Golgi compartments enable controlled biomolecular assembly using sloppy enzymes

Anjali Jaiman[a] and Mukund Thattai[a,1]

[a]Simons Centre for the Study of Living Machines, National Centre for Biological Sciences, Tata Institute of Fundamental Research, Bangalore 560065, India
[1]To whom correspondence should be addressed. Email: thattai@ncbs.res.in

***Abstract*—The synthesis of eukaryotic glycans – branched sugar oligomers attached to cell-surface proteins – is organised like a factory assembly line. Specific enzymes within successive compartments of the Golgi apparatus determine where new monomer building blocks are attached to the growing oligomer. These enzymes are sloppy: they act promiscuously and stochastically, causing variability in the final oligomer products. However, this variability is tightly controlled: a given eukaryotic protein type is typically associated with a sparse, specific glycan oligomer profile. Here we use ideas from the mathematical theory of self-assembly to enumerate the enzymatic causes of oligomer variability, and show how to specifically eliminate each cause. We rigorously demonstrate why oligomer variability can be controlled only when sloppy enzymes are partitioned across multiple Golgi compartments. Finally, we derive a lower bound on the number of distinct compartments required for the controlled synthesis of any given glycan oligomer.**

self-assembly | stochastic chemistry | glycan synthesis

THE surfaces of all living cells are decorated with information-rich molecules known as glycans: branched sugar oligomers, covalently linked to proteins or lipids [1]. Glycans encode cell identity and mediate a variety of intercellular interactions. They play critical roles in development, species recognition, self-nonself discrimination, and host-pathogen co-evolution [2].

Glycan oligomers are composed of a small set of monosaccharide building block types (monomers) and disaccharide bond types (linkages in the branched glycan oligomer) [7] (Fig. 1A). Eukaryotic glycans are built by collections of glycosyltransferase (GTase) enzymes the ER and Golgi apparatus, a process known as glycosylation. GTase enzymes are chemically precise but contextually sloppy [8]: a given enzyme catalyzes a specific bond between a specific pair of monomer types, but can act promiscuously and stochastically on many oligomer types. Even a single cell with a limited set of sloppy GTase enzymes can theoretically synthesize an astronomical array of oligomeric combinations [9]. This enormous potential for variability is exemplified by prokaryotic glycans, which are typically random heteropolymers [1, Chapter 21,22] [7]. In contrast, any glycosylated protein type in a eukaryotic cell is typically associated with a small, specific set of glycan oligomers, referred to as protein's glycan profile [1, Chapter 1]. The observed sparseness of eukaryotic glycan profiles is functionally relevant: specific protein glycan profiles are associated with distinct species [4], [5], [10], distinct individuals (as with ABO blood groups [11]) and distinct cell types in an individual [12]; and altered glycan profiles are implicated a variety of disorders [13].

How are eukaryotic cells able to generate sparse, specific protein glycan profiles despite the variability caused by sloppy enzymes? Patterns of variability in natural glycan profiles appear to fall into multiple classes (Fig. 1B; Table 1; [1, Chapter 1]). Computational models of glycan synthesis suggest that each type of variability is caused by distinct enzymatic mechanisms [14]–[17]. Cells appear to control variability by permitting some of these mechanisms but eliminating others, in different cellular contexts (Fig. 1C). If we could rigorously enumerate the enzymatic mechanisms that caused glycan variability, we would understand how each mechanism could be specifically eliminated, leading to a comprehensive account of how cells control glycan synthesis.

Analogous questions arise in the field of algorithmic self-assembly, which explores how building blocks with sloppy local interactions can be programmed to assemble into a desired final structure [18]–[21]. The self-assembly "forward problem" examines how a given set of building blocks with predetermined interactions assemble into larger structures. In the so-called algorithmic limit, a self-assembly process generates a single specific final structure. The self-assembly "inverse problem" asks how the interactions between building blocks can be designed to obtain a given final structure. Glycans may be considered a natural realization of self-assembly, with monomers being building blocks whose interactions are encoded by sloppy GTase enzymes.

Here we argue that the iconic compartmental organization of the Golgi apparatus, conserved across eukaryotes [22], is a structural adaptation reflecting a functional imperative: control of glycan variability. The key idea is simple: in a single reaction compartment, sloppy enzymes generate many byproducts along with any desired oligomer product. But when appropriately partitioned across multiple Golgi compartments, the same enzymes generate fewer byproducts. We make this idea rigorous. Our results are based on formal theorems, so they apply to a large class of deterministic and stochastic models of glycan synthesis [14]–[17]. We first enumerate the types of variability caused by sloppy enzymes. We then demonstrate how each type of variability can be eliminated by separating enzymes from one another, placing them within distinct compartments. Finally we solve the glycan "inverse problem": we find the minimum number of compartments required for the specific synthesis of any given glycan oligomer.

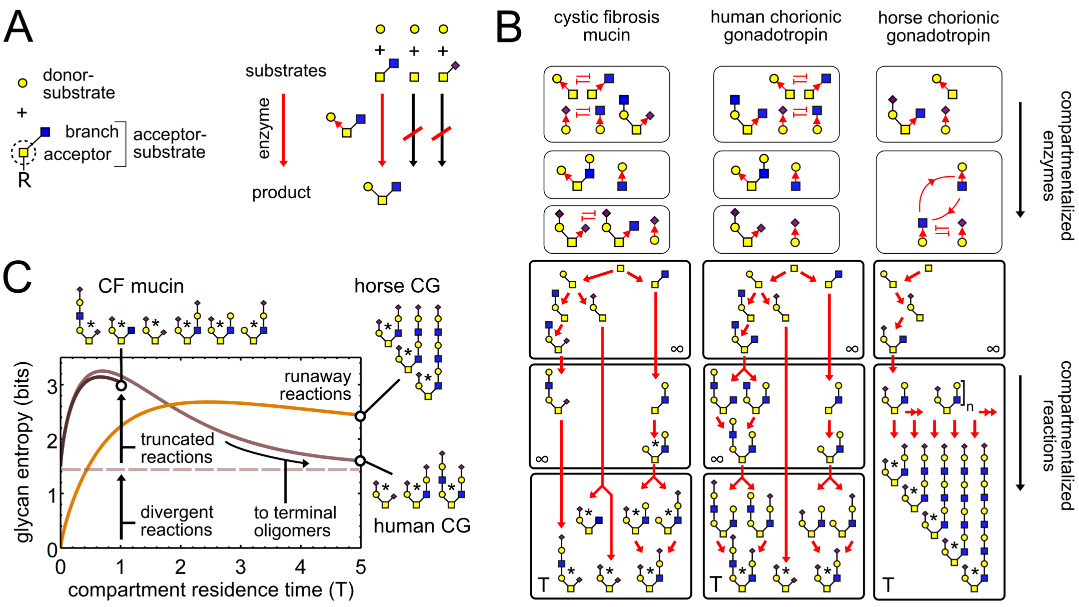

Fig. 1. Glycan synthesis and variability. ***(A)*** A GTase enzyme catalyzes a specific carbon-carbon linkage between a specific donor monomer type (the "donor-substrate") and a specific acceptor monomer type with specific branches (the "acceptor-substrate") on some arbitrary oligomer ("R"). We represent distinct monomer types in an oligomer by shapes/colors, and linkages between distinct monomer carbons by distinct bond angles [1]. We represent an idealized GTase enzyme graphically, showing its acceptor-substrate and the specific monomer-addition reaction it catalyzes (arrow from acceptor to donor at distinct angles for distinct acceptor carbons). This notation makes it easy to identify acceptor-substrates in an input oligomer, and to predict the resulting output oligomer. Enzymes are branch sensitive: they will only act if their acceptor-substrates have or lack specific branches (red arrows) and will not act otherwise (crossed black arrows). ***(B)*** Glycan synthesis in Golgi compartments. We show oligomer profiles of: respiratory mucins of a cystic fibrosis (CF) patient [3]; human chorionic gonadotropin (CG) from a cancer cell line [4]; horse chorionic gonadotropin (CG) [5] (datasets from UniCarbKB [6]; observed oligomers starred; only non-fucosylated oligomers shown). Top: By changing the compartmentalization of GTase enzymes we can generate many different glycan profiles. We refer to internal nodes of the reaction network as intermediate oligomers, and endpoints of the reaction network as terminal oligomers; all oligomers are exported from the compartment as outputs after some average residence time. Bottom: Reaction networks within each compartment, starting from the same root monomer. Red arrows show single-monomer-addition reactions. The residence time of the last compartment in each series is T: it can produce a combination of intermediate and terminal oligomers as outputs. The residence time of all other compartments is ∞: they produce only terminal oligomers as outputs. Each glycan profile highlights a distinct source of variability. "Truncated reactions": For the CF mucins, oligomers exit the last compartment at a random intermediate stage of growth, before being converted to a terminal oligomer. This gives intermediate oligomers. "Divergent reactions": for both CF mucins and human CG, acceptor blocks (blunt red arrows) arise when two enzymes compete for the same acceptor-substrate. This causes the reaction network to diverge and gives mutually exclusive oligomer fates depending on the random order of enzyme action. "Runaway reactions": for horse CG, the last compartment contains two enzymes that drive a runaway reaction (cyclic arrows). This gives oligomers with an arbitrary number of tandem repeats. ***(C)*** Effect of compartment residence time on entropy of glycan profiles. We model the reaction networks in Fig. 1B as Markov processes with constant transition probabilities per unit time (SI PROOFS: Remark 1), and calculate the probability distribution over each possible fate of the final output oligomer. The Shannon entropy of this output distribution in bits captures the variability of the glycan profile; approximately, it is the log-base-two of the number of distinct high-abundance oligomers. The entropy of the output distribution depends on that of the compartment's input distribution, on the structure of the compartment's reaction network, and on its residence time T. For horse CG the input entropy of the final compartment is zero, since there is a unique input oligomer; for human CG and CF mucins the input entropy is 1.5 (since input oligomers are in a 1:1:2 ratio due to divergent reactions in earlier compartments). At short residence times the entropy initially rises due to export of intermediate oligomers from truncated reactions. This corresponds to the CF mucin profile. At longer residence times multiple intermediate oligomers converge to a few terminal oligomers, so the entropy decreases; this corresponds to the human CG profile. The entropy stays high even at long residence times for the horse CG profile, due to synthesis of tandem repeats via runaway reactions.

These results connect a quintessential eukaryotic adaptation (intracellular compartments) to a fundamental biochemical limitation (sloppy enzymes). The idea of the Golgi apparatus as a factory assembly line is more than a metaphor, it is a mathematical necessity.

RESULTS

Universal types of variability in assembly-line reactions

For any assembly-line reaction in which oligomers are built by adding or removing one monomer at a time, there are precisely three ways variability can occur (SI PROOFS: Remark 1) [1, Chapter 1]. Each type of reaction variability corresponds to a specific output product pattern that has been observed in real glycan oligomer profiles (Fig. 1B,C). First: two identical input oligomers might exit the reaction compartment at different stages of growth; this gives a combination of both intermediate and terminal oligomers as outputs ("truncated reactions"; CF mucin in Fig. 1B). Second: The growth of some input oligomer might lead to an infinite runaway reaction; this gives oligomers with arbitrary numbers of tandem repeats as outputs ("runaway reactions"; horse CG in Fig. 1B). Third: the growth of some input oligomer might lead to a divergent reaction; this gives mutually exclusive oligomer fates as outputs ("divergent reactions"; CF mucin and human CG in Fig. 1B).

Sloppy enzymes in glycan synthesis

Though the basic types of variability are universal across all assembly-line reaction systems, their underlying causes depend on the enzymatic details. Here we analyse the highly diverse class of O-glycans, which are associated with most eukaryotic cell-surface proteins [1, Chapter 10]. The synthesis of O-glycan oligomers begins in the Golgi apparatus when a root monomer is attached to a specific serine or threonine on the substrate protein. As the growing oligomer moves through successive Golgi compartments, it encounters distinct collections of GTase enzymes within each compartment [23], [24]. During each such encounter, the GTase enzyme scans the oligomer for a site that matches a structural motif (the "acceptor-substrate") [14], [23] and attaches a single free monomer (the "donor-substrate") to that site (Fig. 1A) [25]. A given GTase enzyme always catalyzes a specific carbon-carbon linkage between a specific free donor monomer type and a specific acceptor monomer type within the acceptor-substrate (Fig. 1A; SI DEFINITIONS). However, as we now show, these enzymes are doubly sloppy: they are both promiscuous and stochastic.

Promiscuity: GTase enzymes are highly conserved across species with diverse glycans [26] (Fig. 1B,C); even in a single species the number of observed glycan oligomers far exceeds the number of available GTase enzymes [27]. By the pigeonhole principle an individual GTase enzyme must therefore be promiscuous, able to act on many distinct oligomer types. This implies that the preferred acceptor-substrate of a GTase enzyme is typically a sub-oligomer rather than the whole oligomer [27]. Here we define an acceptor-substrate to be an acceptor monomer type with specific branches (Fig. 1A; SI DEFINITIONS), and assume each idealized GTase enzyme recognizes a single specific acceptor-substrate anywhere it appears on any oligomer. This is consistent with the observation that O-glycan GTase enzymes are often branch-sensitive (Fig. 1A): some act only if the acceptor monomer is empty, others only if the acceptor monomer is already linked to a specific branch [1, Chapter 6,10] [28]. If an O-glycan GTase enzyme only reads branches up to a certain depth, we can treat it as a union of all idealized GTase enzymes whose acceptor-substrates are identical up to that depth. Our definition of acceptor-substrates thus subsumes many possible models of enzyme promiscuity [14], [16].

Stochasticity: A given Golgi compartment will contain a specific set of GTase enzymes responsible for growing an oligomer on a specific protein type. Each oligomer is exported as an output of the compartment after an average residence time T. While within the compartment, the order in which the growing oligomer stochastically encounters the available GTase enzymes is equivalent to a process of random sampling with replacement, with randomly distributed time intervals between successive encounters [29] (SI PROOFS: Remark 1). The reaction network (SI DEFINITIONS) of the compartment shows every possible oligomer growth order starting from a given input oligomer, as a result of all possible enzyme-catalyzed single-monomer-addition reactions in all possible permutations (Fig. 1B). Since O-glycan oligomers are not pruned, these reactions are irreversible [1, Chapter 6]. Within a reaction network, intermediate oligomers are those that can potentially be further extended by some available GTase enzyme, and terminal oligomers are those that cannot be further extended. Two identical input oligomers might take different paths in the reaction network as they encounter GTase enzymes in different random permutations and at different times. An oligomer might encounter the same enzyme repeatedly (if the enzyme is at high concentrations), or it might exit the reaction compartment without ever encountering some enzyme (if the enzyme is at low concentrations).

Enzymatic causes of glycan variability

We now show how the three types of variability are caused by basic enzymatic mechanisms (Table 1; Fig. 1C).

A *truncated reaction* causes intermediate oligomers to be produced as outputs. This occurs whenever the average waiting time for monomer addition (a quantity inversely proportional to enzyme concentrations) is comparable to or greater than the compartment's residence time (SI PROOFS: Remark 1). The only way all input oligomers are guaranteed to reach a terminal state (assuming no proofreading) is if enzyme concentrations are sufficiently high, or equivalently, the residence time T is sufficiently long (schematically, $T \to \infty$).

A *runaway reaction* is an infinite path in the reaction network, giving oligomers with arbitrary numbers of tandem repeats as outputs. To diagnose runaway reactions we must examine all the orders in which monomer types may be linked to one another through the action of the available GTase enzymes; this is summarized in a mathematical construct known as the compartment's linkage network (Fig. 2A; SI DEFINITIONS). A compartment contains a runaway reaction if and only if its linkage network contains one or more loops (Fig. 2A; SI PROOFS: Lemma 1). Each loop shows an order of monomer linkages that can be iterated to form tandem repeats. Certain GTase enzymes require triggers (branched acceptor-substrates that cannot be synthesized within the compartment itself; Fig. 2D; SI DEFINITIONS). Triggers effectively act as novel monomer types in the linkage network.

A *divergent reaction* is a fork in the reaction network that never reconverges, with distinct paths leading to mutually exclusive oligomer fates as outputs. To diagnose divergent reactions we must examine the acceptor-substrates of every GTase enzyme in the compartment. A fork occurs whenever distinct enzymes can act on the same oligomer to yield distinct products. If these enzymes could act in any order (for example, if they act on distinct empty acceptor monomers on the oligomer) then the reaction paths could reconverge after the fork. If a fork does not reconverge this implies that the action of one enzyme on an oligomer blocks the subsequent action of another (Fig. 2B; SI PROOFS: Lemma 2). There are only two ways for enzyme action to be blocked (SI DEFINITIONS): either two enzymes have the same acceptor-substrate (bidirectional acceptor block; Fig. 2B, left) or the acceptor-substrate

Observed product pattern		Type of reaction variability		Enzymatic cause
intermediate oligomers	⇔	truncated reaction	⇔	low concentration
tandem repeats	⇔	runaway reaction	⇔	linkage loop
mutually exclusive fates	⇔	divergent reaction	⇒	acceptor block

TABLE I
TYPES OF VARIABILITY CAUSED BY SLOPPY ENZYMES

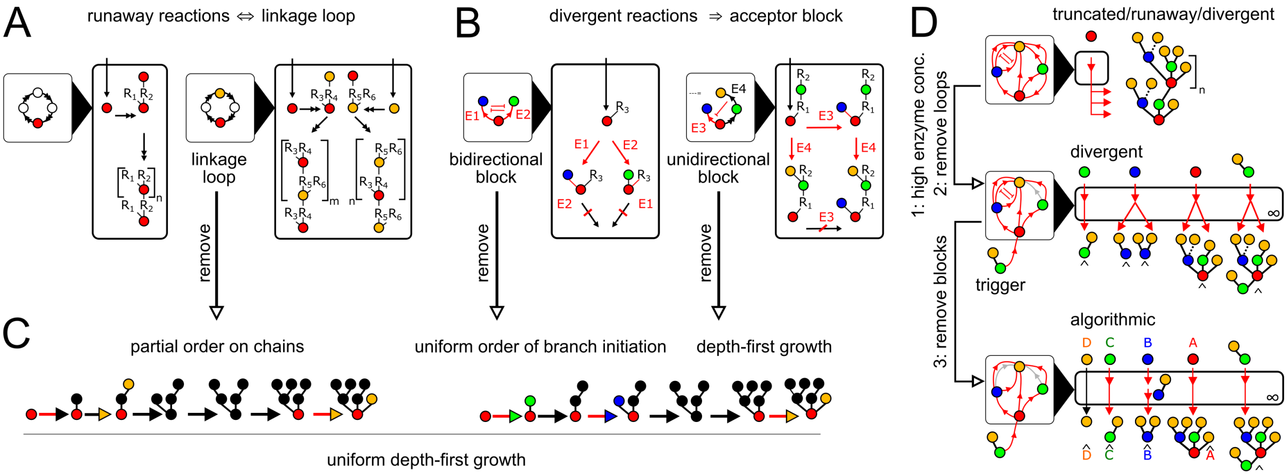

Fig. 2. Enzymatic causes of glycan variability. Colors represent monomer types (red, A; blue, B; green, C; orange, D); "R_i" represents an arbitrary oligomer. Boxes with black triangles represent the linkage network, showing all orders in which monomer types can be linked, with arrows from acceptor monomer types to donor monomer types (SI DEFINITIONS). Double arrows in linkage and reaction networks represent multiple reaction steps; blunt red arrows in the linkage network represent the action of one enzyme blocking the action of another. ***(A)*** Runaway reactions occur whenever certain steps of oligomer growth can be iterated to produce tandem repeats. Loops in the linkage network are necessary and sufficient for runaway reactions (SI PROOFS: Lemma 1). For example: monomer A is added to a branch of monomer A, ad infinitum (left); monomer D is added to a branch of monomer A, and monomer A is added to a branch of monomer D, ad infinitum (right). ***(B)*** Divergent reactions occur whenever the reaction network has a fork that can never reconverge. This occurs when the action of one enzyme blocks the subsequent action of another, so the fate of the oligomer depends on the random order of enzyme action. Acceptor blocks are necessary for divergent reactions (SI PROOFS: Lemma 2). Bidirectional acceptor block: two enzymes compete for the same acceptor-substrate. In this example E1 and E2 compete for the same acceptor-substrate; if E2 acts first it blocks E1 from acting, and vice versa. Unidirectional acceptor block: the acceptor-substrate of one enzyme is on some branch of the acceptor-substrate of another. This effect is always unidirectional. In this example E4 acts on a branch of the acceptor-substrate of E3; if E4 acts first it blocks E3 from acting. ***(C)*** Chain order is the order in which different monomer types are added from root to tip along any chain. Branch initiation order is the order in which different carbons of each acceptor-substrate are linked to donor monomers. Growth order is the order in which an oligomer grows, one monomer at a time. By removing loops in the linkage network we impose a partial order on monomer types along chains, since every allowed chain corresponds to a directed walk along the acyclic linkage network. By removing bidirectional acceptor blocks we ensure that no two enzymes have the same acceptor-substrate, imposing a strict order on branch initiation for each acceptor-substrate. By removing unidirectional acceptor blocks we ensure that all branches of all acceptor-substrates are terminal, imposing a depth-first growth order. The final result is a uniform depth-first growth order: a depth-first growth order in which identical acceptor-substrates arising at any stage of growth are always linked to the same donor monomer type at the same carbon. ***(D)*** Suppose we are given a non-algorithmic compartment. We can eliminate truncated reactions by ensuring high enzyme concentrations. We can eliminate runaway reactions by removing or disabling at least one enzyme involved in each linkage loop. We can eliminate divergent reactions by removing all but one enzyme involved in each block. The result is a block-free algorithmic compartment that converts each input oligomer to a corresponding specific output oligomer, via uniform depth-first growth (SI PROOFS: Lemma 3). Terminal fates (∧) of empty input monomers are: $\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$ (see Fig. S1 for a detailed example).

of one enzyme is on a branch of the acceptor-substrate of another (unidirectional acceptor block; Fig. 2B, right). Note that acceptor blocks need not lead to divergent reactions: the compartment might contain other enzymes that can act on the modified substrate, allowing reaction paths to reconverge after the fork (rightward implication in Table 1).

Eliminating truncated, runaway, and divergent reactions

Having identified the enzymatic causes of variability, how can we specifically eliminate each cause? Suppose we are given a compartment containing some arbitrary collection of GTase enzymes. Such a compartment might contain truncated, runaway, and divergent reactions (Fig. 2D, top). Truncated reactions can arise from insufficient concentrations of some enzyme. Runaway and divergent reactions, in contrast, are not due to any individual enzyme: they arise from interactions within sets of enzymes (Fig. 2A,B). To eliminate variability we must break up these problematic sets by removing enzymes from the reaction compartment. We only allow an enzyme to be removed if it is specifically implicated in one of the three causes from Table 1; this rules out the case of a compartment with no enzymes, which trivially generates no variability.

(1) To eliminate truncated reactions we must ensure that all enzymes are present at sufficiently high concentrations. (2) To eliminate runaway reactions we can remove or disable one enzyme in every linkage loop. (To disable an enzyme X, we must remove some other enzyme Y required for the synthesis of X's acceptor-substrate; see trigger, Fig. 2D). (3) This only leaves divergent reactions leading to a set of mutually exclusive terminal oligomers. It is a powerful result (SI PROOFS: Lemma 3) that we can always select one terminal oligomer from this set, by removing one enzyme from every bidirectional acceptor block, or the blocked enzyme from every unidirectional acceptor block. (Removing blocked enzymes in a different order may select a different terminal oligomer; and not all terminal oligomers can necessarily be selected in this way.) Each step of enzyme removal results in the elimination of an existing linkage loop or acceptor block; importantly, this does not introduce any new loops or blocks. The process is therefore guaranteed to converge, giving a compartment in which all three enzymatic causes of variability are absent. In such a compartment, the reaction network starting from any given input oligomer converges to a specific terminal oligomer, which exits as the output (this follows from the order of

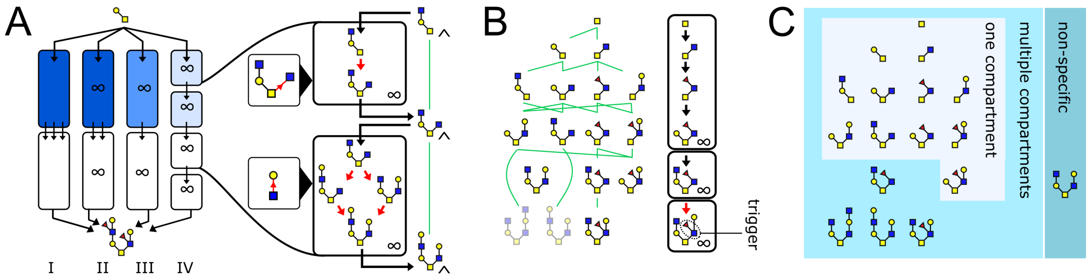

Fig. 3. The oligomer repertoire of multi-compartment systems. ***(A)*** Outline of Theorem 2 (SI PROOFS). Suppose we are given a series of compartments, not necessarily algorithmic, that convert a given input oligomer specifically to a given target oligomer. We now proceed to modify the original compartments through successive steps (the fading shade of blue represents the removal of enzymes in successive steps). We know none of the compartments contains a runaway reaction, since these would produce oligomers with arbitrary numbers of tandem repeats. By assumption every possible growth order through the reaction networks of every compartment in the series leads uniquely to the target oligomer. Step I → II: Ensure enzyme concentrations are high (equivalently, $T \to \infty$) to eliminate intermediate oligomers. The resulting growth orders go via the terminal oligomers of each compartment and still lead to the target. Step II → III: Remove all enzymes with acceptor blocks. This ensures each compartment is algorithmic while retaining at least one of the output oligomers of the corresponding original compartment, so all growth orders still lead uniquely to the target. Step III → IV: There is some growth order in which any given GTase enzyme acts successively on all its acceptor-substrates before another enzyme acts. This growth order persists when the original compartment is replaced with a series of infinite-residence-time compartments each containing a single GTase enzyme (square compartments). We represent the input-to-output map of each such compartment by a green edge. This construction shows that if an input/target pair is algorithmically achievable in a series of compartments, then it is achievable in a series of single-enzyme compartments. ***(B)*** Consider any set of oligomers; the set shown here happens to contain all sub-oligomers of all its oligomers, except for the two faded oligomers on the bottom-left. We connect two oligomers by a green edge if some infinite-residence-time single-enzyme compartment can specifically convert the top oligomer to the bottom oligomer. An input/target oligomer pair is algorithmically achievable if and only if there is a continuous path via green edges from the input to the target (as long as all sub-oligomers of the target are included in the original oligomer set; SI PROOFS: Corollary 2.1). Once such a path is found, we find the required number of compartments by decomposing any path into uniform depth-first stretches. For example, we show an oligomer that requires three compartments for its synthesis. The final compartment makes use of a trigger that was passed to it from an earlier compartment. ***(C)*** By applying this procedure we can classify any oligomer as algorithmically achievable in one compartment or multiple compartments. (The one-compartment oligomers include O-glycan "cores" [1, Chapter 10].) Certain oligomers cannot be specifically synthesized in any number of compartments, but can be synthesized as non-specific intermediates, along with other oligomers, in a non-algorithmic system.

implications in Table 1; Fig. 2D, bottom).

Enzyme promiscuity and stochasticity conspire in multiple ways to cause glycan variability. Table 1 lists three such causes: low concentrations, linkage loops, and acceptor blocks. In fact we can be sure there are no other, cryptic causes. This strong result follows from the following two facts, both demonstrated above. First: all three causes can be simultaneously and specifically eliminated. Second: once all three causes are eliminated, oligomer variability is eliminated. (Of course this does not rule out other, non-enzymatic, causes of variability such as errors in enzyme synthesis or trafficking.)

Algorithmic compartments and uniform depth-first growth

Borrowing the vocabulary of algorithmic self-assembly, we describe as "algorithmic" any compartment that produces a single specific output oligomer for every possible input. An algorithmic compartment which, furthermore, has no acceptor blocks is termed "block-free". Any algorithmic compartment can be converted to a block-free algorithmic compartment with the same input-output map by removing blocked enzymes, as described above (SI PROOFS: Lemma 3). If we could watch an individual oligomer growing within a block-free algorithmic compartment, we would find that its growth order always satisfied certain properties, which we collectively characterize as "uniform depth-first growth" (SI DEFINITIONS; SI PROOFS: Lemma 3; Fig. 2C). "Depth-first" means no new branch is initiated on an acceptor monomer until all its existing branches are fully extended ("fully" is defined by comparison to the final oligomer). "Uniform depth-first" means, for two identical acceptor monomers with identical fully-extended branches, the new branch (if any) is always initiated at the same empty carbon with same donor monomer type. These properties imply that identical acceptor-substrates arising at any stage of growth will have the same fate in the final oligomer, and the same order of branch initiation. If the input is a single monomer, uniform depth-first growth completely fixes the order of growth. However if the input is an oligomer containing many branches ready to be initiated, they can grow in any interleaved permutation without violating the uniform depth-first property. The concept of uniform depth-first growth plays a key role in the glycan "inverse problem", discussed next.

Controlled glycan assembly in multi-compartment systems

Can a given input oligomer be specifically converted to a desired target oligomer? For high yield of the target oligomer without fine-tuning or proofreading, the input-to-target reaction network must have the target as its unique terminal oligomer. If this can be done, we say the input/target pair is algorithmically achievable. As a first pass we could pick an arbitrary growth order and load a single compartment with idealized GTase enzymes corresponding to each successive single-monomer-addition reaction. This guarantees that the target oligomer will be synthesized from the given input. However various other oligomers might also be synthesized due to truncated, runaway, or divergent reactions; and the target oligomer might itself be further extended at long residence times. We show (Fig. S1; SI PROOFS):

Theorem 1. *An input/target oligomer pair is algorithmically achievable in a single compartment if and only if there is a* *uniform depth-first growth order from the input to the target.*

Some input/target oligomer pairs admit no uniform depth-first growth orders. There is always a depth-first growth order to any target oligomer from any input sub-oligomer, but it might fail to be uniform: two identical acceptor-substrates at any stage of growth could have distinct fates in the target (Fig. S1); or an acceptor-substrate extended at an intermediate stage of growth might be unextended in the target, such as at the tip of a tandem repeat (Fig. 2A). These problems signal the presence of truncated, runaway, or divergent reactions. In such cases we could try to split the enzymes over several compartments, as this might help eliminate linkage loops and acceptor blocks. In a multi-compartment series every output oligomer of each compartment type becomes an input for the next compartment type [23], [24]. We show (Fig. 3; SI PROOFS):

Theorem 2. *An input/target oligomer pair is algorithmically achievable in a series of N compartments if and only if there is a growth order from the input to the target that can be fully decomposed into N uniform depth-first stretches.*

The repertoire of algorithmically achievable input/target pairs is larger for a multi-compartment series than for a single compartment. We discuss a detailed example (Fig. S1) of an input-target pair that is not algorithmically achievable in a single compartment, because no uniform depth-first growth order exists. (Note that the uniform depth-first test only establishes that an input/target pair is algorithmically achievable; it does not mean an oligomer will necessarily grow in a uniform depth-first order, since algorithmic compartments with acceptor blocks do permit other types of growth orders.)

Even allowing for multiple compartments, not all input/target pairs are algorithmically achievable. We show (Fig. 3; SI PROOFS):

Corollary 2.1. *An input/target oligomer pair is algorithmically achievable if and only if there is a series of single-enzyme infinite-residence-time compartments that converts the input to the target as the unique final output.*

This result gives a protocol to test if an input oligomer can be specifically converted to a target oligomer in any number of compartments. Moreover, it allows the construction of an explicit set of viable growth orders via enzyme-catalyzed single-monomer-addition reactions. Among these, the growth order with the fewest number of uniform depth-first stretches reveals the smallest number of compartments in which the specific synthesis of the target from the input can be achieved. To achieve this lower bound we require enzymes that read acceptor-substrate branches to arbitrary depth; if the enzymes are less specific, more compartments may be needed.

DISCUSSION

The emergence of intracellular compartments was a watershed step in eukaryotic evolution [30]. Many hypotheses have been advanced about the adaptive function of such compartments [31]. Here we have shown that the compartmental organization of Golgi apparatus allows eukaryotic cells to synthesize sparse, specific glycan oligomer profiles from an astronomical space of possibilities, despite enzymatic sloppiness. Such controlled glycan synthesis would have been advantageous to early eukaryotes, as it enables the sophisticated intra-cellular interactions that underlie sex, cooperation and multicellularity [32].

How do Golgi compartments help reduce glycan variability? If GTase enzymes were extraordinarily specific, any glycan profile could be precisely generated in a single compartment by independently controlling the rates of every reaction [16], [33]. However, the limited number, promiscuity and stochasticity of the enzymes introduces correlations between different reaction rates, driving the synthesis of undesired oligomer byproducts. This lack of control is mainly due to problematic interactions within sets of enzymes. By compartmentalizing enzymes, cells eliminate linkage loops and acceptor blocks, and synthesize triggers in early compartments to control branching in later compartments (Fig. 2,3). These ideas could inform the design of artificial self-assembly systems, in which sticky building blocks are designed to aggregate into desired structures [18]–[21]. In particular, step-assembly strategies [34] (in which the blocks and sticky glues are added in successive steps) are analogous to glycan synthesis in successive Golgi compartments. Our results explain why larger classes of structures can be accessed using step-assembly compared to single-pot assembly.

The control of sloppy enzymes by compartmentalization within the Golgi provides a flexible way to encode cell-specific glycan profiles without requiring a large enzymatic repertoire (Fig. 1B) [12]. It allows the rewiring of glycan synthesis across different cell types and species, and enables rapid evolution in the context of host-pathogen interactions and speciation [1, Chapter 20]. By the same token, aberrant enzyme localization [35] causes changes in glycan profiles, leading to disorders [13] The precise etiology of these disorders are poorly understood, because the mechanisms by which glycan profiles encode information are largely unknown [2]. Our focus here, exploring how cells might control glycan synthesis, is only the first step toward understanding how cells actually read and use glycans.

ACKNOWLEDGEMENTS

We thank Ajit Varki for introducing us to glycans, and Arnab Bhattacharyya for helping us view glycosylation through the algorithmic lens. We thank Ramya Purkanti, Somya Mani, Mugdha Sathe, Kabir Husain and Amit Singh for useful discussions.

AUTHOR CONTRIBUTIONS

MT conceived the project. AJ and MT designed and carried out the analysis, proved the theorems, and wrote the paper.

SUPPORTING INFORMATION

See Supporting Information for definitions, proofs of theorems, and a detailed example in Fig. S1.

Declaration of interests

The authors declare no competing interests.

References

[1] A Varki, ed., *Essentials of Glycobiology, 3rd Edition*. (Cold Spring Harbor Laboratory Press), (2017).

[2] A Varki, Biological roles of glycans. *Glycobiology* **27**, 3–49 (2016).

[3] JM Lo-Guidice, et al., Sialylation and sulfation of the carbohydrate chains in respiratory mucins from a patient with cystic fibrosis. *Journal of Biological Chemistry* **269**, 18794–18813 (1994).

[4] K Harrd, et al., The carbohydrate chains of the beta subunit of human chorionic gonadotropin produced by the choriocarcinoma cell line BeWo. novel o-linked and novel bisecting-GlcNAc-containing n-linked carbohydrates. *European Journal of Biochemistry* **205**, 785–798 (1992).

[5] CH Hokke, MJH Roosenboom, JE Thomas-Oates, JP Kamerling, JFG Vliegenthart, Structure determination of the disialylated poly-(n-acetyllactosamine)-containingO-linked carbohydrate chains of equine chorionic gonadotropin. *Glycoconjugate Journal* **11**, 35–41 (1994).

[6] MP Campbell, et al., UniCarbKB: building a knowledge platform for glycoproteomics. *Nucleic Acids Research* **42**, D215–D221 (2013).

[7] A Adibekian, et al., Comparative bioinformatics analysis of the mammalian and bacterial glycomes. *Chem. Sci.* **2**, 337–344 (2011).

[8] MS .Newton, VL Arcus, ML Gerth, WM Patrick, Enzyme evolution: innovation is easy, optimization is complicated. *Current Opinion in Structural Biology* **48**, 110–116 (2018).

[9] RD Cummings, The repertoire of glycan determinants in the human glycome. *Molecular BioSystems* **5**, 1087 (2009).

[10] J Hamako, T Masui, Y Ozeki, T Mizuochi, Comparative studies of asparagine-linked sugar chains of immunoglobulin g from eleven mammalian species. *Comparative Biochemistry and Physiology* **106B**, 949954 (1993).

[11] FJ Olson, et al., Blood group a glycosyltransferase occurring as alleles with high sequence difference is transiently induced during aNippostrongylus brasiliensisParasite infection. *Journal of Biological Chemistry* **277**, 15044–15052 (2002).

[12] MB West, et al., Analysis of site-specific glycosylation of renal and hepatic -glutamyl transpeptidase from normal human tissue. *Journal of Biological Chemistry* **285**, 29511–29524 (2010).

[13] HH Freeze, BG Ng, Golgi glycosylation and human inherited diseases. *Cold Spring Harbor Perspectives in Biology* **3**, a005371–a005371 (2011).

[14] G Liu, DD Marathe, KL Matta, S Neelamegham, Systems-level modeling of cellular glycosylation reaction networks: O-linked glycan formation on natural selectin ligands. *Bioinformatics* **24**, 2740–2747 (2008).

[15] P Hossler, BC Mulukutla, WS Hu, Systems analysis of n-glycan processing in mammalian cells. *PLoS ONE* **2**, e713 (2007).

[16] PN Spahn, NE Lewis, Systems glycobiology for glycoengineering. *Current Opinion in Biotechnology* **30**, 218–224 (2014).

[17] PN Spahn, et al., A markov chain model for n-linked protein glycosylation – towards a low-parameter tool for model-driven glycoengineering. *Metabolic Engineering* **33**, 52–66 (2016).

[18] PWK Rothemund, E Winfree, The program-size complexity of self-assembled squares (extended abstract) in *Proceedings of the thirty-second annual ACM symposium on Theory of computing - STOC 00*. (ACM Press), (2000).

[19] D Soloveichik, E Winfree, Complexity of self-assembled shapes. *SIAM Journal on Computing* **36**, 1544–1569 (2007).

[20] Z Zeravcic, MP Brenner, Self-replicating colloidal clusters. *Proceedings of the National Academy of Sciences* **111**, 1748–1753 (2014).

[21] A Murugan, J Zou, MP Brenner, Undesired usage and the robust self-assembly of heterogeneous structures. *Nature Communications* **6** (2015).

[22] LD Barlow, E Nývltová, M Aguilar, J Tachezy, JB Dacks, A sophisticated, differentiated golgi in the ancestor of eukaryotes. *BMC Biology* **16** (2018).

[23] KW Moremen, M Tiemeyer, AV Nairn, Vertebrate protein glycosylation: diversity, synthesis and function. *Nature Reviews Molecular Cell Biology* **13**, 448–462 (2012).

[24] S Mani, M Thattai, Stacking the odds for golgi cisternal maturation. *eLife* **5** (2016).

[25] SI Patenaude, et al., The structural basis for specificity in human ABO(h) blood group biosynthesis. *Nature Structural Biology* **9**, 685–690 (2002).

[26] M Kaneko, S Nishihara, H Narimatsu, N Saitou, The evolutionary history of glycosyltransferase genes. *Trends in Glycoscience and Glycotechnology* **13**, 147–155 (2001).

[27] H Narimatsu, et al., GlycoGene database (GGDB) on the semantic web in *A Practical Guide to Using Glycomics Databases*. (Springer Japan), pp. 163–175 (2016).

[28] O Blixt, et al., Glycan microarrays for screening sialyltransferase specificities. *Glycoconjugate Journal* **25**, 59–68 (2007).

[29] DT Gillespie, Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* **81**, 2340–2361 (1977).

[30] R Ramadas, M Thattai, New organelles by gene duplication in a biophysical model of eukaryote endomembrane evolution. *Biophysical Journal* **104**, 2553–2563 (2013).

[31] (2018).

[32] LA Wetzel, et al., Predicted glycosyltransferases promote development and prevent spurious cell clumping in the choanoflagellate s. rosetta. *eLife* **7** (2018).

[33] L Cardelli, M Kwiatkowska, L Laurenti, Programming discrete distributions with chemical reaction networks. *Natural Computing* **17**, 131–145 (2017).

[34] D Doty, Theory of algorithmic self-assembly. *Communications of the ACM* **55**, 78 (2012).

[35] P Fisher, HL Spencer, JE Thomas-Oates, AJ Wood, D Ungar, Modelling glycan processing reveals golgi-enzyme homeostasis upon trafficking defects and cellular differentiation. *Cell Reports* **In Press** (2019).

Supporting Information for
Golgi compartments enable controlled biomolecular assembly using sloppy enzymes

Anjali Jaiman[a] and Mukund Thattai[a]

[a]Simons Centre for the Study of Living Machines, National Centre for Biological Sciences, Tata Institute of Fundamental Research, Bangalore 560065, India

Definitions

Glycan oligomer: A set of monomers linked to form a finite tree. Oligomers are grown one monomer at a time, so every oligomer or sub-oligomer (any subtree of the oligomer) has a well-defined root monomer and a well-defined direction of growth. A chain is a root-to-tip path in an oligomer. (Fig. 1A)

Acceptor monomer: A monomer in an oligomer that can be linked to to a new donor monomer at some specific carbon, through the action of some GTase enzyme. (Fig. 1A)

Empty acceptor monomer: An acceptor monomer with nothing linked to any carbon, except the carbon by which the monomer is linked to the oligomer. A donor monomer becomes an empty acceptor monomer once it is linked to the oligomer. (Fig. 1A)

Branch: The entire sub-oligomer grown on a given carbon of a given monomer. If nothing is linked to a given carbon, we say the corresponding branch is empty. New branches are initiated when empty carbons are linked to donor monomers. (Fig. 1A)

Tandem repeat: A chain in an oligomer that contains repeated instances of the same monomer type. (Fig. 1B)

Acceptor-substrate: The acceptor-substrate of a GTase enzyme is a specific acceptor monomer type with an empty branch at the carbon to be linked, and specific branches or empty branches at all other carbons. (Fig. 1A)

Bidirectional acceptor block: Two enzymes have the same acceptor-substrate, so the action of the first blocks the action of the second, and vice-versa. This is because the each enzyme can only act on the original unmodified acceptor-substrate. A special case of this is when both enzymes compete to act on the same carbon. (Fig. 2B)

Unidirectional acceptor block: The acceptor-substrate of one enzyme is on a branch of the acceptor-substrate of another so that the action of the first blocks the action of the second. This is because the second enzyme can only act on the original unmodified acceptor-substrate. (Fig. 2B)

Compartment: A reaction compartment containing a set of specified GTase enzymes, and characterized by an export rate that generates an average oligomer residence time (Fig. 1B).

Input oligomer: An oligomer or sub-oligomer that is provided as an input to a compartment. (Fig. 1B)

Output oligomer: An oligomer that exits the compartment after some residence time, at some stage of growth. (Fig. 1B)

Series of compartments: An ordered set of compartments in which every output of each compartment is passed as an input to the next compartment. An initial input is provided to the first compartment, and the last compartment produces the final outputs. (Fig. 1B)

Growth order: The order in which an oligomer is grown one monomer at a time, starting from a specific initial oligomer and leading to a specific final oligomer, through the action of successive enzymes in one or more compartments. (Fig. 1B)

Depth-first growth order: A growth order in which, as soon as a new donor monomer is linked to an empty carbon of an acceptor monomer, the preexisting branches of that acceptor monomer no longer grow. (Fig. 2C)

Uniform depth-first growth: A depth-first growth order in which identical acceptor-substrates arising at any stage of growth are always linked to the same donor monomer type at the same empty carbon. This imposes a partial order on monomer types along chains, and a uniform strict order on branch additions for each acceptor-substrate. (Fig. 2C)

Reaction network: The nodes of a reaction network represent distinct oligomers, and its directed edges represent single-monomer-addition reactions. A reaction network shows all possible growth orders in a given compartment starting from a given input oligomer. This corresponds to all possible orders in which the growing oligomer can encounter and be acted upon by the available GTase enzymes. (Fig. 1B)

Terminal oligomer: An oligomer with no outgoing edges in the given reaction network (labeled $\wedge$). (Fig. 1B,2D)

Intermediate oligomer: An oligomer with at least one outgoing edge in the given reaction network. (Fig. 1B)

Trigger: An acceptor-substrate that cannot be fully synthesized within a compartment starting from an empty acceptor monomer input. A trigger is effectively a novel monomer type such that other donor monomers can be added to it, but it cannot be added to other acceptor monomers. (Fig. 2D)

Linkage network: The nodes of a linkage network represent distinct monomer types or triggers; its directed edges represent acceptor-to-donor linkages at specific carbons. To construct a compartment's linkage network we add an arrow from one monomer type to another if the corresponding acceptor-to-donor linkage occurs on any oligomer that can be synthesized starting from any empty acceptor monomer input. We need consider only oligomers whose height is less than or equal to that of the tallest acceptor-substrate of any GTase enzyme in the compartment. An enzyme whose acceptor-substrate is a trigger will not be represented among the arrows we have added so far. We must explicitly list each trigger and draw an arrow from its acceptor monomer to the relevant donor monomer type. This gives the full linkage network of the compartment. (Fig. 2A,B,D)

Runaway reaction: A reaction network, starting from a given input oligomer, that has at least one infinite path. (Fig. 1B)

Divergent reaction: A reaction network, starting from a given input oligomer, that has at least one fork beyond which reaction paths never reconverge. (Fig. 1B)

Algorithmic reaction: A reaction network, starting from a given input oligomer, that has no runaway or divergent reactions and therefore converges to a unique terminal oligomer. (Fig. 2D)

Algorithmic compartment: A compartment that has a well-defined input-to-output map: it converts each possible input to a corresponding unique output. (Fig. 2D)

Algorithmically achievable: An input/target oligomer pair is said to be algorithmically achievable if there is a series of one or more compartments that converts the input to the target as the unique final output. (Fig. 3)

PROOFS

Remark 1. *Universal types of variability in assembly-line reactions.*

Consider a reaction compartment in which monomers are added, one at a time, to a growing oligomer. (Pruning can be represented as the addition of an "anti-monomer"). Assume monomers are in excess, and that oligomers grow independently of one another. The oligomer starts in some input state, proceeds through a series of transitions, and exits the reaction compartment after some residence time T. We model this as a continuous-time Markov process over a discrete, potentially infinite state space: each state is an oligomer configuration; each transition is an enzyme-catalyzed single-monomer-addition reaction, whose probability per unit time is proportional to the corresponding enzyme concentration. These states and transitions form an acyclic assembly-line reaction network. Given a reaction network starting from some input oligomer, define a truncated network by cutting off every reaction path at an arbitrary point (for glycans, this should correspond to an oligomer height much larger than the input oligomer height plus the number of monomer types). The terminal oligomers of the truncated network are then either terminal oligomers of the original network, or truncated oligomers (containing arbitrary numbers of added tandem repeats). For any finite residence time T the exit probability for any oligomer in the truncated network is non-zero. As $T \to \infty$ the exit probability for any intermediate oligomer tends to zero, while the exit probability for any terminal oligomer or truncated oligomer is non-zero. The input oligomer will be fully converted to a single specific final oligomer if and only if all the following conditions hold: (a) The compartment has no runaway reactions so there are no truncated oligomers (no arbitrary tandem repeats); (b) The compartment has no divergent reactions so every input leads to a unique terminal oligomer; (c) The compartment has a sufficiently long residence time (or equivalently, sufficiently high enzyme concentrations) so only this unique terminal oligomer exits as an output.

Lemma 1. *Runaway reactions $\Leftrightarrow$ linkage loop.*

Proof: Consider a reaction network starting from some input oligomer. Keep all enzymes involved in this reaction network, remove other enzymes from the compartment. Suppose the reaction network has an infinite runaway path. Each reaction corresponds to the addition of one monomer to an oligomer. Therefore the reaction network contains at least one oligomer with an arbitrarily long root-to-tip chain. Since the number of monomer types is finite, the chain must include at least two instances of the same monomer type added within the compartment. Therefore the compartment's linkage network contains a loop. Conversely suppose a compartment's linkage network contains a loop. Then there is at least one monomer type, added at some step of the reaction network, on which a branch can be grown that includes another instance of the same monomer type. This process can be iterated ad infinitum to produce arbitrary tandem repeats. Therefore the reaction network contains an infinite runaway path. (Fig. 2A) ■

Lemma 2. *Divergent reactions ⇒ acceptor block.*

Proof: Consider a reaction network starting from some input oligomer. Keep all enzymes involved in this reaction network, remove other enzymes from the compartment. Suppose the reaction network has a divergent reaction. A fork in a reaction network occurs when two enzymes can act on the same oligomer. If the enzymes could act in either order the fork could immediately reconverge. Therefore there is at least one pair of enzymes such that the action of the first enzyme blocks the action of the second. There are only two ways this can occur: both enzymes have the same acceptor-substrate (bidirectional acceptor block) or the acceptor-substrate of the first enzyme is on a branch of the acceptor-substrate of the second (unidirectional acceptor block). The converse is not true: acceptor blocks do not imply divergent reactions, since reaction paths might later reconverge due to the action of some subsequent unblocked enzyme. (Fig. 2B) ■

Lemma 3. *Every finite reaction network contains at least one growth order from the input oligomer to one of its terminal oligomers that is (a) uniform depth-first and (b) achievable in a series of single-enzyme infinite-residence-time compartments.*

Proof: Consider a finite reaction network starting from some input oligomer. Keep all enzymes involved in this reaction network, remove other enzymes from the compartment. Since there are no infinite runaway reactions, there are no linkage loops. Every possible growth order starting from the input oligomer leads to some terminal oligomer. (Any time a branch-insensitive enzyme acts at any step of any growth order, add a new branch-sensitive enzyme with the corresponding acceptor-substrate; then remove all branch-insensitive enzymes. This leaves the reaction network unchanged.) Now there is at least one growth order in which a given enzyme involved in an acceptor block is completely blocked from acting because its acceptor-substrate or a branch of its acceptor-substrate is modified by some other enzyme. This growth order is retained when we remove the blocked enzyme. By iterating this procedure we eliminate every acceptor block; note this process is not unique, since we can eliminate acceptor blocks in many different orders. The resulting compartment is algorithmic, so every growth order leads to just one of the original terminal oligomers. Every such growth order must be depth-first (there are no unidirectional acceptor blocks, so every enzyme only acts on acceptor-substrates with fully extended branches) and uniform (there are no bidirectional acceptor blocks and the final oligomer is terminal, so identical acceptor-substrates are identically extended). Among these growth orders there is at least one in which each given enzyme acts successively on every available instance of its acceptor-substrate on the oligomer, before the next enzyme acts. This growth order is retained even once we replace the compartment by a series of single-enzyme infinite-residence-time compartments. ■

Theorem 1. *An input/target oligomer pair is algorithmically achievable in a single compartment if and only if there is a uniform depth-first growth order from the input to the target.*

Proof: Suppose an input/target oligomer pair is algorithmically achievable in a single compartment. The reaction network starting from the given input oligomer must be algorithmic, with the target as its unique terminal oligomer. Therefore there is a uniform depth-first growth order from the input to the target (by Lemma 3). Conversely, suppose there is a uniform depth-first growth order from the input to the target. Each step of the growth order corresponds to the action of some GTase enzyme. We now check what happens when just these enzymes are simultaneously present in a single compartment. Since growth is depth-first, there are no unidirectional blocks. Since identical acceptor-substrates are identically extended, there are no linkage loops or bidirectional acceptor blocks. Therefore the compartment contains no runaway or divergent reactions, and the final oligomer in the growth order is terminal. In particular the reaction network starting from the given input oligomer is algorithmic, with the target as its unique terminal oligomer. At infinite residence times the compartment will produce the target as the unique output oligomer starting from the given input, so the input/target oligomer pair is algorithmically achievable. (Fig. S1) ■

Theorem 2. *An input/target oligomer pair is algorithmically achievable in a series of N compartments if and only if there is a growth order from the input to the target that can be fully decomposed into N uniform depth-first stretches.*

Proof: Suppose the input/target oligomer pair is algorithmically achievable in a series of N compartments. Every possible growth order starting from the initial input oligomer leads to the target as the unique final output oligomer. The outputs of each compartment are passed as inputs to the next compartment. Each reaction network starting from each input of each compartment is finite (since there are no arbitrary tandem repeats) and therefore has at least one uniform depth-first growth order from its input to one of its terminal oligomers (by Lemma 3). Each compartment will produce all its terminal oligomers as outputs (potentially along with intermediate oligomers at finite residence times). Therefore there is a growth order starting from the initial oligomer, passing via uniform depth-first stretches through just one terminal oligomer of each successive compartment, and producing the terminal oligomer of the last compartment as the final output. It also follows (by Lemma 3) that there is a series of single-enzyme infinite-residence-time compartments that achieves each depth-first stretch. Conversely, suppose there is a growth order from the input to the target that can be fully decomposed into N uniform depth-first stretches. Then each stretch can be achieved within a single compartment (by Theorem 1) so the input/target oligomer pair is algorithmically achievable in a series of N compartments. (Fig. 3) ■

Corollary 2.1. *An input/target oligomer pair is algorithmically achievable if and only if there is a series of single-enzyme infinite-residence-time compartments that converts the input to the target as the unique final output.*

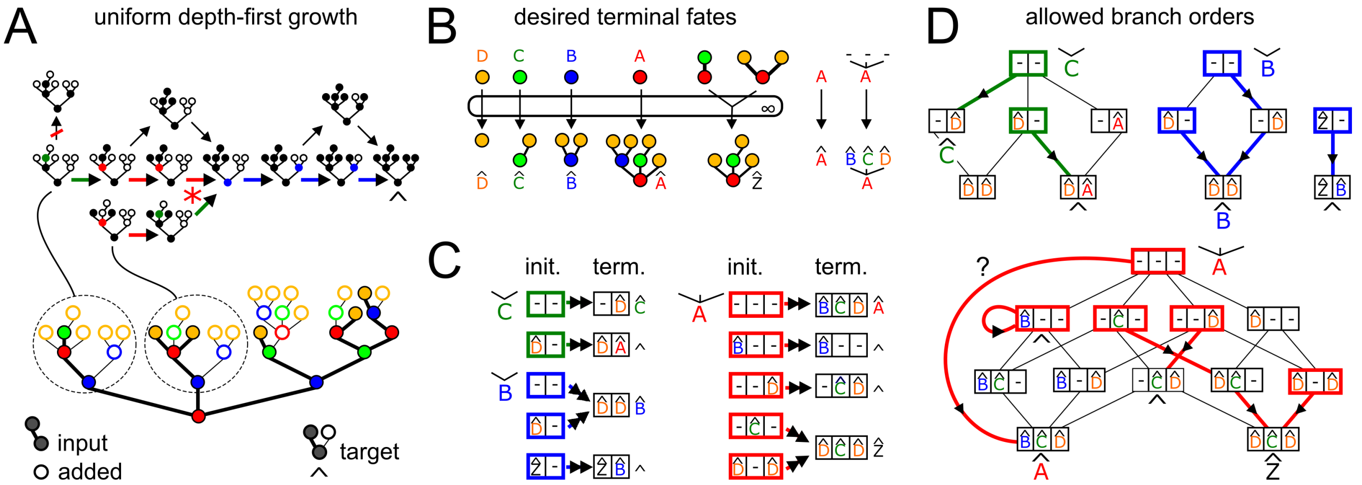

Fig. 1.

Detailed example: Testing if a desired oligomer can be specifically synthesized in a single compartment.

(A) We test if a given input oligomer (filled circles, thick lines), can be specifically converted to a desired target oligomer (full structure) by the enzyme-catalyzed addition of monomers (empty circles). This is possible if and only if there exists a uniform depth-first growth order from the input to the target (Theorem 1). Top: Examples of possible growth orders for two sub-oligomers. At each growth step an empty circle becomes filled as a new monomer is added. Depth-first growth means no new branch can be initiated on a monomer with incomplete existing branches; all reaction arrows are consistent with depth-first growth, except for the leftmost vertical arrow. Uniform growth means identical acceptor-substrates are always linked to the same donor monomer type at the same empty carbon; thus if two growth paths converge to the same acceptor-substrate (red star) they must have the same subsequent growth step and the same terminal fate in the target oligomer ($\wedge$). Alternative choices of branch order for a given acceptor-substrate correspond to alternative uniform depth-first growth orders (bifurcating arrows). The thick arrows show a depth-first growth order consistent with a particular choice of branch order; arrow colors show the monomer type of the growing acceptor-substrate.

(B) Under uniform depth-first growth, identical acceptor-substrates must have identical terminal fates in the target oligomer. The input oligomer contains many distinct acceptor-substrates, each potentially in multiple copies, and each with a desired terminal fate (labeled with a $\wedge$). A few examples of these are shown here. The terminal fates of empty monomers ($\hat{A}$, $\hat{B}$, $\hat{C}$, $\hat{D}$) are of particular interest, since these comprise the allowed fates of any newly-initiated branches. Instead of the graphical representation (left) we use a recursive representation (right) showing the branches linked to each carbon of each acceptor-monomer type. Thus oligomer $\hat{A}$ is monomer A linked to branches $\hat{B}$, $\hat{C}$, $\hat{D}$ on its three carbons; oligomer $\hat{B}$ is monomer B linked to branch $\hat{D}$ on both its carbons; oligomer $\hat{C}$ is monomer C linked to $\hat{D}$ on its right carbon; and oligomer $\hat{D}$ is simply monomer D.

(C) The set of distinct initial acceptor-substrates (bold boxes colored by acceptor monomer type) and their desired terminal fates in the target oligomer ($\wedge$). Under depth-first growth it is sufficient to consider how new branches are initiated on acceptor-substrates whose existing branches are already terminally extended. Boxes are colored according to acceptor monomer type; each slot shows any existing terminally-extended branches; empty carbons are labeled "$-$". We have not shown the trivial case of monomer type D.

(D) A uniform depth-first growth order is essentially determined by the choice of branch order. Depth-first growth is automatically enforced since we consider only acceptor-substrates with terminally-extended existing branches. Uniform growth requires that each instance of a specific acceptor-substrate has the same strict order on branch additions. We must find a single consistent branch order for each monomer type, such that each distinct acceptor-substrate achieves its desired terminal fate. The initiation of all possible branches in all possible orders is represented as a transition graph; we need consider only branches that are actually observed in the target oligomer. Each node of the graph represents distinct acceptor-substrates, using the box notation from Fig. S1C. Each directed edge represents the initiation and terminal extension of a branch on an empty carbon. Bold colored arrows show a possible choice of successive branch additions, from initial acceptor-substrates (bold boxes) to desired terminal fates ($\wedge$). There can be no bold outward arrows from terminal fates; the bold self-loop represents an acceptor-substrate that is already at its desired terminal fate. By the definition of uniform growth there can be only one bold outward arrow from each acceptor-substrate. Any acceptor-substrate not reachable from an empty monomer is a trigger. The branch order search might have a single unique solution, as with monomer type C; or multiple solutions, as with monomer type B (only one solution is shown here). There may be no solutions, since each choice of branch initiation cuts off other paths. In this example there is no set of bold arrows that simultaneously achieves all desired terminal fates for monomer type A. For the choice of arrows shown here, all desired terminal fates are achieved except for that of the empty monomer A. Therefore the given input/target pair is not algorithmically achievable in a single compartment. We leave it as an exercise for the reader to show that the same input/target pair is algorithmically achievable in two compartments.