

We can scroll time slide at bottom of windows to make sure the branch of network isn’t out of bound.


time_ins1_10uM_FCCP.czi - C=0 (75%)
a.
1/145 (c:1/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 67.61x67.61 microns (691x691); 8-bit; 66M
ROI Properties...
Roi Defaults...
Add to Overlay
Add to ROI Manager
Duplicate...
Fit Spline
Create Mask
Measure
Duplicate...
b.
OK
Cancel
time_ins1_10uM_FCCP.czi - C=0-1
c.
1/145 (c:1/2 t:1/145 - time_ins1_10uM_FCCP.czi #1); 36.20x15.65 mi

Title: 0uM_FCCP.czi - C=0-2

□ Ignore selection

☑ Duplicate stack

Range: 1-145

Figure 2.17. Extraction of a single-cell mitochondrial network using the Duplicate function in ImageJ.

(a) A mitochondrial network region is selected by drawing a rectangular region of single cell in Channel 2.

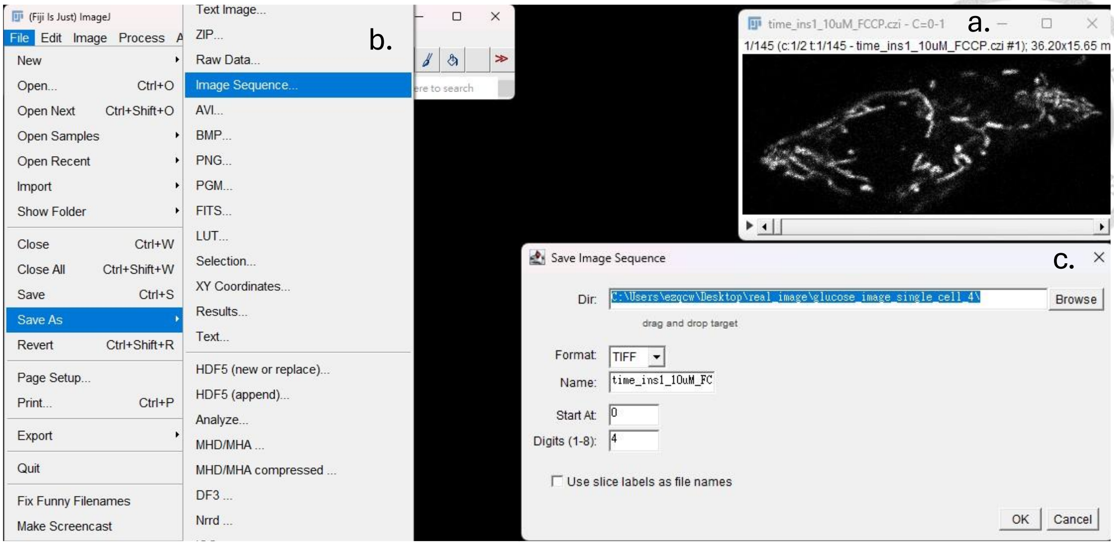
(b) The Duplicate option is accessed via right-click, and the Duplicate stack box is checked in the dialog to preserve all time frames in the stack.

(c) A new window is generated showing the extracted time series of the selected mitochondrial network.

8. Save network of single cell as multiple frame tiff

Press the window of the single cell network and find save as => image sequence. After pressing it, there should be came up with another window, you choose the folder you want to save multiple tiff file, finally press ok to finish this section.

doi:10.6342/NTU202503815



Save Image Sequence

Dir: C:\Users\ezqcw\Desktop\real_image\glucose_image_single_cell_4\ Browse

drag and drop target

Format: TIFF

Name: time_ins1_10uM_FC

Start At: 0

Digits (1-8): 4

□ Use slice labels as file names

OK Cancel

Figure 2.18. Exporting the single-cell mitochondrial network as a multi-frame TIFF sequence in ImageJ.

(a) The extracted single-cell mitochondrial network stack is selected.

(b) Under File > Save As > Image Sequence, the user initiates the export process.

(c) In the dialog window, the output directory, filename prefix, format (TIFF), and frame index settings are specified. Clicking OK saves each time frame as an individual TIFF file, enabling compatibility with downstream image processing tools.

2.5 Image Processing and Network Reconstruction

In this section, the image data imported using ImageJ is processed to extract the underlying mitochondrial network structure. Using tools such as scikit-image, skan, and networkx, the processed images are skeletonized and converted into a node-edge graph representation.

- Nodes represent key structural points such as branch points or endpoints.
- Edges represent the mitochondrial tubules connecting these nodes.

The resulting graph is exported as two CSV files: one containing node coordinates and the other defining edge connectivity. These files serve as the input to initialize the mitochondrial network in the ReaDDy2 simulation environment (fig 2.19.).

doi:10.6342/NTU202503815

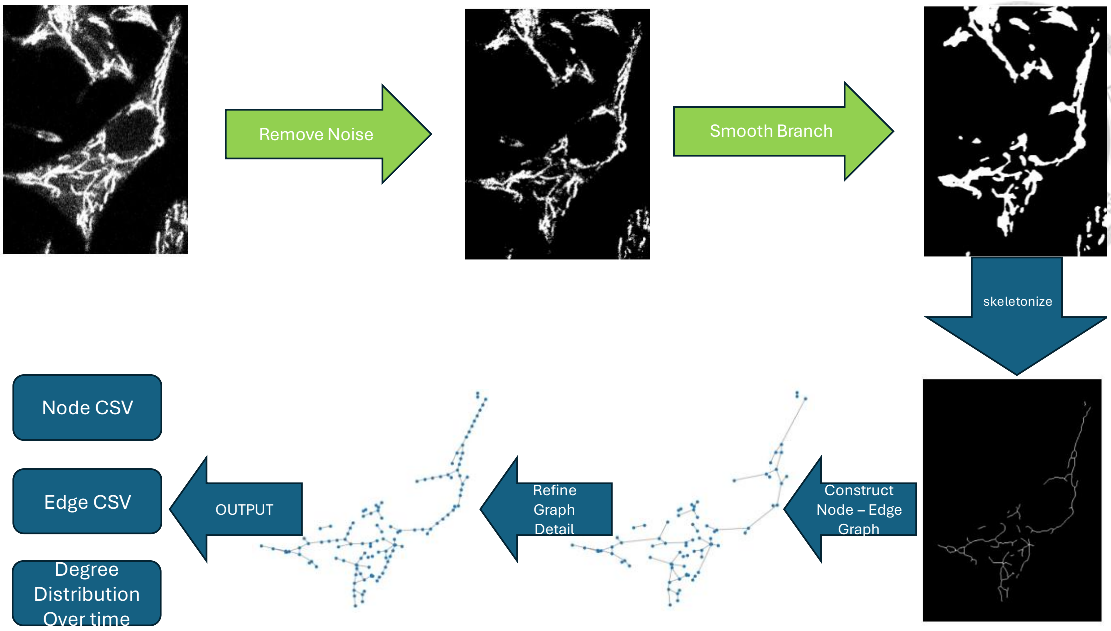


Figure 2.19. Image Processing and Mitochondrial Network Reconstruction Pipeline.

This panel zooms into the middle stage of the end-to-end pipeline introduced in Fig. 2.9, detailing how the experimental image in Fig 2.9 (c) is transformed into the graph that sees the simulation shown in Fig. 2.9(f).

1. Noise removal – The raw frame is first denoised (green arrow, “Remove Noise”) using median filtering and adaptive thresholding to suppress photon shot noise and background speckle while preserving true mitochondrial signal.

2. Branch smoothing – A morphological opening/closing sequence (“Smooth Branch”) eliminates jagged edges and fills sub-pixel gaps, yielding cleaner, contiguous tubules.

3. Skeletonization – The refined binary mask is reduced to a one-pixel-wide skeleton (blue downward arrow), preserving topology while stripping away thickness information.

4. Graph construction – Skeleton nodes (branch points and termini) and edges (tubular segments) are identified with scikit-image’s skan and imported into NetworkX (“Construct Node-Edge graph”).

doi:10.6342/NTU202503815

5. Graph refinement – Short stubs and artifactual fragments are pruned, and missing links are bridged to restore continuity (“Refine Graph Detail”), producing a biologically plausible topology.

6.Data export & diagnostics – The final graph is exported as Node CSV and Edge CSV files, accompanied by summary metrics such as the degree-distribution-over-time trace. These files are the direct inputs that initialise particle positions and connectivity in the simulation stage that follows (Fig 2.9. (f)).

2.5.1. filter and smooth image

a. Remove Background Noise to Isolate Mitochondrial Structures

In the initial step of image processing, we aim to reduce background noise and highlight the true mitochondrial network (fig 2.20.). This is crucial for accurate skeletonization and graph reconstruction in later steps.

```
from skimage.filters import threshold_otsu
# Step 1: Compute the optimal threshold using Otsu's method
thresh = threshold_otsu(image)
# Step 2: adjust the threshold slightly to remove faint noise
adjusted_thresh = thresh * 0.6
# Step 3: Create a binary image by applying the adjusted threshold
filter_image = image > adjusted_thresh
```

doi:10.6342/NTU202503815

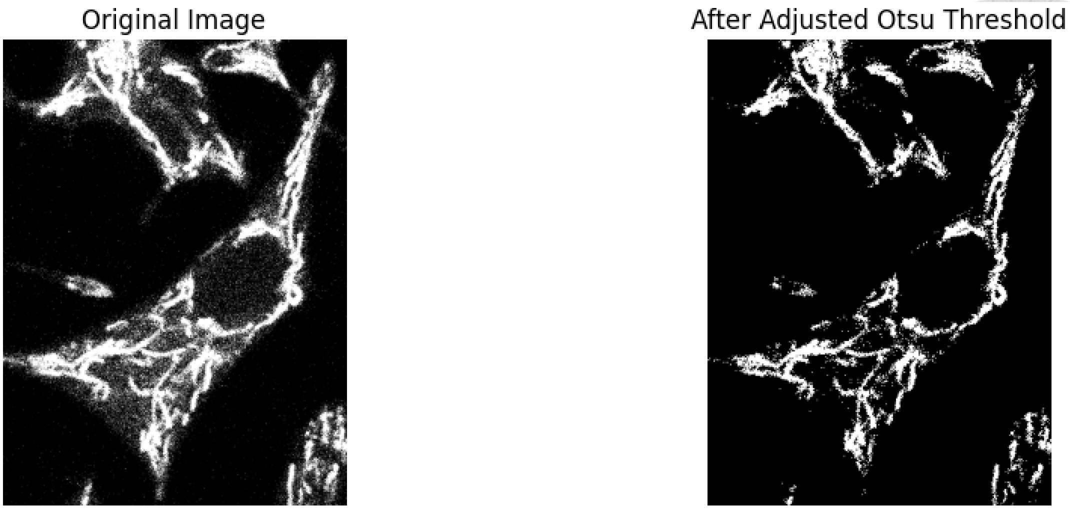


Figure 2.20. Background removal and enhancement of mitochondrial structures using Otsu thresholding.

The left panel shows the original mitochondrial fluorescence image, containing both signal and background noise. The right panel displays the result after applying and manually adjusting Otsu's thresholding method to isolate high-intensity mitochondrial structures. This preprocessing step enhances contrast and prepares the image for accurate binarization, skeletonization, and network extraction.

b. Refine Mitochondrial Structure to Avoid Noise Artifacts in Skeletonization

After filtering out background noise, the next step is to smooth the binary image to eliminate small irregularities or jagged edges—often referred to as "hairy" structures(fig 2.21.). These artifacts can negatively impact the skeletonization process, leading to false or fragmented branches in the final network graph.

```
from skan.pre import threshold
# Smooth the image
bin_image = threshold(
    filter_image,
    sigma=2,   # Controls the strength of Gaussian smoothing
    radius=0   # No local neighborhood dilation)
```

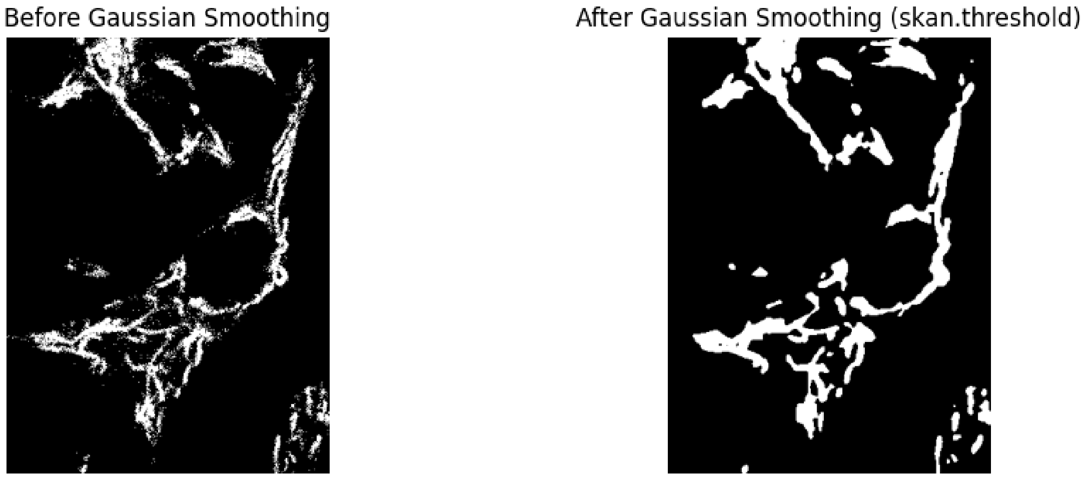

Before Gaussian Smoothing
After Gaussian Smoothing (skan.threshold)

Figure 2.21. Refinement of binarized mitochondrial structures using Gaussian smoothing.

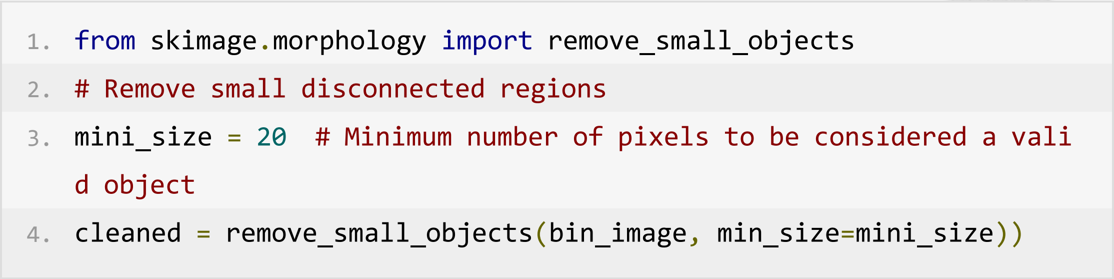
The left panel shows the binarized image after thresholding, still containing fine-grained noise and irregular contours. The right panel presents the result after applying Gaussian smoothing via the skan.threshold module, which reduces spurious edges and sharp protrusions. This refinement step improves the accuracy of subsequent skeletonization by minimizing artifacts that could lead to false branches or misidentified connections in the network.

c. Eliminate Noise and Isolated Fragments to Ensure Structural Consistency

After smoothing the binary image, we perform a small object removal step to eliminate minor artifacts or isolated blobs that are not part of the actual mitochondrial network(fig 2.22.). These small components may arise from background noise or disconnected pixels and can lead to the creation of isolated nodes when constructing the node-edge graph.

In our simulation, isolated nodes violate the assumption of a connected network and could distort the dynamics of mitochondrial behavior. To prevent this, we filter out small components based on their area.

doi:10.6342/NTU202503815



```python
from skimage.morphology import remove_small_objects
# Remove small disconnected regions
mini_size = 20  # Minimum number of pixels to be considered a valid object
cleaned = remove_small_objects(bin_image, min_size=mini_size))
```

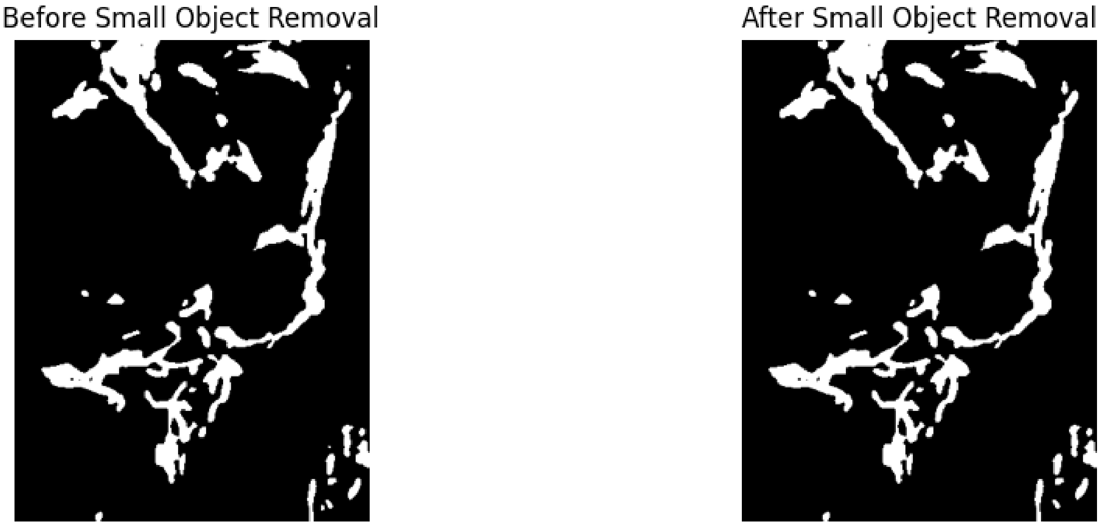


Figure 2.22. Removal of small, disconnected objects to ensure mitochondrial network integrity.

The left panel shows the binary image after smoothing, which may still contain small noise-induced fragments. The right panel displays the result after applying an area-based small object removal filter. This step eliminates minor isolated blobs that could otherwise generate erroneous nodes in the skeletonized graph. Ensuring a structurally consistent and connected network is essential for accurate simulation of mitochondrial dynamics.

2.5.2 Skeletonize the image

After filtering, smoothing, and removing unwanted structures, the cleaned binary image is now ready for skeletonization(fig 2.23.). This step reduces the thick, blob-like mitochondrial shapes into their central axes—thin, one-pixel-wide lines—while preserving the overall topology and connectivity.

This is essential for extracting the graph structure of the mitochondrial network (nodes and edges) in a form suitable for simulation.

```python
from skimage.morphology import skeletonize
# Perform skeletonization
skeleton = skeletonize(final_mask)
```

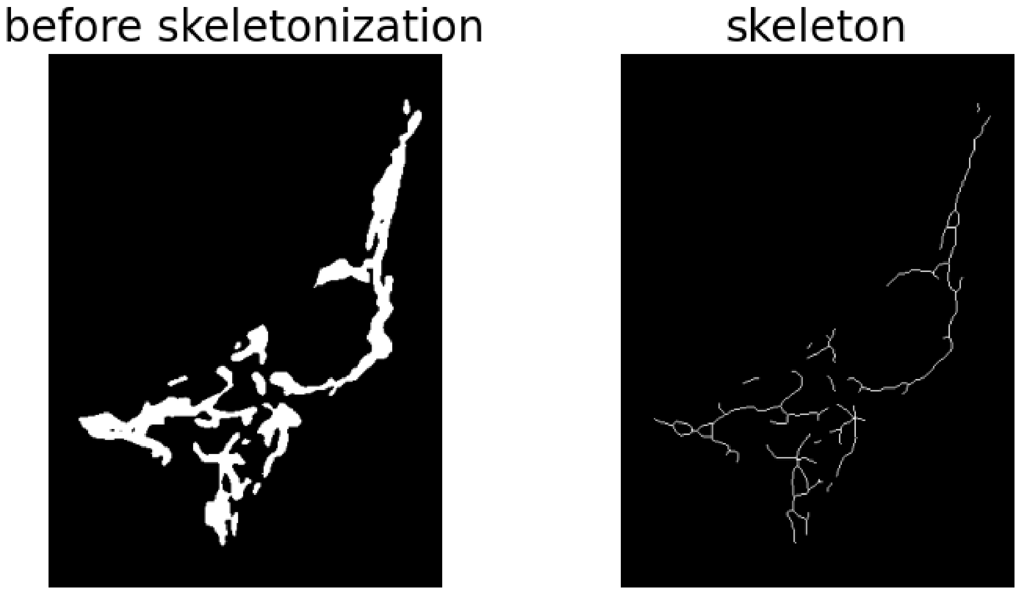


Figure 2.23. Skeletonization of the binarized mitochondrial network.

The left panel shows the pre-processed binary image representing the mitochondrial structures of a single cell. The right panel displays the result of skeletonization, where thick regions are reduced to one-pixel-wide medial axes. This process preserves the topology and connectivity of the network, enabling downstream conversion into a graph structure of nodes and edges for quantitative simulation.

2.5.3 Construct Node-Edge Graph from Skeleton

After skeletonizing the image into 1-pixel-wide paths, the next step is to transform the skeleton into a graph structure (fig 2.24.). This allows the mitochondrial network to be represented as a set of nodes (branch points and endpoints) and edges (connections between nodes), which is ideal for quantitative analysis and simulation in ReaDDy2.

We use the skan package to analyze the skeleton and extract its structural geometry and then use networkX to build the graph.

```python
from skan import Skeleton, summarize
import networkx as nx
# Step 1: Create a Skeleton object and summarize it
skeleton_obj = Skeleton(skeleton)
skeleton_summary = summarize(skeleton_obj, find_main_branch=True)
# Step 2: Initialize a NetworkX graph
graph = nx.Graph()
# Step 3: Add nodes with their coordinates
for node_id in np.unique(skeleton_summary[['node-id-src', 'node-id-dst']]):
    rows = skeleton_summary[(skeleton_summary['node-id-src'] == node_id) |
                            (skeleton_summary['node-id-dst'] == node_id)]
    for _, row in rows.iterrows():
        if row['node-id-src'] == node_id:
            coord = (row['image-coord-src-1'], row['image-coord-src-0'])
        else:
            coord = (row['image-coord-dst-1'], row['image-coord-dst-0'])
        break
    graph.add_node(node_id, coord=coord)
# Step 4: Add edges with their branch distances
for _, row in skeleton_summary.iterrows():
    src = row['node-id-src']
    dst = row['node-id-dst']
    distance = row['branch-distance']
    graph.add_edge(src, dst, weight=distance)
# Step 5: Extract node coordinates for visualization
```

```
node_positions = {node: data['coord'] for node, data in graph.nodes(data=True)}
```

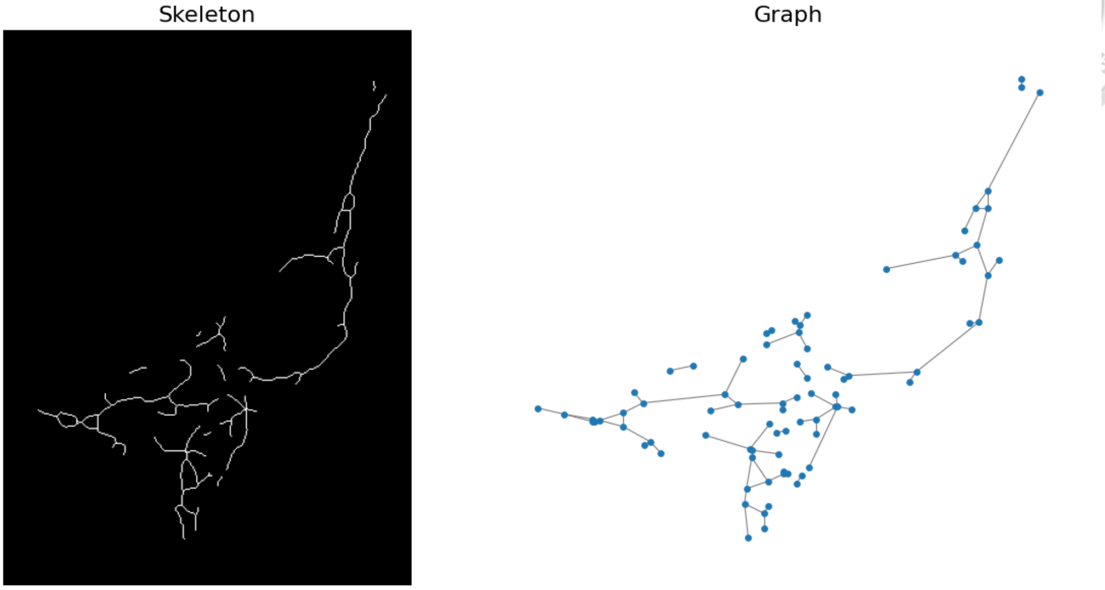


Figure 2.24. Conversion of the mitochondrial skeleton into a node-edge graph.

The left panel shows the one-pixel-wide skeletonized representation of the mitochondrial network. The right panel depicts the corresponding graph structure, where nodes represent endpoints and branch points, and edges represent linear connections between them. This transformation is achieved using the skan library for skeleton analysis and networkx for graph construction, enabling quantitative simulation and topological analysis in ReaDDy2.

2.5.4 Refine the Graph by Adding Intermediate Nodes

The initial node-edge graph only includes skeleton endpoints and branching points, which results in a sparse network. This limited resolution can restrict the accuracy of downstream simulations—particularly when modeling fission and fusion events, which may occur anywhere along a mitochondrial tubule.

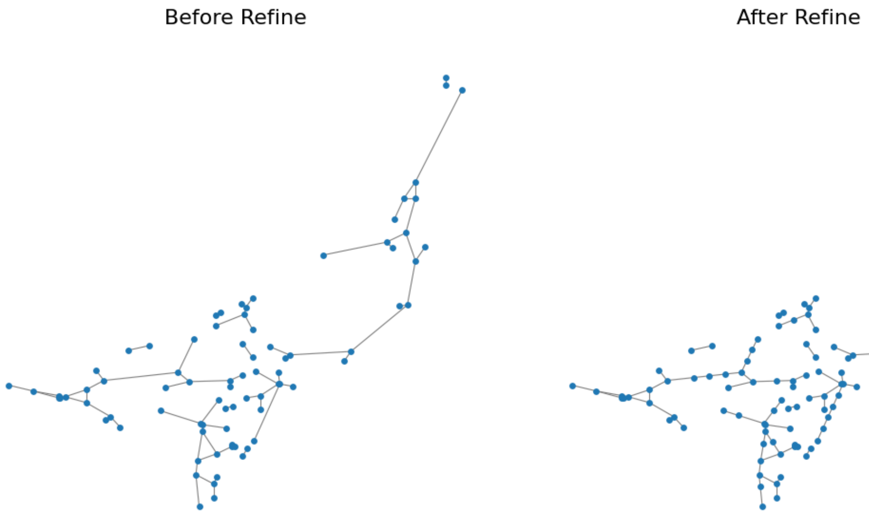
To better mimic the continuous nature of real mitochondrial structures, we refine the graph by inserting additional nodes along each edge at regular intervals (e.g., every 10 pixels) (fig 2.25.). This makes the network denser and more realistic for reaction-diffusion simulations.

```python
import networkx as nx

# Step 1: Create a copy of the original graph
modified_graph = nx.Graph()
modified_graph.add_nodes_from(graph.nodes(data=True))

# Step 2: Define parameters
image_height = skeleton.shape[0]
new_node_id_gen = itertools.count(start=max(graph.nodes) + 1)  # Unique node IDs
bond_length = 10  # Desired spacing between intermediate nodes

# Step 3: Loop over edges and insert intermediate nodes
for u, v, data in graph.edges(data=True):
    x0, y0 = graph.nodes[u]['coord']
    x1, y1 = graph.nodes[v]['coord']
    dx, dy = x1 - x0, y1 - y0
    edge_length = math.hypot(dx, dy)
    num_segments = int(edge_length // bond_length)
    if num_segments == 0:
        modified_graph.add_edge(u, v, weight=edge_length)
        continue
    ux, uy = dx / edge_length, dy / edge_length
    prev_node = u
    for i in range(1, num_segments):
        new_x = x0 + ux * i * bond_length
        new_y = y0 + uy * i * bond_length
        new_id = next(new_node_id_gen)
        modified_graph.add_node(new_id, coord=(new_x, new_y))
        modified_graph.add_edge(prev_node, new_id, weight=bond_length)
        prev_node = new_id
    modified_graph.add_edge(prev_node, v, weight=bond_length)
```



Before Refine

After Refine

Figure 2.25. Refinement of the mitochondrial graph by inserting intermediate nodes.

The left panel shows the original node-edge graph derived from the skeleton, composed only of branch points and endpoints. The right panel illustrates the refined graph, where intermediate nodes are inserted along each edge at uniform spatial intervals (e.g., every 10 pixels). This refinement increases spatial resolution and structural fidelity, enabling more realistic simulation of mitochondrial fission, fusion, and diffusion processes in ReaDDy2.

2.5.5 Export Refined Graph as Node and Edge CSV Files

After refining the mitochondrial network graph by adding intermediate nodes, the final step in image-based preprocessing is to export the graph structure into CSV files (fig 2.26.). These files serve as input to reconstruct the network topology in the ReaDDy2 simulation environment.

We extract and save two components:

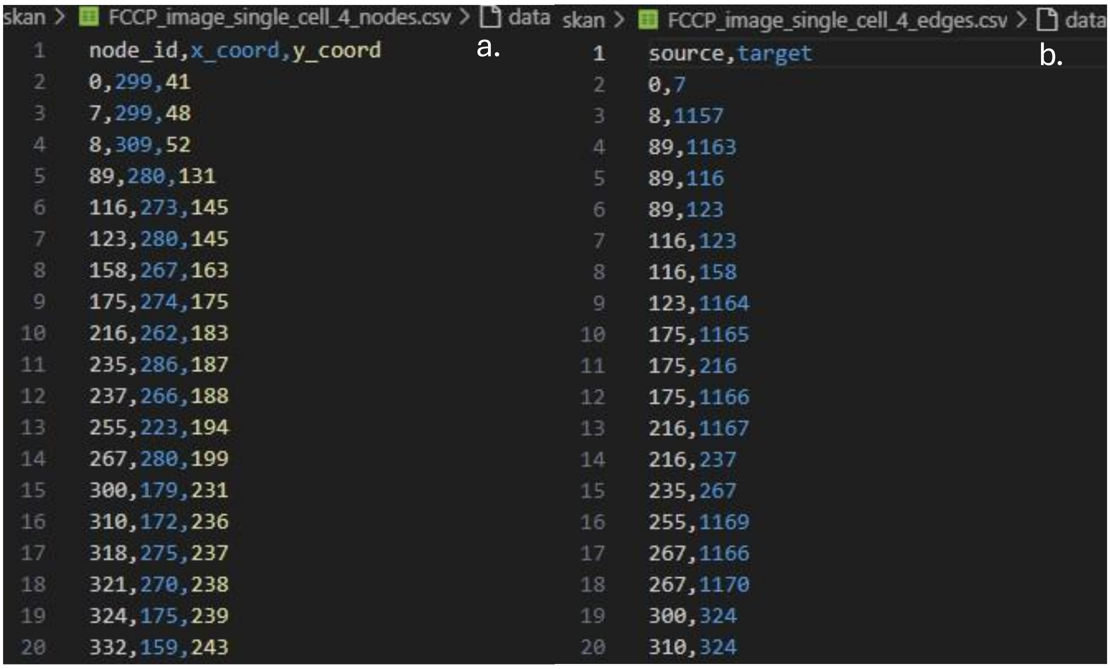
- A node CSV: Contains unique node IDs and their 2D coordinates.

```
import csv
```

```python
# Save node information
with open('FCCP_image_single_cell_4_nodes.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['node_id', 'x_coord', 'y_coord'])
    for node, data in modified_graph.nodes(data=True):
        x, y = data['coord']
        writer.writerow([node, x, y])
```

- An edge CSV: Describes the connectivity between node pairs (source–target).

```python
import csv
with open('FCCP_image_single_cell_4_edges.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['source', 'target'])
    for u, v, data in modified_graph.edges(data=True):
        writer.writerow([u, v])
```


skan > FCCP_image_single_cell_4_nodes.csv > data
a.
skan > FCCP_image_single_cell_4_edges.csv > data
b.

```
node_id,x_coord,y_coord
0,299,41
7,299,48
8,309,52
89,280,131
116,273,145
123,280,145
158,267,163
175,274,175
216,262,183
235,286,187
237,266,188
255,223,194
267,280,199
300,179,231
310,172,236
318,275,237
321,270,238
324,175,239
332,159,243
```

```
source,target
0,7
8,1157
89,1163
89,116
89,123
116,123
116,158
123,1164
175,1165
175,216
175,1166
216,1167
216,237
235,267
255,1169
267,1166
267,1170
300,324
310,324
```

Figure 2.26. Exported mitochondrial graph data as node and edge CSV files.

(a) The node file (_nodes.csv) lists each node's unique ID along with its 2D spatial coordinates.

(b) The edge file (_edges.csv) specifies the connectivity between nodes as source–target pairs.

These CSV files encode the refined mitochondrial network structure and serve as standardized input for initializing topology-based simulations in ReaDDy2.

2.5.6 Process Time-Series Images to Generate Degree Distribution Log

The previous steps process only a single image (one timepoint) of the mitochondrial network. However, to study temporal changes in network topology, we apply the same pipeline to each frame in a time-series image stack. For each frame, we compute the degree distribution of the graph and track how it evolves over time (fig 2.27.).

```python
# Initialize lists to store probabilities
degree_1_probs = []
degree_2_probs = []
degree_3_probs = []

# Open a log file to write the results
with open("FCCP_image_single_cell_4_degree_distrubution_log.txt", "w") as log_file:
    # Write header
    log_file.write("TimeStep,Degree1_Prob,Degree2_Prob,Degree3_Prob\n")

    # Iterate through each graph
    for t, graph in enumerate(modified_graphs):
        degrees = [deg for _, deg in graph.degree()]
        total_nodes = len(degrees)
        degree_counts = Counter(degrees)

        # Normalize counts to probabilities
```

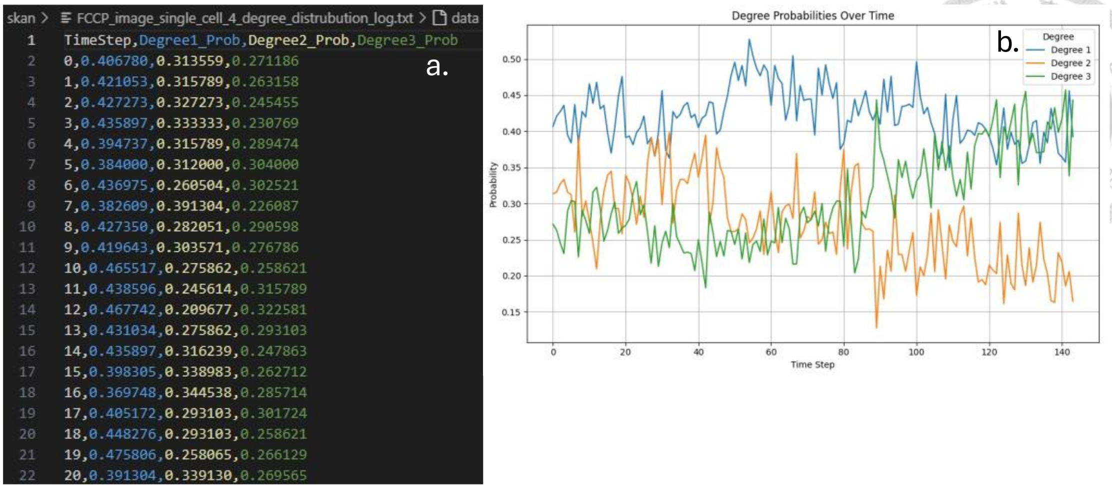
doi:10.6342/NTU202503815

```
        degree_1 = degree_counts.get(1, 0) / total_nodes
        degree_2 = degree_counts.get(2, 0) / total_nodes
        degree_3 = degree_counts.get(3, 0) / total_nodes

        degree_1_probs.append(degree_1)
        degree_2_probs.append(degree_2)
        degree_3_probs.append(degree_3)

        # Log the results
        log_file.write(f"{t},{degree_1:.6f},{degree_2:.6f},{degree_3:.6f}\n")

        # Optionally print
        print(f"Time {t}:")
        print(f"  1-degree nodes (endpoints): {degree_1}")
        print(f"  2-degree nodes (linear path points): {degree_2}")
        print(f"  3-degree nodes (branching points): {degree_3}\n")
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(degree_1_probs, label='Degree 1')
plt.plot(degree_2_probs, label='Degree 2')
plt.plot(degree_3_probs, label='Degree 3')
plt.title("Degree Probabilities Over Time")
plt.xlabel("Time Step")
plt.ylabel("Probability")
plt.legend(title='Degree')
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
TimeStep,Degree1_Prob,Degree2_Prob,Degree3_Prob
0,0.406780,0.313559,0.271186
1,0.421053,0.315789,0.263158
2,0.427273,0.327273,0.245455
3,0.435897,0.333333,0.230769
4,0.394737,0.315789,0.289474
5,0.384000,0.312000,0.304000
6,0.436975,0.260504,0.302521
7,0.382609,0.391304,0.226087
8,0.427350,0.282051,0.290598
9,0.419643,0.303571,0.276786
10,0.465517,0.275862,0.258621
11,0.438596,0.245614,0.315789
12,0.467742,0.209677,0.322581
13,0.431034,0.275862,0.293103
14,0.435897,0.316239,0.247863
15,0.398305,0.338983,0.262712
16,0.369748,0.344538,0.285714
17,0.405172,0.293103,0.301724
18,0.448276,0.293103,0.258621
19,0.475806,0.258065,0.266129
20,0.391304,0.339130,0.269565
```

Figure 2.27. Temporal analysis of degree distribution in a mitochondrial network.

(a) The time-resolved log file records the probability of nodes having degrees 1, 2, or 3 for each frame in the image stack.

(b) The corresponding plot shows how these probabilities fluctuate over time, reflecting dynamic topological remodeling of the mitochondrial network. This time-series analysis enables quantitative comparison between experimental observations and simulated mitochondrial behavior under different conditions.

2.5.7 Cross-Validation of Initial Network Structure

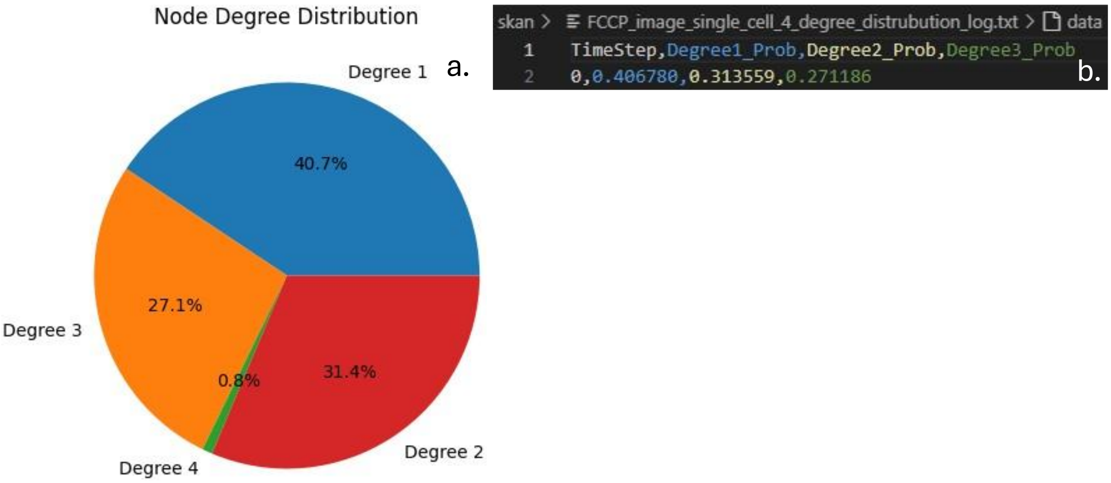
To ensure that your simulation input (node and edge CSV files) and the degree distribution log from time-series images are aligned at the first time point, it is critical to cross-validate their contents. Since these files are often generated by different scripts or parameter settings, mismatches can occur—especially after changing thresholds, smoothing, or refinement settings (fig 2.28.).

This check allows you to verify that the simulated network starts from the same structure observed in the microscopy image stack.

```
from collections import Counter
# Count degrees of all nodes
```

doi:10.6342/NTU202503815

```python
degrees = dict(modified_graph.degree())
degree_counts = Counter(degrees.values())
# Print sorted full histogram
print("Degree Distribution (sorted):")
for deg in sorted(degree_counts):
    print(f"Degree {deg}: {degree_counts[deg]} nodes")
# Optionally: Pie chart for visualization
plt.figure(figsize=(5, 5))
labels = [f'Degree {k}' for k in degree_counts.keys()]
sizes = list(degree_counts.values())
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title('Node Degree Distribution')
plt.show()
```



```
TimeStep,Degree1_Prob,Degree2_Prob,Degree3_Prob
0,0.406780,0.313559,0.271186
```

Figure 2.28. Cross-validation of initial degree distribution between simulation input and image-derived data.

(a) A pie chart summarizes the degree distribution of the initial mitochondrial network used in the simulation, computed directly from the node-edge graph.

(b) The first row of the time-series log file records the degree probabilities at time step 0, derived from the microscopy image stack.

By comparing these two sources, consistency in network structure at initialization is verified, ensuring alignment between experimental observations and simulation setup.

doi:10.6342/NTU202503815

2.6 Simulation Workflow Overview

To investigate the dynamic remodeling of mitochondrial networks, this study employs a particle-based reaction–diffusion simulation framework built in ReaDDy2. The simulation domain is initialized using the graph extracted in the previous step of “Image Processing and Network Reconstruction”. Specifically, the node and edge data shown in the green box of Fig. 2.30. are imported directly into the simulation, ensuring that the starting topology precisely reflects the experimentally observed mitochondrial skeleton. Each node is represented as a diffusing particle, and each edge is treated as a harmonic bond, thereby reconstructing the in-cellular filament architecture in silico. The particles interact through a combination of physical forces: harmonic bond and angular potentials maintain local segment length and bending stiffness, soft repulsion prevents overlapping between nonbonded particles, and reflective box boundaries confine the system to the cytoplasmic region.

Dynamic remodeling of the network emerges from explicit reaction rules that operate throughout the simulation. Structural fission reactions selectively remove internal edges away from endpoints, which causes fragmentation by reducing the connectivity of the affected vertices (degree 3 nodes are downgraded to degree 2, and degree 2 to degree 1). Fusion processes operate through two independent mechanisms: spatial fusion events merge distinct topologies when particles approach within a defined capture radius, and structural fusion detection reactions adjust node types when new junctions are formed. These reaction mechanisms collectively reproduce the interplay between elongation and fragmentation that defines mitochondrial morphology. Individual simulation runs track the time evolution of the network, while multiple replicates are used to calculate ensemble-averaged degree distributions for comparison with experimental observations.

Figure 2.29. summarizes this workflow. The green box on the left corresponds to the imported experimental graph obtained from the previous reconstruction step. The center panel illustrates the force-field representation of bonds, angles, and repulsive

doi:10.6342/NTU202503815

interactions that preserve network geometry. The right panel highlights the reaction engine that applies fission and fusion rules to modify topology over time. The resulting outputs include particle trajectories, reaction event logs, and time-resolved degree distributions, which together form the basis for validating the simulated network dynamics against experimental data.

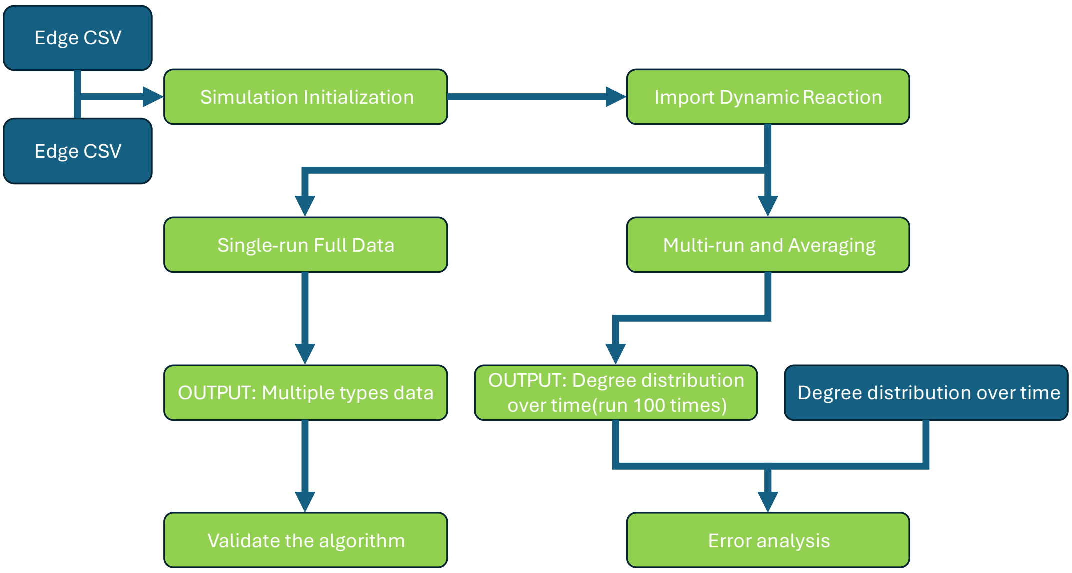

Edge CSV
Edge CSV
Simulation Initialization
Import Dynamic Reaction
Single-run Full Data
Multi-run and Averaging
OUTPUT: Multiple types data
OUTPUT: Degree distribution over time(run 100 times)
Degree distribution over time
Validate the algorithm
Error analysis

Figure 2.29 . Workflow of the ReaDDy2-based mitochondrial network simulation.

Image-derived node and edge data from the previous reconstruction step (green box) are imported as the initial network. Physical interactions (bond, angle, repulsion, and confinement) govern particle motion, while reaction modules implement fission and fusion to remodel network topology. Outputs include particle trajectories, reaction event logs, and time-resolved degree distributions.

2.6.1 Simulation Initialization and Relaxation with ReaDDy2

To construct the initial mitochondrial network for simulation, we employ a multi-stage pipeline that translates node and edge data extracted from microscopy images into a particle-based topological structure in ReaDDy2. This preparatory step ensures a stable and physically meaningful configuration before enabling dynamic reactions such as fission and fusion in subsequent phases.

doi:10.6342/NTU202503815

1. Data Import and Graph Construction

We begin by importing two CSV files: one specifying node coordinates (node_id, x_coord, y_coord) and another defining pairwise connections (source, target). Using the NetworkX library, these entries are used to reconstruct the undirected graph of the mitochondrial network. Each connected component within this graph is treated as an independent mitochondrial structure (topology).

2. Particle-Based Topology Generation

For each connected component:

- Nodes are initialized as ReaDDy2 particles (mito_node_1) with corresponding 2D spatial coordinates (z = 0 to model a quasi-2D network).
- Edges between nodes are translated into harmonic bonds within ReaDDy2 using its topology API.
- Each topology is added to the simulation using simulation.add_topology(...), automatically invoking predefined bonding and angular.

3. System Configuration

The create_ReaDDy2_system() routine assembles a ReactionDiffusionSystem whose numerical specification is captured in Table 2.1. and Fig 2.29. below. All geometric quantities are expressed in nanometers (nm) because the skeleton extracted from fluorescence microscopy has pixel-to-pixel distances on that scale (≈ 100 nm); keeping the simulation grid in the same units avoids unnecessary conversions and preserves sub-pixel precision. Energetic parameters are given in kilojoules per mole per square-nanometer (kJ mol^{-1} nm^{-2}), the natural unit for harmonic spring constants. In a harmonic potential,

$$E = \frac{1}{2}k(r - r_0)^2$$

doi:10.6342/NTU202503815

So the spring constant k (100 kJ mol^{-1} nm^{-2}) specifies the energy cost of stretching a bond by 1 nm per mole of identical bonds; higher values make bonds stiffer. Equilibrium bond lengths (ℓ_0 = 10 nm) are set equal to the average center-to-center distance between neighboring skeleton pixels after isotropic voxel scaling, ensuring that the simulated filament matches the physical length measured in the processed image.

According to Table 2.1., every skeleton node is instantiated as mito_node_1, mito_node_2, or mito_node_3, each diffusing at rate D. Pairwise repulsion eliminates overlaps, harmonic bonds of length ℓ_0 and strength k_bond preserve segment integrity, and angular springs of constant k_angle reproduce the semiflexible nature of mitochondrial tubules. The confining box, enforced by k_box, maintains the network inside the imaged cytoplasmic area. Collectively, these parameters convert the static graph into a mechanically faithful, Brownian-driven model that underpins the stochastic fusion-and-fission reactions introduced in the next section (Fig. 2.30.).

Table 2.1. ReaDDy2 simulation-box and interaction parameters.

Category	Parameter	Value	Unit	Description / Note
Simulation Box	Box size	[1000, 1000, 0.01]	nm	2-D slice of the 3-D simulation volume
	Box potential	100	100 kJ mol^{-1} nm^{-2}	Soft wall that prevents particles from leaving the field of view
Particle Types	mito_node_1	—	—	Represents a node of degree 1 (endpoint)
	mito_node_2	—	—	Represents a node of degree 2 (linear segment)
	mito_node_3	—	—	Represents a node of degree 3 (branch point)

Diffusion	Diffusion constant	0.02	nm^2 ps^{-1}*	Sets Brownian mobility of all node types
Pairwise Interaction	Repulsion force constant	10	100 kJ mol^{-1} nm^{-2}	Soft-core interaction to avoid overlap
Bonded Interaction	Bond length (equilibrium)	10	nm	Target distance between connected nodes
	Bond force constant	100	100 kJ mol^{-1} nm^{-2}	Strength of harmonic bond potential
Angular Interaction	Angle force constant	10	100 kJ mol^{-1} rad^{-2}	Bending stiffness of consecutive bonds

Table concisely lists the numerical settings of the ReaDDy system. A 1 000 nm × 1 000 nm 2-D box with a soft wall potential confines three node species that differ only in their graph degree. All nodes diffuse at 0.02 nm^2 ps^{-1}, repel each other via a gentle harmonic potential ($k = 10$), and connect through 10 nm bonds stiffened by 100 and regulated in bending by an angular constant of 10. These values anchor the simulated network to experimentally observed segment lengths, flexibility, and cytoplasmic viscosity while eliminating edge effects.

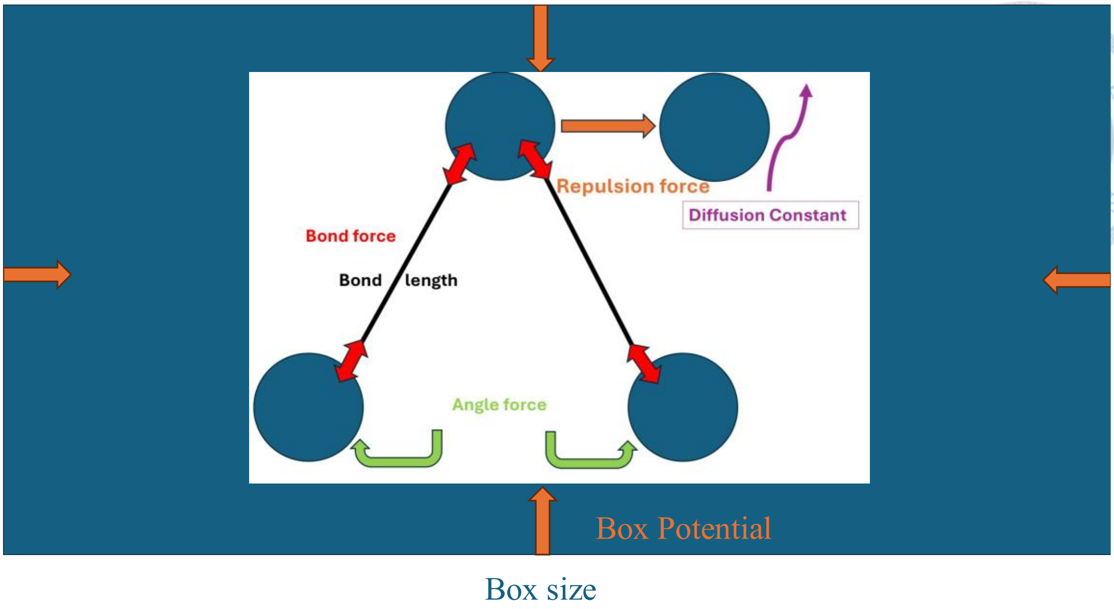


Figure 2.30. ReaDDy2 force-field and boundary situating the physical parameters schematic used in the simulation stage.

This schematic clarifies how the abstract graph iranslated into concrete mechanical interactions inside ReaDDy2:

- Particle representation – Each blue disc corresponds to a skeleton node from the Node CSV. Radii are set by experimental tubule thickness and determine excluded-volume effects.
- Bond force & bond length (red arrows) – Every black line reflects an edge from the Edge CSV. A harmonic bond force maintains the experimental segment length while allowing thermal fluctuation.
- Angle force (green brackets) – Consecutive bonds along a tubule experience a harmonic angle potential that penalizes sharp bending, preserving the semi-flexible nature of mitochondrial membranes.
- Repulsion force & diffusion constant (orange arrows/purple trace) – Non-bonded particles interact via a soft-core repulsion, and each particle's diffusion constant controls its Brownian motion; both parameters are calibrated from literature values and single-particle tracking data.
- Box potential & box size (orange frame) – The simulation volume (blue background) imposes reflective boundaries to mimic the cytoplasmic confines

observed in live-cell imaging. The box size is chosen to encompass the full skeleton with a safety margin, preventing artificial compression.

4. Dynamic Relaxation

The simulation is then executed using the run_simulation() function:

- An EulerBDIntegrator integrates Brownian motion with harmonic constraints.
- Trajectory recording and topology observation are enabled at fixed strides to capture network configurations over time.
- The system is allowed to evolve for 10,000 steps, permitting geometric relaxation under the influence of diffusion, bond/angle forces, and steric repulsion.

This pre-reaction equilibration phase ensures that all particles and bonds settle into physically plausible configurations without overlaps or unrealistic tensions.

2.6.2 Dynamic Reaction-Diffusion Simulation

Following initialization and geometric relaxation, we enable reactive behavior in the system to model the dynamic reorganization of mitochondrial networks. The simulation incorporates spatial diffusion, structural changes (fission and fusion), and topological transformations governed by a well-defined reaction scheme. This section outlines the reaction-driven simulation protocol and the dual analysis strategies employed to capture both detailed temporal dynamics and statistically robust trends.

1. Reaction Scheme

To model the dynamic behavior of mitochondrial networks, we adopt a minimal topological reaction framework inspired by prior work [32]. This framework captures two fundamental fusion-fission motifs based on local connectivity: tip-to-tip and tip-

doi:10.6342/NTU202503815

to-side interactions. These abstractions represent simplified but biophysically relevant approximations of mitochondrial morphological remodeling.

a. Tip-to-Tip Fusion and Fission

This scheme assumes two terminal (degree-1) nodes, denoted as X_1, can undergo a pairwise fusion to form a degree-2 node X_2, and conversely, X_2 can undergo fission to revert into two endpoints:

$$2X_1 \underset{b}{\overset{a}{\leftrightarrows}} X_2$$

- Fusion rate (a): Reflects the probability of two endpoints approaching and connecting.
- Fission rate (b): Encodes the likelihood of a linear segment breaking into two terminal fragments.

b. Tip-to-Side Fusion and Fission

In this scheme, a terminal node X_1 fuses with a linear node X_2, forming a degree-3 branch point X_3.

The reverse reaction models the detachment of an arm from a branched structure:

$$X_1 + X_2 \underset{b}{\overset{a}{\leftrightarrows}} X_3$$

- Fusion: Reflects lateral attachment of a tip to the side of an existing segment, creating a Y-junction.
- Fission: Represents the retraction or detachment of a branch.

These two reactions provide a minimal yet expressive vocabulary for simulating mitochondrial network plasticity through topology-level events. In our simulations, we map these schematic reactions onto particle-level transformations governed by spatial proximity, connectivity rules, and rate functions

2. Implementation in ReaDDy2

Before detailing the structural- and spatial-reaction kernels, we recall how reaction_rate is defined inside ReaDDy2. In the engine, a user-supplied rate constant λ is interpreted as the instantaneous Poisson hazard for a qualified particle pair; during each integration step of length τ the corresponding event is accepted with probability:

$$P = 1 - e^{-\lambda\tau}$$

- λ∈R_(≥0): represent the rection rate.
- τ: time step.

Because this quantity is ultimately a per-step acceptance probability, and to avoid confusion with biological fusion or fission rates that are usually reported as events $\cdot$ cell^{-1} $\cdot$ s^{-1}, we hereafter refer to every ReaDDy2 rate parameter simply as a reaction probability (P_fus, P_fis, …).

In cell biology, mitochondrial fusion and fission rates are experimentally defined as the number of observable events (branch joining or fragment splitting) per cell per unit time, often on the scale of minutes. These biological rates are emergent properties of the network and depend on organelle size, protein machinery activity (e.g., Drp1 for fission, Mfn1/2 and OPA1 for fusion), and local cellular conditions. They are measured statistically from live-cell imaging and are typically in the range of 0.01–0.1 events per mitochondrion per minutefilePRXLife.2.043002. By contrast, in ReaDDy2 the user-specified λ does not directly represent such a per-cell rate; it controls the per-step probability that a qualified particle pair will undergo a reaction.

With this convention, the experimentally calibrated probabilities are mapped onto one of two independent reaction classes in ReaDDy2: (a) spatial reactions that check whether two candidate particles which do not share the same topology fall within the prescribed capture radius with probability $P_{spatial}$ decides whether they instantaneously create a bond between them and change their particle type or not, and (b) structural reactions by contrast, act inside an existing topology: with probability $P_{structrual}$ a rule rewires the internal graph. These two mechanisms are evaluated separately each time-step. This separation enables us to couple detailed topological updates to a distance-based encounter filter while retaining a transparent correspondence between simulation parameters and their biological counterparts.

a. Spatial Reactions: Topology Merging

For internal modifications, topology merging is enabled through spatial rules:

- "mito_node_1 + mito_node_1" → merged topology (tip-to-tip fusion) (fig 2.31.)

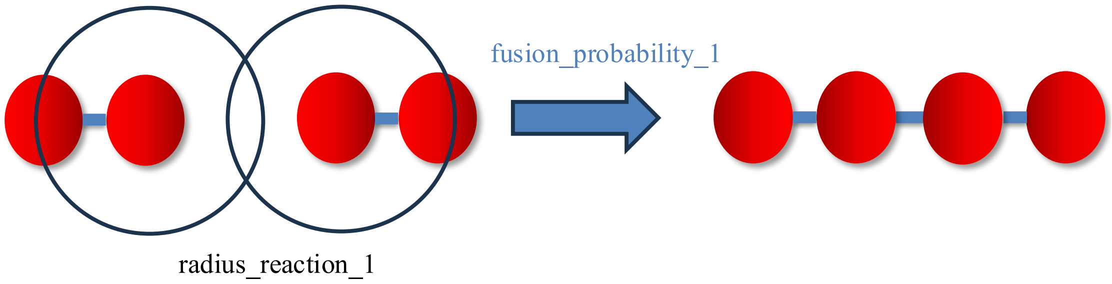


Figure 2.31. Tip-to-tip fusion triggered by a spatial reaction in ReaDDy2.

Two linear mitochondrial fragments approach each other, and when the reactive zones (radius_reaction_1) around their terminal nodes overlap, a probabilistic fusion event may occur with a likelihood defined by fusion_probability_1. Successful fusion merges the two fragments into a single continuous filament, thereby enabling dynamic remodeling of the mitochondrial network through tip-to-tip spatial reactions.

- "mito_node_1 + mito_node_2" → merged topology (tip-to-side fusion) (fig 2.32.).

doi:10.6342/NTU202503815

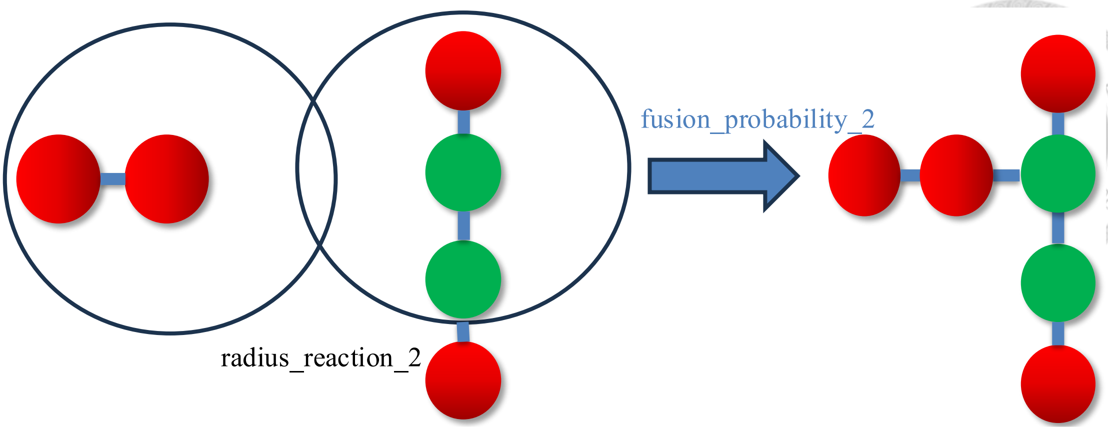


Figure 2.32. Tip-to-side fusion mediated by spatial reactions in ReaDDy2.

A terminal node (red) of one mitochondrial fragment approaches the inear fragment composed of two-degree nodes (green). When their reactive zones (radius_reaction_2) overlap, a fusion event can occur with a probability determined by fusion_ probability _2. Successful fusion attaches the tip to the side of the filament, forming a Y-shaped structure in which the contacted site becomes a higher-degree branching node. This spatial reaction allows the mitochondrial network to develop complex branching architectures through tip-to-side connections.

These spatial reactions are registered via ReaDDy2's add_spatial_reaction API:

```
system.topologies.add_spatial_reaction(
    "fusion_1: mitochondria(mito_node_1) + mitochondria(mito_node_1) -> mitochondria(mito_node_1--mito_node_1)",
    rate=FUSION_PROBABILITY_1,
    radius=RADIUS_REACTION
)
```

The spatial proximity constraint (controlled by RADIUS_REACTION) ensures these reactions only occur when particles are physically near each other, preserving spatial realism.

b. Structural Reactions: Topology-Based Logic

doi:10.6342/NTU202503815

In addition to fission logic via ReaDDy2's structural reaction framework, which operates on particle connectivity:

- Fusion Detection: Encoded via fusion_detect_function. The function scans all vertices for specific degree motifs:
    - Two mito_node_1 with 2 connections → both upgraded to mito_node_2 (tip-to-tip fusion) (fig 2.33.).
    - Mito_node_2 (deg=3) and mito_node_1 (deg=2) → upgraded to mito_node_3 and mito_node_2 respectively (tip-to-side fusion) (fig 2.34.).
    - Export fusion position → record data to validate the tracking algorithm.

This function mirrors the forward reaction logic of the schemes above.

```
if current_type == "mito_node_1" and num_connections == 2:
    if neighbor_type == "mito_node_1" and neighbor_connections == 2:
        # Tip-to-tip fusion
        change both to mito_node_2
elif current_type == "mito_node_2" and num_connections == 3:
    if neighbor_type == "mito_node_1" and neighbor_connections == 2:
        # Tip-to-side fusion
        change to mito_node_3 and mito_node_2
```

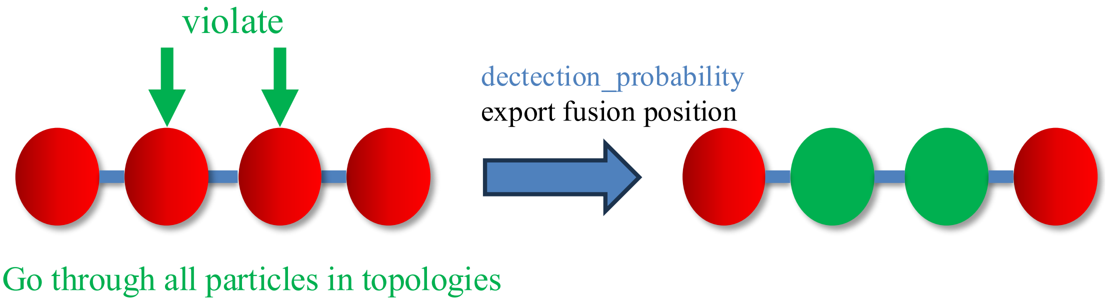


Figure 2.33. Structural reaction: detection of tip-to-tip fusion sites in ReaDDy2.

During structural reaction processing, the algorithm iterates over all particles within each topology to identify degree patterns that violate fusion criteria. When two terminal nodes (mito_node_1) are detected at adjacent positions with a connectivity pattern corresponding to tips, a fusion detection step is triggered with a probability defined by detection_probability. If successful, the local connectivity is updated, converting the involved nodes into internal nodes (mito_node_2) and exporting the fusion position for subsequent topological rewiring.

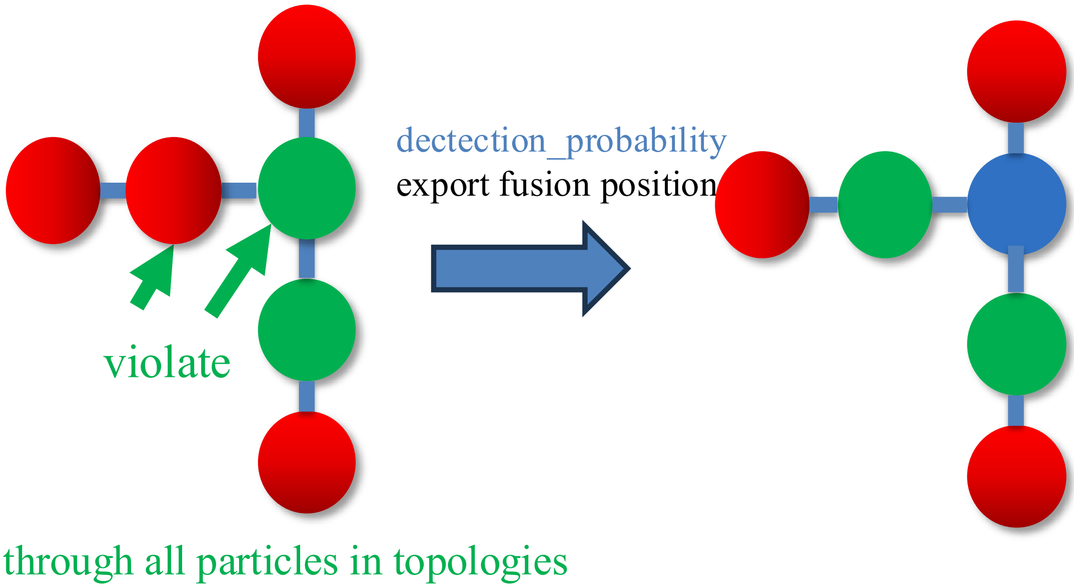


Figure 2.34. Structural reaction: detection of tip-to-side fusion sites in ReaDDy2.

Within the structural reaction framework, the algorithm iterates through all particles in each topology to identify branching violations that correspond to tip-to-side fusion motifs. When a terminal node (mito_node_1) is detected adjacent to the lateral position of a filament containing an internal node (mito_node_2), a fusion detection step is triggered with a probability determined by detection_probability. Upon a successful event, the node at the contact site is converted into a higher-degree branching node (mito_node_3), while the tip is reclassified as mito_node_2, and the updated fusion position is exported for topological rewiring.

- Fission: Implemented using dissociation_reaction_function. A topology is scanned for internal edges (excluding endpoints and their neighbors), and a randomly chosen one is removed. After edge removal:
    - The vertex types are downgraded based on degree:

doi:10.6342/NTU202503815

- $X_2 \rightarrow X_1$ (fig 2.35.) , $X_3 \rightarrow X_2$ (fig 2.36.)
- This maps directly onto the reverse of both tip-to-tip and tip-to-side schemes.
- Export fission position → record data to validate the tracking algorithm.

```
if current_type == "mito_node_3":
    recipe.change_particle_type(vertex, "mito_node_2")
elif current_type == "mito_node_2":
    recipe.change_particle_type(vertex, "mito_node_1")
```

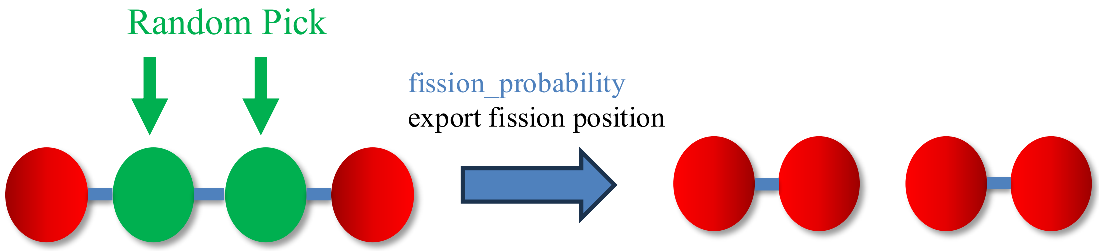


Figure 2.35. Structural reaction: fission of a linear mitochondrial fragment in ReaDDy2.

For fission events, the algorithm identifies internal edges within a filament (excluding endpoints and their immediate neighbors) and randomly selects one as the cut site. With a probability defined by fission_probability, the selected edge is removed, splitting the filament into two smaller fragments. Following edge removal, the node degrees are updated: internal nodes (mito_node_2) at the cut location are downgraded to terminal nodes (mito_node_1). The position of the fission event is recorded to support network topology tracking.

doi:10.6342/NTU202503815

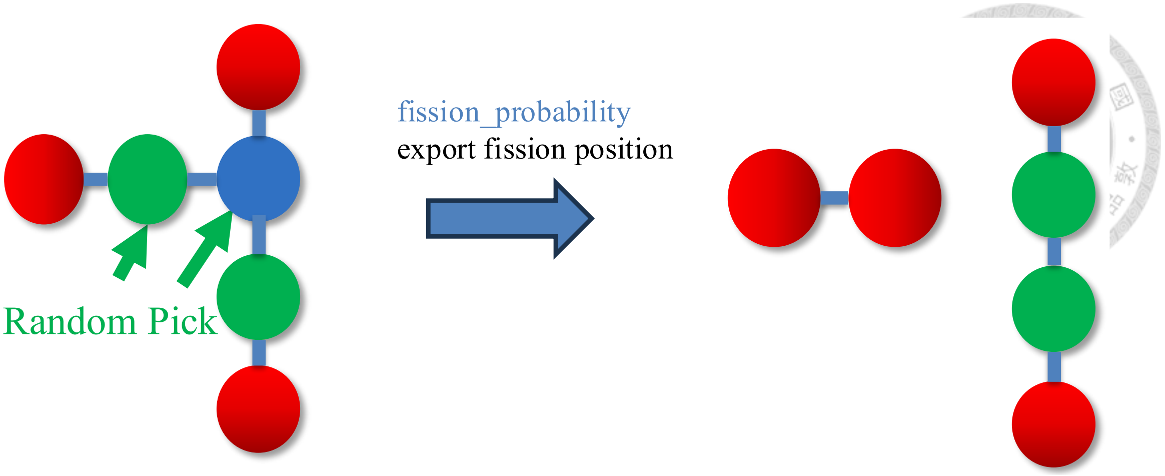


Figure 2.36. Structural reaction: fission of a branched mitochondrial fragment in ReaDDy2.

In branched topologies, the fission algorithm randomly selects an internal edge connected to a branching node (mito_node_3). With a probability determined by fission_probability, this edge is removed, detaching one branch from the network. Following the disconnection, the branching node is downgraded from mito_node_3 to mito_node_2, reflecting its reduced connectivity. The location of the fission event is recorded for downstream analysis and validation of network evolution.

Together, these implementations form a unified framework for simulating network morphogenesis. Each observed fusion or fission event corresponds to a direct mapping of the abstract reaction schemes, grounded in dynamic spatial interactions and topological evolution. This structure allows the simulation to explore mitochondrial fragmentation, branching, and reassembly in a biologically interpretable way.

2.6.3 Single-Run Full Data Simulation

To capture the temporal evolution of mitochondrial morphology, we conduct a single, long simulation with comprehensive logging of system observables:

doi:10.6342/NTU202503815

- Topology Tracking: Particle connectivity and fragmentation are tracked at every stride, enabling analysis of structural evolution.
- Reaction Event Logging: Every fission and fusion event is logged, including timestamp, positions, and particle IDs. This allows spatial-temporal mapping of network remodeling.
- Trajectory Recording: Full 3D coordinates of all particles are saved per frame, facilitating downstream spatial visualization and rendering (e.g., PDB, PSF output for VMD).
- Statistical Metrics:
  - Node degree probabilities (degree-1, 2, 3) over time.
  - Fragment count and fragment size distribution, characterizing network breakdown or merging.
  - Average polymer length per time step.

These results are written to human-readable logs and plotted to visualize trends such as increasing fragmentation, emergence of branching motifs, or stabilization of topology complexity.

2.6.4 Multi-Run Statistical Averaging (Secondary Script)

To quantify the robustness of observed trends, a second script performs repeated simulations (e.g., 100 replicates), each starting from the same relaxed network:

- Only node degree distributions are computed for each run.
- Outputs include time-averaged probabilities for degree-1 (endpoints), degree-2 (chains), and degree-3 (branch points).

By averaging across runs, this approach smooths stochastic fluctuations and allows comparison to experimental data or theoretical baselines. This strategy is particularly effective for validating network topology models under variable conditions such as drug treatments (e.g., FCCP exposure).

2.7 Visualization and Post-Simulation Validation

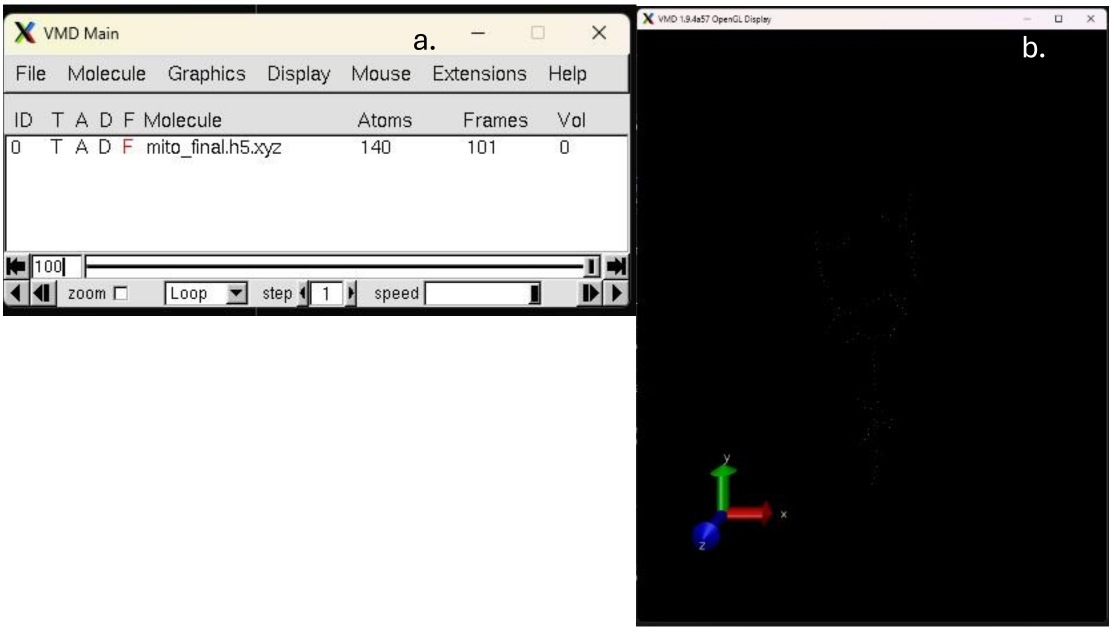
In this section we will teach how to use VMD (Visual Molecular Dynamics)[40] with Single-Run Full Data Simulation’s output (e.g. XYZ file, PDB file, PSF file) to examine whether the simulation crashes or not with XYZ file first, then we use PSF file and PDB file to produce accurate image.

2.7.1 Validation with XYZ file

We assume you already downloaded VMD, and we use MobaXterm as Linux platform with remote server to operate.

a. Open VMD with XYZ.file

After running the simulation, there should be an output called “*.h5.xyz”, then type linux command “vmd *.h5.xyz”, it should come up with two windows (fig 2.37.). We can see that the visualization is barely seen in default status, so you need to change the representation of the network.



ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	mito_final.h5.xyz	140	101	0

Figure 2.37. Initial visualization of simulation output in VMD using XYZ format.

(a) The VMD Main window shows the successful loading of a simulation output file (mito_final.h5.xyz) containing 140 atoms across 101 frames.

(b) The OpenGL Display window renders the particle positions, though visibility is limited due to default representation settings.

To enhance interpretability of the mitochondrial network, representation styles (e.g., particle size, color, rendering method) must be manually adjusted in the VMD graphical interface.

b. adjusts representation of the network

After you press representations, it should come up with another window as well(fig 2.38.).

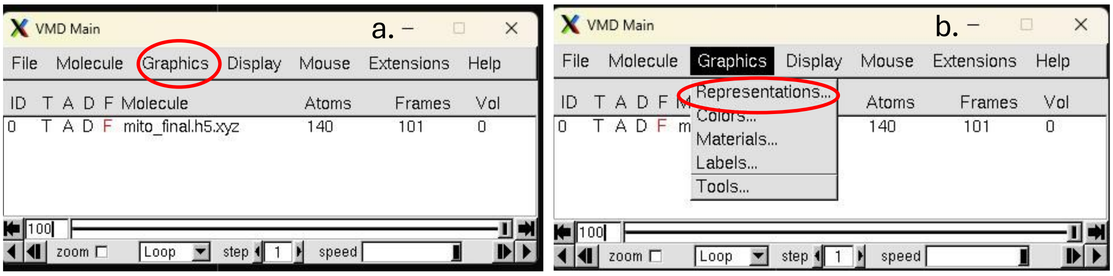
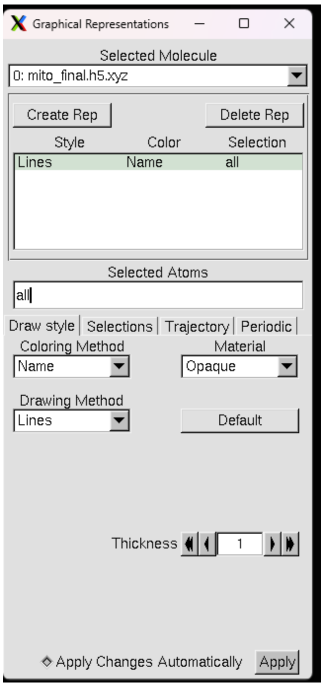


Figure 2.38. Accessing the graphical representation settings in VMD.

(a) To enhance visualization of simulation output, the user navigates to the Graphics menu in the VMD Main window.

(b) From the dropdown, the Representations... option is selected to open the display settings panel

This panel allows users to adjust rendering styles (e.g., particle shape, size, and color) to make mitochondrial network structures clearly visible in the OpenGL display.



Graphical Representations

Selected Molecule: 0: mito_final.h5.xyz

Style	Color	Selection
Lines	Name	all

Selected Atoms: all

Coloring Method: Name

Material: Opaque

Drawing Method: Lines

Thickness: 1

Figure 2.39. Graphical Representations panel in VMD for customizing network visualization.

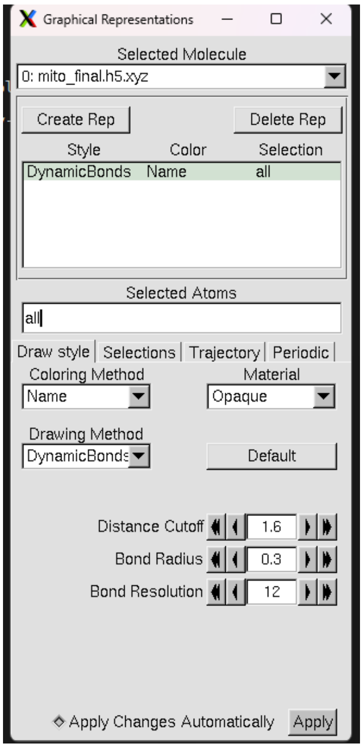
This panel allows users to adjust how particles are rendered in the OpenGL display. Users can define which atoms to display (e.g., using the selection keywords all), choose a coloring method (e.g., by Name), set material appearance (Opaque), and select a drawing method such as Lines, VDW, or Points. Line thickness and other visual parameters can be fine-tuned for clarity. These settings help highlight mitochondrial network structures for qualitative inspection and presentation.

This window (fig 2.39.) can be divided into four parts:

- Rep section (can decide what type of particle you want to change its shape):
  - by press create Rep and type "name type_*" in line of Selected Atoms to select specific particle type. In VMD, particle types are automatically named as type_0(mito_node_1), type_1(mito_node_2), type_2(mito_node_3), and so on, based on the order they appear in the input file (like. xyz or .pdb).
- Coloring Method: Change particle color (except Name and Type, they are same color
- Drawing Method: Change particle representation.

- Material: change the texture of the particle.

Here is the window (fig 2.40.) when I choose dynamicBonds in Drawing Method (branch-like).


Graphical Representations
Create Rep
Delete Rep
Draw style | Selections | Trajectory | Periodic |
Default
Apply

Selected Molecule
0: mito_final.h5.xyz

Style	Color	Selection
DynamicBonds	Name	all

Selected Atoms
all

Coloring Method: Name
Material: Opaque
Drawing Method: DynamicBonds

Distance Cutoff 1.6
Bond Radius 0.3
Bond Resolution 12

◇ Apply Changes Automatically

Figure 2.40. Enhanced mitochondrial network rendering using DynamicBonds in VMD.

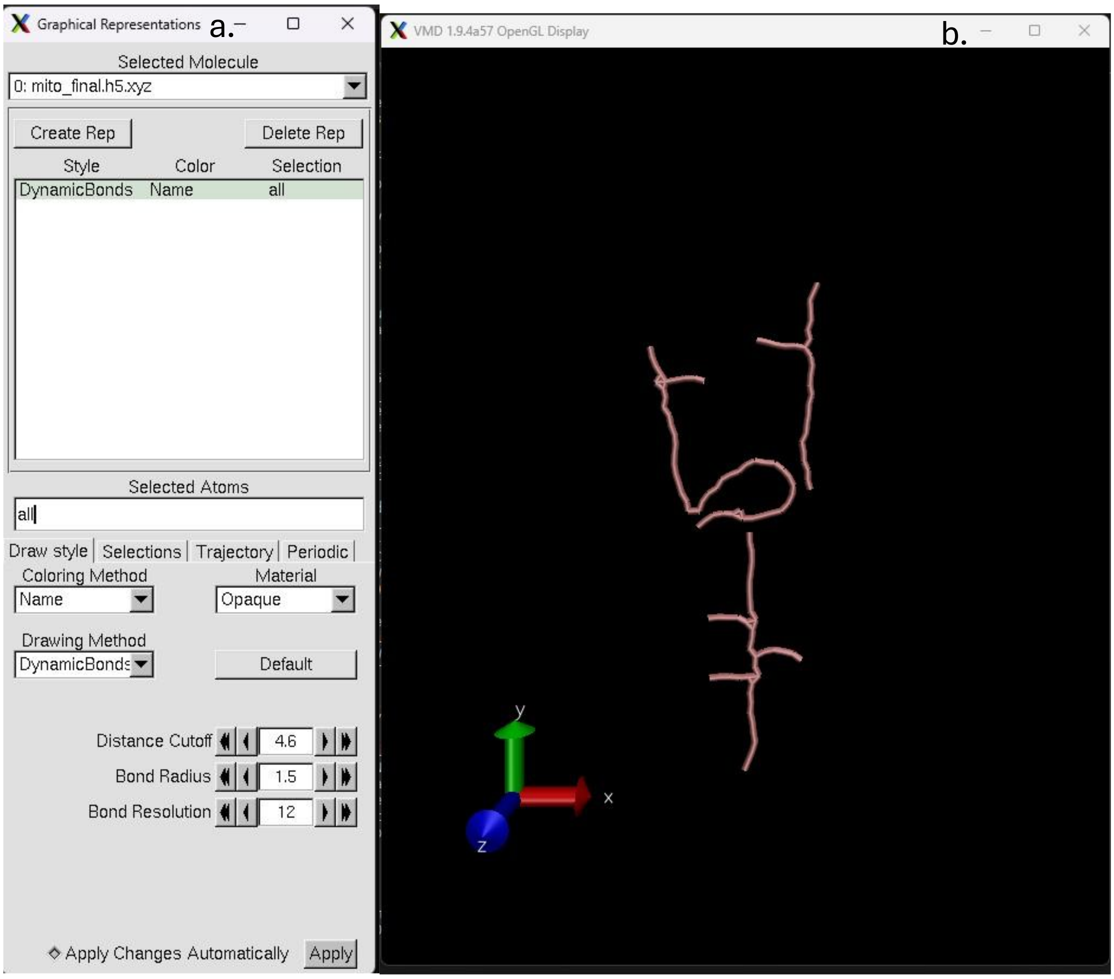
The Graphical Representations panel is configured to use the DynamicBonds drawing method, which visually connects nearby particles based on a distance criterion. The Distance Cutoff is set to 1.6, meaning bonds are drawn between particles within this range. The Bond Radius and Bond Resolution control the visual thickness and smoothness of the bonds. This setting improves interpretability by emphasizing network connectivity and structure in 3D mitochondrial simulations.

There are three parameters we can adjust:

- Distance cutoff: setting determines how close two atoms must be to visually form a bond. If two atoms are within this distance, VMD draws a bond between them. Since the ReaDDy2 define the bond in topology is spring-like, so that is why bond is unstable in visualization, if you want you bond more stable, please set the distance cutoff more than bond_length in ReaDDy2, for example: bond_length =3.5 , distance cutoff should above 4.0.

- Bond radius: setting bond thickness
- Bond resolution: I didn’t see any change from visualization.

After you press apply result should like this (fig 2.41.).



Selected Molecule: 0: mito_final.h5.xyz

Style	Color	Selection
DynamicBonds	Name	all

Selected Atoms: all

Coloring Method: Name

Material: Opaque

Drawing Method: DynamicBonds

Distance Cutoff: 4.6

Bond Radius: 1.5

Bond Resolution: 12

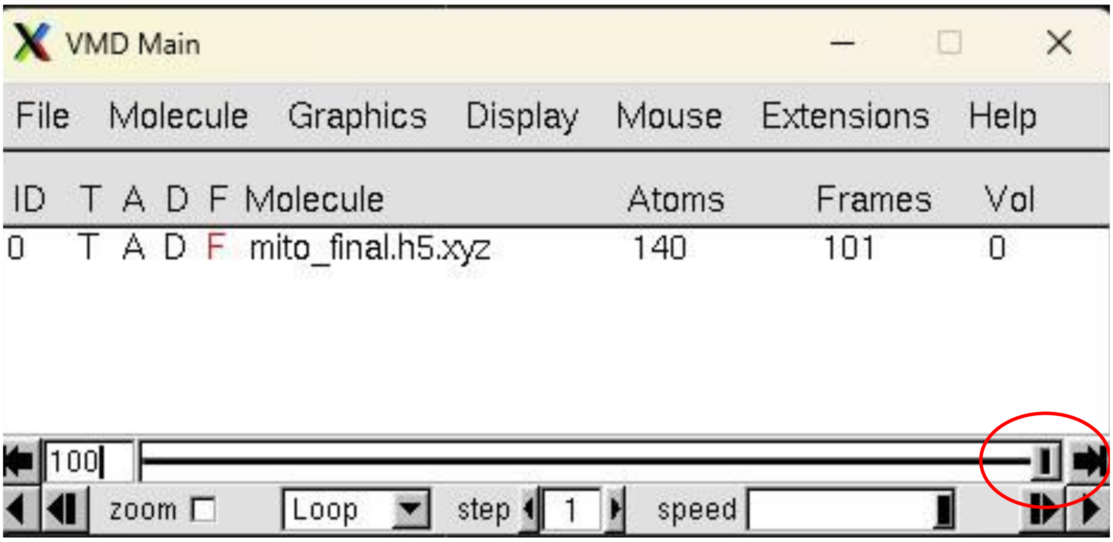
◇ Apply Changes Automatically

Figure 2.41 Final visualization of the mitochondrial network using VMD DynamicBonds representation.

(a) The Graphical Representations panel is configured with the DynamicBonds drawing method. A higher Distance Cutoff (4.6), Bond Radius (1.5), and bond resolution (12) are used to enhance visual continuity and clarity of network segments.

(b) The OpenGL Display window shows a clearly connected mitochondrial network structure rendered with smooth cylindrical bonds, allowing for intuitive inspection of topology and dynamics across simulation frames.

Then we can scroll the time slide to check whether simulation is crash or not (fig 2.42.).



ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	mito_final.h5.xyz	140	101	0

100

Loop

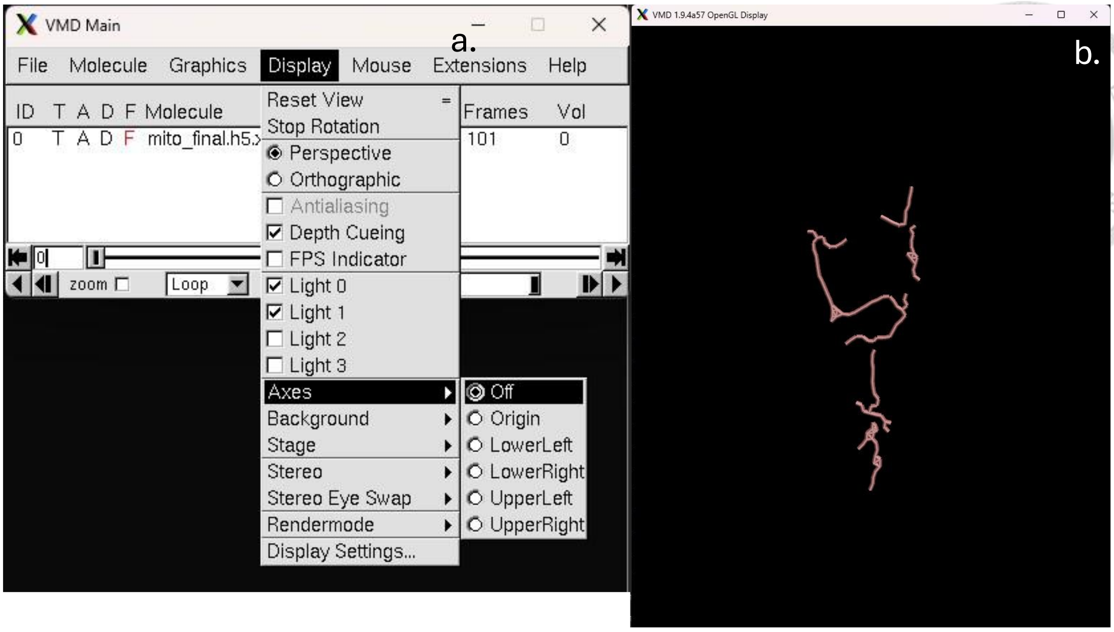
step 1

Figure 2.42. Time navigation in VMD for simulation frame inspection.

The VMD Main window includes a timeline slider (circled in red) that allows users to scroll through simulation frames. By manually advancing through the trajectory, users can visually inspect the mitochondrial network across time steps to verify structural continuity and detect potential simulation crashes, discontinuities, or anomalies.

c. removes the axis from window (optional)

If you want an axis to disappear (fig 2.43.).



ID	T	A	D	F	Molecule	Frames	Vol
0	T	A	D	F	mito_final.h5.x	101	0

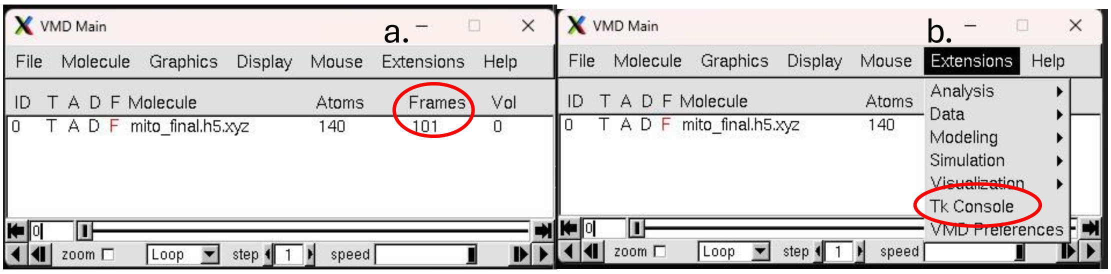
Figure 2.43. Optional removal of coordinate axes in VMD display.

(a) In the VMD Main window, the user accesses the Display menu and selects Axes > Off to hide the coordinate axes.

(b) The OpenGL Display window then presents a cleaner visualization of the mitochondrial network without distraction from axis indicator

d. import simulation visualization as tiff file

Finally, let’s make a tiff file, if you want to know how many frames you make. So here how it goes (fig 2.44.).



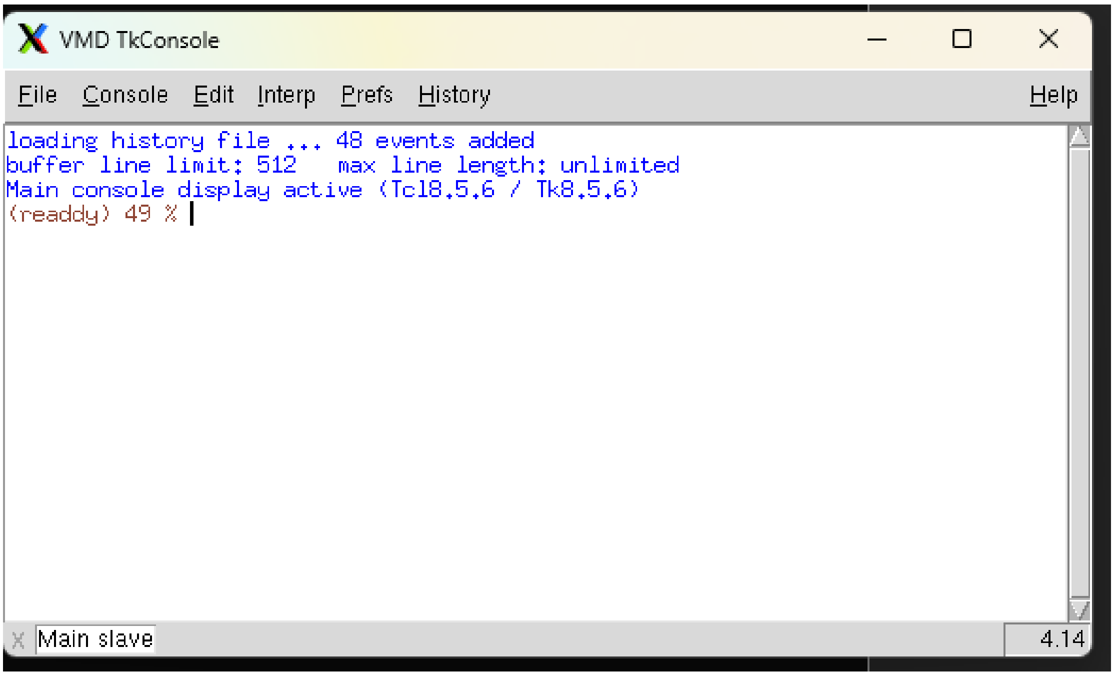
ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	mito_final.h5.xyz	140	101	0

Figure 2.44. Preparing to export simulation frames as TIFF images in VMD.

(a) The Frames column in the VMD Main window indicates the total number of simulation frames available for export (e.g., 101 frames).

(b) To begin exporting images, the user accesses the Tk Console via the Extensions menu. The console allows scripted batch export of each frame as a TIFF file, enabling frame-by-frame visualization or construction of time-lapse animations for mitochondrial network dynamics.

After that, it should come up with another window (fig 2.45.).



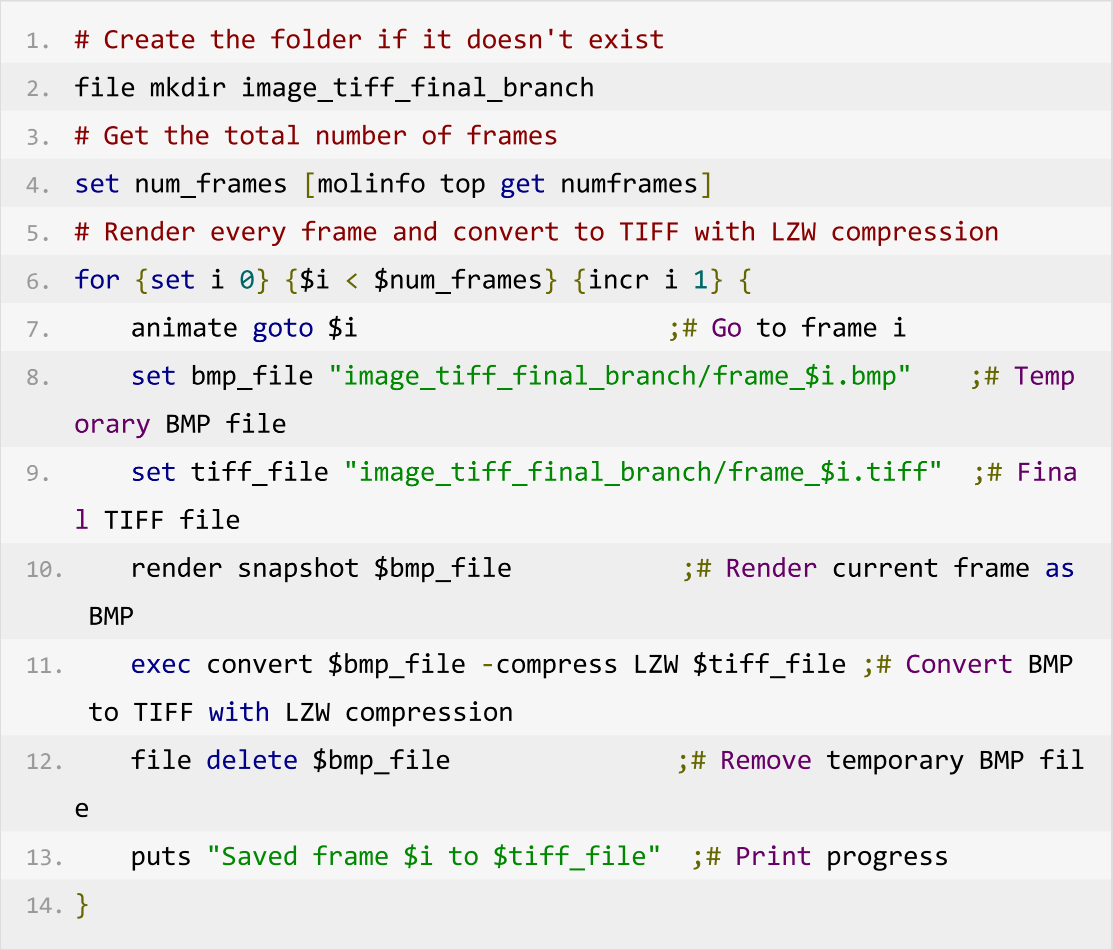
```
loading history file ... 48 events added
buffer line limit: 512   max line length: unlimited
Main console display active (Tcl8.5.6 / Tk8.5.6)
(readdy) 49 %
```

Figure 2.45. Using the VMD TkConsole to export mitochondrial simulation frames as TIFF images.

The TkConsole window in VMD enables scripted control over visualization export. After setting the desired graphical representation and checking the total number of frames (see previous figure), the user enters a Tcl script into the console to export each frame as a .tiff image. This step facilitates generation of high-resolution image sequences for time-lapse analysis or movie creation.

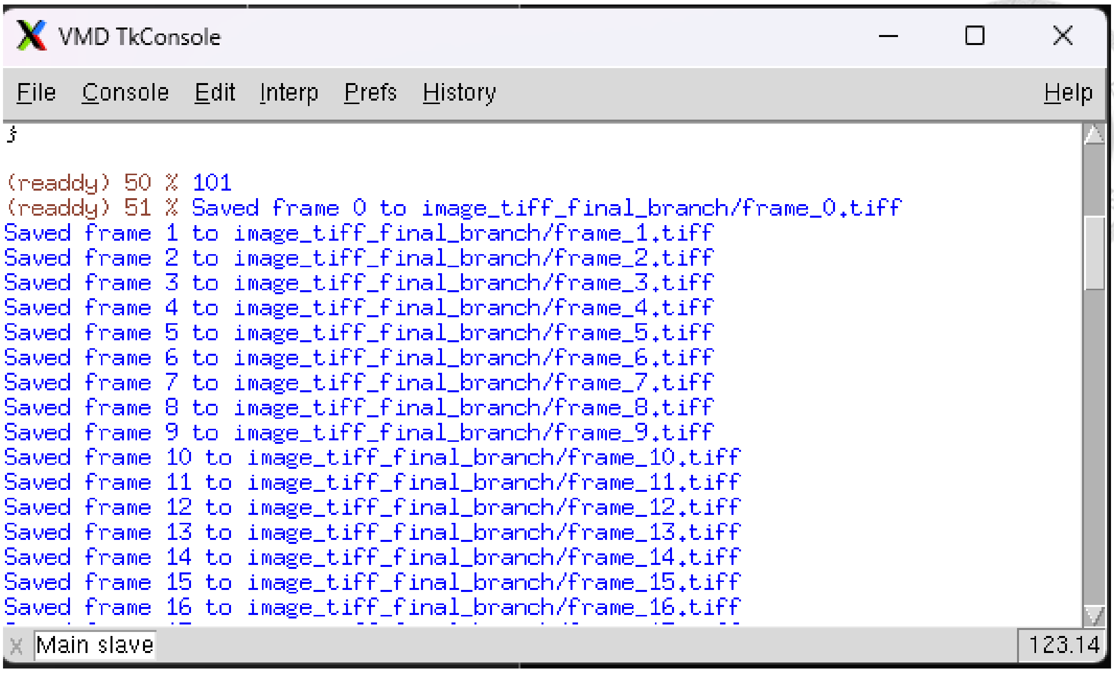
Type command below:

doi:10.6342/NTU202503815



```
# Create the folder if it doesn't exist
file mkdir image_tiff_final_branch
# Get the total number of frames
set num_frames [molinfo top get numframes]
# Render every frame and convert to TIFF with LZW compression
for {set i 0} {$i < $num_frames} {incr i 1} {
    animate goto $i                        ;# Go to frame i
    set bmp_file "image_tiff_final_branch/frame_$i.bmp"    ;# Temporary BMP file
    set tiff_file "image_tiff_final_branch/frame_$i.tiff"  ;# Final TIFF file
    render snapshot $bmp_file              ;# Render current frame as BMP
    exec convert $bmp_file -compress LZW $tiff_file ;# Convert BMP to TIFF with LZW compression
    file delete $bmp_file                  ;# Remove temporary BMP file
    puts "Saved frame $i to $tiff_file"  ;# Print progress
}
```

If code runs successfully, the window should be like this (fig 2.46.).



```
$

(readdy) 50 % 101
(readdy) 51 % Saved frame 0 to image_tiff_final_branch/frame_0.tiff
Saved frame 1 to image_tiff_final_branch/frame_1.tiff
Saved frame 2 to image_tiff_final_branch/frame_2.tiff
Saved frame 3 to image_tiff_final_branch/frame_3.tiff
Saved frame 4 to image_tiff_final_branch/frame_4.tiff
Saved frame 5 to image_tiff_final_branch/frame_5.tiff
Saved frame 6 to image_tiff_final_branch/frame_6.tiff
Saved frame 7 to image_tiff_final_branch/frame_7.tiff
Saved frame 8 to image_tiff_final_branch/frame_8.tiff
Saved frame 9 to image_tiff_final_branch/frame_9.tiff
Saved frame 10 to image_tiff_final_branch/frame_10.tiff
Saved frame 11 to image_tiff_final_branch/frame_11.tiff
Saved frame 12 to image_tiff_final_branch/frame_12.tiff
Saved frame 13 to image_tiff_final_branch/frame_13.tiff
Saved frame 14 to image_tiff_final_branch/frame_14.tiff
Saved frame 15 to image_tiff_final_branch/frame_15.tiff
Saved frame 16 to image_tiff_final_branch/frame_16.tiff
```

Figure 2.46. Batch export of simulation frames to TIFF images via the VMD TkConsole.

Upon running a loop script in the TkConsole, each frame of the trajectory is sequentially saved as a high-resolution .tiff file in the specified output directory (image_tiff_final_branch/). Successful execution is confirmed by a log of messages indicating the saved frame numbers and corresponding filenames.

There should be a folder call image_tiff_final_branch in your terminal.

2.7.2 Import Simulation Visualization as tiff with PDB file and PSF file.

If we want an accurate tiff, you need to make sure that you have a two folders keep PDB files and PSF files separately. Once confirm, type “vmd” in your terminal and open tk console and type command below:

```
# Directories containing the PDB and PSF files
set pdb_dir "mito_final_pdb_files/"
set psf_dir "mito_final_psf_files/"
set output_dir "image_tiff_final_branch/"

# Ensure the output directory exists
```

```
file mkdir $output_dir

# Set the rendering window size
display resize 672 848

# Turn off the axes
axes location off

# Loop over the time indices
for {set i 0} {$i <= 1000} {incr i 10} {
    # Format the time index
    set time_index [format "%d" $i]

    # Construct file paths
    set pdb_file "${pdb_dir}mito_final_time_${time_index}.pdb"
    set psf_file "${psf_dir}mito_final_time_${time_index}.psf"

    # Check if both files exist
    if {[file exists $pdb_file] && [file exists $psf_file]} {
        # Load the PSF and PDB files into a new molecule
        set molid [mol new $psf_file]
        mol addfile $pdb_file molid $molid

        # Apply Bonds representation with specified radius and resolution
        mol delrep 0 $molid
        mol representation Bonds 13.0 12
        mol color Name
        mol addrep $molid

        # Calculate the frame index by dividing time_index by 10
        set frame_index [expr {$time_index / 10}]

        # Render and save the image as TIFF
        set bmp_file "${output_dir}frame_${frame_index}.bmp"
        set tiff_file "${output_dir}frame_${frame_index}.tiff"
        render snapshot $bmp_file
        # Ensure ImageMagick's 'convert' tool is available for this command
        exec convert $bmp_file -compress LZW $tiff_file
        file delete $bmp_file

        # Delete the molecule to free memory
        mol delete $molid

        puts "Rendered and saved: $tiff_file"
    } else {
        puts "Warning: Missing files for time index $time_index"
    }
```

2.8 Quantitative Comparison with Experimental Data

To validate the fidelity of our reaction-diffusion simulation against biological behavior, we conduct a quantitative comparison between the simulated and experimentally observed mitochondrial network structures. Specifically, we focus on node degree distributions—a coarse-grained yet informative metric reflecting topological complexity and branching behavior over time.

1. Data Source and Processing

This analysis leverages outputs from the multi-run statistical averaging simulation, in which each mitochondrial network is simulated 100 times under identical initial conditions. For each replicate, the simulation tracks the probability of node degrees 1, 2, and 3 at every time step. These replicate-level logs are saved as: degree_probabilities_rep*.txt and Averaging across these replicates yields one consolidated file per cell: degree_probabilities_average.txt.

Each row corresponds to a time step, and each column reports the probability of encountering nodes of a specific degree.

2. Experimental Data Extraction

Experimental data consists of time-lapse microscopy images of mitochondrial networks. These are processed using skeletonization and graph-based methods (e.g., via the skan package) to extract degree logs over time. Each processed file is named according to the corresponding simulation: <cell_id>_degree_probabilities_log.txt

This file mirrors the simulated format with columns for degree-1, -2, and -3 probabilities over time.

3. Simulation vs. Experiment Comparison

To evaluate the fidelity of the simulation outputs against experimental observations, a structured comparison pipeline was established. For each cell, the first step involved matching simulation results with the corresponding experimental dataset using a shared cell identifier. Once the file pairs were identified, a validation procedure was applied to confirm that the time-series data from both sources were directly comparable. This validation ensured that the temporal sampling intervals were consistent, that the data arrays were aligned in length, and that both datasets were free from missing or undefined values (Nans).

After validation, quantitative discrepancies between simulation and experiment were assessed through a time-resolved computation of the mean absolute error (MAE) across the three primary node degree classes. At each time step t, the error was defined as:

$$MAE_t = \frac{1}{3}\sum_{d=1}^{3} |P_d^{sim}(t) - P_d^{exp}(t)|$$

Where $P_d^{sim}(t)$ and $P_d^{exp}(t)$ represent the degree-d probabilities obtained from simulation and experimental measurements, respectively.

doi:10.6342/NTU202503815

Following the calculation of time-resolved errors, the pipeline aggregated these values at two levels. First, a per-cell mean error was computed to evaluate how well the simulation reproduced the temporal evolution of network structure for each individual cell. Second, to summarize performance across the dataset, the distribution of per-cell errors was used to report overall statistics, expressed as the mean error and its associated standard error. This two-tiered approach provided a robust metric for assessing the agreement between simulated and experimental mitochondrial network dynamics at both single-cell and population levels.

4. Visualization and Reporting

The analysis pipeline incorporates a standardized procedure for visualization and data reporting. For each individual cell, time-resolved degree probabilities are plotted to depict the evolution of the network structure throughout the simulation. To summarize model performance at the population level, a scatter plot is generated in which each point represents the per-cell mean absolute error; superimposed on these points, the overall mean and associated standard error of the mean (SEM) are shown as error bars. In addition, the computed per-cell errors are exported as a structured CSV file (e.g., 02_control_single_cell_error_summary.csv), enabling further statistical analyses and facilitating comparisons across different experimental or pharmacological conditions.

5. Model Evaluation and Refinement

This quantitative evaluation framework provides an objective basis for assessing the realism of the computational model. By directly comparing simulation-derived degree distributions to their experimental counterparts, it becomes possible to determine whether the current parameterization and reaction schemes reproduce the key features of mitochondrial network dynamics. Furthermore, systematic deviations—such as consistent underestimation of degree-3 branching events—can be readily detected. These insights guide iterative refinement of the model through targeted adjustments to fission and fusion rates, reaction probabilities, and structural

doi:10.6342/NTU202503815

constraints, with the goal of improving the biological fidelity of the simulation outputs.

6. Multi-Condition Summary Analysis

The final script generalizes this analysis ins_1 cell across six experimental conditions:

Table 2.1 . Summary of experimental conditions and cell models used in this study. Five treatments (Control, FCCP, Mdivi-1, Oligomycin, Rotenone) were applied to Ins-1 cell to examine mitochondrial network remodeling. Treatments differentially shift the fission–fusion balance.

Condition	Main Function	Expected Network Impact
Control	Baseline, no perturbation	Balanced fission/fusion[7]
FCCP	ΔΨm uncoupler	↑ Fission / ↓ Fusion[10]
Mdivi-1	DRP1 inhibitor (blocks fission)	↓ Fission / ↑ Fusion[11]
Oligomycin	ATP synthase inhibitor	↑ Fission[10]
Rotenone	Complex I inhibitor	↑ Fission / ↓ Fusion[6]

And three different three types of cells.

Table 2.2 . Baseline characteristics of the three cell types (Ins-1, PANC-1, AC-16) and their intrinsic mitochondrial network states.

Cell Type	Cell State / Characteristics	Mitochondrial Network Status
Ins-1	Rat pancreatic β-cell; glucose-responsive, high oxidative phosphorylation	Moderately interconnected; balanced fission/fusion[7]
PANC-1	Human pancreatic ductal carcinoma; high metabolic plasticity	Highly fragmented; ↑ Fission / ↓ Fusion[8]

AC-16	Human ventricular cardiomyocyte; high ATP demand, oxidative stress	Dense small mitochondria; Slightly ↑ Fission[15]

For each experimental condition, the corresponding summary file (e.g., 02_FCCP_single_cell_error_summary.csv) containing the per-cell mean absolute error values is imported and processed. These data are aggregated to produce a condition-specific distribution of errors, which is subsequently visualized in a comparative plot. In this plot, individual black points represent the error values for each cell, while blue markers with error bars denote the mean error and its associated standard error (SE) for that condition. This comparative visualization facilitates quantitative evaluation of model performance across different pharmacological perturbations. It enables direct ranking of model fidelity between experimental conditions, highlights outlier behaviors or inconsistent simulation fits, and provides a data-driven basis for refining reaction rate parameters in response to specific cellular perturbations.

3. Chapter 3: Results

This study implements a two-pronged simulation framework to explore the structural evolution of mitochondrial networks under varying conditions. The first approach focuses on a high-resolution single-run simulation, offering insight into detailed topological and particle-level behavior. The second adopts a multi-run statistical averaging strategy to extract reliable trends and assess the reproducibility of dynamic outcomes. Together, these approaches balance mechanistic clarity with statistical robustness.

3.1 Single-Run Simulation — Comprehensive Structural and Topological Analysis

The single-run simulation captures the dynamics of mitochondrial morphology by explicitly modeling particle diffusion, topological fusion and fission, and structural constraints (bonds, angles, repulsion). The system is initialized using real skeletonized network data, and reactions are governed by spatial proximity and graph-based logic rules. The output captures both particle-level and topology-level observables.

3.1.1 Particle Trajectories and 3D Structure Files

At every observation step, the 3D position and particle type of each node are recorded. Each node begins as mito_node_1 and may transition to mito_node_2 or mito_node_3 depending on its local connectivity (fig 3.1.). The .csv position logs offer an accessible numeric representation, while .pdb and .psf files provide structured 3D geometry and bond connectivity for external rendering. The resulting PDB-PSF pairs can be loaded into VMD to reconstruct mitochondrial filament arrangements with atomic-style precision, showing the continuous spatial deformation of the network over time. These spatial outputs are essential for visual validation and qualitative interpretation of fusion-fission cycles.

doi:10.6342/NTU202503815

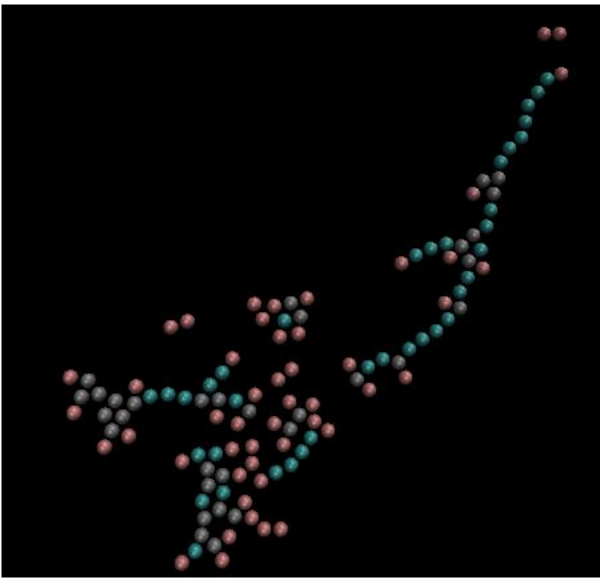


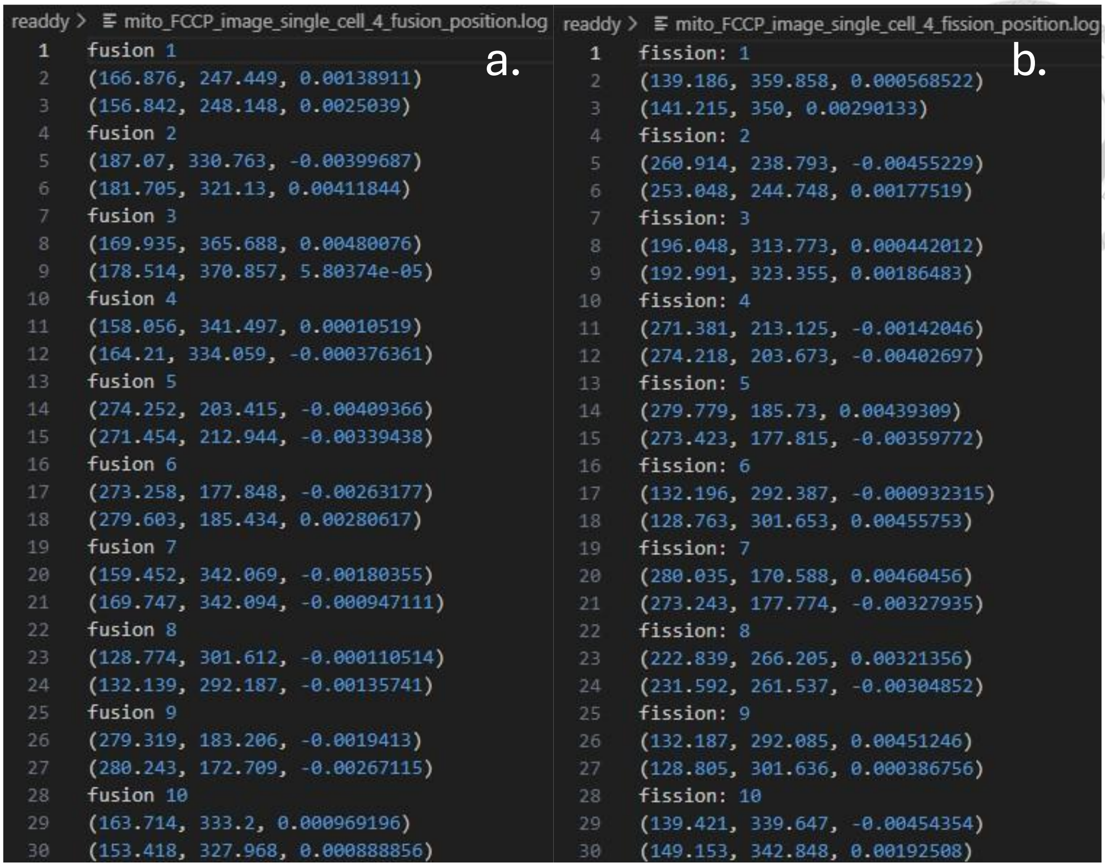
Figure 3.1. 3D rendering of mitochondrial network structure

Each node's spatial position and identity (e.g., mito_node_1, mito_node_2, mito_node_3) are recorded at each observation step and rendered using VMD. The color-coded atoms represent different node types, with connectivity defined by bond information in the PSF file. This high-resolution visualization captures the evolving topology and spatial deformation of mitochondrial filaments, allowing qualitative validation of fusion–fission dynamics modeled in the simulation.

3.1.2 Fusion and Fission Event Logging

Topological remodeling events are logged explicitly, with fusion reactions triggered by local degree constraints—such as two mito_node_1 particles converting into mito_node_2—and fission occurring via selective removal of internal (non-terminal) edges based on a topology-aware dissociation rate function. Each event includes the 3D coordinates of the affected particles, allowing reconstruction of the event landscape in physical space. This spatial logging reveals not only the frequency of reactions but also their localization and spatial distribution, which is crucial for identifying structurally sensitive zones or edge cases in the network dynamics.

doi:10.6342/NTU202503815



readdy > mito_FCCP_image_single_cell_4_fusion_position.log

```
fusion 1
(166.876, 247.449, 0.00138911)
(156.842, 248.148, 0.0025039)
fusion 2
(187.07, 330.763, -0.00399687)
(181.705, 321.13, 0.00411844)
fusion 3
(169.935, 365.688, 0.00480076)
(178.514, 370.857, 5.80374e-05)
fusion 4
(158.056, 341.497, 0.00010519)
(164.21, 334.059, -0.000376361)
fusion 5
(274.252, 203.415, -0.00409366)
(271.454, 212.944, -0.00339438)
fusion 6
(273.258, 177.848, -0.00263177)
(279.603, 185.434, 0.00280617)
fusion 7
(159.452, 342.069, -0.00180355)
(169.747, 342.094, -0.000947111)
fusion 8
(128.774, 301.612, -0.000110514)
(132.139, 292.187, -0.00135741)
fusion 9
(279.319, 183.206, -0.0019413)
(280.243, 172.709, -0.00267115)
fusion 10
(163.714, 333.2, 0.000969196)
(153.418, 327.968, 0.000888856)
```

readdy > mito_FCCP_image_single_cell_4_fission_position.log

```
fission: 1
(139.186, 359.858, 0.000568522)
(141.215, 350, 0.00290133)
fission: 2
(260.914, 238.793, -0.00455229)
(253.048, 244.748, 0.00177519)
fission: 3
(196.048, 313.773, 0.000442012)
(192.991, 323.355, 0.00186483)
fission: 4
(271.381, 213.125, -0.00142046)
(274.218, 203.673, -0.00402697)
fission: 5
(279.779, 185.73, 0.00439309)
(273.423, 177.815, -0.00359772)
fission: 6
(132.196, 292.387, -0.000932315)
(128.763, 301.653, 0.00455753)
fission: 7
(280.035, 170.588, 0.00460456)
(273.243, 177.774, -0.00327935)
fission: 8
(222.839, 266.205, 0.00321356)
(231.592, 261.537, -0.00304852)
fission: 9
(132.187, 292.085, 0.00451246)
(128.805, 301.636, 0.000386756)
fission: 10
(139.421, 339.647, -0.00454354)
(149.153, 342.848, 0.00192508)
```

Figure 3.2. Spatial logging of mitochondrial network remodeling events during simulation.

(a) Fusion events and (b) fission events are recorded in separate .log files, each entry capturing the 3D coordinates of the involved nodes. Fusion typically involves the merging of two mito_node_1 particles into a higher-order node (e.g., mito_node_2), while fission reflects the disassembly of internal bonds leading to node separation. The logs enable precise reconstruction of dynamic topological transitions, offering spatial context for each event and supporting downstream visualization or clustering analyses to identify preferential sites of remodeling activity.

3.1.3 Temporal Degree Probability Tracking

Node degrees, interpreted as the number of neighbors in the topology graph, are dynamically tracked and binned into degree-1, -2, or -3 categories. These degrees represent linear endpoints, internal chain particles, and branched junctions, respectively. By calculating the frequency of each degree class at every time point, we quantify the shift in network complexity as the simulation evolves. For instance, a monotonic increase in degree-1 nodes signifies fragmentation, while a sustained presence of degree-3 nodes would indicate retained branching. These probabilities are exported to .txt files for downstream comparison and plotted to show time-resolved topological simplification.

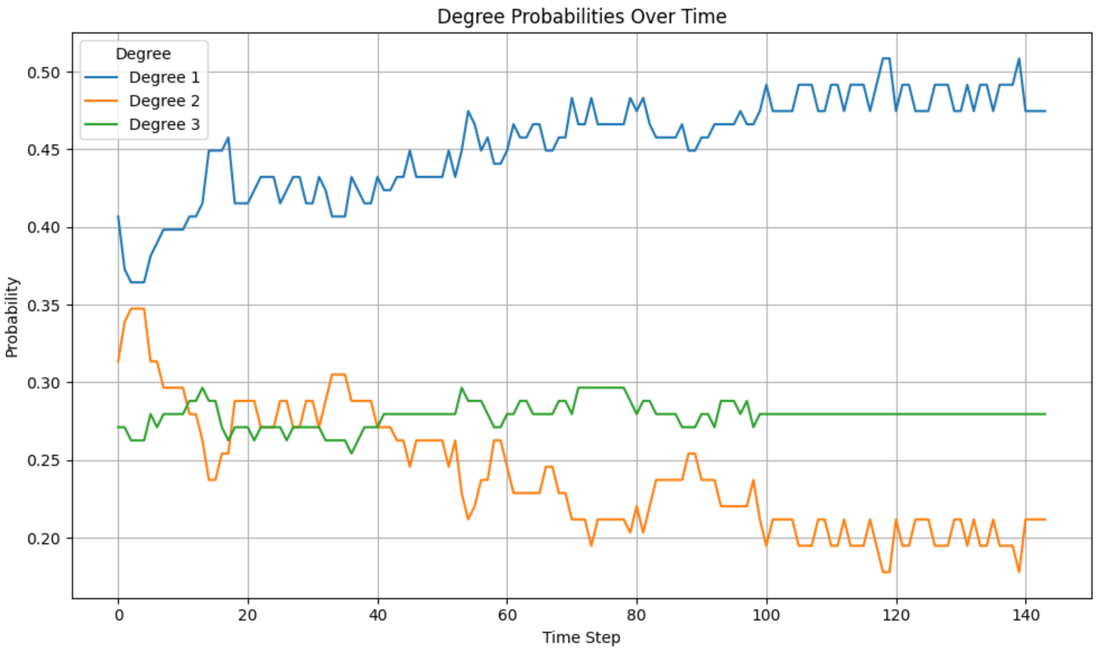


Figure 3.3. Time evolution of topological complexity in the mitochondrial network.

The plot shows the probability of nodes possessing degree 1 (blue), degree 2 (orange), and degree 3 (green) at each simulation time step. Degree-1 nodes correspond to terminal ends, degree-2 to internal linear segments, and degree-3 to branching junctions. Over time, an increasing trend in degree-1 probability and decreasing degree-2 frequency suggests progressive fragmentation. The relatively stable degree-3 proportion indicates that while linearity dominates, occasional branching structures persist. This temporal tracking of node degree distribution provides quantitative insight into mitochondrial network simplification dynamics.

3.1.4 Average Topology Length per Time Step

To further quantify structural degradation, the average number of bonds per connected topology (i.e., average polymer length) is computed at each time step. This

doi:10.6342/NTU202503815

metric accounts for both fragmentation and internal connectivity. The simulation reveals a consistent downward trend in this quantity, suggesting that large, complex filaments are progressively replaced by shorter segments or isolated particles. Since this measure is topology-based rather than particle-based, it serves as a higher-order indicator of network disintegration.

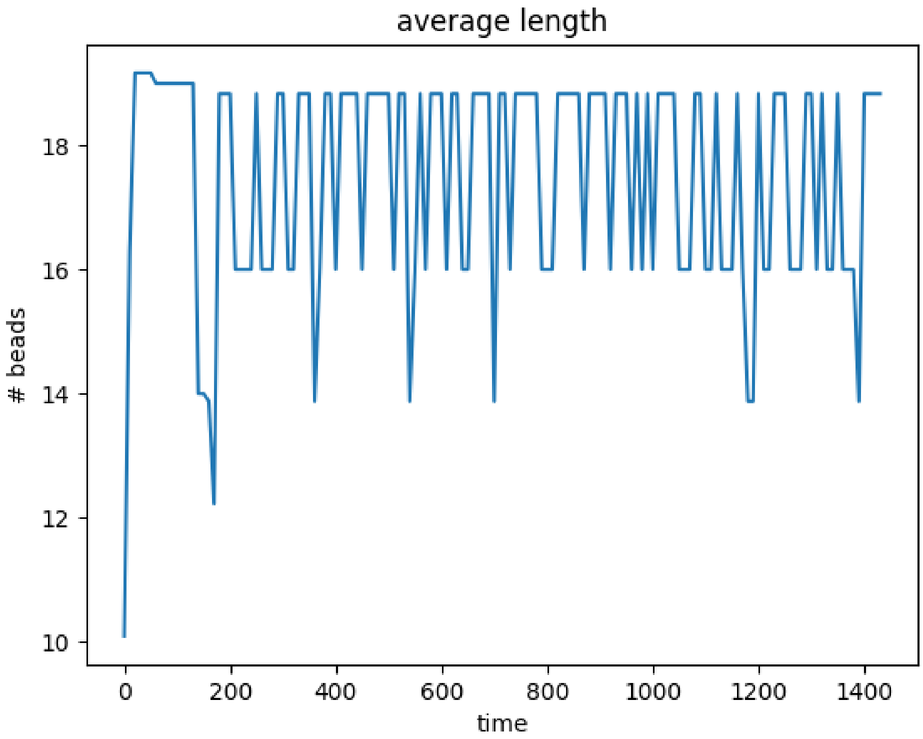


Figure 3.4. Temporal evolution of average mitochondrial network segment length.

The plot shows the average number of beads per connected topology (i.e., polymer segment) over time. Despite fluctuations, a generally stable but lower-bound average length is maintained, with intermittent drops suggesting transient fragmentation events. This measure reflects the evolving integrity of the mitochondrial network structure, where shorter polymers indicate degradation and loss of extended connectivity. As this metric aggregates structural information at the topology level, it provides a high-level view of network simplification trends.

3.1.5 Final Degree Distribution

doi:10.6342/NTU202503815

At the conclusion of the simulation, the system's topological state is analyzed by computing the full histogram of node degrees and normalizing it to obtain a degree probability distribution. This is visualized in log–log space to capture both frequent and rare degree values. The dominance of degree-1 nodes confirms the collapse of the network into disconnected or linear elements. Importantly, this distribution is the basis for quantitative comparison with experimental image-derived networks.

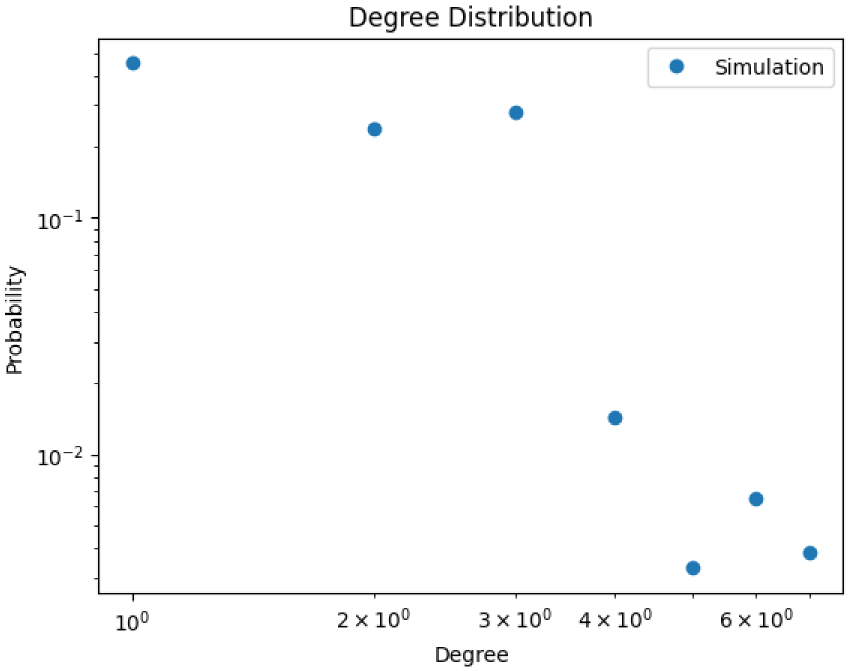


Figure 3.5. Final degree distribution of the simulated mitochondrial network.

The histogram shows the normalized probability of each node degree at the end of the simulation, plotted on a log–log scale to emphasize both common and rare topological states. Degree-1 nodes dominate the distribution, indicating a fragmentation-dominated regime with a predominance of terminal elements. The sharp decay in higher-degree nodes reflects the loss of complex junctions and branching points, consistent with network disintegration under FCCP treatment. This distribution serves as a benchmark for comparing simulation outcomes with experimental imaging data.

3.1.6 Fragment Size Distribution

doi:10.6342/NTU202503815

Each fragment (connected component) is characterized by its size—the number of particles it contains. The simulation computes the frequency of each fragment size and displays it as a distribution. A broad distribution, particularly one with a heavy tail, suggests that while many small fragments exist, some larger structures remain. This analysis reveals the system's heterogeneity: complete fragmentation into isolated monomers results in a narrow peak, while mixed-size fragments suggest partial preservation of higher-order structure.

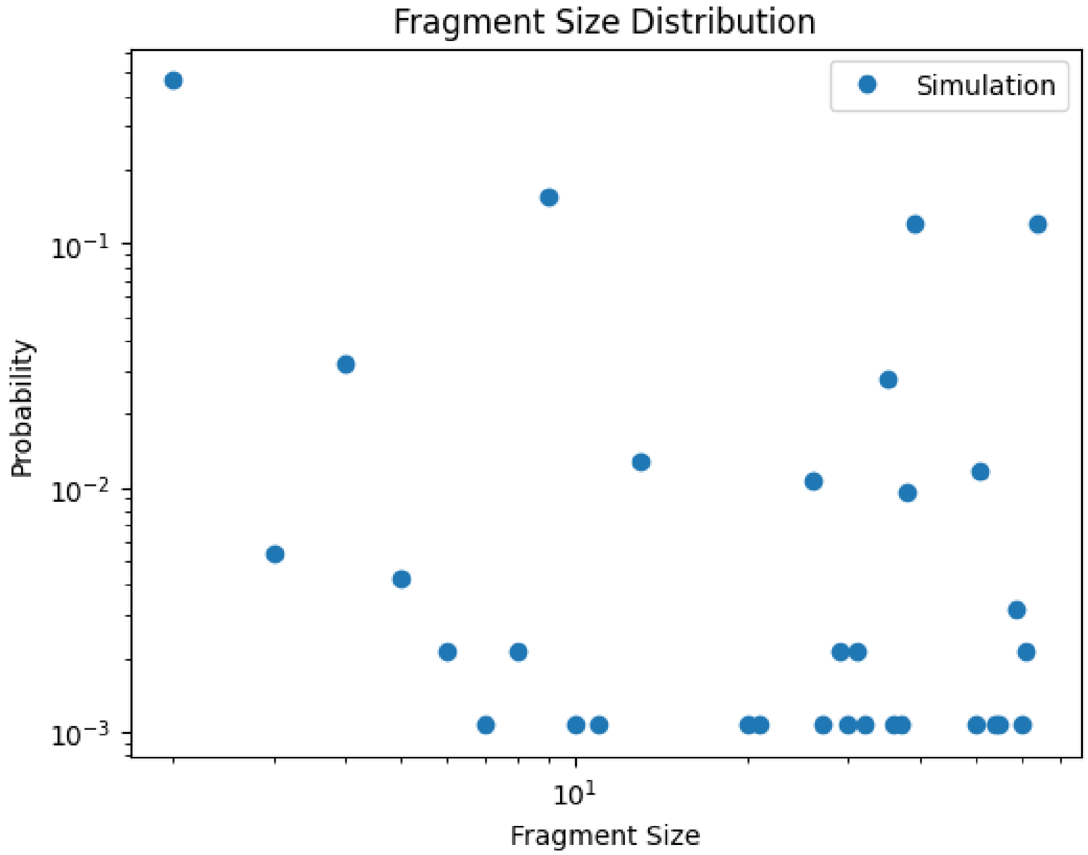


Figure 3.6. Fragment size distribution of the simulated mitochondrial network at the final time point.

The probability distribution of fragment sizes—measured as the number of particles per connected component—is shown on a log–log scale. The presence of a wide range of fragment sizes, including both small and moderately large clusters, indicates a heterogeneous fragmentation process. The heavy-tailed nature of the distribution implies that while many filaments break into small pieces, some larger substructures persist. This pattern supports a model of partial network collapse with residual

doi:10.6342/NTU202503815

topological complexity, potentially reflecting biologically relevant subnetwork stabilization mechanisms.

3.1.7 Fragment Count Over Time

To assess how the network disassembles over time, the simulation tracks the number of fragments at each time point. This directly reflects the balance between fusion and fission events. The increasing number of fragments over time confirms a fission-dominated regime and serves as a macroscopic signature of mitochondrial network collapse. It also helps identify transient equilibrium phases, where the fragment count temporarily plateaus before resuming its rise.

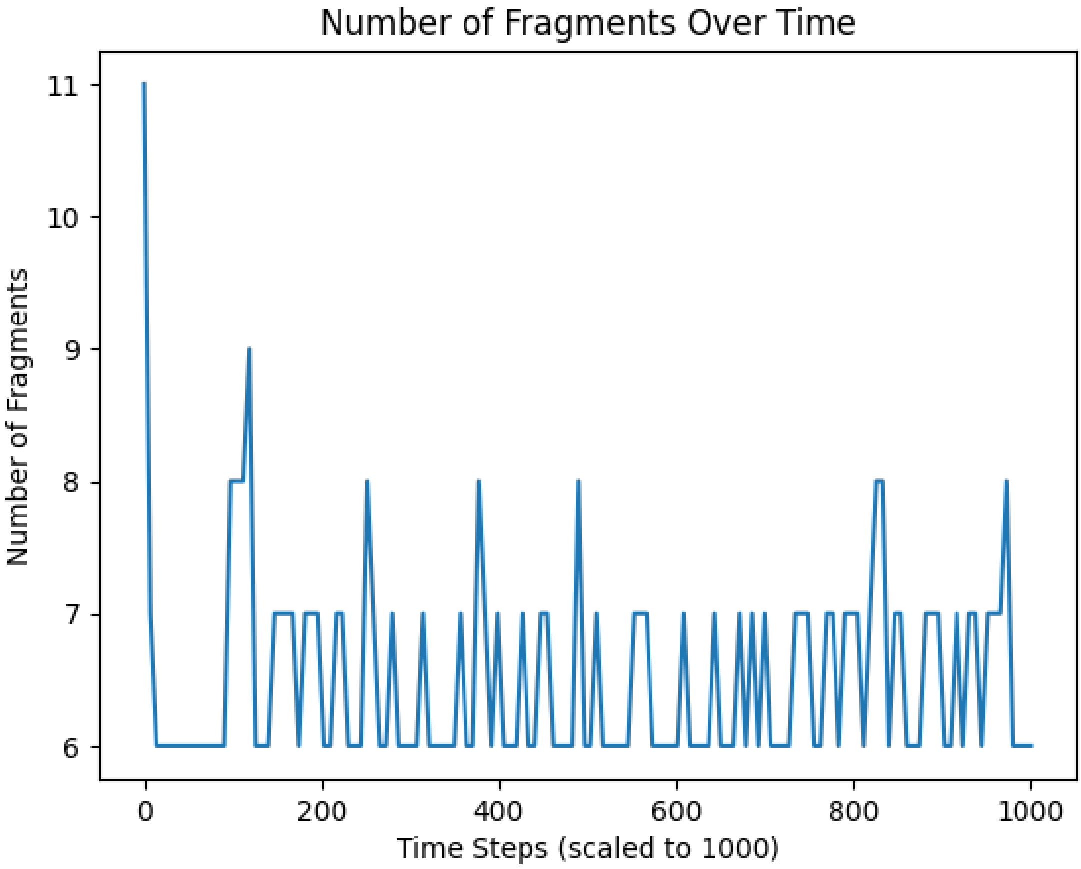


Figure 3.7. Temporal evolution of fragment count during the mitochondrial network simulation.

doi:10.6342/NTU202503815

The number of disconnected topological fragments is plotted against simulation time steps (scaled to 1000). An initial drop followed by persistent fluctuations around a higher baseline indicates progressive fragmentation driven by fission events. Despite transient dips, the absence of a consistent downward trend suggests a lack of sustained fusion activity. This behavior is consistent with a system undergoing structural disintegration, and it offers a coarse-grained view of topological breakdown dynamics complementary to degree and size-based metrics.

3.1.8 Reaction Count Dynamics

In parallel, all reactions—both spatial and structural—are monitored and recorded in real-time. This includes fusion_1, fusion_2, and any internally triggered structural transitions. By parsing reaction_counts.log, the system reconstructs the kinetic landscape of the simulation. Changes in reaction rates over time provide insight into the system's phase behavior, such as initial bursts of activity during early reconfiguration, followed by a steady state where fewer eligible reactions remain due to increasing fragmentation.

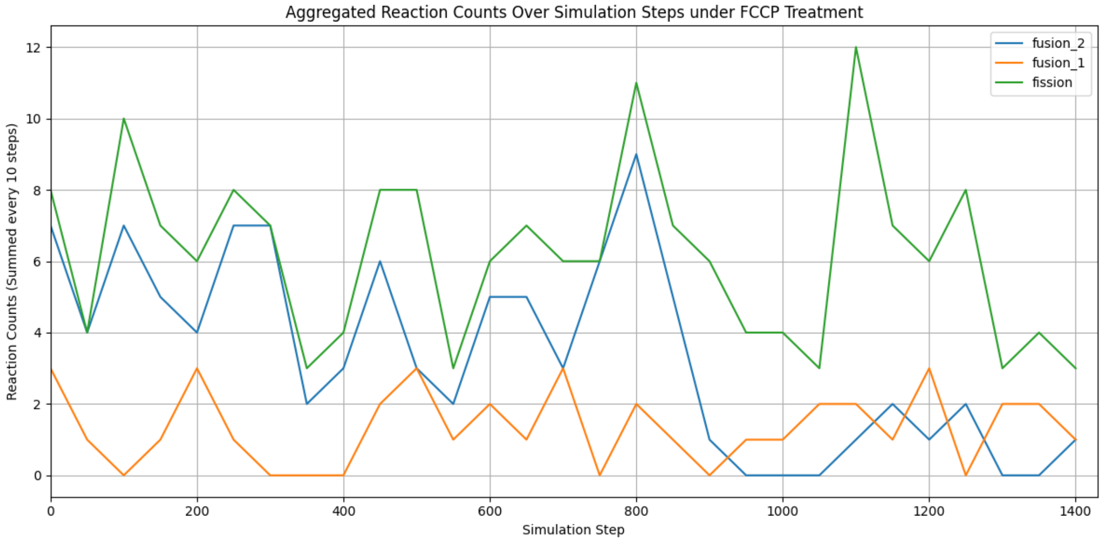


Figure 3.8. Temporal evolution of reaction events

The plot depicts the aggregated counts of fission, fusion_1, and fusion_2 reactions across the simulation timeline, binned every 50 steps for clarity. Fission reactions

doi:10.6342/NTU202503815

(blue) dominate throughout the simulation, reflecting a fragmentation-prone regime. In contrast, fusion_1 (orange) and fusion_2 (green) events occur sporadically and at lower frequencies. This kinetic profile underscores the imbalance between network-building and network-breaking processes, which contributes to the overall structural degradation observed in topological metrics. The declining frequency of all events toward later stages reflects a reduction in eligible reaction sites as the network disintegrates.

3.2 Multi-Run Statistical Averaging — Reproducibility and Error Quantification

To ensure reproducibility and account for stochastic variation in the simulation outcomes, we conducted 100 independent simulations using a batch pipeline (Multi-run Statistical Averaging.py). Each replicate uses identical initial network conditions and parameters and produces its own trajectory (h5), XYZ file, and degree distribution log (degree_probabilities_rep*txt).

3.2.1 Degree Probability Averaging Across Replicates

All replicate logs are aggregated and processed to compute the mean degree probability for each class (degree 1, 2, 3) at each time point. The standard deviation across replicates is also calculated, capturing the run-to-run variability of topological evolution. These statistics are visualized as mean ±1σ bands using 01_mito_FCCP_image_error_analyzepy.py. This analysis provides an error-aware backbone of system dynamics and serves as a reference point when comparing against single-run results or experimental measurements. Notably, the low variability across runs indicates consistent simulation behavior and a well-defined dynamic trajectory.

doi:10.6342/NTU202503815

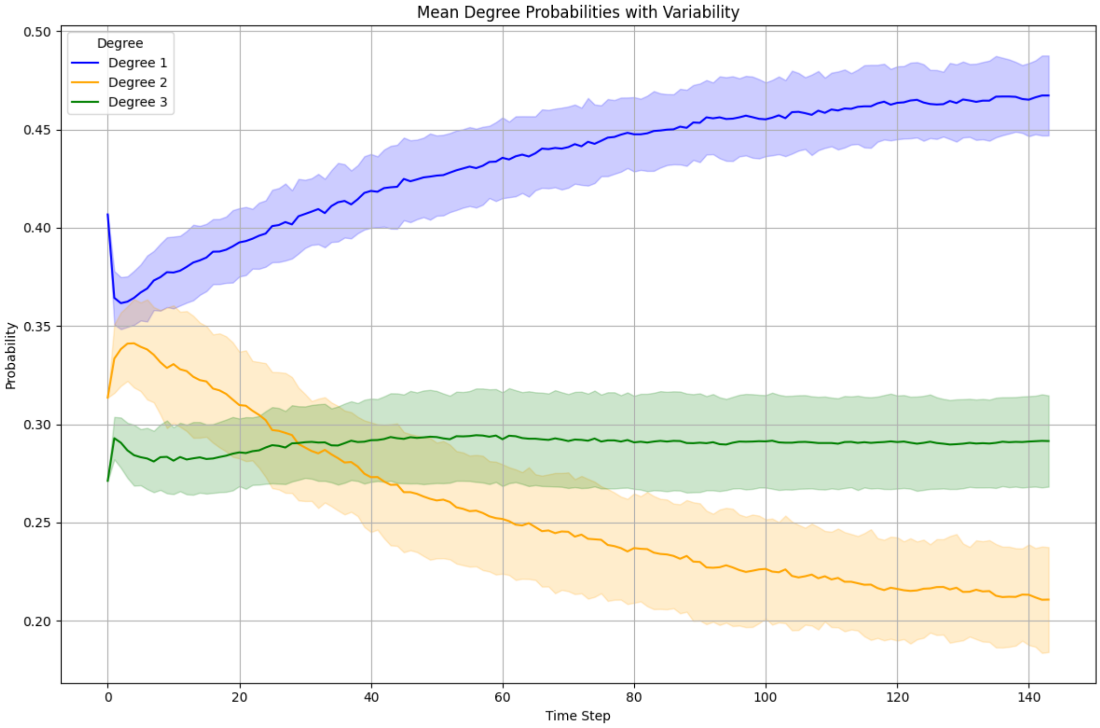


Figure 3.9. Mean degree probabilities over time with replicate variability.

The plot shows the mean probability of each topological degree class—degree 1 (blue), degree 2 (orange), and degree 3 (green)—across multiple simulation replicates. Shaded regions represent ±1 standard deviation at each time step, quantifying the variability of network evolution. A clear upward trend in degree-1 nodes indicates persistent fragmentation, while the decline in degree-2 nodes reflects a reduction in internal filament segments. The relatively stable profile of degree-3 nodes suggests a residual presence of branching structures. The narrow error bands indicate consistent dynamics across replicates, supporting the reliability of simulation outcomes.

3.2.2 Quantitative Comparison of Experimental Image Data

To evaluate how well the simulation captures real mitochondrial dynamics, we directly compare simulated degree trajectories with those extracted from time-series microscopy images. The image data, processed through skan, yields skeletonized networks and corresponding degree logs. For quantitative comparison, per-frame absolute errors are computed for each degree class and averaged to obtain a mean

doi:10.6342/NTU202503815

absolute error (MAE) for every simulated–experimental cell pair. This matching is automated using the 01_mito_FCCP_image_error_analyze.py script. Figure 3.10 illustrates the validation process under FCCP treatment. Panel (c) shows the distribution of MAEs from 100 simulation runs across all individual cells. Each black dot represents the MAE for a single simulated cell, and the red star denotes the simulation with the smallest error, selected as the best-fit case. Based on this selection, panel (a) presents the temporal evolution of degree probabilities (mean ± SD) for this best-fit simulation, demonstrating close agreement with the experimental dynamics. Panel (b) focuses on this same single cell, providing a detailed view of its degree distributions throughout the simulation. These results indicate that the simulation framework accurately reproduces mitochondrial network dynamics for the chosen best-fit replicate, providing a robust basis for interpreting treatment-induced deviations in subsequent analyses.

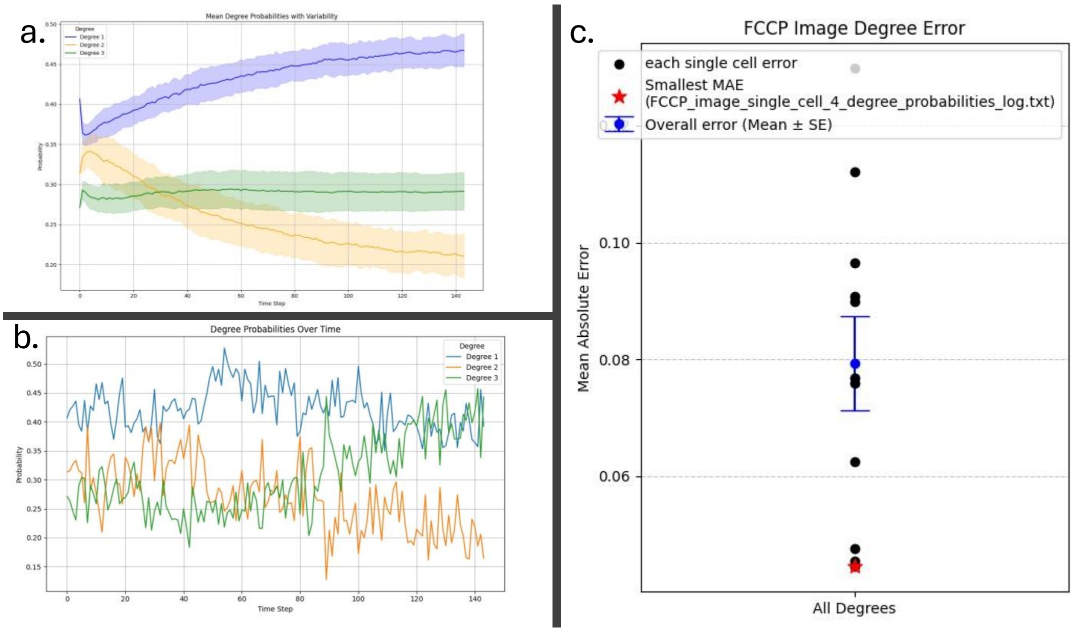


Figure 3.10. Validation of simulation against experimental image-derived dynamics under FCCP treatment.

(a) Temporal evolution of degree probabilities (mean ± SD) for this best-fit replicate, showing close alignment with experimental trends.

(b) Degree distribution of the same single cell across the simulation, illustrating how the best-fit replicate captures mitochondrial network dynamics under FCCP treatment.

doi:10.6342/NTU202503815

(c) Mean absolute errors (MAEs) from 100 simulations compared with experimental degree distributions; each black dot represents an individual simulated–experimental cell pair, and the red star marks the best-fit simulation with the smallest error.

3.3 Results of Simulation–Experiment Validation

This section presents the outcomes of the quantitative comparisons introduced in Section 3.2. For each treatment condition and cell type, the temporal evolution of degree distributions from simulations is evaluated against experimental image-derived data. Both best-fit single-cell replicates, and aggregate analyses are provided to illustrate model performance and treatment-specific deviations.

Before presenting the validation results, it is important to clarify the rationale behind using the mean absolute error (MAE) as our evaluation metric. In the ReaDDy2 simulations, all parameters except the fission and fusion probabilities were kept constant. These two parameters were iteratively adjusted for each single cell to minimize the difference between simulated and experimental degree distributions. The parameter set that yielded the lowest MAE for a cell was recorded as its “best-fit” configuration. For each treatment, these best-fit parameters were then averaged and reapplied to evaluate whether a common parameter set could robustly reproduce the dynamics of all cells in that group. MAE distributions within a treatment provide two key pieces of information: (i) whether the resulting errors are highly dispersed—indicating large variability between cells—and (ii) whether the overall MAE values remain low. In general, an MAE below approximately 0.1–0.15 for degree probabilities is considered to represent good agreement, whereas values approaching 0.3 or higher imply a substantial mismatch that may require additional model assumptions or mechanisms. Based on these evaluations, we focus on analyzing the best-fit simulations as representative cases for each treatment to explore possible network behaviors under that condition. Presentation of the worst-fit cases was not emphasized, as our primary aim was to identify the range of dynamics that can be captured when the model performs optimally.

doi:10.6342/NTU202503815

To present the results systematically, we first summarize the fixed ReaDDy2 parameters all simulations in a table 3.1. These baseline values remain constant across conditions, this table is shown once for reference, and for each treatment only the adjusted fusion/fission probabilities are noted when introduced.

Table 3.1. Fixed ReaDDy2 simulation parameters.

Values define the simulation box geometry, diffusion properties, interaction forces, angular constraints, and detection probability which is extremely high to insure can detect fusion in every step. These parameters are held constants for all conditions; only fusion probabilities, capture radii for spatial fusion reactions and fission probabilities are adjusted for each treatment.

parameter	**value**	**description (concise)**
box_size	[1000, 1000, 0.01]	simulation box dimensions (x, y, z)
origin	[-500, -500, -0.005]	box center coordinates
force_constant_box	100.0	reflective boundary strength
diffusion_constant	0.02	particle diffusion constant
force_constant_repulsion	10.0	soft-core repulsion strength
interaction_distance	10.0	repulsion interaction cutoff distance
force_constant_bond	100.0	harmonic bond force constant
length_bond	10	equilibrium bond length
force_constant_angle	10.0	angle potential force constant
two_degree_angle	π	target angle for degree-2 connections
three_degree_angle	π / 3	target angle for degree-3 junctions
detection_probability	4	rate factor for fission site detection

We then present the average degree distributions over time, aggregated across all cells within each treatment, to provide an overview of the typical temporal evolution of mitochondrial network topology. Following this, error analysis plots (such as Fig. 3.10) are used to assess the overall fit quality between simulations and experimental data, considering both group-level variability and individual best-fit performance. Finally, for each treatment, we examine the network behaviors captured by the best-fit simulations through detailed time-course visualizations and structural analyses, focusing on how well these simulations reproduce key experimental trends.

1. Ins-1 Control Network Condition (n = 15) (baseline value).

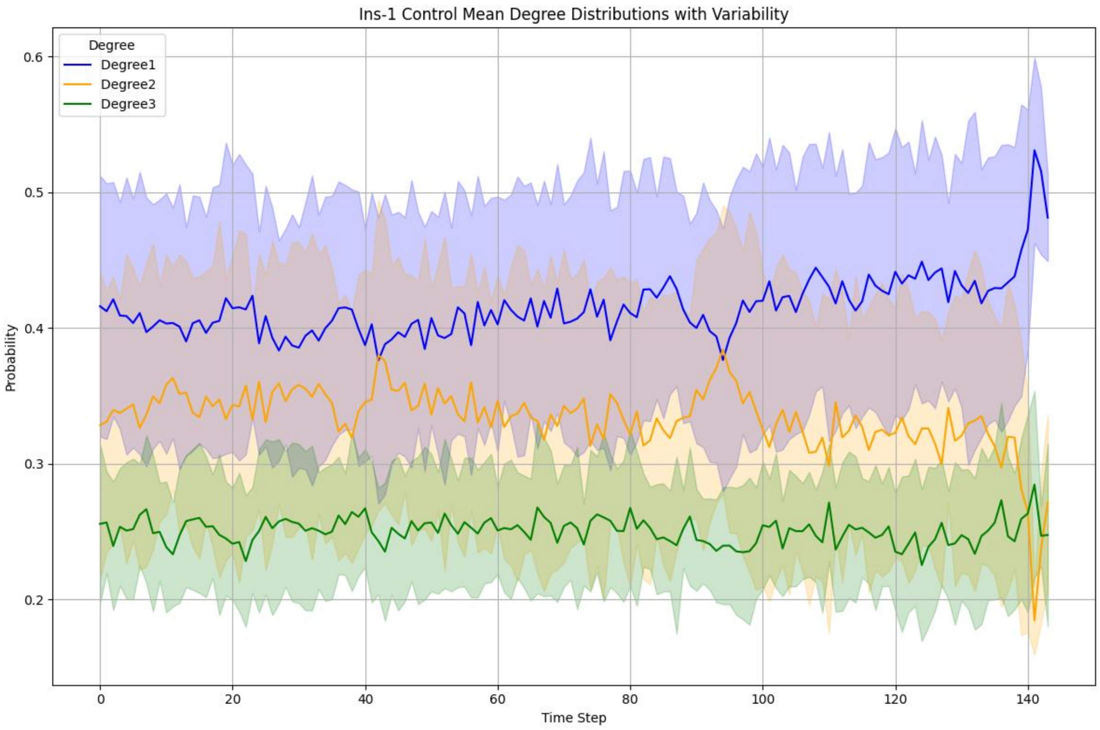


Figure 3.11. Temporal evolution of mean degree distributions for the Ins-1 control condition.

Figure 3.11. illustrates the temporal evolution of the mean degree distributions for the Ins-1 control group, aggregated across all single-cell simulations. Under

doi:10.6342/NTU202503815

control conditions, the mitochondrial networks maintain a relatively balanced topology over time. Degree 1 nodes dominate, with probabilities fluctuating around 0.40–0.45, indicating the presence of numerous endpoints and short branches. Degree 2 nodes remain moderately prevalent, maintaining values near 0.30–0.35, consistent with linear chain segments. Degree 3 nodes are less frequent, with probabilities around 0.20–0.25, reflecting a limited but stable number of branching points. The variability bands indicate moderate cell-to-cell heterogeneity, but no major shifts or instabilities are observed over the 140 simulation steps. These results suggest that, in the absence of treatment, the Ins-1 mitochondrial networks exhibit a stable, moderately connected architecture with limited branching complexity.

x

Table 3.2. Adjustable ReaDDy2 parameters for Ins-1 Control.

parameter	value	description (concise)
Fusion_probability_1	0.3	tip-to-tip fusion probability
Fusion_probability_2	0.4	tip-to-side fusion probability
Fission_base_probability	0.15	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	15	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

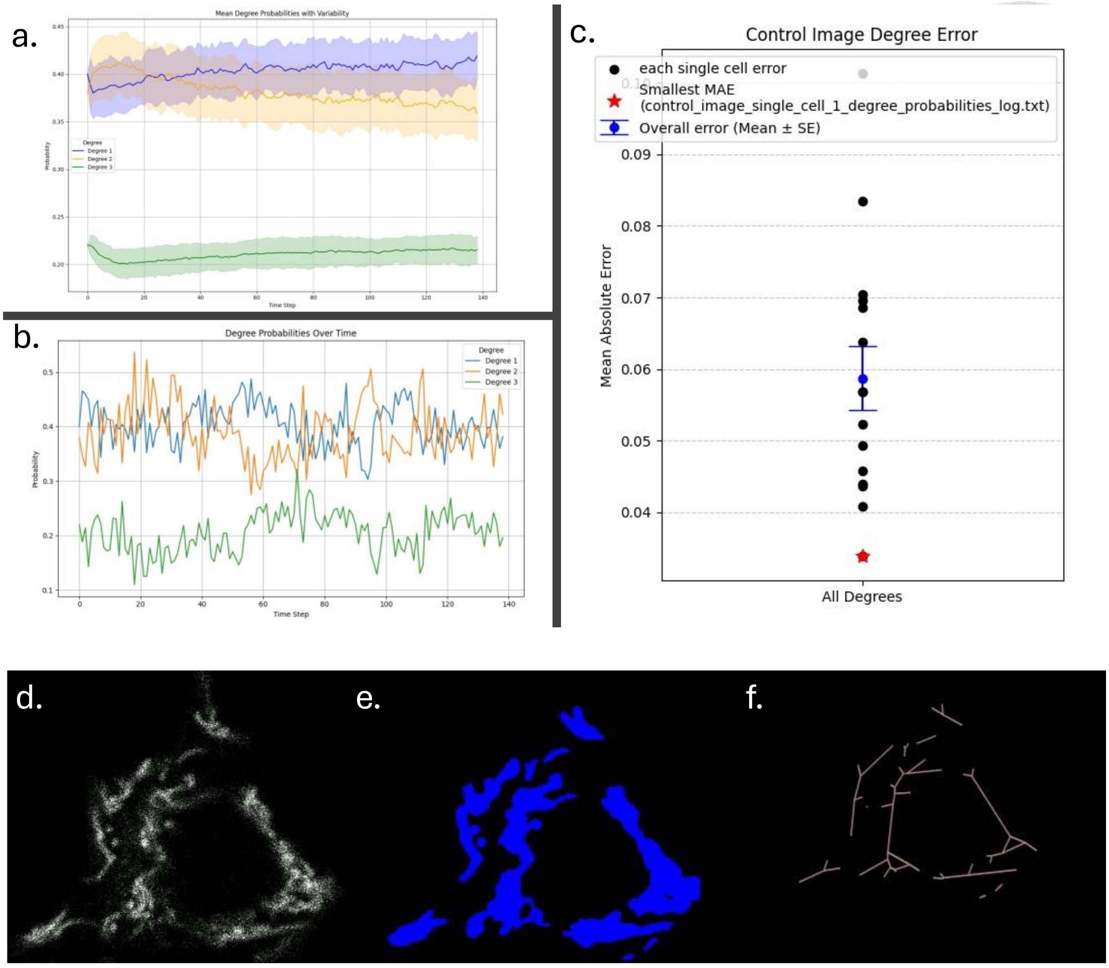


Figure 3.12 . Control Condition Analysis for Ins-1 Cells

Panels (a–c) present quantitative comparisons between simulated and experimental mitochondrial network properties under control conditions.

(a) Time-averaged degree probabilities (± standard deviation) for the best-fit simulation replicate.

(b) Degree probabilities from the corresponding experimental image that the best-fit simulation was matched against.

(c) Mean absolute error (MAE) between experimental data and 100 simulation replicates. The red star marks the simulation with the lowest MAE, selected as the best fit; the blue point and error bar indicate the overall mean ± standard error (SE) across all replicates.

(d) Raw confocal fluorescence image of a representative untreated Ins-1 cells.

doi:10.6342/NTU202503815

(e) Corresponding segmentation mask.

(f) Skeletonized mitochondrial network used for graph extraction and degree quantification.

Under control conditions, the mitochondrial network exhibits relatively stable structural dynamics over time. As shown in Fig. 3.12(a), the best-fit simulation demonstrates a dominant presence of degree 1 nodes (terminal ends), indicating a fragmented but steady morphology. Degree 2 nodes (linear segments) show a slight decline, while degree 3 nodes (branch points) gradually increase, reflecting mild network remodeling. The corresponding experimental degree probabilities used to evaluate the best-fit case are shown in Fig. 3.12(b).

This evolving pattern suggests a slow but persistent incorporation of linear segments into branched structures, consistent with physiological turnover rather than stress-induced restructuring. The network maintains a quasi-steady state characterized by balanced fusion and fission activity, without signs of extreme fragmentation or hyper fusion.

As quantified in Fig. 3.12(c), the best-fit simulation achieves the lowest MAE among 100 replicates, while the overall mean MAE remains below 0.1, indicating a strong agreement between the model and experimental data. This control condition provides a reliable baseline for assessing the effects of pharmacological perturbations in subsequent sections.

doi:10.6342/NTU202503815

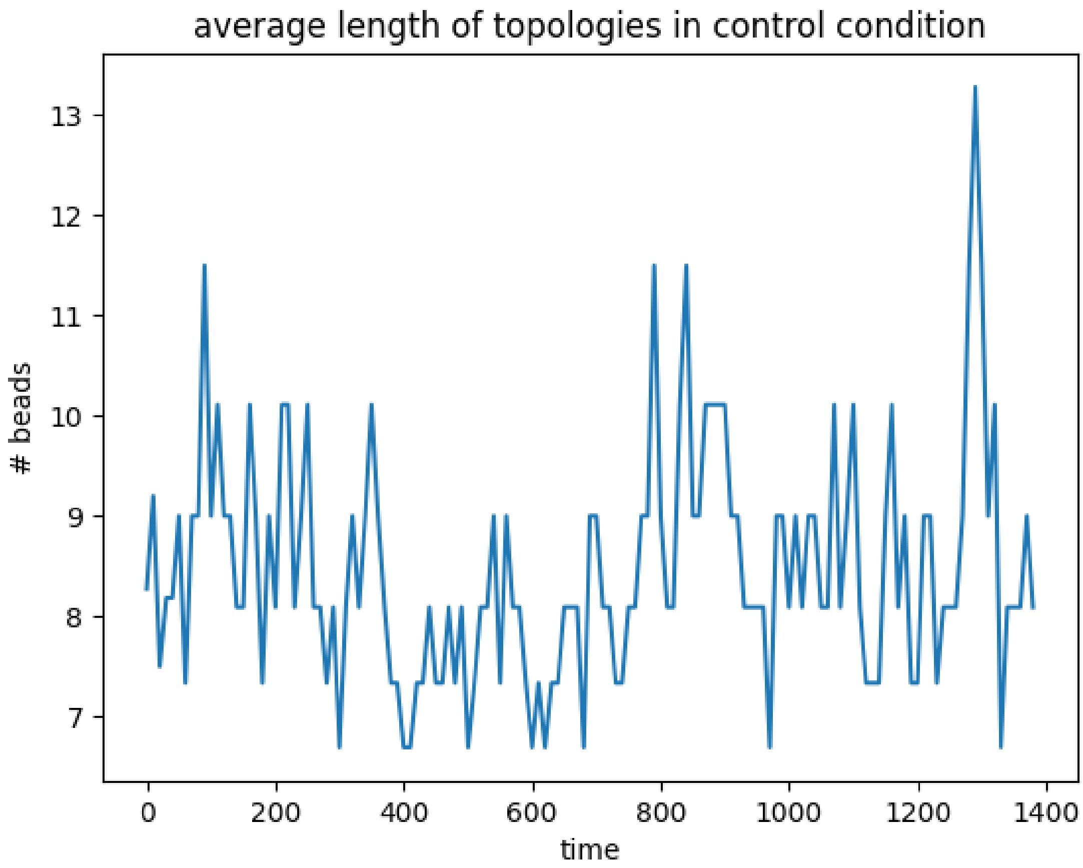


Figure 3.13. Temporal evolution of average fragment length of Ins-1 Cell Control. (measured by node count per topology) indicates continuous remodeling without strong directional trends.

Figure 3.13 remains relatively stable throughout the simulation, fluctuating between approximately 7 and 13 beads per structure. No consistent increasing or decreasing trend is observed, indicating a dynamic equilibrium between fusion and fission events. This variability reflects continuous remodeling activity characteristics of a physiologically balanced mitochondrial network. The preservation of average segment length over time suggests that the system is maintaining morphological homeostasis, with neither excessive fragmentation nor elongation dominating the network dynamics.

doi:10.6342/NTU202503815

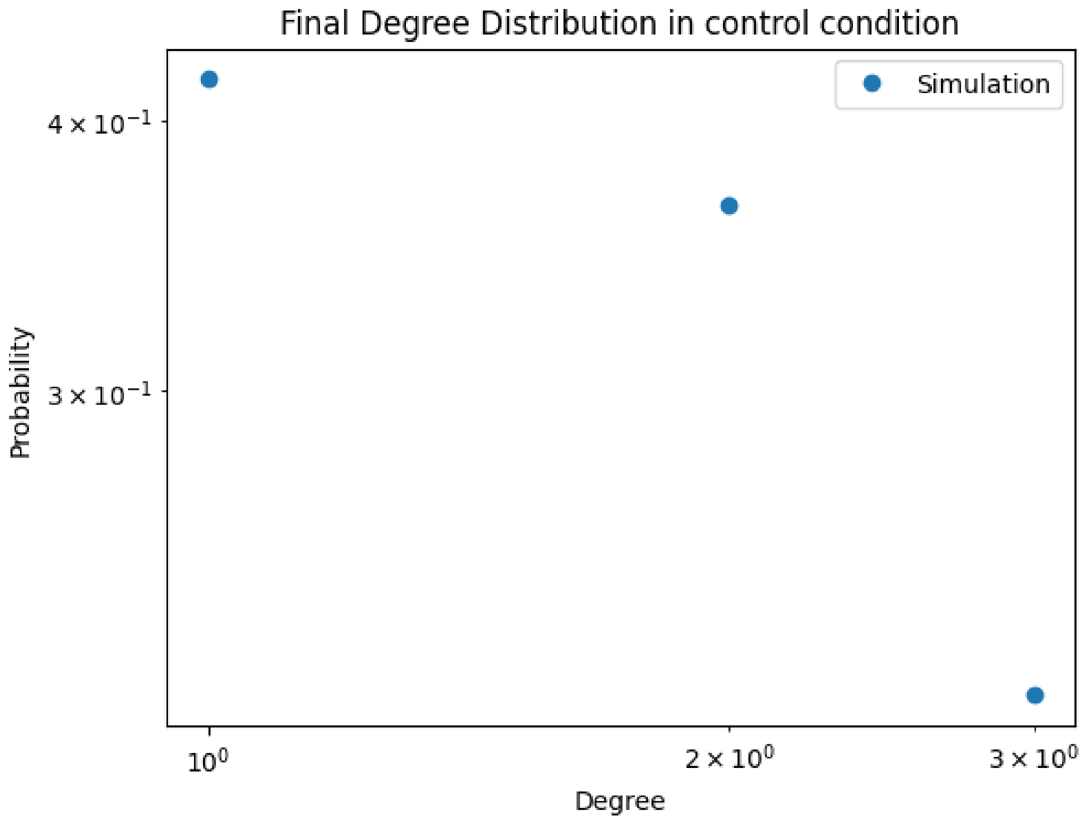


Figure 3.14. Final Degree distribution of Ins-1 Cell Control (log-log scale) reveals a dominant presence of degree-1 nodes, consistent with fragmented and linear structures.

Figure 3.14 is a predominance of low-degree nodes, with degree 1 and degree 2 being the most frequent. Degree 1 nodes (terminal ends) exhibit the highest probability, indicating that fragmented or end-point structures are common within the network. Degree 2 nodes (linear segments) are slightly less frequent, while degree 3 nodes (branch points) occur at lower probability, reflecting moderate network complexity. Nodes with degrees 4 or higher are rare, suggesting limited over-branching. This distribution consists of a homeostatic mitochondrial network characterized by a balance between fragmentation and fusion, supporting moderate connectivity without excessive branching.

doi:10.6342/NTU202503815

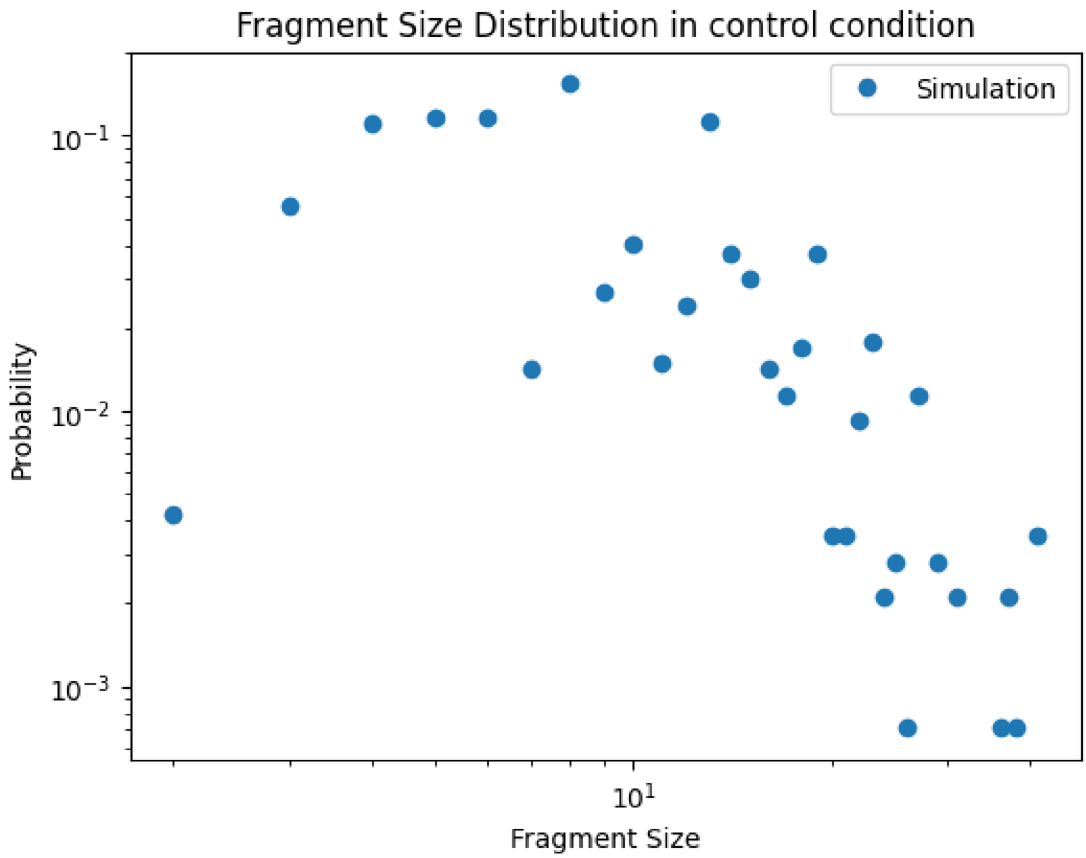


Figure 3.15. Fragment size distribution follows a heavy-tailed profile of Ins-1 Cell Control, with numerous small fragments and a few large clusters, reflecting network heterogeneity.

Figure 3.15 follows a broad-tailed, right-skewed profile. Most mitochondrial fragments are relatively small, but a wide range of larger structures is also present at lower probabilities. The distribution suggests that while fragmentation events are frequent, a subset of mitochondria remains connected as extended structures. This balance reflects the dynamic interplay between fission and fusion in a homeostatic network, allowing both isolated fragments and moderately large mitochondrial clusters to coexist. The presence of a long-tailed distribution is indicative of structural heterogeneity, characteristic of healthy mitochondrial populations.

doi:10.6342/NTU202503815

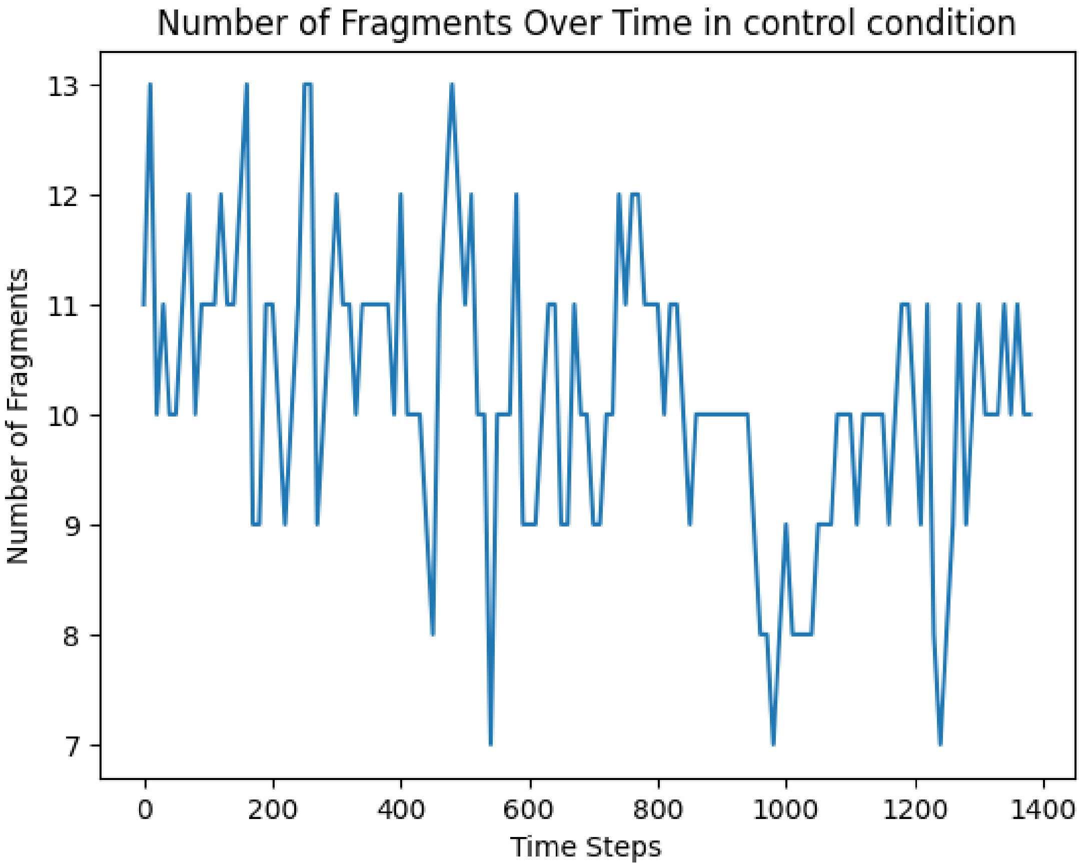


Figure 3.16. The number of fragments tracked over time of Ins-1 Cell Control shows fluctuations around a stable average, indicating a dynamic steady state.

Figure 3.16. remains relatively stable over time under control conditions, fluctuating around a mean of approximately 9 to 11 fragments. No consistent upward or downward trend is observed, indicating that the rates of fission and fusion are well balanced. The fluctuations reflect the stochastic nature of individual reaction events, but the overall fragment count remains within a narrow range. This stability in fragment numbers supports the notion of a homeostatic network, where mitochondrial remodeling occurs dynamically but without net accumulation or loss of structural units.

doi:10.6342/NTU202503815

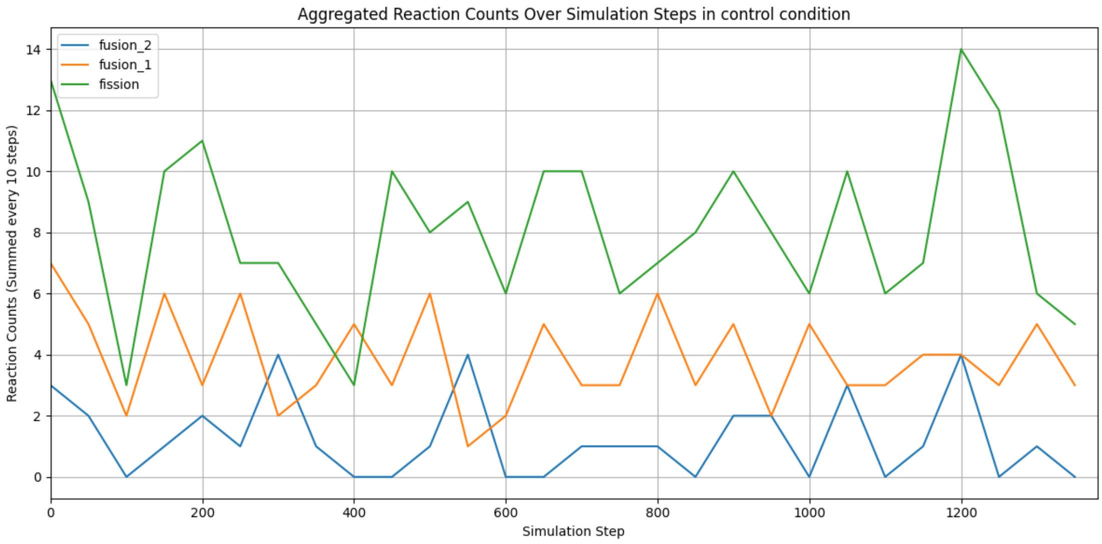


Figure 3.17. Aggregated reaction counts for fission, fusion_1, and fusion_2 events across the simulation of Ins-1 Cell Control. The relatively balanced rates reinforce the notion of a homeostatic regime with ongoing but non-disruptive restructuring.

Figure 3.17. Reaction dynamics remain active and balanced throughout the simulation. Fission events (green) occur frequently and stochastically, maintaining a moderate to high level across all simulation steps.

Fusion_1 events are consistently present and follow a frequency pattern like fission, though slightly lower in count. In contrast, fusion_2 events occur less frequently and display sporadic behavior with many intervals of inactivity. This asymmetry suggests that the primary fusion mechanism under homeostatic conditions is fusion_1, while fusion_2 plays a more limited role.

Overall, the interplay between fission and fusion reactions remains dynamically balanced, supporting the maintenance of a stable mitochondrial network. The persistent activity of all reaction types reflects ongoing remodeling typical of healthy mitochondrial populations.

doi:10.6342/NTU202503815

2. Ins-1 FCCP Network Condition (n =11)

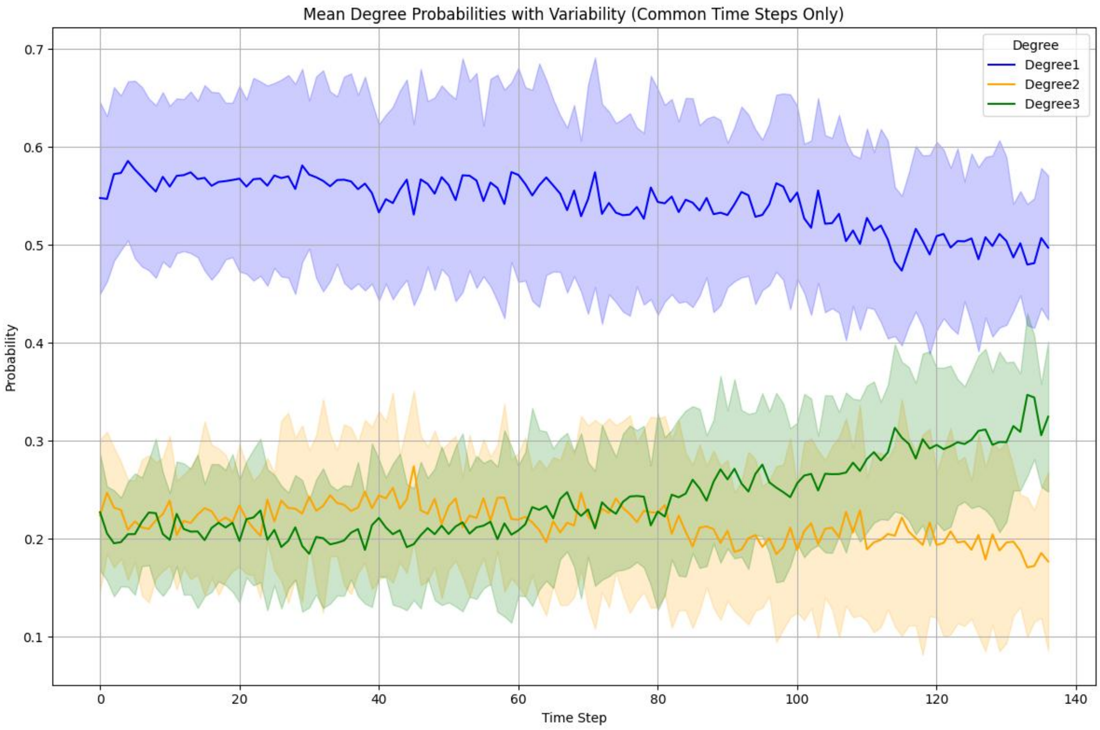


Figure 3.18. Temporal evolution of mean degree distributions for Ins-1 cells under FCCP treatment.

The plot shows the average degree probabilities (mean ± SD) across all single-cell simulations. Degree 1 nodes increase over time, indicating enhanced fragmentation. Degree 2 nodes decline steadily, while degree 3 nodes rise gradually, reflecting local branching amid overall network destabilization caused by FCCP-induced stress.

Figure 3.18 illustrates the temporal evolution of degree distributions in Ins-1 cells treated with FCCP, averaged across all single-cell simulations. The network exhibits a pronounced shift toward increased fragmentation over time. Degree 1 nodes (terminal ends) dominate throughout the simulation, maintaining probabilities above 0.5 and showing a slight upward trend, indicating a growing number of isolated or disconnected segments. In contrast, degree 2 nodes (linear segments) steadily decline, suggesting loss of continuity in tubule structures. Most notably, degree 3 nodes

doi:10.6342/NTU202503815

(branch points) show a gradual and continuous increase, reaching near 0.3 by the end of the time course. This trend suggests that, despite FCCP-induced fragmentation, some local branching persists—potentially due to compensatory fusion events.

These patterns reflect a network that undergoes progressive destabilization under FCCP, consistent with the drug's known role as a mitochondrial uncoupler that collapses membrane potential and biases dynamics toward fission. The overall variability also increases over time, suggesting a loss of structural uniformity among the simulated networks.

Table 3.3. Adjustable ReaDDy2 parameters for Ins-1 FCCP

parameter	**value**	**description (concise)**
Fusion_probability_1	2	tip-to-tip fusion probability
Fusion_probability_2	5	tip-to-side fusion probability
Fission_base_probability	0.1	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	20	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

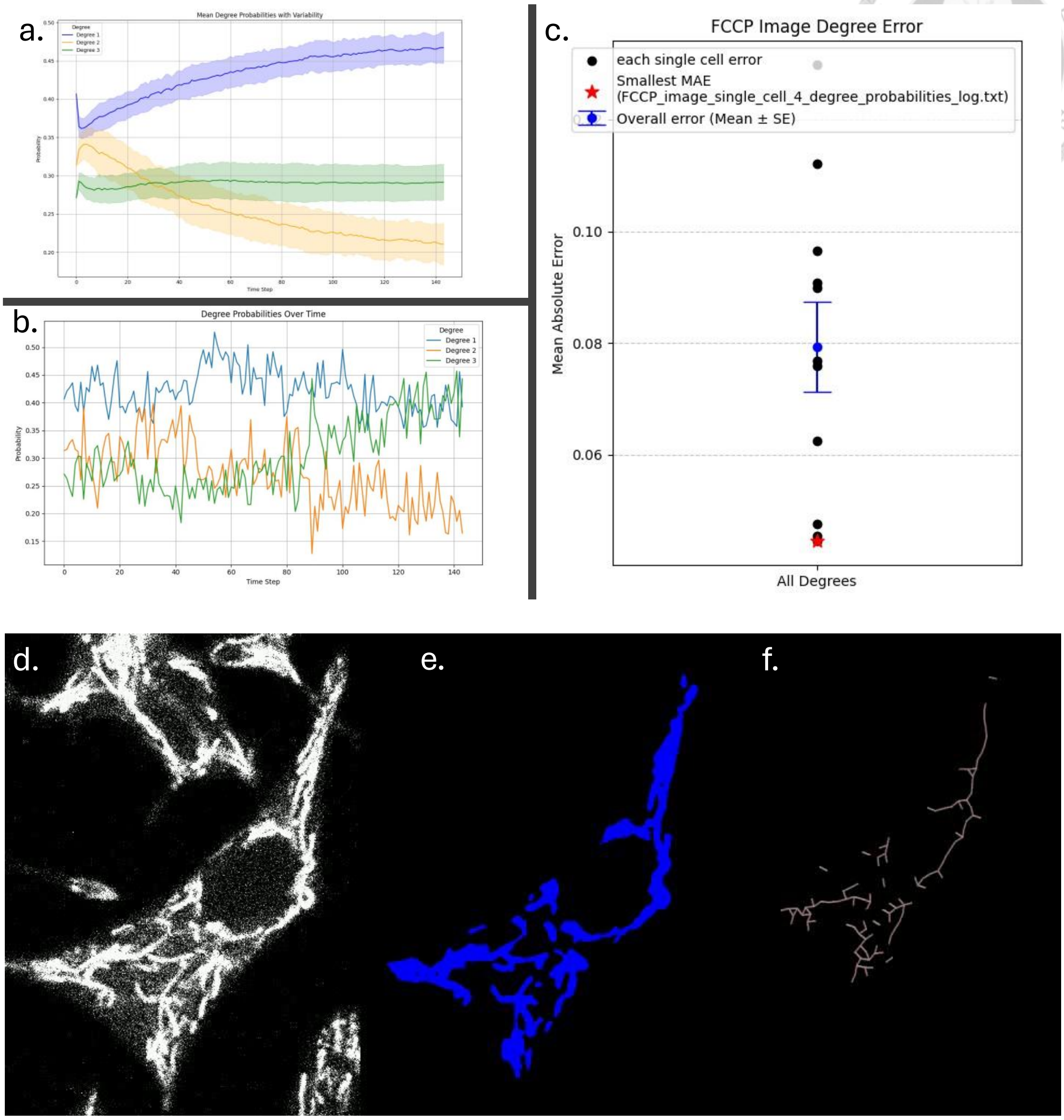


Figure 3.19. FCCP Treatment Analysis for Ins-1 Cells.

Panels (a–c) show comparisons between simulated and experimental mitochondrial network properties under FCCP treatment.

(a) Time-averaged degree probabilities (mean ± SD) for the best-fit simulation, reflecting fragmentation trends.

(b) Degree probabilities from the experimental image used for best-fit evaluation, representing the real mitochondrial network dynamics.

doi:10.6342/NTU202503815

(c) Mean absolute error (MAE) between experimental data and 100 simulation replicates. The red star indicates the simulation with the lowest MAE (best fit), and the blue point with error bar represents the overall mean ± SE.

(d) Raw confocal image of a single FCCP-treated Ins-1 cell.

(e) Corresponding segmentation mask.

(f) Skeletonized network used for graph extraction and degree quantification.

Under FCCP treatment, the mitochondrial network undergoes progressive fragmentation. As shown in Fig. 3.19(a), the best-fit simulation demonstrates a strong increase in degree 1 nodes (terminal ends), consistent with excessive fission activity and suppressed fusion due to membrane potential collapse. Degree 2 nodes remain moderate but slowly decline, while degree 3 nodes (branch points) remain relatively flat.

The corresponding experimental degree probabilities (Fig. 3.19(b)) reveal a more complex dynamic: while fragmentation dominates, there is also a noticeable and continuous increase in degree 3 nodes. This pattern implies ongoing structural complexity and suggests that under stress, mitochondria may still undergo non-canonical or compensatory fusion attempts that form disordered branches.

The quantitative comparison in Fig. 3.19(c) shows that although the best-fit simulation yields a low MAE (red star) relative to other replicates, the overall distribution of MAE values is more dispersed and higher than in control conditions (see Fig. 3.12(c)), indicating reduced model robustness under extreme perturbation.

This discrepancy, particularly the model’s failure to capture the experimental increase in degree 3—highlights a limitation of the current framework. As currently implemented, the agent-based model does not support conversions from linear

doi:10.6342/NTU202503815

segments (degree 2) to branch points (degree 3) unless explicitly triggered by predefined fusion events. It lacks mechanisms for unstructured or stress-induced remodeling pathways, which may account for the experimental complexity observed under FCCP treatment. This situation will explain detailly in the discussion section.

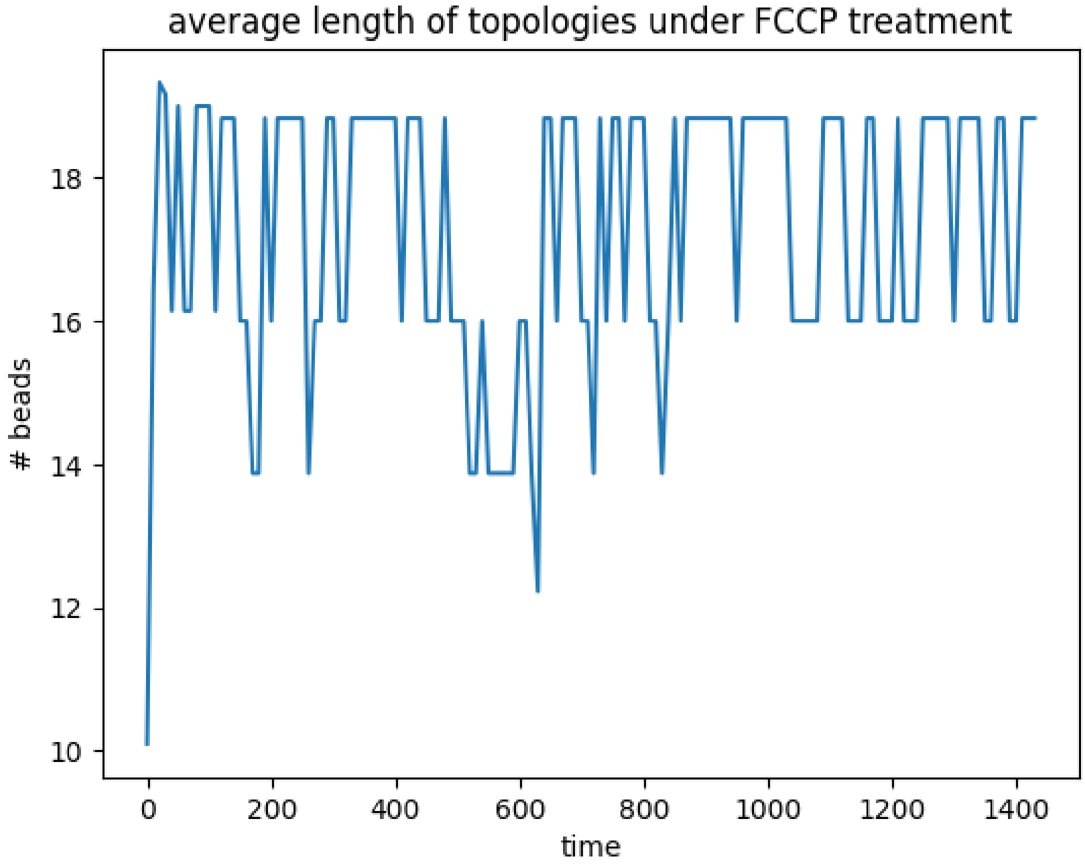


Figure 3.20. The average length of topologies of Ins-1 Cell under FCCP treatment remains consistently high, indicating elongated but disconnected mitochondrial fragments.

Figure 3.20. remains consistently high, fluctuating around 16 to 19 beads, with brief and infrequent drops. This suggests that the network is composed of a small number of elongated structures rather than fragmented units. The sustained high values imply that once fragmentation occurs, few individual topologies persist and grow in length, possibly due to a collapse into fewer surviving structures.

doi:10.6342/NTU202503815

This behavior contrasts with the more balanced remodeling observed under control conditions. FCCP, a mitochondrial uncoupler, disrupts membrane potential and suppresses mitochondrial function, which may impair normal fission activity, allowing elongated fragments to persist abnormally. The observed topology length suggests dysfunctional network dynamics, with impaired fragmentation and abnormal elongation.

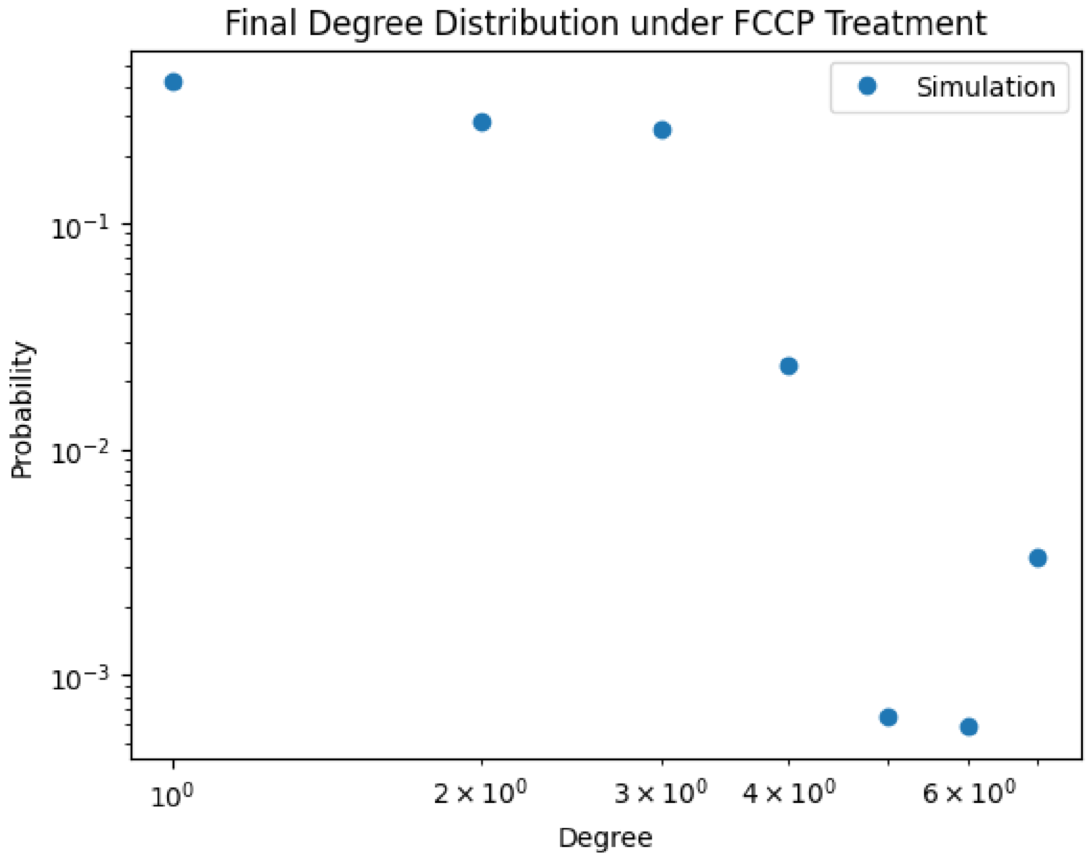


Figure 3.21. The Final degree distribution of Ins-1 Cell under FCCP treatment is dominated by lower-degree nodes, with reduced branching complexity relative to control.

Figure 3.21. is skewed toward low-degree nodes, with degree 1 remaining the most dominant. However, unlike the control condition, degree 3 nodes occur at comparable frequency to degree 2, indicating an unusual accumulation of branch

doi:10.6342/NTU202503815

points despite overall network disruption. Higher-degree nodes (degree ≥ 4) are present but rare, reflecting isolated instances of abnormal connectivity.

This distribution suggests paradoxical behavior: while the network is overall fragmented due to mitochondrial stress, branching still occurs, possibly due to disorganized or non-physiological fusion events. The combination of high fragmentation (degree 1) and non-trivial branching (degree 3) may reflect a pathological remodeling process, characteristic of mitochondrial dysfunction under FCCP-induced depolarization.

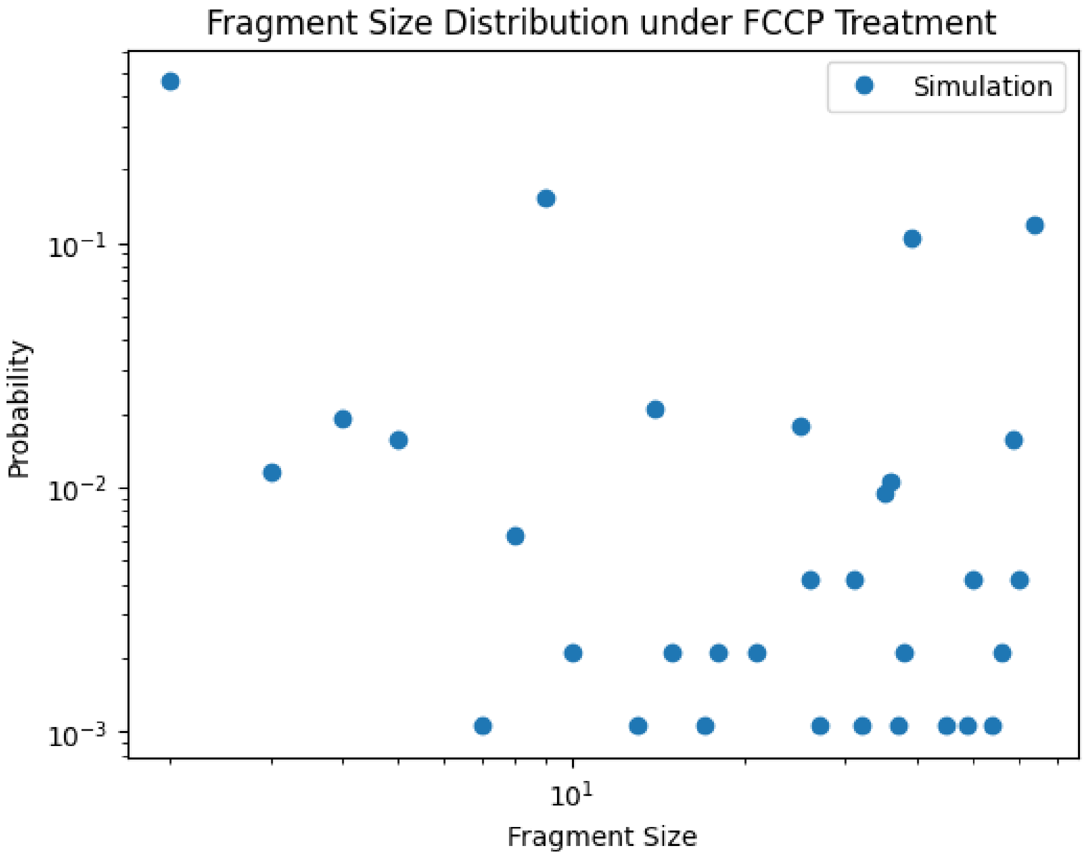


Figure 3.22. The fragment size distribution of Ins-1 Cell under FCCP treatment exhibits a widespread with an abundance of small fragments, reflecting network fragmentation and disassembly.

doi:10.6342/NTU202503815

Figure 3.22. is highly irregular and dispersed, lacking a clear monotonic decay. While small fragments remain dominant, a broad range of larger fragments appears with non-negligible probability, including several outliers at high fragment sizes. Unlike the control condition, where the distribution showed a smooth decline, the FCCP distribution is noisy and discontinuous, indicative of disorganized fragmentation and abnormal aggregation.

This irregularity suggests that the network undergoes pathological remodeling, where fusion and fission are not well regulated. The coexistence of very small and large fragments likely results from FCCP-induced mitochondrial dysfunction, where loss of membrane potential leads to both uncontrolled fission and compensatory or mis regulated fusion, producing a wide range of fragment sizes.

doi:10.6342/NTU202503815

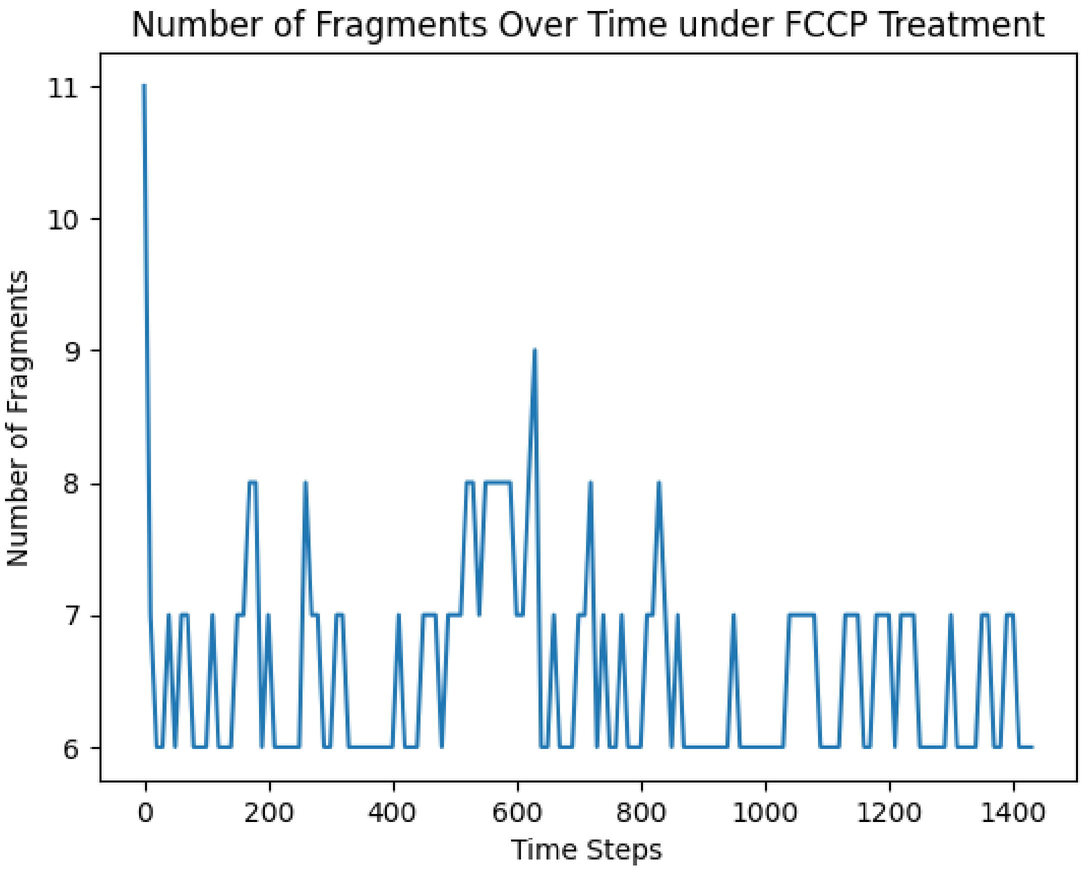


Figure 3.23. The number of fragments over time of Ins-1 Cell under FCCP treatment stabilizes at a higher baseline than control, consistent with persistent fission activity.

Figure 3.23. rapidly decreases and stabilizes at a low value. After an initial transient phase, the fragment count settles around 6–7 fragments, with minimal fluctuation throughout the rest of the simulation. This behavior contrasts sharply with the control condition, where fragment counts remained more variable and higher on average.

The consistently low fragment count suggests that FCCP leads to the collapse of network diversity, possibly through the elimination of smaller fragments or aggregation into a few abnormally large structures. This outcome reflects a dysfunctional dynamic state, in which normal mitochondrial turnover is suppressed, and the network fails to maintain typical levels of fragmentation and remodeling.

doi:10.6342/NTU202503815

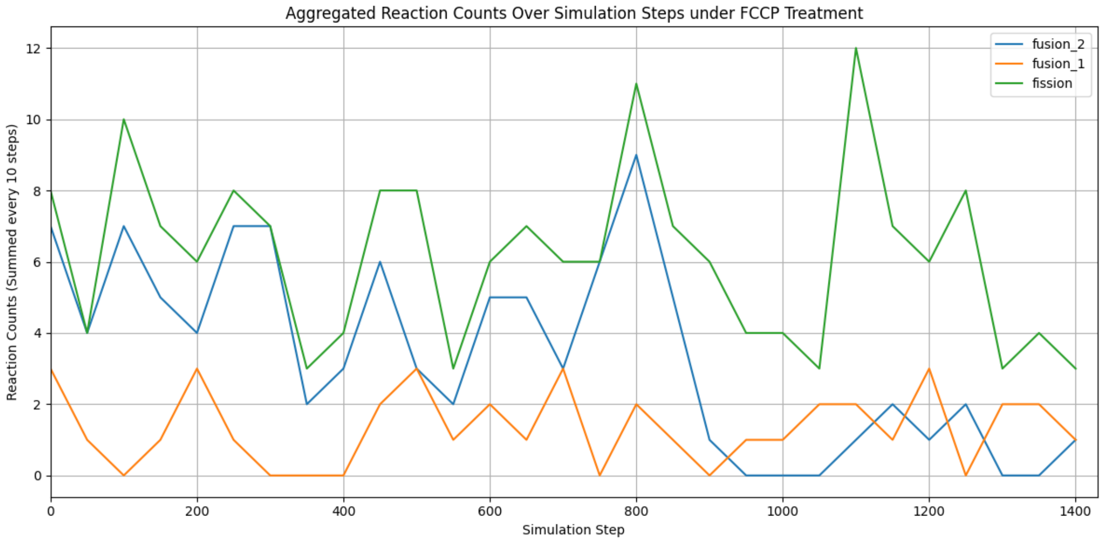


Figure 3.24. Aggregated reaction counts over time of Ins-1 Cell under FCCP treatment show a predominance of fission events with infrequent fusion_1 and fusion_2 events, highlighting a strong fission-dominant regime.

Figure 3.24. leads to an elevated and persistent level of fission activity (green), with reaction counts frequently reaching 6–10 events per aggregation window. In contrast, both fusion reaction types are reduced: fusion_2 (blue) declines gradually over time and occasionally drops to zero, while fusion_1 (orange) remains sporadic and consistently low throughout the simulation.

This imbalance between high fission and suppressed fusion supports the observed fragmentation of the mitochondrial network and the decline in fragment diversity. The depletion of fusion events, particularly fusion_2, suggests that network repair and reconnection mechanisms are impaired, consistent with FCCP's role in dissipating membrane potential and inhibiting fusion machinery.

Overall, these dynamics indicate that FCCP induces a pathological shift in mitochondrial remodeling, dominated by fission and largely unopposed by

doi:10.6342/NTU202503815

compensatory fusion, resulting in structural collapse and fragmentation.

3.Mdivi1 Network Condition (n = 6)

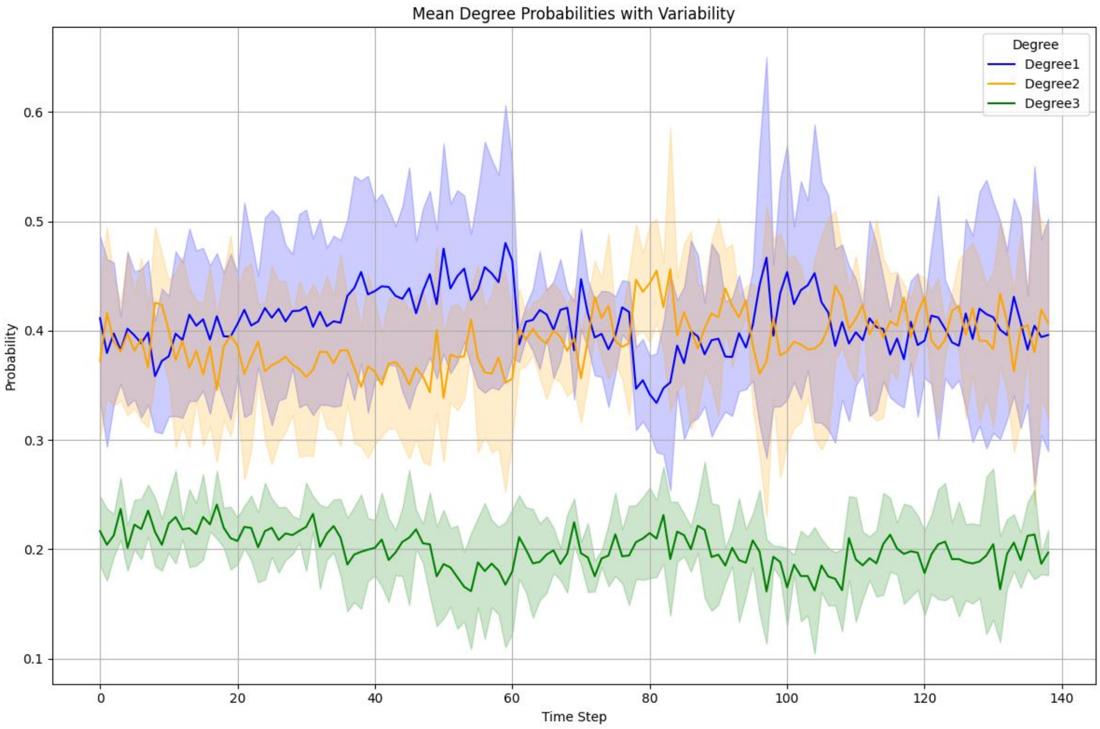


Figure 3.25 . Temporal evolution of mean degree distributions for Ins-1 cells under Mdivi-1 treatment.

Degree 1 (blue) and degree 2 (orange) nodes dominate and remain tightly coupled across time, while degree 3 nodes (green) remain consistently low. The network exhibits morphological stability, reflecting Mdivi-1's inhibition of fission without inducing excessive branching or fusion-driven remodeling.

Figure 3.25 displays the temporal evolution of degree distributions in Ins-1 cells treated with Mdivi-1, a pharmacological inhibitor of mitochondrial fission. The network topology remains notably stable over time. Degree 1 and degree 2 nodes dominate and closely overlap throughout the simulation, each fluctuating around a probability of ~0.40, suggesting a balance between terminal ends and linear segments. Degree 3 nodes (branch points) remain consistently lower, with average probabilities near 0.20 and minimal upward or downward trend.

doi:10.6342/NTU202503815

This behavior indicates that Mdivi-1 treatment preserves the existing mitochondrial architecture and prevents excessive fragmentation. The suppression of fission likely inhibits network breakdown, but without a strong fusion drive, the system does not exhibit extensive branching or elongation either. The narrow variability of bands further suggests reduced dynamical changes and higher topological uniformity among cells. Overall, the mitochondrial network appears morphologically stable and structurally constrained, consistent with a fission-inhibited but fusion-limited condition.

Table 3.4. Adjustable ReaDDy2 parameters for Ins-1 Mdivi1

parameter	**value**	**description (concise)**
Fusion_probability_1	0.6	tip-to-tip fusion probability
Fusion_probability_2	0.4	tip-to-side fusion probability
Fission_base_probability	0.15	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	15	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

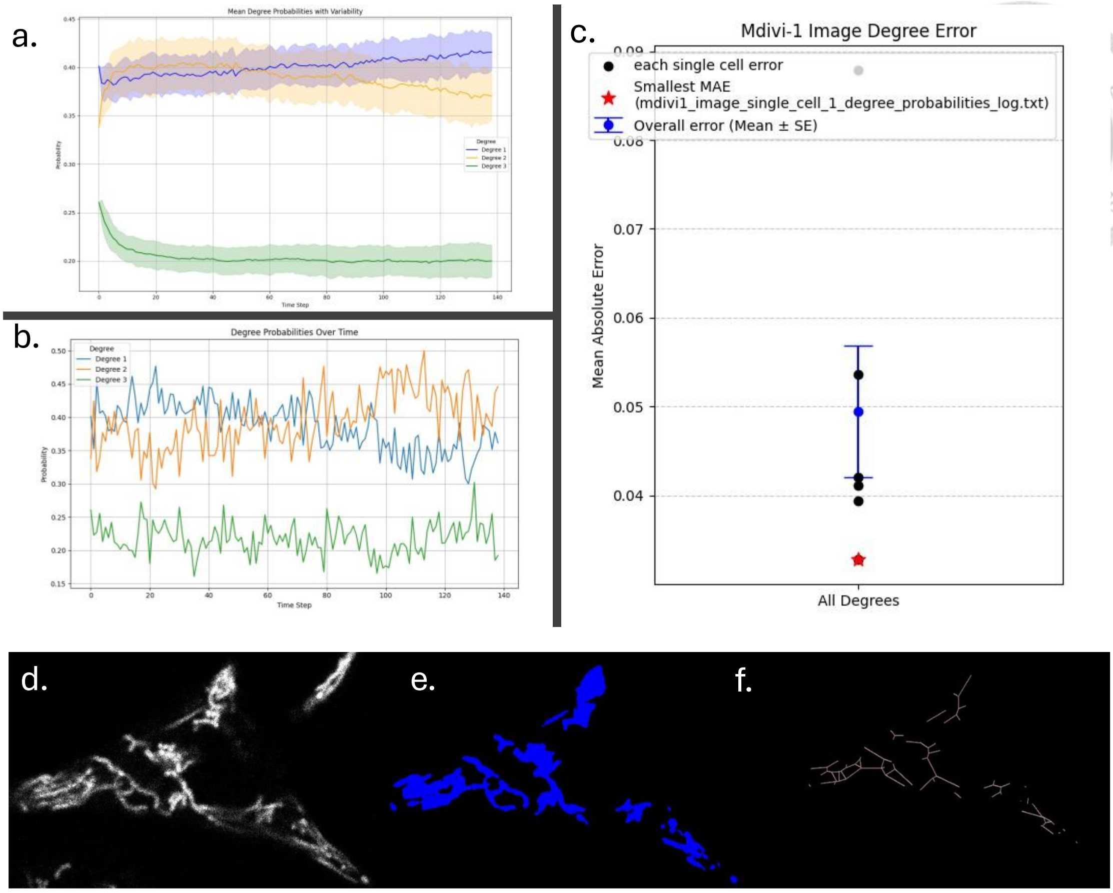


Figure 3.26. Mdivi1 Treatment Analysis for Ins-1 Cells.

Panels (a–c) compare simulation results with experimental mitochondrial network properties following Mdivi-1 treatment.

(a) Degree probability trajectory from the best-fit simulation shows stabilization of degree 1 and 2 nodes with low and declining degree 3, indicating limited remodeling.

(b) Degree probability from the corresponding experimental dataset used for best-fit evaluation.

(c) Mean absolute error (MAE) comparison across 100 simulation replicates. The red star marks the lowest MAE (best fit), while the blue point and error bar indicate the group mean ± SE.

(d) Confocal image of a single Mdivi-1 treated Ins-1 cell showing compact and fused mitochondrial clusters.

(e) Segmentation results in capturing the enhanced mitochondrial mass.

doi:10.6342/NTU202503815

(f) Extracted skeleton network, showing a prevalence of linear and terminal segments with sparse branching.

Under Mdivi-1 treatment results in a stabilized mitochondrial network structure, as illustrated in Fig. 3.26(a). The best-fit simulation shows persistent dominance of degree 1 and degree 2 nodes, both fluctuating near 0.4, with limited changes over time. Degree 3 nodes remain low and show a slight downward trend, indicating minimal new branching activity.

The real experimental degree probabilities from the matched dataset, shown in Fig. 3.26(b), support this observation. Terminal and linear segments are prevalent, while branch points remain sparse and steady. This morphology reflects a condition where fission is pharmacologically suppressed, and the network remains in a structurally constrained state.

As shown in Fig. 3.26(c), the best-fit simulation achieves a low MAE (red star), indicating strong alignment with the experimental profile. The narrow MAE distribution across 100 replicates (blue point with small SE) further confirms that this behavior is reproducible and not limited to one simulation instance.

Overall, Mdivi-1 effectively inhibits fragmentation, as expected from its role as a fission inhibitor. However, without additional cues promoting fusion, the network does not undergo extensive elongation or branching. This results in a topological stasis characterized by balanced terminal and linear segments with low complexity, consistent with the subdued dynamics of a fusion-competent but fission-impaired mitochondrial state.

doi:10.6342/NTU202503815

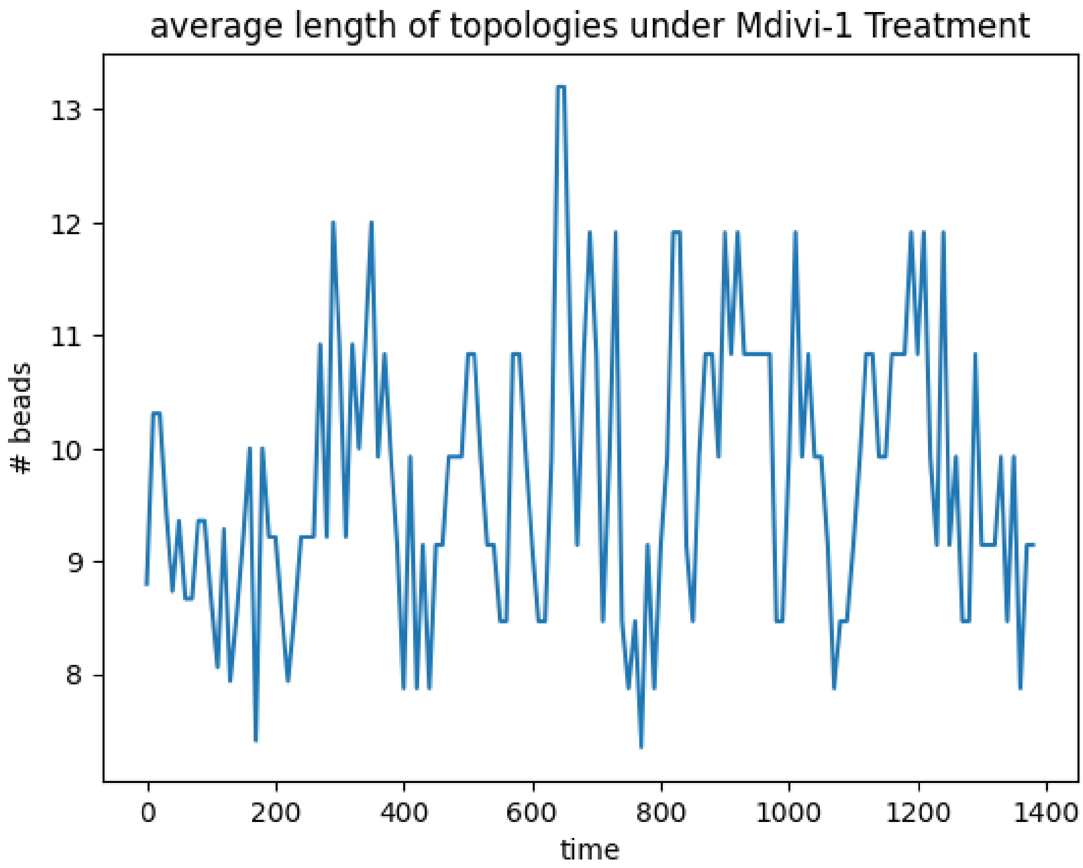


Figure 3.27. The average length of topological fragments of Ins-1 Cell under Mdivi1 treatment fluctuates around a higher mean compared to control, suggesting reduced breakage and enhanced elongation.

Figure 3.27. fluctuates between 8 and 13 beads, with no consistent increasing or decreasing trend over time. While short-term variability is evident, the values remain within a moderate range, suggesting that the network avoids both excessive fragmentation and elongation.

The relatively stable average length observed here suggests that fission suppression may lead to moderate elongation, but compensatory mechanisms—such as reduced fusion or passive fragmentation—may limit unchecked network growth. Overall, the network appears to maintain moderate structural integrity, consistent with partial remodeling under fission inhibition.

doi:10.6342/NTU202503815

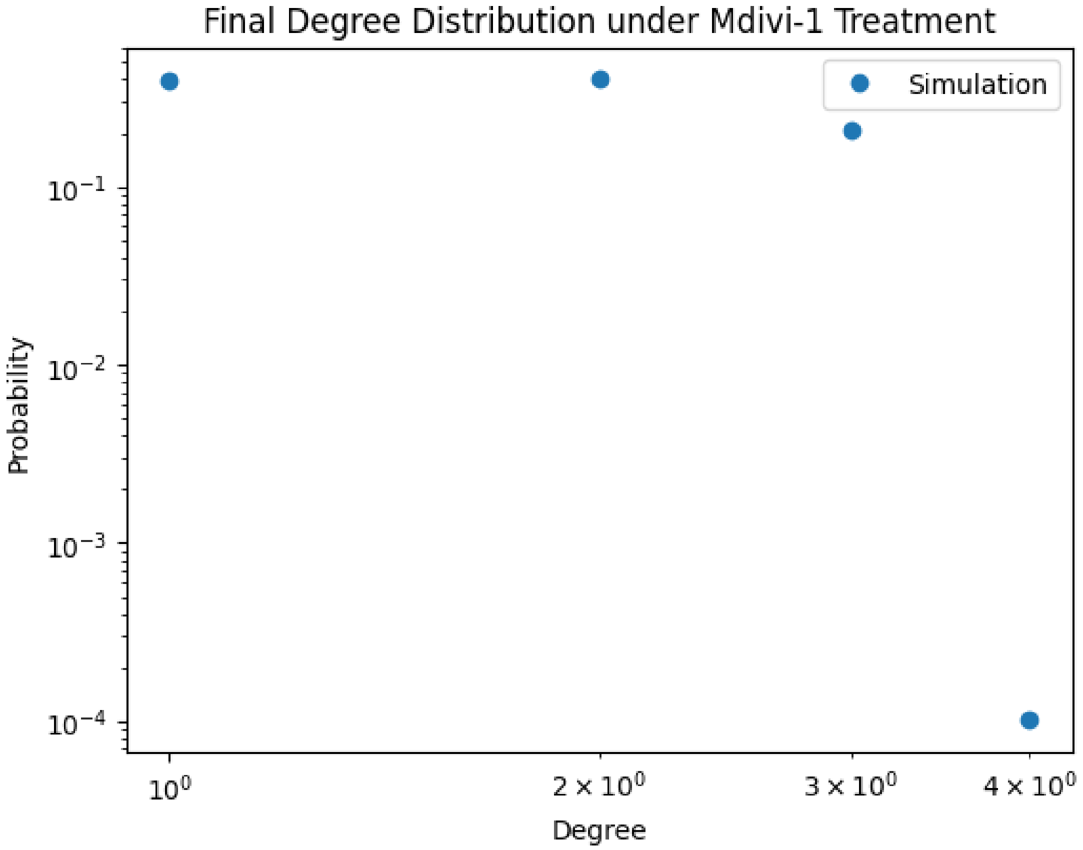


Figure 3.28. Final Degree distribution of Ins-1 Cell under Mdivi1 treatment shows a dominance of linear (degree 2) and terminal (degree 1) nodes, while branch points (degree ≥ 3) are infrequent, consistent with suppressed network complexity.

Figure 3.28. is dominated by degree 1 and degree 2 nodes, which occur with similar and highest probabilities. Degree 3 nodes are less frequent but clearly present, while higher-degree nodes (≥4) are rare.

This distribution suggests that despite the inhibition of fission via fission suppression, the network does not shift toward excessive branching or hyper-connectivity. Instead, it maintains a moderately linear and weakly branched structure, with limited morphological complexity. The presence of degree 3 nodes indicates residual branching capacity, potentially driven by low-level fusion or incomplete

doi:10.6342/NTU202503815

fission suppression. Overall, the network topology remains structured yet restrained, consistent with a partially remodeled state under Mdivi-1 influence.

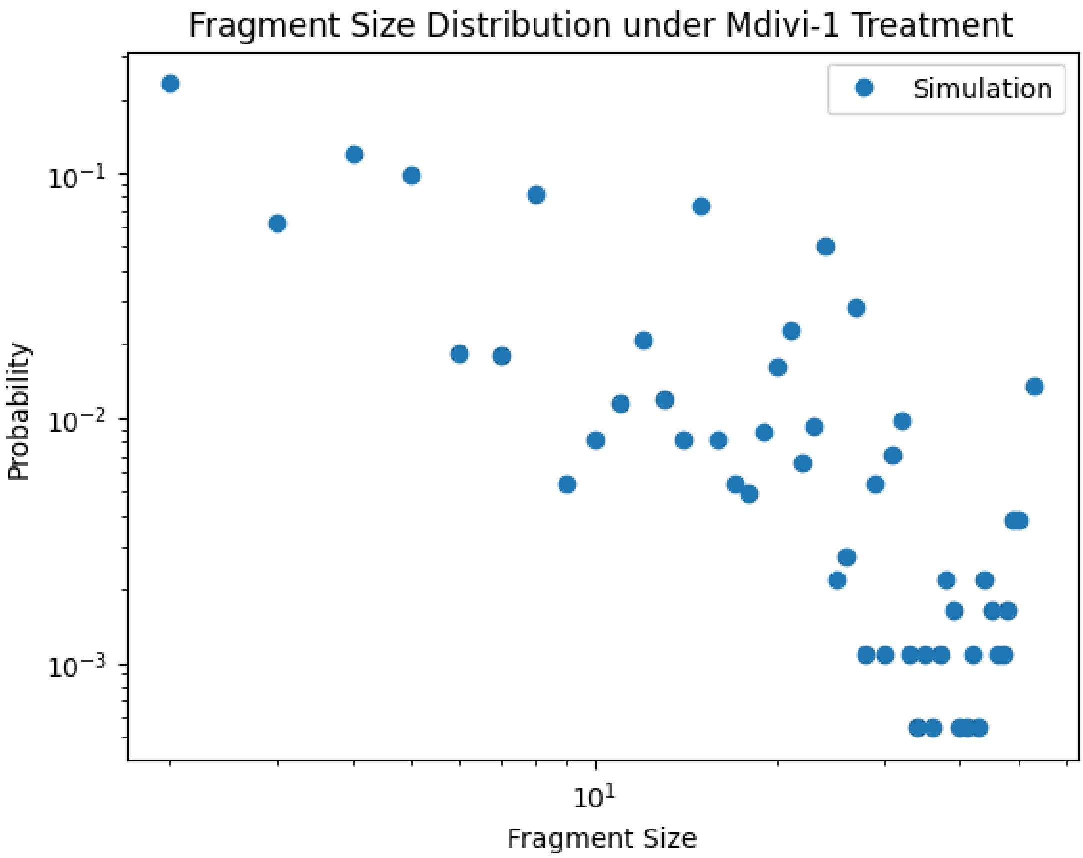


Figure 3.29. Fragment size distribution of Ins-1 Cell under Mdivi1 treatment indicates a wider spread of sizes with a tendency toward larger, fused structures.

Figure 3.29 displays a broad, heterogeneous range of fragment sizes. While small fragments remain the most probable, a wide spread of medium and larger fragments is observed with non-negligible probability. The absence of a sharp drop-off and the dispersed distribution suggest that fragmentation is suppressed but not eliminated.

This distribution aligns with the known action of Mdivi-1 as a fission inhibitor that impairs mitochondrial fission. The persistence of larger fragments reflects reduced fission activity, while the presence of smaller fragments indicates that some

doi:10.6342/NTU202503815

division still occurs—potentially through passive mechanisms or incomplete inhibition. Overall, the network exhibits a structurally diverse but non-collapsing topology, characteristic of partially stabilized mitochondrial dynamics.

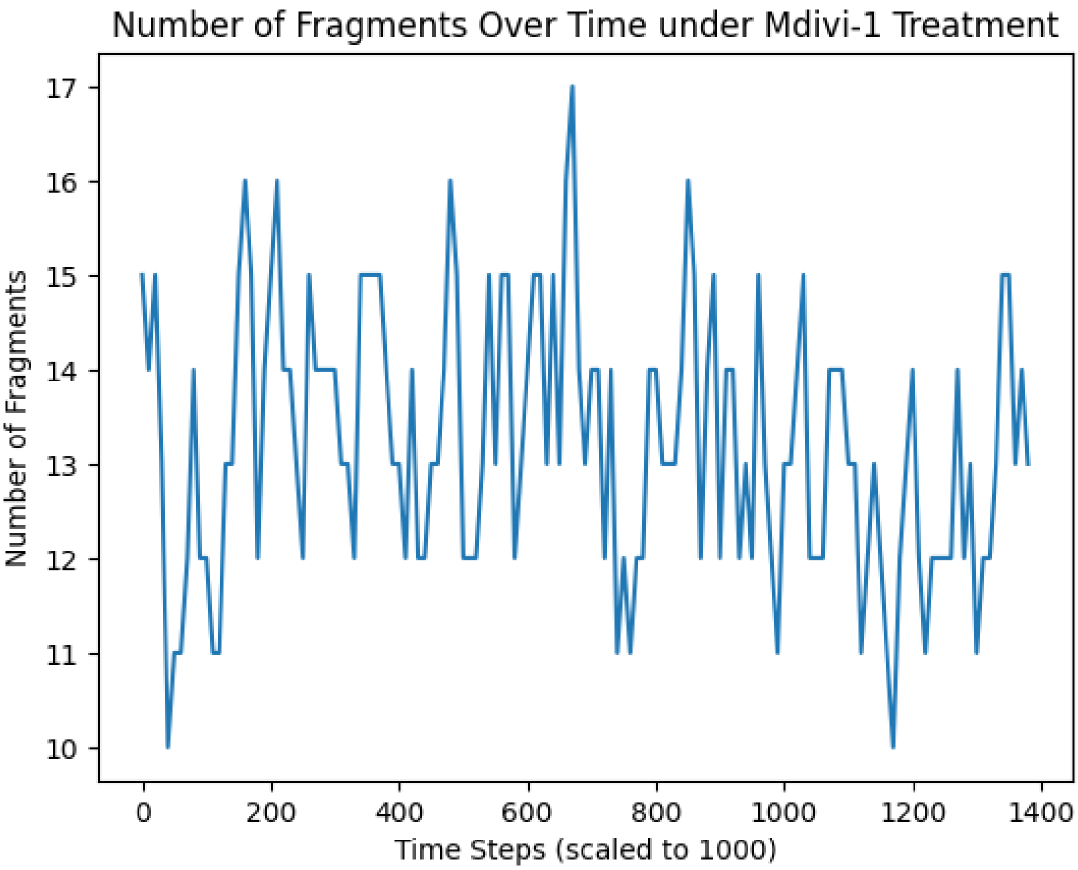


Figure 3.30. The number of network fragments of Ins-1 Cell under Mdivi1 treatment remains moderately stable over time, with higher baseline counts than FCCP, reflecting balanced but reduced fission activity.

Figure 3.30. fluctuates between 10 and 17, showing moderate variability without a clear long-term trend. The fragment count remains relatively stable, suggesting that network turnover continues, albeit with a less dynamic range compared to untreated conditions.

doi:10.6342/NTU202503815

These observations are consistent with Mdivi-1's inhibition of fission -mediated fission. However, the persistent fluctuations indicate that fission is not fully abolished, and compensatory or residual mechanisms may sustain a moderate level of network remodeling. The system appears to settle into a partially remodeled state, maintaining structure without collapsing into hyper fusion or excessive fragmentation.

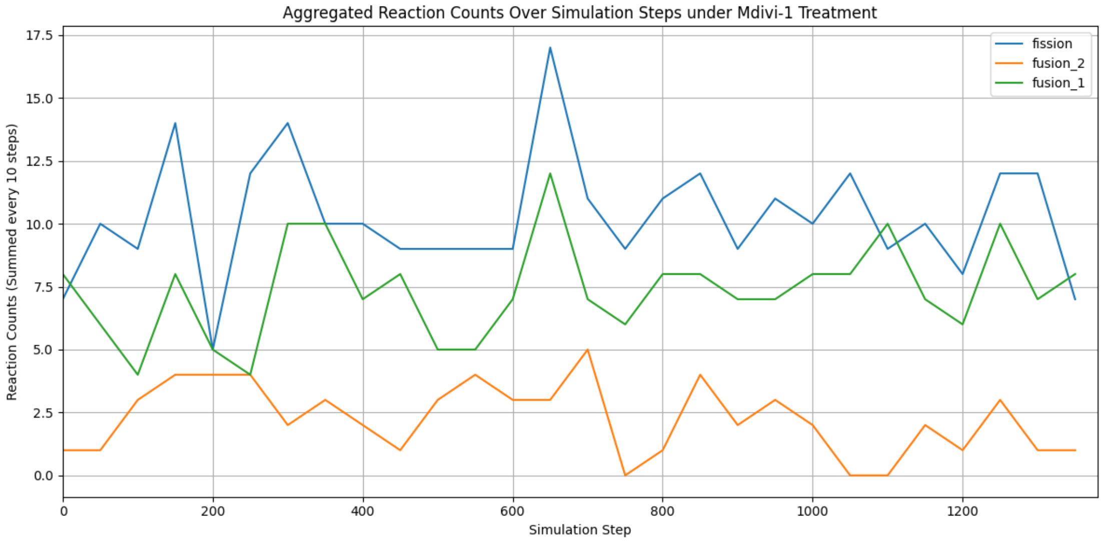


Figure 3.31. Aggregated reaction dynamics of Ins-1 Cell under Mdivi1 treatment reveal a distinct profile in which fusion_2 events are infrequent, whereas fission and fusion_1 remain active. This dynamic is consistent with a network favoring stabilization and elongation without significant remodeling.

Figure 3.31, all three reaction types—fission, fusion_1, and fusion_2—remain active throughout the simulation. Fission events (blue) are consistently present, though rarely dominant, suggesting that fission inhibition only partially suppresses fission activity. Fusion_1 events (green) occur with comparable frequency, indicating a maintained level of tip-to-tip fusion. In contrast, fusion_2 events (orange) occur infrequently and remain at low levels, showing occasional gaps of inactivity.

The relatively balanced presence of fission and fusion_1 suggests that the network remains remodeling-capable, despite pharmacological inhibition of fission.

doi:10.6342/NTU202503815

The diminished fusion_2 activity may reflect a shift toward simpler reconnection mechanisms, as more complex fusion modes are underutilized. Overall, these dynamics support the emergence of a stabilized but still plastic mitochondrial network, consistent with the intermediate morphological features observed under Mdivi-1.

4. Ins-1 Oligomycin Network Condition (n = 7)

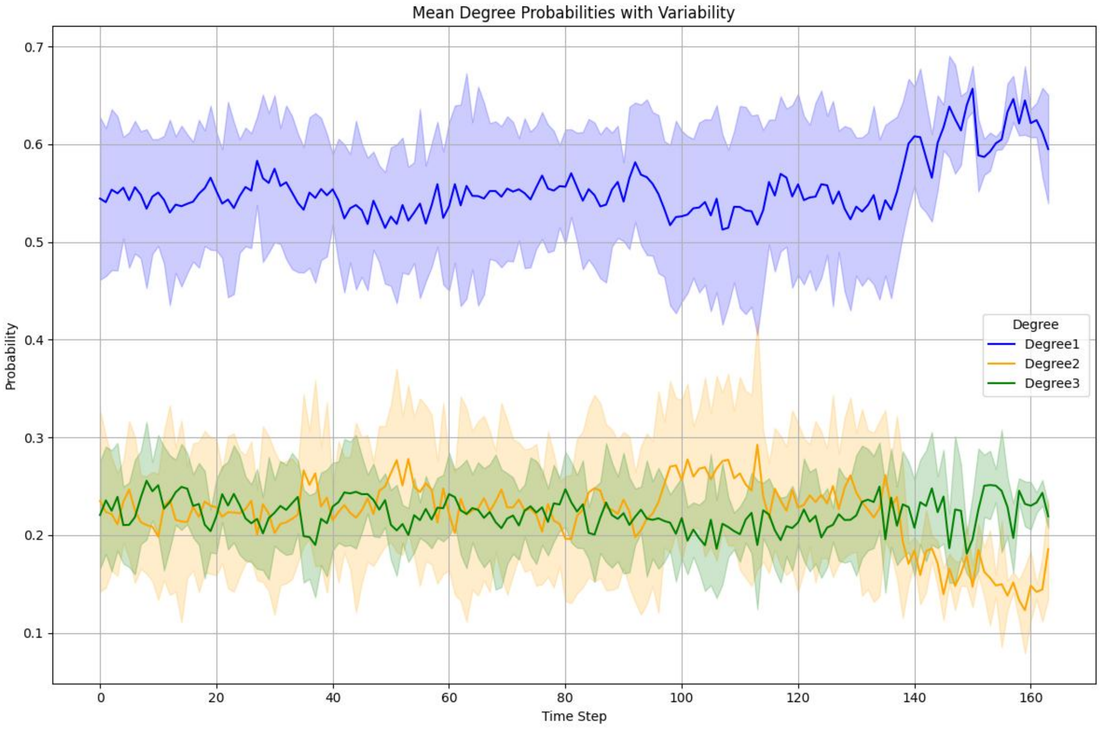


Figure 3.32 . Temporal evolution of mean degree distributions for Ins-1 cells under oligomycin treatment.

Degree 1 nodes (blue) increase over time, indicating growing network fragmentation. Degree 2 nodes (orange) decline in later stages, while degree 3 nodes (green) remain consistently low. These trends reflect impaired fusion and persistent structural disintegration in energy-compromised cells.

Figure 3.32. depicts the time evolution of degree distributions in Ins-1 cells exposed to oligomycin, an ATP synthase inhibitor that alters mitochondrial

doi:10.6342/NTU202503815

bioenergetics. The network structure is characterized by a strong and sustained dominance of degree 1 nodes (terminal ends), with probabilities consistently above 0.5 and increasing slightly over time. This trend indicates progressive fragmentation of the network and accumulation of disconnected tubules.

Degree 2 nodes (linear segments) initially remain relatively stable but gradually decline in prevalence, particularly after step 120, suggesting breakdown of continuous segments. Meanwhile, degree 3 nodes (branch points) remain low and fluctuate within a narrow band, indicating that complex branching events are largely absent under this condition.

These dynamics suggest that oligomycin disrupts mitochondrial structure by suppressing energy production, which reduces fusion competence and favors fission-like phenotypes. The consistent increase in fragmentation without compensatory branching supports the interpretation of an energy-deficient, disorganized network state. Variability bands are wider toward later time points, reflecting increased heterogeneity across cells in response to energetic collapse.

Table 3.5. Adjustable ReaDDy2 parameters for Ins-1 Oligomycin

parameter	value	description (concise)
Fusion_probability_1	0.3	tip-to-tip fusion probability
Fusion_probability_2	5	tip-to-side fusion probability
Fission_base_probability	0.15	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	15	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

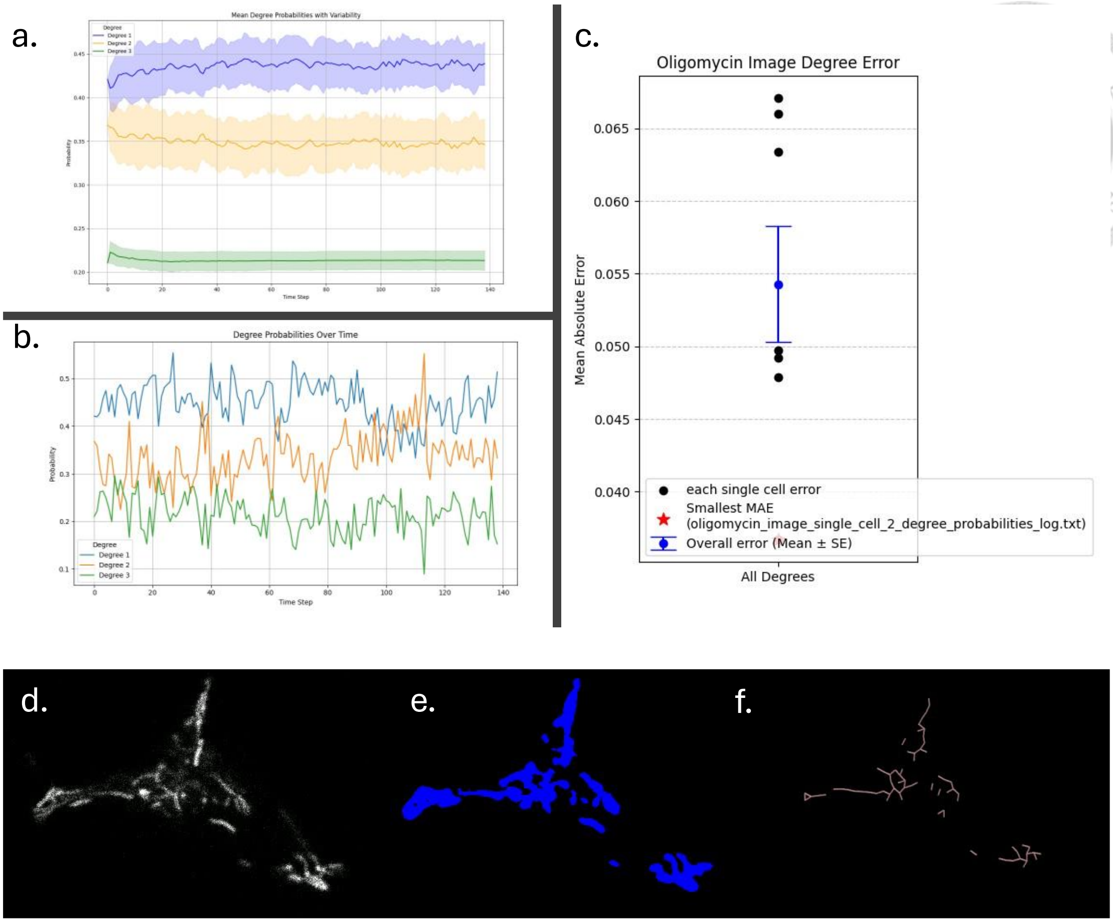


Figure 3.33. Oligomycin treatment analysis for Ins-1 cells.

Panels (a–c) compare simulated and experimental mitochondrial network properties following Oligomycin treatment.

(a) Time-averaged degree probabilities (mean ± SD) from the best-fit simulation replicate, showing persistent dominance of degree 1 nodes and suppression of branching.

(b) Degree probabilities from the experimental image used for best-fit evaluation.

(c) Mean absolute error (MAE) between 100 simulations and the experimental data. The red star marks the simulation with the lowest MAE, and the blue dot with error bar indicates the overall mean ± SE.

(d) Raw fluorescence image of a mitochondrion-labeled Ins-1 cell under Oligomycin treatment.

(e) Corresponding segmentation mask used to extract mitochondrial structure.

doi:10.6342/NTU202503815

(f) Skeletonized network derived from the segmentation, used for graph-based analysis.

Under Oligomycin treatment, the mitochondrial network maintains a fragmented but morphologically stable state. As shown in Fig. 3.33(a), the best-fit simulation features high and consistent levels of degree 1 nodes (terminal ends), with minimal changes over time—reflecting the persistence of disconnected mitochondrial structures. Degree 2 nodes (linear segments) remain at moderate levels, and degree 3 nodes (branch points) stay consistently low.

The experimental degree distribution from the matched dataset, shown in Fig. 3.33(b), closely mirrors this topology, reinforcing the validity of the simulation's structural predictions. This morphology reflects impaired network remodeling, as mitochondria remain energy-deficient due to ATP synthase inhibition but are not subjected to membrane depolarization or acute stress.

The MAE comparison in Fig. 3.33(c) shows a tightly clustered error distribution across replicates, with the best-fit simulation (red star) achieving a very low MAE and the group mean (blue marker) indicating strong model reproducibility.

Taken together, these findings suggest that oligomycin induces a metabolically constrained condition in which both fission and fusion activities are limited. The mitochondrial network exhibits neither catastrophic fragmentation nor compensatory branching, but instead stabilizes in a low-dynamic, low-complexity configuration.

doi:10.6342/NTU202503815

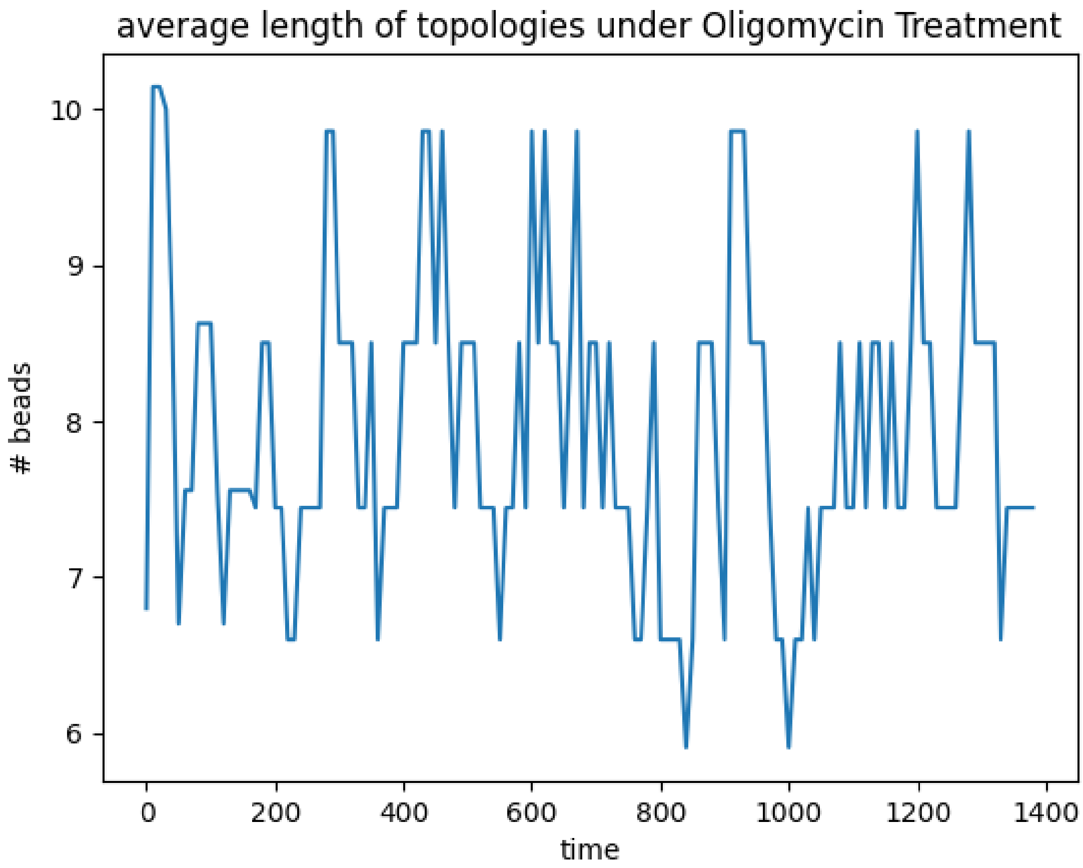


Figure 3.34. The average length of topologies of Ins-1 Cell under Oligomycin treatment remains moderately variable over time, reflecting ongoing dynamic remodeling despite treatment.

Figure 3.34. fluctuates between 7 and 11.5 beads, with no consistent trend of elongation or fragmentation over time. The trajectory shows moderate variability, suggesting that the network undergoes continuous remodeling while maintaining overall structural balance.

Oligomycin, an ATP synthase inhibitor, disrupts mitochondrial energy production without directly altering membrane potential. The observed topology length pattern reflects a partially suppressed but not severely impaired network, where energy limitation slightly constrains fission–fusion dynamics but still allows for moderate

doi:10.6342/NTU202503815

remodeling. The resulting topology indicates a metabolically stressed but morphologically stable mitochondrial state.

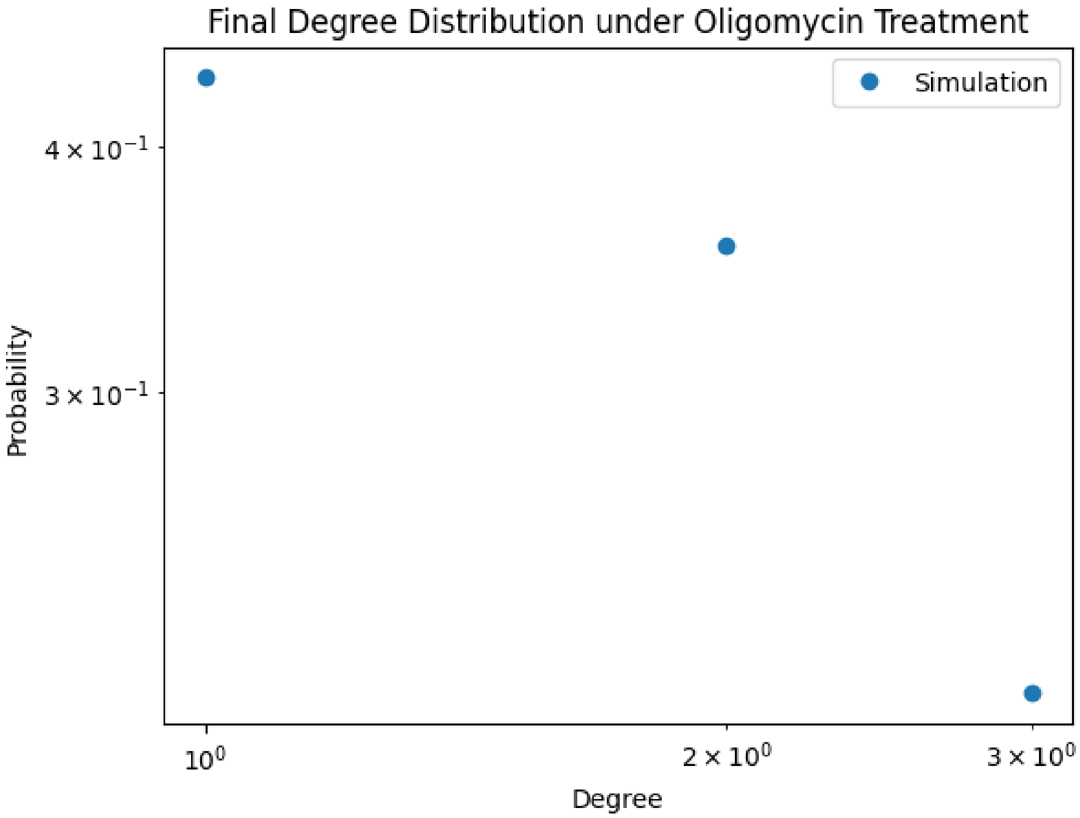


Figure 3.35. The final degree distribution of Ins-1 Cell under Oligomycin treatment shows a pronounced medium peak at degree 3, consistent with a highly branched morphology.

Figure 3.35. reveals a skewed pattern dominated by degree-1 and degree-2 nodes, with probabilities of approximately 0.45 and 0.35, respectively. Degree-3 nodes account for the remaining ~0.2, while higher-order connections are absent.

This configuration reflects a moderately connected network, with limited branching. The dominance of lower-degree nodes suggests that the mitochondrial network remains mostly linear or slightly branched under Oligomycin, which is consistent with reduced fusion activity or a mild constraint on network complexity.

doi:10.6342/NTU202503815

This pattern supports the notion that ATP depletion under Oligomycin subtly restricts mitochondrial remodeling, yielding a structure that is more fragmented than control but more connected than FCCP-induced collapse.

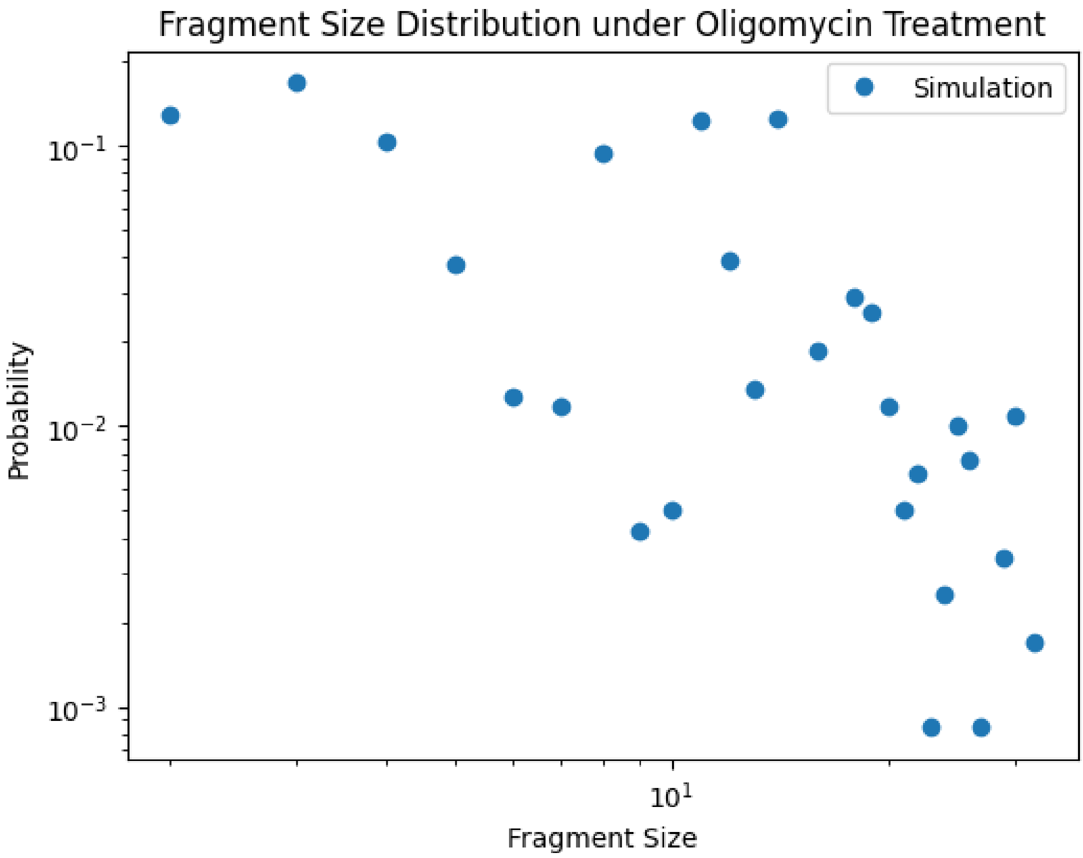


Figure 3.36. Fragment size distribution of Ins-1 Cell under Oligomycin treatment follows a broad-tailed pattern, indicating heterogeneous fragment populations.

Figure 3.36. demonstrates a broad spectrum of fragment sizes, with a long-tail trend extending beyond 30 nodes. Although smaller fragments are more frequent, the presence of larger clusters at lower probabilities indicates intermittent connectivity and partial fusion within the network.

This distribution suggests that Oligomycin treatment, while impairing ATP synthesis, does not completely inhibit fusion events. Instead, the network exhibits

doi:10.6342/NTU202503815

coexisting small and moderately large fragments, possibly resulting from localized energy preservation or spatial clustering of active mitochondria. Compared to FCCP, which collapses the network into smaller units, and Mdivi-1, which enhances connectivity, Oligomycin induces an intermediate fragmentation regime reflective of its partial energetic inhibition.

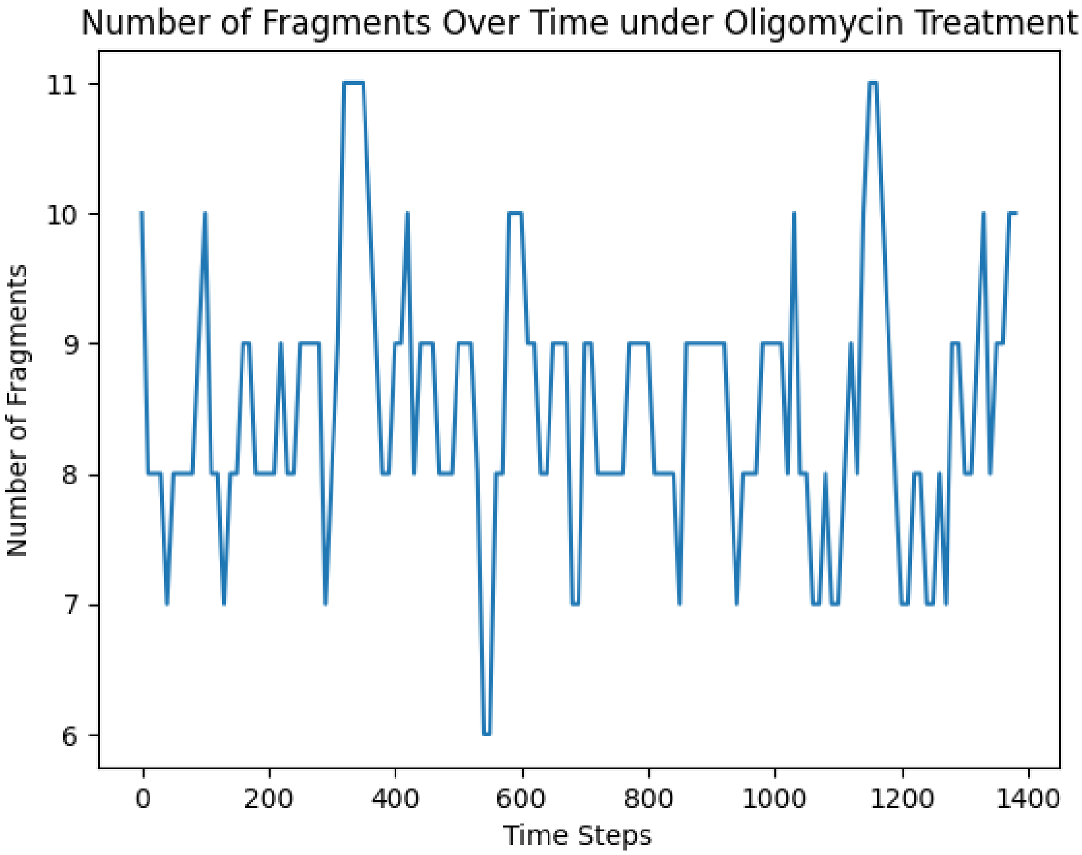


Figure 3.37. The number of network fragments of Ins-1 Cell under Oligomycin treatment fluctuates between 7 and 10 over the course of simulation, indicating partial network preservation.

Figure 3.37. fluctuates within a narrow range between 6 and 10, suggesting a relatively stable network topology throughout the simulation. Unlike FCCP, which leads to persistent fragmentation, or Mdivi-1, which promotes gradual reconnection, Oligomycin maintains a moderate fragmentation state over time.

doi:10.6342/NTU202503815

This behavior reflects the partial inhibitory effect of Oligomycin on mitochondrial function, where fission is not strongly enhanced and fusion is not entirely suppressed. The system remains in a dynamic but steady regime, potentially due to ATP depletion slowing down both fusion and fission reactions without drastically altering network balance.

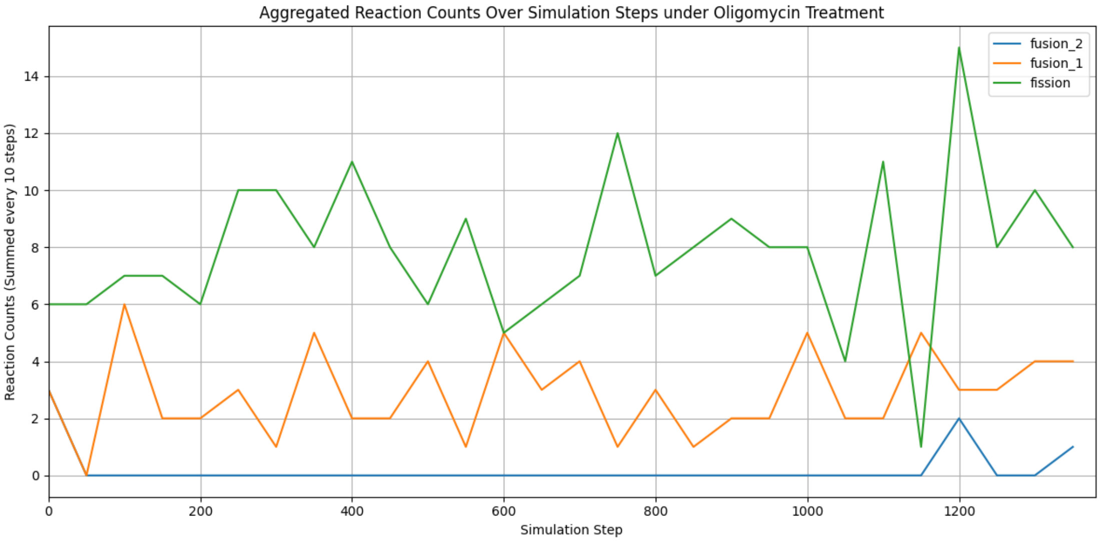


Figure 3.38. Aggregated reaction counts reveal a clear imbalance: fusion_1 reactions dominate, while fusion_2 is nearly absent, and fission events occur sporadically. This kinetic profile aligns with the sustained branching morphology observed.

Figure 3.38. reveals moderate but consistent mitochondrial remodeling activity. Fission reactions occur more frequently than both types of fusion events, particularly fusion_1, which remains largely suppressed throughout the simulation.

While fusion_2 events do persist intermittently, their frequency is relatively lower and lacks prolonged surges, suggesting that intermediate-scale fusion is not dominant in this condition. The overall trend reflects a mild fragmentation bias, consistent with the steady but fragmented network observed in the fragment count and fragment size distribution data.

doi:10.6342/NTU202503815

These results support the hypothesis that Oligomycin, by inhibiting ATP synthase and reducing cellular ATP levels, partially impairs energy-dependent fusion mechanisms while allowing baseline fission to continue. The resulting dynamics favor network maintenance over drastic reorganization, reinforcing a steady-state topology with moderate connectivity.

5. Ins-1 Rotenone Network Condition (n = 6)

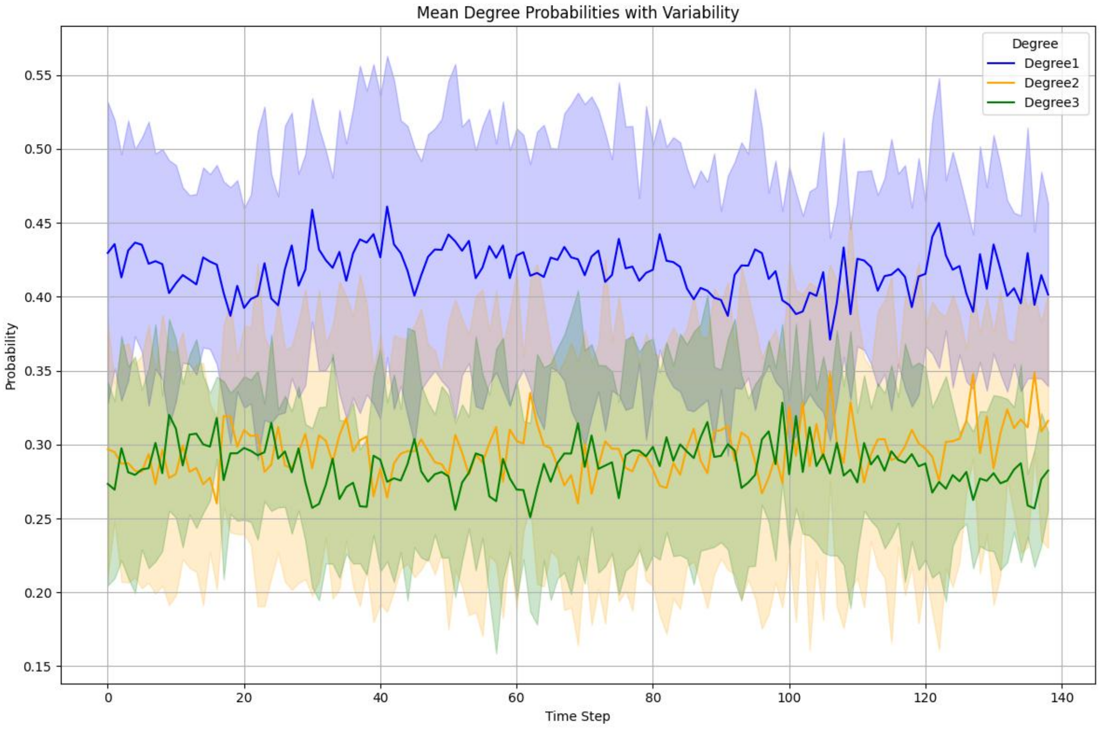


Figure 3.39 . Temporal evolution of mean degree distributions for Ins-1 cells under rotenone treatment.

Degree 1 (blue) nodes dominate but show broad fluctuations, while degree 2 (orange) and degree 3 (green) nodes remain balanced and highly variable. The mitochondrial network exhibits structural heterogeneity and instability, reflecting dysregulated remodeling dynamics under oxidative stress.

Figure 3.39 illustrates the time evolution of degree probabilities in Ins-1 cells subjected to rotenone treatment, a complex I inhibitor known to impair mitochondrial

doi:10.6342/NTU202503815

respiration and increase oxidative stress. The mitochondrial network exhibits notable heterogeneity, with relatively balanced distributions across all three degree classes.

Degree 1 nodes (blue) maintain the highest proportion but fluctuate widely over time, indicating inconsistent fragmentation behavior across cells. Degree 2 (orange) and degree 3 (green) nodes remain moderately represented and largely overlap, with both showing considerable temporal and inter-replicate variability. This pattern suggests that under rotenone treatment, mitochondrial networks do not converge on a single structural outcome but rather oscillate between fragmented, linear, and moderately branched states.

The increased variability across all degrees implies that rotenone induces a dysregulated network state, where mitochondria respond inconsistently to metabolic stress. Some networks undergo excessive fission, while others may attempt compensatory remodeling. This stochastic or heterogeneous behavior is consistent with the pathological complexity observed in rotenone-exposed systems, where bioenergetic failure and ROS accumulation disrupt mitochondrial dynamics in cell-specific ways.

Table 3.6. Adjustable ReaDDy2 parameters for Ins-1 Rotenone

parameter	**value**	**description (concise)**
Fusion_probability_1	3	tip-to-tip fusion probability
Fusion_probability_2	2	tip-to-side fusion probability
Fission_base_probability	0.025	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	20	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

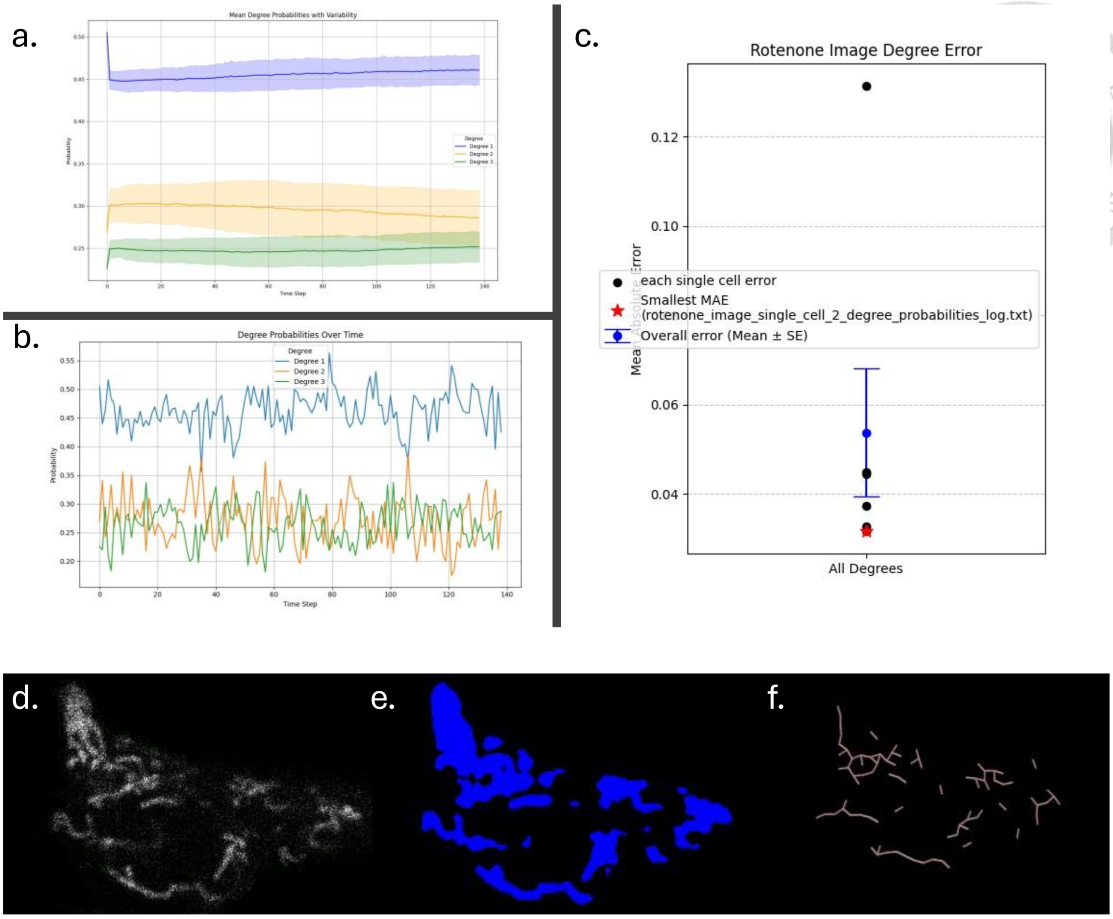


Figure 3.40. Rotenone treatment analysis for Ins-1 cells.

Panels (a–c) compare simulated and experimental mitochondrial network dynamics under rotenone exposure.

(a) Degree probabilities (mean ± SD) over time from the best-fit simulation replicate.

(b) Degree probabilities extracted from the corresponding experimental dataset used for best-fit evaluation.

(c) Mean absolute error (MAE) comparison across 100 simulation replicates. The red star highlights the best-fit case with the lowest MAE, while the blue dot and error bar represent the overall mean ± SE.

(d) Confocal image of a representative Ins-1 cell under Rotenone treatment.

(e) Image segmentation mask highlighting mitochondrial morphology.

(f) Extracted skeleton network used for topological comparison with simulations.

doi:10.6342/NTU202503815

Under Rotenone treatment, Fig. 3.40(a), the best-fit simulation predicts stable and dominant degree 1 nodes (terminal ends), indicating a sustained fragmented network. Degree 2 and degree 3 nodes remain lower but display modest fluctuations throughout the time course.

In contrast to prior figures showing simulation-wide trends, Fig. 3.40(b) depicts the experimental degree probabilities from the single cell most closely matched to the best-fit simulation. These experimental data confirm a structurally fragmented state, with persistent terminal ends and a small but non-negligible proportion of branching (degree 3) over time. The presence of degree 3 nodes suggests incomplete or disordered remodeling events in response to rotenone-induced stress.

The MAE comparison in Fig. 3.40(c) supports the simulation’s accuracy under this condition: the best-fit replicate (red star) achieves minimal deviation from experimental observations, and the overall distribution of errors is relatively narrow, indicating good reproducibility.

Rotenone disrupts mitochondrial respiration by inhibiting complex I, increasing ROS production and impairing network regulation. This condition leads to a fragmented, variably branched network with high inter-cell variability—captured by both simulation and experiment. The data point to a stress-induced phenotype in which fission dominates but stochastic fusion attempts introduce heterogeneity, resulting in a structurally unstable mitochondrial network.

doi:10.6342/NTU202503815

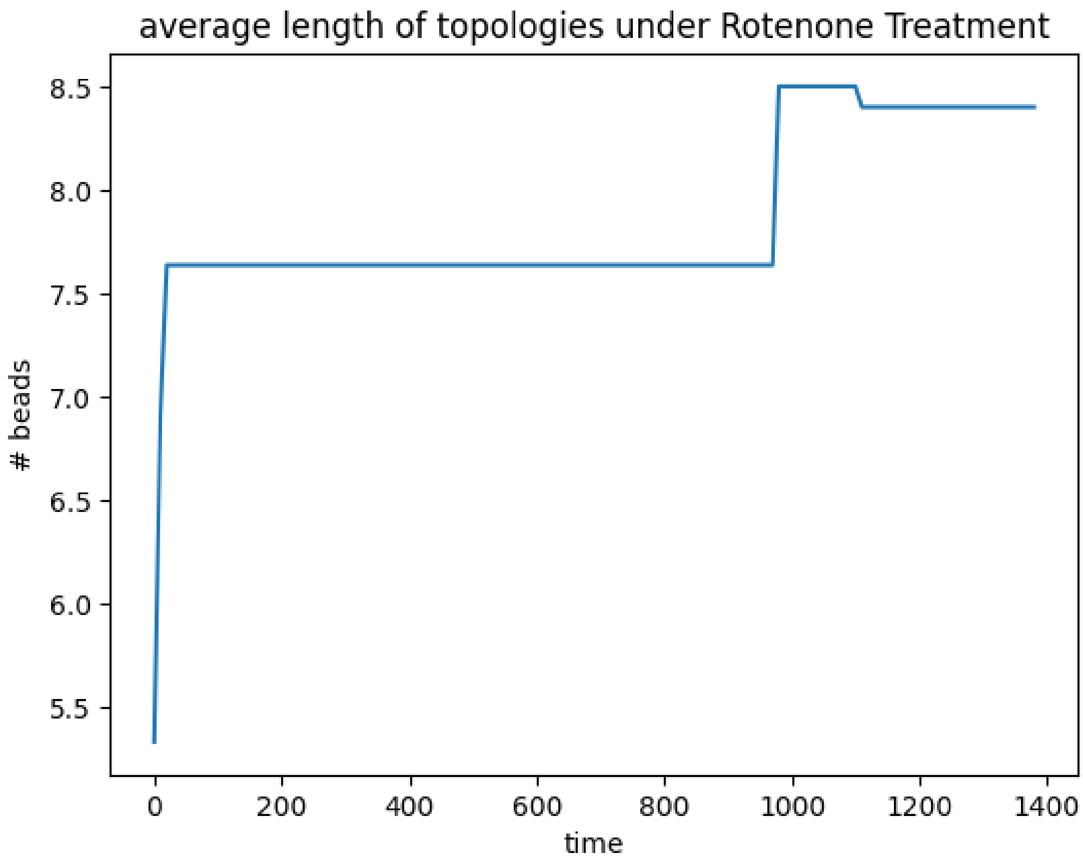


Figure 3.41. Time-series of the average number of topologies of Ins-1 Cell under Rotenone treatment shows a gradual increase in mitochondrial segment length with limited fluctuation, indicating stabilized elongation behavior.

Figure 3.41. exhibits a progressive increase in average topology length throughout the simulation. Initially, the average length stabilizes around ~7.6 beads, indicating the formation of short mitochondrial fragments. However, after approximately 800 simulation steps, a gradual elongation is observed, culminating in an average of ~9.3 beads in the latter part of the simulation.

This upward trend suggests a suppression of fission activity or enhanced stability of existing network structures, potentially reflecting Rotenone's known effects on disrupting mitochondrial respiration and promoting oxidative stress, which may impair the energetic requirements for fission.

doi:10.6342/NTU202503815

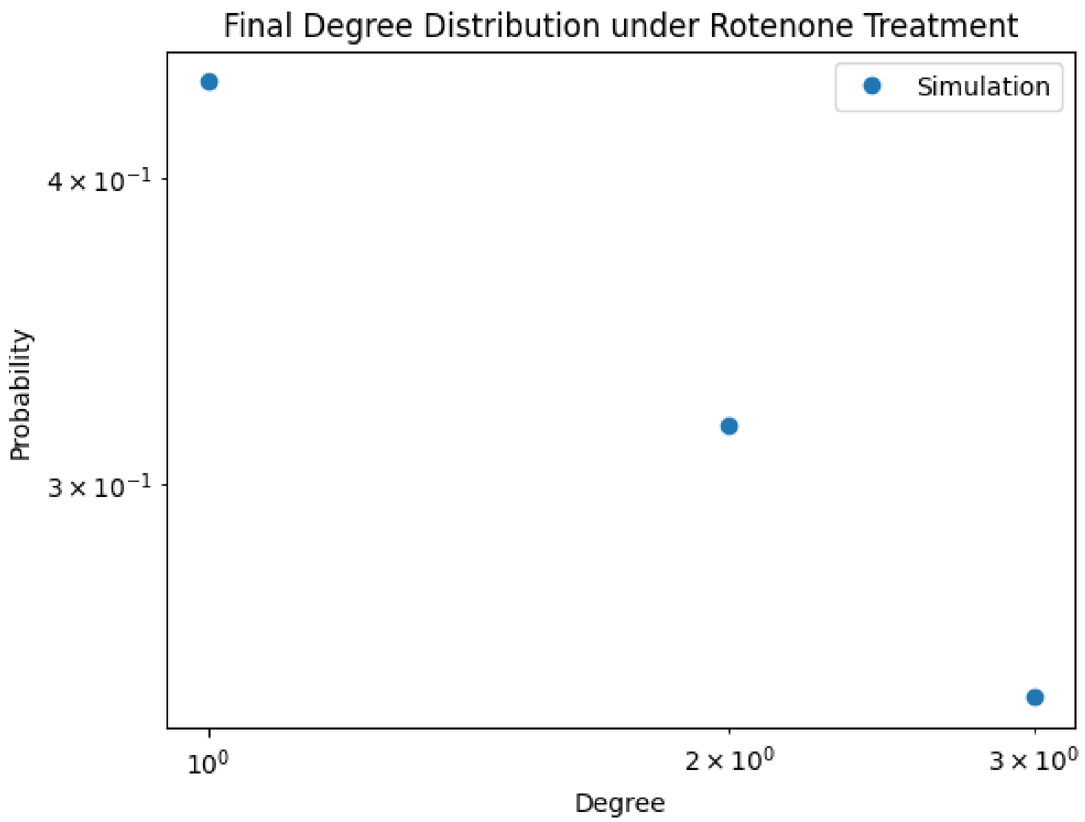


Figure 3.42. The final degree distribution of Ins-1 Cell under Rotenone treatment reveals a predominance of lower-degree nodes with reduced network branching, consistent with fragmented or less connected morphology.

Figure 3.42 reveals a predominance of low-degree nodes, with degree 1 nodes being the most frequent (~44%). The probability declines progressively for higher-degree nodes, with degree 2 nodes comprising ~33%, and degree 3 nodes contributing the least.

This skew toward lower connectivity suggests a fragmented and sparsely connected mitochondrial network, in line with a system undergoing minimal fusion events or incomplete structural recovery. Such a distribution reflects the impairment of network complexity and interconnectivity, which may be attributed to mitochondrial dysfunction induced by Rotenone’s inhibitory effect on complex I of the electron transport chain.

doi:10.6342/NTU202503815

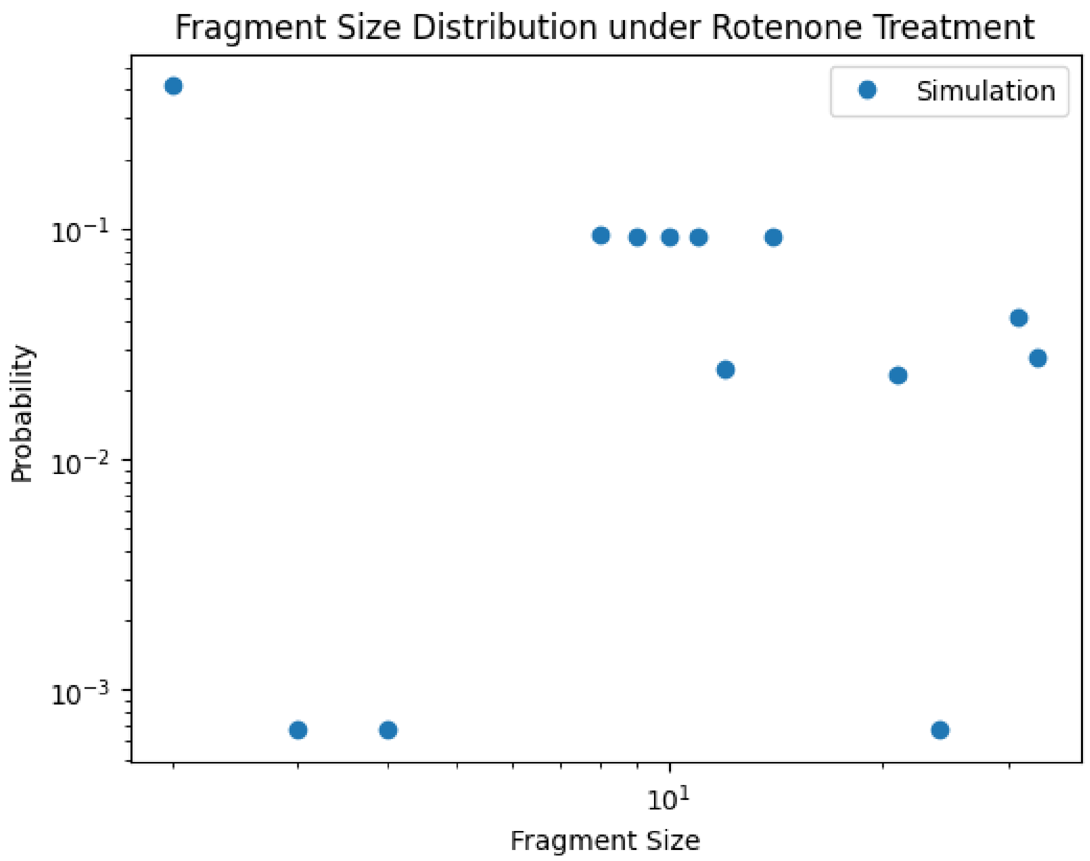


Figure 3.43. Fragment size distribution of Ins-1 Cell under Rotenone treatment on a log-log scale shows a widespread with heavy tail, highlighting the coexistence of small and intermediate-size fragments.

Figure 3.43. displays a bimodal-like pattern with a dominant peak at small fragment sizes and a secondary cluster around intermediate sizes (≈10–20 beads). The highest probability occurs at the smallest fragment size, indicating a prevalence of highly fragmented mitochondrial elements. Beyond this peak, the distribution spreads across a range of larger sizes but with considerably lower probabilities.

This skewed distribution aligns with a system experiencing excessive fission or suppressed fusion, resulting in structural fragmentation. The scarcity of large fragments supports the observation that Rotenone impairs mitochondrial dynamics,

doi:10.6342/NTU202503815

likely through its inhibition of oxidative phosphorylation, which disrupts the bioenergetic balance needed for network maintenance and elongation.

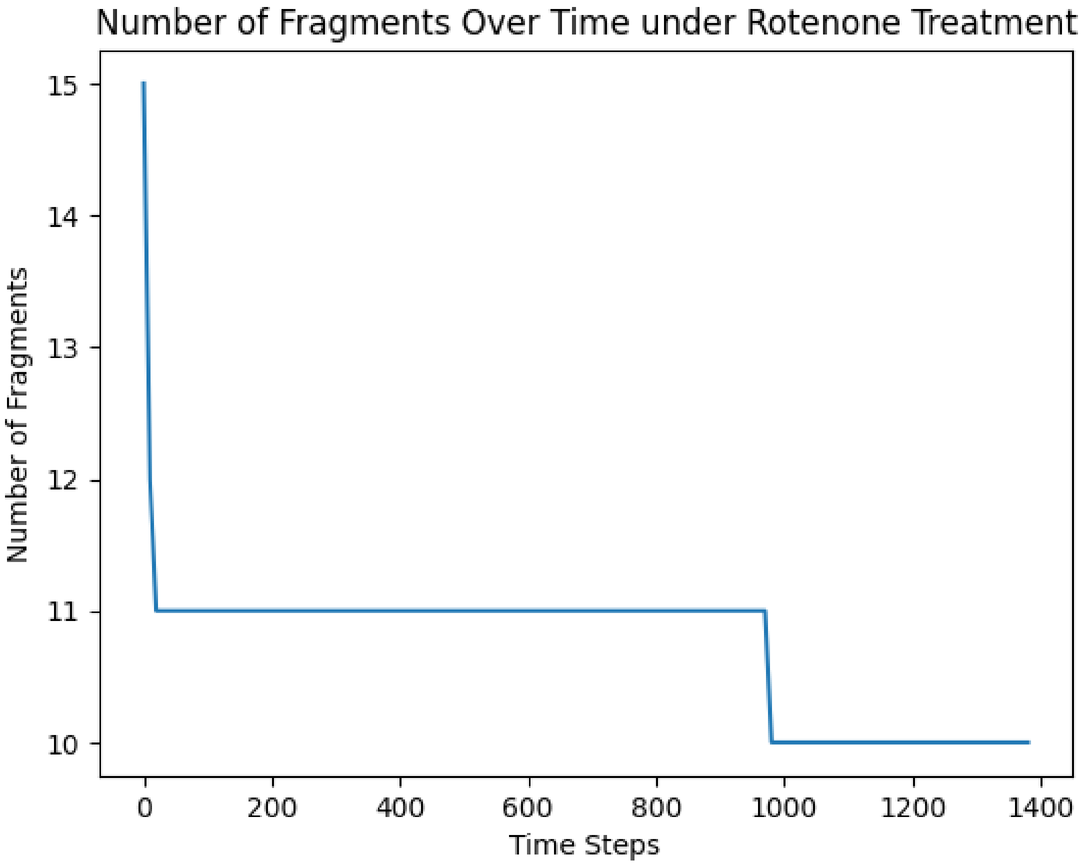


Figure 3.44. The number of fragments over time of Ins-1 Cell under Rotenone treatment demonstrates a rapid decline in the early simulation phase, followed by stabilization, reflecting early fusion-driven coalescence.

Figure 3.44. exhibits an initial sharp decline, followed by a plateau phase with limited fluctuations. The system rapidly transitions from 15 fragments to approximately 11 within the first few steps, and then stabilizes, maintaining a relatively constant fragment count between 9 and 11 for the majority of the simulation.

doi:10.6342/NTU202503815

This behavior indicates a reduction in fission events or a mild increase in fusion efficiency, leading to a slight consolidation of mitochondrial segments. Unlike other stressors that may promote dynamic instability or fragmentation, Rotenone appears to result in a relatively static mitochondrial network after an initial reorganization period.

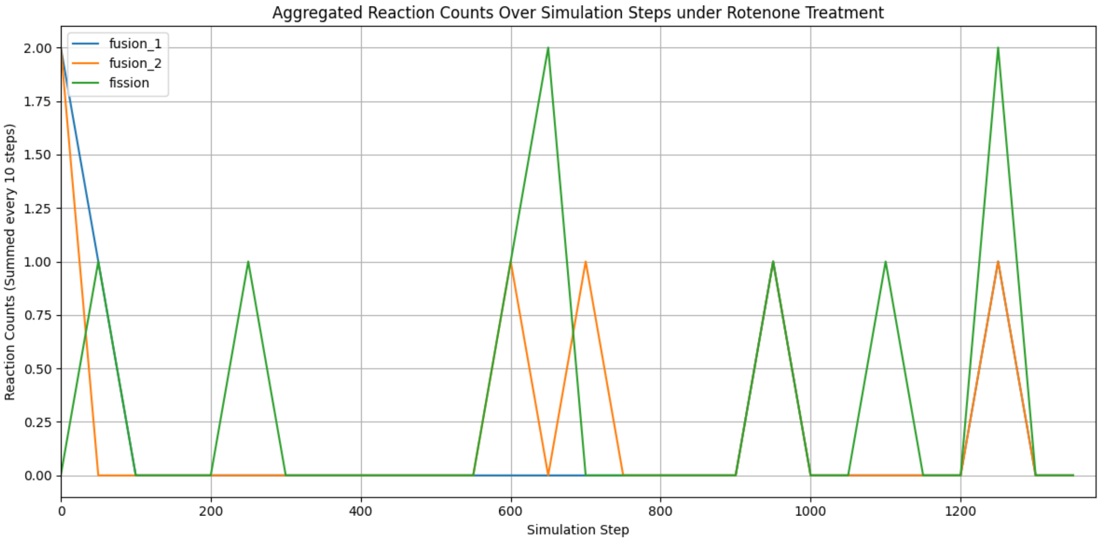


Figure 3.45. Aggregated reaction counts of Ins-1 Cell under Rotenone treatment reveal minimal reaction activity, particularly low fusion and fission frequencies.

Figure 3.45. shows a striking suppression of both fission and fusion events throughout the simulation. Most time intervals exhibit no recorded activity, with only a few sparse reaction spikes—typically of low magnitude (1–2 events per 50 steps)—appearing sporadically across the timeline.

This result suggests a severely diminished dynamic turnover within the mitochondrial network, reflecting a state of reduced remodeling capacity. The minimal activity of both fusion (types 1 and 2) and fission indicates that the network likely stabilizes early and remains static, consistent with the low ATP availability and energy collapse caused by complex I inhibition under Rotenone exposure.

doi:10.6342/NTU202503815

The overall quiescence aligns with the previously observed fragment number plateau and slow evolution of topology length, reinforcing the hypothesis that Rotenone induces a mitochondrial freezing effect in the simulated model.

6. PANC-1 Network Condition (n = 6)

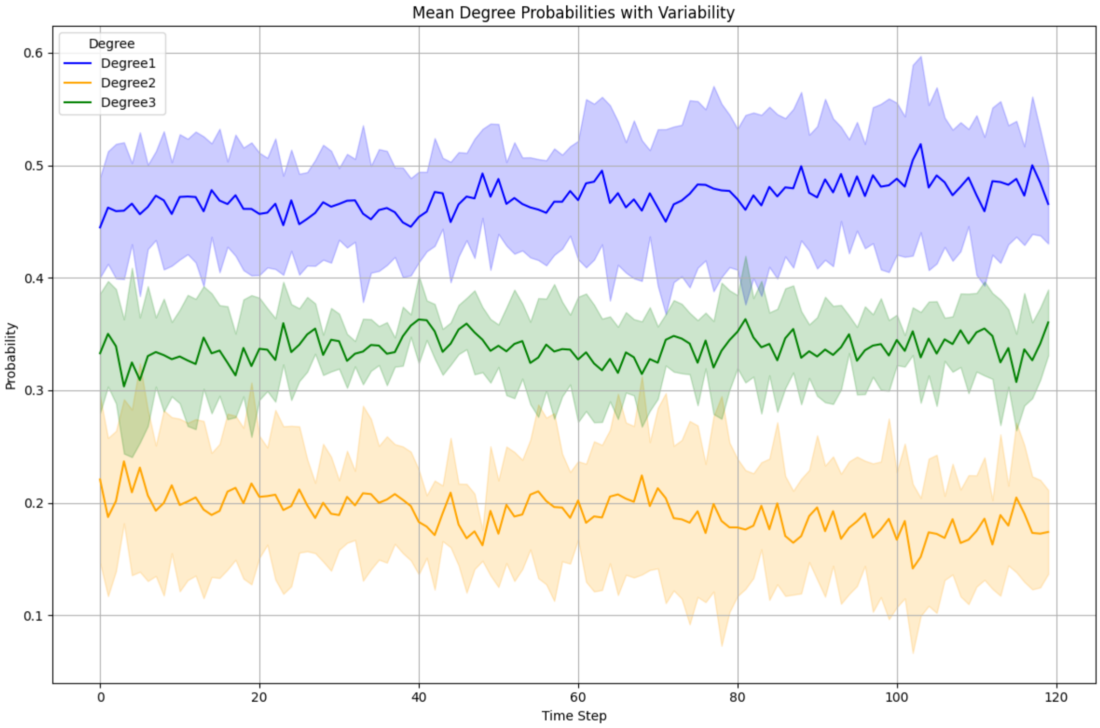


Figure 3.46. Temporal evolution of mean degree distributions for PANC-1 cells.

Degree 1 nodes (blue) remain dominant and stable, while degree 3 nodes (green) consistently surpass degree 2 nodes (orange), indicating persistent network branching. The distribution shows structural stability over time with moderate fragmentation and sustained branching, characteristic of the baseline mitochondrial morphology in PANC-1 cells.

Figure 3.46 displays the time evolution of degree probabilities in PANC-1 cells under baseline conditions. The mitochondrial network exhibits a relatively well-

doi:10.6342/NTU202503815

balanced but structured topology, with three clearly separated degree classes and low overall temporal fluctuation.

Degree 1 nodes (blue), representing terminal ends, dominate throughout the time course with probabilities ranging around 0.45–0.50. This reflects a moderately fragmented state where disconnected tubules are prevalent. Degree 3 nodes (green), associated with branching points, consistently occupy the second-highest proportion, maintaining a stable probability near 0.33. Their prominence suggests that PANC-1 mitochondria possess a relatively higher degree of branching than seen in other cell lines. Degree 2 nodes (orange), corresponding to linear segments, remain the least abundant and show low variability, hovering around 0.20.

This network configuration implies that PANC-1 cells maintain a mixed morphological phenotype: a population of fragmented mitochondria coexisting with moderately complex branches. The low variability in all three-degree classes indicates structural stability over time, and the elevated degree 3 presence may reflect cell-type-specific differences in mitochondrial organization or metabolic demand.

Table 3.7. Adjustable ReaDDy2 parameters for PANC-1

parameter	value	description (concise)
Fusion_probability_1	2	tip-to-tip fusion probability
Fusion_probability_2	5	tip-to-side fusion probability
Fission_base_probability	0.1	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	20	capture radius for fusion_reaction_2

doi:10.6342/NTU202503815

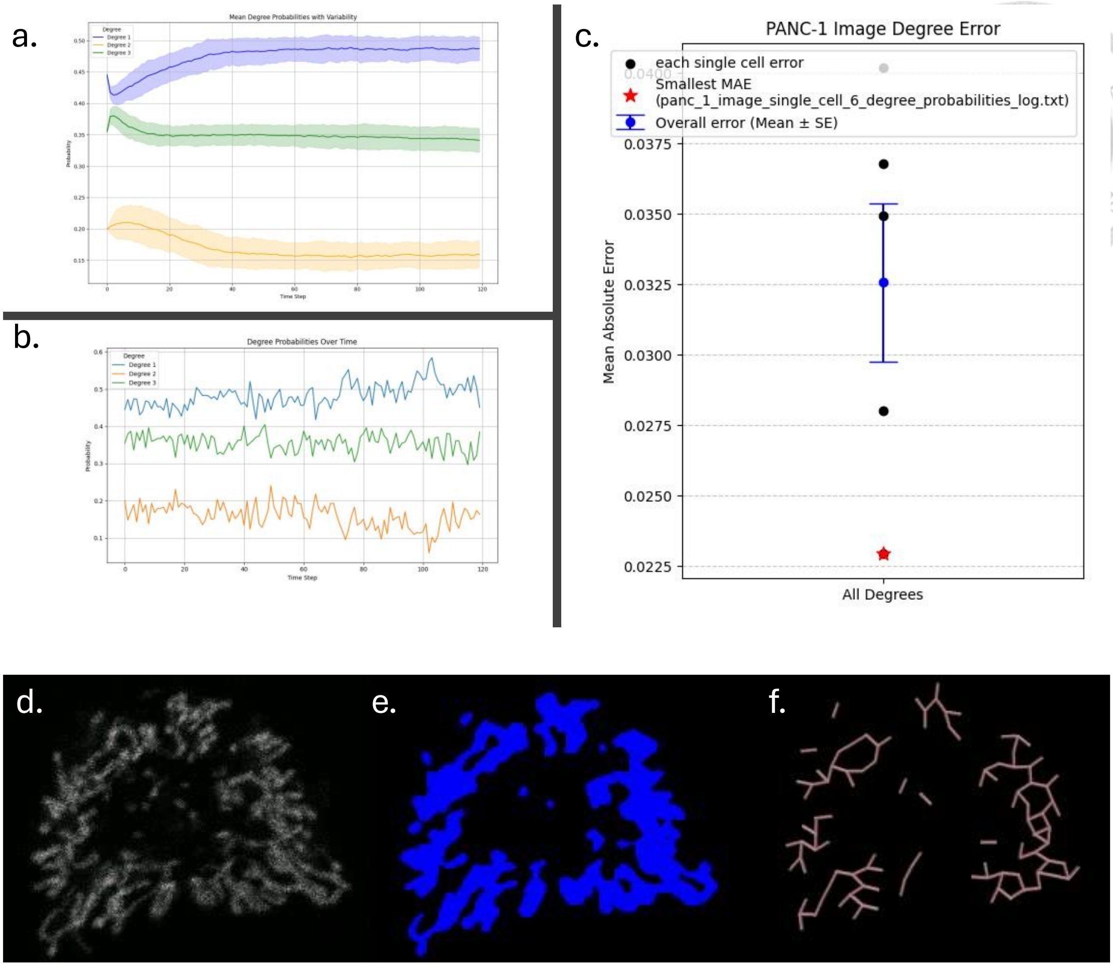


Figure 3.47. PANC-1 condition analysis.

Panels (a–c) present quantitative evaluation and image-derived topology for the best-fit simulation under control conditions in the PANC-1 cell line.

(a) Time-averaged degree probabilities (± SD) from the best-fit simulation, illustrating separation and stability across all degree classes.

(b) Experimental degree probabilities from the single PANC-1 cell that best matches the simulation shown in (a), used for MAE calculation.

(c) Mean absolute error (MAE) across 100 simulation replicates. The red star indicates the best-fit simulation with the lowest MAE, while the blue dot and bar show the population mean ± SE.

(d) Raw fluorescence image of mitochondria from a representative untreated PANC-1 cell.

doi:10.6342/NTU202503815

(e) Corresponding segmentation mask showing mitochondrial regions.

(f) Skeletonized mitochondrial network used for graph extraction and topological analysis.

Under the PANC-1 condition, In Fig. 3.47(a), the best-fit simulation reveals a fragmented but structurally consistent network: degree 1 nodes dominate with a stable probability near 0.48, while degree 3 nodes (branch points) persistently exceed degree 2 (linear segments). This indicates that despite fragmentation, the network retains considerable branching capacity.

Panel (b) shows the degree probability trajectories extracted from the experimental image used for best-fit comparison. The strong alignment between simulation and experiment supports the conclusion that PANC-1 networks exhibit a hybrid phenotype: branched yet non-elongated.

The MAE distribution in Fig. 3.47(c) demonstrates excellent overall model performance, with the best-fit simulation (red star) achieving a low MAE of approximately 0.0225, and the group also remains below 0.035. This narrow error distribution highlights robust reproducibility across independent runs and suggests that the simulation framework effectively captures PANC-1 mitochondrial dynamics under baseline conditions.

Overall, the network morphology in PANC-1 cells reflects a cell-type-specific configuration: fragmented mitochondria with structurally discrete yet branched patterns, potentially shaped by metabolic specialization or cytoskeletal architecture.

doi:10.6342/NTU202503815

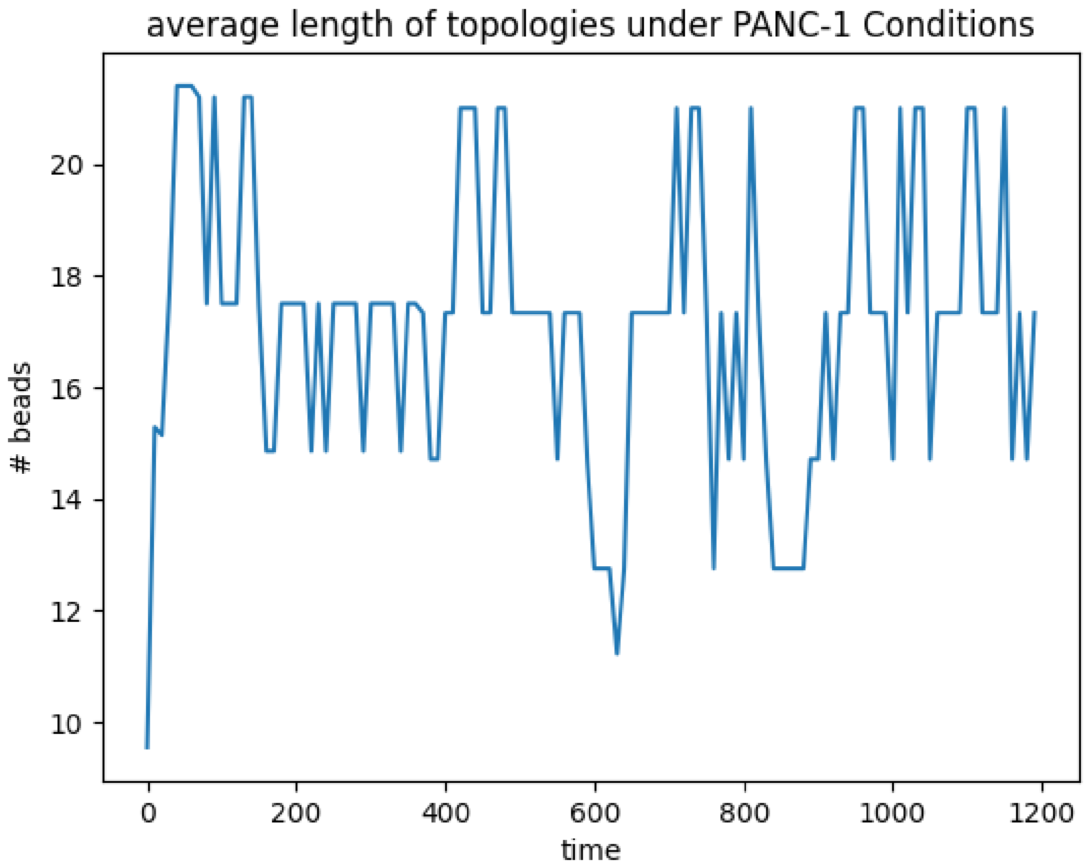


Figure 3.48. Average number of topologies over time of PANC-1 condition, reflecting network elongation and transient fluctuations in filament length.

Figure 3.48. quantified by the number of beads per topology, exhibits high variability over time, indicating a dynamic and fluctuating mitochondrial network. Initially, the network shows a steady elongation phase, reaching a peak average length exceeding 26 beads, which likely reflects an early dominance of fusion activity.

However, this is followed by a pronounced destabilization period marked by frequent oscillations and sharp declines in length. The system settles into a phase characterized by intermittent contraction and extension cycles, with average lengths fluctuating between 12 and 18 beads. This reflects a highly active remodeling regime, potentially mirroring cell-type-specific mitochondrial dynamics inherent to PANC-1 cells, such as elevated basal fusion-fission turnover or sensitivity to fluctuating metabolic cues.

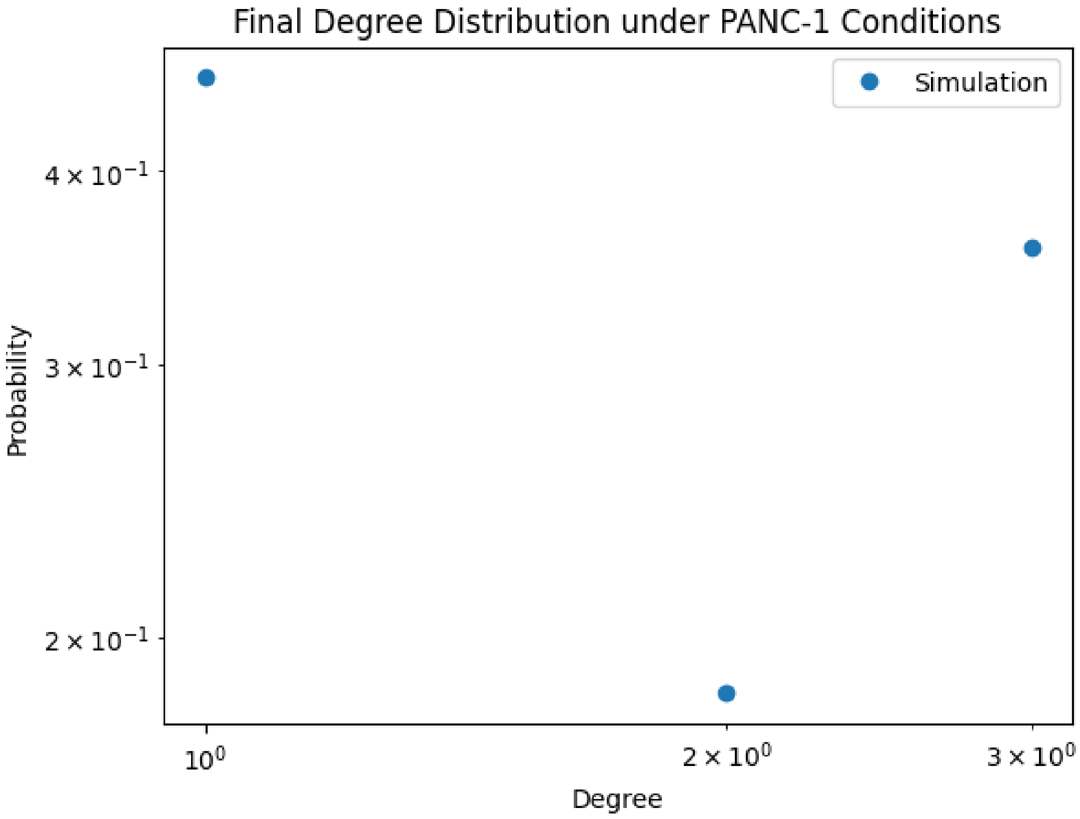


Figure 3.49. Final degree distribution of PANC-1 condition plotted on a log-log scale, indicating dominant degree classes within the simulated network topology.

Figure 3.49. displays a broader and more heterogeneous topology compared to control or drug-treated states. As shown in the log-log plot, nodes of degree 1 and 3 dominate, each contributing significantly to the overall network architecture, while degree 2 nodes are present but less prevalent. A small fraction of nodes with degree 4 are also observed, indicating the presence of higher-order branching events.

This diverse range of connectivity suggests an environment where complex network structures emerge dynamically, likely driven by frequent remodeling through fusion and fission. The occurrence of degree-4 nodes—though rare—hints at localized

doi:10.6342/NTU202503815

regions of dense mitochondrial interconnectivity, a feature often linked to metabolic adaptation or stress resistance in cancerous cells.

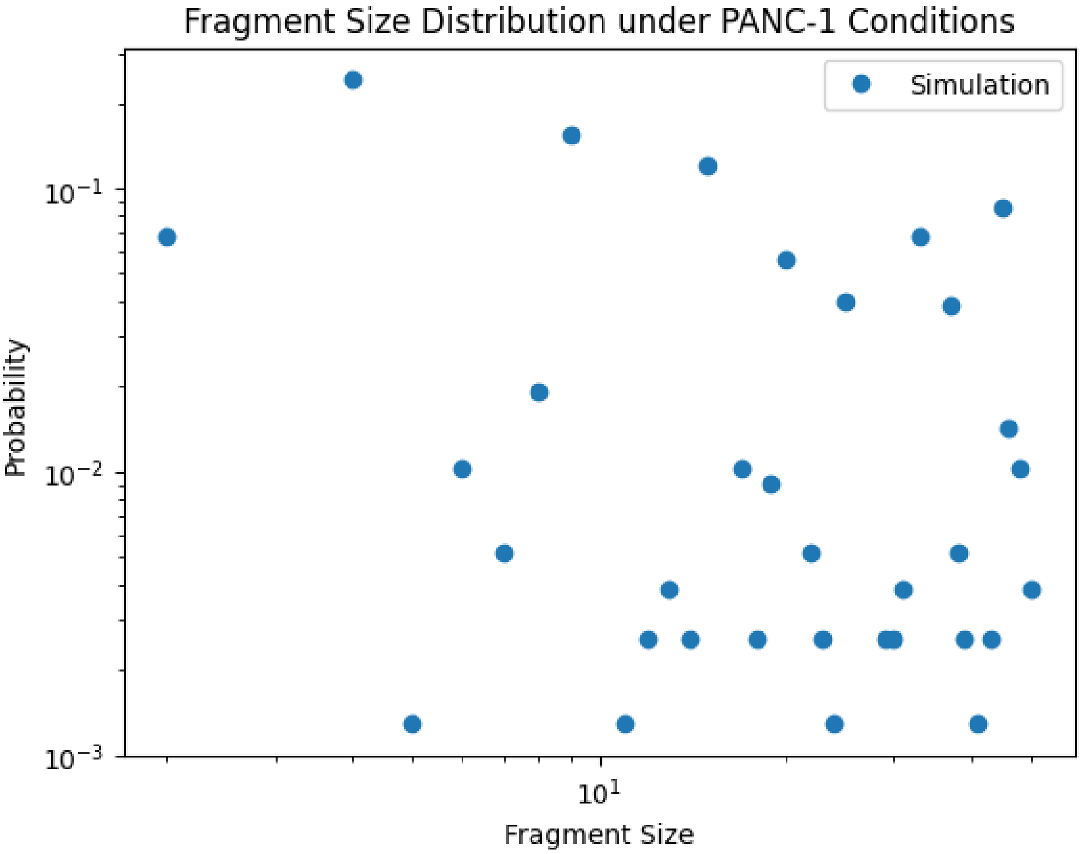


Figure 3.50. Fragment size distribution of PANC-1 condition from the simulation, revealing a broad heterogeneity in mitochondrial segment sizes

Figure 3.50. reveals a broad and dispersed spectrum of mitochondrial segment sizes. The log-log plot demonstrates a heavy-tailed distribution, indicating the presence of both small and large mitochondrial fragments with varying frequencies. Notably, no single fragment size dominates, and the probability is distributed across a wide range, from short fragments (2–5 beads) to extended structures exceeding 20 beads.

doi:10.6342/NTU202503815

This distribution suggests high dynamic heterogeneity, likely reflecting an active balance between mitochondrial fission and fusion processes. The frequent presence of larger fragments may be indicative of elevated fusion activity or suppressed fission, characteristic of cancer cells adapting their mitochondrial networks for metabolic flexibility and resilience.

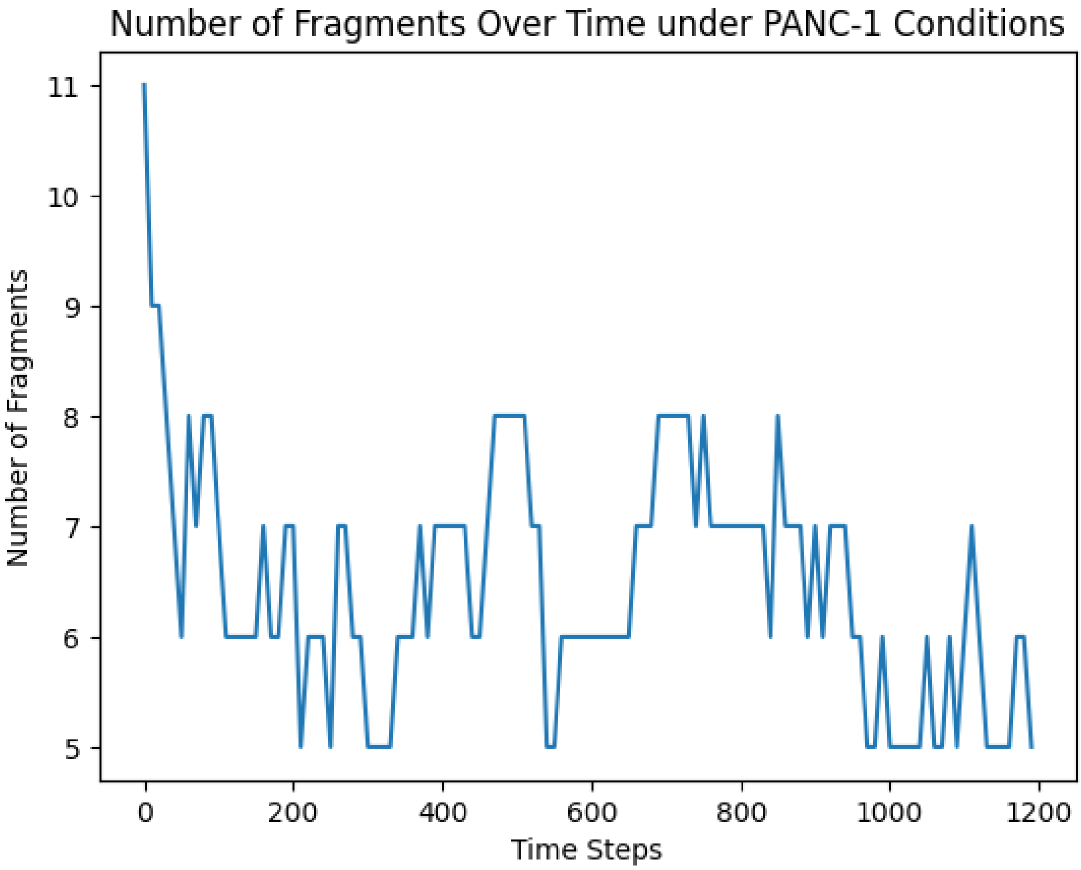


Figure 3.51. Time evolution of the total number of disconnected fragments of PANC-1 condition, indicating partial stabilization following an initial decrease.

Figure 3.51. demonstrates an initial sharp decline from 11 to approximately 6 fragments, followed by sustained fluctuations throughout the simulation. The number of fragments stabilizes within a range of 5 to 9, indicating the system reaches a quasi-steady-state dynamic regime.

doi:10.6342/NTU202503815

This stable fragmentation profile suggests a balanced interplay between fission and fusion events, where neither process dominates persistently. The recurring fluctuations imply ongoing remodeling of the mitochondrial network, which may reflect adaptive bioenergetic regulation in response to the metabolic demands of PANC-1 cells.

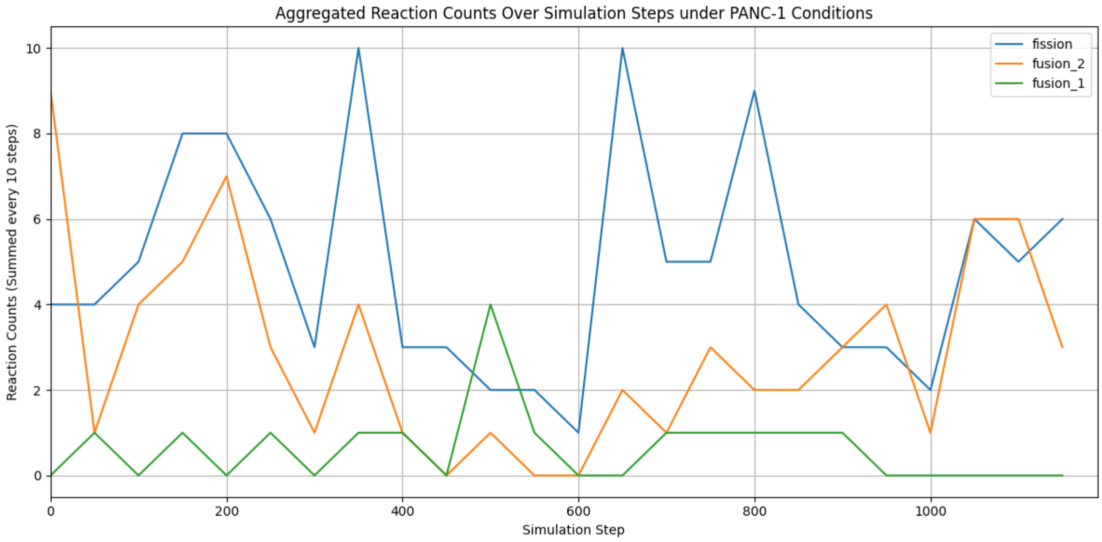


Figure 3.52. Aggregated counts of fusion and fission events over simulation steps of PANC-1 condition, showing active remodeling dynamics with alternating fusion and fission rates.

Figure 3.52. reveals a highly dynamic interplay among mitochondrial fission and fusion events. Fission events (green) occur consistently throughout the simulation, often exceeding both fusion types in frequency, suggesting ongoing network fragmentation pressure.

Fusion-2 reactions (blue), which likely represent fusion between longer fragments, occur more frequently than fusion-1 (orange), consistent with elongation and network consolidation mechanisms. However, neither fusion process fully suppresses fission, leading to a fluctuating balance.

doi:10.6342/NTU202503815

This ongoing competition between fusion and fission maintains a dynamic steady state. The presence of all three reaction types at relatively steady levels suggests a non-terminal, adaptable mitochondrial network, potentially reflecting the enhanced plasticity and metabolic demands of PANC-1 cells.

7. AC-16 Network Condition (n = 15)

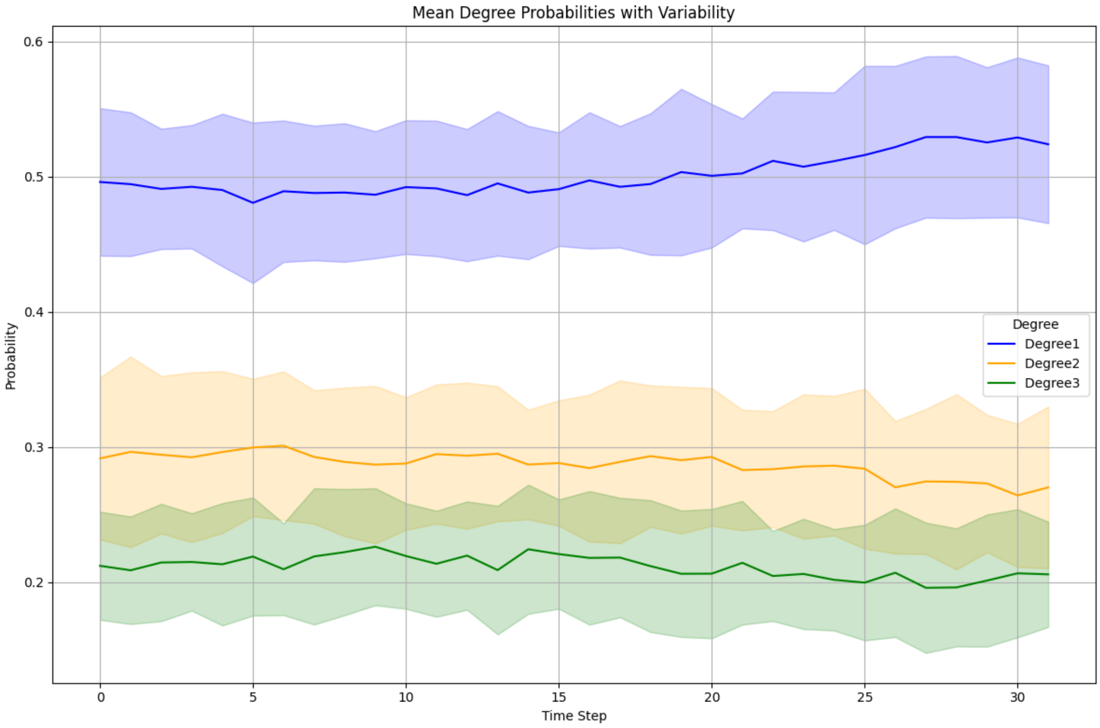


Figure 3.53. Simulated mitochondrial network dynamics in AC-16 cells under control conditions.

Time-averaged degree probabilities (mean ± SD) from the best-fit simulation. Degree 1 nodes dominate the network structure, followed by degree 2 and degree 3 nodes, which remain stable and lower in proportion. The consistent ordering and low variability indicate a steady-state mitochondrial topology with limited remodeling activity.

doi:10.6342/NTU202503815

Figure 3.53 illustrates the simulated temporal evolution of mitochondrial network topology in AC-16 cells under control conditions. The degree probability trajectories derived from the best-fit simulation show a distinct and stable separation across node types.

Degree 1 nodes (terminal ends) exhibit the highest and steadily increasing probabilities, stabilizing around 50–55% over the 30-time steps, indicating a fragmented network architecture. This consistent dominance suggests minimal fusion-driven network elongation or integration.

Degree 2 nodes (linear segments) maintain a moderate presence, approximately 28–30%, with slight downward fluctuation. The absence of significant growth or decline suggests limited remodeling via linear elongation or breakdown.

Degree 3 nodes (branch points) are the least prevalent and remain consistently around 20%, with a narrow range of variability. This low and stable level of branching implies that AC-16 mitochondrial networks under baseline conditions do not engage in significant topological restructuring.

The relatively small standard deviations across all degree classes confirm robust simulation convergence and suggest that the modeled network behavior is representative of a steady-state physiological condition in cardiomyocyte-like cells. This pattern reflects a structurally conservative mitochondrial phenotype, where fragmentation dominates, and branching or elongation events are tightly regulated or infrequent.

Table 3.8. Adjustable ReaDDy2 parameters for AC-16

parameter	**value**	**description (concise)**

Fusion_probability_1	2	tip-to-tip fusion probability
Fusion_probability_2	5	tip-to-side fusion probability
Fission_base_probability	0.1	base probability of fission events
radius_reaction_1	15	capture radius for fusion_reaction_1
radius_reaction_2	20	capture radius for fusion_reaction_2

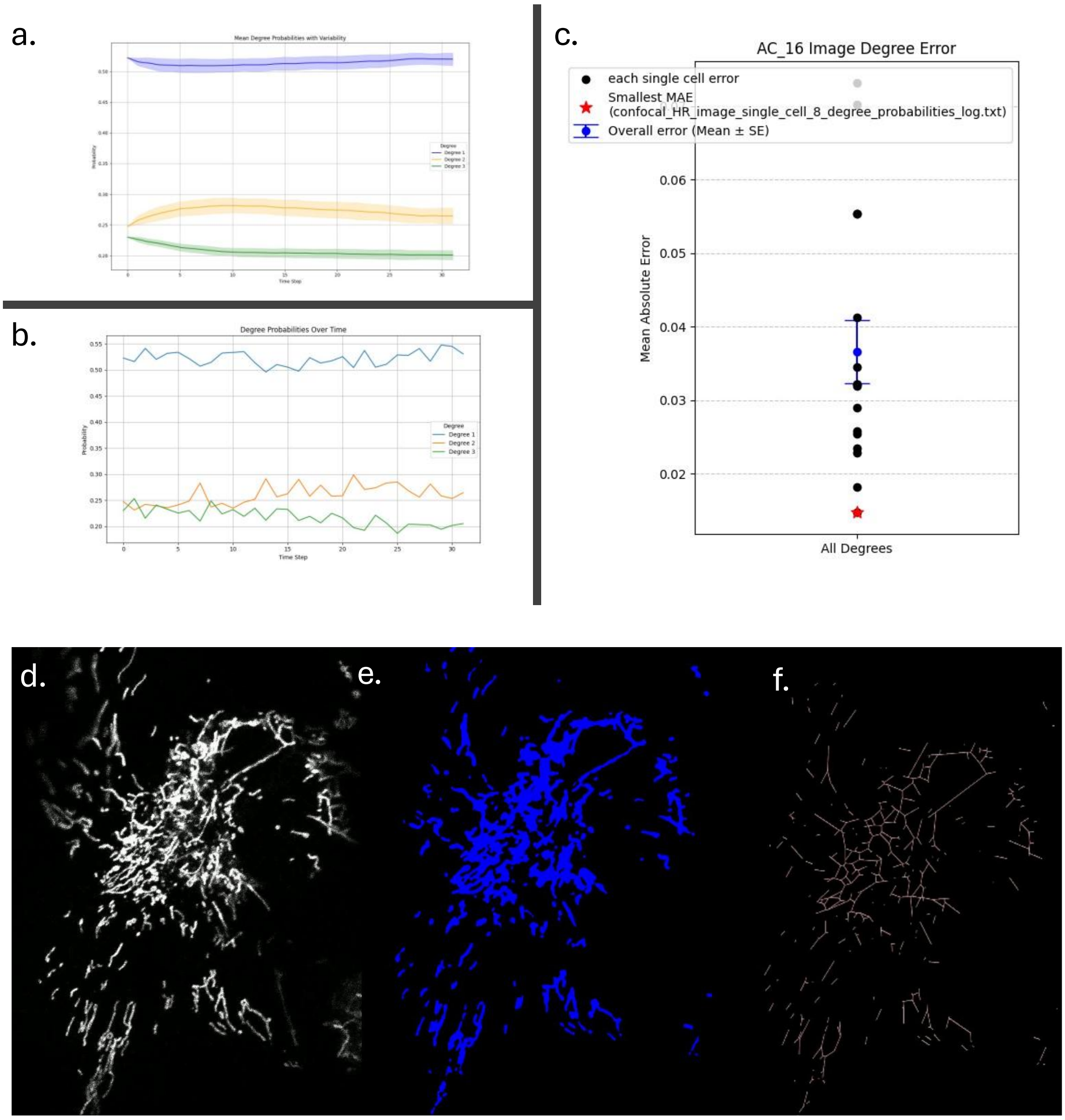


Figure 3.54. AC-16 condition analysis

Panels (a–c) present quantitative and image-derived analyses of mitochondrial networks under the AC-16 cell condition, using the best-fit simulation result for comparison.

doi:10.6342/NTU202503815

(a) Mean degree probabilities (± standard deviation) from 100 simulation replicates show high prevalence of degree 1 nodes and low variability, indicating persistent fragmentation.

(b) Degree probabilities extracted from the real AC-16 image that most closely matches the simulation result (lowest MAE), revealing a topology dominated by terminal ends with sparse branching.

(c) Mean absolute error (MAE) between simulated and experimental degree distributions across all cells. The red star marks the simulation with the lowest MAE (used in a–b), while the blue marker and error bar represent the population mean ± SE.

(d) Raw high-resolution mitochondrial fluorescence image (confocal microscopy).

(e) Segmented mitochondrial regions after preprocessing.

(f) Extracted network structure with nodes and edges, used for simulation input and statistical comparison.

Under AC-16 conditions, the mitochondrial network maintains a highly fragmented architecture with minimal branching or elongation. As shown in fig. 3.54(a), the simulated degree dynamics are characterized by a consistently high prevalence of degree 1 nodes, moderate levels of degree 2 nodes, and a persistent scarcity of degree 3 nodes. The real network topology extracted from the best-fit AC-16 image (fig. 3.54(b)) mirrors this trend, with dominant terminal structures and limited connectivity.

This pattern reflects a stable but structurally constrained mitochondrial state, in which fusion events and network remodeling are either suppressed or ineffective. The overall error between simulation and experimental data remains low (fig. 3.54(c)), with the best-fit replicate exhibiting a minimal MAE of ~0.017. This tight fit confirms the model's capacity to reproduce key structural features of the AC-16 mitochondrial network under baseline conditions.

doi:10.6342/NTU202503815

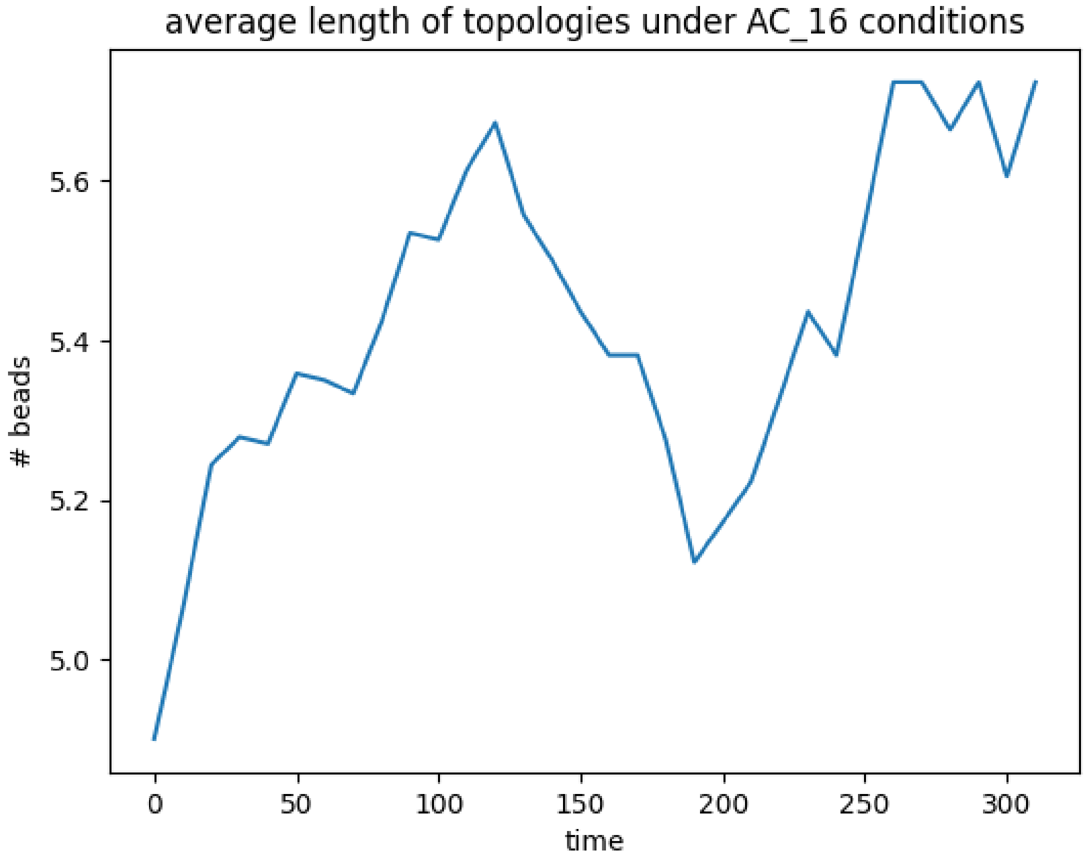


Figure 3.55. Average topologies length over time of AC-16 conditions, measured as the number of connected beads per mitochondrial fragment.

Figure 3.55. shows a gradual increasing trend over time, rising from approximately 4.9 to over 6 beads per structure. This indicates a progressive elongation of mitochondrial segments.

The data exhibits mild fluctuations but maintains a steady upward trajectory, suggesting that fusion events dominate over fission, albeit without abrupt shifts. The absence of sharp fragmentation events points to a moderate but sustained network growth, potentially reflective of restorative or adaptive processes in mitochondrial morphology under this condition.

doi:10.6342/NTU202503815

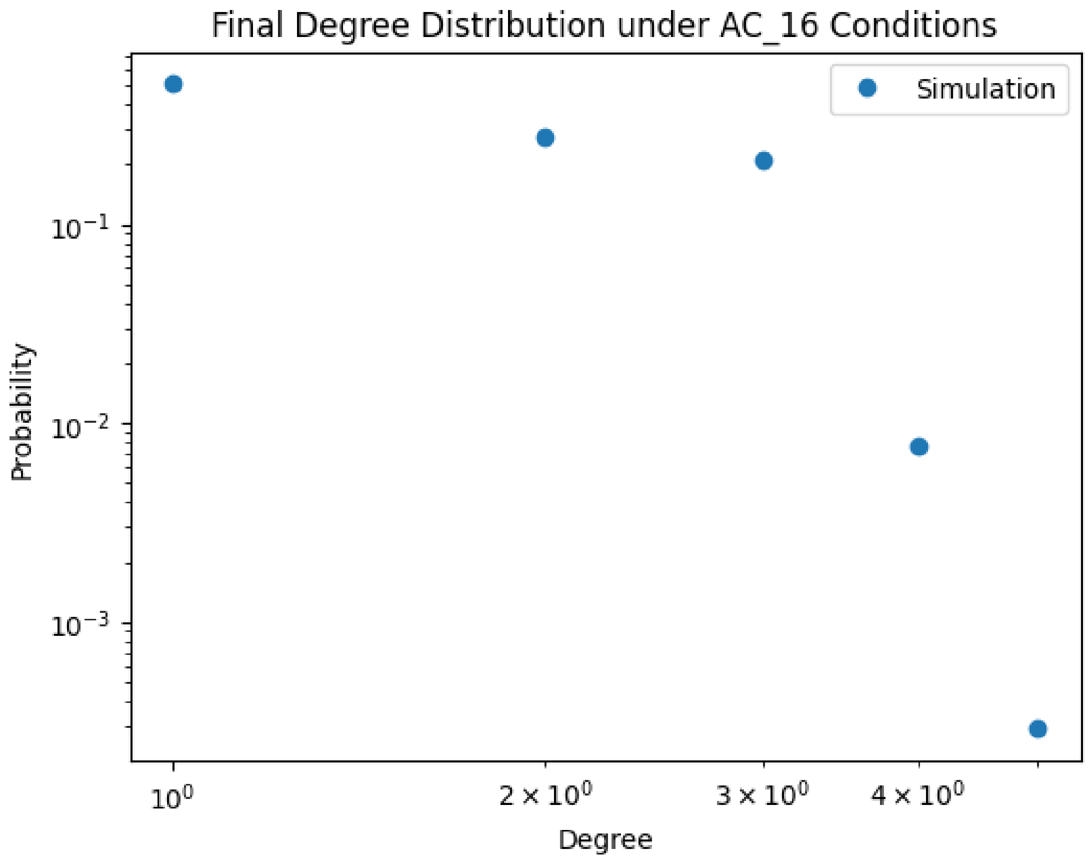


Figure 3.56. Final degree distribution of AC-16 condition plotted in log-log scale, showing the relative probability of encountering nodes of varying connectivity (degree 1–4).

Figure 3.56. reveals a predominance of lower-degree nodes, with degree 1 nodes exhibiting the highest probability. This indicates a prevalence of terminal ends in the network structure, suggesting limited connectivity.

However, a non-negligible fraction of nodes with degrees 2 and 3 are also present, reflecting the existence of linear and branching segments within the network. The probabilities drop sharply for higher degrees (4 and above), indicating that highly connected junctions are rare.

doi:10.6342/NTU202503815

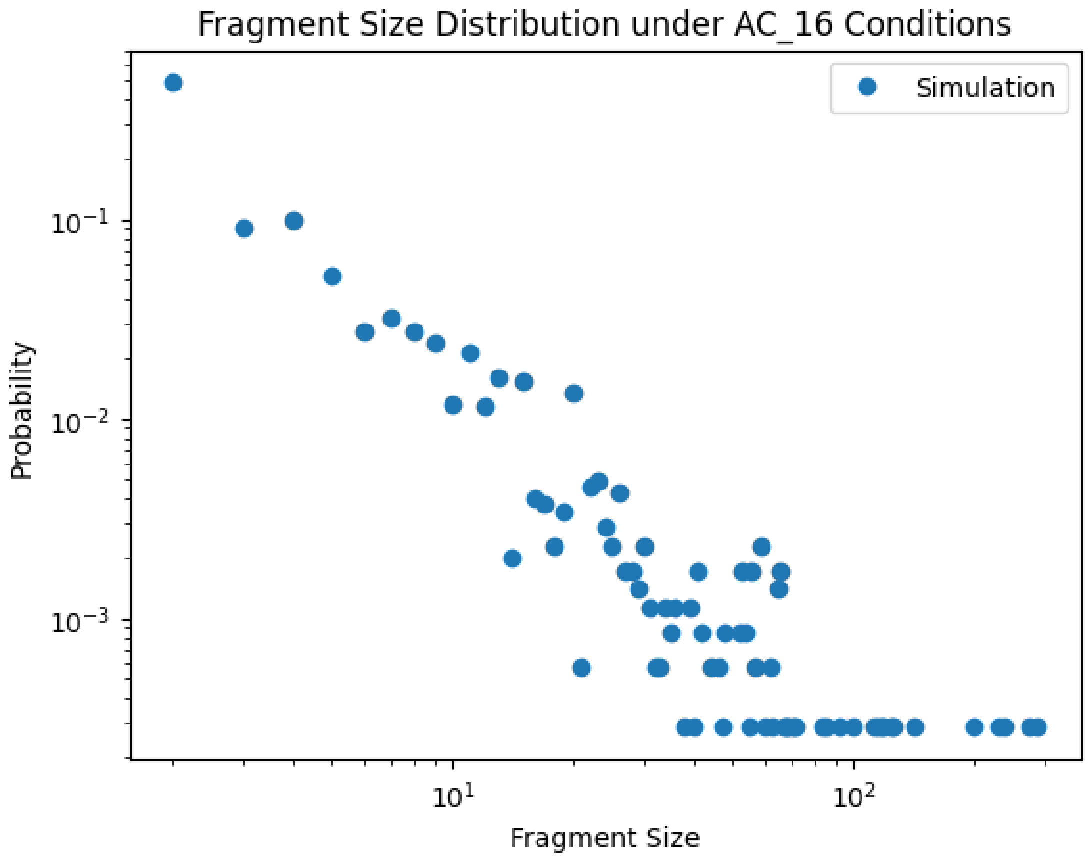


Figure 3.57. Fragment size distribution of AC-16 condition on a log-log scale, reflecting the structural heterogeneity of mitochondrial networks formed during simulation.

Figure 3.57. demonstrates a right-skewed profile with a broad range of fragment sizes. Most fragments are small, with a high probability associated with low fragment sizes, reflecting frequent breakage or limited fusion.

As fragment size increases, the probability declines exponentially, indicating that larger mitochondrial networks are less frequent. However, the distribution extends

doi:10.6342/NTU202503815

over two orders of magnitude, suggesting that a subset of highly fused or elongated structures persists within the population.

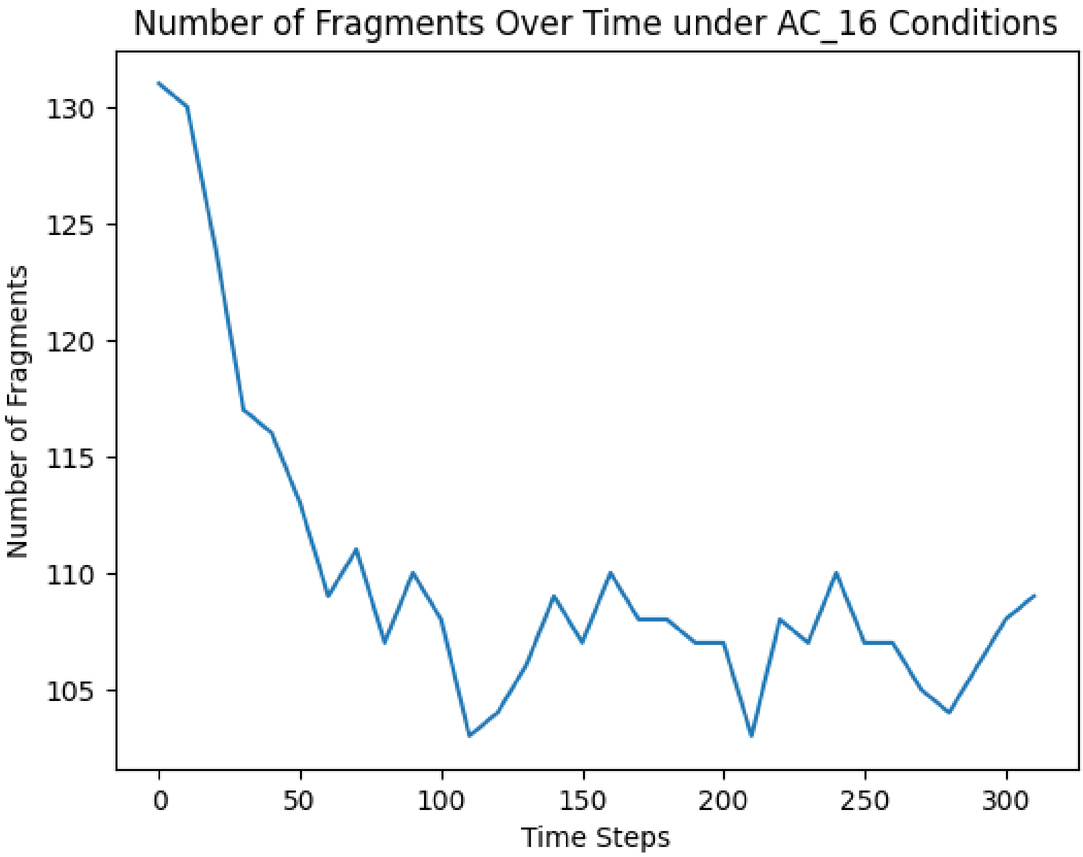


Figure 3.58. Temporal evolution of the number of mitochondrial fragments of AC-16 conditions, indicating a slight decrease followed by stabilization

Figure 3.58. initially declines rapidly, decreasing from over 130 to approximately 105 within the first 100-time steps. This trend suggests an early dominance of fusion events, leading to network consolidation.

After the initial drop, the fragment count exhibits moderate fluctuations between 105 and 120, indicating a dynamic balance between fusion and fission. The absence

doi:10.6342/NTU202503815

of drastic shifts in fragment number after the early phase implies that the system reaches a quasi-steady state, where the network maintains a relatively stable fragmentation profile.

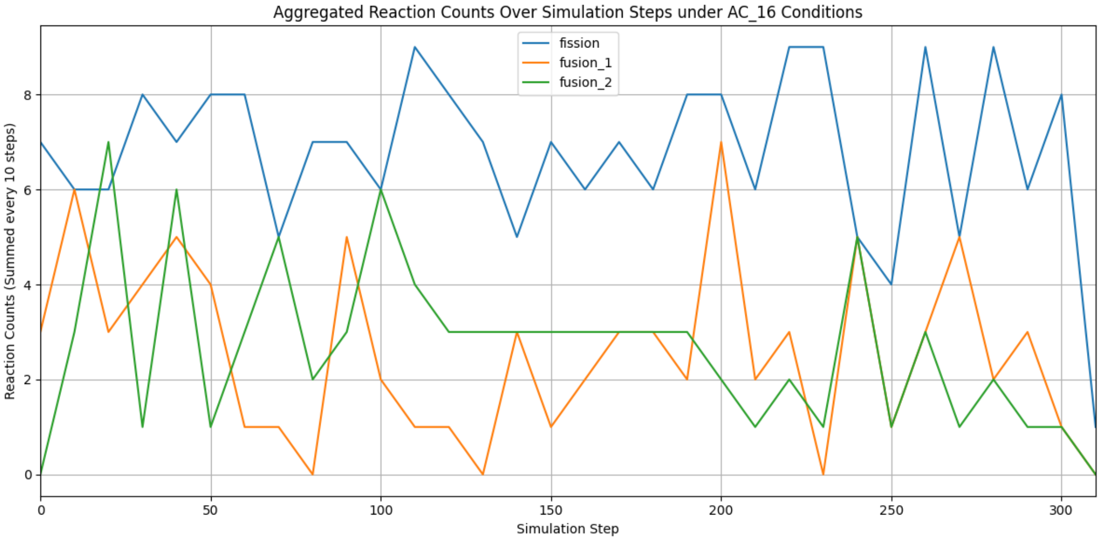


Figure 3.59. Reaction activity over time of AC-16, with fission, fusion_1, and fusion_2 event counts aggregated every 50 simulation steps. Fission events dominate, contributing to persistent fragmentation and shorter structures.

Figure 3.59, fission reactions consistently dominate throughout the simulation, maintaining a high frequency of 6–10 events per aggregated window. This persistent elevation in fission activity supports the elevated fragment count observed in the corresponding morphology data.

In contrast, fusion reactions (both fusion_1 and fusion_2) display lower and more fluctuating frequencies. Fusion_2 events show intermittent bursts (notably around steps 90 and 180) but generally remain under 5 per window. Fusion_1 events are even sparser and irregularly distributed, indicating limited rejoining of small fragments.

doi:10.6342/NTU202503815

The overall pattern reveals that fission outweighs fusion, especially during later phases, reinforcing the fragmentation-prone phenotype of mitochondrial networks under AC-16 exposure. These dynamics suggest that AC-16 disrupts the mitochondrial homeostasis by skewing the fusion–fission balance in favor of fission.

3.2.3 Final Condition-Level Summary

Finally, the MAEs across multiple replicates and experimental treatments are summarized using 01_mito_final_image_error_analyze.py, which aggregates condition-level performance. Each condition—such as control or drug treatment—is associated with a distribution of simulation-image errors. This high-level comparison provides a metric of model generalizability across different biological contexts and offers guidance for parameter tuning or model refinement.

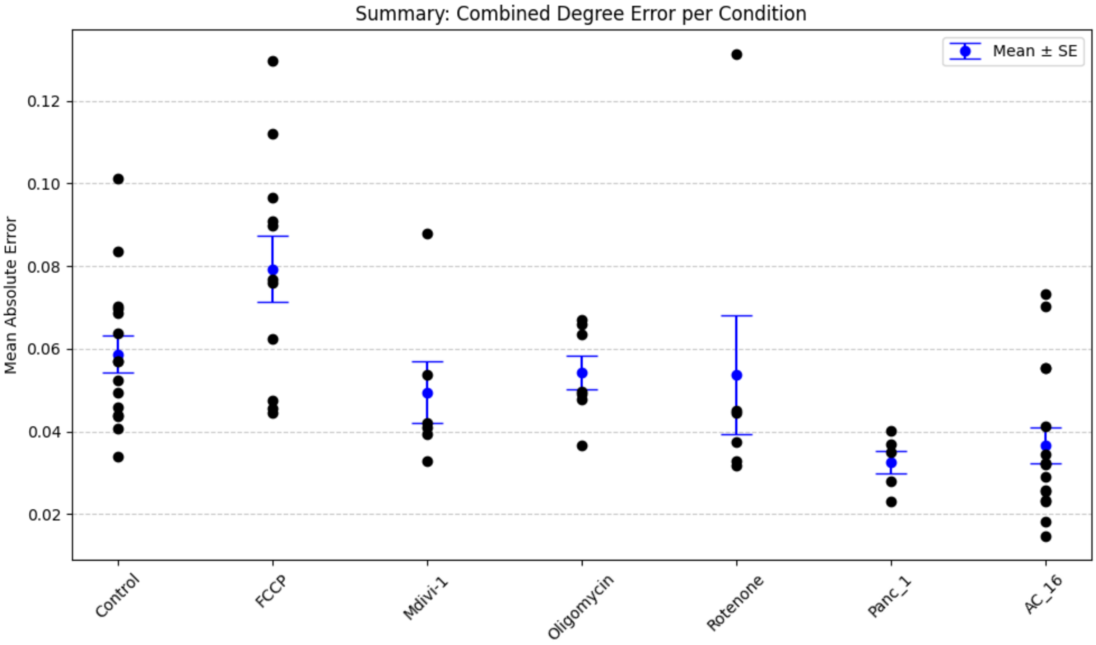


Figure 3.60. Summary of Simulation-Image Degree Error across Experimental Conditions

doi:10.6342/NTU202503815

To quantitatively assess how well the simulation replicates mitochondrial network topology, we computed the mean absolute error (MAE) between the experimental and simulated degree distributions across all cells for each treatment condition.

Among the conditions, PANC-1 yielded the lowest MAE (~0.03 ± SE), indicating the highest fidelity between simulated and experimental topologies. Similarly, Mdivi-1, and AC-16 conditions also demonstrated low error (< 0.045), suggesting that the simulation model effectively captures the structural patterns under these treatments.

In contrast, FCCP conditions exhibited much higher variability and increased error (up to ~0.08–0.13 for some cells), indicating reduced simulation accuracy. This is attributed to the nature of agent-based models and the assumption of the reaction, which are challenging to reproduce under the current reaction scheme or model assumptions, and we will explain in discussion section.

Overall, the error profile highlights both the strength and limitation of the current modeling framework—it reproduces network structures robustly under mild or baseline conditions but diverges under highly disruptive treatments such as FCCP and Rotenone.

This multi-run simulation strategy, combined with image-based validation, provides a robust framework to evaluate both stochastic consistency and biological plausibility of the modeled mitochondrial network dynamics.

3.2.4 Simulation Responses to Drug-Induced Mitochondrial Network.

doi:10.6342/NTU202503815

Furthermore, we summarize the parameters of ins-1 cells under different treatment.

Table 3.9. Treatment-Specific Adjustments of Fusion/Fission Probabilities in ReaDDy2 Simulations

Category:	Fusion_probablity_1	Fusion_probability _2	Fission_base_probablity	Radius_fusion_1	Radius_fusion_2
Control (n = 15)	0.3	0.4	0.15	15	15
FCCP (n = 11)	2	5	0.1	15	20
Mdivi1 (n = 6)	0.6	0.4	0.15	15	15
Oligomycin (n = 7)	0.3	5	0.15	15	15
Rotenone (n = 6)	3	2	0.025	15	20

While table 3.9. lists the key adjusted parameters—namely the two types of fusion probabilities (tip-to-tip and tip-to-side), the base fission probability, and their associated reaction radii—it is not immediately intuitive how these numerical differences map onto phenotypic outcomes.

We also make a ratio of reaction summary to provide insight into the prevailing regime of mitochondrial remodeling

Table 3.10 Normalized Fusion-to-Fission Ratios and Functional Interpretation of Each Treatment Conditions.

Category	Fusion_probability_1 / Fission_base_probability	Fusion_probability_2 / Fission_base_probability
Control (n = 15)	2.00	2.67
FCCP (n = 11)	10.00	50.00
Mdivi1 (n = 6)	4.00	2.67
Oligomycin (n = 7)	2.00	3.33
Rotenone (n = 6)	120.00	80.00

Table 3.10. presents the translated ratios of fusion-to-fission probabilities derived from calibrated ReaDDy2 simulations. By converting the raw values into ratios, the intent is to facilitate cross-condition comparison by placing fusion and fission on a common scale.

However, despite this normalization, several inconsistencies emerge between numerical ratios and biological outcomes. For instance, FCCP and Rotenone, both shown in red, exhibit extremely high fusion-to-fission ratios which typically suggest hyper fusion—but are experimentally characterized by pronounced fragmentation. This apparent contradiction arises because fusion probabilities remain high while fission is suppressed, yet the actual network architecture reflects the failure of fusion due to disrupted membrane potential or ROS-induced instability.

Likewise, Oligomycin (highlighted in orange) presents a more moderate ratio, but its morphological outcome is ambiguous, with mitochondria remaining fragmented yet not severely deteriorated.

doi:10.6342/NTU202503815

These discrepancies underscore an important modeling insight: while ratio-based normalization improves numerical clarity, it can mask qualitative divergence from physiological behavior.

There we find another measurement to verify our prediction of treatment

Table 3.11. Event Frequency per Time Step for Fusion and Fission Events Across treatment conditions.

Category	Fusion_1_Count / Total Time Step	Fusion_2_Count / Total Time Step	Fission_Count / Total Time Step
Control (n = 15)	0.098	0.026	0.156
FCCP (n = 11)	0.027	0.042	0.117
Mdivi1 (n = 6)	0.137	0.054	0.211
Oligomycin (n = 7)	0.053	0.004	0.166
Rotenone (n = 6)	0.004	0.007	0.010

To better capture the dynamic behavior of mitochondrial remodeling, table 3. 11. summarizes the absolute frequency of each event type—tip-to-tip fusion (fusion_1), tip-to-side fusion (fusion_2), and fission—normalized by total simulation steps for each treatment condition.

Unlike the previous ratio-based comparisons (e.g., fusion probability divided by fission probability), which yielded misleading or inflated values (e.g., fusion/fission ratios >100 in stress groups), this count-based approach reflects realized behavior over time and enables direct, interpretable comparisons across conditions.

doi:10.6342/NTU202503815

Finally, we also go through the entire analysis of different types of cells

Table 3.12 Cell-Specific Adjustments of Fusion/Fission Probabilities in ReaDDy2 Simulations

Category:	Fusion_ probability _1	Fusion_ probability _2	Fission_base_probability	Radius_fusion_1	Radius_fusion_2
Ins-1(n = 15)	0.3	0.4	0.15	15	15
Panc-1 (n = 6)	0.02	1	0.05	15	20
AC-16 (n = 15)	0.05	0.1	1	15	20

Same as before, it is not immediately intuitive how these numerical differences map onto phenotypic outcomes according to table 3.12.

Table 3.13 Normalized Fusion-to-Fission Ratios and Functional Interpretation of Each Cell Conditions.

Category:	Fusion_ probability _1 / Fission_ base_probability	Fusion_ probability _2 / Fission_base_ probability
Ins-1 (n = 15)	2.0	2.67

Panc-1 (n = 6)	0.4	20.0
AC-16 (n = 15)	0.05	0.1

Same as before, table 3.13. shows some results are inconsistencies. For instance, Panc-1 displays an anomalously low ratio in the tip-to-tip pathway (0.4) but an extremely high ratio (20.0) in the tip-to-side route, implying an unbalanced or potentially unstable network formation bias. Conversely, AC-16 exhibits uniformly low ratios (<0.1), aligning with its experimentally observed fragmented and static morphology. Although Ins-1 maintains moderate, symmetric ratios (~2), the disparities in Panc-1 highlight parameter mismatches that may stem from under sampled dynamics or cell-line-specific constraints not captured in the current model. As with drug-based comparisons, normalized values help clarify trends but do not always resolve underlying fitting inaccuracies.

Table 3.14. Normalized Event Frequencies per Time Step across Cell Lines

Category:	Fusion_1_Count / Total Time Step	Fusion_2_Count / Total Time Step	Fission_Count/ Total Time Step
Ins-1 (n = 15)	0.098	0.026	0.156
Panc-1 (n = 6)	0.005	0.086	0.107

doi:10.6342/NTU202503815

AC-16 (n = 15)	0.241	0.270	0.733

To more clearly quantify and compare mitochondrial remodeling dynamics across cell lines, table 3.14. show raw event counts were normalized by simulation time steps. This approach eliminates the unit scaling ambiguity observed in previous parameter-ratio tables and provides a direct view of functional remodeling trends.

In this normalized framework, clear distinctions emerge: AC-16 cells exhibit the highest overall activity, with markedly elevated fission (0.733) and balanced contributions from both fusion pathways, aligning with their dense and fragmented network morphology. Ins-1 displays a moderate remodeling regime, dominated by fission (0.156) but with meaningful fusion-1 activity (0.098), reflecting a conditionally stable network. In contrast, Panc-1 shows asymmetric fusion contributions—fusion-2 dominates (0.086) while fusion-1 is nearly absent (0.005)—suggesting an unbalanced and possibly non-canonical remodeling pattern.

Overall, this frequency-based representation offers a clearer and more interpretable view of network dynamics than raw parameter ratios, helping to resolve previously noted inconsistencies in simulation calibration.

doi:10.6342/NTU202503815

4. Chapter 4: Discussion

4.1 Fusion–Fission Reaction Architecture

Our simulation framework emulates mitochondrial network remodeling by combining structural (graph-level) reactions with spatial (distance-triggered) reactions. The network is represented as a dynamic topology whose nodes carry discrete degree-encoded particle types (mito_node_1, _2, _3). Structural reactions operate deterministically on graph logic:

Fusion detection functions scan local neighborhoods and retype participating particles when tip-to-tip or tip-to-side eligibility criteria are met.

Dissociation (fission) functions selectively delete bonds—typically non-terminal internal edges—to create new fragments and "downgrade" node types where degrees fall (e.g., mito_node_3 → mito_node_2).

These graph edits are not purely geometric; they encode biological assumptions about how mitochondrial tubules connect, split, and reclassify their junction states.

Structural logic is coupled with spatial reactions that allow previously disconnected topologies to fuse when reactive particle pairs fall within an interaction radius. Reactions such as fusion_1 and fusion_2 are executed stochastically with user-specified rate constants, introducing a proximity- and concentration-dependent channel for network growth, branch formation, and loop closure.

By running both structural and spatial channels in tandem, the framework captures a range of mitochondrial behaviors observed in live-cell microscopy: filament elongation, branching, progressive fragmentation under stress, and (when

doi:10.6342/NTU202503815

allowed by parameters) reaggregation toward more reticulated geometries. This dual-layer reaction design mirrors the biological interplay between physical encounter frequency and the internal molecular machinery that governs mitochondrial fusion and fission.

4.2 Parameter Sensitivity and Biological Control Knobs

Model behavior is highly sensitive to a small number of tunable parameters that map cleanly onto interpretable biological or mechanical levers:

Table 4.1. Key Model Parameters and Their Biological Interpretations

Parameter	Controls	Biological/Modeling Interpretation
Base fission rate	Frequency of bond removal	Stress-induced fragmentation; Drp1 activity surrogacy
Fusion rates	Likelihood that eligible pairs fuse	MitoFusin/ OPA1 dependent fusion competence
Interaction radius	Spatial encounter eligibility	eligibility Organelle crowding; effective tether reach
Bond force constant	Segment extensibility	Membrane tension / tether stiffness
Angular constraints	Bending stiffness & branching geometry	Curvature regulation; cristae/outer membrane mechanics proxy

Systematic sweeps over these parameters reproduce qualitatively distinct morphologies—from sparse puncta to branched reticula—matching trends seen across pharmacological perturbations (e.g., fragmentation under FCCP vs.

doi:10.6342/NTU202503815

maintenance/elongation under control). Given the multi-parameter coupling and stochasticity, future work should incorporate automated parameter estimation (Bayesian optimization, likelihood-free inference, or ML-guided search) to more rigorously fit condition-specific datasets.

4.3 Emergent Topological Dynamics

Single-run time series illuminate how local reaction rules scale to global network outcomes. A common trajectory under fragmentation-favoring settings shows:

- Rising degree-1 fractions (terminal tips) as filaments sever.
- Falling degree-3 fractions (branch points) reflecting junction loss.
- Convergent fragment size distributions where many small motifs outcompete large reticula.

Reaction logs reveal bursty temporal structure: fusion and fission events cluster when geometry and concentration briefly align, separated by quiescent intervals where topology is sterically locked or diffusion-limited. Tracking fragment counts, mean component size, and node degree histograms together provides a mechanistic narrative—networks either relax toward a quasi-steady branching frequency or cascade into sustained fragmentation depending on the fusion: fission balance and encounter radius.

4.4 Quantitative Validation Against Image-Derived Data

To benchmark biological realism, we compared simulations to degree distributions extracted from time-lapse microscopy across multiple treatments (Control, Glucose, FCCP, Mdivi-1, etc.). Each simulation condition was run in 100 replicates; per-frame degree probabilities were averaged across replicates and matched to experimental time points. We computed mean absolute error (MAE) trajectories cell-by-cell, then aggregated errors at the condition level.

doi:10.6342/NTU202503815

Agreement was strongest under Control and Glucose conditions, indicating that the core fusion–fission + diffusion machinery is sufficient to recapitulate baseline mitochondrial topology. Larger discrepancies emerged under stronger perturbations (e.g., FCCP, Mdivi-1), suggesting missing biology: altered fusion competence, selective degradation, or metabolic feedback not yet encoded. The validation scripts also produce per-cell "best fit" traces (minimum MAE) that highlight heterogeneity within conditions and guide targeted model refinement.

4.5 Semi-Synthetic Ground-Truth for Tracking Validation

The simulation produces a semi-synthetic ground-truth dataset that looks like microscopy output yet retains complete knowledge of the underlying mitochondrial topology and event history. This capability underpins one of the framework's key strengths (Section 4.5 Strengths #5: robust data to validate tracking algorithms). By starting from rule-driven network dynamics rather than hand-drawn cartoons, we obtain image sequences whose complexity—branching, fragmentation, intermittent fusion—closely resembles real cells while remaining perfectly annotated.

The ground truth is multi-layered. Lightweight XYZ frames allow rapid sanity checks (did the run complete, did reactions fire), but the PDB/PSF exports provide bonded structure and node-degree typing that can be rendered in VMD to produce realistic image stacks for algorithm testing. Crucially, every simulation step also writes reaction count trajectories (how many fusions/fissions per time point) and event-level fusion/fission logs with particle/topology IDs and spatial coordinates. Together, these outputs let us score a tracker along complementary axes: (i) temporal detection accuracy (does the algorithm call the right number of events at the right times?), (ii) spatial localization error relative to logged coordinates, and (iii) topology recovery fidelity (degree distributions reconstructed from tracked skeletons vs. known simulation graph; comparable to the MAE analyses in Section 4.4).

doi:10.6342/NTU202503815

Because the imagery is generated from known structure, any discrepancy between tracker output and ground truth reflects limitations in the analysis pipeline rather than biological variability. This gives a high-confidence environment for debugging segmentation thresholds, branch-merging heuristics, and event-linking logic before applying the pipeline to experimental movies. The same dataset can be shared openly, enabling reproducible cross-group benchmarking and parameter sweeps (e.g., varying noise, blur, or sampling frequency) to probe algorithm robustness. In this way, the semi-synthetic ground-truth resource closes the loop between the emergent simulation behavior described in Section 4.3 and the quantitative comparisons to real data in Section 4.4, strengthening the overall validation chain.

4.6 Contributions and Limitations

4.6.1 Contributions

The simulation environment developed in this study provides a flexible and biologically informed platform for exploring mitochondrial network dynamics. Because simulation initial conditions can be seeded from microscopy-derived skeletons, the in-silico networks retain a realistic structural context, narrowing the gap between experimental observation and computational modeling. This imaged driven linkage reduces arbitrary initialization bias and allows condition-specific modeling directly from observed cell states.

A second strength lies in the transparency of the rule set. All biological assumptions governing fusion eligibility, fission logic, node retyping, and bond mechanics are encoded in explicit reaction functions that can be inspected, version-controlled, and systematically modified. This clarity supports hypothesis testing: specific molecular hypotheses (e.g., impaired fusion under drug treatment) can be expressed as rate changes, eligibility filters, or altered retyping behavior and then evaluated against data.

doi:10.6342/NTU202503815

The framework is also highly adaptable. Modest parameter adjustments—fusion and fission rates, interaction radius, bond force constants, angular stiffness—produce qualitatively different morphologies, from highly reticulated networks to fragmented puncta. This responsiveness enables efficient parameter sweeps and facilitates condition-wise comparisons across pharmacological or genetic perturbations.

Quantitative comparison to experimental datasets shows that, when appropriately tuned, the model reproduces baseline metrics with good fidelity. Degree distributions, segment length statistics, and fusion: fission ratios fall within experimental variation under Control and other moderately perturbed conditions, indicating that the core mechanisms encoded in the model capture essential features of unstressed mitochondrial remodeling.

Finally, the simulation generates rich, multi-layer outputs—coordinate trajectories, bonded structural files (PDB/PSF), per-step reaction counts, and spatially resolved fusion/fission event logs—that together constitute a semi-synthetic ground-truth resource for validating tracking and skeletonization algorithms (see Section 4.5). Because the underlying topology and event history are known exactly, discrepancies between analysis output and truth can be attributed to algorithm performance rather than biological uncertainty. Coupled with the open Python code and parameter files, this ecosystem supports reproducible benchmarking and community extension.

4.6.2 Limitations

Table 4.2. Current Model Limitations and Future Considerations

Limitation Category	Description
Topological Constraints	Cannot simulate degree-2 to degree-3 transitions (e.g., under FCCP treatment).

Kinetic Limitations	Unable to capture large-scale mass movement of mitochondrial components.
Dimensional Restriction	Model validation is limited to 2D experimental data only.
Parameter Mapping	Simulated fusion/fission rates do not always match biologically observed rates.
Particle Lifecycle	No support for mitophagy or biogenesis; particles cannot be added or removed.
Scaling Artifacts	Requires parameter tuning to avoid static networks in large spatial domains.
Segmentation Caution	DIC masks are required for cell segmentation; mitochondrial smoothing must be avoided to preserve network details.
Spatial Accuracy	Spatial validation should include pixel-wise graph-image comparisons.
Nucleus Representation	Nucleus should be modeled as a static “mega particle” to influence network behavior.
Reaction Documentation	A full workflow for defining and explaining structural/spatial reactions is needed.

To provide a comprehensive evaluation of the simulation framework, it is essential to systematically address its current limitations and areas requiring refinement. While the model successfully recapitulates major topological trends and treatment-specific mitochondrial behaviors, several intrinsic constraints remain. These include limitations in reaction rule expressiveness, spatial fidelity, scalability, and biological realism. The following table summarizes key technical and conceptual limitations, along with implications for future model development and experimental integration.

1. Topological Constraints — Degree Conversion Limitations

The first constraint concerns node-degree bookkeeping under the implemented tip-to-side fusion rule. In Figure 4.1., when a degree-1 terminal tip attaches to a degree-2 side node, the side node becomes degree-3 while the former tip becomes degree-2; one degree-2 is lost and one is gained, yielding no net reduction in the global degree-2 population. Consequently, experimental trajectories show a coordinated decline in degree-2 nodes alongside an increase in higher-order junctions cannot be reproduced solely through repeated tip-to-side fusion events. Additional structural processes—such as side-to-side consolidation or post-fusion pruning that consumes degree-2 intermediates—would be needed to match those patterns.

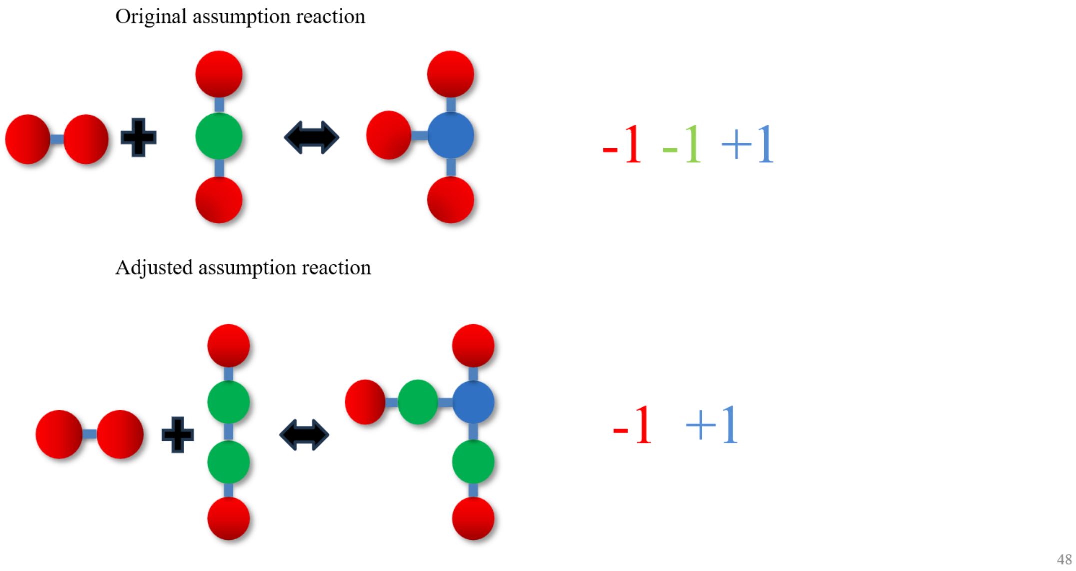


Figure 4.1. Comparison of original and adjusted fusion assumptions in node-degree transitions.

In the original assumption (top), tip-to-side fusion converts a degree-1 tip and a degree-2 node into a new degree-3 node, effectively shifting one degree-1 and one degree-2 into a new configuration without reducing overall degree-2 prevalence. In contrast, the adjusted assumption (bottom) introduces a more complex fusion scheme that directly removes a degree-1 node and consolidates the structure, enabling a net loss of degree-2 nodes and better alignment with experimental observations of coordinated topological remodeling.

2. Kinetic Limitations — Immobilization of Large Components

A second limitation arises from the mechanical treatment of bonded particles. Each particle carries its own stochastic motion; when many are linked into an extended topology, internal spring forces and opposing random displacements tend to cancel at the level of the whole object. The effective center-of-mass diffusion of large

doi:10.6342/NTU202503815

components is therefore markedly reduced, and extended networks can appear unrealistically stationary. This "stand-still" artifact becomes especially apparent in larger simulation domains, where biological mitochondria would normally translate or drift through the cytoplasm.

3. Spatial Dimensionality — Restricted to 2D Simulations

The framework is currently restricted to quasi-two-dimensional geometries, chosen to reflect the projected nature of most fluorescence microscopy datasets and to simplify rendering. While this design is sufficient for validating 2D tracking pipelines, it limits biological realism by underrepresenting vertical stacking, out-of-plane reactions, and 3D spatial exclusions. Extension to full 3D initialization and topology evolution remains a critical direction for future development.

4. Scaling Sensitivity — Fixed Parameter Effects Across Domain Sizes

Scaling also introduces challenges. Simply enlarging the simulation box without tuning encounter radii, reaction propensities, or diffusion constants leads to reduced encounter frequency and worsens the immobilization of large components. As demonstrated in AC-16 simulations, fixed-parameter schemes can produce static network behavior in large regions. Implementing scale-aware strategies, such as density-normalized reaction probabilities or incorporating active transport, will be necessary for realistic modeling at whole-cell resolution.

5. Reaction Rate Discrepancies — Parameter Misalignment with Experimental Rates

In some cases, the event frequencies derived from simulations diverge from expected biological rates. For example, certain conditions yield improbably high fusion-to-fission ratios or inverted network outcomes when compared to experimental data. This suggests that the current fusion/fission rules and tuning protocols may not fully capture condition-specific mitochondrial kinetics, highlighting a need for more mechanistically grounded parameterization.

6. Missing Mitochondrial Turnover — No Biogenesis or Mitophagy

doi:10.6342/NTU202503815

The model does not currently support the creation or removal of mitochondrial particles during runtime. This limits its application to steady-state conditions and prevents simulation of mitochondrial turnover mechanisms such as biogenesis, mitophagy, or stress-induced degradation. Extending the reaction scheme to allow for dynamic particle birth and death will be essential for modeling long-term mitochondrial maintenance.

7. Image Processing Limitations — Cell Masking and Boundary Effects

The simulation environment relies on DIC-based segmentation to define cellular boundaries, which are subsequently applied to fluorescence data. Inaccuracies in segmentation can lead to artifacts in confinement behavior or simulation initialization. Moreover, cellular features such as the nucleus are not explicitly represented and may need to be treated as inert or repulsive "mega-particles" to improve spatial accuracy.

8. Preprocessing Artifacts — Loss of Fine Network Features

Skeleton smoothing steps in the image-processing pipeline can remove small branches or subtle network undulations, potentially biasing the input topology. While smoothing may help reduce noise and clarify connectivity, it risks eliminating important structural heterogeneity, especially near filament termini or in densely packed regions. Strategies to denoise without oversimplifying remain a key area for refinement.

9. Spatial Validation — Lack of Pixel-Level Similarity Metrics

Although topological degree distributions provide one layer of validation, the spatial fidelity between simulated and real networks has not yet been quantified. A useful approach would involve panelizing both simulated and experimental networks and computing similarity scores (e.g., Dice coefficient or IoU). Such metrics would help assess how well the simulated network replicates real mitochondrial architecture beyond node statistics.

10. Unaccounted Structures — Nuclear Exclusion Not Modeled

doi:10.6342/NTU202503815

The current simulation does not explicitly account for the presence of the nucleus or other large subcellular structures, which act as physical obstacles. Simulating these as repulsive or static boundaries could better constrain particle movement and prevent unphysical overlaps, particularly in central cytoplasmic zones.

11. Missing Model Visualization — Reaction Rule Workflow Not Illustrated

Finally, the framework lacks a clear schematic or flowchart that communicates how reactions are triggered, prioritized, or executed in each simulation step. Including such a visual overview would improve transparency and reproducibility, particularly for readers unfamiliar with the rule-based structure of ReaDDy.

doi:10.6342/NTU202503815

5. Chapter 5: Conclusion and Future Work

5.1 Overall Conclusions

This thesis introduced a computational framework for simulating mitochondrial network dynamics that unifies graph-based structural reactions with spatial, proximity-driven interactions. By encoding fusion eligibility, selective internal bond fission, and node retyping rules directly on a dynamic topology—and coupling these with distance-dependent stochastic fusion channels—the model generates time-evolving filament networks that reproduce hallmark mitochondrial behaviors, including tip-to-tip reconnection, tip-to-side branch formation, and progressive fragmentation. Parameterized mechanical elements (bond stiffness, angular constraints) further shape filament geometry, allowing the emergence of loop closure, branch thickening, or collapse into punctate fragments under fragmentation-biased regimes.

Quantitative comparison to microscopy-derived skeleton graphs demonstrated that the framework could recover condition-specific mitochondrial phenotypes. When tuned to baseline or moderately perturbed cellular states, simulated degree distributions, segment length statistics, and fusion: fission ratios closely tracked experimental measurements. Multi-run ensemble averaging reduced stochastic noise and enabled statistical error assessment across conditions, revealing robust agreement under Control and Glucose while exposing systematic deviations under strong pharmacological perturbations (e.g., FCCP, Mdivi-1). These deviations are biologically informative: they point to processes—altered fusion machinery, selective degradation, metabolic feedback—not yet represented in the current rule set and therefore guide targeted model extension.

Beyond reproducing individual trajectories, the framework contributes a reproducible analytic workflow. Simulation outputs are exported in formats that bridge visualization, quantitative analysis, and algorithm testing: lightweight XYZ coordinates for quick integrity checks; bonded PDB/PSF structures suitable for

doi:10.6342/NTU202503815

high-fidelity VMD rendering; and complete temporal logs of fusion and fission events. As shown in Chapter 4, these exports support direct frame-wise comparison to experimental data and enable a semi-synthetic ground-truth resource against which image-analysis pipelines can be objectively validated. Together, these capabilities position the framework as a versatile tool for probing mitochondrial responses to genetic, metabolic, or pharmacological modulation.

The work also surfaced several modeling boundaries. Degree bookkeeping under the current tip-to-side fusion scheme conserves the global count of degree-2 nodes, limiting the model's ability to capture experimental trajectories in which degree-2 frequencies decline as higher-order junctions form. Likewise, emergent kinetic cancellation within large, bonded components suppresses center-of-mass motion, producing "stand-still" aggregates unless mobility is re-scaled. Finally, the quasi-two-dimensional simulation geometry, while matched to most projected microscopy datasets, underrepresents vertical intertwining and out-of-plane encounters that occur in full cellular volumes. These limitations define the frontier for the next phase of development.

5.2 Future Work

Table 5.1. Proposed Extensions for Enhanced Mitochondrial Network Modeling

Future Extension	Description
3D Initialization	Import 3D z-stack segmentations to reconstruct layered topology and resolve projection artifacts from 2D imaging.
Metabolic Coupling Layer	Introduce a per-segment metabolic variable (e.g., $\Delta\Psi$) to modulate fusion/fission behavior based on local energetic state.
Mitophagy & Biogenesis	Simulate network turnover by enabling removal (mitophagy) or creation (biogenesis) of mitochondrial segments based on quality control rules.

doi:10.6342/NTU202503815

Automated Parameter Tuning	Apply optimization methods (e.g., ABC-SMC, Bayesian inference) to fit simulation parameters against experimental statistics efficiently and reproducibly.
High-Content Integration	Scale framework to screen large datasets by automating initialization, simulation, and scoring of treatment effects across hundreds of cells or conditions.

Future extensions will focus on increasing biological realism, improving parameter identifiability, and scaling the framework for large-scale screening and benchmarking applications. Five directions are outlined below.

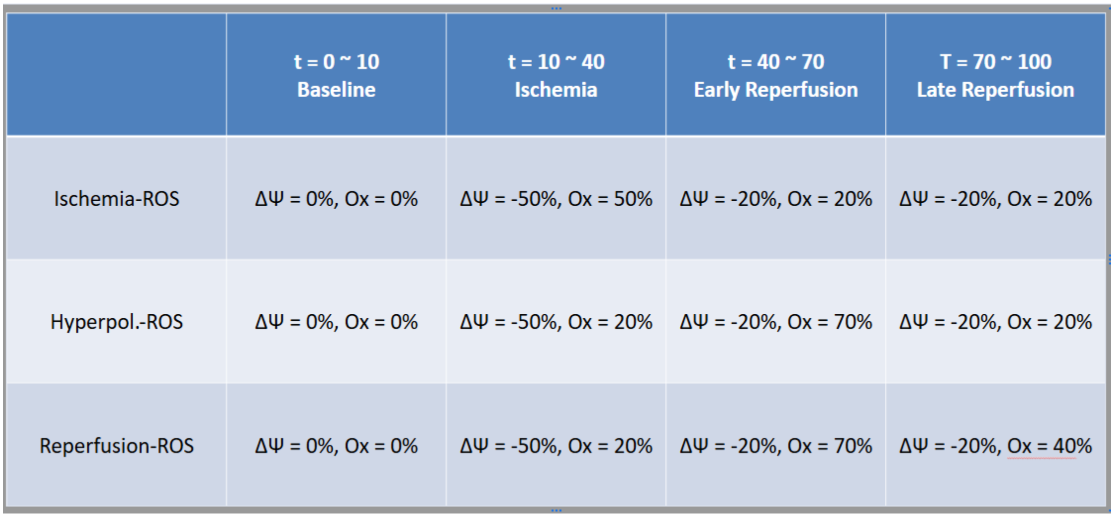
1. 3-D Initialization from Z-Stack Imaging

Extending initialization from 2D projections to volumetric z-stack reconstructions will allow the model to capture vertical crossings, layered branching, and depth-dependent encounter frequencies that are lost in projected geometries. A 3D import pipeline would ingest voxel-level segmentations, skeletonize in three dimensions, and generate topology species with full xyz coordinates. Comparisons between 2D-projected and 3D-resolved simulations would help disentangle apparent fragmentation caused by projection artifacts from true structural disassembly.

2. Metabolic Coupling Layer

Mitochondrial fusion competence is strongly influenced by energetic status, membrane potential (ΔΨ), and redox balance. Incorporating a light-weight metabolic state variable—tracked per topology, per segment, or per spatial sub volume—would enable state-conditioned reaction propensities: high ΔΨ could raise fusion probability; energetic collapse (e.g., FCCP-like uncoupling) could suppress fusion or bias fission. Coupling structural dynamics to metabolic feedback would provide mechanistic traction on stress responses and could reconcile simulation–experiment discrepancies observed under pharmacological treatments [41].

Table 5.2. Time-Resolved Mitochondrial Metabolic States Across Ischemia–Reperfusion Phases.



	t = 0 ~ 10 Baseline	t = 10 ~ 40 Ischemia	t = 40 ~ 70 Early Reperfusion	T = 70 ~ 100 Late Reperfusion
Ischemia-ROS	ΔΨ = 0%, Ox = 0%	ΔΨ = -50%, Ox = 50%	ΔΨ = -20%, Ox = 20%	ΔΨ = -20%, Ox = 20%
Hyperpol.-ROS	ΔΨ = 0%, Ox = 0%	ΔΨ = -50%, Ox = 20%	ΔΨ = -20%, Ox = 70%	ΔΨ = -20%, Ox = 20%
Reperfusion-ROS	ΔΨ = 0%, Ox = 0%	ΔΨ = -50%, Ox = 20%	ΔΨ = -20%, Ox = 70%	ΔΨ = -20%, Ox = 40%

3. Mitophagy and Biogenesis Modules

Network turnover is governed not only by fusion and fission but also by selective removal of damaged fragments (mitophagy) and the birth of new mitochondrial elements through biogenesis. Adding rules that mark, segregate, and remove slow-quality segments—optionally triggered by loss of ΔΨ or accumulation of fragmentation events—would permit study of quality control pathways. Complementary biogenesis rules could nucleate new tubes at specified rates or locations, supporting steady-state turnover studies and modeling of recovery after acute stress.

4. Automated Parameter Tuning

Manual parameter fitting is time-consuming and subjective. Bayesian optimization, likelihood-free inference (e.g., ABC-SMC), or differentiable surrogate models could be used to optimize fusion/fission rates, interaction radii, and mechanical constants directly against experimental time-series statistics (degree distributions, fragment counts, event rates). Automated calibration would accelerate cross-condition studies, reduce user bias, and provide credible intervals on fitted parameters, improving interpretability of condition-to-condition differences.

5. Integration with High-Content Screening Pipelines

The modular export/analysis stack makes the framework well suited for scale-up. Batch initialization from hundreds of microscopy fields could be paired with automated parameter fitting to rank perturbations (drug compounds, genetic hits) by their ability to rescue or degrade network integrity relative to control baselines. Semi-synthetic ground-truth datasets—generated by varying known reaction parameters and adding controlled imaging noise—could serve as benchmarking panels for evaluating image-analysis tools used in screening workflows. Such integration would move the framework from single-cell mechanistic studies toward population-scale discovery applications.

5.3 Final Remarks

In sum, this thesis advances an extensible, data-connected modeling platform that links mitochondrial structure, stochastic reaction dynamics, and image-driven validation within a single reproducible workflow. By exposing the mapping between biological assumptions and emergent network behavior, the framework supports mechanistic hypothesis testing today and sets the stage for coupled metabolic-structural simulations tomorrow. Continued development along the future directions outlined above should enable more faithful modeling of mitochondrial life cycles, improved interpretation of pharmacological screens, and deeper insight into how organelle connectivity supports cellular health.

doi:10.6342/NTU202503815

REFERENCES

[1] Ahola S (2024), Editorial: Mitochondrial bioenergetics and metabolism: implication for human health and disease, Front. Mol. Biosci. 11:1468758, doi: 10.3389/fmolb.2024.1468758.

[2] Douglas R. Green, Guido Kroemer, The Pathophysiology of Mitochondrial Cell Death.Science305,626-629(2004). DOI:10.1126/science.1099320.

[3] Laura A. Sena, Navdeep S. Chandel, Physiological Roles of Mitochondrial Reactive Oxygen Species, Molecular Cell, Volume 48, Issue 2, 2012, Pages 158-167, ISSN 1097-2765.

[4] Nunnari, J., & Suomalainen, A. (2012). Mitochondria: In sickness and in health. Cell, 148(6), 1145–1159.

[5] Spinelli, J.B., Haigis, M.C. The multifaceted contributions of mitochondria to cellular metabolism. Nat Cell Biol 20, 745–754 (2018).

[6] Lin, M., Beal, M. Mitochondrial dysfunction and oxidative stress in neurodegenerative diseases. Nature 443, 787–795 (2006).

[7] Bradford B. Lowell, Gerald I. Shulman, Mitochondrial Dysfunction and Type 2 Diabetes.Science307,384-387(2005).DOI:10.1126/science.1104343.

[8] Vyas, S., Zaganjor, E., & Haigis, M. C. (2016). Mitochondria and cancer. Cell, 166(3), 555–566

[9] Douglas C. Wallace, Mitochondrial Diseases in Man and Mouse.Science283,1482-1488(1999).DOI:10.1126/science.283.5407.1482.

[10] Richard J. Youle, Alexander M. van der Bliek, Mitochondrial Fission, Fusion, and Stress.Science337,1062-1065(2012). DOI:10.1126/science.1219855.

[11] Yu R, Lendahl U, Nistér M and Zhao J (2020) Regulation of Mammalian Mitochondrial Dynamics: Opportunities and Challenges. Front. Endocrinol. 11:374. doi: 10.3389/fendo.2020.00374.

[12] Timothy Wai. Is mitochondrial morphology important for cellular physiology?. Trends in Endocrinology and Metabolism = Trends in Endocrinology & Metabolism , 2024, 35 (10), pp.854-871. ff10.1016/j.tem.2024.05.005ff.

[13] Shah SI, Paine JG, Perez C, Ullah G (2019) Mitochondrial fragmentation and network architecture in degenerative diseases. PLOS ONE 14(9): e0223014.

[14] Westrate LM, Drocco JA, Martin KR, Hlavacek WS, MacKeigan JP (2014) Mitochondrial Morphological Features Are Associated with Fission and Fusion Events. PLOS ONE 9(4): e95265.

[15] Knowlton, A.A. and Liu, T.T. (2016), Mitochondrial Dynamics and Heart Failure. Comprehensive Physiology, 6: 507-526.

[16] Lagos, D., de Santiago, P. R., Pérez, N., Cartes-Saavedra, B., Vial-Brizzi, J., Podmanicky, O., Horvath, R., & Eisner, V. (2025, April 4). Disease-causing MFN2 mutants impair mitochondrial fission dynamics by distinct DRP1 dysregulation [Preprint]. bioRxiv.

[17] Fischer, Tara, "Mitochondrial Fission After Traumatic Brain Injury" (2017). The University of Texas MD Anderson Cancer Center UTHealth Graduate School of Biomedical Sciences Dissertations and Theses (Open Access). 807.

[18] Song, M., Mihara, K., Chen, Y., Scorrano, L., & Dorn, G. W., II. (2015). Mitochondrial fission and fusion factors reciprocally orchestrate mitophagic culling in mouse hearts and cultured fibroblasts. Cell Metabolism, 21(2), 273–286.

[19] Sukhorukov VM, Dikov D, Reichert AS, Meyer-Hermann M (2012) Emergence of the Mitochondrial Reticulum from Fission and Fusion Dynamics. PLOS Computational Biology 8(10): e1002745.

[20] Tam, Z. Y., Gruber, J., Halliwell, B., & Gunawan, R. (2013). Mathematical modeling of the role of mitochondrial fusion and fission in mitochondrial DNA maintenance. PLoS ONE, 8(10), e76230.

[21] Vessoni, A. T., Quinet, A., de Andrade-Lima, L. C., Martins, D. J., Garcia, C. C. M., Rocha, C. R. R., Vieira, D. B., & Menck, C. F. M. (2016). Chloroquine-induced

glioma cells death is associated with mitochondrial membrane potential loss, but not oxidative stress. Free Radical Biology and Medicine, 90, 91–100.

[22] Clemens, L., Kutuzov, M., Bayer, K. V., Goyette, J., Allard, J., & Dushek, O. (2021). Determination of the molecular reach of the protein tyrosine phosphatase SHP-1. Biophysical Journal, 120(10), 2054–2066.

[23] Brodland GW. How computational models can help unlock biological systems. Semin Cell Dev Biol. 2015 Dec;47-48:62-73. doi: 10.1016/j.semcdb.2015.07.001. Epub 2015 Jul 9. PMID: 26165820.

[24] Kitano H. Computational systems biology. Nature. 2002 Nov 14;420(6912):206-10. doi: 10.1038/nature01254. PMID: 12432404.

[25] Naomi Oreskes et al. ,Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences.Science263,641-646(1994).DOI:10.1126/science.263.5147.641.

[26] Walpole J, Papin JA, Peirce SM. Multiscale computational models of complex biological systems. Annu Rev Biomed Eng. 2013;15:137-54. doi: 10.1146/annurev-bioeng-071811-150104. Epub 2013 Apr 29. PMID: 23642247; PMCID: PMC3970111.

[27] Yu JS, Bagheri N (2024) Model design choices impact biological insight: Unpacking the broad landscape of spatial-temporal model development decisions. PLOS Computational Biology 20(3): e1011917.

[28] Viana, M. P., Brown, A. I., Mueller, I. A., Goul, C., Koslover, E. F., & Rafelski, S. M. (2020). Mitochondrial fission and fusion dynamics generate efficient, robust, and evenly distributed network topologies in budding yeast cells. Cell Systems, 10(3), 287–297.e5.

[29] Zamponi, N., Zamponi, E., Billoni, O. V., Cannas, S. A., Helguera, P. R., & Chialvo, D. R. (2018). Mitochondrial network complexity emerges from fission/fusion dynamics. Scientific Reports, 8, 363.

[30] Holt, K. B., Winter, J., Manley, S., & Koslover, E. F. (2024). Spatiotemporal modeling of mitochondrial network architecture. PRX Life, 2(4), 043002.

doi:10.6342/NTU202503815

[31] Liesa, M., & Shirihai, O. S. (2013). Mitochondrial dynamics in the regulation of nutrient utilization and energy expenditure. Cell Metabolism, 17(4), 491–506.

[32] Wang Z, Natekar P, Tea C, Tamir S, Hakozaki H, et al. (2023) MitoTNT: Mitochondrial Temporal Network Tracking for 4D live-cell fluorescence microscopy data. PLOS Computational Biology 19(4): e1011060.

[33] Hoffmann M, Fröhner C, Noé F (2019) ReaDDy 2: Fast and flexible software framework for interacting-particle reaction dynamics. PLOS Computational Biology 15(2): e1006830.

[34] Bi-Chang Chen et al. ,Lattice light-sheet microscopy: Imaging molecules to embryos at high spatiotemporal resolution.Science346,1257998(2014).DOI:10.1126/science.1257998.

[35] Richard J. Youle, Alexander M. van der Bliek ,Mitochondrial Fission, Fusion, and Stress.Science337,1062-1065(2012).DOI:10.1126/science.1219855.

[36] Chuphal, P., Lanctôt, J. D., Cornelius, S. P., & Brown, A. I. (2024, October 22). Mitochondrial network branching enables rapid protein spread with slower mitochondrial dynamics. PRX Life, 2(4), 043005.

[37] Krukowski, K., Nolan, A., Frias, E. S., Boone, M., Ureta, G., Grue, K., Paladini, M.-S., Elizarraras, E., Delgado, L., Bernales, S., Walter, P., & Rosi, S. (2020). Small molecule cognitive enhancer reverses age-related memory decline in mice. eLife, 9, e62048.

[38] Erban, R., Chapman, J., & Maini, P. (2007, April 15). A practical guide to stochastic simulations of reaction-diffusion processes (arXiv:0704.1908v2 [q-bio.SC]). arXiv.

[39] Schneider, C., Rasband, W. & Eliceiri, K. NIH Image to ImageJ: 25 years of image analysis. Nat Methods 9, 671–675 (2012).

[40] Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD – Visual Molecular Dynamics. Journal of Molecular Graphics, 14(1), 33–38.

doi:10.6342/NTU202503815

[41] Fogo, G. M., Torres Torres, F. J., Speas, R. L., Anzell, A. R., & Sanderson, T. H. (2025). Agent-based modeling of neuronal mitochondrial dynamics using intrinsic variables of individual mitochondria. iScience, 28(5), Article 112390.

doi:10.6342/NTU202503815