Exploring the Emergence of Complexity in Vesicle Traffic Networks through Molecular and Topological Constraints

A thesis

Submitted towards the partial fullfilment of

BS-MS dual degree programme

by

Kritika Kumari

Date: March 15th, 2024

under the guidance of

Prof. Mukund Thattai

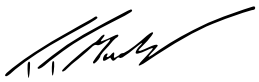
National Centre for Biological Sciences and Research-TIFR

from May 2022 to Mar 2023

Indian Institute of Science Education and Research Pune

Certificate

This is to certify that this dissertation entitled **Exploring the Emergence of Complexity in Vesicle Traffic Networks through Molecular and Topological Constraints** submitted towards the partial fulfillment of the BS-MS degree at the Indian Institute of Science Education and Research, Pune represents original research carried out by Kritika Kumari at National Centre for Biological Sciences and Research-TIFR, under the supervision of Prof. Mukund Thattai during academic year May 2023 to March 2024.



Mukund Thattai

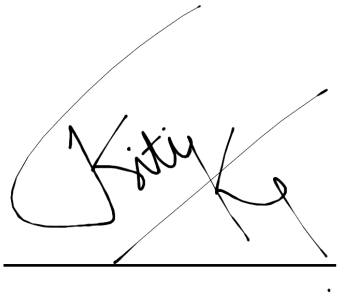
Committee:

Prof. Mukund Thattai

Prof. Nagaraj Balasubramanian

Declaration

I hereby declare that the matter embodied in the report entitled **Exploring the Emergence of Complexity in Vesicle Traffic Networks through Molecular and Topological Constraints** are the results of the work carried out by me at the Department of Biology, Indian Institute of Science Education and Research (IISER) Pune, under the supervision of Prof. Mukund Thattai (National Centre for Biological Sciences and Research-TIFR) and the same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgment of collaborative research and discussions.



Kritika Kumari
Roll no.: 20191077

Date: 16/05/2024

Acknowledgements

I would like to acknowledge the Indian Institute of Science Education and Research, Pune, for introducing and engraining in me the spirit of Scientific inquiry and for the most beautiful years of my life until now.

I would like to acknowledge Prof. Mukund Thattai and the research scholar- Sahana K.S., who is a collaborator on this project, for their support and guidance during this project.
I would like to acknowledge the Simons Foundation for the funding I received during this project.
I would like to acknowledge the National Centre of Biological Sciences for the excellent facilities provided.
I want to thank Madhu (Prof. M.S. Madhusudhan) for raising me in the field of computational biology, and always believing in me.
I extend my special thanks to Mukundan and other COSPI lab members at IISER Pune.
In the end, I want to thank my friends- Jezer, Sudeepta, Sachit, Krishnan, Aashish, and all the friends who made exposure to pure research easier and more enriching and also helped me get through the few tough times I had.

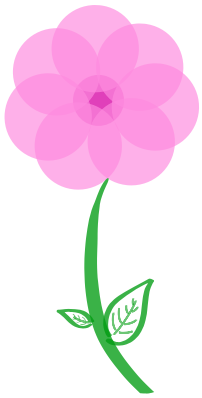
Special thanks to Sahana for her amazing jokes and for being the partner in crime through this project.

Contributions

Contributor name	Contributor role
Sahana K.S. and Mukund Thattai	Creation of the Model(s)
Kritika Kumari	Graph Search Parallelization
Kritika Kumari	Graph Properties and Proofs
Kritika Kumari	Symmetry in Graphs
Kritika Kumari	Molecules to Graph
Kritika Kumari	Lit. Review: FECA to LECA
Sahana K.S., Kritika Kumari, Mukund Thattai	The Evolutionary Landscape
Sahana K.S., Kritika Kumari	References
Kritika Kumari	Figures and Tables
Kritika Kumari	Thesis Writing

[1]

[1] https://journals.biologists.com/jcs/pages/author-contributions



To my mother, and my father, and my brother. And to all the beautiful people who crossed my path, guided me, and took care of me. And to every moment of my life, that the universe conspired, that led me to IISER and to write this thesis.

[2]

[2]The flower above was created by the author in Inkscape (Inkscape Project. (2020)).

Abstract

The Endomembrane Trafficking machinery is a defining complex feature of Eukaryotes. This Intracellular Trafficking is regulated by Specificity Modules that are the budding and Fusion Machinery. Along the path of Eukarya evolution, various questions have been raised regarding the emergence of complexity, creation of new organelles, and retargeting of vesicle pathways in eukaryotes.

While these complex features have been ascribed to Eukaryotic success in the Phanerozoic era and have been studied with various approaches, this thesis is an attempt at using the simple yet mathematically rich framework of Graph theoretic Boolean Modelling of the vesicle trafficking. Using Boolean modeling of cargo flows we can get to a molecular representation of the network and its specificity modules.

The creation of the molecular regulatory landscape of graphs has allowed us to study how topological changes in vesicle pathways are constrained at the molecular level and functionally. We discover how the emergence of molecular properties takes place across various kinds of cells from the underlying constraints of Biophysical properties upon which a Graph is considered a valid cell. We also try resurrecting intermediates in the path of emergence of topological properties starting from the basic biochemical interactions in a genome, and how it implicates the creation of a non-endosymbiotic compartment or pathways to an engulfed compartment through the raw material of existing molecular machinery following specific genetic events.

Contents

Chapter 1

Introduction

1.1 The History of Studying Eukaryotes

From the discovery of first cells by Robert Hooke (Hooke, 2010) to the formalization of the first definition of Eukaryotes and Prokaryotes by Roger Stanier and C. B. van Niel in 1962 (Stanier and Van Niel, 1962), that initiated the cascade of ever-deepening dichotomy in between, finally leading to being classified into separate domains of life by Robert Whittaker (Whittaker, 1969; Sapp, 2005). Several attempts have been made to discover the phylogenetic divergence of Eukaryotes and Prokaryotes, with questions spanning from the timing, the nature of changes, biochemical and biophysical, along with changes in the genomic landscape through various processes and through external environmental changes/stresses including their interaction with other organisms. In short, how far are Eukaryotes in complexity to Prokaryotes, how did this Complexity come about, and what was the evolutionary time scale they diverged from each other?

This thesis is an attempt to study this rift by examining the emergence of the complexity by studying one of the most important features of Eukaryotes that mark their identity- the Endomembrane Vesicle Trafficking System (Cavalier-Smith, 2002; Koonin, 2010; Koumandou *et al.*, 2013; Marijuán *et al.*, 2013; Thattai, 2023). The reason behind choosing this feature is simple, we do not find any extant Eukaryote lacking this sophisticated sub-cellular communication mechanism through the transport of vesicles between compartmentalized entities in the cell (Koumandou *et al.*, 2007; Field and Dacks, 2009a; Koonin, 2010; Fritz-Laylin *et al.*, 2010; Koumandou *et al.*, 2013; Thattai, 2019; Zaremba-Niedzwiedzka *et al.*, 2017). The very conservation of this system across the current-day Eukaryotes, also known as "Crown-group" Eukaryotes, begs for its existence in the Last Common Ancestor of Eukaryotes (LECA) (Koumandou *et al.*, 2007; Field and Dacks, 2009a; Koonin, 2010; Fritz-Laylin *et al.*, 2010; Koumandou *et al.*, 2013; Zaremba-Niedzwiedzka *et al.*, 2017; Dacks and Field, 2018; Barlow *et al.*, 2018; Thattai, 2019; More *et al.*, 2020; Vosseberg *et al.*, 2021), and its diversity in extant organisms denotes

the phylogenetic distance of LECA from crown group eukaryotes (Embley and Martin, 2006; Schlacht *et al.*, 2014; Vosseberg *et al.*, 2021). While it also hints towards the evolutionary time period at which it could have happened, it is not the focus of this thesis. The purpose of my thesis is to illuminate, given that the LECA already had such complex features, the path from the most primitive First Eukaryotic Common Ancestor (FECA) to LECA.

1.2 Evolution of the Last Common Eukarytic Ancestor: A Literature Review

Several attempts at reconstructing the history of Eukarya involve getting the complete picture through phylogenomics, biophysical and ecological studies, and fossil evidence findings of organisms with eukaryotic structures that might belong to the "stem group," the taxa of organisms that came before LECA. This problem was made easier with the discovery that the TACK and Asgardarchaeota superphyllum, which have been shown to be not only genetically closer but also possess important characteristics of eukaryotes, such as the "eukaryotic signature proteins" (Field and Dacks, 2009b; Koumandou *et al.*, 2013; Zaremba-Niedzwiedzka *et al.*, 2017). There is conclusive evidence to show that the first eukaryotic ancestor belongs to the Asgard Archaeal Superphylum, and more specifically to the phylla - Lokiarchaeota (Lambert, 2019; Imachi *et al.*, 2020). Genomic analyses have shown that certain membrane trafficking machinery components or "Eukaryotic Signature Proteins" such as proto-coatomers, ESCRT complex proteins, plethora of small GTPases, SRP, etc., exist in members of this superphylum (Cao and Saier, 2003; Field and Dacks, 2009b; Koumandou *et al.*, 2013; Koonin, 2015; Zaremba-Niedzwiedzka *et al.*, 2017; Eme *et al.*, 2023).

However, the intermediate paths from FECA to LECA remain unclear. There are disputes in timings and time scales over which stem lineages and LECA arrived, with pieces of evidence towards the Archaean and Mesoproterozoic (Brocks *et al.*, 2023), respectively. For example, there have been some attempts to discover the existence of tetracyclic sterol pathways, characteristic of eukaryotes, in the preservation of intermediate protosteroids in fossils before the Tonian period, in which modern Eukaryotes appeared, to map the path that preceded the LECA. However, recent studies also reveal earlier Proterozoic era had significant primitive eukaryote species richness suggesting a much deeper history before the extinction of stem lineages (Riedman *et al.*, 2023). Again, this is not the focus of my question, which involves studying the path taken from FECA to LECA. We might make statements about the relative timings of the emergence of certain complex features in the vesicle trafficking system.

Regardless of a narrowed-down starting point to an Archeal ancestor, there are several possible routes that evolution could taken while going through intermediates that have now gone extinct. Going along the earlier line of reasoning of what prevails must have existed in the common ancestor with the example of mitochondria; we see that all primary eukaryotes have mitochondria. It is a widely held view that the engulfment of mitochondria (De Duve, 2005) by the archeal ancestor (Imachi *et al.*, 2020) was a major part of this transition from FECA to LECA as it helped in the creation of new compartments (Martin and Koonin, 2006; Cavalier-Smith, 2010; Martin *et al.*, 2015; Baum and Spang, 2023). However, there are very few reasons to believe that it was coincident with the FECA (Koonin, 2010; Martijn and Ettema, 2013). Instead, various works have cited the need for advancement in cytological control, like loss of cell wall and primordial cytoskeletal elements deforming membrane into protrusions before the cell could host an external bacteria and have a nucleus as suggested in the 'Eocyte-hypothesis' (Embley and Martin, 2006; Martijn and Ettema, 2013; Baum and Baum, 2014, 2020). It is important to note that the evolution of stem group lineages was not finished at the internalization of mitochondria or nucleus as a set of molecular motifs needed to be created for the Endomembrane organization and trafficking, which are found in all extant eukaryotes today (Cavalier-Smith, 2002; De Duve, 2005; Field and Dacks, 2009b; Koumandou *et al.*, 2013; Koonin, 2015; More *et al.*, 2020; Vosseberg *et al.*, 2021). Hypothesizing a rapid transformation of an archeal ancestor through the engulfment of mitochondria because we can't find any intermediates of this process today is, therefore, a short-sighted way of thinking. An extinction event is, of course, implied by the very definition of the LECA as a complex organism and no mitochondriate or amitochondriate archaea with any subcellular organization seen today.

Thus, from the above discussion, it is clear that we need to look at every aspect of complexity, which must have a causal relationship with the pre-existing state and a result of the selection operating on the organism, to discover the true path from FECA to LECA.

Our approach to answering this question takes the perspective of Assembly theory (Sharma *et al.*, 2023), where we define the smallest unit of selection, here the cell, using molecular and functional elements that can be built or assembled in evolutionary time based on our definition of transitions between cells. This is studied in the framework of Graph Theory (Biggs *et al.*, 1986) and Boolean Modelling (Albert and Robeva, 2015), discussed in detail in the next chapter. Laying down these cells sequentially through a mode of selection unravels the path of evolution of eukaryotes and gives a glimpse of the cells that could have been contingent in the assembly of the complex LECA. We also explore the multiple equivalent and non-equivalent intermediates of this process which could have formed the space for assembly to take place or for selection to operate, and

how they can weigh in on evolutionary trajectories in the case when natural selection could have been non-unidirectional.

To realize how far FECA had to traverse the different paths of evolution before becoming the LECA is the goal of the upcoming paper by Sahana K.S., Kritika Kumari, and Mukund Thattai 2024 (unpublished). The pathways or constraints discovered in this process of eukaryogensis might also be an underlying principle of how eukaryotes diversified, as has been suggested in "slow-drip" models of early eukaryotic evolution (e.g. (Moreira and López-García, 1998; Lester *et al.*, 2006; Martijn and Ettema, 2013; Butterfield, 2015)).

1.2.1 The Evolution of Vesicle Trafficking and The Current Picture of Vesicle Trafficking

As discussed in the above sections, the endomembrane system of the cell became a very key identifying feature of eukaryotes and therefore is the communication between these compartmentalized systems, whether at short or long distances. While we are not concerned about the distances of transport of cargo between compartments in this thesis, we take a sharp look at vesicle pathways that allow this communication between compartmentalized regions of the cell. These pathways are indispensable for a eukaryotic cell and are not missing in any of the extant eukaryotes (Cavalier-Smith, 2002; Cao and Saier, 2003; De Duve, 2005; Field and Dacks, 2009b; Koumandou *et al.*, 2013; Koonin, 2015; Zaremba-Niedzwiedzka *et al.*, 2017; More *et al.*, 2020; Eme *et al.*, 2023), and therefore would have been part of the process of evolution from FECA (Schlacht *et al.*, 2014) as no archaea, existing today, has any endomembrane definition let alone intricate trafficking (McInerney *et al.*, 2011; Lambert, 2019; Imachi *et al.*, 2020).

Based on the existing diversity of molecular repertoire of molecules regulating this sophisticated mechanism of trafficking, which became apparent through the Nobel Prize winning work of Rothman, Schekman, and Sudhof (Novick and Schekman, 1979; Balch *et al.*, 1984; Perin *et al.*, 1990; Kaiser and Schekman, 1990; Hata *et al.*, 1993; Söllner *et al.*, 1993) includes the following key features: -

- Specialized directed flow of proteins and lipids exist between identifiable compartments which maintain distinct compositions homeostatically within certain timescales. (Thattai, 2023; Bonifacino and Glick, 2004; Zaremba-Niedzwiedzka *et al.*, 2017). The specificity of these flows is conferred by budding and fusion machinery.
- Specificity to the operations of budding and fusion to the target is encoded within

the vesicular transport system, requires concerted action by SNARE proteins that are recycled back to the source, Rab GTPases, and tethering complexes, as well as the participation of the cytoskeletal elements and many other factors.

- These set of Specificity modules covering each operation taking place in a vesicle trafficking network are unique for each of the specific pathways from a particular source organelle to a target organelle (Gerst, 2003; Bonifacino and Glick, 2004; Cai *et al.*, 2007; Hutagalung and Novick, 2011; Faini *et al.*, 2013; Hariri *et al.*, 2014; Dubuke and Munson, 2016; Pfeffer, 2017; Zaremba-Niedzwiedzka *et al.*, 2017; Arakel and Schwappach, 2018; Koike and Jahn, 2019; Sauvola and Littleton, 2021; Cui *et al.*, 2022; Thattai, 2023).

A closer look at the budding and fusion machinery of this process-

Where budding machinery is constituted by Coats, their recruiters, adaptors, with loading of specific cargos via interaction with adaptins or cargo receptors (Bonifacino and Glick, 2004; Faini *et al.*, 2013; Hariri *et al.*, 2014; Zaremba-Niedzwiedzka *et al.*, 2017; Arakel and Schwappach, 2018; Cui *et al.*, 2022).

Most of the intracellular vesicles are made by these coat complexes- COPI, COPII and Clathrin (Bonifacino and Glick, 2004; Faini *et al.*, 2013; Cui *et al.*, 2022). The mechanism of their recruitment from the cytosol and the coat-specific cargo loading at the membrane involves recruiter molecules (small GTPases of the Arf1/Sar1 family) in GTP bound state initiating coat-assembly at the membrane (Bonifacino and Glick, 2004; Faini *et al.*, 2013; Cui *et al.*, 2022). Adaptin subunits with cargo binding motif sites (Arora and Damme, 2021) are primarily involved in recruiting specific cargo and regulating further coat assembly (Faini *et al.*, 2013; Cui *et al.*, 2022). Small GTPase molecules induce scission of the vesicle from the membrane by inserting an amphipathic terminal helix into the membrane to induce membrane constriction after cargo are sorted into the vesicle (Zhukovsky *et al.*, 2019; Cheng *et al.*, 2021).

The fusion machinery is constituted by small GTPases like Rabs, which are important markers of compartment identity and (Pfeffer, 2017; Hutagalung and Novick, 2011; Cai *et al.*, 2007) mediate the uncoating of the vesicle. Their activity state is regulated by specific guanine-nucleotide-exchange factors (GEFs) and GTPase-activating proteins (GAPs). Distinct Rab GTPases are recruited to different membrane compartments defining their identity (Kjos *et al.*, 2018; Lamber *et al.*, 2019; Homma *et al.*, 2021). Rabs also further recruit Rab effector molecules, indicating several regulatory mechanism during the fusion state happening at the level of coat. The interaction of Rabs and its effector with tethers and SNARE Regulator mediate assembly of SNARE complexes on

the vesicle(vSNARE) and the target(t-SNARE) compartment (Gerst, 2003; Bonifacino and Glick, 2004; Cai *et al.*, 2007; Hutagalung and Novick, 2011; Dubuke and Munson, 2016; Pfeffer, 2017; Koike and Jahn, 2019; Sauvola and Littleton, 2021; Cui *et al.*, 2022). SNARE regulators promote the fusion process by influencing how the SNARE complex assembles (Ungermann and Kümmel, 2019; Han *et al.*, 2017; Travis *et al.*, 2020).

There are certain underlying trends in the evolution of the above molecules that regulate this process in the crown-group archaea, which have been widespread throughout Eukaryotic evolution and could have existed pre-LECA as well: -

- Paralogous protein families exist across eukaryotes, but these protein families are organelle-specific; they guide the shipment of cargo to a specific compartment.(Bonifacino and Glick, 2004; Gabaldón and Koonin, 2013; Dacks and Field, 2018; More *et al.*, 2020; Thattai, 2023)

- Different organelle targeting machinery cluster analyses suggest the creation of paralogous protein families within species (Gabaldón *et al.*, 2006; Dacks and Field, 2018; More *et al.*, 2020; Thattai, 2023) and orthologous protein families across species (Gabaldón and Koonin, 2013).

- Duplication events at evolutionary time-scale are at the heart of paralogous protein families creation. Suggesting the creation of new vesicle pathways using the raw material from existing molecular machinery (Schlacht *et al.*, 2014; Dacks and Field, 2018; More *et al.*, 2020; Vosseberg *et al.*, 2021; Purkanti and Thattai, 2022; Thattai, 2023).

- Minor mistargeting or the "imperfection" of Eukaryotes- Dual targeting (Martin, 2010), and even multiple organellar proteome analyses reveal the co-localization of protein families on multiple different organelles (Gabaldón and Pittis, 2015).

- Retargetting of protein cargo through modification in the vesicle pathways from one compartment to another(s) has been pervasive and is often preceded by a paralogous gene creation by duplication and co-option (Gabaldón *et al.*, 2006).

- Natural selection-driven retargeting has been observed too. For example: change in Alanine Glyoxylate Aminotransferase trafficking due to change in the site of substrate production - glyoxylate - from peroxisomes to mitochondria while going

from complete Herbivory to Carnivory, with a co-localization observed in Omnivores as well (Danpure, 1997; Gabaldón *et al.*, 2006).

- The targeting signals are generally encoded by short stretches of amino acid sequences. Hence, any given protein is not many mutations away from acquiring or losing a given targeting signal (Danpure, 1997; Gabaldón *et al.*, 2006; Gabaldón and Pittis, 2015).

There are several hypotheses that have been proposed in terms of how duplication is a key event in eukaryotic rise in complexity, as well as the metabolic constraints on creation of connection pathways to a new compartment .

Given Protein Retargeting in Membrane Traffic Evolution has been thought to be rampant across eukaryotes even after the Last Eukaryotic Common Ancestor. It has been hypothesized that multi-enzyme reactions pose a challenge for retargeting of cargo to occur in evolutionary steps, single retargeting could be lost to mutation (Martin, 2010).

Regarding the creation of a new pathway to endosymbiosed organelle or internally derived one: Minor mistargeting or Dual targeting has allowed the retargeting (or specialization of the pathway to one compartment) to be selected across eukaryotes (Martin, 2010) and form novel pathways. Whether or not this hypothesis is true remains an open question and will be addressed by this thesis towards the end.

It is clear that the evolutionary path would have incurred the emergence of complexity slowly, assembling each block step by step. We will look at how discrete or continuous these steps could be in detail in this thesis.

Chapter 2

Model

Sahana K.S. and Mukund Thattai (2022, Unpublished) created the Graph theoretic model of the Vesicle Trafficking Network(VTN) based on minimalistic boolean modeling (Albert and Robeva, 2015). Boolean models allow an efficient description of the system by using only binary numerics to represent the presence (1) and absence(0) of any encoded quantity. This model builds upon the existing work on Vesicle Trafficking Approaches starting at purely biophysical modeling invoking biochemical rules at cargo to coat interaction level (Mani and Thattai, 2016), and Shukla *et al.* (2017)'s and (Bhattacharyya *et al.*, 2021)'s satisfiability approach to check for VTNs which are consistent with different models of Snare regulation, with (Mani *et al.*, 2022)'s work using a pure graph connectivity framework that is property of specific regulation model solutions to describe constraints and check the consistency of any VTN.

The model created by Sahana and Mukund takes a deeper dive at defining molecular rules over the budding and fusion functions of VTNs. It also introduces a novel approach to describing the genomic architecture of a VTN, which is consistent with all the defined molecular rules. This description of a regulatory molecular landscape allows us to describe genomic changes between graphs enabling us to describe the phylogeny between graphs when including function and to see how the molecular network maps onto the topological network of a VTN.

2.1 Graph Theoretic Modelling of Vesicle Trafficking

A network of vesicle trafficking in a homeostatic state has been established using Rothman-Schekman-Sudhof's Nobel prize winning work as described before, and the constraints leading to certain properties of the emerging network have been encoded by Graph-theoretic modeling (Mani and Thattai, 2016; Shukla *et al.*, 2017) and model checking through Boolean Satisfiability(SAT) (Shukla *et al.*, 2017). However, none of the previous models have been able to capture a molecular network emerging from the topology operations of budding and fusion. These Graph encoding of networks are directed graphs, and they represent compartments as nodes and vesicles transported in between source compartment and target compartment as directed edges. Some of the basic constraints upon which these Graphs are generated include the steady state of the compartments, which is made by a cyclical transport of cargo to maintain homeostasis of compartment compositions, and the distribution of cargo across vesicle and compartments building a molecular interaction network that has specificity modules directing the edges to their target. These constraints only allow Graphs with certain properties like Graph Connectivity, self-fusions for certain cargo to be flowing, fusion to the same compartment if vesicles carry the same set of molecules regardless of coats, etc., to be relevant in the biological framework created.

2.1.1 Modelling Vesicle Traffic Network as a Directed Transport Graph

This is a sub section 2.1.

We use a boolean framework similar to the previous models (Mani and Thattai, 2016; Shukla *et al.*, 2017; Bhattacharyya *et al.*, 2021), which involves nodes as compartments, labeled directed edges as specific vesicle pathways, and boolean representation of cargos. Sahana K.S., and Mukund Thattai (2022, Unpublished) created a Vesicle Trafficking network model incorporating a very important missing piece in all modeling approaches in sub-cellular traffic models, which is the creation of not only the molecular space which regulates the budding flows and fusions but also the realizing the molecular space which can destroy them if it comes to existence, of specific graphs in an emergent manner from a given topology.
Not only does the model involve the creation of graphs with bio-physical constraints, such as connectedness of compartments and uniqueness of compositions, but also biochemical constraints which must be satisfied and which are not merely characterized signified by cyclical flows of Snares as in the previous model. The realization of a molecular space that can be spanned maximally by any graph through a combinatorial approach allows

the creation of the molecular space that makes the graph, as well molecular space that breaks it. But the true importance of this approach is that it allows us to compare the genomic landscape of two different VTNs which occupy a sub-space of the whole space. There are several regulations or 'rules' of the specific operation of budding and fusion events for a given directed graph, also called a transport graph to be valid. This thesis is majorly focussed on a particular model where the fusion events are regulated by not just V- and T- Snares but also a V-Snare-Activator, as an analog to the biological Rabs on specific vesicles, which must exist on the vesicle to allow its fusion.

This Model of transport of cargo across compartments, which invokes biochemical specificity at steady state, is based on coat types and defining cargo types based on their binary interaction with each of the coats. For a given set of coats with cardinality n, we can have 2^n cargo types based on all combinatorial permutations that a binary interaction can make with the n coats.
It is important to note that only starting at 3 coat types- $a1, a2$, and $a3$ do we start seeing the non-trivial distribution of cargo and higher compartment connectivity of the network. Therefore this thesis will focus on the minimally complex 3-coat system that could have existed in the FECA. The emergence of the number of coats is looked upon in Sahana K.S.'s work and will be part of the upcoming paper (K.Shridhar, Kumari, Thattai, 2024 Unpublished).
Based on a cargo-type's interaction with the coat type (as shown in Figure 2.1), they will be recruited into the budding Vesicle, and based on the vesicle's pathway, they will be shipped to a compartment.

2.2 The Algorithmic Framework of Discovering Valid Vesicle Traffic Network Graphs

Figure 2.2 above demonstrates the heart of discovering a solution to the Model.
With these minimal interaction and flow rules for any directed graph taken, how do we arrive at a valid VTN and its molecular network?

- After labeling the edges with n (here 3) coats, one of the compartments is labeled as the ER where all the cargo types present in the system exist initially. The steady state of the compartment and vesicle compositions is then got from cargo types and their loading onto specific coat labeled edges based on their affinity (given coat interaction matrix in Figure 2.1) and flowing to the target compartments and recycled based on same affinities to coats available.

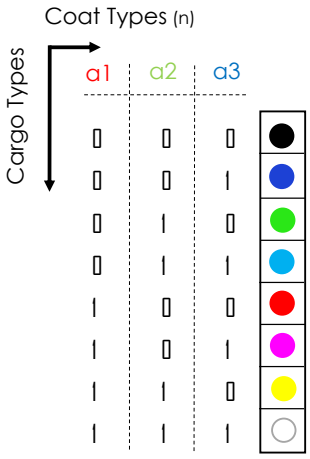


Figure 2.1: **Coat Interaction Matrix for 3-Coat System**: creation of cargo-types (color-coded on the right column) based on their interaction with the rest of the coats.

- Now, after this flow is set up, the cargo flows sent from the initial source compartment, ER, to the target, will only be allowed in the steady state of the network if there exists a vesicle type at the target upon which they can return to the source compartment, i.e., they are flowing on a strongly connected component of the graph.

- Therefore, only vesicles carrying cargo flowing cyclically survive (which is always the case due to ubiquitously flowing cargo type 8, unless there are no return edges). Therefore, there will be some cargo types that get stuck on end-point cargo and won't be recycled back. These cargoes are assumed to be flowing at a very low rate and are called Replenishment flows which replenish the target compartment, in a low flux, of the cargo type that gets stuck at the target compartment and acts as a marker in the case of it's stuck at only one compartment. The replenishment cargo flows are not taken amongst the cargo on a cyclical route flowing at a higher rate.

- Graphs that have more than one compartment with exact same compositions are discarded.

- Graphs that remain connected by strong fluxes of cargo types at the steady state are then sent for the next regulatory rules satisfiability check (discussed in the next section).

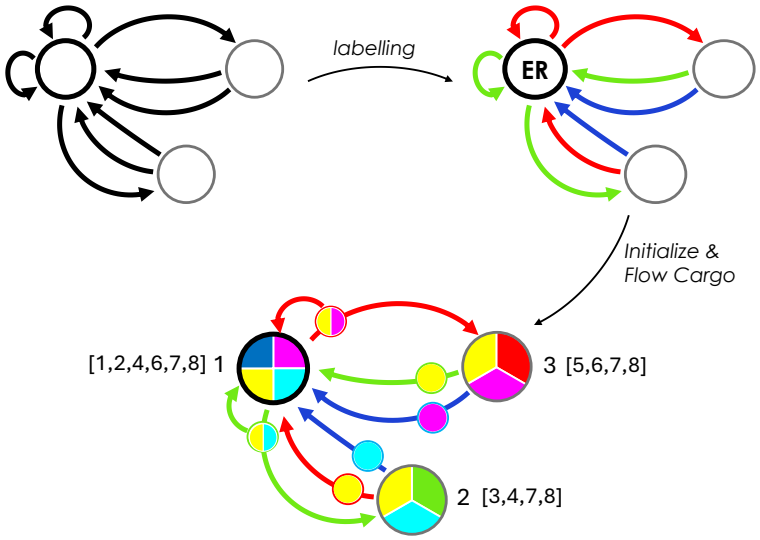


Figure 2.2: **Vesicle Trafficking Model Solution Search Algorithm** Discovery of a Model solution Graph G.

2.3 From Solution Graphs to Regulatory Molecules

The Algorithm to create the regulatory space of a graph is described below.
The Graph got at the steady state with unique compositions of compartments and compositions of vesicles are checked for whether the operations occurring are satisfied when using their compositions in the clause. There are two sets of operations, each of which must be satisfied under the specific clause(s) of the regulation model taken for a molecular machinery to exist as a solution to this graph.
If and only if a molecular machinery exists covering every operation will the graph be called a valid graph or a solution graph to our model.

We call the cargo types or their combinations, which may include other components of the graphs like coats and other cargos - 'Specificity modules'. So Budding and Fusion machinery of these steady-state networks are these combinatorial object of components that emerge out of the unique compartment compositions.

- The budding molecular machinery defined with respect to (hereby initialized as - w.r.t.) a coat-type is the property of the cargo types that are present uniquely on the compartment(s) that recruit the coat-type and are not found in any non-recruiting compartments. So they represent a union of intersection over recruiting

compartment compositions. If two coats are recruited by the same compartment(s), as well are not recruited by the same compartments, they can share the same set of recruiters or a single recruiter. Even if the coat-type is recruited on multiple compartments it can be covered minimally by a single cargo type which exists on all the recruiting compartments.

- Similarly, the fusion machinery is given by Vesicle-SNARE, and under the main Model of Snare activation, Activator-SNARE, and Target-SNARE(VAT) combinations, with the requirement of both V-SNARE and its activator to occur in the vesicles. A VAT combination is defined for a specific Vesicle-Type given by – ((Coat-type), source), e.g.-(a1,1), of the vesicle along with its target compartment. A valid VAT cargo-type combination is found by sampling the vesicle and target cargos, V and A from the vesicle (these could be identical) and T from the target, and then finding combinations created with non-target compartment cargo types keeping the vesicle the same. There must remain at least one VAT combination in every target compartment after taking out the intersection of these VATs with non-target VATs for any vesicle. There could be multiple VATs covering the fusion for a (vesicle-type target).
  If two different vesicle types have similar cargos on them as well as they have the same targets, then they can share the same set of VATs.

The possible space of molecules that can regulate this network is given by $8^3 = 512$ VAT molecules, given that a V-snare or A-snare, or a T-snare could maximally be of 8 types give the 8 cargo types in our 3 coat system, and $8 * 3 = 24$ recruiter molecules, making up the molecular space 536 bit-string long containing regulators. Please note this as we will come back to the size of this space.
For a graph to be a valid network or a "Solution Graph," all its coat types must have at least one recruiter each and every vesicle fusion event must have at least one VAT molecule regulating its fusion. Therefore, a solution - possible molecular types regulating budding and fusion- occupy specific positions in this 536 bit-string long space, and VAT molecules that are never formed in the graph via the above check approach are referred to as neutral molecules to the graph, as they are considered neutral to its stability even if they are added to the graph. The unoccupied sites after removing the neutral VATs are the ones that relate to operations that were specifically not happening in the graph and are therefore combinedly referred to as the Forbidden Zone of the Graph and are reminiscent of cargo combinations that can break the specificity of the traffic network.

What is a Genome in this molecular landscape?

Among the molecules or molecular types that cover certain operations in the graph, some can cover multiple vesicle types' recruitment or fusion. Thus, a subset of these molecular types can be used to irreducibly 'cover' a graph. The molecular types that cover the same set of vesicle types get clubbed together to be called a Component, and there are multiple ways with which we can combine such components to cover every vesicle type's regulation and create an irreducible cover.

An irreducible cover can be visualized as a bipartite graph with the specific regulatory molecules in one column mapping to a specific function, either budding or fusion of specific vesicle(s) in the other column. This bipartite representation, as shown in Figure 2.3 below, can help us visualize which regulatory molecules can be clubbed into the same 'Component' by virtue of covering the same function for the same set of vesicles. Therefore a minimum molecular cover can be discovered by just finding the minimum vertex cover(explained in the next chapter in 6 of this bipartite graph in the column of Regulatory Molecules.

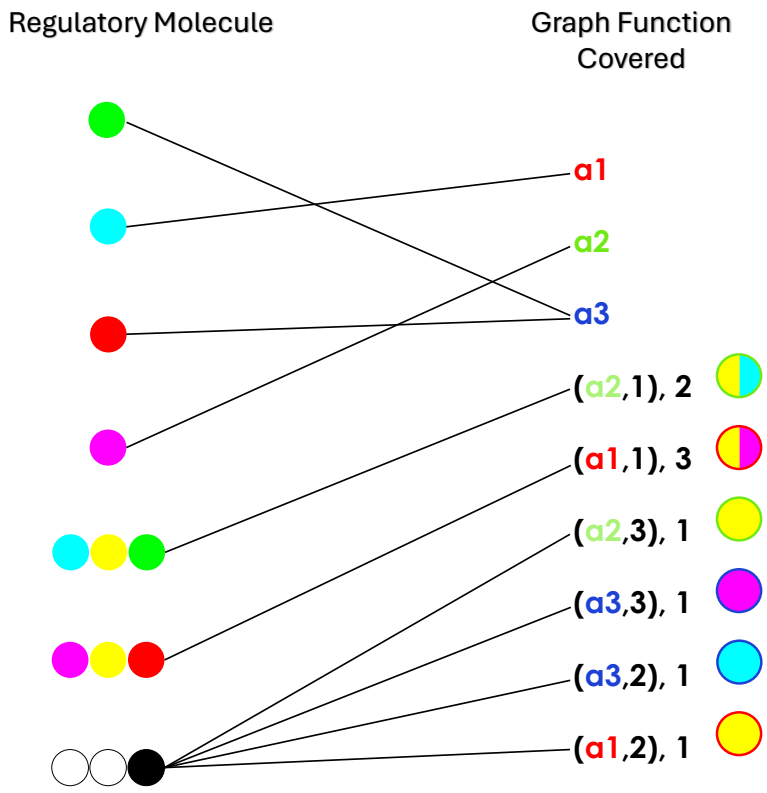


Figure 2.3: **A Bipartite Graph Representation of a Cover**: This is a visualization of an irreducible cover of previously graph G in Figure 2.2 (shown above).

We use the definition of an irreducible cover to make the process of becoming a different valid graph during evolution more simplistic as well as present all different but equivalent in terms of emergency from the same graph, raw materials upon which evolution can act, mutations can happen which lead to the irreducible cover becoming equal to the irreducible cover of another graph. In short, taking a minimal cover set of a larger set might allow us to cover a larger evolutionary landscape.
However, is this really 'The' genome of a VTN? If it is really the genome, then it should give rise to the same topology when clauses of these operations are reversed. However, is that really the case that we get back the same graph, or a much simpler question, is it possible to have multiple graphs with the same cover? Is it even tractable to compute the biophysical topology with just the blue-print of the genome as is the dogma of biology? These are some very important questions that we will come back to in the end.

2.4 Assumptions of the Model

Following are the assumptions made by Sahana and Mukund while creating the V-SNARE Activator Model: -

1. unique compositions of the compartments characterize the current-day Eukaryotic Endomembrane system.
2. Cargo types are not degraded in the ER (Rajakumar *et al.*, 2020; Hetz *et al.*, 2020; Smith *et al.*, 2011; Qu *et al.*, 2021; Read and Schröder, 2021)
3. The cargo types that are stuck at compartments, i.e., not on the cyclic route, i.e., on replenishment flow, are degraded slowly in the compartment(s) they are stuck at and are synthesized at a low rate in the ER, making the replenishment flow a weak flux of these cargoes that still manage to stay homeostatically at the target compartment(s).(Aridor and Traub, 2002; Kirchhausen *et al.*, 2014)
4. Only vesicles that carry cargo type(s) cyclically occur at different time scales than the replenishment cargo flows and are assumed to be flowing in a stronger flux.
5. There are no direct organelle-organelle interfaces for rapid transport of cargo types between compartments, only transport that happens through vesicles flowing is significant to the model.

It is important to note that the Coat-Cargo interaction matrix (Figure 2.1), provides asymmetry to the cargo types in terms of the number of interactions, and we can observe that in a 3-Coat system, we have cargo-types which can be classified into 3 Classes based on their multiplicity of interaction with coats: -

1. Class-1: These are cargo types 2, 3, and 5 or 001, 010, and 100 in their bit-string representation. These cargo types only interact with one coat type, and therefore it is important to note their probability of occurring on compartments is inherently lower.
2. Class-2: These are cargo types 4 (011), 6 (101), and 7 (110), which interact with exactly two coats.
3. Class-0: These are cargo types 1 (000) and 8 (111). Although these cargoes are completely flipped bit-strings of each other, i.e., have exactly opposite amounts of coat interactions, it is useful to classify them into one Class because both of them remain invariant in case of symmetric transformation. Cargo type 1 never leaves the ER, and cargo type 8 is present in all Compartments (assumed to be the lipid molecule).

Chapter 3

Methods

The Model of the Vesicle Trafficking Network described in the previous chapter, and its various variations, are created in the framework of a Classic Information Network, which is based on Directed graphs, with nodes specifying specific objects and the directed edge linking the flow of information between the two. Therefore, Directed Graphs, Graph Symmetries, Graph Traversal Algorithms, and Graph Connectedness are the core of the methods that we have used. These are explained in detail, keeping the V-Snare Activator Model in main focus, with illustrations created by the author in this chapter. The Algorithms are either taken from the Book Algorithms Design by Kleinberg, Jon; Tardos, Éva (Kleinberg and Tardos, 2006) or a modified version of them are used tailored to the question, some of which were also taught by Prof. Prafullkumar Tale in the course MT3264 ALGORITHM. (West *et al.*, 2001).

3.1 Search for Model Solutions Over Directed Graph Space

As explained in the Model section, we sample all possible Directed graphs to find solutions to our Model. A great deal of optimization was required for searching higher compartment (node) directed graph spaces.

3.1.1 Model Search Algorithm and Nature of Directed Graph Space

This is a subsection to Sec. 3.1.

Sahana and Mukund 's Algorithm has been summarized in the previous Chapter 2 Model. What is important in the Directed Graph space search is that the filtering of Graphs to be selected as topology to be labeled and checked for composition and consistency in emergent molecular features, happens at multiple levels.
First, the only topologies allowed are the ones with the maximum outdegree of each node equal to the number of coats in the system selected.
Second, the only topologies which are connected go further down the pipeline.
We then see the topologies being labeled with Coats and the ER selected.
A distribution of how the number of Solutions over the finally selected topologies to be labelled looks is given in Figure 3.1.

The way of selecting a compartment as the ER could be redundant because two nodes can have the same out- and in-edges, ultimately having the same labeling, thus selecting either of them as the ER does not yield a different graph, but merely an isomorphic (Graph Isomorphism explained in detail in 3.6.2). However, Sahana's Algorithm does not filter this as it creates different labelings for the two compartments' edges as well.

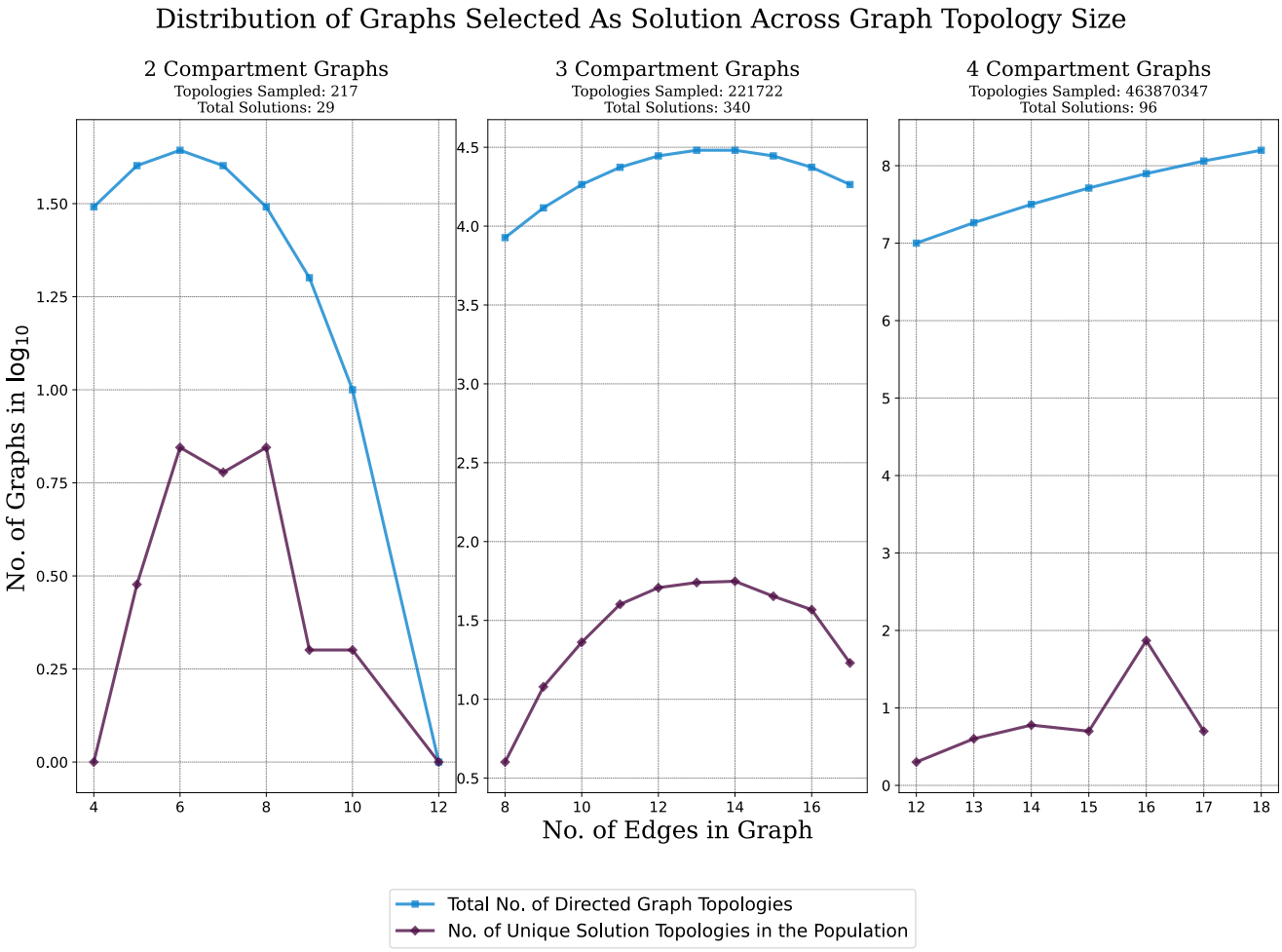


Figure 3.1: **Trend of No. of Topologies selected as Solution over no. of Topologies Sampled.**The trend of Solutions looks similar but more smoother than the topologies sampled, with an occasional rise in solutions peaking near but not always at the highest sampling. The trend of this rise and fall also suggests our sampling of 4 Compartment graphs has only just begun and is far from the endpoint.

3.1.2 Making Search Algorithm More Efficient: Creation of Whole Graph Space.

This is a subsection to Sec. 3.1.

Sahana's algorithm for creating all of directed graphs for a given number of nodes and edges is an intelligent method of first creating the adjacency matrix, as frequency of edges from row to column node of the graph as a multiset list with the sum of all elements in the multiset list equal to the total number of edges required in the graph. This adjacency matrix multiset list also has the constraint of maintaining the maximum element to be less than equal to the number of Coats selected.
We then need an algorithm that creates all possible permutations of the valid Multisets generated. Since Sahana's algorithm involved $O(n^2)$ computation to check whether the created permutations are unique, I had assumed the algorithm to be a Brute Force Algorithm which by definition creates all elements to all elements permutation, ultimately taking $O(n! * P)$ or simply $O(P^2)$which is again for number of permutations P being bounded above $n!$, and the P^2 checking time to retrieve the unique permutations amongst the created.

Therefore, I proposed the Recursive Backtracking Algorithm (Aggarwal *et al.*, 1999) for the creation of all multiset permutations.
Recursive algorithms involve the creation of sub-problems within problems which can be visualized in the form of a tree branching. This method is especially effective in Combinatorics, series, or sequences involving the selection of elements in possible sequences. Calling the same function over multiple times is at the heart of Recursion. And if we can prevent the redundancy in the number of times recursion is called, we can minimize the time complexity, and that is what the goal of Backtracking is in the Algorithm explained below.

All Graph Permutations using Recursive Backtracking Algorithm

Algorithm 1: Recursive Backtracking Algorithm for Multiset Permutations

Data: List '*lst*' of size n and unique elements k '*element*', and $m_{i1}, m_{i2}, \ldots, m_{ik}$ for m_i or '**count**' denoting multiplicity of i^{th} element.
Result: List '*result*' containing all the permutations of multiset *lst*

```
begin
    Initialize a Dictionary object 'counter' to count element occurrences in lst
    Initialize an empty list result to store the permutations
    Call permute() with counter, an empty list for the current permutation,
     len(lst), and the list result
end
Function permute(counter, current_permutation, remaining, result)
    if remaining == 0 then
        (Valid Length Permutation Creation Finished)
        Append current_permutation to result
        return
    end
    for each element, count in the list of items from counter do
        if count > 0 then
            Decrement counter[element] by 1
            current_permutation + [element]
            Call permute() recursively with current counter,
             current_permutation, remaining − 1, and result
            Increment counter[element] by 1 (backtrack)
        end
    end
return result
```

A Description of the Algorithm:

A Dictionary 'Counter' object is created to store the element types and their occurrences from the inputted list, which is a list of edges –$[1, 1, 2, 2]$ (Example given in Figure 3.2. The Algorithm's recursive function 'Permute' loops over different types of elements in this Counter, and inside each iteration, it calls itself again over a Counter object which has one less occurrence of the element selected in the outer iteration, creating a new layer with all possible path from one element to another. This creates trees with multiple branches on every element type in the original list, and the Counter object ensures that no element type is repeated in the next level more than its oc-

currence value, effectively giving unique branches as we reach the end-points.

Another upside of this algorithm is that we don't need to traverse the created tree again because each path was stored in the *current_permutations* sub-list. Upon reaching the leaf node, these paths are directly stored in the *result* list, which is then returned.

The total number of permutations of multiset of k unique elements and $m_{i1}, m_{i2}, \ldots, m_{ik}$ for m_i denoting the multiplicity of unique element $i \in i_1, i_2, \ldots, i_k$ and n equal to the length of multiset is

$$= \frac{n!}{\prod m_i!}$$

The Counter object helps in keeping track of elements already encountered therefore recursion calls are minimized, and if we try to check the time complexity of this algorithm which is basically counting how the number of nodes has grown layer by layer, we get an upper bound of $O\left((m_1 + m_2 + \ldots + m_k)!\right)$ where $m_1, m_2, \ldots, m_k$ represent the multiplicities of the unique elements in $\{i_1, i_2, \ldots, i_k\}$ from the Input Multiset list. This is a significant improvement over brute force because, for a fixed number of compartments, the length of the list remains the same equal to $|nodes| * |nodes|$ but the multiplicity is bounded above by number of coats, and their sum by number of edges. Thus the worst-case Time complexity $O(n!)$ (Aggarwal *et al.*, 1999; Juric and Siljak, 2011) is encountered in the case when the number of unique elements equals the length of the list, which does not happen in 3-coat systems.

Improvement by Recursive Backtrack Algorithm and use of Hash Data Structure

It was only after some time that I discovered that the pre-existing method of creation of permutations of multiset was done by a recursive algorithm through the use of the Python module *sympy.utilities.iterables.multiset_permutations* which is a generator type object and also uses keeping the count of each

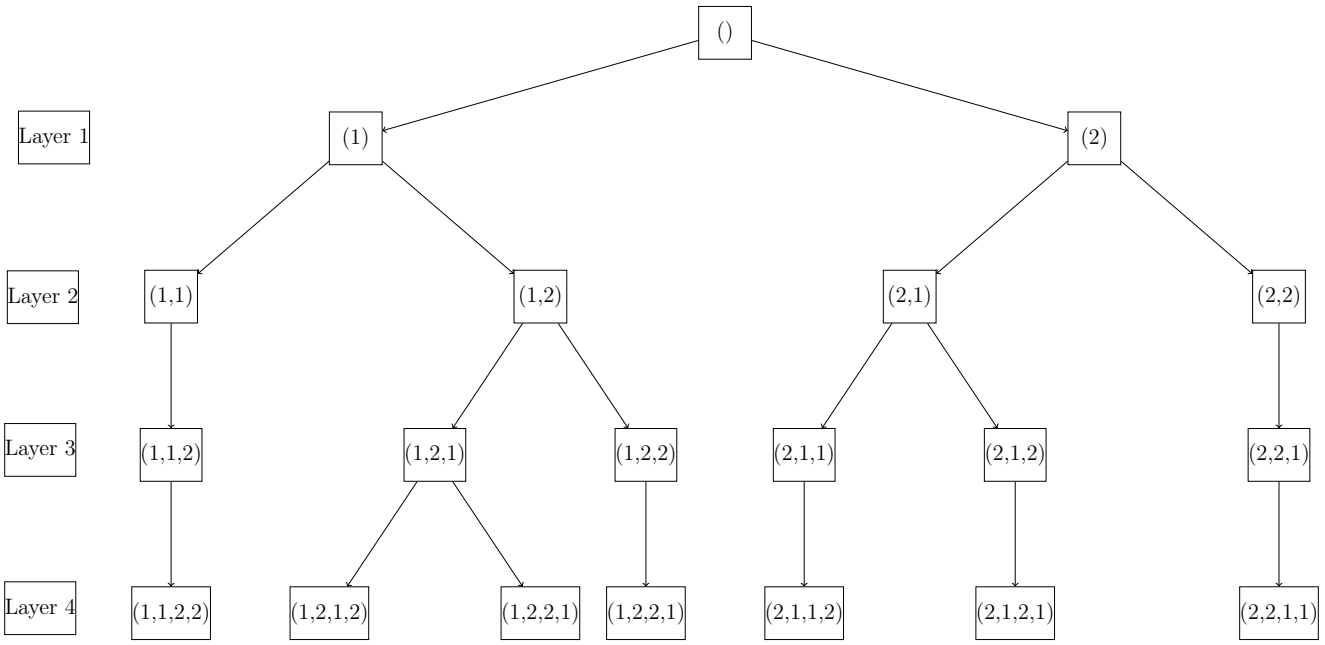


Figure 3.2: **Recursive Backtrack Algorithm Creates Tree of permutations** An example for multiset –$[1, 1, 2, 2]$.

unique element for the creation of permutations, however due to the Generator object nature of the algorithm a tree is created which needs to traversed (which takes constant time $O(n)$) to retrieve all the permutations at the leaf nodes (nodes at the bottom most layer of the tree where it does not grow further). But this overhead is not very significant as we can see in the Figure .

Why, then, do we experience an extended period of time delay in this process?

it turned out that a check was performed after converting the multiset lists, which are actually adjacency matrices back into edges list format. Now all permutations, say, cardinality P, were being checked for whether they are connected in a simple undirected graph format. However, after this check was performed, there was $O(n^2)$ operation of taking out unique elements being performed for the P graphs in the data type of list of lists. So a $O(P^2)$ operations were being performed unnecessarily when all the permutations were already unique.

For the check of uniqueness or any data comparisons, Hash tables have long been known to be the most efficient data type because of their specific na-

ture of arranging elements by the calculation of their hash value which takes constant time (Maurer and Lewis, 1975; Konheim, 2010). An analog of the hash table is the '*set*()' function in Python (Van Rossum and Drake Jr, 2014). Thus for every element in the list which is now converted to a set using this function, instead of checking every element in the set to see if it already exists, a hash value of the element is calculated and the data point corresponding to this hash value is checked for whether it is already occupied or not. This decreases the computation significantly for higher edges, as shown in the figure 3.3.

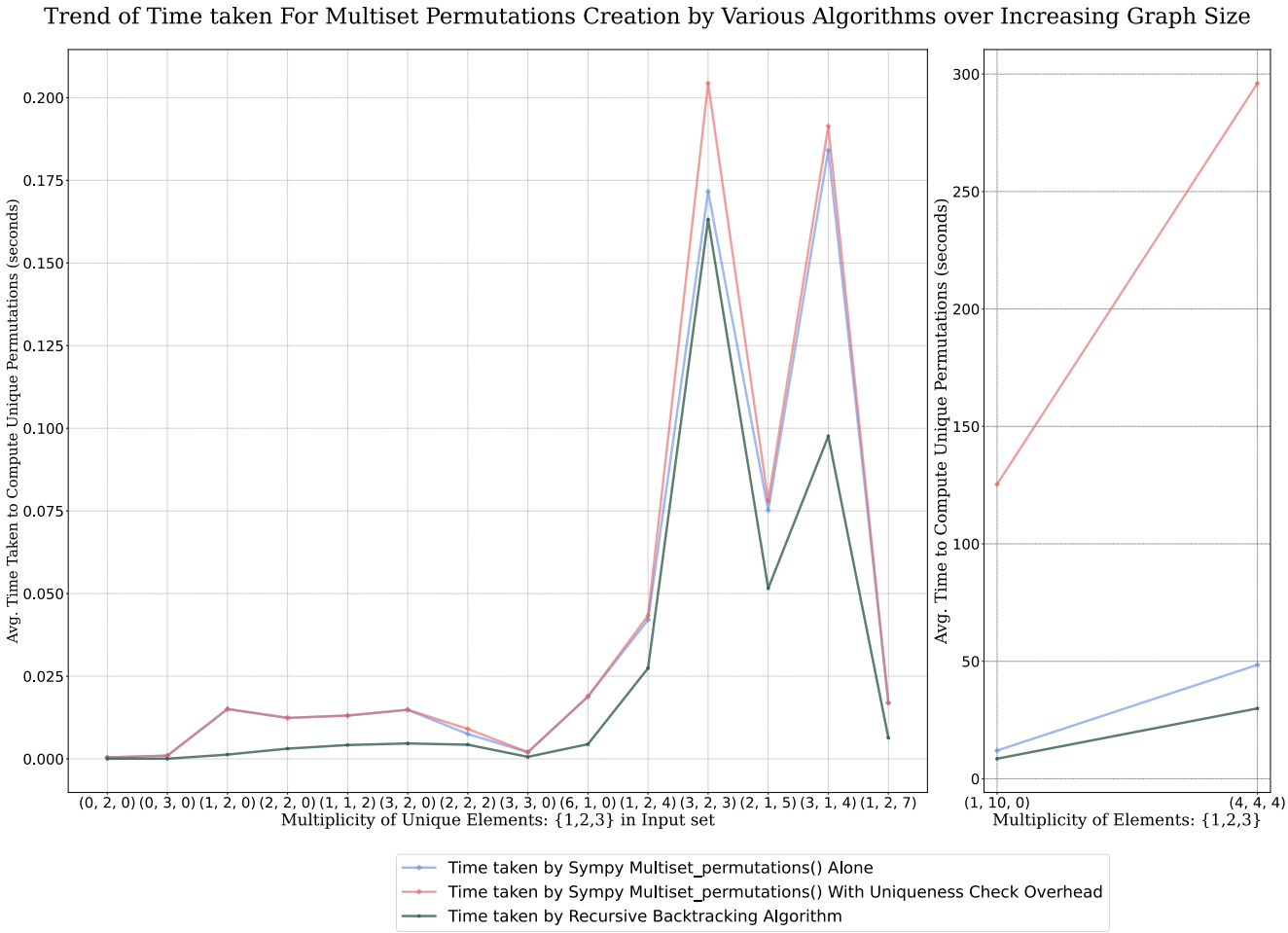


Figure 3.3: No. of Solutions Distribution in Background of Regulatory Landscape of Valid and Invalid Molecules, Across Graph Sizes

3.2 Cluster Parallelization Methods in Solution and Non-Solution Search

Then the next step in making Solution search more efficient was breaking the large space of Graphs generated down into chunks, and parallely run the Model Algorithm on them to get solutions and save time.

Initial methods of parallelization included MPI (dalcin2021mpi4py) (mpi4py python module) on a cluster of cores on the cluster. However, it was not efficient and even gave memory allocation exceeded errors even when a lot of cores were called.

3.2.1 The 3 Clusters

This is a subsection to Sec. 3.2.

It is important to define what a cluster is: -
A cluster is a connected set of nodes with interactive User interface and fast internet connectivity for performing heavy computation, and a cluster with a rapidly connected and numerous cores with a software framework to handle large data processing is called High Performance Computing Cluster.
At the National Centre for Biological Sciences, there are 3 Clusters: Tiramisu, Nargis and Pakeeza, ordered in terms of lowest to highest efficiency. The Schedular system, which is software or Dynamic Resource Manager of relaying program running requests to a set of cores and/or nodes, is Open Grid Engine/ Sun Grid Engine in the 3 aforementioned clusters.

I utilized 2 methods of parallelization to SGE Schedular, one of which is hard to implement in SGE Schedular because it is an older software and Cluster permissions were not available.
These methods are written in BASH, which is a language used to communicate with the system and receive text based output. Also allows encoding of complex operations like redirection of outputs and creation of pipelines. A

specific Scheduler, like SGE or Slurm or PBS has its own specific lingo for communicating with the user through BASH.

3.2.2 Print Commands to Shell

This is a subsection to Sec. 3.2.

Isolation of certain sets of nodes and using a list that has all the commands. Now the nature of these commands is such that they call a subset of the big graph space through the index of chunks of the whole list stored outside.

Algorithm 2: Parallelization of Chunks of Lists in an Isolated Pool Cores

Data: Path to the list of commands *cmd_list*.
Result: Submit each command as the job to run from the *cmd_list* on a separate core.

Function *print-cmds*(*cmd_list*)
  foreach *command cmd in cmd_list* **do**
    submit-cmd(*cmd*);
  end
  wait();

Function *submit-cmd*(*cmd*)
  Submit the job to the individual core using *qsub* command with required flag parameters;
  $qsub - pesmp1 - bindinglinear : 1 - Nmy\_job\_name << EOF$;
  Execute command using $bash - c$"cmd" EOF;

Function *wait*()
  Holds the isolated cores until all submitted jobs after printing each command finish;
  return;

This algorithm was adapted from its Slurm implementation and therefore had a problem; such submission to individual cores was not allowed in any of the 3 Clusters. Therefore I had to remove the line where the submission of jobs takes place inside the submitted job to a specific core/slot and keep printing commands directly to bash using just $bash - c$"cmd" .
This resulted in sub-optimal enhancement of speed, and therefore the method Array Job Submission was called into the picture.

3.2.3 Array Job Parallelization

This is a subsection to Sec. 3.2.

Array job submissions are known to be faster because of the complete isolation of each task onto a job and the return of output directly without the need for multiple cores communicating.

Algorithm 3: Parallelization of Chunks of Graph Space by Array Job Parallelization

Data: A Specified index of arrays contained in *cmd_list* as commands, and the number of Array jobs that can be running at one timestamp.

Result: Submit each command in the specified index of arrays from the *cmd_list* as an array job to run on 1 core as a separate isolated job.

Function ***Read-Execute****(cmd_list)*

Set Internal Field Separator to newline: $IFS \leftarrow' \backslash n'$;
Read lines from *cmd_list* into array: $array \leftarrow$ read -d '' -r -a array $<$ *cmd_list*;
Index the array with SGE_TASK_ID as a variable
$var \leftarrow array[SGE\_TASK\_ID - 1]$;
Bash prints or 'echos' into command line *var*;
The Array Job with the specific ID gets submitted through *var*;
return;

3.3 Graph Transitions

To define if a transition is allowed between two graphs that are regulated by their molecular types, most of which are only one coat interaction change away (see Figure 3.4) from each other, we define how far is the molecular type in the other graph in terms of coat-affinities, while comparing every irreducible cover of one graph to every irreducible cover of the other graph until we reach the exact same pair. Molecular types that only differ by one coat interaction change are called one-step mutation or flip away, with the underlying assumption of point mutations changing targeting signals in the vesicle trafficking of a protein. (Ref)

A VAT molecule is also one step away from 24 other VAT molecules, which differ from it in either its V, A or T - SNARE cargo type, which can be any among the 8 cargo types. This creates a Hypercube of 512 nodes.

With the molecular distances, we have two kinds of molecular transitions.
(1) **Bistable transition** where the exact same minimum set of molecules from the available components of the two networks can be found. Here bistability refers to the fact that two different graphs could be achieved with the same minimum set of molecules. In our system, we also see multis-stable graphs with the same minimal(and not always minimum) molecules regulating them.
(2) (1) **One step away transition** when there is an overlap of one graph's component and the other graph's forbidden zone. If this transition is possible then it implies that a graph needs to destroy certain or all operations of its network in order to become the other graph, as it is acquiring a molecule that perturbs the network (or in some cases, deleting a molecule that is integral to itself). These transitions are possible if-

at least one of the molecules in this overlapping component can become any molecule present in the cover of the other graph by one flip or

in case one component from each graph overlaps the other graph's forbidden zone, then the transition is allowed only if there's at least one molecule from each of these different graph components that can reach the other in one step.

3.3.1 Hamming Distance and Grey's Code

This is a subsection to Sec. 3.3. **Hamming Distance**: The Hamming distance is defined between two strings of equal length. It is the number of positions at which the corresponding symbols differ. For example, for the binary strings of cargo types 2 '001' and 4 '011', the Hamming distance is 1, as only bit at position 2 differs.

Gray Code: Gray code, or reflected binary code (RBC), is sequential ordering in Binary systems of bit-strings in way that the succeeding bit-string only differs from the preceding one at one bit-position, i.e., the successive strings occur at one Hamming Distance from the adjacently previous ones.

According to Sahana and Mukund's Model, 2022 (Unpublished), when a component of one graph is completely inside the forbidden zone of another graph, the number of mutations it requires to get out is calculated by ensuring that there is an element in the other graph's non-forbidden zone (which may not be in the components) which is only one Hamming Distance from it.

Neutral Walk and Minimum Hamming Distance Continuing the above argument of mutating out of the forbidden zone to some neutral or non-neutral zone of the other graph, but it still needs to become the other graph by becoming the necessary regulator(s). This takes place by a "neutral-walk" which is going from one molecule to the required non-neutral molecule in the component, which will require minimum flipping of the cargo types' bits in bit-strings. Thus the path of "neutral-walk" involves going through the minimum Hamming distance to the required bit-string set. This is a very computationally hard problem to solve since the number of Bit-string sets a single VAT molecule can go to 24 other VATs with one bit change. Therefore discovering the most efficient path will involve the comparison of the order 24^n for n bits that needs to be changed to reach the other graph's VAT molecule, and not just that because some of these VATs along the branching paths possible might be in the forbidden zone and therefore a careful graph traversal be required.

3.3.2 The One-Mutation Bit-String Cube of Cargo-types.

This is a subsection to Sec. 3.3. This cube in Figure 3.4 represents the allowed evolutionary mutations that are considered single-step mutations.

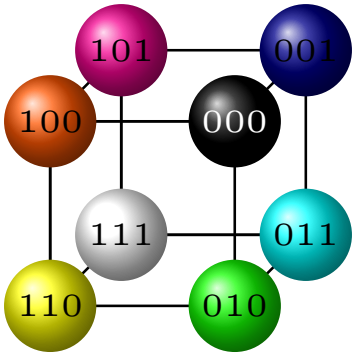


Figure 3.4: The Cube of Cargo types' Bit-Strings at One Hamming Distance

3.4 Graph Properties

3.4.1 SCC, Closed Cycles, Self-Loops, Cliques, and Topological Motifs

This is a subsection to Sec. 3.4.

As mentioned before, the definitions are either taken from the Book Algorithms Design by Kleinberg, Jon; Tardos, Éva (Kleinberg and Tardos, 2006) or from (West *et al.*, 2001). I use the terms 'nodes' and 'vertices' to describe the points in a graph interchangeably.
Please see the book (West et al.*, 2001) or online sources for the definitions of a Graph, a Directed Graph, an out-edge, and an in-edge, because I am not defining them here.*

Strongly Connected Components (SCC) in a Directed Graph
We know that a directed graph is strongly connected- " if, for every pair of nodes i and j for $i, j \in Nodeset$, there is a path from i to j and a path from j to i." (Kleinberg and Tardos, 2006)

Cycles: "In a directed graph, we can define a cycle as a sub-graph on two vertices, both of which are connecting the other by an out-edge to the other" (West *et al.*, 2001). Unless specified otherwise, when a 'cycle' is mentioned in the graph it means on two and only two nodes/compartments of the graph. Instead of calling this a 'directed cycle,' we will use only 'cycle' because we are majorly dealing with directed graphs here.

Self-Loops: In graph theory, a self-loop is defined as the path of an edge which starts and ends at the same vertex. Or, simply put, a closed loop on a vertex in a graph, in a directed graph, this represents the flow of cargo from the same node back to itself.

Closed Cycles: Based on the aforementioned definitions of self-loops and Cycles, we define a Closed Cycle on our edge-labeled graphs to have a closed cycle if and only if two or more nodes form pairwise Cycles with the exact same labeled edge and a self-loop formed by the same label exist on all of these nodes.
Such closed cycles represent a specific kind of flow, which we will talk about in Results Chapter 4.

Half Closed Cycles: A half-closed cycle, as opposed to a Closed cycle only defined between two nodes, where a directed edge exists from one vertex to another and no backward edge to the first source vertex (we will simply use the term 'source' instead of 'source node/vertex' from hereon for simplicity), and there exists a self-loop on the same source. Both these forward and self-loop edges must have the same label. These Hald Closed cycles by definition have only one source, and we would refer this as the 'root' of the half-closed cycle or 'this is where half-closed is rooted at' in the text and the proofs in 4 chapter.

Heterogenously Closed Cycles: When a half-closed cycle from a source to target exist with another half-closed cycle which is , as opposed to a Closed cycle only defined between two nodes, where a directed edge exists from one vertex to another and no backward edge to the first source vertex (we will simply use the term 'source' instead of 'source node/vertex' from hereon for simplicity), and there exists a self-loop on the same source. Both these forward and self-loop edges must have the same label.

Cliques: "A clique in a graph is a subset of vertices where an edge connects every pair of vertices"(West *et al.*, 2001), i.e., a complete subgraph with all nodes adjacent to each other.

Topological Motifs: We define topological motifs as a specific arrangement of nodes with respect to each through the specific connectivity, which can include directed flow by specific labeled edges. The motifs might be recurring or emergent patterns in networks or graphs of graphs. A Clique, or an arrangement of two different labels opposite half-closed cycles on the adjacent nodes, the occurrence of all three labels as self-loops, the occurrence of one half-closed cycle of the specific label and two backward edges with different labels from the target node, etc., are a few examples of topological motifs which will be defined and used in the thesis.

3.5 Graph Traversal Algorithms

To explore the structure and properties of directed graphs, such as the aforementioned closed cycles and motifs, a systematic method was required which allows iterating through connections between the nodes and stopping when the property(s) is/are found. I explain the two most basic traversal algorithms that are only for the flavor of the actual tailored algorithms that I have used in my research below. Mentioning all the algorithms I used would take unnecessary space and time, and will be supplementary information which maybe presented in the upcoming paper (Shridhar, K. Kumari, M. Thattai, 2024 Unpublished).

3.5.1 Breadth First Search

This is a subsection to Sec. 3.5.

'Breadth-First Search' or 'BFS' Algorithm is way of exploring the graph by creating a layered tree rooted at (starting from) from a specified source

node and traversal, or enlisting of all neighboring nodes in the next adjacent layer before moving to the next. The structure of the tree created is such that if we go from layer 0 with the source node at layer 0 to the nth layer, we would traverse a sequence of nodes which are at increasing distance from the source(for proof, please refer to (Kleinberg and Tardos, 2006)). BFS is, therefore, particularly useful for finding the shortest paths in graphs and for the identification of connected components.

BFS Algorithm is 4, described below in the pseudocode below.

Algorithm 4: Breadth First Search (BFS)

```
Data: Graph G represented as an adjacency list, starting node s.
Result: Traversal order of nodes in BFS.
Function BFS(G, s)
    queue ← empty queue;
    visited[s] ← true;
    queue.enqueue(s);
    while queue is not empty do
        v ← queue.dequeue();
        visit v;
        foreach neighbor w of v do
            if w is not visited then
                visited[w] ← true;
                queue.enqueue(w);
            end
        end
    end
```

3.5.2 Depth First Search

This is a subsection to Sec. 3.5.

'Depth-First Search' or 'DFS', unlike BFS, is not focussed on finding all neighbors of a particular node selected as it explores a graph by traversing as far along each branch as possible before backtracking and looking at other immediate neighbors of the starting node. This allows DFS to quickly detect cycles or strongly connected components in a graph and is used in the VTN Models in the part of discovering the compositions of nodes and vesicles.

DFS Algorithm is 5, as described below in the pseudocode below.

Algorithm 5: Depth First Search (DFS)

Data: Graph G represented as an adjacency list, starting node s.
Result: Traversal order of nodes in DFS.

Function ***DFS***(G, s)
    $visited[s] \leftarrow$ true;
    visit s;
    foreach *neighbor* v *of* s **do**
        if v *is not visited* **then**
            DFS(G, v);
        end
    end

3.5.3 Minimum Vertex Cover

This is a subsection to Sec. 3.4.

"A vertex cover of a graph $G = (V, E)$ is a subset $S \subseteq V$ such that every edge in E is incident to at least one vertex in S."(West *et al.*, 2001) The problem of finding a minimum vertex seeks to uncover the smallest possible set, or minimum cardinality, of vertices from which you can reach all the other vertices in the graph using the edge connections that exist.

The Minimum Vertex Cover problem proves to be an integral part of realizing the compression of data in a connected system, and we use this to realize the properties, such as symmetry and molecular space, that cluster our graphs into groups so that we can see the minimum principles upon which individual graphs are built.

Although the Minimum Vertex Cover(MVC) problem is $\mathcal{NP}$-Hard and can take up enormous time if an absolute algorithm such as as given below 6, to work of larger graphs. I have tried using Approximate Algorithms to discover MVCs where an absolute Algorithm was not possible but required. Fortunately, this problem is easier if each cluster of nodes segregates into a component and the component is a complete graph (West *et al.*, 2001) or a 'clique' on the larger landscape of graphs.

Algorithm 6: Minimum Vertex Cover in an Undirected Connected Graph

```
Data: An Undirected tree T with n nodes.
Result: Minimum Vertex Cover of the tree T.
Pick a vertex r to be the root of this Graph tree T.
Modify the way T is encoded to ensure that nodes further away from r are
 represented by smaller indices (if T is stored as an adjacency matrix, this would
 mean rearranging the rows and columns).
Initialize array vertex_cover of size n with all entries set to 0. for i ← 0 to
 n − 1 do
    Let u be i-th node of T;
    if u has no children then
        vertex_cover[i] ← 1;
    else
        all_children_covered ← true;
        for each child v of u do
            if vertex_cover[v] = 0 then
                all_children_covered ← false;
            end
        end
        if all_children_covered then
            vertex_cover[i] ← 0;
        end
        else
            vertex_cover[i] ← 1;
        end
    end
end
return vertex_cover;
```

3.6 Symmetries and Groups

A symmetry group of an object, in mathematics, is defined as the set of all transformations that preserve the object's shape, size, and orientation. (Weyl, 1952; Alt *et al.*, 1988; Hoffmann and Kunze, 1971; Armstrong, 1997; Martin, 2012; Conway *et al.*, 2016)
As depicted in the Figure 3.4 The Cube of Cargo types' Bit-Strings at One Hamming Distance, we see a cube, and a 3-D cube is known to have 48 isometries or 48 elements in its symmetry group (Armstrong, 1997; Martin, 2012; Conway *et al.*, 2016). 24 of which are formed by the 3 Rotational Axis(Diagonals: rotation along x-,y-, and z- axes) and the formation of 24 permutations of rotation transformation by any two axes of rotation, along with 9 Reflection planes.

However, in our Model, we can clearly see due to the symmetry among cargo-type bit-strings, we have only 1 Rotation Axis that passes through cargo type 1(000) and 8 (111) because these remain unchanged by these operations. And this Rotation Diagonal has the order symmetry equal to 3, as in 3 rotations we reach the same cargo type bit=string again. Given that the order is 3, we can see that this is an odd permutation group.
There are 3, instead of 9, planes of reflection in our model due to inherent symmetry in bit-string again. Each of these reflection transformations has a symmetry order of 2, making reflection in our system an even symmetry operation.
A Reflection plane can be visualized on the Bit-string Cube using the string vertices, which remain unchanged by the specific reflection operation, and the plane passing through the Rotational symmetry axis, intersecting 000 and 111 nodes that remain unchanged under any reflection plane.

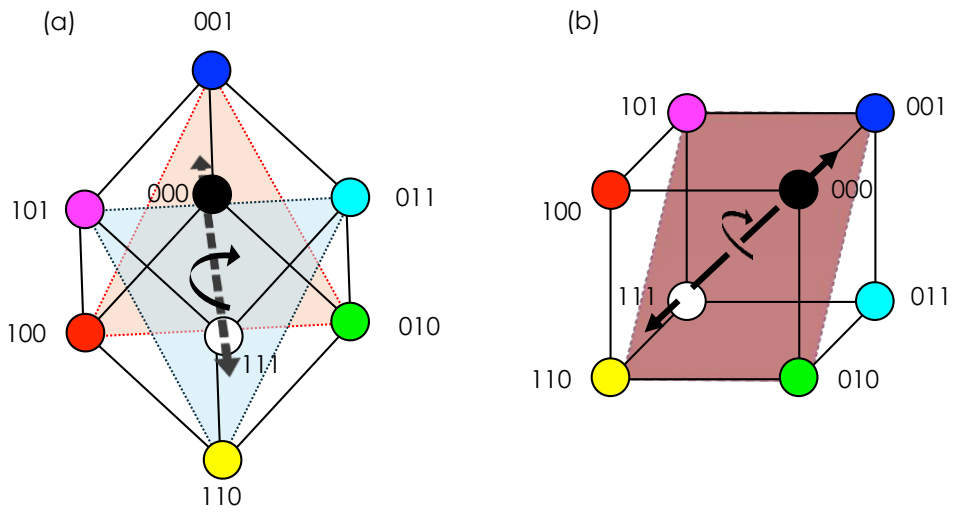


Figure 3.5: **Symmetry Operations on the Coat-Rotation-Reflection Symmetry Cube**: (a) Grey Line from Bit-string 111 to 000 shows the Main Rotation Axis going through the Blue and Pink shaded triangles. The Triangles mark the bit-string with the frequency of 1s and 0s, demarking Bit-string inside each triangular plane is reachable via symmetry but not to strings on the other triangle. (b) An example of Reflection plane when (a1,a2,a3) goes to (a2,a1,a3). We can create the Reflection plane on the Bit-string Cube using the string vertices 001 and 110 which remain unchanged by the specific reflection operation and the plane through the Rotational symmetry axis that intersects them. The diametrically opposite strings not touched by the plane are the resulting strings related by the reflection applied.

3.6.1 Rotation-Reflection Symmetry and Cargo-type Relations: The Coat-Rotation-Reflection Symmetry Cube

This is a subsection to Sec. 3.4.

We can also observe from Figure 3.5 above that Class-1 cargo types can transform only to Class-1 cargo types, and the same for Class-2 cargo types. This is captured in the triangular planes in the same figure.

It is also important to note the cargo types on these planes are connected by both Rotations as well as reflection symmetry operations, one from each

symmetry operation.
For a specific operation over bit-string, by rotation or reflection, we observe that there are flipping of two bits that takes place, and if again a rotation of 120 Degrees is applied two more times, the position comes back to its initial state, while undergoing 6 bit-flips.
It is also important to note these cargo types that are reachable by symmetry operations are 2-Hamming Distances away from each other.

Theorem 3.6.1. *Each Bit-string-Pair Related by Symmetry Operation is either only 2 or 0 Hamming Distance Away on the Coat-Rotation-Reflection Symmetry Cube.*

Proof. This is apparent from the fact that bit-strings that are 3 hamming distance away will require at least 3 bit-flips or changes in 3 different positions. We also know that if two bit-strings are related by 3-bit flips must have different frequency of (both) the 1s and 0s when compared to each other.
Whereas in our 7 isometric operations with 3 rotations, 3 reflections and one identity transformation, the distribution of frequency of either 1s or 0s is always preserved. Given that we have 3 bits and only 2 Classes of frequencies (either 1 1 and 2 0s, or 1 0 and 2 1s), we observe a 2 bit flip change under non-identity transformations over Cargos 2 to 7 or 0 bit flip change for 1 and 8, or 0 bit flip change for all cargos over Identity transformation. □

3.6.2 Graph Isomorphism

This is a subsection to Sec. 3.4.

"Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be isomorphic if there exists a bijective function $f : V_1 \to V_2$ such that for every pair of

vertices $u, v \in V_1$, $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$." (Kleinberg and Tardos, 2006)

Therefore, two graphs are isomorphic if and only if they can be superimposed and are completely congruent, i.e., we make the current graph G_1 into the other G_2 by a simple relabelling of the vertices and edges without disturbing the connectivity between nodes.

Although Graph Isomorphism is not considered a $\mathcal{NP}$-complete problem, it is a difficult problem to solve in larger graphs, and we also had an added complexity of labeled edges. However, I exploited the added complexity of labels to make this problem easier by first superimposing the ER of the two graphs (this is a step that might be skipped in some analyses, but if it is used, then it is mentioned) and then only comparing the edges belonging to a specific label in the graph to the specifically labeled edges in the other graph so as to ensure if the coats have rotated or reflected symmetrically the other coat-labeled edges would acquire the coat, and again finding a bijection of one coat-labeled edge set to another for each coat in the pair of graphs. This has been visualized in a very small and simple example below.

Figure 3.6 shows a simple example of Isomorphism in Directed graphs.

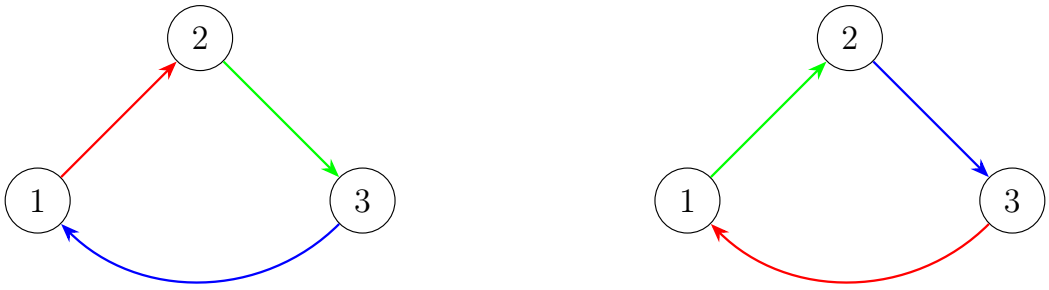


Figure 3.6: **Two Directed Isomorphic Graphs** We can clearly see that the bijective function here is 1 -> 3, 2 -> 1, and 3->2 for the nodes and (1,2) -> (3,1), (2,3) -> (1,2) and (3,1)-> (2,3).

Chapter 4

Results and Discussions

We discuss the results of this thesis in 4 Sub-parts, which are- Model Solution Search, Discovering the Evolutionary Trajectories Through Molecular and Functional Definitions , Discovering the Intermediates and Constraints along the routes between FECA and LECA by the Study of the Splittability of Smaller Graphs into larger ones.

4.1 Graph Search Parallelization Results

We have sampled over at least 100 Billion Graphs with a total of 1.4 Million CPU Hours in the course of 9 months of this Thesis. And we have discovered only 22,788 Graphs that are solutions to our V-Snare Activator Model. The time it takes for the Model's Algorithm on the most efficient Cluster (Pakeeza), to discover whether a graph is a solution, is on average 78.765 seconds on one core. Interestingly, if the number of cores are increased for one input topology, the time taken increases by 5 seconds! This is a gist of the reason of justifying why Array Job parallelization is much more efficient than Print Command to Cores in an isolated Node-pool (which does not work in an SGE cluster scheduler such as the Clusters of NCBS).

An estimate of how much improvement was achieved by either of these methods to previous runs is slightly difficult task because of multiple failed runs with general MPI parallelization before a definition node-size directed graph space search could complete.
Not counting the improvement in speed from creation of Graph Space which was improved by 10 times, we can calculate the improvement through parallelization by dividing the product of total number topological permutations sampled and time taken for each by the number of average cores utilized during permutation: -

$$463870347 \times 78 = 36181887066$$

$$\frac{36181887066}{500} = 72363774.132 \qquad \text{let average cores used be} = 500$$

$$\frac{72363774.132}{3600} = 20101.04837$$

20,101 Hours for the search of all possible labellings of 463,870,347 4 compartment graphs (bounded above by 1 Trillion)

But how much more efficient did Bash Cluster Parallelization prove over MPI?
For the same number of cores, an estimate of 1.5 months for MPI parallelization to 5 days for Array Job Bash parallelization for the graph of size 4 compartment and 12 edges is the efficiency achieved by this method. Why was MPI parallelization not as efficient is a question for Computer Scientists and the High Performance Computing Experts. Array Job Bash parallelization also scales linearly with the number of cores used.

In the Table 4.1 below, the results for the space sampled, the solutions found and sparsity of solutions in the space is shown.

Compartments	Max-Edges	Networks-Sampled	Solutions	Sparsity:Solutions/Space
1	3/3	1	1	1
2	12/12	3,622	990	0.273
3	17/27	37,977,283	17,646	0.0005
4	15/48	$\leq 1 \times 10^{12*}$	4,152	$< 1 \times 10^{-9}$

Table 4.1: **No. of Solutions Over Sampled Space, and the Sparsity in Solutions across Node-sizes in graphs.** The * denotes an approximated calculation of all possible labellings over 463,870,347 graphs.(Table was made in pdfplots LateX)

4.1.1 Graph Solutions and Molecular Landscape

This is a subsection to Sec. 4.1.

Why does the number of solutions decline when the number of Valid molecules are increasing? Easy, the answer is what topologies allow the graphs to be a solution. It is the variation in these valid graphs that determines the nature of Molecular space. As we can see in Figure 4.1 the number of VATs go down, while the number of recruiters increases, and that's natural because recruiters are a union of certain operations, whereas VATs need to be an intersection or unions outside of a forbidden set. As the number of vesicle types increases, the number of coats recruited to compartments

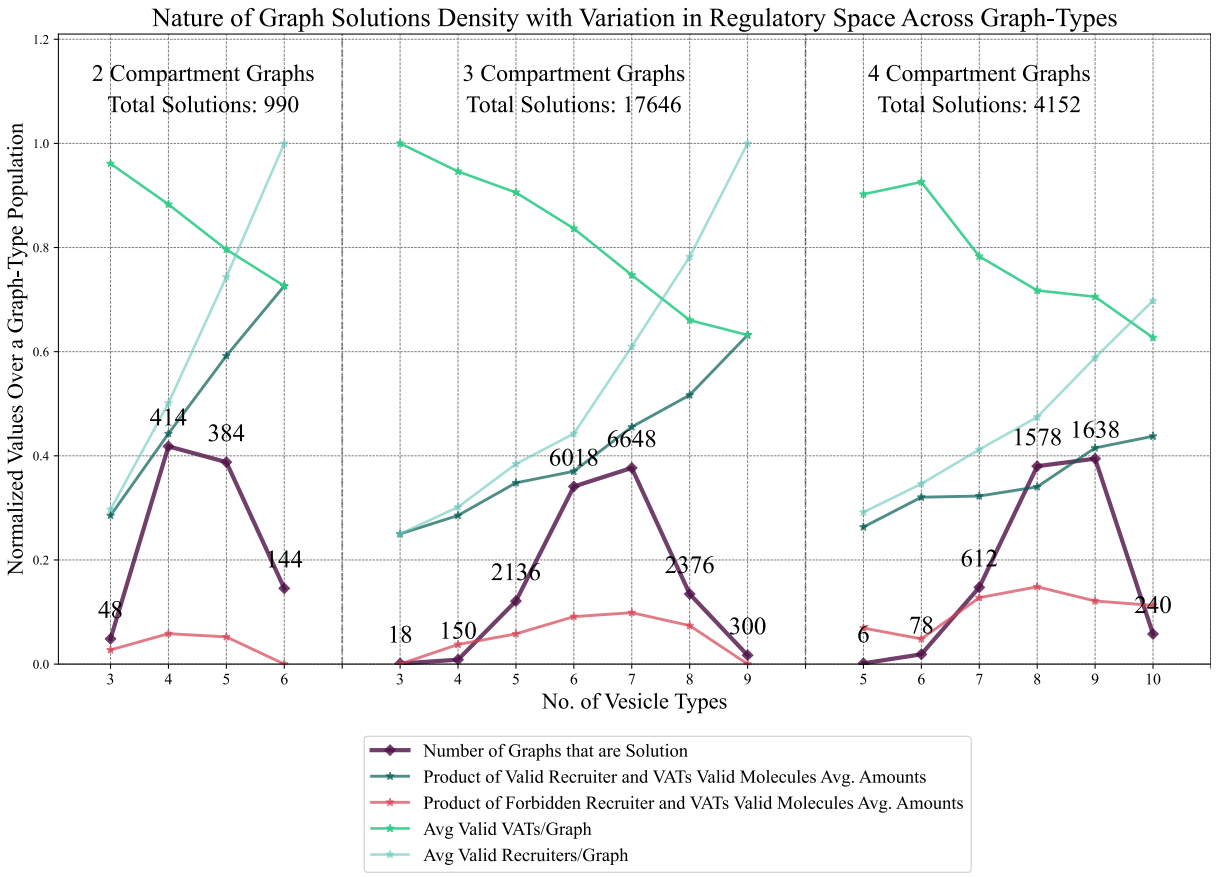


Figure 4.1: **No. of Solutions Distribution in Background of Regulatory Landscape of Valid and Invalid Molecules, Across Graph Sizes** The unit of Vesicle types (source compartment, coat-label) for edge is used to check the variation in other topological or emergent molecular properties. The number of solutions peaks slightly towards the peak of forbidden VATs and recruiters. The product of valid VATs and recruiters keeps going higher.

increases.

However, we observe a dip in the number of solutions which is not due to lowered sampling of graphs which yield higher vesicle-type size. It is due to the fact that the number of markers on compartment go down because the number of cycles in topologies increases as shown in Figure 4.2, because the increase in vesicle types indicates the number of coats on a compartment has increased if this increase in coats on all compartments then an increase in a number of self-loops to allow fusion of same cluster of cargo going out and coming in and therefore closed cycles is bound to happen. This, however, constraints the topologies which can have unique compartment compositions at a steady state.

Theorem 4.1.1. *Vesicle with exact same cargo types composition will not have any valid VATs unless they have the same targets they are fusing to.*

Proof. Let's assume that this is not true, and vesicles with same composition can have valid VATs even when their targets differ. A valid VATs are created by V-Snare and A-Snare selected with replacement from the Vesicle composition, while the T-Snare is selected from the Target compartment composition to which it is fusing. Similarly, invalid or forbidden VATs are created by keeping the V-and A-Snare combinations the same but selecting T-Snares from non-fusion compartment(s) composition.

Having exactly the same vesicle composition yields exactly the same V- and A- Snare combinations. Therefore we get the exact same combinations of VA and T-snare if we select T-snares keeping the target compartment composition the same.

However, if even a single vesicle with the exact same composition does not fuse to the same target, then all its V-, A-Snare combinations with T-snare from this non-target composition become invalid. But these combinations are exactly the same as formed by another vesicle(s) with the exact same composition, which is fusing to this non-target compartment. This implies that all the combinations have now become invalid, and assumption has been contradicted. □

Lemma 4.1.2. *[A sub-set of cargo types fusing in a separate vesicle(s) from a bigger vesicle cannot have valid VATs unless the bigger vesicle has the same fusions to the compartments they are fusing to.]*

Theorem 4.1.3. *VATs with Class-1 cargo type in both V- or A-Snare and T-snare position (21 VATs) are never forbidden.*

Proof. We know that Class-1 cargo types only have 1 coat interacting, which can carry them on a vesicle pathway, and therefore for them to exist as either V- or A- snare, they must be cyclically flowing using the same coat. This implies either a big multi-step cycle exists over which they flow, or they are being cycles between pair(s) of the compartment(s). The latter turns out to be the case as there are no cycles that exist on more than two compartments without having a route back, i.e., only pair-wise cyclical flow exists. Therefore whenever these Class-1 cargo types exist as V-snares, we know the same coat exists in a cycle between pair(s) of compartment(s), and these will be carrying the exact same cargo types to and fro, so by Theorem 4.1.1 they must have self-loops to their sources, making a closed cycle. Therefore, either these Class-1 V/A- and T-snare form and are valid and not part of the forbidden zone, or they never form, and VATs that do not form by virtue of existing on a flow (not depletion) are not part of the forbidden zone. □

Lemma 4.1.4. *Class-1 cargo types only exist on the closed cyclical flow made by the respective coats they interact with singularly.*

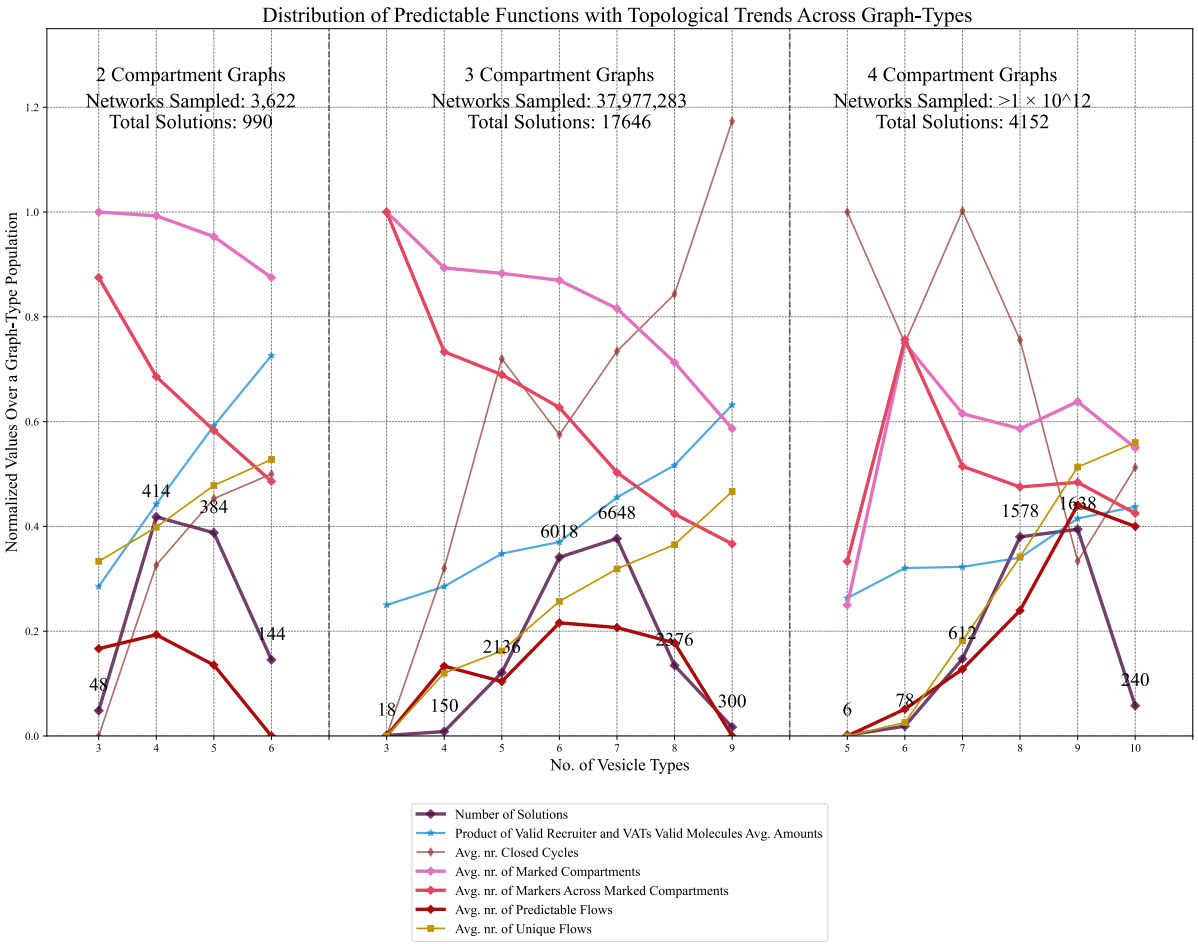


Figure 4.2: **Distribution of Graph Properties in Background of Regulatory Landscape of Valid and Invalid Molecules, Across Graph Sizes** The unit of Vesicle types (source compartment, coat-label) as a measure of how many unique flows exist in Graph, and the variation in important properties such as number of Closed Cycles, Marked Compartments and Flows with Unique targets(Unique Flows). The rise in Closed Cycles is mostly steady with the rise in vesicle types because the number of coats is limited. If more vesicles were to be added at multiple compartments, then cycles are bound to form, and we know that these cycles need to have a self-loop at each compartment by Theorem 4.1.1.

Theorem 4.1.5. ***The V-Snare Activator Model Only 296 Molecules exist in the real System in both V-snare Activator and V-Snare Inhibitor Model***

Proof. The number of cargo types that each of the Coat-types can carry is only 4 for the V-Snare, and again at least 3 of these cargo types are the same between two Coat-types' affine cargo types and cargo type 8 same across all Coats.
Whereas the number of 'Complement cargo types' associated with each coat, in terms of never flowing them in itself, is also 4, but again, certain Complement cargo types also occur twice (Cargo-type 2 and 3) and some thrice (Cargo-type 1) across coats. Thus the actual number of VATs is lower than 512 in all possible combinations. A formula to calculate the space of all possible VATs is -

$$(3\times(4\times4\times8))-((3\times1\times8)+(2\times(1\times1\times8))+(3\times((1\times1\times8)+(1\times1\times8)))) = 296$$

Or, $384 - (24 + (16 + 48)) = 296$
Or, $384 - 88 = 296$
Breakdown -

$$3 \times (4 \times 4 \times 8) = 384$$

3 coats, each carrying at most 4 cargo, with at most 8 cargo types at the target compartment.

Redundant Permutations among these 384 Tuples -

- The 3 cargo types that have 2 different coat interactions occur more than once, and each of the 3 cargo types occurs at least twice while counting for each Coat's 128 permutations. The combination with which each of them exists inside one Coat-type is unique, i.e., each Coat has a unique combination of such Class-II cargo types. Thus we subtract one for each

of these existing with themselves in the VAT, $3 \times 1 \times 8$.

- Similarly, the Cargo-type 8 interacts with all three Coats, and its VAT permutations with itself will occur thrice, so we only keep one of them by subtracting the other two times - $2 \times (1 \times 1 \times 8)$.

- Cargo-type 8 will also co-occur with the 3 Class-II cargo types, which occur twice across the Coats. Cargo-type can exist in either position 1 or position 2 with each of the Cargo-types - 4, 6, and 7. Thus, we subtract redundant ones - $3 \times 2 \times (1 \times 1 \times 8)$.

□

Theorem 4.1.6. ***The V-Snare Inhibitor Model Only 296 Molecules exist in the real System in both V-snare Activator and V-Snare Inhibitor Model***

Proof. The number of Cargo-types that each of the Coat-types can carry is only 4 for the V-Snare, and again at least 3 of these cargo types are the same between two Coat-types' affine Cargo-types and cargo type 8 same across all Coats.
Again, the number of 'Complement cargo types' associated with each coat is also 4, but again certain Complement cargo types also occur twice (Cargo-type 2 and 3) and some thrice (Cargo-type 1). Thus, the number of VATs is lower than 512 in all possible combinations. A formula to calculate the space of all possible VATs is -

$$(3\times(4\times4\times8))-((3\times(2\times(1\times1\times8)))+(2\times(1\times1\times8))+(3\times(1\times1\times8))) = 296$$

Or, $384 - (48 + (16 + 24)) = 296$
Or, $384 - 88 = 296$
Breakdown

$$3 \times (4 \times 4 \times 8) = 384$$

3 coats, each carrying at most 4 cargo, with at most 8 cargo types at the target compartment.

Redundant Permutations among these 384 Tuples -

- The 3 Cargo-types occurring twice across all Coat affinities always occur with 1 at the A-Snare position, as well with the same Class-I cargo types in the A-Snare position when Complement Cargo-types are considered. Thus we subtract one for both of these redundancies, $2 \times (1 \times 1 \times 8)$.
- For the Cargo-type 8, which interacts with all three Coats, will always have Cargo-type 1 at the A-Snare position in all the 3 coat-cases and, therefore will be unique only once. Thus we subtract from the 384 - $2 \times (1 \times 1 \times 8)$.
- Cargo-type 8 will also occur with 3 Class-I Cargo-types - 2,3,5 - in the A-Snare position, which occur twice across the Complement-Cargo-types list associated with each of the Coats. Thus we subtract redundant ones - $3 \times (1 \times 1 \times 8)$.

□

4.2 Evolution of Vesicle Trafficking in Early Eukaryotes

As discussed before, there is conclusive evidence to show that the first eukaryotic ancestor belongs to the Asgard Archaeal Superphylum (Zaremba-Niedzwiedzka *et al.*, 2017; Field and Dacks, 2009b; Koumandou *et al.*, 2013). And that duplication and co-option or creation of paralogous protein families within species (Thattai, 2023; Gabaldón *et al.*, 2006; More *et al.*, 2020), and orthologous protein families across species (Gabaldón and Koonin, 2013) is at the heart of the creation of different vesicle pathways. Therefore to invoke modification in complexity, we needed to allow molecular changes to create new vesicle pathways or retarget them. This system of checking and allowing transitions between different graph topologies with few molecular mutations was created by Sahana and Mukund (2022, Unpublished). However, to invoke Evolution, we needed to define functions in the solution graphs and a Selection model that acts on these functions. This work was done jointly by Sahana, Mukund, and me.

4.2.1 Defining Function

This is a subsection to Sec. 4.2.

Our assumption is that for sub-cellular machinery to be useful to the cell, it must have an identity that separates or marks it in the cell and allows it to function in a specialized way.

We have in our system, two major morphological features that must have identity in each graph, which we call the cell in our model.

Identifying Compartments

This is achieved by defining the identity of compartments across graphs based on the existence of same Marker cargotype, i.e., compartments with identical markers in different graphs are the same.

Identifying Flows

Similarly, a unique flow or pathway is defined by the flow of a vesicle-type strictly between two marked compartments, i.e., a single target.

These two coordinates of identifiable compartments and identifiable targets guide the path evolution takes.

4.2.2 Proofs in Predictable Functions

This is a subsection to Sec. 4.2. The nature of topologies selected for certain number of functions follows a specific pattern which can be reasoned through the nature of emergent molecular landscape creation which validates or invalidates a topology.

4.2.3 Defining the Ratchet Model of Selection: Constraining Transitions

This is a subsection to Sec. 4.2. Upon the two functions of identity and unique flows, the model of selection, a transition to a different graph is allowed if and only if-
(1) Transition in the molecular landscape is allowed and,
(2) markers of the compartments must be preserved on compartments of the other graph, and
(3) if there is a unique flow between these marked compartments, then it must also be preserved with at least one cargo type on both source and target compartments.
The next graph is allowed to have more functions than the first one, i.e., more markers or flows. This selection model is called the Ratchet model of selection, as it goes on ratcheting the functions across graphs.

Results from Markov Chain Analysis on Directed Graph of Directed Graphs

This is a subsection to Sec. 4.2.3. The trajectories we get by deploying such a selection over solution set graphs with defined functions, we get trajectories as shown in Figure 4.3. We call each discrete point in this coor-

dinate space made by the number of functions an Evolutionary Layer. We can observe that graphs at a certain layer are connected only to either the layer on their right or above, both of which imply an increase in function in number of marked compartments and predictable flows, respectively.

Markovian analysis based on Directed Graph probabilities implemented by Sahana over all possible transitions allowed by both molecular and functional selection yielded the most probable evolutionary trajectory ending in a graph that has a lot of markers and flows and, according to labeling, seems to represent the current day eukaryote with endocytosis and secretion like functions.

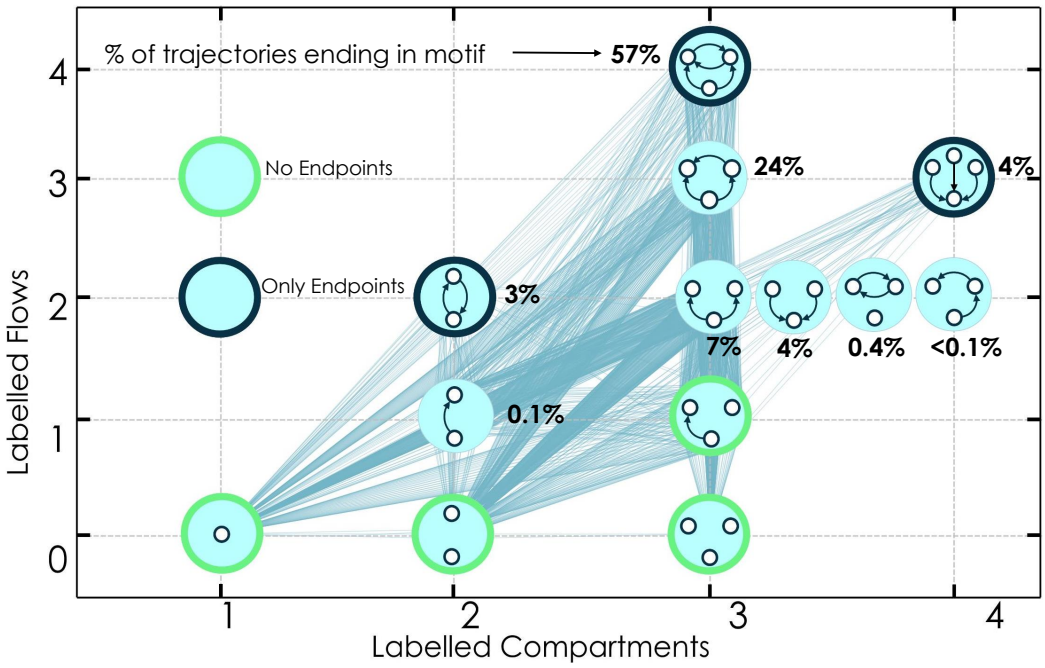


Figure 4.3: **Evolutionary Trajectories** Markovian Trajectories on a Weighted Directed Graph of Solution Directed Graphs

The 2 Motifs That allow a Vesicle Path to be Predictable

A predictable path exists in the following two cases, as shown in figure 4.4-

•

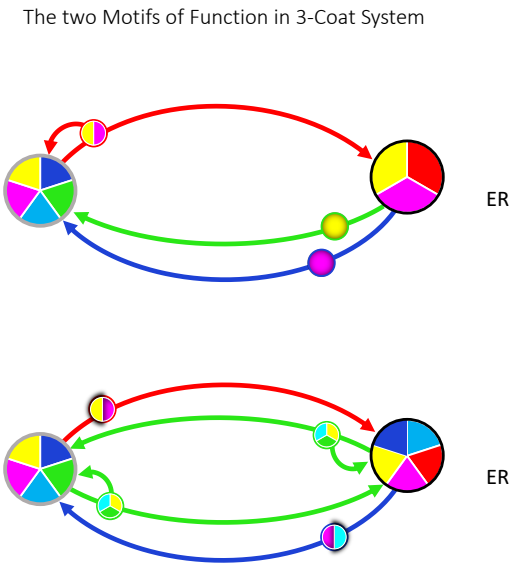


Figure 4.4: **Predictable Flow Motifs of 3-Coat Model** The only possible topologies which allow a predictable flow function(s).

Theorem 4.2.1. ***Motif-1*** *At most two different Coat Vesicle pathways between source and target can be predictable if and only if there is a half-closed cycle routed at the target and flowing back to their source with the third label.*

Proof. It can be proved very simply why the half-closed cycle must have the third coat different from the first two coat vesicles- if the half-closed cycle had the same coat as the any one of them then both that vesicle and vesicle on the half-closed cycle will have the exact same composition and by Theorem 4.1.1 they will need to have the same coat and therefore a closed cycle would form. One might wonder why the other predictable vesicle also loses its function, if one more vesicle exists, and that is because the cargo-types being carried by it are a subset of what is being returned to the source as opposed to when novel Class-2 cargo was able to flow on it because a novel Class-2 was getting shipped, but now the cargo carried by it are also fusing back to the source and thus it no longer remains predictable. □

- **Theorem 4.2.2.** ***Motif-2*** *A cyclical predictable flow made by two different coats on the same pair of predictable compartments is possible if and only if the third coat forms a closed cycle on the same pair of compartments.*

  Proof. As shown in Theorem 4.2.1, a Coat labeled Vesicle is dependent on what cargo types are being returned by the return edges from its target back to its source for the cargo types composition on it at steady state.
  If suppose the return edge, which is not a part of the closed cyclical flow gets deleted, then we have the same motif as in Theorem 4.2.1 and none of pathways remain predictable anymore.
  The second requirement of a closed cycle existing is again based on the fact that every pair of the Coats shares a unique Class-2 molecules as shown in Figure 2.1, and if only two different coat vesicles exist cyclically on a pair of nodes, they will be transporting the same set of cargo types because they are dependent on only each for what gets cycled back to their compartment and again from Theorem 4.1.1. □

  Lemma 4.2.3. *A Class-1 Cargo type is never on a predictable path because they are only flowing strongly at a steady state when a closed cycle with their affine coat exists (Lemma 4.1.4).*

Why no individual Cargo type exists on a 2-step Predictable Flows

It is a very important result that comes out of the observations of what kind of topologies that exist in a valid graph are predictable, that the flow of a cargo can't be predictable on more than one source target compartment-pair flow.
This can be simply proved by the fact that none of the 2 motifs that allow any vesicle paths to be predictable cannot be propagated to flow any of cargo types that were previously predictably flowing in the motif through any topological re-arrangements.

Theorem 4.2.4. *No Cargo types have a 2-step Predictable Flow.*

Proof.

- **Motif-1**: The only predictable flows are formed by Class-2 cargo types on one vesicle or two vesicles of different coats flowing between the same source and target pair. It is also implied that the predictable cargo types are also on the backward half-closed routed at their targets because these cargo types are flowing (as invoked in Theorem 4.1.2. But a half-closed cycle invokes the presence if the self-loop on the source compartment, which in effect, demolishes the predictability of any cargo type flowing inside it. Therefore, the existing Class-2 cargo types which are predictable cannot be predictable propagated using any coat.

- **Motif-2**: The two opposite flowing different coat vesicles have the predictable cargo, and they have a unique Class-2 because of their interaction with the third Coat, which forms the closed cycle. Therefore, the closed cycle needs to be propagated to the next compartment. However, if it is indeed propagated, along with one or both edges in the same forward and backward arrangement or the reverse of that arrangement, we can always find that one of these new vesicles carries either that exact same or a subset of cargo types of the same coat vesicle which was pre-existing, which implies that it cannot exist without a self-loop by Lemma 4.1.2.

□

Remark. **Hint Towards Constraints on Maximum Functions**

It is evident that even though we have not found the upper bound on the maximum number of compartments that can form a valid graph, the number of unique predictable functions is bounded by 3 Unique Class-2 cargo types: 4,6,and 7. Given that the same cargo type can't be propagated onto any predictable second step flow, we find the only motif in the layer $(3, 4)$ as shown in the figure 4.3 is the only possible motif which comes from a valid graph, all the other versions of this motif with any other edge changes makes

the graph invalid to the model (please note that here if two different vesicles are budding from the same compartment and fusing to the same target predictably, then they are considered one predictable flow only). Therefore only a specific arrangement of the motifs can form the valid graph, and there will be limited permutations that this can take along with the limit on the number of uniquely marked compartments other than the ER is capped by 6 (Class-0 cargo types 1 and 8 are not counted).
We have not even included the constraint of compartments being labelled to the restrictions we have imposed in this subsection on predictable flows, and if we were to invoke them, we might be able to get at the limit of what could be the most complex network under this regulation model of V-snare activation.

Realizing this restriction of motifs of functionality missing under this regulation model, Sahana and Mukund (2024, Unpublished) are also looking at other regulation models to expand the span of what this kind of boolean modeling can achieve in terms of complexity and also allow us to study where molecular regulation was a constraint and a change in regulation opened pathways to the current day eukaryotes, or led to the extinction of the networks. We obviously also see an increase in the number of functions by increasing the number of coats(and hence cargo types) in the system as we see the 3-step predictable of cargotype-12 in 4-Coat system as shown in figure 4.5.

Remark. **The best Definition of Function**

The trajectories we have achieved capture some evolutionary concepts, such as the paths taken to the creation or internalization of a new compartment (which we will come back to at the end) through a process of preservation of the functions gained by the graph and gaining new functions while allowing the loss of unpredictable events if required to go higher in the functional layer. We also observe retargeting of pathways taking place in the case of

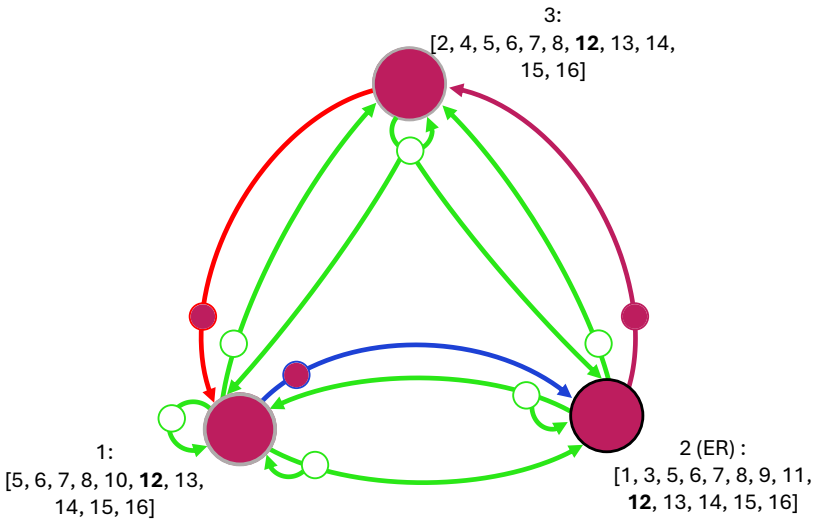


Figure 4.5: **3-Step Predictable Flow Motifs of 4-Coat Model** 2-Step and even 3-step Predictable flow, as shown in the figure with the focus on cargo-type-12 shown in purple.

specialization of a depletion flow from two to one compartment, with the flow of material also becoming predictable if present on the correct motif arrangement, also called 'Neo-functionalization' in this case or in case of loss of recycling route. 'Sub-functionalization' happens in the case of the complete splitting of markers from one compartment to another, which can brought about by one-step or multi-step depletion.

However, we see there are stronger constraints on what kind of molecules are involved in these functions, for instance, the cargo type of Class-1, which is never accepted to be predictable in flow function and, therefore, cannot be retargeted ever. This gives insight into the fact that under this definition of function being unique fusions, only those cargo types that have different recycling routes (Class-2 cargo are affine to 2 coats) can be functional. Even though this is a strong result, it is not in alignment with the fact that most targeting signals are only a few residue stretches, and therefore, one mutation could potentially lead to a change in targeting. However, we never see such a one mutation switch of cargo flows, which is allowed at the molecular regulation, taking place with the same or different markers on targets

taking place because of the definition being used. To keep a Class-1 cargo out of functional bounds is strange as it could represent a more specialized state when cargo types have been selected for a specific recycling route only. This, however, is taken care of in the preservation of markers where the replenishment flow specialization is realized in terms of gain in a marked compartment, or 'neo' or 'sub-functionalization.'

4.3 Topological Transitions: An Attempt in Studying Viral Hijack

Viruses are known to use endocytic vesicles, which leads them into the Vesicle traffic network unobstructed from the crowded cytoplasmic components (29), and have co-evolved machinery to use endocytic maturation (31). While doing so Viruses rewire the retrograde vesicle traffic paths to retarget the required molecules in a specific organelle where they need to replicate and assemble while also evading the host immune system, which is usually their self-created Replication Organelle near the ER(29,31), similarly they hijack the secretory pathway to egress from the cell (29). Therefore, we can define this rewiring by looking at graphs that have small changes in topology like an extra edge or a different edge, or a new compartment altogether with the rest of the smaller graph preserved inside the larger graph.
Let's redefine what we mean by Topology in these transitions, as a specific definition was used-

Topology is the arrangement of specific vesicle types and their various targets. No self-fusions were considered while defining the topology of a graph.

Several analyses were done to realize the nature of these transitions. The distribution of the minimum no. of components that will be completely forbidden with respect to the other graph came out to be mostly 1 or 2 forbidden components between them, suggesting that the rewiring is molecularly loosely constrained.
Distribution of forbidden components varies w.r.t the two kinds of topological changes-(1) creating an extra edge or (2) rewiring source and/or target of one edge in the graph pair.
The creation of an additional edge creates fewer conflicts with the existing molecular cover than a change of an edge (Figure **??**(b)).

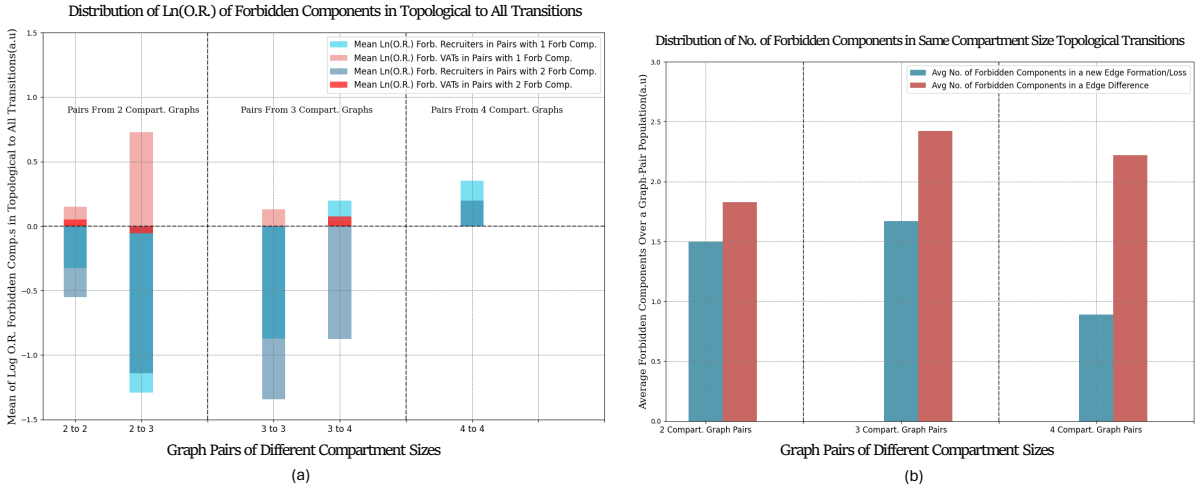


Figure 4.6: **Molecular Trends in Topological Transitions**
(a) Distribution of Mean of Objective Natural Log of O.R. of Topological to All Possible Graph-Pairs. VATs across Graph Pairs are more enriched than recruiters except in pairs with 4-compartment graphs. 4 to 4 4-compartment pairs do not even have any VATs forbidden between them, consistent with the large drop in a number of valid Recruiters as compared to drop VATs for 4-compartment graphs w.r.t. smaller size graphs. Individual 4-compartment graphs can have very different recruiters amongst each other. (b)) Distribution of forbidden components varies w.r.t the two kinds of topological changes. The creation of an additional edge creates fewer conflicts with the existing molecular cover than a change of an edge

4.3.1 Nature of Conflicting Molecular Space Between Graph Pairs

This is a subsection to Sec. 4.3.

If we observe carefully in Figure 4.7(a-b) that shows the heatmap of molecules that are forbidden together in these topological transitions among the 296 VATs and 24 recruiters, all the molecules that exist in our solution set at some point, there is still a chunk of them that never found forbidden in these transitions as depicted in the empty 0 space of these heatmaps.
This can be explained by the following previous observations: -
We know by Theorem 4.1.3 that the 21 VATs with Class-1 cargo types - 2,3,5 never exist in the forbidden zone.
VATs with Class-1 cargo type at V and/or A positions(152 molecules) are never part of the overlap of the selected(one mutation or no mutation) ir-

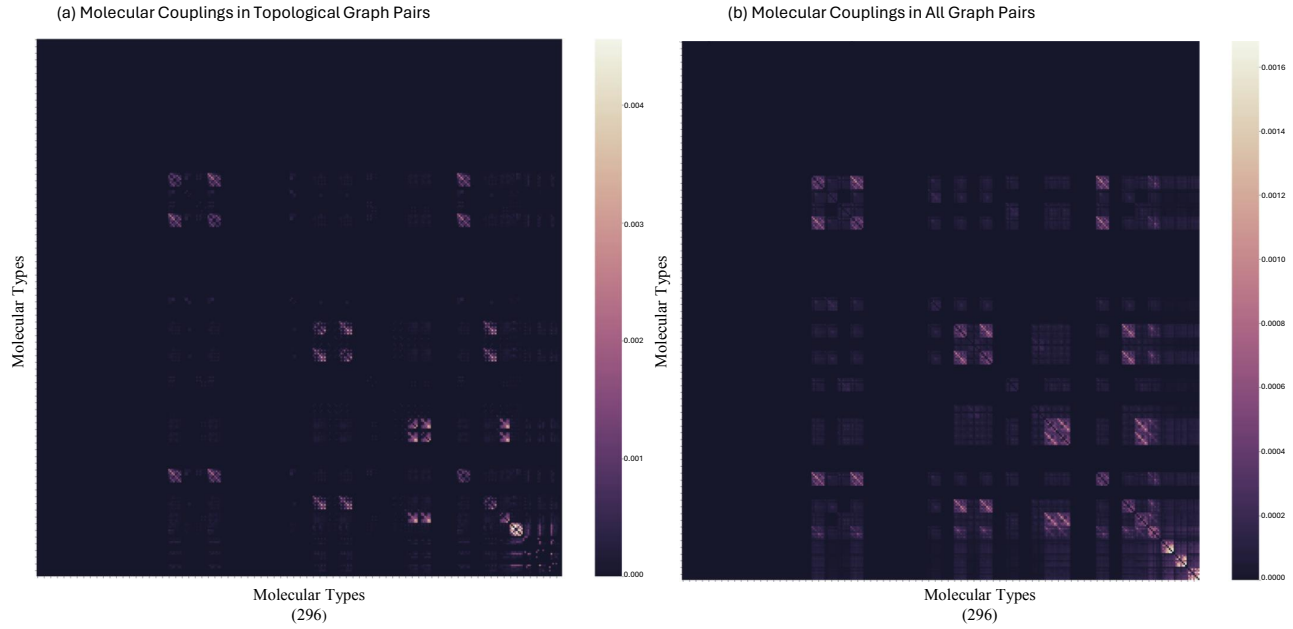


Figure 4.7: **Molecular Coupling in Forbidden Components in Transitions** (a) Molecules that co-occur in the Topological Transitions among the total 296 molecules. (b) Molecules that co-occur in the All possible transitions a graph can have, among a total 296 molecules.

reducible cover of one graph with another cover's forbidden zone. This is because the existence of Class-1 at the V/A position suggests that Class-1 cargo type is flowing, which is only the case when it is a closed cycle (or a self-loop). Remember when I talked about certain VATs (21) that are never found in forbidden zones in any graph (both V and/or A and T - snares are the same Class-1); so there's no chance these VATs will not exist covering the same vesicle fusions, we can prove this by contradiction if they did not exist, then they are not flowing. These are never forbidden in graphs that do not have Class-1 flows as well. So when comparing irreducible covers from two graphs, an irreducible cover will always exist that has these 'always valid' molecules. That component will never be completely inside the forbidden zone of the other graph, so this cover is the one with minimum distance from the other graph as we create all possible combinations of components.

This leaves us with an important biological insight: the molecules that have

a smaller interaction network allow transitions more easily by serving as the raw material because few or no other molecules in the molecular network are getting perturbed.

To realize the nature of these molecular sets which irreducibly cover a graph, and upon which evolution has been defined to be acting we looked at various approaches, including Decision Trees and data compression like Huffman codes. However, these methods didn't yield the results required. Thus, we looked at a more innate thing in our model, which is pervasive in nature as well: Symmetries, and it has allowed us to look deeper into the features that control this Model. In the end, we look at how based on the study of these relations of symmetries of coat-cargo interactions 3.5 gives an idea of how we can go from molecules to graphs.

4.4 Symmetries, Creation - Compression and Proofs in Model Solutions

4.4.1 All Symmetric Coat-Labellings are Solution

This is a subsection to Sec. 4.4.

Theorem 4.4.1. *All Symmetric Coat-Labellings are Solution*

Proof. If a specific coat-labeling is a solution to a topology, it creates a specific distribution Class-0,1 and two cargo types distribution based on its affinities and the connectedness of topology in the Graph. A specific composition of any compartment is constituted by the coat-labeled edge flowing towards and out of it. The composition in the vesicle is constituted by the coat-labeled edge(s) parallel(between the same source-target node pair) to itself and the edge(s) recycled back to the source.

From the definition of Isomorphism as defined in **??** and the definition of Symmetry 3.5, we know that when any coat-symmetry operation applied to the labels will yield a clear one-to-one bijection of mappings between the original labeling and the symmetric labeling for every edge on the same topology. This implies that no Strongly connected Cycles or number of out or in-edges to any nodes is lost; only coats are switched amongst themselves. If this was not the case, we can show that the labeling created is not Isomorphic to the initial labeling.

We define the loss of cyclic routes and a certain degree of out-edges or any specific topology motif based on a specific coat. So, as long as these motifs are preserved in any other or its own coat's labeled edges set, and if the same holds true for all the coats, then it implies that the source-target pairs along with edges between them remain the same but with a different coat label for each previous labels. However, there is a symmetry at the level of coat-cargo interactions across coats; each pair of coats will have exactly

the same cardinality of cargo types that overlap or those that do not. This implies that the distribution of cargo on a compartment will be switched to cargo types that exist at the steady state after the result of the SCC it is a part of with cargo being taken in and out with a certain arrangement of coats on the edges. We can prove that if a compartment has a certain arrangement of a pair or a triplet of edges coming going in and out, it will be the same as well as pair or triplet of coats will be preserved according to our definition of Isomorphism. Thus, because the pairs of coats are conserved, and because each coat has identical cardinality of cargo types affine to it and overlapping with the other two, we will end up with the same cardinality of compositions as the topology motifs formed by each coat is either preserved or transferred to another coat. However, through the definition of symmetry operations, a bijection is created for each cargo type as well. Thus the new compositions on the compartments can be found just by applying symmetry operation to the cargo types. The same is evident for vesicle compositions as the pair of labeled edges undergo a transformation preserving the number labels and, therefore, the overlap in cargo type cardinalities, except with cargo types that are bijective to the previous (original) cargo type set.

We showed in the previous chapter the triangular planes upon which cargo can transform into any other vertex preserves the Class (number of affinities to different coats) of the cargo type 3.5. Thus, as long as the distribution of cargo types remains the same, we know that multiplicity in the classes of cargo types is also the same. Therefore any combinations made under a certain class to a certain other class is either completely preserved or preserved in terms of cardinality. This implies that the number of valid VATs we can get will remain the same over a pair of edges that are isomorphic to each other as a result of bijection. Similarly, for recruiters, the distribution of the number of unique cargo types per class remains the same, and the unique number of coats budding from them remains constant. Therefore, the number of recuiters remains the same but may get shifted around each

coat based on the symmetry operation.

□

Lemma 4.4.2. *All Symmetric Coat-Labellings have Symmetric Covers*

Creation of Complete Solution Set

I realized that the coat-labelled solution set that is generated from the Model's Algorithm prevents creation of redundant different orderings of the same labellings for edges with multiplicity more than 1. However, this was only only done by Sahana after 2 Compartment solutions were done, and 3 compartment solution search needed to be optimized. So the number of actual solutions for 2 compartments instead decreased after recreating all symmetric versions of coats.
The algorithm for the creation of all symmetric labelings is straightforward and does not require the recreation of compositions and molecular covers because the only labelings missing were redundant re-orderings of existing ones to which they needed to be mapped.
The creation of all symmetric labelings was important to check topological properties, which we will encounter ahead. Creation of all symmetric labellings lef to the following change in the numbers in the solutions set - 2 Compartment Solutions decreased from 1,164 to 990.
3 Compartment Solutions increased from 10,224 to 17,646.
4 Compartment Solutions increased from 882 to 4,152.

4.4.2 Symmetry in Molecular Landscape and Compression of Solutions

To realize the nature of the regulatory landscape, we needed to remove redundancy to look at true features, and therefore this experiment of Compression regulating molecules was carried out, and it resulted in specific symmetry patterns in different kinds of regulatory molecules that we have - VATs and Recruiters. In effect, these led to the clustering of graphs into groups where

their whole cover can be related by VAT and recruiter molecular symmetries under the same operation of the Coat-Rotation-Reflection Symmetry Cube. And it allowed us to realize the mapping from the molecular description to graphs and its properties.

Symmetry in VATs

The symmetries in the cargo type triplets that are the VAT are visualized in the Figure 4.8

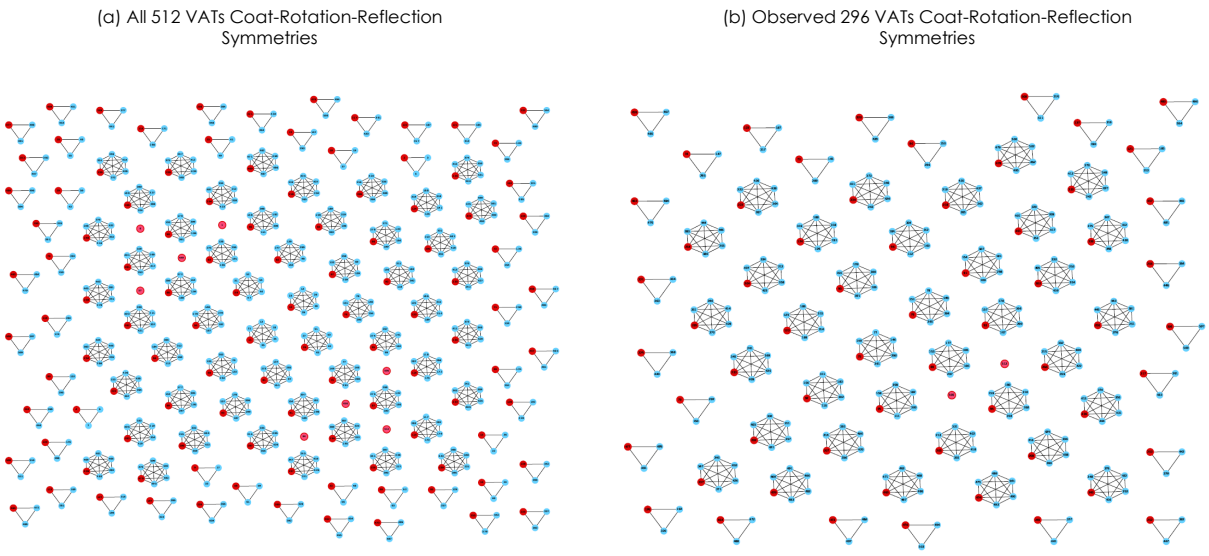


Figure 4.8: **(a) All 512 VAT Coat-Symmetry Relations** There are 56 6-Cliques and 5 3-Cliques, along with 8 isolated VATs with different permutations of 8s and 1s. **(b) Observed 296 VATs Coat-Symmetry Relations** VAT Coat-Symmetry Relations for 296 VATs that exist in Solution set

Theorem 4.4.3. *Only 56 3-Cliques exist under the Coat-Rotation-Reflection Symmetry Cube.*

Proof. 3-Cliques exist via the Symmetry operations under Coat-Rotation-Reflection Symmetry Group if and only if minimum symmetry in Coat Columns is equal 2 or uniqueness of bits among the 3 bits is equal to 1 in the cargo types in V-, A-,T- Snare positions. This confers 3 permutation of the unique bit which we know is related by both reflection and rotation

as well as identity symmetry operations.
There 3 groups into which cargo types of into which cargo types with only 1 unique bit can be classified based on which position among 3 bits does it exist in one of these Position Classes-
Class-A: 2 (001) , 7 (110)
Class-B: 3 (010) , 6 (101)
Class-C: 5 (100) , 4 (011)
Note that Cargo types 1 (000) and 8 (111) vacuously belong to each of the above classes, but for simplicity let's say they belong to Class-A.
Now VATs in 3-Cliques belong to these Classes of distinct symmetries. So a VAT belongs to any of these Classes if all its elements(cargo types) belong to the class, e.g., (2,1,1) belong to Class-A because 2 and 1 belong to Class A.
However, VATs from one of the Position Class can either reach themselves via identity symmetry which is trivial, or become a part of other Position Classes through some symmetric reflection and rotation operation. E.g., (2,1,1) connects to (5,1,1) and (3,1,1).
Each non-redundant non-trivial permutation changes the position of the unique bit among the 3 Position Classes and therefore is bound to lead to a VAT that belongs to a different Class than itself.
Therefore, in each 3-Cliques one and one VAT from each of the 3-Classes exists.
Thus, we can count the number of cliques just by finding the total number of VATs that will fall into one of the Position Classes.
For example, for Class-A the cargo types that are $\{1, 2, 7, 8\}$
all possible permutations of length 3 (all VATs) from this set $= 4^3$
but we are also counting the VATs with only 1s and 8s which do not belong to any of the Position Classes,
Therefore Total Class-A VATs $= 4^3 - 2^3 = 56$ □

Lemma 4.4.4. *Only 56 Cliques of 6-Cliques exist under the Coat-Rotation-Reflection Symmetry Cube.*

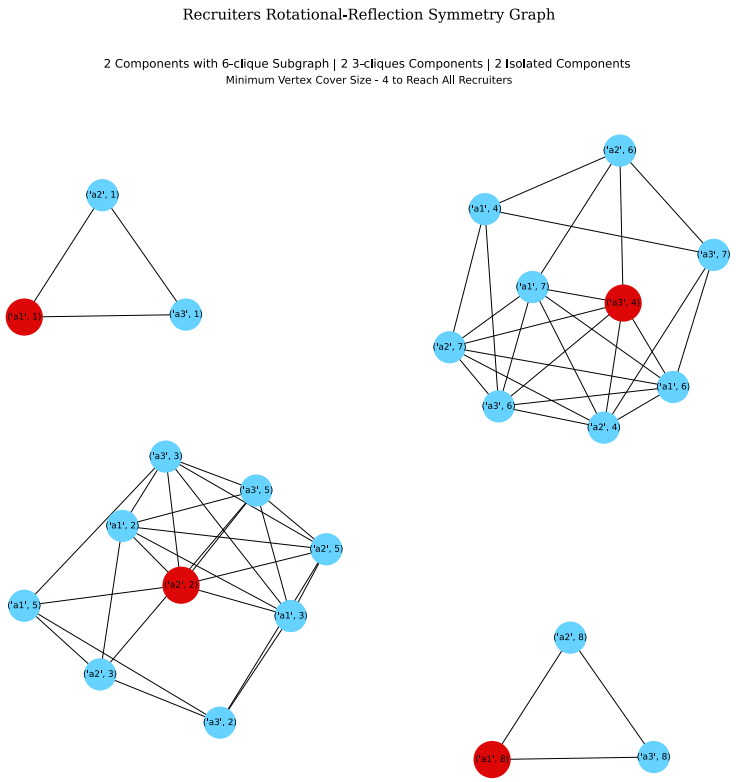


Figure 4.9: **Recruiters Coat-Symmetry Relations** Graph Topology Trends Across Increasing Graph Size.

Symmetry in Coat-Recruiters

This is a subsection to Sec. 4.4.

Figure 4.9. We observe that there are, in total, 24 Recruiters all which appear in the molecular description of our solution set. We can see the symmetry operations do not result in 2 complete clique components of Class-1 and Class-2 cargo types, respectively. This is because coat-cargo types pairs in the triangle in each of the 9 component subgraphs relate to each other via Reflection symmetry operation but to only 2 coat-cargo types in the 6-clique layer below them via Rotation symmetry. On the other hand, coat-cargo types in the 6-clique layer relate to each other through reflection or rotation.

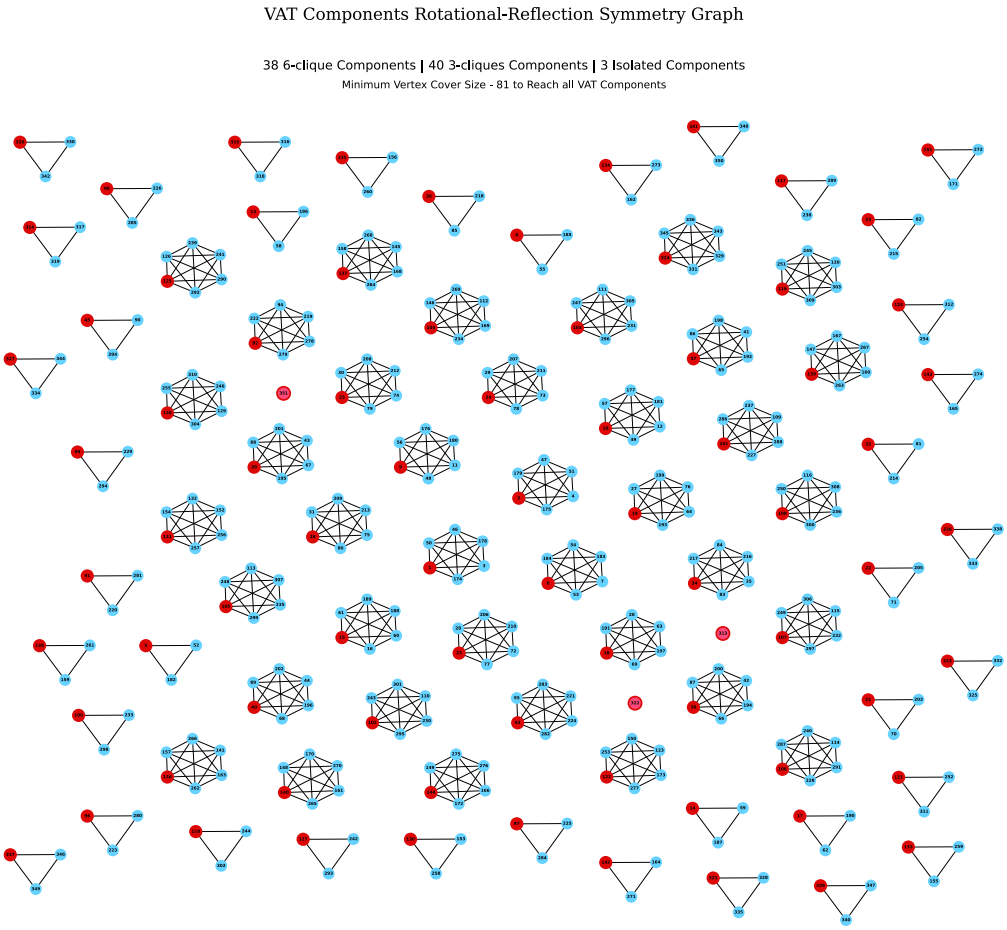


Figure 4.10: **The 351 VAT-Components Coat-Symmetry Relations** Graph Topology Trends Across Increasing Graph Size.

Symmetry in Components

Figure 4.10. We observe that there are, in total, 351 VAT components from all irreducible in our solution set. Only 3 of these are Isolated Components, 2 of which involve cargo types only 8 and 1 among the V-,A-,T-Snares, the third component is $((8,8,2),(8,8,3),(8,8,4),(8,8,5),(8,8,6),(8,8,7),(8,8,8))$ but none of its symmetric partners exists in solution. It remains to be proved if it can or can not have a partner if higher graph size solutions are discovered.

Symmetry in Covers

Symmetry in Covers has been discussed in great detail in the sub-section 4.5 of Molecules to Graphs.

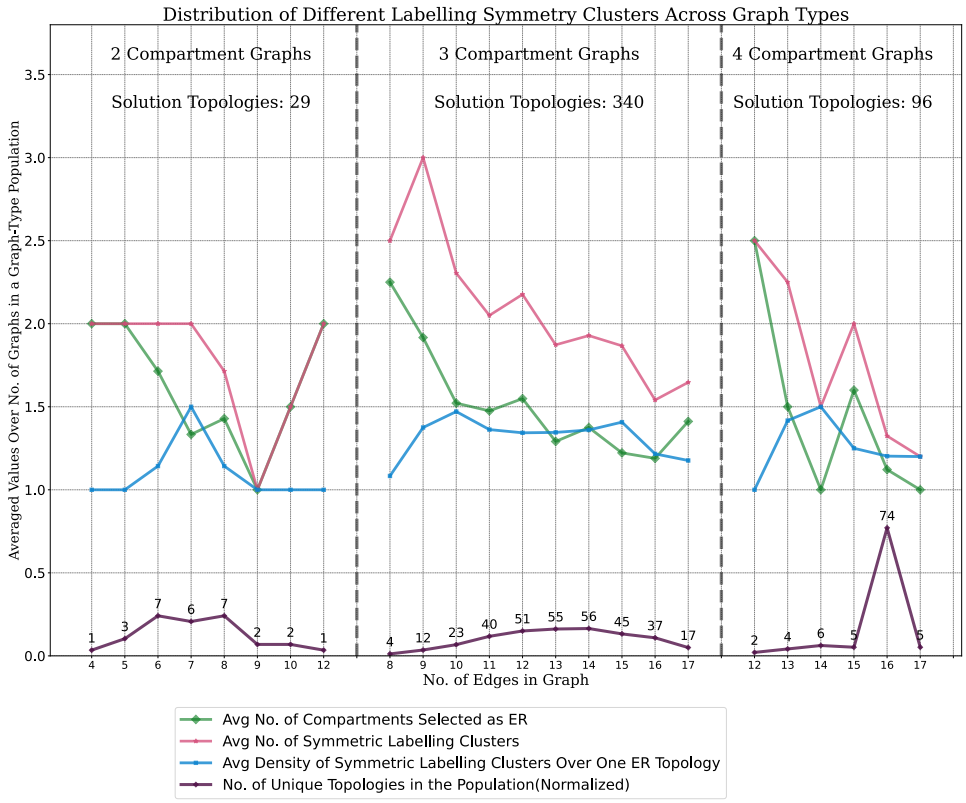


Figure 4.11: Distribution of Symmetric labellings Clusters Across Increasing Graph Size.

4.4.3 Distribution of Symmetric Clusters In Graph Solution Topologies

This is a sub-section to Sec. 4.4.

To see how many Clusters of graphs, by the symmetry operation on Coat labelings for graph, can form under the same nodes and edges topology, we looked at Clusters of symmetric labelings under a topology using the following definition of topological symmetry-

Two graphs belong to the same coat-labeling symmetry cluster if and only if they are topologically isomorphic by the superimposition of their ERs, and there is a one-to-one mapping, an injective function which corresponds to a coat-labeling symmetry operation between edges related by Isomorphism, i.e., their source and target nodes have an isomorphic mapping.

If we try compressing the 22788 labeled-topology solution Graphs by this Topology Symmetry Clustering definition, we get 862 Clusters of topology graphs that are directly or Isomorphically symmetric to each other. We can

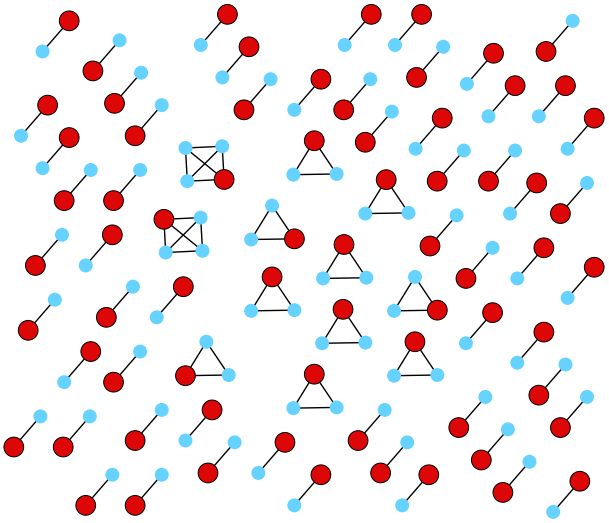


Figure 4.12: **Topological Coat-Labelling Cluster Covers Rotational-Reflection Symmetry Graph**. The isolated 679 Covers (nodes in the graph) are not shown in this graph. Other than isolated these, we observe 2 4-clique Covers, 10 3-clique Covers, and 58 2-clique Covers among the total 833 Cover Clusters, which capture different isomorphically symmetric graphs, some of them encapsulating more than one such topologically symmetric cluster. We only need 749 out of 833 of these Covers to create each and every solution graph in each topological symmetry cluster.

observe this in the Figure 4.11 that number of unique disjoint clusters of Now if we were to look at the molecules that form these graphs, we find that there are actually just 833 Covers, i.e., some graphs that are not topologically symmetric can still be the same at the molecular description level.

This is not surprising as in the next sub-section 4.5, we will discuss that a Cover or a genome can have multiple graphs that can be created by it but which are quite different in the number of edges and the number of compartments.

If we try to Cluster further these 833 Clusters of Covers, we get 749 Clusters, which implies even more redundancy occurs at the molecular level, and that very different non-isomorphic graph solutions are possible from the same molecular description, which we will describe in greater detail in the section 4.5.1 in Cover-Definition-1.

4.5 From Molecules to Graphs

During the search for how we could generate a clause statement for a set of labelings to decide whether it is a valid labeling/graph, I came across many rules that uncovered the existing phenomenons we have encountered previously, like why no 2–step path exists, why some VATs are never forbidden, etc. The same search, along with the discovery of symmetric relations between the covers of different graphs, also led to the realization of the relations of topology with the molecules and revealed the possibility of creating a graph or some possible graph from the same cover, as we will see in this section below. I tried to realize the more specialized description of the 'Cover' a graph has, which will be referred to as its 'genome.' The purpose of irreducible covers is indeed helpful in formalizing transitions between graphs. However, that is not the only regulation set a graph has, and from an evolutionary perspective, it would be incorrect not to expand our scrutiny on the evolution of VTNs with the exact molecular set upon which a graph exists.

4.5.1 Choosing the Molecular Description: The 2 Definitions of Genome

This is a sub-section to Sec. 4.5.

There are two ways in which we define the 'Genome' of a graph-

- Consider all the components and the molecules in the components that minimally cover the graph across all irreducible covers, and leave out molecules that lay in the lowest layer of covering a certain number of functions, so molecules with the lowest multiplicity of function coverage(lowest edges to the second partition in the complete version of the graph shown in 2.3 in the molecular sets will be left because molecules with higher multiplicity exist to cover every function.

- Consider every valid molecule that emerges as a result of fusions and budding events, not considering whether they generate a minimal cover or not.

Cover-2: Selecting Components that Make the Network

We define Cover-Definition-2 as - The set of all components that are selected in the irreducible cover definitions, which encompasses all recruiters but may not include all VATs that are non-neutral to the Solution graph. There are 4427 Covers by this definition. When we try to cluster these Covers via the definition of symmetry of Coat-rotations and reflections, we see 792 Symmetric Clusters.

The number of symmetric Cover Clusters and the Minimum Number of Covers to Reach All Graphs

We observe 689 6-Clique Covers, 95 3-Cliques Covers, and 1 Component which has two clusters of 6-Cliques that are connected pair-wise with each other out of the 792 clusters. The minimum required Covers to create or 'reach' all graphs in the solution set is equal to 787, as depicted in Figure 4.13.

The Covers under this definition rarely allow different Compartment-size graphs to exist on the same genome. No 2-compartment graph Covers share the same Cover with 3-compartment graphs. Only very few 4 Compartments graphs share Cover with 3 Compartments graphs in the Solution set. The reasons for this have not yet been discovered. But we have looked at whether the graphs with the same cover can exist in separate Evolutionary layers by virtue of having different functions.

Covers with Graphs in Different Evolutionary Layers

If we look at how graphs that have the same (or symmetric) Covers are distributed across the evolutionary layers, we get the plot as seen in Figure 4.14. We observe that a very low number of Covers span different layers. There is no Cover that spans more than two layers. Most of the Covers have graphs existing in the same layer, which is represented by the thickness of the nodes at each layer. Please note that this has not been normalized to

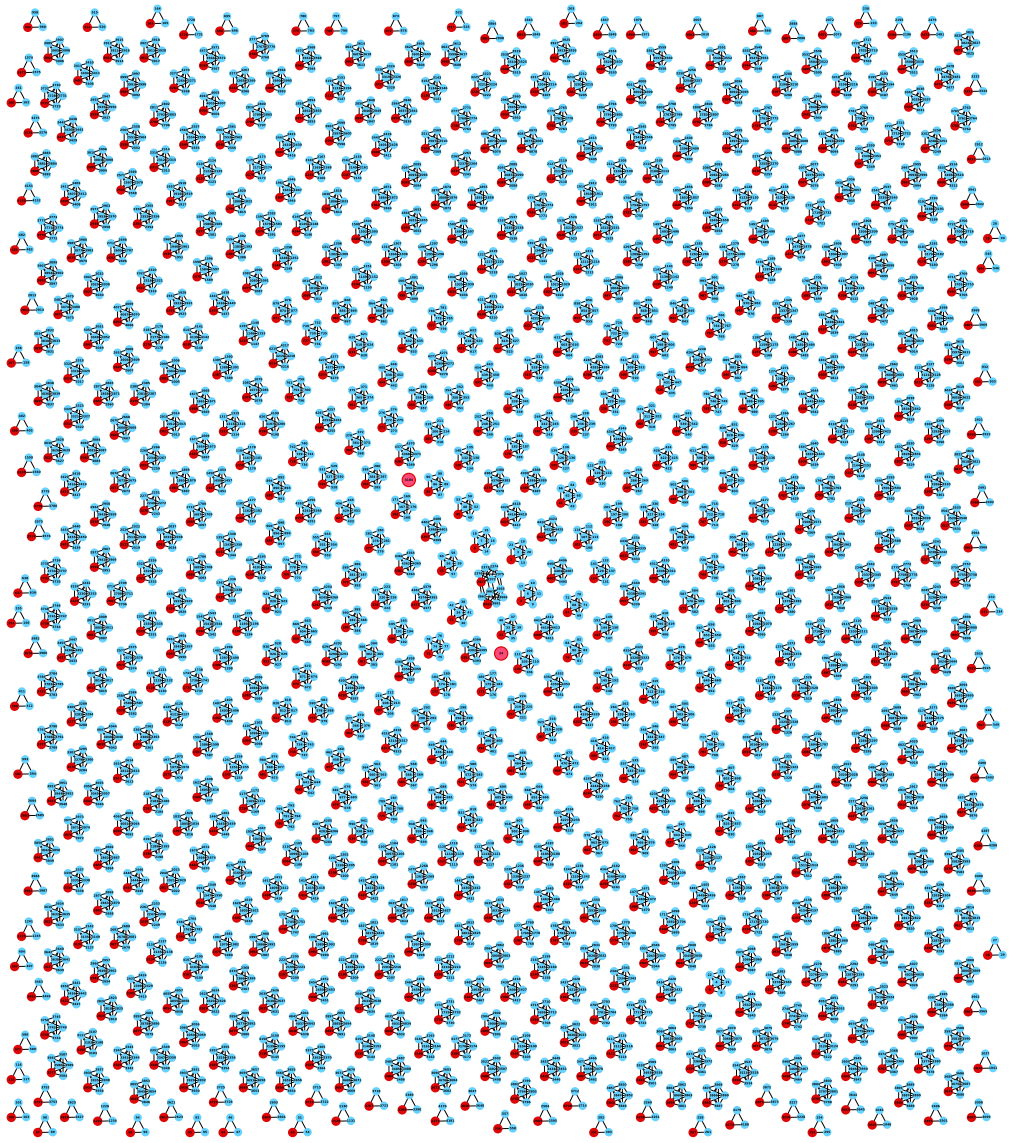


Figure 4.13: **Covers by Definition-2 Related by Symmetry.**

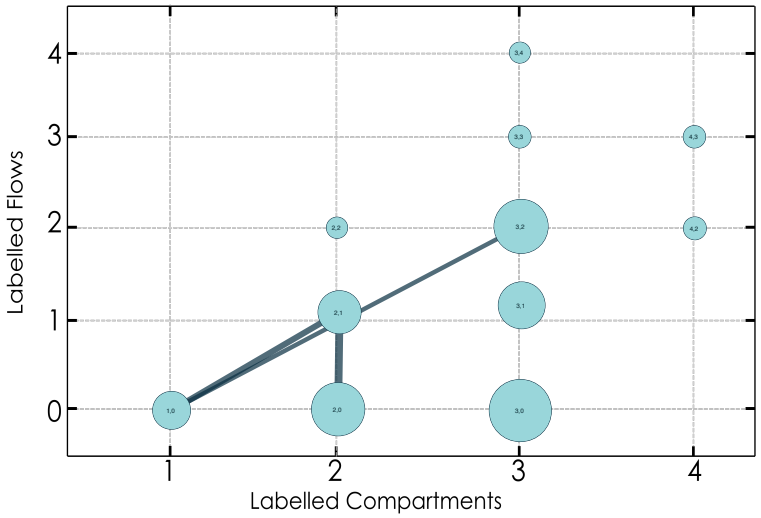


Figure 4.14: **Weighted Nodes and Edges Graph of Span of Evolutionary Layers by Covers by Cover-Definition-2.** Please note that this has not been normalized to the actual number of Covers in the specific layer.

the actual number of Covers in the specific layer.

Going Higher in Compartment Size under the Same Genome

What is an algorithm to generate a minimal vesicle network which will allow us to reach the graphs bistable to the Cover under the core definition?
If the symmetric version of a topology still allows for a graph to remain a solution, the heavier constraint is really in the modification of the topology which perturbs ether VAT or recruiter in the cover.
We also need to remember that a VAT is deemed invalid when the same set VA are present on vesicle types which have conflicting targets.

Input: Valid Cover containing all the VATs and Recruiters
Output: Compositions of Compartments from given Cover

- Create a Dictionary with a key as the identical cargo type existing at positions 1 and 2 in the tuple VAT, and value as the list of Invalid T-snares
- Perform Multiple Sequence Alignment (MSA) over the value list of the dictionary
- Cluster the T-snares invalid to cargo types based on the precise similarity of a conserved set of T-snares, and make it into a compartment composition
- Create a Dictionary with the key as the VA pair in the first two positions among the molecules given in the Cover, and the value as the list of T-snares for the VA pair
- Cluster T-snares as one compartment composition only if the exact same T-snares exist across certain keys of VAs

Algorithm 7: Graph Construction Algorithm

Data: A Valid Cover list containing the VATs and Recruiters and a List of Compositions of Compartments

Result: A graph which has operation covered by the given molecules in the Cover

Given : Valid Cover containing all the VATs and Recruiters and Compositions of Compartments.

Output: Graph representing operations covered by the given molecules

Put the VAs in vesicles with affine coats so there is no conflict in targets, and the compartment compositions are achievable through depletion and cyclic routes.

Flag=False **if** *Every VA in the cover forms a vesicle fusing to its T-snare containing compartment* **then**

    Flag=True;

else

    return 'Cover not Valid';

end

if *All coats have the same recruiters as given in Cover* **then**

    pass;

else

    Flag=False;

    return 'Cover not Valid';

end

if *Flag==True* **then**

    return Graph representing operations covered by the given molecules;

else

    return 'Cover not Valid';

end

As shown in an absolute algorithm above, it turned out to be possible, but the graphs created were not unique. A formal proof for this algorithm remains, but it was not followed through due to the apparent intractability to go higher in graph size with the same cover definition: -
Because the sub-graph may not be unique as the compartment size changes, along with its Class-1 cargo compositions. And even if we relax our constraint on fixed compartment size and exact same compositions, the problem of getting to a higher graph that does not perturb the existing components, which would incur a calculation of which flow is covered by which VATs and discovering a set of new flows or a new compartment (a new composition) requiring all possible combinations of Flows and the newly created VATs to exist in the same component.

Cover-1: Selecting All Non-Neutral Molecules that are Allowed in the Graph

We define Cover-Definition-2 as - The set of all non-neutral molecules, V-, A-, T- Snares, and Recruiters, which are formed and regulate the Solution graph, i.e., those that are non-neutral to the Solution graph.
There are 4226 Covers by this definition. When we try to cluster these Covers via the definition of symmetry of Coat-rotations and reflections, we see 777 Symmetric Clusters.

The number of symmetric Cover Clusters and the Minimum Number of Covers to Reach All Graphs

We observe 655 6-Clique Covers, 86 3-Cliques Covers, and 6 Components which has two clusters of 6-Cliques that are connected pair-wise with each other, and 2 isolated Covers among the 792 clusters. The minimum required Covers to create or 'reach' all graphs in the solution set is equal to 749 as we had described in Sub-section 4.4.3 in the distribution of Symmetric Cover Clusters of solution graphs that are isomorphic to each other (figure- 4.15).

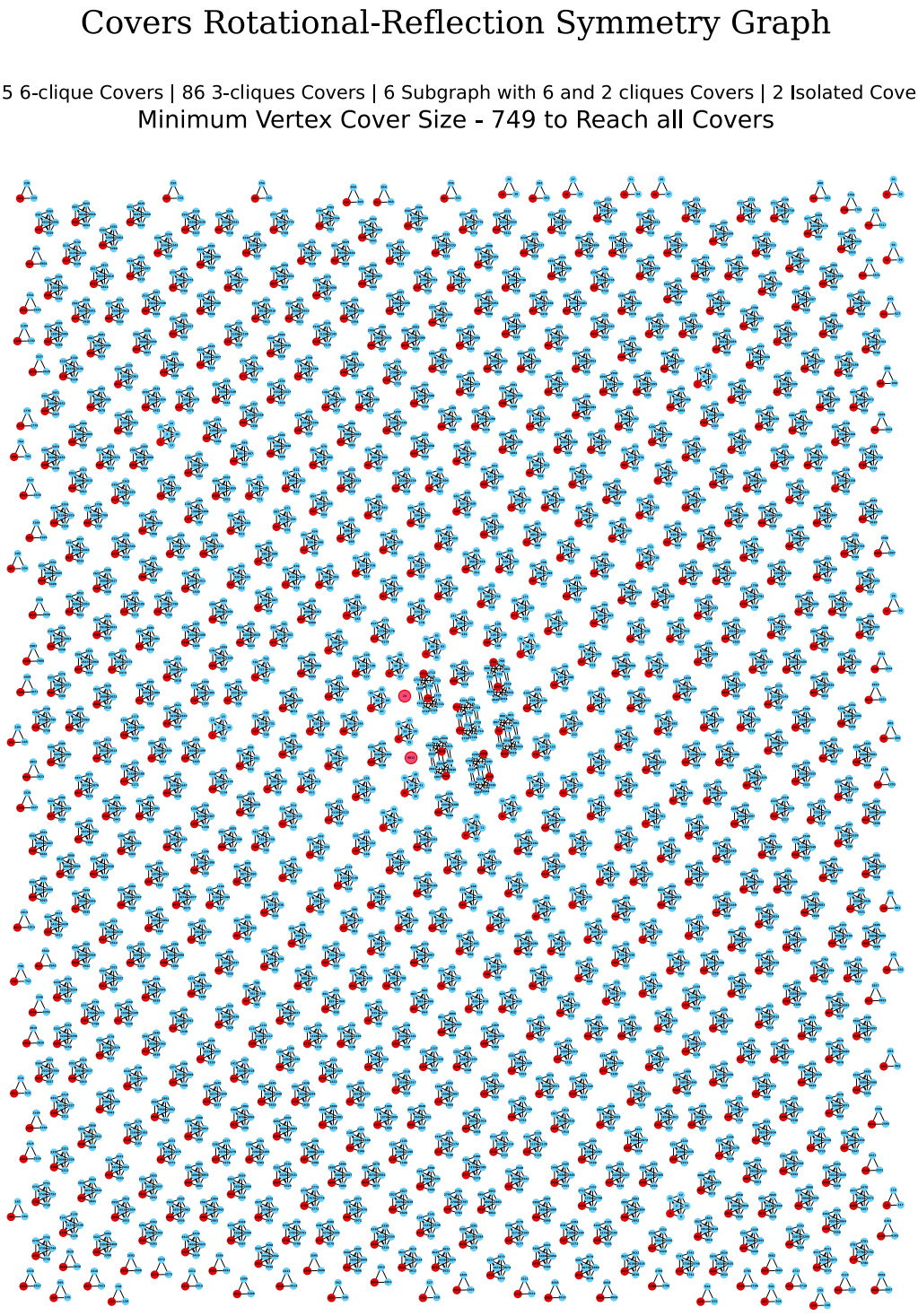


Figure 4.15: **Symmetric Clustering by Covers' Rotational-Reflection Symmetry** A plot of what the symmetric relations between solution covers look like. We see 6 clusters reflecting underlying Coat symmetry relations, the minimum vertex cover for all the Covers symmetry graphs is 749, and all symmetric Clique relations are seen in individual molecular types are observed in Covers, signifying all the original combinations of molecular types that are possible in the Covers exist.

Covers with Graphs in Different Evolutionary Layers

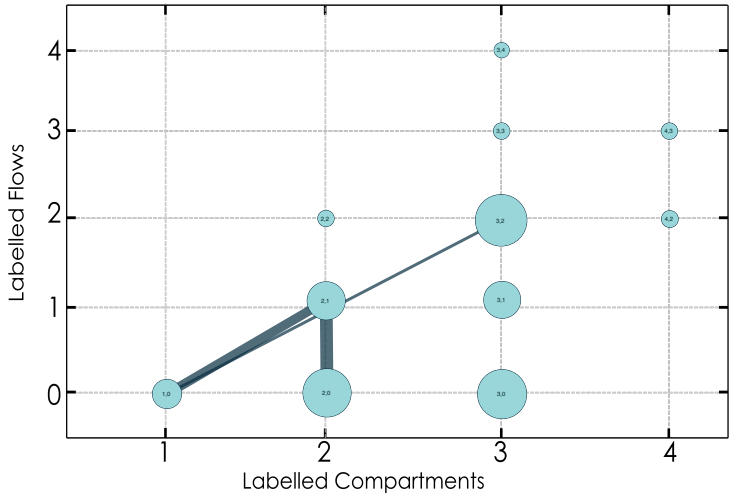


Figure 4.16: Topological Coat-Labelling Cluster Covers Rotational-Reflection Symmetry Graph

Figure 4.16 shows how same (or symmetric) Covers are distributed across the evolutionary layers and we can observe that a relatively higher number of Covers span different layers, but the internal layer-wise count of Covers with graphs in that layer has complementarily reduced. There is, again, no Cover that spans more than two layers.

Surprisingly, among the graphs possible under a genome, it is always the smallest Graph with the highest function! The smallest in the number of edges when bigger compartment graphs are not possible and the smallest in the number of compartments if higher compartment graphs are possible.

While observing the different sizes of graphs possible under a genome, we observe we can go higher in Graph size just the sub-duplication or the split of cargo types that are being flown (which means they are cycled) between the ER and another compartment. This observation resulted in a systematic manner of creating higher compartment graphs while keeping the genome size the same.

In fact, if we look at how likely the genome of a certain compartment-size graph is to create a graph of a higher compartment, we get an average prob-

ability of 0.346 (as shown in Figure 4.17), which is extremely high compared to Cover-Definition-2 Covers' probability of 0.005 of creating solution graphs of different node size in their allowed topological space.

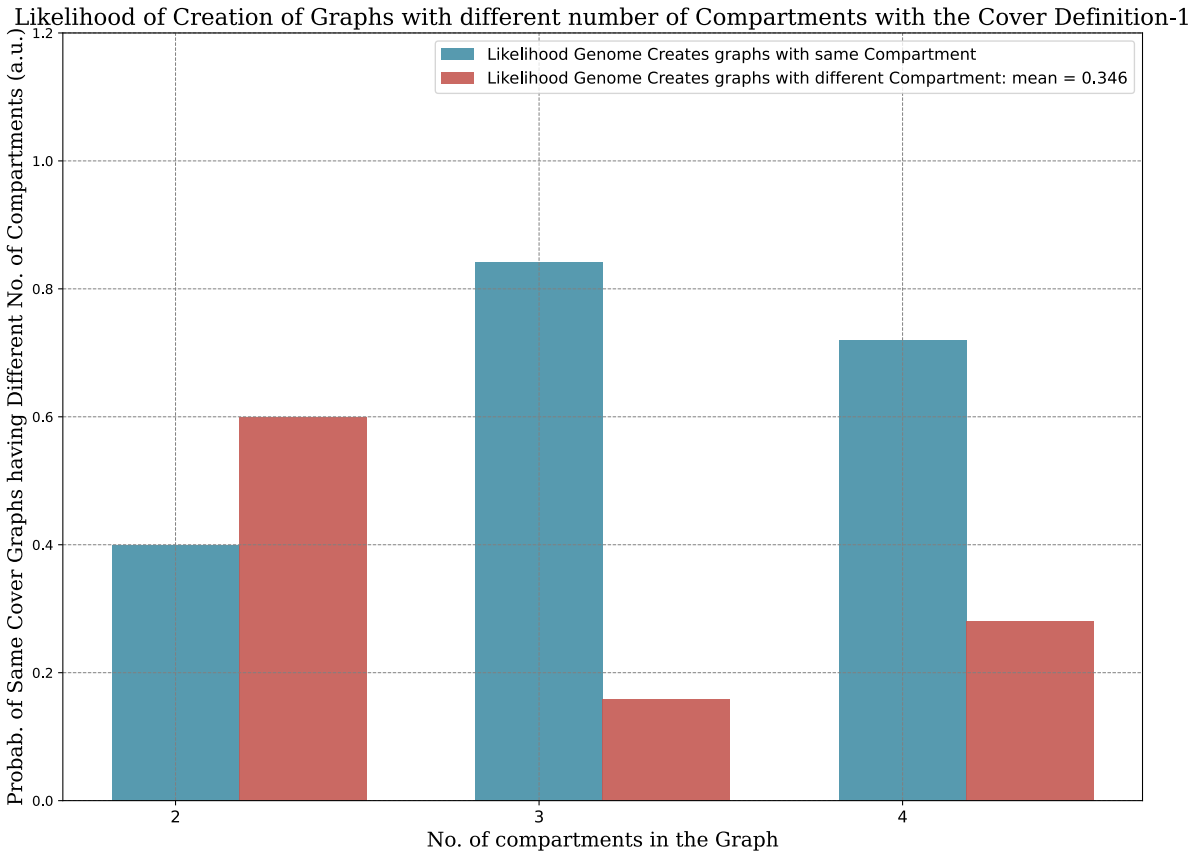


Figure 4.17: Topological Coat-Labelling Cluster Covers Rotational-Reflection Symmetry Graph

In fact we see highly diverse graphs co-existing under a genome, as shown in figure 4.18 below-

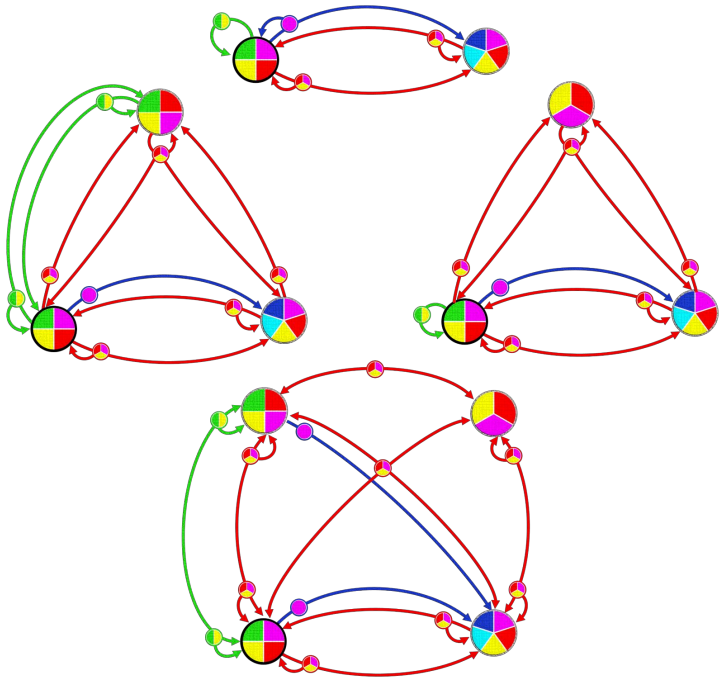


Figure 4.18: **Creation of new compartments by increasing cyclicity in specificity module flows.** This set of graphs have the same genome, i.e., they exist in the same topological space of a genome. Higher compartment graphs can be created by splitting cyclically flowing cargo types.

But how does this duplication take place? How does it not perturb the genome? Why doesn't any other compartment split to make the higher compartment? What does it stay about evolution of complexity in evolutionary time? All these questions have been attempted to be answered in the next sub-section.

4.5.2 The Splitting of ER and Creation of New Compartment(s) under same Genome

This is a sub-section to Sec. 4.5.

In this section, we address the issue of how well-defined a genome is to capture the topological state when taken in its entirety of events it creates in a graph.
We look at underlying principles which allow different topologies to co-occur within a molecular space which is the genome and we scrutinize whether such a genome existed during the evolutionary pre-LECA period. We also take a shot at going higher in topological space with the principles learned, and hinting towards a limit on the topological space from a graph.

How new Compartments are created through sub-duplication or Split of Existing Flows.

What kind of topological state variations are allowed while keeping the genome the same, and what is the underlying mechanism through which this could be happening?

Underlying Molecular landscape over which new Compartments can be created, the main rule: -

Terms used in this section:-

- *Split* or *duplicated*: unless specified otherwise, it means a part of the cargo type sets on the set of compartments being duplicated and split into the new compartment.
- *Common T-snare cluster*: the T-snares that are common between the set of compartments being looked at, as well as the same cargo types which makes the T-snares are also either V- and/or A- SNAREs. This

is the potentially splittable cargo-type cluster between/among the connected compartments.

- *The Circuit*: for a graph that allows splitting to create a new compartment, 'The Circuit' refers to the set of vesicles that carry in them only the set or a subset of the Common T-snare cluster. This might not always be self-sustainable and needs other return to ER depletion edges with the required coat.

- *Splittable* or *Unsplittable*: A graph is 'Splittable' if it can create a new compartment with a subset of cargo types shared between certain compartments through duplication such that the existing compartment compositions and vesicle compositions and the 'Genome' of the graph remains exactly the same. If it is not, then it is called 'Unsplittable.'

- *Conflict in targets*: The term 'conflict in targets' refers to the condition of the Theorem 4.1.1.

A Universal rule to check splittability under the same genome in any graph: Cyclical flows on the same ER-compartment pairs share the same T-snares, some of which are also cyclically flowing. When this shared set of cargo types is split or duplicated into another compartment, all the flows carrying the respective V-snares of the respective common T-snares cluster must also fuse to the new compartment, and in order to maintain the connectivity as well as to ensure pre-existing V/A and T-snares are maintained these again fuse back to pre-existing compartments, but this process involves budding from the newly created compartment. This implies that all the cyclically flowing vesicles that share common T-snares, as well as those vesicles that allow the cargo types in them to flow cyclically but may not share the same T-snares, must have this common T-snares cluster as their recruiters as well, to allow the split to take place. The creation of new compartments only takes place through the splitting of an existing budding and fusion machinery that is already uniquely flowing in a cyclical route.

Duplication of pathways takes place to new and pre-existing compartments during this process.

The way splitting of Compartments is allowed in this Activator-V-Snare Vesicle trafficking Model constraints creation of new compartments in existing networks by necessitating the existence of a shared machinery of targeting or fusion and of budding cyclically flowing between the ER and the pre-existing compartment for the vesicle pathways that carry this machinery cyclically.

We are not going to look at splitting of cargo that are stuck at ER because of never flowing- not even on replenishment or depletion flows. Their splitting can create a trivial topology in the case when all flows are cyclized, i.e., all routes have this stationary cargo type as their t-snare, and therefore they all fuse to it when it splits into a new compartment, but homeostasis can only be maintained if this stationary cargo is also a recruiter of the all the recycling routes of the V-snare fusing.

Theorem 4.5.1. *Depletion cargo types are never part of the splitter compartment.*

Proof. If a Cargo stuck at any of the compartments via depletion, but is also placed on a new compartment, then it would either imply a new depletion has taken place, and all the vesicles with their t-snare as this cargo type must fuse to this compartment. This incurrs two cases-(1) vesicle with cargo-types which do not have all their cargo types the same as their t-snares fusing to the split compartment, but in the new graph these also become the t-snare creating more molecules than there are existing in the genome. (2) The depletion is caused by a Half-closed cycle from the ER, which, when it exists, will cause even those vesicles that are other depletion to fuse to this new compartment, causing either the creation of new VATs or the same composition of the new compartment entirely. □

Theorem 4.5.2. *For a split of a set of cargo types into a new compartment under the same genome, all Coats that flow them cyclically between two compartments, i.e., the Coats allowing shared cargo types to flow cyclically between two compartments, must have these shared cargo types that serve as the t-snare as their recruiters as well.*

Proof. The idea of splitting is that cyclically flowing T-snares are duplicated to new compartment, and all the vesicles that are responsible for maintaining them in a cyclic flow state needs to get replicated even if they are not part of 'The Ciruit', this being referred to in the case of a depletion flow edge from the pre-existing compartment(s) to the ER, this vesicle doesn't have all its cargo in the shared T-snare cluster and by Theorem 4.5.1 cannot fuse to new compartment but still needs to bud from it in order to recycle the specific Class-2 cargo type which is on one of The Circuit vesicles to recycle. □

Lemma 4.5.3. *Half-closed flows on the ER will cause the new split Compartment to have the same composition under the same genome. This is because a half-closed cycle from ER to the pre-existing compartment indicates the depletion of certain cargo types while at the same time having the same shared T-snare cluster as the cyclical closed or unclosed flows; this includes a cycle formed by two different labels or closed cycle by one label and one depletion flow by another label. Thus everything has the same T-snare cluster and therefore needs to fuse to the new compartment.*

Lemma 4.5.4. *Half-closed cyclical flows on the Pre-existing Compartment, which has a closed cyclical flow(s) with ER of other coat(s), will cause the new split Compartment to either have the same composition in case of 1 closed cycle exists with ER or will cause depletion of cargo-type from pre-existing compartment to the new compartment and thereby change in T-snares for vesicles, under the same genome.*

Theorem 4.5.5. *The maximum number of Unique Compartments (Compositions) that can be created from a Graph, keeping the genome the same, is equal to two.*

Proof. There are only two kinds of shared t-snare clusters in the Graph-
(1) T-snares flowing Cyclically between ER and Some Compartments. This is a closed flow amongst these compartments to avoid conflict of targets. Even in the case of multiple such closed cycles between ER and other compartments, the closed cycles either have coupled T-snare clusters and will form only one new compartment with them or will form a new compartment with the same composition or are maximally split already (2) T-snares flowing only the ER by virtue of a coat-label only existing on the ER as a self-loop. There could be multiple such coat-label self-loops stuck at ER but they are all coupled to each other by the shared T-snare pool and will all form the new compartment together unless genome is perturbed. □

Theorem 4.5.6. *Creation of a new Compartment only Takes place Through the Split of cyclized T-snare between or among compartment(s) if and only if ER is among the compartments.*

Proof. This proof has not been formalized to the finish yet.
It, however, seems intuitive to be derived from the Class-0 cargo type 1 on the ER, always maintaining its unique composition as no flow can take it away. A cyclical flow that is one or more steps away from the ER implies either the cargo on the cyclical flow is completely formed by depletion, which can only happen if only one coat-labeled closed cycle forms it and, therefore, it is already in the maximally split state
or if in case more than one coat is cycling cargo types, then either these cycles carry just the same depleted cargo, which implies that there was a single depletion that took place but for keeping the compartments connected, some other coat was used in recycling the other depleted cargo, leaving a single Class-2 for this pair of compartments to cycle. But in this case, as well at least one coat will be the same as depletion flow feeding into the pair, which will lead to conflict in the target without self-loops or the recycling route of the depletion, implying that the cargo carried by the recycling is also depleted leaving both this vesicle to flow only Cargo type 8 causing conflict in targets.

In the case when there are more than one coats and more than cycled cargo type: cycling of more than one Class-2 cargo types which cannot be brought to into flow unless at least two depletion coats exist at one or both the compartments in the pair. But again for complete segregation of t-snares from previous compartment a third label will be required to format half cycle in the recycling path to make the graph valid. However, now we have used up all the edges and any combination of coats that can recycle the 2 Class-2 cargos will either lead to depletion and conflict, of targets or a direct conflict in targets. □

Addendum: Theorem 4.5.1, Lemma 4.5.3, Lemma 4.5.4 do not apply in the case of presence of a different splittable SNARE Cluster which is basically a self-loop at the ER. In the case when the Circuit does not satisfy the conditions of Theorem 4.5.1, or Lemma 4.5.3, or Lemma 4.5.4. the other SNARE cluster can be coupled to the splitting Circuit to create a unique compartment which does not perturb the existing compartment.

Theorem 4.5.7. *Self-loop clusters of SNAREs are always split along with the SNARE cluster which is shared between compartments in a coupled manner.*

Proof. Due to Theorem 4.5.6, we know that other SNARE clusters will always be rooted at the ER, and because any pair of coated vesicles will share at least one Class-2 cargo types, therefore, the two different SNARE clusters will share at least one of the SNAREs they carry this. The shared SNARE cluster between compartments can into split a new compartment without the self-loop cluster also splitting to the new compartment it creates because this shared circuit has targets at the ER as well therefore no Conflict in Target for shared SNARE(s) cargo types takes place. The same is not the case when

a new compartment is formed by the self-loop because now the SNAREs are being split into a different compartment, and therefore the circuit must also acquire its targets and cyclical routes their to avoid Conflict in targets and in recruiters as well. □

Not all 2 Compartment Graphs can create a higher compartment graph because, even when there exists a closed cycle at the ER, this is because there could be a Heterogenously-closed cycle formed by two different labels between the ER and the other compartment, and since either of them will have recruiters involving Cargo-type 1 at the ER and the other will cause depletion of a second class cargo because the first label edge from the ER can't occur at any other compartment than the other.

Is it possible, then, to create 3 new compartments from 2 compartment graphs, because we have 3-Coats? No, it's not possible while keeping the genome the same because of the rule of Half-closed-cycles. If we assume there are no Half-closed-cycle for any coat, then either we are left with no self-loops on the ER or all 2 or 3 self-loops on the ER at least one of which has to be a closed cycle to keep the graph connected. But in the latter case, assuming 3 self-loops on the ER which we can split into 3 compartments, we observe that the self-loops share T-snares amongst each other cargo types as well as their cargo types are T-snares of the closed; at the end, we are left with at most two unique cargo-type clusters! If we consider only two self-loops existing and one of the other Coats does not form a closed cycle, it will be a depletion edge. Again we are left with two clusters of cargo types that can form two unique compartments because one of the self-loops on the ER does not share T-snares with the closed cyclical flow.
Therefore, this can only be made possible by allowing changes in the genome.

When the higher compartment graph lies in the forbidden zone of the current graph.

An interesting case where there is no half cycle but a higher compartment graph lies in the conflict between having different T-snares on different compartments isolated but still having to duplicate into the new compartment to maintain the recruiters, and this new compartment will have neither of these self-loop vesicles T-snare as it is a shared flow, now vesicles have to decide which compartment to go to. Regardless of where it chooses to go, it will occur in the forbidden zone of the graph because it is that shared subset of the two self-loop vesicles that is forbidden because of conflict in the targets.

How the creation of a New Compartment under the same Genome Impacts the Functions in the Network

In the context of evolution, it is necessary to note that the new compartment never has a function, as defined in our system, when it is formed under the same genome. Therefore, mutations are required to confer functionality to the new compartment and cannot be changed.

Theorem 4.5.8. *Creation of a new Compartment, while keeping the genome the same, Never Increases Functions*

Proof. Obvious from the fact that cargo types are already shared between the compartments, and therefore, no markers are getting split into new compartments, so labeled compartment numbers never rise or decrease.
The labeled flows, which are a property of unique flows, cannot increase as the new compartment created doesn't have any unique labels. The labeled flow decreases when a predictable motif-1 exists with the predictable flow coming from the pre-existing compartment to the ER because the share-t-snare is splitting these two different coat vesicles will be required to fuse to both Er and the newly formed compartment to maintain connectivity and Genome integrity. Still, it no longer has a unique target and is no longer

predictable. □

This implies that for a genome to encode the highest number of functions in a cell, it selects the smallest region in its allowed topological space. And, while the same genome can create larger compartment graphs to encode the same or more amount of functions in higher compartment graphs or to get selected during evolution, it needs to mutate into a different genome.
This in a way, suggests a very important result in our model that the creation of molecular machinery of budding and fusion precedes the formation of a new compartment.

abcdefgh

An important observation constraining Higher Compartment Networks

The way splitting of Compartments is allowed in this Activator-V-Snare Vesicle trafficking Model constraints creation of new compartments in existing networks by necessitating the existence of a shared machinery of targeting or fusion and of budding cyclically flowing between the ER and the pre-existing compartment for the vesicle pathways that carry this machinery cyclically.

This phenomenon in our Model alludes to Martin W., 2010's (Martin, 2010) Hypothesis that - Origin of Metabolic Compartments in Eukaryotes takes place after the transfer of an enzymatic pathway from one compartment to another is established slowly over evolutionary time by the gain of the required targeting signals individually by each enzyme.
The context of requiring mutations to be retargeted, and that each individual enzyme has to undergo, is captured by the constraint of pre-existing shared t-snare and recruiter machinery, which, if not present in the genome, will cause the genome to change if the paths are still included. Even with mutation, certain graphs cannot retarget all their pathways between two compartments onto a new one because the machinery required falls into the forbidden region of the graph, which will destroy the existing network, i.e.,

rewiring of certain paths onto a different compartment is not always possible. However, as we observe very clearly, there are graphs where we need no mutation to create a new compartment with all flows established to and from it because the machinery pre-existed. This implies that there does not need to be dual-targetting of proteins to a gained compartment but only the intermediate step of the split of shared cyclized cargo between compartments. This hints strongly to a non-endosymbiotic origin of the 'Third' compartment.

4.5.3 A hint towards intermediate stages during evolutionary process of acquiring complexity

The Splittability of a Genome: A case Study

A Genome, if it has some cargo types that are never found as V-snares or A-snares, are depleted cargo types. If such a genome still shows sign of 'splittability', i.e., a cluster(s) of V/A-snares that are also T-snares, where the whole of the V/A-snares in the cluster, or a subset of them will form a complete individual vesicle-type composition; however, there exist some cargo types in VATs which are never both V-snares or A-snares simultaneously indicating these cargo types are also on vesicles that have a different target(s) then this cluster's vesicles. These are vesicles that are either fusing to only ER from the pre-existing compartment(s) or to only the pre-existing compartment (from the ER). As we saw in the theorems and proofs in the previous sub-sections, splitting is only possible in the former case as these vesicles serve the purpose of cycling the cargos; they can be sourced at the new compartment while keeping their fusion target intact if and only if they also have one of the cargo-types in this t-snare cluster as the recruiter.

Thus, if there is a Half-cycle on the ER, which is part of this shared t-snare cluster between/among compartments, then splitting cannot take place with-

out perturbing the existing cover in case of a depletion flow outside of the split cluster or in case of no vesicles outside this cluster from ER exist then the only possible higher compartment graph which can form via splitting is if the new compartment which forms has exactly the same composition as the preexisting one.

What does this imply?

This suggests that a genome can have a set of pathways to transport cargo using the same set of VATs but these pathways will still remain splittable as long as all the cargos in the new vesicle are being cycled back. However, this is only the case when there exists vesicle(s) with exactly the same coat and forming a half-cycle from the target(s) end that recycles the cargo because Class-1 cargo types cannot recycled otherwise. If in case a pathway emerges in this circuit that is splittable, through the deletion of half of a closed cycle at the target or creation of one at the ER, bearing no closed recycling route will cause a splittable component of flow to affect either redundancy in the new compartment or a genomic perturbation in the case, it takes place as depletion of a cargo type (can be Class-1 or Class 2 depending on what returning vesicle coats exist) would have taken place binding the splittable circuit to the specific compartment pair or group by virtue of becoming a pathway which carries all T-snares for all flows between/among those compartments or by virtue of providing T-snares to the compartments' non-splittable flows. And this is how the un-splittability of these pathways can be a coincidence detector for when different pathways become dependent on each other for their functions.

That a graph that has cyclic pathways for a shared set of snares, must not have a pathway which is also a depletion flow that marks the target compartment(s). If it does, then it implies that not just the pathway that carries depleted cargo but also the other pathways that share the snares

with this pathway are bound to fuse to the same target composition. In the case of no other unique pathway carrying depletion cargo that exists from ER, then the new compartment can only have the same composition as the previous one. In case a unique pathway exists, which does not share a t-snare cluster by virtue of not fusing to all of the compartments containing the t-snares, marks the compartment(s) with another unique cargo, then the shared t-snare cluster can't form another compartment without perturbing the VATs. If it preserves the VATs, then it means that the unique pathway was also replicated, and the new compartment has the same composition as the pre-existing one.

The hint toward operations leading to redundancy: co-occurring vesicle budding and fusions between or among compartments have an inherent redundancy in what is being flown by virtue of our coat-cargo interaction matrix; this further makes molecules, especially t-snares, redundant due to existing across the same targets, unless in the special case of depletion, which allows unique identification of not just the compartment(s) it marks but also uniquely unites the co-occurring flows from the source compartment to the target(s), and prevents their split.

Remark. ***Conclusions from this Section: Significance of Unsplittability in Evolution***

Either new recycling paths for the split cargo require new recruiters for coats, or the vesicle pathways sharing specificity modules have become significant in the genome as a result of marking a compartment that happened through the loss of closed recycling. This is when the cell is required to lose specificity modules (except in the case when it can still have cargo types that are not depleted yet, in which case loss of recruiters would incur on the

half cycle*) and to decouple the depletion pathway from the rest of circuit and allow it to split into a new compartment. So what was the topological nature of the primitive cell that gave rise to a new, non-endosymbiotic compartment, and which route did it take? Clearly, to decouple a pathway from the circuit that shares t-snares and recruiters, we have to delete certain t-snares of the circuit as well as recruiters for the coat(s) causing depletion in the circuit. Whereas gaining a recruiter to allow recycling at the new compartment formed seems like a much easier route for the cell. Therefore, being in a state of coupled depletion flow imposes more constraints from the vantage point of the genome on creating a new compartment. However, another case must be considered while commenting on what could have been the case for primitive cells and another likely route evolution could have taken in the intermediate stages.

* Takes place when the ER still has coats that are only a self loop on it and are not yet spread to other compartments.

It remains to be defined and seen at a greater depth, the coupling of the constraint on the functionality of the cell and the biophysical constraint resulting from the integrity of the cell's genome.

4.5.4 Creation of Higher Compartment Graphs Through Splitting Unconstrained by same Genome.

This is a sub-section to Sec. 4.5.
As we saw in the previous section, cyclization of cargo types, the budding and fusion machinery, is required. We also saw that co-localized different SNARE clusters will require mutation in the case they need to split into different compartments altogether.
Therefore, the experiment of allowing mutation was undertaken, and the following constraints were discovered in order to split a vesicle or set of vesicles into a new compartment.

A weaker version of Theorem 4.5.2 holds in these cases -

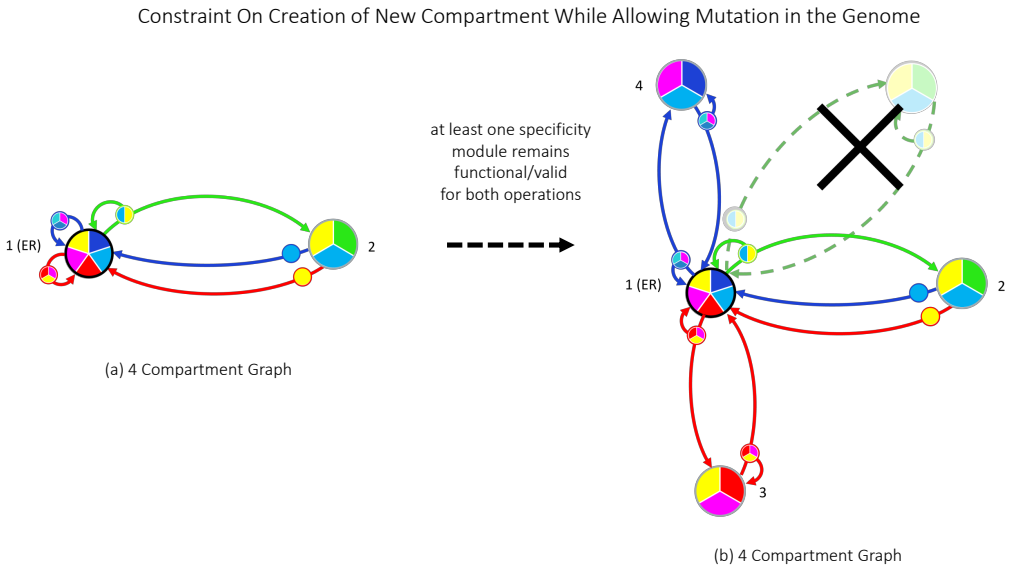


Figure 4.19: **Creation of New Compartment While Allowing Mutation in the Genome** There are two constraints while splitting a vesicle or set of them into a new compartment - 1st is the requirement of a unique cargo or their combination in the vesicles to exist on a strongly connected component with the same coats, and 2nd is that at least one of the cargo types must either be unique to the vesicle or have the same budding and fusion events in all vesicles it exists in. This is to ensure at least one regulator for the new paths created exists. Here, the blue and red coat vesicles satisfy this constraint and can, therefore, be split into different compartments or even into one single compartment. The green coat, however, doesn't satisfy any of these constraints and, therefore, can't split. .

Theorem 4.5.9. *A subset of shared T-snares can be removed into a new compartment if and only if this subset has a specific Coat with at least one cargo type in this subset as its recruiter flows it in a closed cycle.*

Proof. The cycle made by non-closed coats cannot be separated even if it carries a subset of cargo types currently existing on another compartment because it conflicts with the targets or recruiters. □

The requirement is that at least one regulator for paths that already exist must continue to exist, that is there must be no Conflict in targets or lack in recruiter, as well as for the paths that are being created anew for the cargo types being split into a new compartment. Please see the figure 4.19 below for visual representation of the same constraints.

Following is a more deeper analysis of these constraints-

Universal Rules for Splitting Constrained only by Validity of Graph, molecules can mutate:

For a vesicle composition to be turned into a compartment using a closed cyclical flow, the following conditions must be satisfied-
1. The vesicle has at least one cargo type or a combination of cargo types inside it, which has a strongly connected cycle from its source made of the same coat vesicles, which includes self-loop if present on only one compartment and a closed cycle when present on multiple compartments. This ensures at least one t-snare in the set remains functional/valid and allows fusion to the new compartment.
2. To ensure the recruiter condition, among the cargo types found to be flowing on the strongly connected cycle, at least one of them must either be uniquely present on the vesicle being split or, if they exist on different coat vesicles, they all have exactly the same budding and fusion compartments as the splitting vesicle.
3. In the case when the vesicle to be split is not on, or rooted at, the ER, it must be ensured that there are no vesicles with a subset of its cargo because if they are present, then they also need to fuse to the new compartment, along with any other vesicles which carry their cargo types in their composition to maintain some functional t-snares to continue existing. In many cases this would lead to a compartment which will have the same composition as the pre-existing one, and therefore, the vesicles with cargo type subset must be eliminated.

Following is an example of genetically unrestrained splitting of compartments creating a 5-compartment graph-

We found many cases of smaller compartment graphs that were able to create the higher compartment graph, but more of them required disrupting

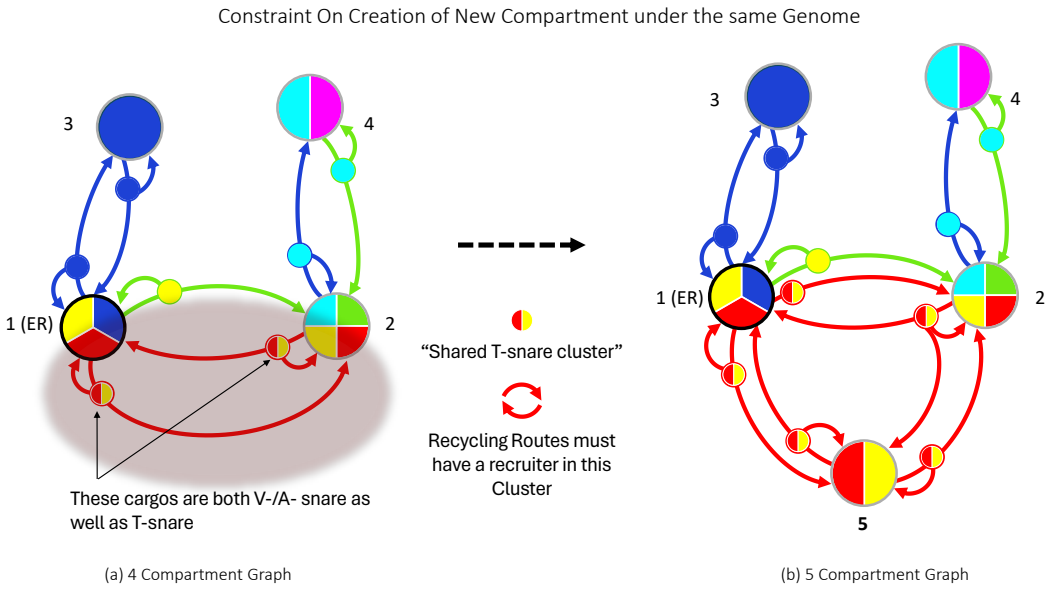


Figure 4.20: **Splitting of Cluster SNAREs from the Circuit.**: (a) A 4 Compartment 14 edges graph with a closed cycle by coat 'a1' but marker cargos on both compartments at the closed cycle. (b) The 5-compartment graph, with 19 edges (the smallest 5-compartment graph obtained from the solutions. The 5th compartment is created by splitting the closed cycle circuit of the 'a1' coat.

some elements in the genome. As we can see from the figure 4.18 here the compartments can split further if mutation is allowed, as depicted in figure 4.22, and also gain function as shown in an example in figure 4.21 below.

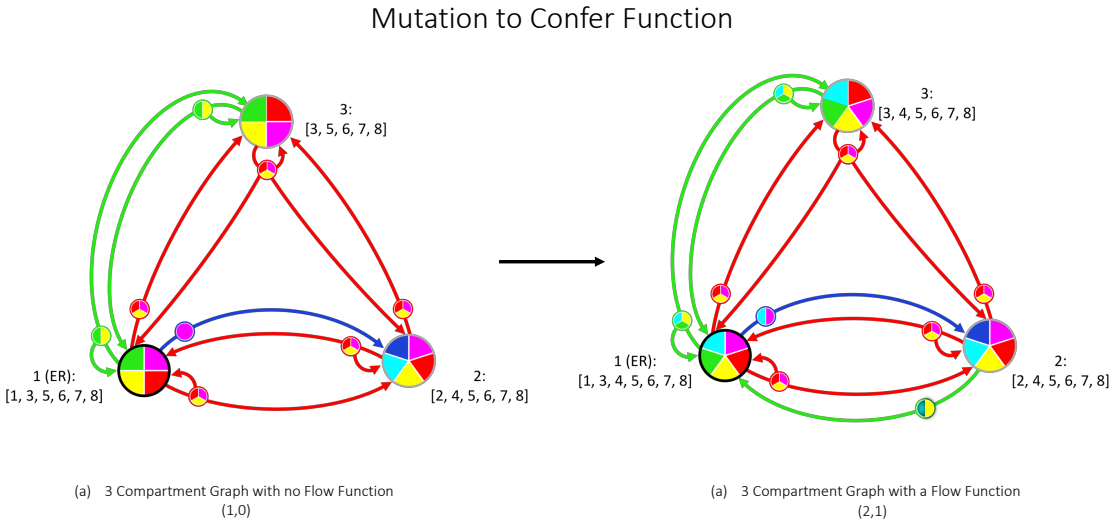


Figure 4.21: **Conferring Function to newly created compartment by allowing mutation.** We see that the 3 Compartment graph (on the left) created from the two-compartment graph has no flow function, but the addition of a new pathway allows an identifiable flow.

However, all of these cases involve the splitting of closed cycles, and none of them lead to an increase or decrease in functions. More sampling of lower and higher compartment graphs is required to ascertain the nature of such unrestrained splittings.

Remark. There is a way depleted cargo can split into a compartment; this is by the use of a 3rd label(if not already in use) for the creation of a heterogenously closed cycle from the depleted compartment, forming a half cycle, and use of the depleting coat to complete half cycle rooted at the new compartment.

We can never have depletion to two compartments and the isolated flow of the depleted cargo type to those compartments.

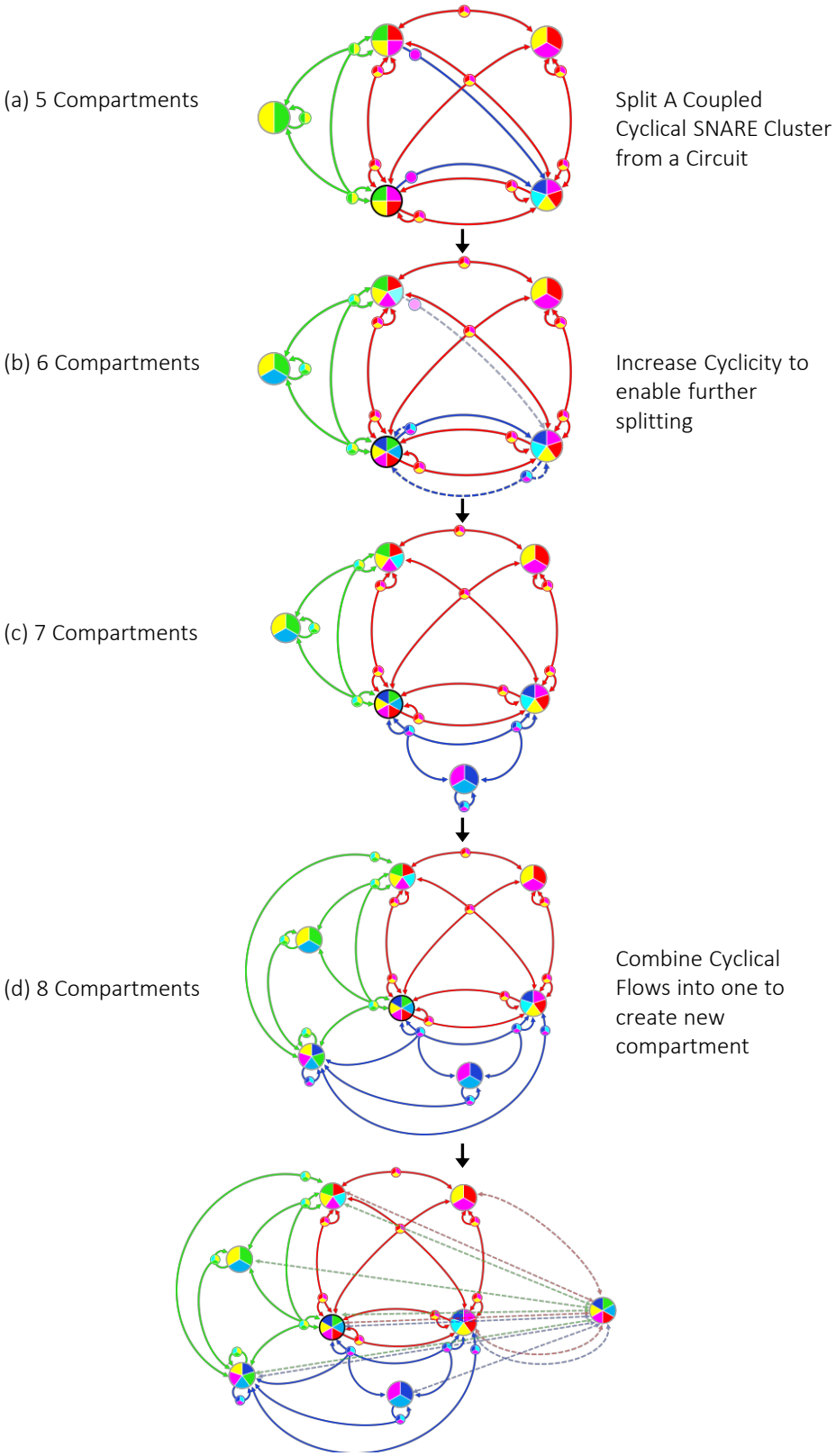


Figure 4.22: **Creation of new compartments by increasing cyclicity in specificity module flows.**

Biological Implication:

Upon Multi-targetting of Proteins to Organelles The fact that we need to increase cyclicity to allow a new compartment under the same regulation and coat system hints towards a plausible reason for the co-occurrence of trafficked proteins in multiple compartments, as well as duplication preceding innovations in trafficking routes. It might not be so much of an "imperfection" but a reminiscence of the concurrent duplication and split of compartments and specialization with the gain of new routes or different modes of regulation thereafter to gain function.

This result is doubly complemented by the fact that cyclization allows molecules that are more specialized to the specific coat to come into play as regulators. At the same time, if the path regulators are fewer and more specialized, it will allow for the change of paths(addition or deletion) more easily.

Upon Underlying Duplication Events Based on the explicit definition of the "Genome" we get two kinds of duplication events in our system. The same genome when it means the same set of genes, including the multiplicity of genes, inokes the question of how two cells can have different compartment sizes when the genome is identical. We cannot circumvent it without duplicating all the machinery involved in creating the new compartment; hence, the hypothesized co-duplication has occurred (Dacks and Field, 2007).

However, suppose you say these are not two different cells but states of the same cell in which the lower compartment size is chosen as a result of the function. In that case, it is simplified to the requirement of events of mutation - duplication and/or deletion- to realize the higher compartment size cell. In this case, the split compartment is just a transient state upon which selection operates discretely as mutational steps.

A final Algorithm to go higher in Compartments while keeping the genome the same.

This algorithm can be improved to create a higher compartment with the sub-set of the t-snares as well in the cases when the recycling condition cannot be satisfied. It will allow us to sample the creation of a larger graph more exhaustively.

Algorithm 8: Generate Higher Compartment Graph

```
Require: Genome, Graph
Ensure : Valid Graph with Higher Compartment(s)
Function GenerateHigherCompartmentGraph(Genome, Graph):
    Cluster T-Snares for each coat type ;                          // (1)
    Identify vesicle types with shared T-Snares clusters ;         // (2)
    Retrieve edges of identified vesicle types ;                   // (3)
    Segregate vesicle types based on shared compartments ;         // (3)
    for each group of vesicle types do
        for each vesicle type in the group do
            if forms a closed cycle with other vesicle types of one coat
             then
                Add (edges,coat) with new compartment as source ;
                 // (4)
                Add (edges,coat) from targets to new compartment;
                Add (self-loop,coat) on new compartment;
            else if self loop then
                Create (edges,coat) to new compartment to form closed
                 cycle with same coat;
                Create closed cycle (edges,coat) for vesicle types' coats
                 that have their t-snare in the Shared t-snare group.
            else
                Check for recycling vesicle types for each pair of
                 compartments do
                    if minimum set of recycling vesicle types exists then
                        Append (edges,coat) going to new compartment;
                        Append (edges,coat) from new compartment to
                         target compartments;
        Check common T-Snare cluster for coat-labeled edges ; // (5)
        Add missing edges sourced at new compartment;
        for each coats with recruiters in common T-Snare cluster do
            Add missing edges sourced at new compartment ;     // (6)
    return New edges created;
```

Chapter 5

Conclusions and Future Directions

We now conclude all the results and reflect upon where they point us to. The results are open to interpretations and many hypotheses seem possible.

The number of graph solutions is very scarce and the emergent landscapes are vast. We need to look through the vision of finding relations between topology and the emergent phenomenons to account for how the creation of Vesicle traffic networks took place through the numerous routes and driving factors and constraints that appear as bottlenecks in the information as well as evolution. An overview of the results and future prospects is as follows: -

1. For a genome, there exists an allowed topological space that is bounded on both sides. This can be proved quite easily based on the Algorithm 7.

2. Adding edges or compartments while keeping the cover the same, cannot increase the the number of predictable functions the graph has.

3. This implies that for a genome to encode the highest number of functions in a cell, it selects the smallest region in its allowed topological space. And, while the same genome can create larger compartment graphs, to encode the same or more amount of functions in higher compartment graphs, it needs to mutate into a different genome.

4. The number of unique cargos flowing on the predictable path between labeled compartments can only be 3, belonging to Class-2 cargo types. This suggests that only those cargo types with having a different molecular machinery for recycling back to the source can be predictable.

5. A specific cargo type can't have a 2-step predictable flow because the 2 motifs allowing the predictability of a cargo type cannot exist concurrently adjacent to each other.

6. (4) and (5), along with the constraint on the fixed number of cargo types being carried by each vesicle and the constraint of Conflict of Targets indicate- not only a limit to the number of predictable functions any graph of any node- or edges- size can have, but also an upper bound on the highest possible functions, as well as highest possible graph size,

our Model will allow at a certain coat-system because the number of Class-2 cargo types are limited. This needs to be proved rigorously.

7. With the same genome, ER can sub-duplicate or split the cargo types on a cyclical route with another compartment(s) or even those that are flowing on a self-loop at the ER and form a new compartment; this compartment maintains the cyclical routes of its cargo types as it has the same molecular machinery to recruit, ship and fuse vesicles. The number of cargo types flowing in separate cyclical clusters is the number of new compartments that could be formed, I have shown this is at most 2.

8. It is evident from (7) why predictable cargo paths shipped from one compartment to the ER can get lost during this sub-compartmentalization. If we allow the genome to mutate during this process of gaining a compartment we can either lose or gain a marker during sub-compartmentalization.

9. (7) Also suggests a very important result in our model that the creation of molecular machinery of budding and fusion precedes the formation of a new compartment. Together, (7) and (8) suggest that while creating a new sub-compartment can take place via splitting with the same genome, this new cell will only be selected under our Selection model, or this new compartment only gains function after the genome mutates.

10. During selection, when we look at transitions the way defined by Sahana and Mukund 2022 (Unpublished), the molecular network or an irreducible cover of a graph that has the cargo types with minimum interactions, the more 'specialized' class-1 cargo types, with other molecules are selected, as they are less likely to mutate and form new molecules, the reason for which is also very nicely captured in the fact that the functions they cover are very unique and if they are only once covering it then their mutation would be lethal, while their creation won't perturb any other event. even exist as a molecule in the other graph.

11. Our model doesn't allow the split of marker or cargos on self-loops or cyclically shared cargos with other compartments for any other compartment than ER: because if they do split, then all the vesicles that are cognate to them, i.e., have them as t-snares, will also follow including depletion flows, as well as the self-loops which will form a closed cycle, and the various reasons mentioned in theorem 4.5.6. This puts ER at an even higher pedestal in determining the internalization of a new endo-symbiotic compartment into the VTN as well as in the creation of a non-endosymbiotic compartment via splitting.

12. Hypothesis of all higher compartment graphs occurring via sub-duplication of lower compartment graph is wrong; other than the explanation provided at the observation of some 2 compartment graphs that cannot split, we can explain it in another way. Some compositions of higher compartments never have a subset of their composition, a smaller graph in the solution set, or not all genomes have different compartment systems.

13. Since we could not find 5-compartment graph-pairs via sub-duplication because sampling of 3 compartments and four compartments is not complete, we create higher compartment graphs via either sub-duplication by keeping the cover intact- which is not true even for the case of 3 compartments trying to create from 2 compartments, I sought the method of sub-duplication with loss of cover integrity.

14. Hypothesis of all higher compartment graphs occurring via sub-duplication or 'splitting' of lower compartment graph is wrong; other than the explanation provided at the observation of some 2 compartment graphs that cannot split, we can explain it in another way. Some compositions of higher compartments never have a subset of their composition, a smaller graph in the solution set, or not all genomes have different compartment systems.

15. The number of compartments can increase under the same genome

through sub-compartmentalization, but these cells could go lower in our evolutionary layers, and they'll have to undergo a separate set of mutations to climb the evolutionary ladder. This seems counterintuitive to the school of thinking that taking up a new compartment gave rise to more functions. However, it supports the randomness in evolution, which can take up any direction along the functions' axis. Our model ratchets the functions as long as two cells are one mutation away, but what if the actual story of going from the first graph to the other higher compartment graphs involved a step backward? That is, if a graph only has 2 compartments, but its whole genome allows the creation of a 3rd compartment, then won't it be easier for it to go that state before directly gaining a labeled compartment? A loss of function could be costly, however, there could be biophysical constraints of not having another compartment, let alone it being identifiable with a maker. Thus the concept of splittability of genomes is fascinating.

16. The way splitting of Compartments is allowed in this Activator-V-Snare Vesicle trafficking Model constraints creation of new compartments in existing networks by necessitating the existence of a shared machinery of targeting or fusion and of budding cyclically flowing between the ER and the pre-existing compartment for the vesicle pathways that carry this machinery cyclically.

17. There is a deeper significance to depletion or replenishment flows: such a pathway which is acting to flow very slowly degrading cargo to its target, is definitely one of the more specialized functions in the solution graphs. As shown in the previous chapter, we can trace the minimum Hamming distance, or minimum mutational distance path, starting at just a single compartment and all vesicle fusions to itself, going through the loss of this self-loop recycling to the new compartment, eventually gaining a recycling route, and then slowly acquiring all coats to be recycled again in closed cycles which pushes the ticker back to zero and the process of gaining a replenishment flow starts by loss of recycling. This is when the

cell is required to lose specificity modules and to decouple the depletion or replenishment pathway from the rest of the circuit and allow it to split into a new compartment.

18. It remains to be defined and seen at a greater depth, the coupling of the constraint on the functionality of the cell and the biophysical constraint resulting from the integrity of the cell's genome.

19. The phenomenon of splitting or sub-compartmentalization in our Model implies that there does not need to be dual-targetting of proteins to a gained compartment but only the intermediate step of the split of shared cyclized cargo between compartments.

20. This strongly hints towards a non-endosymbiotic origin of the 'Third' compartment.

21. Since we could not find 5-compartment graph-pairs via sub-duplication because sampling of 3 compartments and four compartments is not complete, we create higher compartment graphs via either sub-duplication by keeping the cover intact- which is not true even for the case of 3 compartments trying to create from 2 compartments, I sought the method of sub-duplication with loss of cover integrity. However, a more nuanced algorithm than Algorithm 8 is required to check this, and it might allow us to get to the limits of our model finally, the highest compartment it can generate, and see whether the rise in the compartments is really a rise in complexity.

22. The whole analyses done for the Activator Model needs to be done for other regulation models, as evolution could have switched path with regulation at any point to get to the current regulatory network of Eukaryotic Vesicle Trafficking preserved across extant eukaryotes today.

23. It has been hypothesized that splitting of pathways involved in multi-enzyme reactions to a new compartment poses a challenge and would have occurred in evolutionary steps with the raw material of dual targetings of proteins involved. However, we have shown pathways could be split into a new compartment if their existing regulatory machinery is being recycled. This indicates that this targeting of proteins to multiple different organelles could have been a result of this process of concurrent duplication of pathways while creating new organelles.

24. The genome of a Vesicle Traffic Network needs to be defined precisely in order for us to trace the underlying evolutionary genetic events. The hypothesis of co-duplication of existing machinery for the creation of new organelles by (Dacks and Field, 2007) is disproved when taking on the definition of irreducible covers where a single mutation can create a new compartment but is supported by the biophysical constraints in Splittability of a VTN. This needs to be studied more carefully through definitions and could validated via Genomic studies of events preceding the creation of new compartments in the Crown group eukaryotes' evolutionary trends or in the genomic traces collected from the fossil evidence of the extinct Stem Lineages.

References

Aggarwal A, Rangan CP, Takaoka T (1999). An o (1) time algorithm for generating multiset permutations. In Algorithms and Computation: 10th International Symposium, ISAAC'99 Chennai, India, December 16–18, 1999 Proceedings 10, 237–246. Springer.

Albert R, Robeva R (2015). Signaling networks: Asynchronous boolean models. In Algebraic and discrete mathematical methods for modern biology, 65–91. Elsevier.

Alt H, Mehlhorn K, Wagener H, Welzl E (1988). Congruence, similarity, and symmetries of geometric objects. Discrete & Computational Geometry 3(3), 237–256.

Arakel EC, Schwappach B (2018). Formation of COPI-coated vesicles at a glance. Journal of Cell Science 131(5), jcs209890.

Aridor M, Traub LM (2002). Cargo selection in vesicular transport: the making and breaking of a coat. Traffic 3(8), 537–546.

Armstrong MA (1997). Groups and symmetry. Springer Science & Business Media.

Arora D, Damme DV (2021). Motif-based endomembrane trafficking. Plant Physiology 186(1), 221–238.

Balch WE, Dunphy WG, Braell WA, Rothman JE (1984). Reconstitution of the transport of protein between successive compartments of the golgi measured by the coupled incorporation of n-acetylglucosamine. Cell 39(2), 405–416.

Barlow LD, Nỳvltová E, Aguilar M, Tachezy J, Dacks JB (2018). A sophisticated, differentiated golgi in the ancestor of eukaryotes. BMC biology 16, 1–15.

Baum B, Baum DA (2020). The merger that made us. BMC biology 18, 1–4.

Baum B, Spang A (2023). On the origin of the nucleus: a hypothesis. Microbiology and Molecular Biology Reviews 87(4), e00186–21.

Baum DA, Baum B (2014). An inside-out origin for the eukaryotic cell. BMC biology 12, 1–22.

Bhattacharyya A, Gupta A, Kuppusamy L, Mani S, Shukla A, Srivas M, Thattai M (2021). A formal methods approach to predicting new features of the eukaryotic vesicle traffic system. Acta Informatica 58, 57–93.

Biggs N, Lloyd EK, Wilson RJ (1986). Graph Theory, 1736-1936. Oxford University Press.

Bonifacino JS, Glick BS (2004). The mechanisms of vesicle budding and fusion. cell 116(2), 153–166.

Brocks JJ, Nettersheim BJ, Adam P, Schaeffer P, Jarrett AJ, Güneli N, Liyanage T, van Maldegem LM, Hallmann C, Hope JM (2023). Lost world of complex life and the late rise of the eukaryotic crown. Nature 618(7966), 767–773.

Butterfield NJ (2015). Early evolution of the eukaryota. Palaeontology 58(1), 5–17.

Cai H, Reinisch K, Ferro-Novick S (2007). Coats, tethers, Rabs, and SNAREs work together to mediate the intracellular destination of a transport vesicle. Developmental Cell 12(5), 671–682.

Cao TB, Saier MH (2003). The general protein secretory pathway: phylogenetic analyses leading to evolutionary conclusions. Biochimica et Biophysica Acta (BBA) - Biomembranes 1609(1), 115–125.

Cavalier-Smith T (2002). The phagotrophic origin of eukaryotes and phylogenetic classification of protozoa. International journal of systematic and evolutionary microbiology 52(2), 297–354.

Cavalier-Smith T (2010). Origin of the cell nucleus, mitosis and sex: roles of intracellular coevolution. Biology direct 5, 1–78.

Cheng X, Chen K, Dong B, Yang M, Filbrun SL, Myoung Y, Huang TX, Gu Y, Wang G, Fang N (2021). Dynamin-dependent vesicle twist at the final stage of clathrin-mediated endocytosis. Nature Cell Biology 23(8), 859–869.

Conway JH, Burgiel H, Goodman-Strauss C (2016). The symmetries of things. CRC Press.

Cui L, Li H, Xi Y, Hu Q, Liu H, Fan J, Xiang Y, Zhang X, Shui W, Lai Y (2022). Vesicle trafficking and vesicle fusion: mechanisms, biological functions, and their implications for potential disease therapy. Molecular Biomedicine 3(1), 29.

Dacks JB, Field MC (2007). Evolution of the eukaryotic membrane-trafficking system: origin, tempo and mode. Journal of cell science 120(17), 2977–2985.

Dacks JB, Field MC (2018). Evolutionary origins and specialisation of membrane transport. Current opinion in cell biology 53, 70–76.

Danpure CJ (1997). Variable peroxisomal and mitochondrial targeting of alanine: Glyoxylate aminotransferase in mammalian evolution and disease. BioEssays 19(4), 317–326. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bies.950190409.

De Duve C (2005). Singularities: landmarks on the pathways of life. Cambridge University Press.

Dubuke ML, Munson M (2016). The Secret Life of Tethers: The Role of Tethering Factors in SNARE Complex Regulation. Frontiers in Cell and Developmental Biology 4, 42.

Embley TM, Martin W (2006). Eukaryotic evolution, changes and challenges. Nature 440(7084), 623–630.

Eme L, Tamarit D, Caceres EF, Stairs CW, De Anda V, Schön ME, Seitz KW, Dombrowski N, Lewis WH, Homa F, Saw JH, Lombard J, Nunoura T, Li WJ, Hua ZS, Chen LX, Banfield JF, John ES, Reysenbach AL, Stott MB, Schramm A, Kjeldsen KU, Teske AP, Baker BJ, Ettema TJG (2023). Inference and reconstruction of the heimdallarchaeial ancestry of eukaryotes. Nature 618(7967), 992–999. Publisher: Nature Publishing Group.

Faini M, Beck R, Wieland FT, Briggs JAG (2013). Vesicle coats: structure, function, and general principles of assembly. Trends in Cell Biology 23(6), 279–288.

Field MC, Dacks JB (2009a). First and last ancestors: reconstructing evolution of the endomembrane system with escrts, vesicle coat proteins, and nuclear pore complexes. Current opinion in cell biology 21(1), 4–13.

Field MC, Dacks JB (2009b). First and last ancestors: reconstructing evolution of the endomembrane system with ESCRTs, vesicle coat proteins, and nuclear pore complexes. Current Opinion in Cell Biology 21(1), 4–13.

Fritz-Laylin LK, Prochnik SE, Ginger ML, Dacks JB, Carpenter ML, Field MC, Kuo A, Paredez A, Chapman J, Pham J, *et al.* (2010). The genome of naegleria gruberi illuminates early eukaryotic versatility. Cell 140(5), 631–642.

Gabaldón T, Koonin EV (2013). Functional and evolutionary implications of gene orthology. Nature Reviews Genetics 14(5), 360–366. Publisher: Nature Publishing Group.

Gabaldón T, Pittis AA (2015). Origin and evolution of metabolic sub-cellular compartmentalization in eukaryotes. Biochimie 119, 262–268.

Gabaldón T, Snel B, Zimmeren Fv, Hemrika W, Tabak H, Huynen MA (2006). Origin and evolution of the peroxisomal proteome. Biology Direct 1(1), 8.

Gerst JE (2003). SNARE regulators: matchmakers and matchbreakers. Biochimica Et Biophysica Acta 1641(2-3), 99–110.

Han J, Pluhackova K, Böckmann RA (2017). The Multifaceted Role of SNARE Proteins in Membrane Fusion. Frontiers in Physiology 8, 5.

Hariri H, Bhattacharya N, Johnson K, Noble AJ, Stagg SM (2014). Insights into the mechanisms of membrane curvature and vesicle scission by the small GTPase Sar1 in the early secretory pathway. Journal of Molecular Biology 426(22), 3811–3826.

Hata Y, Slaughter CA, Südhof TC (1993). Synaptic vesicle fusion complex contains unc-18 homologue bound to syntaxin. Nature 366(6453), 347–351.

Hetz C, Zhang K, Kaufman RJ (2020). Mechanisms, regulation and functions of the unfolded protein response. Nature Reviews Molecular Cell Biology 21(8), 421–438. Number: 8 Publisher: Nature Publishing Group.

Hoffmann K, Kunze RA (1971). Linear algebra. Prentice-Hall New Jersey.

Homma Y, Hiragi S, Fukuda M (2021). Rab family of small GTPases: an updated view on their regulation and functions. The FEBS journal 288(1), 36–55.

Hooke R (2010). Robert hooke's micrographia of 1665 and 1667. JR Coll Physicians Edinb 40, 374–6.

Hutagalung AH, Novick PJ (2011). Role of Rab GTPases in membrane traffic and cell physiology. Physiological Reviews 91(1), 119–149.

Imachi H, Nobu MK, Nakahara N, Morono Y, Ogawara M, Takaki Y, Takano Y, Uematsu K, Ikuta T, Ito M, *et al.* (2020). Isolation of an archaeon at the prokaryote–eukaryote interface. Nature 577(7791), 519–525.

Juric Z, Siljak H (2011). A new formula for the number of combinations and permutations of multisets. Appl Math Sci 5(18), 875–881.

Kaiser CA, Schekman R (1990). Distinct sets of sec genes govern transport vesicle formation and fusion early in the secretory pathway. Cell 61(4), 723–733.

Kirchhausen T, Owen D, Harrison SC (2014). Molecular structure, function, and dynamics of clathrin-mediated membrane traffic. Cold Spring Harbor perspectives in biology 6(5), a016725.

Kjos I, Vestre K, Guadagno NA, Borg Distefano M, Progida C (2018). Rab and Arf proteins at the crossroad between membrane transport and cytoskeleton dynamics. Biochimica Et Biophysica Acta Molecular Cell Research 1865(10), 1397–1409.

Kleinberg J, Tardos E (2006). Algorithm design. Pearson Education India.

Koike S, Jahn R (2019). SNAREs define targeting specificity of trafficking vesicles by combinatorial interaction with tethering factors. Nature Communications 10(1), 1608.

Konheim AG (2010). Hashing in computer science: Fifty years of slicing and dicing. John Wiley & Sons.

Koonin EV (2010). The origin and early evolution of eukaryotes in the light of phylogenomics. Genome biology 11, 1–12.

Koonin EV (2015). Origin of eukaryotes from within archaea, archaeal eukaryome and bursts of gene gain: eukaryogenesis just made easier? Philosophical Transactions of the Royal Society B: Biological Sciences 370(1678), 20140333. Publisher: Royal Society.

Koumandou VL, Dacks JB, Coulson RM, Field MC (2007). Control systems for membrane fusion in the ancestral eukaryote; evolution of tethering complexes and sm proteins. BMC evolutionary biology 7, 1–17.

Koumandou VL, Wickstead B, Ginger ML, Van Der Giezen M, Dacks JB, Field MC (2013). Molecular paleontology and complexity in the last eukaryotic common ancestor. Critical reviews in biochemistry and molecular biology 48(4), 373–396.

Lamber EP, Siedenburg AC, Barr FA (2019). Rab regulation by GEFs and GAPs during membrane traffic. Current Opinion in Cell Biology 59, 34–39.

Lambert J (2019). Scientists glimpse oddball microbe that could help explain rise of complex life. Nature 572(7769), 294–295.

Lester L, Meade A, Pagel M (2006). The slow road to the eukaryotic genome. BioEssays 28(1), 57–64.

Mani S, Krishnan K, Thattai M (2022). Graph-theoretic constraints on vesicle traffic networks. Journal of Biosciences 47(1), 11.

Mani S, Thattai M (2016). Wine glasses and hourglasses: Non-adaptive complexity of vesicle traffic in microbial eukaryotes. Molecular and Biochemical Parasitology 209(1), 58–63.

Marijuán PC, del Moral R, Navarro J (2013). On eukaryotic intelligence: signaling system's guidance in the evolution of multicellular organization. Biosystems 114(1), 8–24.

Martijn J, Ettema T (2013). From archaeon to eukaryote: the evolutionary dark ages of the eukaryotic cell. Biochemical Society Transactions 41(1), 451–457.

Martin GE (2012). Transformation geometry: An introduction to symmetry. Springer Science & Business Media.

Martin W (2010). Evolutionary origins of metabolic compartmentalization in eukaryotes. Philosophical Transactions of the Royal Society B: Biological Sciences 365(1541), 847–855. Publisher: Royal Society.

Martin W, Koonin EV (2006). Introns and the origin of nucleus–cytosol compartmentalization. Nature 440(7080), 41–45.

Martin WF, Garg S, Zimorski V (2015). Endosymbiotic theories for eukaryote origin. Philosophical Transactions of the Royal Society B: Biological Sciences 370(1678), 20140330.

Maurer WD, Lewis TG (1975). Hash table methods. ACM Computing Surveys (CSUR) 7(1), 5–19.

McInerney JO, Martin WF, Koonin EV, Allen JF, Galperin MY, Lane N, Archibald JM, Embley TM (2011). Planctomycetes and eukaryotes: a case of analogy not homology. Bioessays 33(11), 810–817.

More K, Klinger CM, Barlow LD, Dacks JB (2020). Evolution and natural history of membrane trafficking in eukaryotes. Current Biology 30(10), R553–R564.

Moreira D, López-García P (1998). Symbiosis between methanogenic archaea and δ-proteobacteria as the origin of eukaryotes: the syntrophic hypothesis. Journal of molecular evolution 47, 517–530.

Novick P, Schekman R (1979). Secretion and cell-surface growth are blocked in a temperature-sensitive mutant of saccharomyces cerevisiae. Proceedings of the national academy of sciences 76(4), 1858–1862.

Perin MS, Fried VA, Mignery GA, Jahn R, Südhof TC (1990). Phospholipid binding by a synaptic vesicle protein homologous to the regulatory region of protein kinase c. Nature 345(6272), 260–263.

Pfeffer SR (2017). Rab GTPases: master regulators that establish the secretory and endocytic pathways. Molecular Biology of the Cell 28(6), 712–715.

Purkanti R, Thattai M (2022). Genome doubling enabled the expansion of yeast vesicle traffic pathways. Scientific Reports 12(1), 11213.

Qu J, Zou T, Lin Z (2021). The Roles of the Ubiquitin–Proteasome System in the Endoplasmic Reticulum Stress Pathway. International Journal of Molecular Sciences 22(4), 1526.

Rajakumar S, Vijayakumar R, Abhishek A, Selvam GS, Nachiappan V (2020). Loss of ERAD bridging factor UBX2 modulates lipid metabolism and leads to ER stress-associated apoptosis during cadmium toxicity in Saccharomyces cerevisiae. Current Genetics 66(5), 1003–1017.

Read A, Schröder M (2021). The Unfolded Protein Response: An Overview. Biology 10(5), 384.

Riedman LA, Porter SM, Lechte MA, dos Santos A, Halverson GP (2023). Early eukaryotic microfossils of the late palaeoproterozoic limbunya group, birrindudu basin, northern australia. Papers in Palaeontology 9(6), e1538.

Sapp J (2005). The prokaryote-eukaryote dichotomy: meanings and mythology. Microbiology and molecular biology reviews 69(2), 292–305.

Sauvola CW, Littleton JT (2021). SNARE Regulatory Proteins in Synaptic Vesicle Fusion and Recycling. Frontiers in Molecular Neuroscience 14, 733138.

Schlacht A, Herman EK, Klute MJ, Field MC, Dacks JB (2014). Missing pieces of an ancient puzzle: evolution of the eukaryotic membrane-trafficking system. Cold Spring Harbor perspectives in biology 6(10), a016048.

Sharma A, Czégel D, Lachmann M, Kempes CP, Walker SI, Cronin L (2023). Assembly theory explains and quantifies selection and evolution. Nature 622(7982), 321–328.

Shukla A, Bhattacharyya A, Kuppusamy L, Srivas M, Thattai M (2017). Discovering vesicle traffic network constraints by model checking. PLOS ONE 12(7), e0180692.

Smith MH, Ploegh HL, Weissman JS (2011). Road to Ruin: Targeting Proteins for Degradation in the Endoplasmic Reticulum. Science 334(6059), 1086–1090. Publisher: American Association for the Advancement of Science.

Söllner T, Whiteheart SW, Brunner M, Erdjument-Bromage H, Geromanos S, Tempst P, Rothman JE (1993). Snap receptors implicated in vesicle targeting and fusion. Nature 362(6418), 318–324.

Stanier RY, Van Niel C (1962). The concept of a bacterium. Archiv für Mikrobiologie 42, 17–35.

Thattai M (2019). How contraction has shaped evolution. Elife 8, e52805.

Thattai M (2023). Molecular and cellular constraints on vesicle traffic evolution. Current Opinion in Cell Biology 80, 102151.

Travis SM, DAmico K, Yu IM, McMahon C, Hamid S, Ramirez-Arellano G, Jeffrey PD, Hughson FM (2020). Structural basis for the binding of SNAREs to the multisubunit tethering complex Dsl1. The Journal of Biological Chemistry 295(30), 10125–10135.

Ungermann C, Kümmel D (2019). Structure of membrane tethers and their role in fusion. Traffic (Copenhagen, Denmark) 20(7), 479–490.

Van Rossum G, Drake Jr FL (2014). The python language reference. Python Software Foundation: Wilmington, DE, USA .

Vosseberg J, van Hooff JJ, Marcet-Houben M, van Vlimmeren A, van Wijk LM, Gabaldón T, Snel B (2021). Timing the origin of eukaryotic cellular complexity with ancient duplications. Nature ecology & evolution 5(1), 92–100.

West DB, *et al.* (2001). Introduction to graph theory, volume 2. Prentice hall Upper Saddle River.

Weyl H (1952). Symmetry. Princeton University Press.

Whittaker RH (1969). New concepts of kingdoms of organisms: Evolutionary relations are better represented by new classifications than by the traditional two kingdoms. Science 163(3863), 150–160.

Zaremba-Niedzwiedzka K, Caceres EF, Saw JH, Bäckström D, Juzokaite L, Vancaester E, Seitz KW, Anantharaman K, Starnawski P, Kjeldsen KU, Stott MB, Nunoura T, Banfield JF, Schramm A, Baker BJ, Spang A, Ettema TJG (2017). Asgard archaea illuminate the origin of eukaryotic cellular complexity. Nature 541(7637), 353–358. Publisher: Nature Publishing Group.

Zhukovsky MA, Filograna A, Luini A, Corda D, Valente C (2019). Protein Amphipathic Helix Insertion: A Mechanism to Induce Membrane Fission. Frontiers in Cell and Developmental Biology 7, 291.