

(51) International Patent Classification:
G06F 19/20 (2011.01) *C12Q 1/68* (2006.01)

(21) International Application Number:

PCT/US2013/051167

(22) International Filing Date:

18 July 2013 (18.07.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/672,855	18 July 2012 (18.07.2012)	US
61/783,826	14 March 2013 (14.03.2013)	US

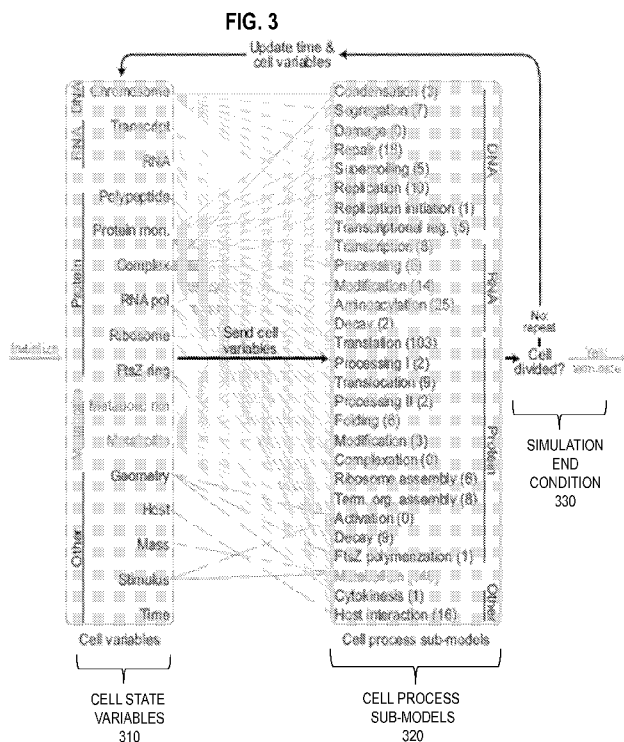
(71) Applicant: **THE BOARD OF TRUSTEES OF THE LELAND STANFORD JUNIOR UNIVERSITY** [US/US];
1705 El Camino Real, Palo Alto, California 94306-1106 (US).(72) Inventors: **COVERT, Markus**; 519 Olmsted Rd, Stanford, California 94305 (US). **KARR, Jonathan**; 2052 Ashton Ave, Menlo Park, California 94025 (US). **SANGHVI, Jayodita**; 45131 Manzanita Court, Fremont,California 94539 (US). **JACOBS, Jared**; 1756 Emerson St, Palo Alto, California 94301 (US). **MACKLIN, Derek**; 16620 Bohlman Rd, Saratoga, California 95070 (US).(74) Agents: **MOLINELLI, Eugene** et al.; P.O. Box 231, Manassas, Virginia 20108 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

[Continued on next page]

(54) Title: TECHNIQUES FOR PREDICTING PHENOTYPE FROM GENOTYPE BASED ON A WHOLE CELL COMPUTATIONAL MODEL



(57) Abstract: Techniques for simulations using a whole-cell model include retrieving cell state data that indicates chromosome data that represents over 40 %, and preferably 100%, of genes in a single cell of an organism, and which also indicates populations of gene products in multiple functional compartments. The compartments include, at least, an external conditions compartment, a membrane compartment, a cytosol compartment, and a DNA interactions compartment. Multiple sub-models are executed that each simulate, for a same time step, a different cell process selected from, at least, transcription, translation, ribosome assembly, protein translocation, and metabolism. Values for the cell state data are updated based on results from executing the sub-models; and, the steps of executing the sub-models and updating the values for the cell state data are repeated for multiple time steps. Values of the cell state data after the multiple time steps are stored on a computer-readable medium.



MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

— *as to the applicant's entitlement to claim the priority of
the earlier application (Rule 4.17(iii))*

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted
a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished
upon receipt of that report (Rule 48.2(g))*

TECHNIQUES FOR PREDICTING PHENOTYPE FROM GENOTYPE BASED ON A WHOLE CELL COMPUTATIONAL MODEL

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit of Provisional Appln. 61/672,855, filed July 18, 2012, the entire contents of which are hereby incorporated by reference as if fully set forth herein, under 35 U.S.C. §119(e). This application also claims benefit of Provisional Appln. 61/783,826, filed March 14, 2013, the entire contents of which are hereby incorporated by reference as if fully set forth herein, under 35 U.S.C. §119(e).

STATEMENT OF GOVERNMENTAL INTEREST

[0002] This invention was made with government support under contracts LM011510 and OD006413 awarded by the National Institutes of Health. The government has certain rights in the invention.

BACKGROUND OF THE INVENTION

[0003] Computer models that can account for the integrated function of every gene in a cell have the potential to revolutionize biology and medicine, as they increasingly contribute to how we understand, discover and design biological systems. For example, it would be very useful to predict the effects of adding, removing or substituting one or more genes to a base organism, without having to actually engineer the modified organism. Models of biological processes have been increasing in complexity and scope. However, previous models have not been able to include more than about 35% of the genome.

[0004] The approaches to modeling cellular physiology, based on ordinary differential equations (ODEs) (Atlas et al., 2008; Browning et al., 2004; Castellanos et al., 2004; Castellanos et al., 2007; Domach et al., 1984; Tomita et al., 1999), were limited by the difficulty in obtaining the necessary model parameters. Subsequently, alternative approaches were developed that require fewer parameters, including Boolean network modeling (Davidson et al., 2002) and constraint-based modeling (Orth et al., 2010; Thiele et al., 2009). However, the underlying assumptions of these methods do not apply to all cellular processes

and conditions, and building a whole-cell model entirely based on either method is therefore impractical.

SUMMARY OF THE INVENTION

[0005] Techniques are provided for predicting phenotype from genotype based on a whole cell computational model.

[0006] In a first set of embodiments, a method comprises retrieving on a processor cell state data that indicates chromosome data that represents over 40% of genes in a single cell of an organism, and which also indicates populations of gene products in multiple functional compartments. The functional compartments include at least an external conditions compartment, a membrane compartment, a cytosol compartment, and a DNA interactions compartment. The method includes executing on a processor multiple sub-models that each simulate, for a same time duration called a time step, a different cell process selected from a group of cell processes comprising at least transcription, translation, ribosome assembly, protein translocation, and metabolism. The method also includes updating on a processor values for the cell state data based on results from executing the plurality of sub-models; and, repeating, for multiple time steps, the steps of executing the plurality of sub-models and updating the values for the cell state data. The process further includes storing, on a computer-readable medium, values of the cell state data after the plurality of time steps.

[0007] In some embodiments, the chromosome data represents all genes in a single cell of an organism. In some embodiments, the method also includes determining a first cell phenotype, such as cell mass doubling time or cell division time, based on the values of the cell state data after the plurality of time steps. In some of these embodiments, the method also includes determining whether the first cell phenotype matches a target phenotype; and, if so, then genetically modifying an actual cell of the organism to correspond to the chromosome data.

[0008] In various embodiments, the sub-models include a variety of types of models, including models based on ordinary differential equations, Boolean networks models, constraint-based modeling, and flux mass balance models.

[0009] In other sets of embodiments, an apparatus, a system, or a computer-readable medium is configured to cause an apparatus to perform one or more steps of the above methods.

[0010] Still other aspects, features, and advantages of the invention are readily apparent from the following detailed description, simply by illustrating a number of particular embodiments and implementations, including the best mode contemplated for carrying out the invention. The invention is also capable of other and different embodiments, and its several details can be modified in various obvious respects, all without departing from the spirit and scope of the invention. Accordingly, the drawings and description are to be regarded as illustrative in nature, and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0012] FIG. 1 is a block diagram that illustrates example processes in a cell to be modeled, according to an embodiment;

[0013] FIG. 2 is a block diagram that illustrates useful compartments, according to some embodiments;

[0014] FIG. 3 is a block diagram that illustrates example simulations using cell state variables and sub-models, according to an embodiment;

[0015] FIG. 4 is a flowchart that illustrates an example method for forming a whole-cell model, according to an embodiment;

[0016] FIG. 5A through FIG. 5C are flowcharts that illustrate an example use of whole-cell model simulation, according to an embodiment;

[0017] FIG. 6A through FIG. 6C are block diagrams that illustrate example fields and arrays and initial values for representing chromosome data in the Chromosome state variable class, according to an embodiment;

[0018] FIG. 7 is a block diagram that illustrates example fields in the Transcript state variable class, according to an embodiment;

[0019] FIG. 8A is a block diagram that illustrates example stages of RNA maturation and change tracked in the RNA state variable class, according to an embodiment;

[0020] FIG. 8B is a block diagram that illustrates example fields in the RNA state variable class, according to an embodiment;

[0021] FIG. 8C is a table that lists example interactions between different cell sub-models and the RNA state variable class, according to an embodiment;

[0022] FIG. 9 is a block diagram that illustrates example fields in the Polypeptide state variable class, according to an embodiment;

[0023] FIG. 10A is a block diagram that illustrates example stages of protein monomer maturation and change tracked in the Protein Monomer state variable class, according to an embodiment;

[0024] FIG. 10B is a block diagram that illustrates example fields in the Protein Monomer state variable class, according to an embodiment;

[0025] FIG. 10C is a table that list example interactions between different cell sub-models and the Protein Monomer state variable class, according to an embodiment;

[0026] FIG. 11A is a block diagram that illustrates example fields in the Protein Complex state variable class, according to an embodiment;

[0027] FIG. 11B is a table that list example interactions between different cell sub-models and the Protein Complex state variable class, according to an embodiment;

[0028] FIG. 12 is a block diagram that illustrates example fields in the RNA Polymerase state variable class, according to an embodiment;

[0029] FIG. 13 is a block diagram that illustrates example fields in the Ribosome state variable class, according to an embodiment;

[0030] FIG. 14 is a block diagram that illustrates example fields in the Cytokinesis state variable class, according to an embodiment;

[0031] FIG. 15 is a block diagram that illustrates example fields in the Metabolic Reaction state variable class, according to an embodiment;

[0032] FIG. 16 is a block diagram that illustrates example fields in the Metabolite state variable class, according to an embodiment;

[0033] FIG. 17A is a block diagram that illustrates example cell geometry with a cell division septum region, according to an embodiment;

[0034] FIG. 17B is a block diagram that illustrates example fields in the Geometry state variable class, according to an embodiment;

[0035] FIG. 18 is a block diagram that illustrates example fields in the Host state variable class, according to an embodiment;

[0036] FIG. 19 is a block diagram that illustrates example fields in the Mass state variable class, according to an embodiment;

[0037] FIG. 20 is a block diagram that illustrates example fields in the Stimulus state variable class, according to an embodiment;

[0038] FIG. 21 is a block diagram that illustrates example fields in the Time state variable class, according to an embodiment;

[0039] FIG. 22A is a block diagram that illustrates features of DNA condensation, according to an embodiment;

[0040] FIG. 22B is a flowchart that illustrates an example DNA Condensation sub-model, according to an embodiment;

[0041] FIG. 23A is a flowchart that illustrates an example DNA Segregation sub-model, according to an embodiment;

[0042] FIG. 23B is a list showing the relationship of the DNA Segregation sub-model to cell state variables, according to an embodiment

[0043] FIG. 24A is a flowchart that illustrates an example DNA Damage sub-model, according to an embodiment;

[0044] FIG. 24B is a list of instructions that illustrates an example DNA Damage sub-model for *M. genitalium*, according to an embodiment;

[0045] FIG. 25A is a flowchart that illustrates an example DNA Repair sub-model, according to an embodiment;

[0046] FIG. 25B is a list of instructions that illustrates an example DNA Repair sub-model for *M. genitalium*, according to an embodiment;

[0047] FIG. 26A is block diagram that illustrates different regions for defining helicity on a replicating chromosome, according to an embodiment;

[0048] FIG. 26B through FIG. 26D are graphs that illustrate the effect of superhelical density on the activity of certain DNA binding proteins, according to an embodiment;

[0049] FIG. 26E and FIG. 26F are flow charts that illustrate an example method for a DNA Supercoiling sub-model, according to an embodiment;

[0050] FIG. 26G is a list of parameter values set for the DNA Supercoiling sub-model, according to an embodiment;

[0051] FIG. 27A through FIG. 27C are a block diagram that illustrates the progress of DNA replication and components of replication machinery in a cell, modeled according to an embodiment;

[0052] FIG. 27D through FIG. 27F are flowcharts that illustrate an example method for a DNA Replication sub-model, according to an embodiment;

[0053] FIG. 27G is a list of parameter values set for the DNA Replication sub-model, according to an embodiment;

[0054] FIG. 27H is a list of cell state variables and other sub-models interacting with the DNA Replication sub-model, according to an embodiment;

[0055] FIG. 27I through FIG. 27K are lists of instructions that illustrate an example DNA Replication sub-model, according to an embodiment;

[0056] FIG. 28A is a list of parameters and values for an example DNA Replication Initiation sub-model, according to an embodiment;

[0057] FIG. 28B and FIG. 28C are block diagrams that illustrate binding state cooperativity for an example DNA Replication Initiation sub-model, according to an embodiment;

[0058] FIG. 28D is a list of cell state variables and other sub-models interacting with the DNA Replication Initiation sub-model, according to an embodiment;

[0059] FIG. 29A is a flowchart that illustrates an example method for a Transcriptional Regulation sub-model, according to an embodiment;

[0060] FIG. 29B is a list of instructions that illustrates an example Transcriptional Regulation sub-model, according to an embodiment;

[0061] FIG. 30A and FIG. 30B are flowcharts that illustrate an example method for a Transcription sub-model, according to an embodiment;

[0062] FIG. 30C is a list of cell state variables and other sub-models interacting with the Transcription sub-model, according to an embodiment;

[0063] FIG. 30D is a list of parameters and values for a Transcription sub-model, according to an embodiment;

[0064] FIG. 31A is a flowchart that illustrates an example method for a RNA Processing sub-model, according to an embodiment;

[0065] FIG. 31B is a list of instructions that illustrates an example RNA Processing sub-model, according to an embodiment;

[0066] FIG. 32A is a flowchart that illustrates an example method for a RNA Modification sub-model, according to an embodiment;

[0067] FIG. 32B is a list of instructions that illustrates an example RNA Modification sub-model, according to an embodiment;

[0068] FIG. 33A is a flowchart that illustrates an example method for an RNA Aminoacylation sub-model, according to an embodiment;

[0069] FIG. 33B is a list of parameters for an RNA Aminoacylation sub-model, according to an embodiment;

[0070] FIG. 34A is a flowchart that illustrates an example method for an RNA Decay sub-model, according to an embodiment;

[0071] FIG. 34B is a list of cell state variables and other sub-models interacting with the RNA Decay sub-model, according to an embodiment;

[0072] FIG. 34C is a list of parameters and values for an RNA Decay sub-model, according to an embodiment;

[0073] FIG. 35A and FIG. 35B are flowcharts that illustrate an example method for a Translation sub-model, according to an embodiment;

[0074] FIG. 35C is a list of cell state variables and other sub-models interacting with the Translation sub-model, according to an embodiment;

[0075] FIG. 35D is a list of parameters and values for a Translation sub-model, according to an embodiment;

[0076] FIG. 35E is a list of instructions that illustrates an example method for initializing ribosomes for a Translation sub-model, according to an embodiment;

[0077] FIG. 36A is a flowchart that illustrates an example method for a Protein Processing I sub-model, according to an embodiment;

[0078] FIG. 36B is a list of instructions that illustrates an example Protein Processing I sub-model, according to an embodiment;

[0079] FIG. 37A is a flowchart that illustrates an example method for a Protein Translocation sub-model, according to an embodiment;

[0080] FIG. 37B is a list of instructions that illustrates an example Protein Translocation sub-model, according to an embodiment;

[0081] FIG. 38A is a flowchart that illustrates an example method for a Protein Processing II sub-model, according to an embodiment;

[0082] FIG. 38B is a list of instructions that illustrates an example Protein Processing II sub-model, according to an embodiment;

[0083] FIG. 39A is a flowchart that illustrates an example method for a Protein Folding sub-model, according to an embodiment;

[0084] FIG. 39B is a list of instructions that illustrates an example Protein Folding sub-model, according to an embodiment;

[0085] FIG. 40A is a flowchart that illustrates an example method for a Protein Modification sub-model, according to an embodiment;

[0086] FIG. 40B is a list of instructions that illustrates an example Protein Modification sub-model, according to an embodiment;

[0087] FIG. 41A is a flowchart that illustrates an example method for a Protein Complexation sub-model, according to an embodiment;

[0088] FIG. 41B is a list of instructions that illustrates an example Protein Complexation sub-model, according to an embodiment;

[0089] FIG. 42A is a flowchart that illustrates an example method for a Ribosome Assembly sub-model, according to an embodiment;

[0090] FIG. 42B is a list of instructions that illustrates an example Ribosome Assembly sub-model, according to an embodiment;

[0091] FIG. 43A is a flowchart that illustrates an example method for a Organelle Assembly sub-model, according to an embodiment;

[0092] FIG. 43B is a block diagram of hierarchical protein organization of a Terminal Organelle, according to an embodiment;

[0093] FIG. 43C is a list of instructions that illustrates an example Terminal Organelle Assembly sub-model, according to an embodiment;

[0094] FIG. 44A is a flowchart that illustrates an example method for a Protein Activation sub-model, according to an embodiment;

[0095] FIG. 44B is a list of instructions that illustrates an example Protein Activation sub-model, according to an embodiment;

[0096] FIG. 45A through FIG. 45D are flowcharts that illustrate an example method for a Protein Decay sub-model, according to an embodiment;

[0097] FIG. 45E through FIG. 45G are lists of instructions that illustrate an example Protein Decay sub-model, according to an embodiment;

[0098] FIG. 46 is a list of parameters and values for an FtsZ polymerization sub-model useful for modeling cell division, according to an embodiment;

[0099] FIG. 47A and FIG. 47B are block diagrams that illustrate a metabolic network of *M. genitalium*, according to an embodiment;

[0100] FIG. 47C and FIG. 47D are block diagrams that illustrate an integrated flux balance analysis for *M. genitalium*, according to an embodiment;

[0101] FIG. 47E and FIG. 47F are lists of instructions that illustrate an example Metabolism sub-model, according to an embodiment;

[0102] FIG. 48A is a list of parameters and values for a Cytokinesis sub-model useful for modeling cell division, according to an embodiment;

[0103] FIG. 48B is a block diagram that illustrates an example cycle of filament bending during cytokinesis in the Cytokinesis sub-model, according to an embodiment;

[0104] FIG. 48C is a list of state variables and sub-models that interact with the Cytokinesis sub-model, according to an embodiment;

[0105] FIG. 49 is a list of instructions that illustrates an example Host Interaction sub-model, according to an embodiment;

[0106] FIG. 50 is a graph that shows example comparisons between measured and predicted growth rates for wild-type and 12 single-gene disrupted strains, according to an embodiment; and

[0107] FIG. 51 is a block diagram that illustrates a computer system 5100 upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION

[0108] Techniques are described for a producing and using a whole cell computational model capable of predicting phenotype from genotype. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0109] As used herein, a whole-cell model is a processor-based method that generates populations of gene products in multiple functional compartments in each of multiple time steps from over 40% of genes in all chromosomes of a cell, preferably over 50% of the genes, more preferably over 90% of the genes, and, in some embodiments, from every gene in all chromosomes. Such models are used to predict the evolution of an individual cell with a particular genotype through cell division, or the variability among a population of cells with the same genotype, or a range of genotypes, including a tissue constructed from a population of such cells, for time scales in a range from one time step to a duration corresponding to many thousands of cell mass doublings and cell divisions.

[0110] Some embodiments of the invention are described below in the context of modeling *Mycoplasma genitalium* (herein after “*M. genitalium*”), a human urogenital prokaryotic cell parasite whose genome contains 525 genes. However, the invention is not limited to this context. In other embodiments, a whole-cell model is formed and used for other prokaryotic cells (cells without a nucleus in which genetic material is free in the cytoplasm) as well as eukaryotic cells (that carry out more internal chemical mechanisms and so utilize a nuclear membrane to separate DNA processes), using one or more techniques described herein. In still other embodiments, a population of cells, of all the same or of multiple different cell types, is simulated by rerunning the whole-cell model multiple times, either in parallel at each time step, or separately. Such population simulations may be used, in various embodiments, to determine variability within the population or interactions between cells for

cells of the same tissue or of the same organism, or for cell populations of mixed organisms, or some combination.

1. Definitions

[0111] The terms used herein have the meanings in the following table.

Table 1. DEFINITIONS

allele	One of two or more forms of a gene or a genetic locus (generally a group of genes) on at least two chromosomes
ADP	Adenosine Di-Phosphate: a molecule that results from hydrolysis of ATP to release energy
ATP	Adenosine Tri-Phosphate: a molecule used by cells to store energy
BER	base excision repair
cell compartment	A portion of cell in which a particular set of processes are performed and populated by molecules that serve as reactants and products of those processes.
chromosome	A single piece of coiled DNA containing many genes, regulatory elements and other nucleotide sequences also containing DNA-bound proteins, which serve to package the DNA and control its functions
DDR	direct damage reversal
DNA	Deoxyribonucleic acid (DNA) is a usually double-stranded long molecule composed of repeating chemical units known as “nucleotides” or “bases.” There are four bases: adenine, thymine, cytosine, and guanine, represented by the letters A, T, C and G, respectively. Adenine on one strand of DNA always binds to thymine on the other strand of DNA, and guanine on one strand always binds to cytosine on the other strand; such bonds are called base pairs. Any order of A, T, C and G is allowed on one strand, and that order determines the complementary order on the other strand. When occurring in chromosomes, the actual order encodes other shorter molecules, such as proteins, used to build and control all living organisms and thus determines the function of that portion of the DNA

	molecule.
DNA-binding proteins	Proteins that are composed of DNA-binding domains that have a specific or general affinity for either single or double stranded DNA. DNA-binding proteins include transcription factors which modulate the process of transcription, various polymerases, nucleases which cleave DNA molecules, and histones which are involved in chromosome packaging and transcription in the cell nucleus.
DSBR	double-strand break repair
FBA	flux-balance analysis
GAM	growth-associated maintenance energy
GDP	Guanosine diphosphate, a molecule that results from hydrolysis of GTP to release energy.
Gene	A segment of DNA involved in producing a functional RNA or polypeptide chain. Specifically, a gene includes, without limitation, regions preceding and following the coding region, such as the promoter and 3'-untranslated region, respectively, as well as intervening sequences (introns) between individual coding segments (exons).
gene product	RNA or protein coded for in a gene
Genome	The entirety of an organism's hereditary information, encoded either in DNA or, for many types of viruses, in RNA. The genome includes both the genes and the non-coding sequences of the DNA/RNA.
genotype	The genetic composition of an individual organism.
GG	global genomic
GTP	Guanosine-5'-triphosphate, a molecule used by cells to store energy.
HR	homologous recombination
logD	Log of distribution coefficient that is the ratio of the sum of the concentrations of all forms of the compound (ionized plus un-ionized)

	in each of the two phases.
logP	Logarithm of the ratio of the concentrations of an un-ionized solute in two solvents: a measure of lipophilicity.
marker	A gene or unique DNA sequence with a known location on a chromosome that can be used to determine the chromosome location of an arbitrary sequenced portion of DNA.
MOMA	minimization of metabolic adjustment
mRNA	messenger RNA
NER	nucleotide excision repair
NF- κ B	nuclear factor kappa-light-chain-enhancer of activated B cells
nucleic acid	A molecule comprising a sequence of one or more repeating chemical units known as “nucleotides” or “bases” of “base pairs” (bp). There are four bases in deoxyribonucleic acid (DNA): adenine, thymine, cytosine, and guanine, represented by the letters A, T, C and G, respectively. In ribonucleic acid (RNA) the base uracil (U) replaces the base thymine (T).
ODE	ordinary differential equation
(p)ppGpp	guanosine pentaphosphate or tetraphosphate
phenotype	An individual organism’s observable characteristics or traits, such as its morphology, development, biochemical or physiological properties, phenology, behavior, and products of behavior.
pI	Isoelectric point that indicates the pH at which a particular molecule or surface carries no net electrical charge.
protein	A generic term referring to molecules made up of a chain of amino acids and including native protein, fragments, peptides, or analogs of a polypeptide sequence. Synonym used herein includes “polypeptide.” Hence, native protein, fragments, and analogs are species of a polypeptide genus.

R/M	restriction/modification
RNA	Ribonucleic acid (RNA) comprises a single stranded chain of nucleotides in which uracil (U) replaces thymine (T).
rRNA	ribosomal RNA
sample	Includes any biological specimen obtained from a subject.
sequencing	Determining the order, or sequence, of bases on one strand of DNA or RNA.
SMC	structural maintenance of chromosome protein
SNP	Single-nucleotide polymorphism, a DNA sequence variation occurring when a single nucleotide in the genome (or other shared sequence) differs between members of a biological species or paired chromosomes in an individual. The vast majority of polymorphic sites are bi-allelic, meaning that no more than two possible distinct alleles can be observed at these sites. As used herein, a 0 indicates one nucleotide at the position of the SNP and a 1 indicates a different nucleotide.
SP-4	<i>Spiroplasma</i> medium #4
sRNA	small non-coding RNA
SRP	signal recognition particle
SSB	single-stranded binding protein
subject	An organism that is an object of a method or material, including mammals, e.g., humans, dogs, cows, horses, kangaroos, pigs, sheep, goats, cats, mice, rabbits, rats, and transgenic non-human animals as well as plants and protists. Synonyms used herein include “patient” and “animal.”
TLR	toll-like receptor
treating	Taking steps to obtain beneficial or desired results, including clinical results, such as alleviating or ameliorating one or more symptoms of a

	disease; diminishing the extent of disease; delaying or slowing disease progression; ameliorating and palliating or stabilizing a metric (statistic) of disease. "Treatment" refers to the steps taken.
tRNA	transfer RNA
TSS	transcription start site
UV	ultraviolet
whole-cell model (also, whole cell computational model)	A computational process that generates populations of gene products in multiple functional compartments in each of multiple time steps from over 40% of genes in all chromosomes of a cell, preferably over 50% of the genes, more preferably over 90% of the genes, and when possible every gene in all chromosomes.

1. Overview

[0112] A useful goal is to predict the complex phenotypes of individual cells in terms of individual molecules and their interactions. The primary challenges to building a unified, whole-cell model of a single cell are threefold: (1) *complexity*, processes relevant to cellular behavior are diverse and span a wide range of length and time scales, (2) *heterogeneity*, cellular networks have heterogeneous mathematical structures and are typically investigated using heterogeneous experimental methods, and (3) *sparsity*, little quantitative data rigorously describing single cell physiology is available. The flexibility of a hybrid model best allowed navigating these challenges.

[0113] Biological systems are herein viewed as modular at certain time scales.

Consequently the cell is modeled herein by (1) dividing cellular processes into separate sub-models corresponding to separate functional processes, (2) independently running each sub-model on a short time scale, and (3) integrating longer time scales by using de-conflicted sub-model outputs at the end of one time step as input to one or more sub-models for the next time step. The sub-models span six areas of cell biology: (1) transport and metabolism, (2) DNA replication and maintenance, (3) RNA synthesis and maturation, (4) protein synthesis

and maturation and (6) interaction, if any. In some embodiments, the protein synthesis and maturation sub-models include an additional area for cell-type specific functions, such as cell-specific organelle functions. For example, *M. genitalium* performs certain functions in an attachment organelle for adhering to the urogenital membrane wall of its human host. Other cells have other organelles such as Golgi bodies and endoplasmic reticulum. In various embodiments, the sub-models include a variety of types of models, including models based on ordinary differential equations, Boolean networks models, constraint-based modeling, and flux mass balance models.

[0114] It is a challenge to identify appropriate sub-models for each process and to determine the appropriate time scale and sub-model inputs and outputs for a whole-cell model. FIG. 1 is a block diagram that illustrates example processes in a cell to be modeled, according to an embodiment. FIG. 1 depicts 28 sub-models grouped by category as metabolic 140, RNA 120, protein 130, and DNA 110—in the context of a single cell. Sub-models are connected through common metabolites, RNA, protein, and the chromosome which are depicted as solid filled arrows. It was discovered that much of the cell phenotype, including growth and cell division is described with these 28 sub models listed in Table 2. These sub-models are described in more detail below with reference to FIGs. 22 to 48. In some embodiments, some sub-models are excluded. For example, if cell division is not modeled, sub-models directed to replication and cell division are omitted. In other embodiments, other sub-models are added, such as sub-models for producing other organelles.

Table 2. List of sub-models for whole-cell model.

number	Area of biology	Sub-model name	processes
1	DNA	Condensation	Chromosome folding
2	DNA	Segregation	Replicated chromosomes moving to opposite poles in cell
3	DNA	Damage	Change in DNA base due to external factors
4	DNA	Repair	Remediate wrong, foreign or damaged DNA bases

5	DNA	Supercoiling	Effect of enzymes that over coil DNA
6	DNA	Replication	Generating a copy of a chromosome
7	DNA	Replication Initiation	Establishing conditions to initiate replication
8	RNA	Transcriptional Reg	Controlling rate of transcription of one or more genes
9	RNA	Transcription	Generating an mRNA coded by DNA
10	RNA	RNA Processing	Cleavage of completed mRNA to remove introns
11	RNA	RNA Modification	Modifications of tRNA to reduce use of redundant codons
12	RNA	Aminoacylation	Binding an amino acid to a tRNA
13	RNA	RNA Decay	Breaking an RNA molecule into its constituent bases
14	protein	Translation	Growing a polypeptide to form a protein coded by mRNA
15	protein	Processing I	N-terminal formylmethionine deformylation and N-terminal methionine cleavage
16	protein	Translocation	Moving proteins to and through membrane
17	protein	Processing II	Lipoprotein diacylglycerol adduction and lipoprotein and secreted protein signal peptide cleavage for protein in a membrane
18	protein	Folding	Chaperone mediated folding of protein to functional form
19	protein	Protein Modification	Covalent bonding with

			small molecules to complete mature protein
20	protein	Complexation	Assembling a gene coded protein with a different gene coded protein or RNA molecule not covered by another sub-model
21	protein	Ribosome Assembly	Assembling a ribosome from proteins and RNA molecules
22	protein	Organelle Assembly	Assembling proteins and complexes specific to a special organelle, such as terminal organelle of <i>M.genitalium</i> .
23	protein	Protein Activation	Causing a protein to be active for performing a function based on external conditions falling in a particular range
24	protein	Protein Decay	Breaking a protein down into separate amino acids
25	protein	FtsZ Polymerization	Assembling proteins and complexes (such as FtsZ) specific to cell division
26	transport/metabolism	Metabolism	Importing extracellular nutrients and converting them into macromolecule building blocks
27	transport/metabolism	Cytokinesis	Pinching of the cell membrane in a certain location until the membrane is separated and division is complete
28	host interaction	Host Interaction	Determining external conditions caused by host, if any, in response

			to proteins and complexes produced in or secreted through the cell membrane
--	--	--	---

[0115] It was also recognized that the above processes act in groups that involve different portions of the cell, herein called compartments. A cell compartment is a portion of cell in which a particular set of processes are performed and populated by molecules that serve as reactants and products of those processes. The compartment might correspond to one or more locations in a cell, such as the membrane. It was found that useful whole-cell modeling could be performed with four or more compartments representing, at least, external conditions, cell membrane, cytosol and DNA interactions. Additional compartments are included in some embodiments, such as one or more compartments for special organelles and a nucleus membrane compartment for cells with a nucleus. In some embodiments, there are sub-compartments, such as sub-compartments for organelle membranes in the special organelle compartment or a compartment for lipoprotein interactions within the cell membrane compartment. FIG. 2 is a block diagram that illustrates useful compartments 200, according to some embodiments. The compartments 200 include a DNA interactions compartment 210, a cytosol compartment 220, a cell membrane compartment 230 and an external conditions compartment 240. Also depicted are a compartment 250 for a nucleus membrane a compartment 260 for special organelles, and sub-compartment 262 for organelle membranes in the organelle compartment 260 and sub-compartment 232 for lipoprotein interactions in cell membrane compartment 230.

[0116] A whole-cell model includes the gene products of a large fraction, if not all, of the genome. As used herein, the term whole-cell model means a computational process that generates populations of gene products in multiple functional compartments in each of multiple time steps from over 40% of genes in all chromosomes of a cell, preferably over 50% of the genes, more preferably over 90% of the genes, and when possible every gene in all chromosomes. In the example embodiment, the entire genome of a simple one-celled organism, *M. genitalium*, is included in the whole cell mode.

[0117] Thus, in some embodiments, a method includes determining sub-models for each compartment including: transcription regulation, transcription and RNA maturation in the DNA interactions compartment; translation, protein maturation, protein translocation and ribosome assembly in the cytosol compartment; and metabolism in the cell membrane compartment.

[0118] Once the compartments and sub-models have been selected, the inputs to each sub-model and the outputs of each sub-model are determined. Variable inputs used by more than one sub model and outputs of each model are collected as state variables. The values of the state variables describe the phenotype of the cell at a particular time. Constants used by the sub-models are considered model parameters, but values of those constants often have to be reconciled among multiple sub-models. The 28 models use the populations and configurations of various gene products and metabolites that have been usefully assembled into 16 multidimensional state variables. The multidimensional state variables are listed in Table 3. These multidimensional cell state variables are described in more detail below with reference to FIGs. 22 to 48.

Table 3. Multidimensional cell state variables.

number	State variable name	Description
1	Chromosome	For each gene: location, start coordinate, length, direction, type, for tRNA genes codon and amino acid, essential, expression shock, half life of each RNA transcript, sequence, G/C content, weight of each transcript, pI of each transcript, half-life RNA, decay rate RNA, synthesis rate RNA, polymerase binding probability RNA, homologs, database ref, transcription unit, reactions involving gene product.
2	Transcript	Transcription units: constituent genes. DnaA boxes. Restriction/modification (R/M) sites. Short tandem repeats, coordinate, direction, sequence and length, progress of the RNA polymerase along the transcription unit, whether aborted at this location, calculated dry weight.
3	RNA	For each RNA species: copy count, weights, compositions, lengths, and localizations, decay rates, modified and unmodified copy numbers, cleaved and

		uncleaved copy counts.
4	Polypeptide	Which mRNA are bound to ribosomes, how far translation has proceeded at current time step, partial polypeptides sequences tagged with proteolysis tag for degradation, length, tRNA sequence, and amino acid sequence of every monomer and the length, molecular weight, tRNA sequence, and amino acid sequence of every possible proteolysis tag.
5	Protein Monomers	For each gene coded protein monomer species: copy count, DNA-bound copy count, result of successful translation. Localization, signal sequence, N-terminal methionine cleavage, prosthetic groups, chaperone, topology, catalysis, DNA footprint (instability, stability, aliphatic index, GRAVY, extinction coefficient and absorption of gene), folded or unfolded, modified or unmodified, nascent or processed. Processing enzyme copy count, cytosolic- and membrane- and organelle cytosol- and organelle membrane-localized copy counts.
6	Protein Complex	For each protein complex species: copy count, DNA-bound copy number, complex subunit composition, molecular weights, amino acid composition, half-lives, and localization, minimum expression of each complex for each process with a minimum, folded and unfolded copy counts, and modified and unmodified copy counts.
7	RNA polymerase	Copy count, chromosome location of DNA-bound copy, expected probabilities of a polymerase being in four conditions (free, non-specific bound, specific bound, actively transcribing), fold changes to the base binding probabilities.
8	Ribosome	Copy count. For each copy: (1) the status – actively translating or stalled, (2) the mRNA (or tmRNA if stalled) species bound to and translating, and (3) the position, in codons from the start codon.
9	Cytokinesis protein	Number of edges and status of each edge (bent or not bent) in a protein complex associated with cell division, such as a FtsZ ring.
10	Metabolic reaction (rxn)	Records the calculated flux of each metabolic reaction.
11	Metabolite	Represents both the extracellular and cellular copy numbers of each metabolite (including ATP, ADP, H ₂ O, H ⁺).
12	Geometry	Diameter of cell midline (progress of cytosin), length of

		cylinder-shaped cell, width, volume (assuming constant density), surface area, septum length.
13	Host	Boolean variables indicating membrane signaling protein expression, host inflammatory response activation.
14	Mass	Cell mass and mass of each component: water mass, chromosome mass, RNA mass, protein monomer and complex masses, transcript mass, polypeptide mass, metabolites mass, mass of bi-lipid membrane (based on mass of other components and density).
15	Stimulus	Temperature in °C, six types of radiation in Wm ⁻² , Gy s ⁻¹ , or m ⁻² s ⁻¹ sr ⁻¹ , and three Boolean-valued stress conditions.
16	Time	Time elapsed from the start of the simulation in seconds as a single-integer variable.

[0119] Simulations are randomly initialized to the beginning of the cell cycle. Next, for each time step, the sub-models retrieve the current values of the cell state variables, calculate separate contributions to the temporal evolution of the cell state variables, and update the values of the cell state variables. This is repeated thousands of times over the course of each simulation. FIG. 3 is a block diagram that illustrates example simulations using cell state variables and sub-models, according to an embodiment. Values of 16 multidimensional cell state variables 310 at one time are used as input to the 28 cell process sub-models 320. Cell functions and variables are grouped into 5 physiologic categories: DNA, RNA, protein, metabolite, and other. Lines between the variables and sub-models indicate the cell variables predicted by each sub-model. The number in parentheses next to each sub-model indicates the number of genes involved in the sub-model processes in the example organism, *M. genitalium*. The sub-models are executed (in random order in some embodiments to fairly apportion limited resources) to model corresponding processes for the same time duration (called the time step) to produce a phenotype made up of new values of the cell state variables at a later time. That phenotype is compared to a simulation end condition 330, such as the cell dividing (occurs when a septum diameter output by cytokinesis sub-model equals zero), or a certain number of rounds of executing the sub-models. If the condition is satisfied, the simulation ends. Otherwise the values of the cell state variables output by the sub-models

are used as input to the sub-models for another round of execution to advance the cell state another time step. In other embodiments, a whole-cell model includes more or fewer multidimensional cell state variables or sub-models or some combination.

[0120] An advantage of the whole-cell model approach, in particular because of the focus on the individual cell, is that uncertain values for any variables or parameters can be determined in order to produce molecular closure and temporal closure. Molecular closure means that the populations of gene products are limited by the availability of the molecules that make them up. When all molecular resources are consumed, no new gene products or derivative complexes can be generated, until the resident gene products are broken down into their constituent molecules for reuse or new molecules are imported through the cell membrane. Because energy, required for many reactions involving gene products, is provided by certain molecular species (e.g., Adenosine Tri-Phosphate, ATP), the molecular closure for the changes between such species (e.g., between ATP and Adenosine Di-Phosphate, ADP) also provides for energy closure. Temporal closure means that if the cell is modeled through to cell division, the state the daughter cell population should be within acceptable variations of the original state of the parent cell population.

[0121] FIG. 4 is a flowchart that illustrates an example method for forming a whole-cell model, according to an embodiment. Although steps are depicted in FIG. 4, and in subsequent flowcharts, as integral steps in a particular order for purposes of illustration, in other embodiments, one or more steps, or portions thereof, are performed in a different order, or overlapping in time, in series or in parallel, or are omitted, or one or more additional steps are added, or the method is changed in some combination of ways.

[0122] In step 401, compartments are selected for a cell type. The compartments include at least the DNA interactions compartment, cytosol compartment, cell membrane compartment; and external conditions compartment. In other embodiments, other compartments, such as those depicted in FIG. 2 are also included. For modeling *M. genitalium*, the special organelle compartment is a so-called terminal organelle used for attaching to the urogenital passage of a host organism. In some embodiments, the determination of compartments includes performing a comprehensive search of the available literature on the subject. In many cases,

direct information about an organism compartment is not available. In such cases information about a closely related organism is used. For example, for instances in which the knowledge base of *M. genitalium* is lacking, information about *Escherichia coli* (*E. coli*) was substituted. In some cases other *Mycoplasmas* are used; in other cases, *E. coli* was used; and in yet other cases organisms were not limited to *Mycoplasmas* or *E. coli*.

[0123] In step 403, sub-models are determined for modeling the processes in each compartment including, at least: transcription regulation, transcription and RNA maturation in the DNA interactions compartment; translation, protein maturation, and ribosome assembly in the cytosol compartment; and, protein translocation and metabolism in the cell membrane compartment. In other embodiments, more sub-models are included for simulating other processes in various compartments, such as one or more of the sub-models listed above in Table 2, or others not listed there. In some embodiments, the determination of processes or the determination of existing sub-models for those processes includes performing a comprehensive search of the available literature on the subject. In many cases, direct information about an organism genome and processes is not available. In such cases information about a closely related gene is used, such as information about homologs in other organisms. As before, in some cases other *Mycoplasmas* are used, in other cases, *E. coli* was used, and in yet other cases organisms were not limited to *Mycoplasmas* or *E. coli*.

[0124] In step 405, variable inputs and outputs are determined for each sub-model (such as chromosome and copy counts of molecules in compartment and cell mass) and parameters (such as reaction rate statistics, percentage of cell mass associated with an important molecule). In some embodiments, the determination of parameter values for those processes includes performing a comprehensive search of the available literature on the subject.

[0125] In step 407, cell states variables are determined based on sub-model inputs and outputs. Because each cell state variable is multidimensional, each is stored in a data structure that includes one or more Boolean, scalar or array fields, or some combination. The cell state data structure is generated to store each multidimensional attribute of cell state and to indicate cell state methods to initialize, update, retrieve or derive values for every attribute of the cell state. As is well known, object-oriented programming structures provide a

computational mechanism for storing multidimensional variable values and methods for manipulation of those values, such as to initialize, update, retrieve or derive values. Objects of object-oriented programming are instances that can inherit methods and attributes, such as parameter values, from an object-oriented class. Thus, in some embodiments, one or more of the cell state variable data structures are objects of a cell state variable class.

[0126] In step 409, for each sub-model, methods are generated to utilize cell state data structures as input or output or both, and to set parameter values. In object oriented programming, methods are implemented as a method class. Thus, in some embodiments, step 409 includes generating or update a cell process sub-model class for each sub-model so that the class utilizes values stored in cell state variable objects.

[0127] In step 411, a time scale for which input and output state variables for each sub-model are well mixed in every compartment where the sub-model's processes occur is determined. In step 413 steady state conditions are determined for at least some sub-model. Steady state conditions are those values of the cell state variables utilized by the sub-model that cause the model to output cell state variable values that are substantively the same as the input cell state variable values. Steady state values are useful for comparison with observations to ensure that sub-model parameters are properly set.

[0128] In step 421 it is determined whether there is a another sub-model for which to perform steps 405 through 413. If so, control passes back to step 405. If not, control passes to step 431. In step 431 a time step for simulation of the cell is determined based on the well mixed time scale. The time step should be short enough to resolve the evolution of most cell state variables over time, but not so short that many extraneous time steps are taken to model even the fastest processes. To keep the time step from being too long, the time step it is selected to be shorter than the time for at least some sub-model processes to complete. To keep the time step from being too short, the time step it is taken to be about the same as the fastest well mixed time scale. In the example embodiment, the time step is selected to be 1 second.

[0129] As a result of method 400, data structures are constructed to store knowledge base for much or most of the genome (40% to 100%) including sub-models for separate processes and state variables in multiple compartments of the single cell.

[0130] Some differences between the whole-cell model algorithm and numerical ordinary differential equation (ODE) integration methods are: (1) the whole-cell model “equations” are grouped into multiple sub-models, (2) the whole-cell model “variables” are grouped into a plurality of multidimensional cell state variables, (3) the values of the cell state variables should be allocated among and within the sub-models at each time step to satisfy the assumption that the sub-models can execute independently, and (4) the discrete nature of many of the variables due to the focus on single cells.

2. Application of whole-cell model to cellular engineering

[0131] FIG. 5A through FIG. 5C are flowcharts that illustrate an example use of whole-cell model simulation, according to an embodiment. FIG. 5A is directed either to training the whole-cell model based on observations, or to engineering a cell that exhibits a desired phenotype, e.g., the production of a certain product, such as an enzyme for digesting oil, or behavior, such as growth under conditions of abnormally high temperatures. In step 501 it is determined what behavior or product is desired in an organism and what known subject organism is suitable as a starting organism because that subject organism phenotype is close enough to the desired phenotype. In some embodiments directed to training or calibrating the whole-cell model, the desired phenotype is the actual phenotype of a subject organism.

[0132] In step 511, a whole-cell model of the subject organism is constructed, or retrieved if already constructed. Such a model can be constructed as described above with reference to FIG. 4. In some embodiments, the sub-models included in step 511 have not been optimized for use in a particular organism. During step 513, the values of parameters of the sub-models, such as kinetic rates, are aligned so as not to be contradictory with parameter values used in other sub-models or so as to produce concentrations and proportions of gene products in a normal range under normal conditions. In some embodiments, the parameter values are set to

provide molecular closure. In some embodiments, this optimization has already occurred in step 511, and step 513 is omitted.

[0133] In step 515, initial conditions variance control is determined. Cell theory states that all cells are created from old cells, or, more mathematically, that on the time scale of a single generation, mother and daughter cells are statistically similar. This principle was used to relate the initial and final cell state distributions of the whole-cell model, providing temporal closure and a rigorous way to define the initial cell state distribution in terms of the dynamic model. The total cell mass variance σ_m , and initial copy counts η_r of all RNA and η_p of all protein species, are used to control initial conditions variance.

[0134] A method that calculates the expectation value of each cell state variable was developed based on the whole-cell model of step 511. This method is depicted in FIG. 5B. FIG. 5C shows how to use the method of FIG. 5B to apply the principle of statistical equivalence of mother and daughter cells. First, the inputs of FIG. 5B were determined initially with σ_m , η_r and η_p all set equal to zero. Second, as shown in FIG. 5B, the distribution of each cell state variable was approximated by a standard, well-behaved statistical distribution, and set the mean of each distribution to its calculated value. For example, it was assumed that the copy number of each RNA and protein species is multinomially distributed, and that the total cell mass is normally distributed. Third, also shown in FIG. 5B, the variance of the distribution of each cell state variable was parameterized and initially set the variance of each distribution to zero. These distributions were sampled and used to set the initial value of each state variable. Fifth, as shown in FIG. 5C, a population of wild type cells was simulated using this cell state initialization procedure and the variance of the final distribution of each cell state variable calculated. Sixth, as shown in FIG. 5C, the initial variance of each state variable was set to its calculated final variance. Finally, steps five and six were repeated until convergence of the values of the variance control parameters.

[0135] In step 523, the elapsed time is set, e.g., initially the elapsed time set to zero, the time of the initial conditions set in step 521; but, subsequently, the elapsed time is incremented by one time step. In an illustrated embodiment, the time step is 1 second. In step

525, the external conditions are set, so that time varying external conditions can be simulated, such as rising temperatures or falling levels of metabolites.

[0136] In step 530 shared resources, such as metabolites, are divided among the processes competing for those resources. For example, in the illustrated embodiment, step 530 includes steps 531 through 539. In some embodiments, the shared resources are not evenly distributed but can be differentially concentrated in one compartment compared to another. Such embodiments include step 531 to determine the next compartment with shared resource to divide. In some embodiments, in which processes in all compartments compete for the same resources, step 531 is omitted. In step 533, the next resource and amount available (e.g., in cell or compartment) is determined. In step 535, the next process that uses the shared resource is determined. In order to prevent one or a few processes from always getting the resources, the process is determined at random. Note that one sub-model may include multiple processes; but the sub-model is configured to apportion the amount of the shared resource allotted to the sub-model among the processes included in the sub-model. Because the sub-models run independently, process 530 divides the resources among multiple sub-models before any sub-model is invoked.

[0137] In step 537, the demand for the resource by the sub-model is determined. For example, one sub-model may not require a particular resource at all, another requires only one copy of the resource, and another may require multiple copies of the resource, e.g. for large reactions or a large number of small reactions included in one sub-model. Steps 535 and 537 are repeated until all sub-models have been considered, steps 533 through 537 are repeated until all resources have been considered, and steps 531 through 537 are repeated until all compartments have been considered. When the demand for every shared resource is known for all compartments, control passes to step 539.

[0138] In step 539, each shared resource is divided among the sub-models based on the relative demand by those sub-models, i.e., proportional to the demand. Thus a first sub-model that demands twice as many ATP molecules as a second sub-model will receive twice as many as the second sub-model.

[0139] In step 550 the evolution of values for each cell state variable is determined. For example, step 530 includes steps 551 through step 555. In step 551 the next sub-model to be executed is determined. In some embodiments, the next sub-model is chosen at random so that a second sub-model that depends on an unshared output of a first sub-model will, at least on occasion, be executed after the first sub-model and not have to wait until the next time step.

[0140] In step 553, the next sub-model is executed by retrieving the values of any unshared cell state variables and retrieving the divided part of any shared cell state variable. In step 555, the contributions (negative or positive) to the values of the cell state variables by executing the process are combined with the values already in those cell state variables and the values of those cell state variables updated. In some embodiments, the effects are not additive; step 555 includes any switching, saturation, or multiplicative effects of each contribution to the value of the cell state variable.

[0141] After all sub-models have been executed, control passes to step 571 to determine if simulation end conditions have been satisfied. If not, control passes back to step 523 and following to increment the elapsed time by the time step, divide the shared resources and execute all the sub-models.

[0142] If simulation end conditions have been satisfied, then in step 573 it is determined if the simulated phenotype is the desired phenotype. For example, does the simulation indicate that the genotype used in step 511 produces an organism with the desired behavior or product? If so, then an organism should be genetically engineered with the genotype used in step 511. If not, then the genotype should be adjusted in step 575 and the model run again with the new genotype by returning control to step 511.

[0143] Any method may be used to genetically engineer the organism. Introducing genes into microorganisms or cell culture lines is foundational to the biotechnology industry, enabling the production of proteins, biofuels, and biochemical. This process known as genetic engineering is often accomplished by inserting genes into a larger piece of DNA, such as a plasmid, that then gets transferred to a host cell. Genetic engineering alters the genetic makeup of an organism using techniques that remove heritable material or that

introduce DNA prepared outside the organism either directly into the host or into a cell that is then fused or hybridized with the host. Genetic perturbations include the introduction of a plasmid or virus containing heterologous genes, knock-outs and knock-ins of chromosomal genes, and modifications of functional and regulatory elements encoded in DNA (e.g., promoters, transcription-factor binding sites, ribosomal binding sites, regions of secondary structure in the transcribed RNA, termination sites). Alternatively, an entire chromosome can be synthesized and transplanted into a competent recipient host.

[0144] The molecular biology methods and techniques are generally performed according to conventional methods well known in the art and as described in various general and more specific references that are cited and discussed throughout the present specification unless otherwise indicated. See, e.g., Sambrook *et al.*, *Molecular Cloning: A Laboratory Manual*, 2d ed., Cold Spring Harbor Laboratory Press: Cold Spring Harbor, N.Y. (1989); Ausubel *et al.*, *Current Protocols in Molecular Biology*, Greene Publishing Associates (1992, and Supplements to 2002); Harlow and Lan, *Antibodies: A Laboratory Manual*, Cold Spring Harbor Laboratory Press: Cold Spring Harbor, N.Y. (1990); *Principles of Neural Science*, 4th ed., Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, editors, McGraw-Hill/Appleton & Lange: New York, N.Y. (2000).

[0145] Expressing newly-introduced genes often has unintended and negative consequences for a host cell, diverting resources away from critical processes. The precise behaviors that occur are complex and difficult to anticipate. Application of whole-cell modeling can address these problems. Whole-cell modeling incorporates known molecular biology into a quantitative framework for running computational simulations. These simulations are capable of predicting the multifactorial effects of the addition or removal of one or more genes in a cell. Additional genes can be modeled either by defining additional genes on one of the existing chromosomes, including possibly extending the sequence of the existing chromosome, or by modeling extrachromosomal plasmids analogously to genomic DNA. Genes can be removed from the model either functionally by eliminating their RNA and protein expression, or structurally by removing the gene's transcription start and stop sites and possibly altering the chromosome sequences.

[0146] In step 581, it is determined whether the actual phenotype of an organism, such as the genetically engineered organism, matches the simulated phenotype. If so, then the process is done. If not, then the model is flawed and should be adjusted in step 583, and run again with the adjustments by returning control to step 511.

[0147] In some embodiments, the model is trained by comparisons between simulated phenotypes and observed phenotypes, as described in more detail below with reference to an example whole-cell model for *M. genitalium*. In such embodiments, there is considered to be no desired phenotype different from the simulated phenotype, and step 573 is omitted. Control passes from step 571 on the yes branch directly to step 581. If the simulated phenotype matches the actual phenotype, then the model training is done. If not, the model is adjusted in step 583, and control passes back to step 511.

3. Cell state variable classes

[0148] In this section the attributes and methods of the cell state variable classes are presented. For brevity, a state variable class is often referred to as a state class in the following. Although data structures and fields are depicted in many of the diagrams in this section as integral blocks in a particular order for purposes of illustration, in other embodiments, one or more data structures or messages or fields, or portions thereof, are arranged in a different order, in the same or different number of data structures or databases in one or more hosts or messages, or are omitted, or one or more additional fields are included, or the data structures and messages are changed in some combination of ways.

3.1 Chromosome state variable

[0149] Chromosomes encode the structure and function of every RNA and protein, and thereby control cellular behavior. Due to their critical function and large size, cells dedicate considerable resources to chromosome replication, maintenance, and compaction. This state represents the polymerization, winding, modification, protein occupancy, and (de)catenation status of the chromosome(s). This state represents the polymerization, winding, modification, and protein occupancy of each nucleotide of each strand of each copy of the organism

chromosome, and the (de)catenation status of the two sister chromosomes following replication.

[0150] FIG. 6A through FIG. 6B are block diagrams that illustrate example fields and arrays for representing chromosome data in the Chromosome state variable class, according to an embodiment. Many of the properties associated with these fields are described in more detail below with reference to the sub-models. FIG. 6A depicts example fields according to one embodiment. Chromosome ID field 601 holds data that indicates an identifier (ID) for a particular chromosome. In some embodiments, the subject organism has only one chromosome and chromosome ID field 601 is omitted. Chromosome length field 602 holds data that indicates length of the chromosome in nucleotides (bases). OriC position field 603 holds data that indicates a position along the chromosome where replication commences; TerC position field 604 holds data that indicates a position along the chromosome where replication is completed. Chromosome sequence field 605 holds data that indicates the sequence of nucleotides of the forward strand. For brevity going forward, the expression that a field “indicates” means that the field “holds data that indicates.”

[0151] The regions accessible for protein binding field 606 indicates the position ranges in bases where the chromosome can be accessed for binding with proteins of sufficient affinity. The locations of DNA-bound proteins field 607 indicates the position in bases where a protein is actually bound and the species of the bound protein. The DNA footprints of proteins field 608 indicates the DNA footprints of the bound protein species. The DnaA complex at OriC field 610 indicates whether that complex is currently bound to the chromosome at that location, signaling that replication can begin. The polymerized regions field indicates the range of bases that have been polymerized (copied by the growing complementary strand during replication). The superhelicity field 612 indicates the change in linking number from relaxed coiling for the chromosome or a replicating portion of the chromosome.

[0152] The strand breaks 613 field indicates locations of strand break, such as introduced when fragments are formed during replicating of a lagging strand or during chromosome damage. The unbound bases field 614 and damaged bases field 615 indicate types of

chromosome configurations that invoke chromosome repair. Segregation status field 616 indicates whether replicated chromosomes have separated and moved to opposite poles of the cell.

[0153] The chromosome includes many genes, transcribed in groups of one or more genes called transcription units. In various embodiments, properties of each gene or transcription unit are also included in this state class. For example each gene is described by a field 620 with multiple component fields, which is repeated for each gene or transcription unit, as indicated by ellipsis below field 720. Gene ID field 621 indicates a unique identifier for a gene, such as gene nomenclature name or MG number. Gene start coordinate field 622 indicates the location on the chromosome where the gene begins, the direction field 623 the direction of the gene from that location, and the length field 624 the number of bases in the gene. The transcription unit ID field 625 indicates the transcription unit to which the gene belongs with a unique identifier for the transcription unit. The essentiality field 626 indicates, e.g., with a Boolean value, whether expression of the gene is essential for the cell. The type field 627 indicates the type of RNA produced from the gene (mRNA, rRNA, sRNA, tRNA) and the promoter field 628 indicates the location and base sequence of the promoter where the RNA polymerase binds to the DNA.

[0154] In some embodiments, the chromosome state variable class includes a number of wild variations field 631 that indicates how many variations of this gene are found in nature and the sequence of wild type field 632 or single nucleotide polymorphisms (SNPs) field 633 indicates the bases in the wild variation. The G/C content field 634 indicates the percent of the bases that are either guanine or cytosine, indicative of increased probability for recombination.

[0155] In support of transcription, the Chromosome state variable class includes a polymerase ID field 635 indicates uniquely one of the polymerases produced and the polymerase binding probability field 636 indicates the probability that polymerase will bind to the promoter for the current gene. The weight of the transcript produced, the rate at which the transcript is synthesized, the half life of the transcript, and the decay rate of the transcript are indicated by the contents of field 641, field 642, field 643 and field 644.

[0156] The product of this gene, either RNA or protein from mRNA is then involved in one or more reactions modeled by the sub-models. So in the illustrated embodiment, the Chromosome state variable class 600 includes field 634 to indicate the reactions involving the products of this gene. If the gene is for tRNA, the associated codon and amino acid are indicated by field 646. The tendency of the gene to be expressed under shock conditions, such as temperature shock, chemical shock, or radiation shock, e.g., at four ultraviolet wavelengths, is indicated by the contents of field 647.

[0157] Homologs field 651 holds data that indicates genes in other organisms with very similar sequences, which can be used in some embodiments to determine the properties of the gene in the subject organism. The database references field 652 indicates what databases were the sources of the values placed in one or more fields for this gene. Other fields for the gene, if any, are indicated by ellipsis.

[0158] The methods of the Chromosome state class include: a method 691 to retrieve values from an object of the class; a method 692 to update values in one or more fields; and a method 693 to initialize values in one or more fields. The illustrated embodiment includes one or more methods 694 to derive a value in one field from values in one or more other fields (such as a method to dissociate DnaA from the chromosome after initiation of replication). Other methods for the chromosome state variable class are indicated by ellipses below method 694.

[0159] FIG. 6B summarizes the mathematical representation of several properties of chromosome(s) including the size and type of multiple attributes, according to another embodiment. Mathematically, several properties are each represented by a tensor for base (nucleotide) i on strand j of chromosome k . The index i varies from 1 to L indicating the length of the chromosome in bases. The index j varies from 1 up to 2 strands; and the index k varies from 1 up to the number of chromosomes (2 for replication of an organism with a single chromosome, such as *M. genitalium*). Properties of bound metabolites, proteins or protein complexes include a fourth dimension with index l to indicate the bound species. Column 671 indicates the name of the physical property. Column 672 indicates a mathematical notation symbol for the property. Column 673 indicates the size of the array

specifying the property at each base of the chromosome and column 674 indicates the type of value that fills the array, either Boolean (True or False) or integer or real.

[0160] Each property except winding, base and sugar-phosphate modification, protein occupancy and catenation, is represented as a 3-dimensional Boolean tensor. Winding (w in the mathematical notation) is represented as a 3-dimensional real tensor which indicates the linking number density of each nucleotide. Base and sugar modification (m^b and m^p , respectively, in the mathematical notation) are represented as 4-dimensional tensors which indicate the chemical identity $l = \{1..M\}$ of a species bound to each nucleotide, where M is the number of distinct metabolite species represented by the Metabolite state. Protein monomer and protein complex occupancy (b^m and b^c , respectively in this notation) are represented as 4-dimensional tensors which indicate the identity, $l = \{1..B^m\}$ and $l = \{1..B^c\}$ respectively, of the protein bound at each nucleotide, where B^m and B^c are the numbers of distinct protein monomer and complex species represented by the Protein Monomer and Protein Complex states. (De)catenation (s) is represented as a Boolean scalar. The other properties represented by 3-D Boolean tensors include polymerization (p), gap site modification (m^g), abasic site modification (m^a), intra-strand cross link modification (m^c), strand break modification (m^s), and Holliday junction modification (m^h). The total size is within the data structures easily maintained by modern computational equipment.

[0161] The Metabolite state describes the identity of each modified base and sugar-phosphate. The Protein Monomer and Protein Complex states represent the total DNA-bound copy count of each protein monomer and macromolecular complex. For computational efficiency, the RNA Polymerase state redundantly represents the chromosomal location of each bound RNA polymerase. The mass of the chromosome(s), including all modifications and all DNA-bound proteins, is included in the cell mass calculated by the Mass state.

[0162] Ten processes access and modify the Chromosome state variable class. The Replication Initiation sub-model models DnaA DNA-binding and formation of the oriC DnaA complex which promotes replication initiation. The Replication sub-model models bidirectional DNA polymerization from the oriC, continuously on the leading strand and

discontinuously on the lagging strand, and Okazaki fragment ligation. The Chromosome Segregation sub-model models sister chromosome decatenation following successful chromosome replication. The Cytokinesis sub-model models cell division following successful decatenation. The Chromosome Condensation and DNA Supercoiling sub-models model SMC-and supercoiling-mediated chromosome compaction. The DNA Damage sub-model models stochastic and radiation- and chemically-induced DNA damage. The DNA Repair sub-model models three DNA repair pathways—base excision repair, nucleotide excision repair, and homologous recombination. The DNA Repair sub-model also models methylation and restriction of MuiI (MG184) restriction/modification sites. Finally, the Transcriptional Regulation and Transcription sub-models model the binding of transcription factors and RNA polymerases to promoters and the synthesis of RNA.

[0163] Initial conditions are determined as follows and illustrated in FIG. 6C by the values 679 listed on the right side of the setting equations, and the symbol 672 on the left side.

First, the chromosome state variable is initialized with un-replicated, supercoiled, and fully methylated chromosome(s), as indicated in FIG. 6C. Here σ_{ss} indicates the observed steady state superhelicity and h is the equilibrium helical repeat length. Second, the Protein Monomer and Protein Complex states initialize the total copy count of each protein species. The Chromosome Condensation, DNA Supercoiling, Replication Initiation, Transcriptional Regulation, and Transcription processes initialize the protein occupancy of the chromosome.

[0164] The values of some properties are obtained by fitting using training data. For example, the affinity of RNA polymerase for each promoter can be fit to provide the gene products required by the 28 modeled processes to reproduce the observed mass doubling time.

[0165] Based on the literature, the reconstructed example *M. genitalium* chromosome contains 525 genes based on the Comprehensive Microbial Resource (CMR) genomic annotation organized into 335 transcription units based on the *M. pneumoniae* operon organization experimentally defined and the *M. genitalium* promoters computationally predicted. The reconstructed genome also contains 17 transcriptional regulatory elements based on 15 studies and databases. 2,283 DnaA binding sites were computationally identified

based on the consensus binding motif reported. MunI restriction/modification (R/M) sites were computationally identified based on the reported MunI binding motif, and 19 short tandem repeats were identified. Additionally, the location of the replication origin and the predicted DnaA box sites were reconstructed based on studies. Finally, the reconstructed steady-state superhelicity is based on the observed equilibrium helical repeat length of plasmid DNA and the observed superhelicity of *E. coli* DNA.

[0166] The binding motif and footprint of each DNA-binding protein was reconstructed from several experimental studies and the databases 3D-Footprint, NDB, and ProNIT. DNA-bound protein displacement reactions were assembled describing which proteins each protein species is able to displace from the chromosome. Functionally, displacement of DNA-bound proteins enables proteins to access the chromosome and fulfill their chromosomal replication or maintenance role. The affinity of RNA polymerase for each promoter was first reconstructed from the observed RNA composition of *E. coli*, the expression of each *M. genitalium* mRNA reported, the half-life of each *E. coli* mRNA reported, the observed amino acid composition of *M. gallisepticum*, and the Mycoplasma genetic code. Subsequently, the affinity of RNA polymerase for each promoter was fit to match the additional data used to train the 28 modeled cellular processes.

[0167] In the example embodiment for *M. genitalium*, there is one chromosome, $h = 10.5$, $\sigma_{ss} = -0.06$, $L = 580076$, $M = 722$, $B^m = 482$ and $B^c = 201$. The chemical composition of *M. genitalium* and the objective of the flux-balance analysis metabolic model were fit to match the mass and dNMP composition of the *M. genitalium* chromosome. The affinity of RNA polymerase for each promoter was fit to provide the gene products required by the 28 modeled processes to reproduce the observed nine-hour mass doubling time of *M. genitalium*. The oriC DnaA DNA-binding cooperativity was fit to match expectations for the duration of the replication initiation cell cycle phase. The expression and activity of DNA gyrase and topoisomerase I were balanced to match the observed steady state superhelical density.

3.2 Transcript state variable

[0168] During Transcription, an RNA polymerase binds to a gene promoter and facilitates the synthesis of an RNA transcript. This state stores all of the information that pertains to nascent RNA transcripts and aids in the time evolution of Transcription. Transcription proceeds at a maximal rate of about 50 nucleotides per second. Therefore, it may take several time steps to synthesize a transcript, and the progress of the RNA polymerase along the transcription unit must be stored. This state stores the transcription unit to which each RNA polymerase is bound, including which chromosome it is bound to and the progress of the RNA polymerase along the transcription unit.

[0169] Furthermore, an RNA polymerase may stall in the transcription process or be knocked off of a gene by another protein. In these cases, an incomplete transcript may result, and will be targeted for degradation by the RNA Decay process. Therefore, the Transcript state variable also holds the sequence of each aborted transcript, such that the nucleotides can correctly be accounted for upon degradation.

[0170] Multiple RNA polymerases may be bound to a given transcription unit, and this state can calculate the total number of polymerases on each unit. The Transcript state also calculates the dry weight of the nascent transcripts.

[0171] FIG. 7 is a block diagram that illustrates example fields in the Transcript state variable class 700, according to an embodiment. A copy count field 701 indicates how many transcription units populate the chromosome. An ID field 703 uniquely identifies a particular transcription unit. The length, start coordinate, direction of transcription and nucleotide sequence are indicated in fields 703, 704, 705 and 706, respectively. The genes included in the transcription unit are listed in field 711, the location of DnaA boxes are indicated in field 712, the location of R/M sites are indicated in field 713, and the locations of short tandem repeats are indicated in field 714. The current location of an RNA polymerase along the transcription unit, in number of nucleotides from the start coordinate, is given in field 721. The abort status field 722 indicates whether the polymerase is progressing normally or has stalled, e.g., with a Boolean value. Other fields for a particular transcription unit are indicated

by ellipses to the right of field 722. Fields 702 through 722, and any following fields, are repeated for the next transcription unit, as indicated by ellipsis below field 721.

[0172] The methods 790 of the transcript state class include: a method 791 to retrieve values from an object of the class; a method 792 to update values in one or more fields; and a method 793 to initialize values in one or more fields. The illustrated embodiment includes one or more methods 794 to derive a value in one field from values in one or more other fields, such as calculating the dry weight of nascent transcripts. Other methods for the chromosome state variable class are indicated by ellipses below method 794. For brevity, similar methods for the remaining state variables are represented by a single method field.

[0173] The Transcript state class retrieves the existence of an aborted sequence from the RNA polymerase state class. Upon RNA polymerase displacement, the Transcript state class updates the Transcript state in the Chromosome state class. It also records the start sites and directions of transcription units in the Chromosome state class. The Transcription sub-model class uses all of the fixed properties and time evolving properties housed in the Transcript state class.

[0174] In addition, there are multiple fixed properties (essentially parameters of several sub-models) relating to the gene templates of transcripts that are useful for transcription and advantageously stored in the Transcript state. Thus, as depicted in FIG. 7, for each transcription unit, the 5' coordinate of the gene on the genome is stored. This is the template for the 1st nucleotide in a growing transcript. In addition to this, the direction in which the template is read, towards or away from the origin, is stored. The sequence and length of each transcription unit is also stored.

[0175] An RNA polymerase may be initialized in the actively transcribing state. For all such polymerases, the growing transcript is accounted for in the Transcript state class.

3.3 RNA state variable

[0176] Transcription leads to the production of nascent RNA that, once mature, may be used in various cell functions. The mRNAs serve as a template for protein synthesis, rRNAs are a part of the ribosome structure, tRNAs carry amino acids to growing polypeptides, and

sRNAs act as regulators. The tmRNAs come into action when ribosomes stall on an mRNA during translation, helping recycle the stalled ribosome and adding a proteolysis tag to the incomplete polypeptide. FIG. 8A is a block diagram that illustrates example stages of RNA maturation and change tracked in the RNA state variable class, according to an embodiment. Different nascent RNAs 761 resulting from transcription undergo various steps towards maturation, including cleavage from polycistronic (transcription unit) to single RNA forms such as intergenic RNA segments and processed RNA forms 765, including tRNA, mmRNA, sRNA, tRNA and tmRNA. Further processes of methylation and thiolation produce mature RNA 767 of the same types. Maturation is carried out by the RNA Processing and RNA Modification process classes.

[0177] Subsequent changes include aminoacylation to bind amino acids to tRNA, producing aminoacylated tRNA 771, binding with other gene products, such as ribosomes, to produce bound RNA 773, and damage to produce damaged RNA 775. Thus, an RNA can also transition between various non-functional and functional forms. RNAs may exist with proteins in a macromolecular complex, and when the complex is marked for decay by the Protein Decay process class, its RNA subunits are marked as damaged in the Protein Monomer state class. Damaged RNAs are degraded by the RNA Decay process class. Further, a mature RNA may be freely floating in the cytoplasm or bound to another molecule in the cell. For example, an mRNA may be bound to a ribosome. This is accounted for in the Bound RNA form. The count of RNA in each of these immature/mature and functional/non-functional forms is stored in the RNA state variable class.

[0178] Thus, this state class holds information about RNAs such as their weights, compositions, lengths, and localizations. It also holds important parameters for the maturation pathway of RNA, such as a number of matrices that map one form of RNA to another, and the gene composition of transcription units. Other parameters such as expected RNA half-lives and RNA weight fractions are included in this class, and are used to determine the RNA Polymerase-DNA binding parameters associated with gene expression.

[0179] FIG. 8B is a block diagram that illustrates example fields in the RNA state variable class 800, according to an embodiment. Field 801 indicates a number of RNA species with

unique nucleotide sequences populating the cell. For each RNA species there is a field 810 with multiple component fields. Other fields 810 are indicated by ellipsis below field 810. State class methods 890 are also included in the class 800.

[0180] Each RNA species field 810 includes an RNA species ID field 811, uniquely identifying the species, e.g., by an alphanumeric code or name or name of gene of which the RNA is a product. The length of the RNA in number of nucleotides and the nucleotide sequence are indicated in fields 813 and 815. In some embodiments, the RNA is identified by its sequence in field 815; and, field 811 is omitted. In the illustrated embodiment, the RNA composition during several stages is tracked by fields 821, 831 and 841, among others indicated by ellipsis, for unmodified (nascent) RNA, cleaved (processed) RNA composition, and modified (mature) RNA composition, and for fields implied by ellipsis, aminoacylated or bound or damaged RNA compositions, or some combination. For each stage of RNA maturation and change, the compartment localization, if any, is indicated in fields 823, 833 and 843, among others, respectively; the copy count is indicated in fields 825, 835, 845, among others, respectively; the molecular weight is indicated in fields 827, 837, 847, among others, respectively; and whether marked for decay and decay rate or half-life is indicated in fields 829, 839, 849, among others, respectively.

[0181] Other properties are calculated by methods 890 of this state class. The RNA decay rates are calculated from the experimentally measured half-lives. The actual decay rates and RNA expression values used in the model are fit from the experimentally measured values such that the cell will double its mass and successfully divide by the end of the cell cycle.

[0182] FIG. 8C is a table that lists example interactions between different cell processes and the fields in the RNA state variable class.

[0183] The system is initialized with a set of RNAs. The determination of which RNAs are expressed and at what quantities is determined randomly based on expected RNA expression and expected total RNA mass fraction.

3.4 Polypeptide state variable

[0184] The Polypeptide state variable class keeps track of the progress of translation. In some embodiments, binding of free ribosomes to mature mRNAs is determined by mRNA availability. Once bound, a polypeptide is synthesized at a rate of up to 16 amino acids per second. Therefore, polypeptides may take multiple time steps to be completed as a protein monomer. The Polypeptide state class holds information about which mRNAs are ribosome bound, as well as ribosomal progress of translating a transcript across time steps. For reasons such as limited resources, a ribosome may stall resulting in an incomplete polypeptide. In such cases, a proteolysis tag (a short peptide added to the end of a nascent polypeptide) may be synthesized to mark the ribosome for release and mark the aborted polypeptide for degradation. This state also houses the progress of proteolysis tag synthesis and the amino acid sequences of the incomplete polypeptides that are to be degraded.

[0185] The Polypeptide state class serves as a “support system” for translation, providing it with information that is required to translate polypeptides. The state holds fixed information such as the length, tRNA sequence, and amino acid sequence of every protein monomer for the cell, and the length, molecular weight, tRNA sequence, and amino acid sequence of every possible proteolysis tag for the cell.

[0186] FIG. 9 is a block diagram that illustrates example fields in the Polypeptide state variable class 900, according to an embodiment. A polypeptide field 910 is utilized for every ribosome bound to an mRNA, whether the ribosome is actively translating or not. Thus, field 901 indicates the copy count for bound ribosomes. This is redundant with information included in the Ribosome state variable; thus, field 901 is omitted in some embodiments.

[0187] The ribosome field 911 uniquely identifies a particular ribosome. The associated mRNA species that the ribosome is bound to is indicated in field 913, such as with a unique species or copy identifier. The length is indicated in field 915. The corresponding tRNA sequence and amino acid sequence are explicitly stored in fields 917 and 919, respectively. Progress is indicated by the current translated length in field 921, and polypeptide weight in field 923. A unique identifier for a proteolysis tag species or copy, if any, is indicated in field 925. A non-null value in field 925 indicates the translation has stalled or aborted and the

corresponding polypeptide is marked for degradation. Similar sets of fields for other polypeptides being formed by ribosomes are indicated by ellipsis below field 910.

[0188] Information about the proteolysis tags available for marking an aborted or stalled polypeptide are provided in following fields. For example a unique identifier for the proteolysis species or copy is indicated in fields 931a and 931b, among others indicted by ellipsis, collectively referenced as field 931, for multiple species or copies. Similarly, the proteolysis tag length is indicated in fields 933a and 933b, among others indicted by ellipsis, collectively referenced as field 933, for multiple species or copies. The tRNA and amino acid sequences for the proteolysis are indicated in fields 935a and 935b, among others indicted by ellipsis, collectively referenced as field 935 for the tRNA sequences, and in fields 937a and 937b, among others indicted by ellipsis, collectively referenced as field 937 for the amino acid sequences. The proteolysis tag weight is indicated in fields 939a and 939b, among others indicted by ellipsis, collectively referenced as field 939, for multiple species or copies.

[0189] The Polypeptide state variable class includes methods 990.

[0190] The Translation process class reads all of the fixed parameters describing polypeptides and proteolysis tags. It both reads and updates the time evolving parameters describing polypeptides and proteolysis tags. The Protein Decay process class reads the sequence and length of each aborted polypeptide.

[0191] In an example simulation the state is initialized so that ribosomes are already bound to mRNAs and in the process of elongating a polypeptide. Each growing polypeptide is accounted for in the Polypeptide state class. No proteolysis tags exist at the start of the example simulation.

3.5 Protein Monomers state variable

[0192] Protein Monomers are the direct result of successful translation events. Upon translation, a monomer undergoes various steps towards maturation including deformylation, translocation, folding, and phosphorylation. As a result, a monomer can exist in many forms. FIG. 10A is a block diagram that illustrates example stages of protein monomer maturation and change tracked in the Protein Monomer state variable class, according to an embodiment.

As diagrammed in FIG. 10A protein monomers are found in nascent form 1081 after translation, processed (I) form 1082 after deformylation or methionine cleavage, translocated into or through the cell membrane 1083, processed (II) to remain in the membrane 1084 or through the membrane as signal sequences 1084, folded 1086, and mature 1087 after phosphorylation, lipoate ligation or glutamate ligation as the protein monomer moves through the maturation pipeline. Mature protein monomers can be combined with other macromolecules to form protein complexes 1089, as described in more detail next.

[0193] In addition to maturation, various processes can render a monomer non-functional, resulting in the misfolded 1088a, inactivated 1088b, or damaged 1088c forms. Further, a mature monomer may be freely floating in the cytoplasm or bound to another molecule in the cell to generate a bound form 1088d. For example, a translation factor may be bound to an mRNA, or a topoisomerase may be bound to a chromosome. The counts of monomers in each of the maturation phases, functional forms, and non-functional forms are stored in the Protein Monomer state variable class.

[0194] A purpose of the Protein Monomer state variable class is to hold the counts and attributes of the monomeric species in the system. This state class holds important information describing protein monomers such as their molecular weights, amino acids composition, length (in amino acids), and half-lives. A subset of monomers is translocated to different compartments, such as the membrane or terminal organelle. The Protein Monomer state class contains information about where each monomer localizes. The Protein Monomer state class also houses information that is important for the fitting and initialization of the model. Some essential functions require a certain abundance of particular proteins. For example, the Cell Division sub-model may require at least some threshold abundance of the division protein FtsZ for cell division to ever be possible. The system is then fit such that in an unperturbed state, at least this threshold amount of FtsZ is produced. This state stores the minimum expression of each monomer. The current version of this model involves a stochastic generation of the initial protein abundances in the cell. The initial abundance of certain monomers is more rigid than others for the maintenance of a stable system that can

grow and divide. The Protein Monomer state class stores the degree of variation allowed in the initial abundances of each monomer.

[0195] FIG. 10B is a block diagram that illustrates example fields in the Protein Monomer state variable class 1000, according to an embodiment. The number of protein monomer species, each species with a different amino acid chain, is indicated in field 1001. For each species a protein monomer species field 1010 includes multiple component fields. Similar fields 1010 for other species are indicated by the ellipsis below field 1010. One or more class methods 1090 are also included in the class 1000.

[0196] The component fields of field 1010 include the following. Field 1011 indicates a unique identifier for the monomer species. The length, amino acid sequence and weight of the monomer are indicated in fields 1013, 1015 and 1017, respectively. Fields 1021, 1023, 1025 and 1027 indicate the catalysis reactions, prosthetic groups, chaperones that assist in folding the monomer, if any, and topology. In addition, fields 1013, 1015, and 1017 indicate the reactions catalyzed by each monomer (or parent complex), the prosthetic groups required by each monomer, chaperones that assist in folding the monomer, and each monomer's topology. Catalysis may be defined in this instance as the reaction catalyzed by the protein monomer (or parent complex). Prosthetic groups include the metal ion cofactors that provide metal ion chemistry critical for chemical catalysis. Topology can be defined as the 3D-dimensional structure (e.g. domains). The DNA footprint of the monomer is indicated in field 1031. The processing enzyme that advances the nascent monomer to a processed monomer is indicated in field 1033. The count for the processing enzyme is indicated in field 1035. The counts for the various forms in the various compartments are indicated in the next fields.

[0197] The total count, nascent count, processed count, folded count, and mature count in the cytosol compartment are indicated in fields 1041, 1043, 1045, 1047 and 1049, respectively. The same breakdown counts for the membrane compartment are indicated in fields 1051, 1053, 1055, 1057 and 1059, respectively. The same breakdown counts for the organelle compartment are indicated in fields 1061, 1063, 1065, 1067 and 1069, respectively. The counts of misfolded, inactivated, damaged and bound monomers across all compartments are indicated in fields 1071, 1073, 1075 and 1077, respectively.

[0198] A purpose of the Protein Monomer state variable class is to hold the counts and attributes of the monomeric species in the system. This state class holds important information describing protein monomers such as their molecular weights, amino acids composition, length (in amino acids), and half-lives. A subset of monomers is translocated to different compartments, such as the membrane or terminal organelle. The Protein Monomer state class contains information about where each monomer localizes. The Protein Monomer state class also houses information that is important for the fitting and initialization of the model. Some essential functions require a certain abundance of particular proteins. For example, the Cell Division sub-model may require at least some threshold abundance of the division protein FtsZ for cell division to ever be possible. The system is then fit such that in an unperturbed state, at least this threshold amount of FtsZ is produced. This state stores the minimum expression of each monomer. The current version of this model involves a stochastic generation of the initial protein abundances in the cell. The initial abundance of certain monomers is more rigid than others for the maintenance of a stable system that can grow and divide. The Protein Monomer state class stores the degree of variation allowed in the initial abundances of each monomer.

[0199] FIG. 8C is a table that lists example interactions between different cell sub-models and the Protein Monomer state variable class, according to an embodiment.

[0200] The system is initialized with a set of proteins. The determination of which proteins are expressed and at what counts is determined randomly based on expected protein expression and expected total protein mass fraction.

3.6 Protein Complex state variable

[0201] Protein complexes are used by almost all of the sub-models in the system to perform their respective functions. A protein complex could be as small as transketolase, a dimer, or as large as pyruvate dehydrogenase that has 192 monomeric subunits. In organisms other than *M. genitalium*, a protein complex could be larger than 192 monomeric subunits. A complex may contain a mixture of protein and RNA subunits (e.g. ribosome), metabolites (e.g. DnaA protein bound to ATP), or ions (e.g. oxidized form of the protein thioredoxin). A

complex can also exist in various locations within a cell including in the cytoplasm, membrane, special organelle, or bound to the chromosome. The Protein Complex state class holds information about complex composition, as well as the counts of each complex in the cell, except for complexes, such as ribosomes and RNA polymerase that are specifically handled in other state variable classes.

[0202] The Protein Complex state class also holds values for fixed parameters used by multiple sub-models, including the complex subunit composition, molecular weights, amino acid composition, half-lives, and localization. Some functions essential to cell survival require a certain threshold abundance of particular protein complexes, and so the Protein Complex state class also holds the minimum expression of each complex.

[0203] FIG. 11A is a block diagram that illustrates example fields in the Protein Complex state variable class 1100, according to an embodiment. Field 1101 indicates a number of protein complex species that populate the cell. For each species a protein complex species field 1110 holds data for several component fields, including a complex species identifier (ID), molecular weight, half-life, number of subunits and subunit identifier in field 1111, field 1113, field 1115, field 1131 and field 1133. The species copy counts in various compartments are indicated in cytosolic count field 1121, membrane count field 1123, organelle count field 1125, among others indicated by ellipsis to the right of field 1125. Other fields included in some embodiments are indicated by ellipsis to the right of field 1133. Additional fields 1119 for other species are indicated by ellipsis below field 1110. One or more class methods 1190 are also included in the protein complex state class 1100.

[0204] FIG. 11B is a table that lists example interactions between different cell sub-models and the Protein Complex state variable class, according to an embodiment.

3.7 RNA Polymerase state variable

[0205] RNA polymerases are protein complexes that bind to gene promoters on the chromosomes and mediate the synthesis of RNA transcripts. This state class helps keep track of the conditions and chromosomal locations of RNA polymerases in the cell. In the illustrated embodiment, the condition of the RNA is tracked among four conditions: free (not

bound to DNA); non-specifically bound (bound to DNA but not to a specific gene promoter and not transcribing); specifically bound (bound to a specific gene promoter); and, actively transcribing (moving along a gene adding RNA bases to produce an RNA molecule).

[0206] As the simulation progresses, the RNA Polymerase state class stores the condition of each RNA polymerase in the simulated cell. Transition between the conditions may involve RNA polymerase association and dissociation from the cell's chromosome(s). This state class holds the precise positions on the chromosome(s) where polymerases are bound. The RNA Polymerase state class also handles basic accounting. For example, upon an RNA polymerase decay event, a free polymerase is decremented. Further, this class records the premature release of RNA polymerases from the chromosome(s) and passes information about the aborted transcript to the Transcript state class. Further, the Transcription sub-model requires the RNA Polymerase-promoter binding probability for each transcription unit. These probabilities may vary during the cell cycle due to the effects of other processes in the system, such as modeled by Transcriptional Regulation or DNA Supercoiling sub-models. The fold changes to the base binding probabilities incurred by these other processes is stored in the RNA Polymerase state class. This state class can also calculate the total number of RNA polymerases in each condition

[0207] FIG. 12 is a block diagram that illustrates example fields in the RNA Polymerase state variable class 1200, according to an embodiment. The number of RNA Polymerase molecules in the cell is indicated in field 1201. Details about each copy are indicated in field 1210 which is repeated for each copy as indicated by ellipsis below field 1210. Class methods 1290 are included in the RNA Polymerase state class 1200.

[0208] Field 1210 includes component field 1211 that identifies a particular copy, and field 1213 that indicates a location on the chromosome where the RNA polymerase is bound. Also included are values for some parameters, such as the probability that an unfolded RNA polymerase is free, bound, specifically bound or actively transcribing, as indicated in field 1221, field 1223, field 1225 and field 1227, respectively. Different probabilities may apply if the RNA polymerase is folded. The probability that a folded RNA polymerase is free, bound, specifically bound or actively transcribing, is indicated in field 1231, field 1233, field 1235

and field 1237, respectively. In some embodiments, the probability that unfolded RNA is free, bound, specifically bound or actively transcribing, is also included in other fields represented by ellipsis.

[0209] This state class interacts with the sub-models and other state classes as follows. The Chromosome state class updates the chromosomal positions of RNA polymerases in the RNA Polymerase state class. The Transcript state class reads the aborted RNA sequences from and writes the updated aborted sequences to the RNA Polymerase state class. The Transcription sub-model reads the RNA polymerase conditions and progress from this state class and updates the RNA polymerase conditions and progress to the RNA Polymerase state class. The Transcription sub-model also reads the RNA polymerase condition transition probabilities and RNA polymerase binding probabilities from the RNA Polymerase state class. The Transcriptional Regulation and DNA Supercoiling sub-models record the fold change in RNA polymerase binding probabilities to the RNA Polymerase state class.

[0210] RNA polymerases are initialized as follows: 1. Each RNA polymerase is randomly assigned (with replacement) to one of the actively transcribing, specifically bound, non-specifically bound, or free states weighted by the expected occupancy of each state. 2. Actively transcribing and specifically bound polymerases are randomly assigned to transcription units weighted by the transcription unit binding probabilities (P_{tu}). 3. Actively transcribing polymerases are randomly assigned to positions within the assigned segment of their assigned transcription unit (positions near the segment border are not allowed to prevent polymerases from being too close to each other) with uniform probability. 4. Non-specifically bound polymerases are randomly assigned to an accessible region on the chromosome.

3.8 Ribosome state variable

[0211] Ribosomes are a specific kind of protein complex handled separately. Ribosomes are large ribonucleoproteins which synthesize polypeptides. Following ribosome assembly, ribosomes synthesize amino acid polymers according to nucleic acid templates specified by mRNA and decoded by tRNA. Upon reaching the stop codon UAG, ribosome release factor

binds to the ribosome, recognizes the stop codon, hydrolyzes the peptidyl-tRNA bond, and releases the terminal tRNA. Finally, an elongation factor and an initiation factor dissociate the ribosomal subunits, the mRNA, and the ribosome release factor.

[0212] Upon prolonged translational stalling, tmRNA can displace both ribosome-bound tRNA and mRNA, leading to the generation of chimeric polypeptides containing an N-terminal mRNA-coded domain and a C-terminal tmRNA-coded degradation domain, or SsrA tag. Following translation termination, the protein degradation machinery recognizes the SsrA tag and degrades the chimeric polypeptide.

[0213] The Ribosome state represents (1) the status—actively translating or stalled—of each ribosome, (2) the mRNA, or tmRNA in the case of stalled ribosomes, species each ribosome is bound to and translating, and (3) the position, in codons, of each ribosome from the start codon. The status, bound (t)mRNA, and (t)mRNA position of each ribosome are implemented as scalar integer variables in the illustrated embodiment. FIG. 13 is a block diagram that illustrates example fields in the Ribosome state variable class 1300, according to an embodiment. The number of ribosome copies is indicated in field 1301. For each ribosome copy there is a field 1310 that is repeated for every copy as indicated by ellipsis. The field 1310 includes a copy identifier (ID) in field 1311, the status of the copy in field 1313, RNA bound to the ribosome in field 1315 and the position of the ribosome from the start codon in field 1317, among others indicated by ellipsis to the right of field 1317.

[0214] The Ribosome state class is related to other state classes and sub-models as follows. The Polypeptide state represents the sequence of each nascent polypeptide. The Protein Complex state represents the copy numbers of free ribosomal subunits (e.g., 30S and 50S in *M. genitalium*), and of mRNA-bound ribosomes (e.g., 70S in *M. genitalium*). The RNA state represents the copy number of each mRNA species and of tmRNA. Three sub-models—Translation, RNA Decay, and Protein Decay—access and modify the Ribosome state.

[0215] The Translation sub-model initializes the values in the state class and models the formation of ribosomes on mRNA, polypeptide synthesis catalyzed by ribosomes, and ribosome disassembly following translation termination. The Translation sub-model also

models ribosome stalling, tmRNA substitution, and SsrA degradation tag synthesis. Computationally, the Translation sub-model predicts the state of each ribosome, the mRNA or tmRNA species each ribosome is bound to, and the elongation rate of each nascent polypeptide. The mRNA and tmRNA degradation events modeled by the RNA Decay sub-model trigger early ribosome disassembly resulting in incomplete polypeptides. Similarly, ribosome degradation events modeled by the Protein Decay sub-model result in incomplete polypeptides. Both sub-models decrement the copy count of mRNA-bound ribosomes represented by the Protein Complex and Ribosome states. The Ribosome Assembly sub-model models the assembly of ribosomal subunits from rRNA and ribosomal protein monomers. The Protein Activation sub-model models the effect of antibiotics on the catalytic activity of ribosomal particles.

[0216] The initial conditions for values in the fields of the Ribosome state class are set as follows. After the RNA and Protein Monomer states initialize the total copy count of each RNA and protein monomer species and the Ribosome Assembly sub-model initializes the total copy count of the ribosomal subunits, the Translation sub-model initializes the Ribosome and Polypeptide states, and forms ribosomes equal to the minimum of the subunit copy counts, and randomly positions each ribosome on mRNA weighted by the copy count of each mRNA species.

3.9 Cytokinesis protein state variable

[0217] Cytokinesis is the division of a cell into two daughter cells. Many cells, including example organism *M. genitalium*, contain a cytokinesis protein called FtsZ that assembles into long filaments that are implicated in cell pinching during cytokinesis. These filaments bind to the cell membrane at the midline of the cell. FtsZ is a GTPase, and when the GTPs bound in a membrane-bound FtsZ filament hydrolyze to GDP, the filaments bend, thus constricting the membrane. In an illustrated embodiment, a geometric process of iterative filament bending is used in a cytokinesis sub-model, modified from a model proposed by Li et al. In summary, the FtsZ ring at the midline of the cell can exist in many states of various

numbers of filaments in the bent and straight configurations. The purpose of the FtsA ring state class is to keep track of the state of the FtsZ ring across time steps.

[0218] In an illustrated embodiment, the FtsZ ring is represented as a polygon of FtsZ filaments. As the circumference of the midline of the cell decreases with time, the number of edges in the inscribed FtsZ polygon decreases. The FtsZ filaments are of a fixed length (Fl) of 40 nanometers (nm, $1\text{ nm} = 10^{-9}$ meters). The number of edges (NE) in the polygon is determined by the maximum number of edges of length Fl that can be inscribed in the given cell circumference. The use of a fixed filament length is a simplification. This state class computes the integer number of edges NE in the FtsZ polygon using the inscribed polygon formula and the current cell midline diameter (CD) as given by Equation 1.

$$NE = \text{rounddown} \{ \pi / \arcsin (1/CD) \} \quad (1)$$

Where rounddown is a function that rounds the value of its argument to the greatest lower integer.

[0219] The Cytokinesis sub-model determines how many straight and/or bent filaments reside at each edge of the FtsZ polygon. At each edge, any of the following occupancies are possible: 1 straight filament; 2 straight filaments; 1 bent filament; 2 bent filaments; 1 bent filament and 1 straight filament; 1 bent filament and 2 straight filaments. The FtsZ Ring state class keeps track of the filament occupancy of each edge. As the filament occupancy in a time step is highly dependent on the occupancy at the previous time step, the tracking of the state of the FtsZ ring is considered important in the time evolution of Cytokinesis.

[0220] FIG. 14 is a block diagram that illustrates example fields in the Cytokinesis state variable class 140, according to an embodiment. The copy count of a particular cytokinesis protein, such as the number of FtsZ edges, is indicated in field 1401. For each copy, a cytokinesis protein copy field 1310 indicates an identifier (ID) for the copy in field 1411 and the cytokinesis phase (e.g., number of bent or straight edges or both) of the copy in field 1413, among other properties indicated by ellipsis to the right of field 1413. The properties of other copies, as well as the properties of other species of cytokinesis proteins, are indicated by ellipsis below field 1400. The class 1400 also includes class methods 1490.

[0221] In an illustrated embodiment, the cytokinesis protein state class is represented by a FtsZ Ring state class. Integration with other states and sub models are as follows. A method 1409 of this class reads the diameter of the cell's midline (*CD*) from the Geometry state class and computes *NE* using Equation 1. The Cytokinesis sub-model reads the number of FtsZ ring edges (*NE*) and the filament phase at each edge from the FtsZ Ring state class. The Cytokinesis sub-model updates the filament phase at each edge and stores the updated value in the FtsZ Ring state class.

[0222] In an illustrated embodiment, the values of the field are initially set to zero, because a FtsZ ring is not assembled at the beginning of the simulation.

3.10 Metabolic reaction state variable

[0223] Cells grow by importing nutrients from the external environment and using those nutrients to construct macromolecules. The Metabolism sub-model models the dynamics of the transport and chemical reactions which provide the metabolic building blocks required for macromolecular synthesis and drive cellular growth. *M. genitalium*, for example, is modeled with metabolic 645 reactions. The Metabolic Reaction state class records the instantaneous flux of each metabolic reaction in reactions per second as a floating point vector. This state also records the instantaneous cellular growth rate predicted by the Metabolism sub-model in cells per second as a floating point scalar.

[0224] FIG. 15 is a block diagram that illustrates example fields in the Metabolic Reaction state variable class 1500, according to an embodiment. The number of different metabolic reactions is indicated in field 1501. For each metabolic reaction species, field 1510 indicates a unique identifier (ID) for the reaction in field 1511, a description of the reaction in field 1513, such as the stoichiometry of the reaction *Mmr*, which indicates the number of particular metabolite *m* in reaction *r*, and the calculated flux in field 1515, among other properties indicated by ellipsis. Field 1510 is repeated for each different reaction species, as indicated by ellipsis under field 1510. One or more class methods 1590 are included in state class 1500.

3.11 Metabolite state variable

[0225] Cells use nucleic and amino acids to synthesize DNA, RNA, and protein. Cells also use ions and other prosthetic groups to stabilize macromolecules. In addition, cells use small molecule bonds and gradients to store energy and drive cellular processes. In particular, cells drive many energetically unfavorable reactions through hydrolysis of the high energy intermediates ATP and Guanosine-5'-triphosphate (GTP). Furthermore, cells use coenzyme functional moieties to facilitate chemical catalysis. Additionally, cells use small molecules such as Ca^{2+} , ATP, and guanosine pentaphosphate or tetraphosphate [(p)ppGpp] for communication and regulation. Still further, cells use small molecule antibiotics to defend against predators and attack prey.

[0226] In an illustrated embodiment, the Metabolic state variable class represents the copy number of each metabolite in each of 3 compartments—cytosol, membrane, and extracellular space—as an integer matrix. FIG. 16 is a block diagram that illustrates example fields in the Metabolite state variable class 1600, according to an embodiment. The number of metabolite species is indicated in field 1601, and for each metabolite species field 1610, and subsequent fields 1610 indicated by ellipsis below field 1610, hold information about each metabolite species. A unique identifier for the metabolite species is indicated in field 1611, a description, such as the chemical formula, is indicated in field 1613. The copy counts for the species in the external conditions compartment is indicated in extracellular count 1615, in the membrane compartment in field 1617, and in the cytosol compartment in field 1619. Metabolite class methods 1690 are also included in class 1600.

[0227] The Metabolite state class is integrated with other state classes and sub-models as follows. At each time step of the simulation, a Simulation object (e.g., method 500 depicted in FIG. 5A) simulates a cell culture incubator which maintains the extracellular partial pressures of carbon dioxide and oxygen by setting the dissolved extracellular copy numbers of these gases. To satisfy the assumption that each process is independent on the time scale of the time step, and because many metabolite species participate in multiple processes, the Simulation object also allocates shared metabolites among processes at each time step,

during step 530. Cytosolic- and membrane-localized metabolites are included in the cell mass calculated by the Mass state.

[0228] The Metabolism sub-model models the import of extracellular nutrients into the cytosol and membrane, the conversion of those metabolites into the precursors of DNA, RNA, protein, and lipids, and the export of byproducts into the extracellular space. The exchange rate of each metabolite of the flux-balance analysis metabolic model is limited by its extracellular copy number. In an illustrated embodiment, 23 additional sub-models access and modify the Metabolite state, primarily drawing cytosolic metabolites to support macromolecule synthesis. Four processes do not directly interact with the Metabolite state: Host Interaction, Macromolecular Complexation, Special Organelle Assembly, and Transcriptional Regulation.

[0229] Values are initialized as follows. After the Mass state initializes the total cell mass, the Metabolite state initializes the total cytosolic and membrane copy numbers of each metabolite according to the calculated chemical composition. Additionally, after the Geometry state initializes the volume of the extracellular compartment, the Metabolite state initializes the extracellular copy number of each metabolite according to the reconstructed medium composition. Following Metabolite state initialization, the Chromosome, RNA, Protein Monomer, Protein Complex, Transcript, and Polypeptide states initialize the copy number of each macromolecule species, and decrement the metabolite copy numbers to maintain the initial cell mass.

[0230] In the example embodiment for *M. genitalium*, the chemical compositions of *M. genitalium* and the extracellular medium were initially reconstructed based on the experimentally characterized compositions of *M. genitalium* and SP-4 medium. The cell and medium composition were both supplemented to support the metabolic demands of the 28 sub-models, and included 722 metabolite species that serve many important functions in over 1,100 chemical reactions across three compartments—cytosol, membrane, and extracellular space. The empirical formula and structure of each metabolite was curated based on an extensive review of the primary literature including two genome-scale metabolic models of *M. genitalium* and *E. coli*, and the databases BioCyc, ChEBI, Delta Mass, FindMod, KEGG,

LIPID MAPS, Modomics, PubChem, RESID, and UniMod. The molecular weight, van der Waals volume, the isoelectric point pI, logP, and logD of each metabolite was computed using ChemAxon Marvin. The regulatory properties of several antibiotics were reconstructed from the primary literature and the database DrugBank. The molecular composition of *M. genitalium* was reconstructed based on an extensive review of the primary literature.

[0231] During model testing and training, the actual phenotype was determined by growing *M. genitalium* in culture, so the metabolite composition of the culture is incorporated into the simulations. *M. genitalium* was cultured in complex *Spiroplasma* medium #4 (SP-4 medium) for all experiments presented here. Accordingly, the chemical composition of the *in silico* external environment modeled that of SP-4 medium. Because the chemical composition of SP-4 medium is undefined, the composition of *in silico* medium was reconstructed based on the characterized composition of each SP-4 medium component, and supplemented with additional metabolites to support *in silico* growth. Addition of supplemental metabolites was guided by the Metabolism sub-model and by the *M. pneumoniae* minimal medium.

3.12 Geometry state variable

[0232] Binary cell division is modeled as the cell varying from a spherical to short rod shape. The cell is modeled as a cylinder with two hemispherical caps, and growth is modeled in the cylinder length. Once cell pinching commences at the midline of the cell, the shape and size of a “septum region” is also modeled. The Geometry state class calculates and keeps track of the physical shape of the cell including its width, length, volume, and surface area. This state also keeps track of the progress of cytokinesis, and the completion of cytokinesis is the trigger to end the entire simulation.

[0233] Geometry state uses a set of geometric equations to calculate the shape of the cell, given the width (CW), density ($C\rho$), and cytokinesis progress of the cell. The state methods provide for the calculations of cell length (Cl), volume (CV), surface area (CA), and progress of cell pinching. The geometric representation involves an assumption that the cell density ($C\rho$) and cell width (CW) are constant across the cell cycle. Various sources have indicated that the volumetric density of a cell remains constant over time. Several sources have

described the width of a rod-shaped bacterial cell as remaining approximately constant across the cell cycle and across cell divisions. The cell width is calculated based on the initial cell mass (CM_0) and density and the assumption that the cell is a sphere. The geometric equations 2 through 8 use the fixed width and fixed cell density assumptions to calculate all the other aspects of the cell geometry. The mass (CM) and density ($C\rho$) enable calculation of the volume (CV) at all time points as given by Equation 2.

$$CV = CM / C\rho \quad (2)$$

[0234] In an illustrated embodiment, before pinching begins the cell geometry is approximated as two hemispheres separated by a cylinder.

$$\begin{aligned} CV &= \text{two hemispheres} + \text{cylinder} \\ &= \pi CW^3 / 6 + \pi CW^2 lc / 4 \end{aligned} \quad (3)$$

$$Cl = CW + lc \quad (4)$$

$$SA = \pi CW^2 + \pi CW * lc \quad (5)$$

Where lc is the length of the cylinder between the hemispheres and is initially equal to zero, and can be stated in terms of the cell volume (CV) based on rearranging Equation 3 and substituted into Equation 4.

[0235] Once cell pinching commences at the midline of the cell, there is a “septum region” as well. FIG. 17A is a block diagram that illustrates example cell geometry with a cell division septum region, according to an embodiment. The cell retains the two hemispheric caps, the cylinder is split in half and a septum region is inserted at the cell center line. The distance from the cell center to the edge of the septum region is ls . Here the cylinder length (lc), is the combined length of the two separated cylinders. The septum length (ls) is the length from the cell midpoint to the edge of the septum region. This septum length is calculated from the “pinched diameter” property that is calculated in the Cytokinesis sub-model. Each half of the septum volume is calculated as the area of two quarter circles (radius: ls) and one rectangle (width: ls , height: $CW - 2ls$), integrated around the midline cylindrically. We use the cell volume to calculate the cell length lc and surface area CA .

Once the pinched diameter is zero, the Geometry state records that the cell has divided, and this determines the end point of the simulation.

$$\begin{aligned} CV &= \text{two hemispheres} + \text{cylinder} + \text{septum region} \\ &= \pi CW^3/6 + \pi CW^2 lc/4 + \pi ls \{ (8ls^2 - 4ls CW + CW^2) + \pi ls (CW - 2ls) - 4ls^2/3 \} / 2 \end{aligned} \quad (6)$$

$$Cl = CW + lc + 2ls \quad (7)$$

$$SA = \pi CW^2 + \pi CW * lc + 4\pi ls(CW - ls) \quad (8)$$

The cell length lc can be stated in terms of the cell volume (CV) based on rearranging Equation 6 and substituted into Equation 7.

[0236] FIG. 17B is a block diagram that illustrates example fields in the Geometry state variable class 1700, according to an embodiment. The cell length lc is indicated in field 1711, width CW in field 1713, volume CV in field 1715, surface area CA I field 1717 and progress of cytosol, e.g., septum length ls , in field 1719. Other fields are indicated by ellipsis. On or more Geometry state class methods 1790 are included in Geometry state class 1700.

[0237] The Geometry state variable class interacts with other classes and sub-models as follows. The Geometry state class obtains the initial mass of the cell (to determine the cell width) from the Mass state class. It also obtains the mass of the cell from the Mass state class at all time points to determine the cell volume. The Metabolism, Protein Activation, Replication Initiation, and FtsZ Polymerization sub-models all obtain the cell volume from the Geometry state class. The Cytokinesis sub-model class retrieves the pinched diameter ls from the Geometry state class, and sends back the updated pinched diameter.

[0238] In some embodiments, initial conditions are determined as follows. The initial cell volume is calculated based on the initial cell mass ($CM0$) and the fixed density (Cp). The initial cell width $CW0$ is calculated from the volume and the assumption that the cell is a sphere. All of the other parameters describing cell geometry are evaluated from the mass (CM), volume (CV), density (Cp), and width (CW).

[0239] For the *M. genitalium* example whole-cell model, the cell density (Cp) used is that of *E. coli*, which has been estimated as 1100 grams per liter (g/L). The initial cell mass ($CM0$) was fit to result in the measured *M. genitalium* cell width (CW) of 200 nm.

3.13 Host state variable

[0240] A cell hosted by another organism can change the state of the host, such as by an eliciting an immune response by the host to one or more proteins in the membrane or secreted by the cell. Modeling such a response is supported by the properties stored in the Host state variable class. The Host state class represents the instantaneous configuration of the host. FIG. 18 is a block diagram that illustrates example fields in the Host state variable class 1800, according to an embodiment. A host interaction is indicated in field 1811 and additional interactions are indicated in additional fields represented by ellipses. The class 1800 includes one or more Host state class methods 1890.

[0241] None of the other 15 state classes directly interacts with the Host state class. The Host Interaction sub-model completely determines the initial conditions and temporal dynamics of variables represented by this state. The values of properties in the Host state class are initialized to the steady-state of the host interaction sub-model.

[0242] The *M. genitalium* terminal organelle is believed to mediate attachment to the human urogenital epithelium and enable its parasitic lifestyle. Upon attachment, lipoproteins are believed to activate host Toll-like receptors (TLRs) 1, 2, and 6, the host transcriptional regulator NF- κ B, and ultimately the host inflammatory response. In this example embodiment, the Host state variable class uses six Boolean variables to represent the instantaneous configuration of the human host: Adherent (true if *M. genitalium* is attached to its human host and false otherwise); TLR 1 activation (true if TLR receptor 1 is active and false otherwise); TLR 2 activation (true if TLR receptor 2 is active and false otherwise); TLR 6 activation (true if TLR receptor 6 is active and false otherwise); NF- κ B activation (true if NF- κ B is active and false otherwise); and Inflammatory response activation (true if the host inflammatory response is active and false otherwise).

3.14 Mass state variable

[0243] Cells are composed primarily of water, DNA, RNA, and protein enclosed in a bilipid membrane. The Metabolite, Chromosome, RNA, Protein Monomer, and Protein Complex states represent the detailed molecular composition of the cell. The Mass state

calculates the total cell mass from those states. The total cell mass is used as a proxy for membrane surface area in several processes which are based on mathematical models originally developed outside the whole-cell modeling framework. The Mass state calculates the total cell mass. The total cell mass is equal to the sum of the masses represented by the Chromosome, RNA, Protein Monomer, Protein Complex, Transcript, Polypeptide, and Metabolite states.

[0244] FIG. 19 is a block diagram that illustrates example fields in the Mass state variable class 1900, according to an embodiment. The water mass is indicated in field 191, the chromosome mass in field 1913, the RNA mass in field 1915, the transcript mass in field 1917, the polypeptide mass in field 1921, the protein monomer mass in field 1923, the protein complex mass in field 1925, the metabolites mass in field 1931 and the bilipid membrane mass in field 1933, among other fields indicated by ellipsis. Though some of these values are redundant with information in other state classes, in the illustrated embodiment, these fields are included in the Mass state for computational efficiency. One or more mass state class methods 1990 are also included in the class 1900.

[0245] The interaction of the Mass state class with other state classes and sub-models is as follows. The Mass state interacts with the Geometry state which calculates the cell shape, surface area, and volume from the calculated cell mass and observed density. None of the sub-models directly modify the Mass state. The Metabolism sub-model uses the cell mass to bound the rates of exchange reactions. The Replication Initiation sub-model models DnaA-ADP reactivation to DnaA-ATP as a function of membrane lipid mass. The Metabolism and Replication Initiation sub-models use the cell mass rather than the cell volume because these processes are based on mathematical models that were originally developed outside the whole-cell modeling framework.

[0246] The values of one or more fields are initialized hierarchically. First, the total cell mass is initialized according to a normal distribution with mean and standard deviation appropriate for the organism (e.g., a mean of 13.1 femtograms, fg, $1 \text{ fg} = 10^{-15}$ grams, and standard deviation 0.66 fg for *M. genitalium*). The mean initial cell mass was set to the reconstructed value. The variance was fit to match that of the predicted final cell mass

following cell division. Second, the total cell mass is divided among individual metabolites according to the reconstructed chemical composition. Third, the Chromosome, RNA, Protein Monomer, Protein Complex, Transcript, and Polypeptide states are initialized as functions of the metabolite copy counts, and the metabolite copy counts are decremented to maintain the initial cell mass.

[0247] In the example embodiment, the mass and chemical composition of *M. genitalium* were reconstructed based on an extensive review of the primary literature and fit to match the 28 sub-models. First, the average initial *M. genitalium* cell mass (13.1 fg) and dry mass (3.93 fg) were self-consistently calculated assuming a spherical geometry with 200 nm diameter, 1.1 gm/l density, and 70% water composition by mass. Note, this is comparable to the 18.9 fg *M. genitalium* cell mass reported. Second, the molecular composition of the dry mass was hierarchically reconstructed: 1. The *M. genitalium* dry mass was divided into eight classes—DNA/dNMPs, RNA/NMPs, protein/amino acids, lipids, carbohydrates, polyamines, vitamins & cofactors, and ions. 2. The fractional mass and molecular composition of each class was reconstructed from the literature. 3. The complete molecular composition of *M. genitalium* was assembled. 4. The dry mass composition was fit to match the 28 sub-models.

[0248] The mean initial cell mass was fit to the reconstructed value. The variance was fit to match that of the predicted final cell mass following cell division. Cellular composition is the balance of the production and usage of molecules by cellular processes. Consequently, it was necessary to reconcile the reconstructed cellular composition with that predicted by the 28 sub-models. First, the DNA dry mass fraction, including both chromosomal DNA and free dNTP, was increased 322% to be consistent with the predicted mass of the chromosome and the free dNTP pool. The predicted chromosome mass accounted for the observed chromosome sequence, predicted modifications, and predicted time-average copy number from Replication, Replication Initiation, and Metabolism sub-models. The predicted free dNTP pool accounted for the free dNTPs needed support the observed nine-hour *M. genitalium* mass doubling time. Second, the RNA dry mass fraction, including both RNA and free NTPs, was increased 39% to be consistent with the amounts of mRNA, rRNA, sRNA, and tRNA desirable to support the observed nine-hour mass doubling time. The NMP

composition of the RNA fraction was set equal to that predicted by the expression, sequence, and modification of each RNA species (as implemented in the Transcription, RNA Processing, and RNA Modification sub-models). Third, the vitamin & cofactor and ion dry mass fractions were increased to match the amounts of vitamins, cofactors, and ions needed for protein folding and catalysis (as implemented in the Protein Folding sub-model). Finally, the protein dry mass fraction, including both proteins and free amino acids, was decreased 23% to offset the increased DNA and RNA mass fractions. The amino acid composition of the protein fraction was set equal to that predicted by the expression, sequence, and modification of each protein species (as implemented in the Translation, Protein Processing I, Protein Processing II, and Protein Modification sub-models).

3.15 Stimulus state variable

[0249] Cells live in complex environments where they are exposed to physical and chemical stresses including antibiotics, radiation, heat and cold. Furthermore, manipulation of the external environment is a powerful tool for discovering new biology. This state represents the status of properties of the external environment that may affect one or more compartments.

[0250] In an illustrated embodiment, the Stimulus state class represents ten properties: the temperature in °C; six types (wavelengths) of radiation in units of Watts per square meter (Wm^{-2}) or greys per second (Gys^{-1}); and three Boolean-valued stress conditions.

Computationally, each property is implemented as a floating point scalar.

[0251] FIG. 20 is a block diagram that illustrates example fields in the Stimulus state variable class 2000, according to an embodiment. The temperature is indicated in field 2011, a first radiation type in field 2013, a radiation level in field 2015 for the radiation type specified in field 2013, and ellipses indicating other radiation types. Stress conditions are indicated in field 2021 for pH induced stress, and other stress conditions are indicated in field 2023 and 2025 and other fields indicated by ellipsis. More stimulus conditions are represented by other fields indicated by ellipsis below field 2021. On or more Stimulus state

class methods 2090 are also included in class 2000. In an illustrated embodiment, the stress factors of Table 4 are included in the Stimulus state variable class.

Table 4. Stimulus state variable fields.

Field name	description
temperature	temperature
stimulus_gluconate	gluconate
stimulus_thiolStress	thiol stress
stimulus_ironStress	iron stress
UVB_radiation	UV-B radiation (4 eV)
electron_radiation	electron radiation (1.5 keV)
alpha_radiation	alpha particle radiation (4 MeV)
proton_radiation	proton radiation (1 MeV)
gamma_radiation	gamma radiation (1.2 MeV)
UVA_radiation	UV-radiation (365 nm)

[0252] This state does not directly interact with any of the other 15 states. The Geometry, Metabolite, Protein Monomer, Protein Complex, and Host states represent additional properties of the extracellular environment. The Geometry state represents the volume of the external environment. The Metabolite state represents the copy counts of extracellular metabolites including antibiotics. The Protein Monomer and Protein Complex states represent the copy counts of extracellularly localized proteins. The Host state represents six properties of the host urogenital epithelium. Three sub-models—Protein Activation, DNA Damage, and Metabolism—access the Stimulus state. The Protein Activation sub-model regulates the activity of four transcription factors as functions of temperature and three stress conditions. The DNA Damage sub-model models the rates of several types of DNA damage as functions of radiation. The Metabolism sub-model models the generation of hydroxyl radicals from water and γ -radiation. None of the sub-models modify the 10 properties of the external environment.

[0253] In an example simulation of the *M. genitalium* example model, temperature was set at 37 degrees C, electron radiation at 2×10^{-11} Gy/s and gamma radiation at 2.8×10^{-11} Gy/s. The contents of the other state variable fields were set to zero.

3.16 Time state variable

[0254] The physical, chemical, and biological processes relevant to cell physiology span a wide range of time scales. By selecting an intermediate time step short compared to the slower processes and long compared to the fastest processes, the whole-cell modeling becomes tractable. The effects of the faster processes are averaged over the time step. In some embodiments, the time step was determined as described in method 400 depicted in FIG. 4 at step 411. Using similar reasoning, it was found that a time step of about 1 second is effective in the example whole-cell modeling of *M. genitalium*, which can be approximated as a well-mixed system at this time scale. This state represents the time elapsed from the start of the simulation in seconds as a single integer variable.

[0255] FIG. 21 is a block diagram that illustrates example fields in the Time state variable class 2100, according to an embodiment. Elapsed time is indicated in field 2111. Other fields are indicated by ellipses. One or more Time state variable class methods 2190 are included.

[0256] The Time state directly interacts with only three parts of the simulation: The Simulation model 500 linearly advances time in even time increments (e.g., 1 second increments for the example organism). If the cell has not yet divided, each simulation terminates at a predetermined maximum time approximately equal to 150% of the observed mean mass doubling time of the organism (for example at of 50,000 s for *M. genitalium*). The concentrations of extracellular metabolites and the values of extracellular stimuli can be functions of elapsed time (although held constant in the example simulations). None of the processes or other states depends directly on the Time state.

4. Cell process sub-model classes

[0257] In this section are presented the attributes and methods of the 28 cell process classes (called simply “sub-models” in the following for brevity). Although data structures and fields are depicted in many of the diagrams in this section as integral blocks in a particular

order for purposes of illustration, in other embodiments, one or more data structures or messages or fields, or portions thereof, are arranged in a different order, in the same or different number of data structures or databases in one or more hosts or messages, or are omitted, or one or more additional fields are included, or the data structures and messages are changed in some combination of ways.

[0258] Throughout this description indexes *i, j, k, l, m, n*, etc. are used repeatedly and are not to be given a particular meaning. The variable letter preceding the index is also used repeatedly and must be interpreted in the light of the local context. In the accompanying flowcharts, an attempt has been made to reduce the repetitive reuse of the same indexes and variable letters for different properties, to limit confusion. For example, in these flowcharts, the indices *i, j, k* are reserved as much as is convenient for nucleotide, strand and chromosome copy, respectively; and variable names are capital italicized letters. However, the descriptions in this section and the instruction listings in some figures reuse indices and variable letters more heavily.

4.1 DNA Condensation sub-model

[0259] Cell chromosomes are compacted (e.g., bacterial cell chromosomes are compacted 10^4 -fold) in volume in order to physically fit inside a cell. Compaction is also advantageous to support various cellular processes including chromosome replication, cell division, and chromosome segregation to opposing poles of the cell. Cells, such as bacteria, employ several mechanisms to compact DNA including “clamping” of the DNA by structural maintenance of chromosome (SMC) proteins, DNA supercoiling, macromolecular crowding, and charge neutralization. DNA Supercoiling is modeled in a separate sub-model. The Condensation sub-model explicitly models DNA clamping by SMC complexes.

[0260] FIG. 22A is a block diagram that illustrates features of DNA condensation, according to an embodiment. SMC complexes 2210a and 2210b (collectively referenced hereinafter as SMC complexes 2210) consist of an SMC core protein 2211 (e.g., product of gene MG298) and segregation and condensation proteins A 2212 and B 2213 (e.g., ScpA of gene MG213, ScpB of gene MG214). Together, the SMC complex 2210 is a “V” shaped

structure (with a head and two legs) that induces positive supercoils in double stranded DNA 2290. The complexes are believed to work with a lock and key mechanism in which first DNA is looped around the legs of the SMC complex, and then an ATP is hydrolyzed to ADP to bind between the two tails to lock the SMC complex in place. The complexes 2210 bind and clamp the DNA at corresponding different locations, causing many loops in the DNA and resulting in DNA compaction. The loops 2222 around each leg occupy 90 base pairs (bp). A loop 2224 of about 450 bp forms between the two SMC complex legs. Further, it has been inferred that there is about one SMC complex 2210 bound for an average spacing $S = 7130$ bp of DNA. Herein, base pairs on a double strand and nucleotides are used interchangeably when specifying position or distance along a DNA molecule.

[0261] Given that the DNA footprint of an SMC complex is much less than its average spacing, if one simply applied a $1/S$ probability of each base being bound at each time step, over time, given an SMC abundance, one would see SMCs bound at intervals much less than S_{av} . An SMC complex threshold spacing parameter (l) 2220 prevents this phenomenon by inhibiting SMC complex binding where the gap between existing SMC complexes 2210 on the DNA 2290 is already much less than the desired spacing. Biologically, this parameter represents the notion that SMC complex binding is not only determined by SMC complex abundance, but also the writhe and physical properties of the DNA that prevent SMC complex from binding too close to each other. This spacing parameter 2220 was fit such that the average SMC complex spacing S in the model is 7130 bp. The fit of this parameter was assessed by various tests to assure an average SMC spacing S of 7130 bp.

[0262] SMCs are bound at an average spacing (S) of 7130 bases. Since it is unknown whether SMC complexes bind to specific DNA motifs, at each time step, SMC complexes are bound to random positions on the chromosomes. The binding is weighed by calculated probabilities (P) of a free SMC complex binding to each accessible double-stranded DNA base. The probability distribution is calculated as a step function: bases within a threshold spacing parameter (l) from another SMC complex have a zero probability of being bound, and bases beyond the threshold spacing parameter distance from other SMC complexes bind

at a probability of $1/S$. SMC complex disassociation from the DNA occurs upon interaction with other DNA binding proteins and is handled by the Chromosome state class.

[0263] The DNA Condensation sub-model models the contribution of SMC proteins to chromosome condensation. SMC-related DNA condensing could be considered along with the effects of topoisomerases in some embodiments, but the effect of the topoisomerases is modeled by tracking the effect of each topoisomerase activity on the DNA linking number (in the DNA supercoiling sub-model). Because SMCs do not act by strand-passing events (the causing of a nick in the DNA, enabling two double-stranded DNA regions to pass through each other, and the re-ligation of the DNA), and since the exact effect of SMC complex activity on the DNA linking number is not known, SMC condensation is considered as compacting the DNA at a different level and therefore it is modeled independently of the DNA linking number calculations in the DNA Supercoiling sub-model. Macromolecular crowding and charge neutralization are not presently included in any sub-model of the illustrated embodiment, but these processes are modeled in one or more new or extant sub-models in other embodiments. An advantage of computational efficiency is achieved by omitting these other processes, without apparent hindrance to obtaining good agreement with experiments, as described in more detail in another section.

[0264] To initiate the Chromosome state, before the start of a simulation, the SMC complex binding calculations described above are iteratively run to bind SMC complexes to the first chromosome until a steady state is reached, in which no more SMC complexes can be bound.

[0265] FIG. 22B is a flowchart that illustrates an example method 2250 for a DNA Condensation sub-model, according to an embodiment. In step 2251, parameters are set for structural maintenance of chromosome (SMC) protein complex. The parameter S indicates average separation of SMC complexes on chromosome = 7130 nt; l indicates closest spacing of SMC complex (fit); footprint of SMC on DNA = 630nt; footprint of SMC arm on DNA = 90nt; DNA loop length between SMC arms = 450nt.

[0266] In step 2261, a maximum number of SMC that can bind during time step is determined. E.g.: SMC binding limit is the minimum of: a count of free SMC complexes and a count of available ATP molecules.

[0267] In step 2271, SMC complexes are randomly bound to a chromosome. In the illustrated embodiment, step 2271 includes steps 2273 through 2277. In step 2273, it is determined based on data in Chromosome state, free bases where SMC complexes can bind that are more than l bases from a different, already bound SMC complex. In step 2275, sites are randomly chosen for binding SMC complexes up to the limit determined in step 2261. For any site, the probability of binding an SMC complex is $1 / (S - 2l)$. If the SMC binds, then in step 2277 an SMC-ATP complex is formed and hydrolyzed to an SMC-ADP complex. The copy count of free SMC complexes I reduced and the metabolites involved in hydrolysis are updated (e.g., decrease H₂O count and ATP count and increase ADP count and H⁺ count).

[0268] Thus, the DNA Condensation sub-model reads from and writes to the Chromosome state class. It reads in the regions of DNA that SMC complexes can bind to, and writes back the positions of SMC complexes on the chromosome.

[0269] *M. genitalium* has the machinery to perform all of these levels of DNA compaction and was modeled using the method 2250 depicted in FIG. 22B.

4.2 DNA Segregation sub-model

[0270] DNA replication produces two chromosomes that are catenated, or connected like links of a chain. Before cell division can take place, these chromosomes need to migrate to

opposing sides of the cell and be decatenated. Chromosome segregation is believed to result from both entropic factors and enzymatic activity.

[0271] The Chromosome Segregation sub-model assumes that the chromosomes entropically segregate during replication. That is, it is unfavorable for the two copies of replicated DNA to exist too close to each other. Thus as the replication fork moves, the replicated DNA starts to migrate towards the poles.

[0272] Enzymatic segregation activity is also modeled. Five chromosome segregation proteins are modeled, including a nucleotide binding domain (e.g., CobQ/CobB/MinD/ParA from MG470), an MraZ protein (e.g., MraZ from MG221), a GTP binding protein (Era from MG387), a GTPase (Obg from MG384), and Topoisomerase IV (ParE from MG203, ParC from MG204). Another segregation protein, FtsZ, is accounted for in the Cytokinesis sub-model. These proteins assist in the chromosomal migration towards the cell poles and decatenate the chromosomes following the completion of replication. However, there is no detailed understanding of the function of these proteins. Further, the specific kinetics, and metabolic costs of these proteins are not known.

[0273] The DNA Segregation sub-model models enzymatically catalyzed sister chromosome separation and decatenation. Entropically driven sister chromosome separation is not well understood and is not presently modeled. An advantage of computational efficiency is achieved by omitting these other processes, without apparent hindrance to obtaining good agreement with experiments, as described in more detail in another section.

[0274] Because the molecular biology of chromosome segregation is not well understood, it was decided to implement this process as a simple Boolean process: the chromosomes are regarded as segregated immediately after the following four conditions are met: the chromosome is replicated; the chromosomes are properly supercoiled within a given tolerance of superhelical density; there is at least one free and functional molecule of each of the five segregation proteins; and, there are enough available GTP molecules.

[0275] For parameter values, the energetic cost of segregation has not been experimentally characterized. The energetic cost is set to a nominal value of 1 GTP per segregation event. The superhelical density tolerance is also unknown, so the allowed range of DNA

superhelical density was set to be quite wide. The chromosome state is initialized with 1 chromosome with superhelical density within the acceptable tolerance.

[0276] FIG. 23A is a flowchart that illustrates an example method 2300 for a DNA Segregation sub-model, according to an embodiment. In step 2301, parameters for DNA segregation are set. The energy for segregation *Eseg* is set equal to the GTP cost of segregation (~ 1 GTP); σ_{tol} , the superhelical density tolerance is set (~ 0.1); σ , the equilibrium superhelical density is set (~ -0.06); segregation enzymes are indicated by their protein monomer identifiers (e.g., the proteins from genes MG470, MG221, MG387, MG384, MG204).

[0277] In step 2302 the state of the chromosome is checked to determine whether the chromosome is fully replicated. If not, the process ends without completing segregation. If so, then in step 2305 it is determined whether the chromosome winding is within range for segregation, e.g., between $\sigma \pm \sigma_{tol}$. If not, the process ends without completing segregation. If so, then in step 2307 it is determined whether there is at least one free copy of each of the segregation enzymes. If not, the process ends without completing segregation. If so, then in step 2309 it is determined whether there is sufficient energy (e.g., available GTP greater than *Eseg* demand for GTP). If not, the process ends without completing segregation. If so, then in step 2311 the chromosomes are marked as segregated, e.g., the status in field 616 is updated to segregated.

[0278] FIG. 23B is a list showing the relationship of the DNA Segregation sub-model to cell state variables, according to an embodiment. Thus the sub-model interacts with the Chromosome state class for the properties listed in FIG. 23B.

[0279] Many fewer segregation proteins are present in *M. genitalium* than that required for the detailed mechanisms for segregation described for other bacterial species. Thus in some embodiments, other segregation enzymes are added to the list of enzymes that need be present for segregation to be completed.

4.3 DNA Damage sub-model

[0280] DNA is not absolutely stable. Rather, DNA is susceptible to spontaneous modification including base loss and deamination. DNA is also susceptible to modification by many exotic agents including UV-A and UV-B radiation, α -, β -, and γ -radiation, oxygen, hydroxyl, carbon, and nitrogen radicals, and alkylating agents. These diverse DNA modification modes generate equally diverse DNA structures including missing and modified sugar-phosphates, missing and modified nucleotides, intra- and inter-strand cross links, and strand breaks. DNA damage is an evolutionarily important source of genetic variation. However, damaged DNA has also been shown to be deleterious to many processes including transcription and replication. Consequently, cells employ dedicated DNA repair machinery. The exact effect of DNA damage on physiology is not well characterized.

[0281] Because DNA damage is rare, this process models each DNA damage reaction as an independent Poisson process. The rate, ri , of each spontaneous DNA damage reaction i is given by the observed specific rate ki . The effective rate, ri , of each DNA damage reaction i caused by radiation j is given by the product of the experimentally observed specific rate, ki , and the radiation flux, sj . Similar rates are determined in some embodiments for chemically induced damage, caused by chemical agent n , given by the product of the experimentally observed specific rate, ki , and the concentration cn of chemical agent n .

[0282] The Chromosome state initializes all chromosome bases, fully methylated at R/M sites and otherwise unmodified. Several other processes including Transcription sub-models initialize the protein occupancy of the chromosome.

[0283] FIG. 24A is a flowchart that illustrates an example method 2400 for a DNA Damage sub-model, according to an embodiment. A DNA damage reaction n refers to damage to any nucleotide in the chromosome. So the specific reaction rate for damage to a nucleotide by reaction n is Kn . The amount of stimulus s associated with the reaction n is Ss and refers not only to radiation flux, but also chemical concentration of a chemical agent, or temperature, or any other real valued stimulus. For reactions n dependent on amount of stimulus, the effective rate $Rn = Kn * Ss$. For spontaneous damage not associated with a stimulus, $Rn = Kn$.

[0284] In step 2401, parameters are set for reaction n to stimulus s in stimulus state class. Zn = resultant base configuration of reaction n (e.g. Boolean values for gap, abasci, sugar-phosphate, base, cross-link and strand break modifications); Kn = specific reaction rates of reaction n ; Mmn = stoichiometry of metabolite m in reaction n . In step 2403 it is determined whether there is another reaction n to test. If not, the sub-model is completed and the method ends. If so, control passes to step 240 to determine if the reaction n is dependent on the amount Ss of stimulus s . If not, then in step 2407 the effective rate is set equal to the specific rate, and control passes to step 2411. If so, then in step 2409 the effective rate Rn is set equal to the product of the stimulus amount and the specific rate, $Rn = Kn * Ss$; and control passes to step 2411.

[0285] In step 2411, it is determined if the next base i in strand j of chromosome copy k is susceptible to reaction n . If not, control passes back to step 2403 for the next reaction, if any. If so, then in step 2413, the copy count Nm of metabolite m is retrieved from Metabolite state variable for every metabolite in reaction n . In step 2415, it is determined if there are insufficient metabolites for the reaction to occur, e.g., it is determined whether $Nm < Mmn$ for any metabolite m . If there are insufficient metabolites, control passes back to step 2403 for the next reaction, if any. If there are sufficient metabolites, then control passes to step 2421.

[0286] In step 2421 a Poisson distribution is used to determine whether damage is done to the current base in the current time step. For example, a random number X is selected from a Poisson distribution with occurrence rate = Rn and time interval = time step. In step 2423 it is determined whether the random number $X \geq 1$. If not, no damage is done to the base and control passes back to step 2411 for the next base. If so, then damage takes the form of one or more modifications to the base as indicated by Zn , e.g., one or more of gap, abasci, sugar-phosphate, base, cross-link and strand break modifications. Control passes back to step 2411 for the next base.

[0287] The Chromosome state represents the copy count, modification status, and protein occupancy of each chromosome. The Metabolite state represents the copy number of each metabolite. The Stimulus state represents the stimulus, such as fluxes of six types of radiation: α -, β -, and γ -particles, electrons, protons, and UV-A and UV-B in an example embodiment. The Metabolism sub-model models the generation of hydroxyl radicals. The DNA Repair sub-model models four DNA repair pathways: direct damage reversal, base excision repair, nucleotide excision repair, and homologous recombination. The DNA Repair sub-model also models DNA methylation at restriction/modification (R/M) sites. All DNA modifications except methylations of R/M sites are assumed to impede protein DNA-binding

[0288] FIG. 24B is a list of instructions that illustrates an example DNA Damage sub-model for *M. genitalium*, according to an embodiment.

4.4 DNA Repair sub-model

[0289] DNA is susceptible to several intrinsic and extrinsic modes of damage. Because damaged DNA is deleterious to many processes including transcription and replication, organisms have evolved specialized machinery to detect and repair damaged DNA. The machinery includes proteins serving as enzymes for detection and repair of damage, including damage sensor DisA and four DNA repair pathways: direct damage reversal (DDR), base excision repair (BER), global genomic nucleotide excision repair (GG-NER), and homologous recombination double strand break repair (HR-DSBR) and type II restriction/modification (R/M) system MunI to selectively degrade foreign DNA.

[0290] DisA recognizes strand breaks, base damage, and cross links. The mechanism of DisA damage recognition is not well understood. DisA is believed to signal DNA damage through the absence of the second messenger cyclic-di-AMP. In the absence of DNA damage, DisA cyclizes ATP to cyclic-di-AMP. In the presence of DNA damage, DisA is believed to bind DNA, adopt a catalytically inactive conformation, and via an unknown mechanism possibly involving cyclic-di-AMP and the transcriptional regulator Spo0A, to induce expression of genes.

[0291] The primary role of BER is to repair small patches of oxidized nucleobases. One BER pathway involves four steps. First, glycosylase Fpg(MG498) or Ung(MG097) hydrolytically cleaves the glycosidic bond between the damaged nucleobase and the DNA backbone, creating an abasic site. Second, two parallel subpathways cleave the phosphodiester bonds 3' and 5' to the abasic site, introducing a gap site. The first subpathway begins with cleavage of the 3' phosphodiester bond site by AP lyase Fpg and ends with cleavage of the 5' phosphodiester bond by 5'-deoxyribosephosphodiesterase Nfo. The second subpathway begins with cleavage of the 5' phosphodiester bond by 5'-AP endonuclease Nfo (MG235) and ends with cleavage of the 3' phosphodiester bond by 3'-(deoxyribose-5'-phosphate) lyase DnaN (MG001). Third, DNA polymerase DnaN restores the missing base using the template provided by the opposite strand. Finally, DNA ligase ligates the inserted base.

[0292] The primary role of NER is to repair bulky distortions in the shape of the DNA helix of up to 12-13 bases in length caused by UV radiation and nitricoxide. Compared to BER, NER employs less specific endonucleases and consequently has broader repair capacity. NER repairs DNA via a four-step mechanism. First, UvrABC (MG421, MG073, MG206) identifies a DNA lesion and cleaves the phosphodiester bonds 6-7 bases 3' and 5' to the lesion. Second, helicase PrcA excises the cleaved DNA. Third, DNA polymerase DnaN (MG001) restores the missing bases using the template provided by the opposite strand. Finally, DNA ligase ligates the inserted base.

[0293] HR repairs double strand breaks caused by ionizing radiation and stalled replication forks. HR also repairs strand gaps generated by the interaction of replication forks with

unrepaired lesions. Recombination repair is modeled as an eight-step process: 1. Initiation: 5'-3' exonuclease removes dNMPs from the 5' ends of the strand break, leaving 3' overhangs of at least 8 bases. 2. Strand exchange: RecA (MG339) catalyzes the formation of a Holliday junction between one of the damaged 3' overhangs and the undamaged homologous chromosome. 3. Polymerization: DnaN (MG001) polymerizes DNA guided by the undamaged chromosome template. 4. Ligation: LigA (MG254) ligates the newly produced DNA. 5. Second strand exchange: RecA catalyzes the formation of a second Holliday junction. 6. Holliday junction migration: RuvA and RuvB widen the distance between the two strand crossover points by moving the Holliday junctions to the preferred sequence 5'-[AT]TTN[GC]-3'⁵³². 7. Resolution: RecU nicks the DNA at the cross over points, creating four single strand breaks. 8. Ligation: LigA ligates the four strand breaks.

[0294] Organisms employ R/M systems to distinguish foreign from self DNA. These systems maintain self DNA in a fully methylated configuration by methylating self DNA during chromosomal replication and cleaving unmethylated foreign DNA which has not been exposed to self DNA methylases. MunI methylates the third base of both strands of the palindromic sequence 5'-CAATTG-3', producing N6-methyladenine.

[0295] The metabolic byproducts of DNA repair are salvaged by oligonucleases. Candidate oligonucleotide salvage genes include pcrA (MG244), mgpA (MG190), and polA (MG262). Of these genes, we assumed that mgpA is responsible for oligonucleotide salvage. Individual damaged nucleotides are further metabolized and exported.

[0296] The DNA Repair sub-model models four DNA repair pathways and methylation and restriction of MunIR/Msites. Because DNA repair is not well understood on the genomic level, this process makes several simplifying assumptions. First, this sub-model represents each step in DNA repair, methylation, and restriction as a separate reaction, and assumes that each reaction is independent. Consequently, the rate of each reaction is determined only by the configuration of the DNA. Second, this sub-model assumes that the mean arrive rate, v_i , of each reaction i is independently limited by (1) the copy counts of intracellular metabolites, m_j , and (2) the DNA repair enzyme copy numbers, e_j . Based on these assumptions, the function form of v_i is given by Equation 9.

$$v_i = \min \left(\left[\overbrace{\frac{m_j}{M_{ji}^s}}^{\text{metabolites}} \right], \left[\overbrace{\text{poissonRand} \left(\frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right] \right) \quad (9)$$

where M_{ji} is the stoichiometry of metabolite j in reaction i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in reaction i , and Δt is the simulation time step. This process also stochastically models the binding of free DisA to DNA lesions.

[0297] The Chromosome state initializes one set of chromosomes (e.g., one Chromosome for *M. genitalium*), modified only at R/M sites. Several other processes including Transcription initialize the protein occupancy of the chromosome.

[0298] FIG. 25A is a flowchart that illustrates an example method 2500 for a DNA Repair sub-model, according to an embodiment. Here a repair reaction is indicated by index r , a metabolite by index m and enzymes by index e . The mean arrive rate of repair reaction r is V_r . The copy count is indicated by variable N . In step 2501, parameters are set for each repair reaction r , including M_{mr} = stoichiometry of metabolite m in reaction r ; protein IDs of repair enzymes for each reaction r (e.g., DisA, DnaA, PrcA, DNAligase, RecA, RuvA, RuvB, RecU, LigA . . .); K_{er} = catalytic rate of enzyme e in reaction r .

[0299] In step 2503, the next repair reaction r is selected at random. If none, the method is complete and the sub-model ends. If a reaction remains, then in step 2505 it is determined whether there is a next randomly selected base susceptible to the repair reaction. If none remain, control returns to step 2503 to check the next reaction. If there is a base for the reaction, then in step 2511, copy counts N_m for metabolites and N_e for repair enzymes are retrieved from the cell state variables. In step 2513, the mean arrive rate V_r is determined based on Equation 9. In the revised notation, this is V_r = minimum of $\{N_m / \text{maximum of}$

[0, - Mmr] over all m ; or random number X over all e from a Poisson distribution with occurrence rate = (time step)* Ne / Ker . In step 2521 it is determined whether $Vr > 1$. If not, control returns to step 2505. If so, then in step 2523 the repair reaction r is run on the current base and the metabolite counts are updated and enzymatic capacity decreased. Control returns to step 2505 for the next base, if any.

[0300] The *M. genitalium* genome contains the DNA damage sensor DisA, the four DNA repair pathways described above, and the type II restriction/modification (R/M) system MunI. *M. genitalium* employs only one dedicated machine—DisA (MG105)—to recognize DNA damage. *M. genitalium*, however, does not have most of the downstream signaling machinery associated with *B. subtilis* DisA-dependent induction of expression of genes.

[0301] DNA ligation is the only DDR pathway employed by *M. genitalium*. DNA ligation is catalyzed by DNA ligase LigA (MG254) by a Nicotinamide adenine dinucleotide (NAD)-dependent mechanism. DNA ligase repairs single strand breaks as well as ligates DNA synthesized during replication, BER, NER, and HR.

[0302] *M. genitalium* BER repairs single damaged nucleobases. *M. genitalium* does not undergo long patch BER because *M. genitalium* does not have a flap endonuclease. *M. genitalium* BER repairs DNA via the four-step mechanism described above.

[0303] *M. genitalium* NER provides global protection against small DNA lesions. *M. genitalium* does not undergo transcription-coupled NER because *M. genitalium* does have the transcription-repair coupling factor mfd. *M. genitalium* NER repairs DNA via the four-step mechanism described above.

[0304] *M. genitalium* HR has a very reduced complement of recombination repair proteins, and follows the eight steps described above with modifications. No traditional initiation gene has been identified in *M. genitalium*; so, it was assumed that polI-like 5'-3' exonuclease (PolA, MG262) is the *M. genitalium* HR initiator. Additionally, HR is believed to be critical for *M. genitalium* antigenic variation.

[0305] *M. genitalium* contains a reduced R/M system consisting of the type I DNA recognition subunit EcoD (MG438) and the type II methylase MunI (MG184). EcoD is the DNA binding domain of a type I multimeric methylation and restriction complex which

recognizes the sequence 5'-TCARTTC-3'. Because *M. genitalium* contains only the DNA recognition subunit and not the methylation and restriction subunits, type I R/M was not modeled. Because *M. genitalium* does not contain a separate type II endonuclease and because type II R/M systems are often monomeric, it was assumed MunI also has restriction activity.

[0306] The mechanism of *M. genitalium* oligonucleotide salvage is not well understood. No gene has been identified which cleaves DNA oligonucleotides.

[0307] FIG. 25B is a list of instructions that illustrates an example DNA Repair sub-model for *M. genitalium*, according to an embodiment.

4.5 Supercoiling sub-model

[0308] DNA naturally exists at a certain level of helicity, and this level of helical density is important for the DNA's stability, its ability to fit in the cell, and its ability to bind proteins. DNA supercoiling can also have an effect on RNA transcription rates as the helicity of the DNA may make given genes more or less accessible.

[0309] For example, three topoisomerase proteins—DNA gyrase, Topoisomerase I, and Topoisomerase IV—transiently break a DNA strand to wind (Topoisomerase I) or unwind (Topoisomerase IV, gyrase) the DNA. The opposing actions of the topoisomerase enzymes help maintain a stable level of DNA helicity. This is especially important during replication because while the replication loops progress along the chromosome unwinding the DNA, the coils that previously existed in the DNA persist and accumulate downstream of the replication loop. This over-coiled region is relieved by the actions of topoisomerases.

[0310] DNA gyrase and Topoisomerase IV each require 2 ATP to induce 2 negative supercoils each time they act. Topoisomerase IV induces positive supercoils relatively rarely and its positive coiling activity is therefore not considered in some embodiments.

Topoisomerase I acts to induce positive supercoils. Gyrase, Topoisomerase IV, and Topoisomerase I act at rates of 1.2, 2.5, and 1 strand passing events per second, respectively.

[0311] The supercoiling of the DNA is quantified using a metric known as the DNA linking number, LK, defined as (# of base pairs)/10.5, where 10.5 is the number of bases per turn in a

relaxed double helix. The relaxed LK is designated *LKrelaxed*, and indicates how many complete turns are on the portion of DNA of interest. The current LK, designated *LKcurrent*, may deviate from the *LKrelaxed* due to the actions of topoisomerases and the progression of replication. The ΔLK is defined as the difference between the current level of DNA supercoiling and the relaxed level of DNA supercoiling.

[0312] The helicity of different regions on the DNA are modeled separately. Before replication, the entire chromosome is represented as one region. During replication, two additional regions are defined and accounted for: the replicated DNA upstream of the replication loops on each of the two chromosomes. FIG. 26A is block diagram that illustrates different regions for defining helicity on a replicating chromosome, according to an embodiment. During replication, the unreplicated region 2600 of DNA (region downstream of the two replication loops) is shrinking in terms of number of bases, and as a result, the *LKrelaxed* decreases. However, as the DNA replicative helicases unwind the DNA they push the existing coils into this region 2600, and there are more coils for fewer bases. This over-coiled region has an *LKcurrent* that is too high, and must be brought down towards the *LKrelaxed*. Gyrases and topoisomerases help bring the DNA back to the relaxed state by inducing negative supercoils. It is assumed that the newly replicated DNA in region 1 2601 and region 2 2602 is made in the relaxed state: $LK_{current} = LK_{relaxed}$, but the supercoiling enzymes act in this region as well, helping maintain equilibrium helicity. It is essential that the ΔLK in the region downstream of the replication loops be brought down to 0 by the end of replication. After replication is complete, only the two replicated regions 2601 and 2602 exist (two separate chromosomes). While in general, DNA supercoiling involves the regions described above, DNA damage and other processes can cause gaps in the DNA that result in additional regions.

[0313] The superhelical density of a DNA region, σ , is defined in Equation 10.

$$\sigma = (LK_{current} - LK_{relaxed}) / LK_{relaxed} \quad (10)$$

Experimentally it has been shown that the steady state superhelical density, σ_0 , is -0.06923 . The activity of topoisomerases depends on the σ of the DNA. Although there is likely a more complex relationship between the DNA supercoiling and enzyme activity, enzyme activity is

modeled as a set of step functions and logistic functions. FIG. 26B through FIG. 26D are graphs that illustrate the effect of superhelical density on the activity of certain DNA binding proteins, according to an embodiment. FIG. 26B is graph 2610 that shows the activity of Topoisomerase I as curve 2616. The horizontal axis 2612 represents superhelical density σ , which is dimensionless; and the vertical axis 2614 indicates activity in relative units. FIG. 26C is graph 2620 that shows the activity of Gyrase as curve 2626. The horizontal axis 2622 represents superhelical density σ , which is dimensionless; and the vertical axis 2624 indicates activity in relative units. FIG. 26D is graph 2630 that shows the activity of Topoisomerase IV as curve 2636. The horizontal axis 2632 represents superhelical density σ , which is dimensionless; and the vertical axis 2634 indicates activity in relative units.

[0314] It is known that the topoisomerases are unable to act beyond certain σ thresholds (gyrase can only act at σ above threshold designated T_{gyr} , Topoisomerase IV can only act at σ above threshold designated T_{TopIV} , and Topoisomerase I can only act at σ threshold designated T_{TopI}), so the activities of the topoisomerases outside of the thresholds are set to zero. However, the exact enzyme activity profiles within these bounds is unknown. Fitting a logistic function within the thresholds of gyrase and Topoisomerase I allows the maintenance of σ approximately near σ_0 . Topoisomerase IV is not active near σ_0 and therefore acts more rarely than the other topoisomerases. As a result, there was no need to assume a logistic profile for its activity. The proportion of full activity for each topoisomerase was obtained from the profiles. This proportion was multiplied by the maximal activity to determine the number of strand passing events for each topoisomerase for a given time step.

[0315] Further, the sub-model keeps track of enzyme processivity, or how long an enzyme can remain bound to the DNA and continue to act. Topoisomerase IV and DNA gyrase are highly processive and may perform several strand-passing events before falling off of the DNA. Topoisomerase I is not highly processive; it acts on the DNA and falls back off right away.

[0316] It has been shown that there is an effect of supercoiling on transcription rates, and experimental data in *E. coli* compares superhelical density to fold change of gene expression.

Expression fold changes at different superhelical densities have been calculated for many genes. However, in an example embodiment only the effects of supercoiling on the expression of the 5 supercoiling genes *gyrB*, *gyrA*, *parC*, *parE*, and *topA*, are included. These genes exist in 3 transcription units. The data for the three transcription units has been combined and approximated by three linear fits. In other embodiments, other genes and transcription units are included.

[0317] The activities of DNA coiling enzymes, such as gyrase, Topoisomerase I, and Topoisomerase IV, and the progress of replication loops can all affect the linking number (and superhelical density) of the DNA. The DNA Supercoiling sub-model evaluates the effects of all the free coiling enzymes in random order, determines what regions they can bind in, and randomly binds them to large enough open positions on the DNA. The linking number is then adjusted based on the actions of all the bound enzymes, and the usage of ATP is determined. The processivity of gyrase and Topoisomerase IV are also tracked so that they can be unbound from the DNA when appropriate.

[0318] The superhelicity of the chromosome is tracked separately for each different double stranded region. The exact base positions and length in base pairs, l , of each region is determined based on the progress of replication and positions of breaks in the DNA. If there are no breaks in the DNA, the following DNA regions exist. Before Replication: Region 1 includes unreplicated DNA. During Replication: Region 1 includes Replicated DNA, Chromosome 1 (upstream of replication loops, increasing in length over time); Region 2 includes Replicated DNA, Chromosome 2 (upstream of replication loops, increasing in length over time); Region 3 includes Unreplicated DNA (downstream of replication loops, decreasing in length over time). After Replication: Region 1 includes Replicated DNA, Chromosome 1; Region 2 includes Replicated DNA, Chromosome 2.

[0319] FIG. 26E and FIG. 26F are flow charts that illustrate an example method 2640 for a DNA Supercoiling sub-model, according to an embodiment. In step 2641, parameters values are set: DNA linking number in relaxed state per base = $1/10.5$; protein IDs of coiling enzymes (*gyrB*, *gyrA*, *parE*, *parC*, *topA*); kinetic rates K_e of enzymes; ATP hydrolyzed per

strand passing event E_e of enzymes; change in linking number δe by strand passing event of enzyme e ; and response curves for effect of superhelical density on gene expression.

[0320] In step 2643, the next region is selected. In step 2645, length of region LR and $LK_{current}$ are determined; and bound coiling enzymes are determined based on Chromosome state variable. In step 2647, superhelical density σ is determined using Equation 11a and 11b.

$$LK_{relaxed} = LR / 10.5, \text{ and} \quad (11a)$$

$$\sigma = (LK_{current} - LK_{relaxed}) / LK_{relaxed}. \quad (11b)$$

In step 2651 it is determined whether superhelical density is greater than an activity threshold (e.g., 0). If not, then in step 2653 unbind coiling enzymes not active below this threshold such as all parC and parE . In either case control passes to step 2661.

[0321] In step 2661, the next coiling enzyme still bound is found, if any. If not, control returns to step 2671. Otherwise, in step 2663, a random number X is determined from a Poisson distribution with occurrence rate = (time step)/(45 seconds), the rate at which some coiling enzymes naturally unbind. In other embodiments, the 45 seconds is replaced by a different time scale more appropriate for the natural unbinding time of the different coiling enzyme. If it is determined in step 2665 that $X > 1$, then unbind the coiling enzyme and get the next bound enzyme. Otherwise, the coiling enzyme is left bound and the next bound coiling enzyme, if any, is selected in step 2661.

[0322] In step 2671, the next free coiling enzyme that binds at superhelical density greater than zero is selected at random, if any. If found, and superhelical density is greater than zero, then the enzyme is bound to the DNA in step 2677 and the copy counts are updated, if it is determined in step 2675 that there is a length available on the DNA greater than the footprint of the enzyme. Otherwise the next free coiling enzyme, if any, is selected. When there are no more such free enzymes, control passes to step 2681.

[0323] In step 2681, the next copy of the remaining enzymes is selected, if any. If so, then in step 2683 the number of strand passing events SP_e by enzyme e is determined based on the probability P_e that enzyme e will act at current superhelical density, according to Equation 12.

$$SPe = \text{MIN} \{ Ce \text{ Ke } Pe; (\text{count free ATP})/Ee; (\text{count free H}_2\text{O})/Ee \} \quad (12).$$

In step 2685 $LK_{current}$ is updated by adding the product of δe and SPe . In step 2687, counts of metabolites ATP and H₂O consumed and metabolites ADP, PI and H⁺ produced are updated. In step 2689 all non processive enzymes, such as topA, are unbound. Control passes back to step 2681 to determine the next enzyme, if any.

[0324] If no other coiling enzyme remains, then in step 2691, σ for the region is updated and fold change in probability is determined based on response curves for effect of superhelical density on gene expression. Also, the polymerase binding probability field is updated in Chromosome state for genes in transcription unit with any of the coiling enzymes. In step 2693, it is determined if there is another region. If so, control returns to step 2643. Otherwise the method ends.

[0325] *M. genitalium* has the three topoisomerase proteins: DNA gyrase, Topoisomerase I, and Topoisomerase IV. The logistic function parameters (L_{gyr} , L_{topI}) were fit such that the equilibrium superhelical density, σ_0 , could be maintained while still allowing both gyrase and Topoisomerase I to act at the equilibrium σ_0 . Linear fits were approximated for gene expression data experimentally measured at varying σ . The upper and lower bounds of σ for applying the linear fit of the σ -gene expression data (σ_{lower} , σ_{upper}) were set in the range where a linear fit was appropriate for the data. FIG. 26G is a list of parameter values set for the DNA Supercoiling sub-model, according to an embodiment for *M. genitalium*.

4.6 DNA Replication sub-model

[0326] DNA replication is an integral part of the cell cycle which produces a complete chromosome for each daughter cell. DNA replication is initiated by the formation of a large multimeric DnaA complex at the origin of replication ($oriC$), which enables the recruiting of the DNA replication machinery to the $oriC$. Replication proceeds bidirectionally from the $oriC$ to the terminus ($terC$), half way around the circular chromosome. The chromosome has two strands of base pairing DNA called the leading and lagging strands. Replication of the leading strand proceeds continuously in the 5' to 3' direction. Replication of the lagging strand occurs in short Okazaki fragments, also in the 5' to 3' direction.

[0327] FIG. 27A through FIG. 27C are block diagrams that illustrates the progress of DNA replication and components of replication machinery in a cell, modeled according to an embodiment. The DNA replication machinery 2700 consists of multiple replisome proteins, and there is one set of machinery for each of the two replication bubbles (on each side of the *oriC*). The replicative DNA helicase 2704 serves to unwind the coiled DNA. The DNA primase 2705 makes short primers 2707 that help initiate the polymerization of a long stretch of DNA 2709a and 2709b (collectively referenced hereinafter as DNA 2709). Primers are made once at the origin for the replication of the leading strand, and one primer is made to start each Okazaki fragment. DNA polymerization is carried out by DNA polymerase III molecules, consisting of two core subunits 2701a and 2701b (collectively referenced as core subunits 2701), a gamma complex 2702, and beta-clamps 2703a and 2703b (collectively referenced as beta clamps 2703). One core 2701a or 2701b resides on each of the leading strand 2709a or lagging strand 2709b, respectively, of the DNA 2709, and is made up of two alpha subunits. The two cores 2701 are held together by a gamma complex 2702, consisting of delta, delta prime, gamma, and tau subunits. The core 2701 is also bound to a sliding beta-clamp 2703 which helps anchor the polymerase to the DNA and maintain processivity. On the lagging strand 2709b, beta-clamps 2703b are swapped out each time a new DNA loop 2710 is created to make a new Okazaki fragment. A “back-up” beta-clamp 2703c binds downstream of the lagging DNA loop 2710 to facilitate this switch and formation of the next loop. (The lagging strand polymerization requires the formation of DNA loops, so that the DNA can be polymerized in the opposite, 5’ to 3’, direction.) DNA ligases 2708 connect the separate polymerized Okazaki fragments together. Finally, single-stranded binding proteins (SSBs) 2706 stabilize and protect single-stranded DNA, which is created as the DNA is unwound to make room for the DNA machinery. For example, the lagging strand DNA loop 2710 often has long stretches of unwound DNA waiting to be polymerized. All of the replication proteins and all of the parameters used in replication of *M. genitalium* are described in FIG. 27G. FIG. 27G is a list of parameter values set for the DNA Replication sub-model, according to an embodiment.

[0328] Upon replication initiation (the binding of initiation molecules near the *oriC* by the Replication Initiation sub-model), the DNA Replication sub-model (also called the Replication sub-model for brevity) tracks the progression of the replication proteins on the known chromosome sequences specified in the Chromosome state variable.

[0329] To duplicate the chromosome, the exact position of each replisome protein is tracked over time on both the leading and lagging strands. The polymerization of Okazaki fragments is explicitly modeled. Since the exact primer binding sites for Okazaki fragments are uncharacterized for some organisms, including *M. genitalium*, the Okazaki fragment lengths are randomly determined based on a Poisson distribution centered around the mean fragment length (*IOF*). Polymerization of both the leading and lagging strands is limited by the average rate of polymerization (*kel*) in the organism (e.g., *Mycoplasmas*), the availability of nucleotides and energy, and the available protein binding sites on the chromosome. Further, polymerization of the leading strand is prevented if additional unwinding will lead to too many unprotected single-stranded bases on the lagging strand, or if the leading polymerase will progress a distance that is greater than two Okazaki fragment lengths beyond the lagging strand polymerase. Leading strands are not polymerized past the *terC*, and the lagging strands are not polymerized past the ends of Okazaki fragments.

[0330] The formation of a new Okazaki fragment requires the formation of a new lagging strand DNA loop. To prepare for this event, while the current Okazaki fragment is being polymerized, a “back-up” beta-clamp is bound just downstream of the start position of the next Okazaki fragment. The beta-clamp assembles as a dimer on the chromosome, and the binding event requires hydrolysis of one ATP molecule to ADP (e.g., *ebeta* = 1). When an Okazaki fragment has been completely polymerized, its beta-clamp is released, and the end of the Okazaki fragment is marked as having a single-strand break to be ligated by DNA ligase. The lagging strand primase and polymerase associate with the “back-up” beta clamp that was previously bound. This forms a new lagging strand DNA loop.

[0331] The Replication sub-model outputs the progress of replication, which has a big impact on many other processes in the system and corresponding sub-models. For example, the copy count of each gene plays a role in RNA polymerase binding in the Transcription sub-

model, and affects the cell's gene expression. The duplicated regions of the chromosome have to be wound appropriately by the DNA Supercoiling sub-model, and the completion of replication triggers other cell cycle events such as the decatenation of the chromosomes by the Chromosome Segregation sub-model.

[0332] Collisions of the replication machinery with the transcription machinery (as determined by the Chromosome state class) are also handled by the Replication sub-model. A helicase may collide with a RNA polymerase in two ways: head-on (RNA polymerase is traveling in a direction opposite to the helicase), or co-directionally (RNA polymerase is traveling in the same direction as the helicase). There is some debate in the literature regarding the pausing of the replication loop upon collisions and whether RNA polymerases are displaced upon collisions. Based on these studies as well as modeling considerations, it is assumed that the RNA polymerase is always displaced upon collisions, and the replication loop is only stalled upon a head-on collision. If the helicase hits an RNA polymerase, polymerization pauses, the RNA polymerase falls off, and its transcript is degraded. If it is a head-on collision, the impact will stall progression of the replication bubble for some amount of time (*tstall*). If it is a co-directional collision, polymerization will continue at full speed the following time step.

[0333] For initial conditions, at the beginning of the simulation, Okazaki fragment lengths were chosen randomly based on a Poisson distribution around the mean length *IOF* (e.g., about 1500 nucleotides). From this, the start position of each Okazaki fragment is obtained to be used to determine primer, beta-clamp, and polymerase core binding during the simulation.

[0334] The Replication sub-model involves keeping track of the specific positions of the replication proteins on the chromosome. The position on the chromosome where the proteins bind is determined based on the following rules.

1. The Helicase is centered on the boundary between double stranded DNA (wound) and single-stranded DNA (unwound). The position over which it is centered is the next position to be unwound.
2. There is no gap between the helicase and leading strand DNA polymerase core or between the leading strand polymerase core and the leading strand beta-clamp. Therefore,

- movement of the helicase controls the movement of the leading strand polymerase core.
3. There is no gap between the lagging strand DNA polymerase core and the beta-clamp on the current Okazaki fragment.
 4. The positions over which the polymerase cores are centered are the next positions to be polymerized.
 5. "Back-up" beta-clamps bind slightly upstream of the start site of the next Okazaki fragment to be polymerized such that when the next Okazaki fragment starts polymerizing, there will be no gap between the polymerase and beta-clamp, and the polymerase core will be centered on the Okazaki fragment start site.
 6. At replication initiation, the mother strands are separated such that the leading polymerase cores are centered at $\text{oriC} \pm 1$ base and the helicases are a distance l_{prim} (e.g., 11 nucleotides) ahead.
 7. During replication initiation (and the final step of replication after the last Okazaki fragment has completed), the lagging polymerase, beta-clamp, and primase are accounted for as part of a complex on the leading strand (containing also the helicase, leading polymerase, gamma complex, and leading beta-clamp). At all other times, the lagging polymerase, lagging beta-clamp, and primase are accounted for as a complex on a different strand. This allows for separate tracking of the leading and lagging polymerase positions.
 8. SSBs are bound at a fixed spacing (sss_b) in the single-stranded regions of DNA.

[0335] The Replication sub-model is built up of several sub-functions which move the replication proteins along the chromosomes and evolve the chromosome copy count of the cell to double (e.g., from 1 to 2 for *Mycoplasmas*). These sub-functions are evaluated in random order, so that shared resources, such as energy, are allocated fairly across sub-functions.

[0336] FIG. 27D through FIG. 27F are flowcharts that illustrate an example method 2720 for a DNA Replication sub-model, according to an embodiment. In step 2720 parameters for the model are set, such as a list of the protein monomers that serve as replisome enzymes, the SSB dissociation rate. Following step 2721, the other sub-functions (represented by steps 2722, 2730, 2740, 2750, 2760, 2770, 2780) are performed in random order.

[0337] Step 2722 is a sub-function for starting the replication. Step 2722 includes steps 2723, 2725 and 2727. In step 2723, it is determined if conditions are satisfied for replication to start, such as an initiation protein at the OriC sites, and sufficient enzymes and metabolites (including nucleic acid bases and ATP and water) to make two replisome complexes, each including 1 helicase, 1 primase, 2 polymerase cores, 1 beta clamp and 1 gamma complex. In step 2725, enough DNA base pairs are unwound to accommodate the DNA footprint of and thus allow binding of the two replisome complexes, and those locations are marked as unwound in the Chromosome state class. In step 2727 the copy counts of the metabolites are updated, e.g., the copy counts of ATP and water (H₂O, also H₂O) are decremented by the ATP energy cost for each unwound base times the number of unwound bases plus twice the ATP energy cost for binding each beta-clamp, and the copy counts of reaction products ADP and inorganic Phosphate (Pi, also PI) and hydrogen ion (H⁺) are incremented by a like number.

[0338] Step 2730 is a sub-function for advancing the replisome complexes along the DNA. Step 2730 includes steps 2731, 2733, 2735 and 2737. In step 2731, it is determined if a Okazaki fragment is starting, e.g., by the current base being a multiple of the fixed length determined during initialization. If not, then control passes to step 2735 to select the next strand. If so, then in step 2733 a primase and polymerase core are bound to the lagging strand and control passes to step 2735 to select the next strand. In step 2737, the unwinding and polymerizing of the DNA is modeled by advancing the positions of the leading and lagging polymerases and helicases as far as allowed while following conditions are met, where N indicates the number of polymerized/unwound bases on a given strand in the given time step, and P indicates the last polymerized position varying from 0 at OriC to $L/2$ at TerC, and L is the length of the chromosome. (For *M. genitalium*, $L/2 = 290,038$.) The conditions are: $N < \text{primer length}$ if primer; $N < \text{DNA polymerase elongation rate in time step}$; $\text{dNTP consumed} < \text{dNTP available}$; $N * \text{ATP cost of unwinding} < \text{available ATP}$; $\# \text{ bound SSB} = \text{length of lagging single strand region} / (\text{SSB 8mer footprint} * \text{SSB complex spacing})$; DNA accessible; $P < \text{terC on leading strand or Okazaki frag length on lagging strand}$; leading strand $P < \text{lagging strand } P + 2 * \text{mean Okazaki frag length}$; $P < \text{position of colliding RNA polymerase}$.

[0339] Step 2740 is a sub-function for binding and unbinding SSBs along the unwound DNA. Step 2740 includes steps 2741, 2743, 2745 and 2747 in which the positions are determined at fixed spacing (*sssb*), randomly bind a free SSB to each site randomly release a bound SSB based on the SSB dissociation rate, and dissociate free SSB 8-mers into two SSB tetramers, respectively.

[0340] Step 2750 is a sub-function for binding a back-up beta clamp. Step 2750 includes step 2751 through 2755. In step 2751 conditions that favor binding a free back-up beta clamp are evaluated. In the illustrated embodiment, the conditions include: copy count for available beta-clamp monomers > 2; copy count for ATP > ATP cost of beta-clamp binding; position on DNA is accessible; helicase has passed beta-clamp binding site; Okazaki fragment length > minimum for beta-clamp binding. In step 2753, it is determined if all conditions are satisfied. If not, the sub-function ends. If so, then in step 2755, the state of the beta-clamp protein complex is marked bound, the copy counts of metabolites are updated, e.g., counts are decremented for ATP and H₂O based on the ATP costs of binding, and the counts of metabolites produced ADP, PI and H⁺ are incremented by a like number.

[0341] Step 2760 is a sub-function for terminating Okazaki fragments. Step 2760 includes step 2761 through 2765. In step 2761 conditions that favor terminating are evaluated. In the illustrated embodiment, the conditions include: Okazaki fragment completely polymerized; # bound SSB = length of lagging single strand region / (SSB 8mer footprint *SSB complex spacing); back-up clamp bound on lagging strand; and leading strand advanced past next Okazaki start position. In step 2763, it is determined if all conditions are satisfied. If not, the sub-function ends. If so, then in step 2765, the state of the beta-clamp protein complex is marked unbound, the Okazaki fragment is marked with a break to be ligated, the lagging strand primase and polymerase are associated with the released clamp, and the copy counts of metabolites are updated.

[0342] Step 2770 is a sub-function for ligating terminated Okazaki fragments. Step 2770 includes step 2771 through 2775. In step 2771 the next single strand with a break, if any, is selected randomly. If none, the sub-function ends. If such a strand is found, then, in step 2773, conditions that favor ligation are evaluated. In the illustrated embodiment, the conditions include: DNA ligase available; Nicotinamide adenine dinucleotide (NAD) available, and a random number X from a Poisson distribution ≥ 1 , where the Poisson distribution has an occurrence rate = time step divided by the ligase kinetic rate. In step 2775, it is determined if all conditions are satisfied. If not, the sub-function ends. If so, then in step 2777, the strand break is repaired by ligation, and the copy counts of metabolites are updated, e.g., copy count of NAD is decremented and copy counts of adenosine monophosphate (AMP) produced during ATP synthesis, N-methyl mesoporphorin IX (NMM) and H^+ are incremented.

[0343] Step 2780 is a sub-function for terminating replication. Step 2780 includes step 2781 through 2785. In step 2781 conditions that favor terminating are evaluated. In the illustrated embodiment, the conditions include: leading strand polymerized to one side of *terC*; lagging strand polymerized to other side of *terC*. In step 2783, it is determined if all conditions are satisfied. If not, the sub-function ends. If so, then in step 2785, release all replisome complexes from chromosome so complexes are free and not bound, mark *terC* with a single strand break for subsequent repair in the Chromosome and Protein Complex state variables.

[0344] Some accommodations and simplifications apply to the sub-model for *M. genitalium*. For example, some bacterial species are known to have multiple replication initiation events during their life cycles. This has never been demonstrated in *M. genitalium*, so in example embodiments for *M. genitalium* the sub-model performs up to 1 chromosome duplication event per cell life cycle. Further, the exact mechanism of replication initiation in *M. genitalium* is unknown. *M. genitalium* does not include a DnaC homolog, which in other species is an essential cue for the binding of the replication machinery to the *oriC*. For the *M. genitalium* whole-cell model, the binding of 29 DnaA-ATP molecules to the *oriC* is the cue for replication initiation.

[0345] FIG. 27H is a list of cell state variables and other sub-models interacting with the DNA Replication sub-model, according to an embodiment. FIG. 27I through FIG. 27K are lists of instructions that illustrate an example DNA Replication sub-model, according to an embodiment. This example uses the parameter and values listed in FIG. 27G for the *M. genitalium* whole-cell model.

4.7 Replication Initiation sub-model

[0346] The Replication Initiation sub-model determines when during the cell cycle chromosome duplication begins. This replication initiation time is therefore very important in determining the cell's division time. The mechanism of replication initiation used here is modeled after that described for *E. coli* involving the protein DnaA (MG469). Chromosome Replication begins when a complex of 29 DnaA-ATP molecules assembles near the replication origin, OriC, at specific DNA motifs called R1-R5. The assembly of this complex is rare because the DnaA molecules are titrated out by approximately 2000 additional binding sites, or "DnaA boxes," that exist all around the chromosome. DnaA needs to be bound to ATP or ADP to bind to these sites, and binds on and off these sites throughout the cell cycle. Binding of the DnaA-ATP at the R1-R5 sites is cooperative, enabling the large complex to form at this specific location on the chromosome.

[0347] FIG. 28A is a list of parameters and values for an example DNA Replication Initiation sub-model, according to an embodiment. All of the parameters used in the Replication Initiation sub-model are described in FIG. 28A. All of the rate constants used in this sub-model were obtained from previous models of replication initiation. The site and state cooperativity constants were fit according to the cell cycle length. The site and state cooperativity constants directly affect the time required for replication initiation. The total cell cycle length has been experimentally measured, and the time required for Cytokinesis and Chromosome Replication can be estimated from their sub-models. By Equation 13.

$$timeReplicationInitiation = timeCellCycle - timeCytokinesis - timeReplication \quad (13)$$

The site and state cooperativity constants were fit such that on average the simulated cell divides in the experimentally measured amount of time.

[0348] First, DnaA rapidly associates with ATP, and in some cases DnaA-ATP can be converted to DnaA-ADP. DnaA-ATP and DnaA-ADP molecules bind to and release from the binding sites on the chromosome. The number of binding/unbinding events that occur in a given period of time is determined by the number of available DnaA-ATP and DnaA-ADP molecules and calculated binding/unbinding rates.

[0349] In addition to the R1-R5 sites at the origin, there are DnaA binding sites all around the chromosome. There are 2227 DnaA binding sites based on the *M. genitalium* motifs, which are 9 bases long. A high affinity site is defined as one that exactly matches the motif or its reverse complement (e.g., 148 sites for *M. genitalium*). A medium affinity motif is defined as one that matches 8 of the 9 bases in the motif or its reverse complement (e.g., 2079 sites for *M. genitalium*); a low affinity site is one that only matches 7 of 9 bases. The replication initiation sub-model allows binding to all of the high and medium affinity sites outside of the oriC region.

[0350] The sub-model recognizes 5 DnaA binding sites in the OriC region: one high affinity (R4), three medium affinity (R1-R3), and one low affinity (R5), to mimic *E. coli*'s 5 R-sites and the 5 R-sites predicted for *M. genitalium*. These sites reside between the genes MG469 and MG470 (bases: 578581-579224). There are 9 high and medium affinity motifs in this region, but the sub-model only recognizes 4 to match the *E. coli* pattern, and therefore ignore the sites at genome positions 578837, 578855, 578881, 578966, and 579139. R5 matches 7 of 9 bases of the motif, so it is a very weak binder of DnaA. Since its exact mechanism/purpose is unknown, the sub-model binds this site after a 28-mer DnaA-ATP initiator complex at R1-R4 is formed, and the R5 binding triggers initiation. This is the only low affinity binding site included in the sub-model.

[0351] The sub-model recognizes 7 states at each binding site. The 28-mer initiator complex consists of 7 DnaA-ATP molecules bound to each of the R1-R4 sites. This binding is split up into 7 serial states, and in each state one additional DnaA-ATP molecule binds to each of the R1-R4 sites. All four sites must be bound before transition to the next state.

[0352] DnaA-ATP binding at the origin is cooperative. There are two forms of cooperativity in the Replication Initiation sub-model. FIG. 28B and FIG. 28C are block

diagrams that illustrate binding state cooperativity for an example DNA Replication Initiation sub-model, according to an embodiment. The first type of cooperativity is state cooperativity, which helps increase the probability of transitioning into higher states. Once the binding within a state is completed, the cooperativity to transition into the next state is calculated as

$$\text{State Cooperativity Factor} = C_{\text{state}} (\text{State} - 1)$$

where C_{state} is a fixed cooperativity constant. This calculation is depicted in FIG. 28B. The second type of cooperativity is site cooperativity, which describes the effects of the binding of one or more of the R1-R4 sites, on the probability of binding the other sites. The site cooperativity factor of a given site is 1 unless certain other R-sites are already bound, in which case the cooperativity factor changes to a variable C_{site} . R4 is a high affinity site, so it generally is the first to bind and has the highest cooperative effect on the other sites. The site cooperativity rules can be seen in FIG. 28C.

[0353] R5 is also bound by cooperativity, given the presence of a complete 28-mer DnaA-ATP initiator complex at the R1-R4 sites. The resulting DnaA-ATP29-mer is the trigger to begin the Replication of the chromosome.

[0354] Following initiation, the DnaA-ATP complex at the origin dissociates and hydrolyzes. DnaA molecules can continue to bind around the chromosome. The formation of a second complete initiation complex is rare because the time left in the cell cycle is generally less than that needed for an initiation event and because DnaA molecules are further titrated with a second set of binding sites on the second chromosome. For simplicity, the sub-model does not include a second DNA replication event.

[0355] This sub-model is initialized to a steady state. The steady state amounts of free, medium affinity site bound, and high affinity site bound DnaA-ATPs and DnaA-ADPs are found using non-linear constrained optimization where we try to identify a state which is a stable point and which maximizes the amount of high affinity site bound DnaA-ATP. All DnaA proteins are either ADP or ATP bound at the initial time step. There are also no DnaA polymers at the R1-R4 sites at the start of the simulation.

[0356] FIG. 28D is a list of cell state variables and other sub-models interacting with the DNA Replication Initiation sub-model, according to an embodiment. State classes interacting with the DNA Replication Initiation sub-model are listed in FIG. 28D.

[0357] At each time step, the following algorithm is performed by the Replication Initiation sub-model, using Equations 14a through 14i.

1. DnaA activation. Deterministically form DnaA-ATP complexes up to the limit of available DnaA and ATP. The kinetics of DnaA activation are not known, and are not modeled in this embodiment.

$$\text{Number of activation events} = \min \{ \text{Number of available ATP molecules}; \\ \text{Number of free DnaA molecules} \} \quad (14a)$$

Update the counts of free DnaA, DnaA-ATP, and ATP.

[0358] 2. DnaA-ATP complex dissociation. A complex of multiple bound DnaA-ATPs is only found near the oriC. This complex can be released from the chromosome by the DNA polymerase at the start of DNA replication. The resulting DnaA-ADP molecules are able to re-bind to the chromosome or reactivate to DnaA-ATP. If 29-mer DnaA-ATP complex is disrupted by the replisome machinery, then the following steps are taken: Dissociate the complex to individual DnaA-ATP molecules; Hydrolyze the ATP to form free DnaA-ADP molecules; Decrement the counts of DnaA-ATP complexes and H₂O molecules; Increment the counts of free DnaA-ADP, H⁺, and Pi.

[0359] 3. Binding DnaA-ATP and DnaA-ADP Up to 7 DnaA-ATPs can polymerize at each of the origin sites and only one DnaA-ATP can bind at each site outside of the origin. Only one DnaA-ADP can bind at each site. If the chromosomes are sufficiently supercoiled, then do the following: Stochastically select both high and low affinity sites at the origin and around the chromosome to bind DnaA-ATP and DnaA-ADP molecules. Binding proceeds such that the rate of binding each site is as follows (where the cooperativity factor, depends on the polymerization status of the R1-R4 boxes):

$$\text{rate of DnaA-ATP binding R4 (high affinity)} = \text{kb1ATP} \times \text{numFreeDnaAATP} / \\ (\text{cell volume} \times \text{cooperativity factor}) \quad (14b)$$

$$\text{rate of DnaA-ATP binding R1-R3 (medium affinity)} = \frac{\text{kb2ATP} \times \text{numFreeDnaAATP}}{\text{cell volume} \times \text{cooperativity factor}} \quad (14c)$$

$$\text{rate of DnaA-ATP binding non-origin high affinity sites} = \frac{\text{kb1ATP} \times \text{numFreeDnaAATP}}{\text{cell volume}} \quad (14d)$$

$$\text{rate of DnaA-ATP binding non-origin medium affinity sites} = \frac{\text{kb2ATP} \times \text{numFreeDnaAATP}}{\text{cell volume}} \quad (14e)$$

The rate equations for DnaA-ADP binding are the same as above, except that the rate constants are kb1ADP and kb2ADP.

[0360] 4. Cooperativity. The binding of DnaA-ATP to the R1-R4 sites is cooperative at two levels. Site cooperativity increases the probability of binding successive sites within a state. State cooperativity increases the probability of entering successive states. The cooperativity factors for sites R1-R4 are calculated as follows: Calculate the State Cooperativity by Equation 14f:

$$\text{State Cooperativity Factor} = C_{\text{state}} (\text{State}-1). \quad (14f)$$

Calculate the Site Cooperativity. The Site Cooperativity Factor for a given site is 1 or Csite depending on the occupancy of the other R-sites. Assign the total Cooperativity Factor for sites R1, R2, and R3 as their respective Site Cooperativity Factors. Assign the total Cooperativity Factor for R4 as its Site Cooperativity Factor + State Cooperativity Factor. The cooperativity factors calculated here are used in the rate calculation in Step 3.

[0361] 5. Displacing DnaA-ATP and DnaA-ADP. The release of DnaA-ATP and DnaA-ADP molecules from their binding sites throughout the cell cycle is modeled at the following rates: Stochastically select DnaA-ATP and DnaA-ADP molecules to unbind from both high and low affinity sites at the origin and around the chromosome, such that the number of unbinding events is in agreement with the following rates:

$$\text{rate of DnaA-ATP displacement} = \text{kd1ATP} \quad (14g)$$

$$\text{rate of DnaA-ADP displacement} = \text{kd1ADP} \quad (14h)$$

[0362] 6. Reactivation of DnaA-ADP to DnaA-ATP. The rejuvenation of DnaA-ADP to DnaA-ATP is incorporated. This reaction is promoted by the acidic phospholipids cardiolipin

and phosphatidylglycerol. Deterministically reactivate free DnaA-ADP at a rate of:

$$\text{rate of reactivation} = \text{numFreeDnaAADP} * \text{k_Regen} * \text{membraneConcentration} / \text{K_Regen_P4+} \quad (14i)$$

where the membrane concentration is the grams of membrane in the cell divided by the volume of the cell. These values are read from the Mass and Geometry state classes.

[0363] 7. Replication-dependent bound DnaA-ATP release. Include a global term for the effect of active beta-clamps (a part of the DNA polymerase machinery) on bound DnaA-ATP inactivation. If the model predicts the exact position of the active beta-clamps on the chromosome, a global term is not needed; instead, model the release of a DnaA-ATP molecule exactly when the beta-clamp encounters it on the DNA. The free DnaA-ATP in the cytosol is not distinguished from the DnaA-ATP that has been recently released by a beta-clamp. Therefore, the release is modeled in the form of DnaA-ATP and not the hydrolysis to DnaA-ADP. This beta-clamp-dependent release is handled by the DNA Replication sub-model described above and Chromosome state class.

4.8 Transcriptional Regulation sub-model

[0364] Transcription factors are one of many mechanisms cells employ to respond to external signals and maintain homeostasis. Transcription factors regulate the synthesis rates of RNA by modulating the affinity of RNA polymerase for promoters. Transcriptional enhancers stabilize RNA polymerase-promoter complexes by contributing negative free energy to the complex, for example by providing additional surfaces for RNA polymerase binding. Transcriptional repressors destabilize RNA polymerase-promoter complexes, for example by sterically blocking promoters.

[0365] The Transcriptional Regulation sub-model models the binding of transcriptional regulators to promoters and the fold-change effect of transcriptional regulators on RNA polymerase-promoter binding for individual promoters. Because transcriptional regulation is not always well characterized, the sub-model makes several simplifying assumptions. First, it is assumed that transcriptional regulator-DNA binding is kinetically fast and energetically favorable, and therefore proceeds to completion within the simulation time step. Second,

because transcriptional regulator-promoter affinities have not been systematically characterized, it is assumed that transcriptional regulators bind promoters with affinity proportional to their fold change effect. Third, it is assumed that only one copy of each transcriptional regulator can bind at each promoter. Fourth, it is assumed that transcriptional regulators stably bind DNA. Consequently, transcriptional regulator dissociation is ignored except for displacement by other DNA-binding proteins, which is modeled by the Transcription sub-model and Replication sub-model. It is also assumed that transcriptional regulators independently affect RNA polymerase, and thus their fold change effects add multiplicatively.

[0366] Initial conditions are set as described above and summarized here. Briefly, after the Protein Monomer and Protein Complex states initialize the total copy number of each transcriptional regulator, the Transcriptional Regulation sub-model initializes the status—free or DNA-bound—and chromosomal location of the transcriptional regulators to a steady-state of the transcriptional regulatory network by iteratively evaluating sub-model results until convergence. Because the transcriptional regulatory model assumes that transcriptional regulator-promoter binding proceeds to completion within the simulation time step and is stable, the sub-model converges in one iteration.

[0367] FIG. 29A is a flowchart that illustrates an example method for a Transcriptional Regulation sub-model, according to an embodiment. In step 2901, parameters are set for the sub-model, including identifying the transcription regulation enzymes e for the organism, e.g., by the protein monomer species or protein complex species or copy identifier. For each transcription regulation enzyme e , its binding site Xep on promoter p , as well as its motif, affinity, and fold-change effect Fep is also retrieved.

[0368] In step 2903, the next promoter p expressed in chromosome k is selected; and, in step 2905, the count CNe of that enzyme in the cytosol compartment is retrieved and used with Fep to compute a transcription rate $Repk$ for transcription of the gene at promoter p on chromosome k by enzyme e . This step is repeated until there are no more promoters. Then control passes to step 2910.

[0369] In step 2910, enzymes are bound to the chromosome in proportion to the rates $Repk$. Step 2910 includes steps 2911, 2913 and 2915. In step 2911 a random value for enzyme e , promoter p and chromosome k is selected from a multinomial distribution, with relative probabilities based on the rates $Repk$. In step 2913 it is determined whether conditions allow binding of enzyme to promoter on that chromosome. If not, then step 2915 is skipped. If so, then the enzyme is bound to the promoter in step 2915 and the copy counts for free and bound enzymes in the cytosol are adjusted and the combination is removed from the distribution by setting $Repk = 0$. If it is determined in step 2917 that $Repk = 0$ for all e, p, k , then step 2910 is finished and control passes to step 2920. If not, then control passes back to step 2911 to draw another random combination of e, p, k .

[0370] In step 2920, the fold change effect fp of the bound enzymes on the affinity of RNA polymerase for the promoter p is determined. Step 2920 includes steps 2921, 2923 and 2925. In step 2921, all promoters are initialized with no change effect ($fp = 1$). In step 2923 the next promoter bound to an enzyme is selected. If none are left, then the method ends. Otherwise the fold change effect of all enzymes on the promoter is accumulated multiplicatively— fp is updated with the product of fp and Fep .

[0371] The Protein Monomer and Protein Complex states represent the free and DNA-bound copy numbers of each transcriptional regulator. The Chromosome state represents the exact chromosomal location of each DNA-bound transcriptional regulator. The RNA Polymerase state represents the fold change effect of transcriptional regulation on the affinity of RNA polymerase for each promoter. Several sub-models, including the Translation sub-model, model protein synthesis. The Transcription sub-model models: (1) transcription initiation involving RNA polymerase-promoter binding including the fold-change effects of transcriptional regulation; (2) transcript elongation; and (3) transcription termination.

[0372] FIG. 29B is a list of instructions that illustrates an example Transcriptional Regulation sub-model, according to an embodiment. These instructions re-use different variable names and indices, than used in FIG. 29A.

[0373] For application to *M. genitalium*, the applicable transcriptional regulatory network was reconstructed based on an extensive review of the primary literature and the proteomic database DBTBS. *M. genitalium* has few homologs to reported transcription factors. The reconstructed network contains five regulators which regulate 54 genes through 29 regulatory interactions, including one regulator (Spx, MG127) which interacts directly with RNA polymerase. The five reconstructed transcriptional regulators and the binding site, motif, affinity, and fold-change effect of each regulatory interaction were curated.

4.9 Transcription sub-model

[0374] Transcription is the first step in the synthesis of functional gene products where RNA polymerase and several accessory enzymes translate transcription units (regions of the DNA containing 1 or more genes) into RNA molecules. An RNA polymerase can bind on and off of the DNA either specifically to a gene promoter or non-specifically to a non-promoter site. Transcription begins with the recruitment of RNA polymerase to a gene promoter with the help of a sigma initiation factor and possibly transcription factors. Next elongation factors are recruited, RNA begins to be polymerized, and the sigma factor is released. In this stage, the RNA polymerase is said to be in the actively transcribing state. Finally the RNA polymerase reaches a terminator at the end of the transcription unit, and with the help of termination factors releases the polymerized RNA and dissociates from the DNA.

[0375] As soon as the RNA begins to polymerize, even prior to termination, the mRNA transcripts may be bound by ribosomes and translated to polypeptides. For simplicity, in some embodiments, the transcription sub-model does not represent this phenomenon, allowing translation only of completed mRNAs.

[0376] The parameters required for transcription were derived from multiples sources including microarray gene expression data and RNA half-life data. The data used and

calculations performed pre-fitting are described here. For some applications, tRNA expression estimates analogous to the mRNA microarray data are useful. Since no microarray data is available for many organisms, cell composition data was used to approximate tRNA expression. The total tRNA expression is approximated as the total mRNA expression multiplied by the ratio of the tRNA to mRNA weight fractions of the cell. To approximate the expression of each individual tRNA, measurements of the relative abundances of each amino acid in the cell were used. A tRNA expression is the total tRNA expression multiplied by the fraction of the total amino acid weight represented by its amino acid. For example, alanine accounts for about 8% of the total amino acid mass. Then the expression of tRNA for alanine, MG471, is 8% of the total tRNA expression. In cases of degeneracy, where multiple tRNAs bind to the same amino acid, the amino acid weight fractions is split evenly between the tRNAs.

[0377] Similar to the tRNA expression calculations, the total rRNA expression is calculated as the total mRNA expression multiplied by the ratio of the rRNA to mRNA weight fractions of the cell. This total rRNA expression is split between the rRNA species based on their relative abundances in the cell. Also sRNA expression is calculated in the same way, such that expression is proportional to RNA abundance in the cell.

[0378] The rate of change of the concentration of an RNA species i , designated RNA_i , is its synthesis net its degradation, as given by Equations 15a.

$$dRNA_i/dt = k_{synth,i} - k_{deg,i} |RNA_i| \quad (15a)$$

where k_{synth} is the rate of synthesis and k_{deg} is the rate of degradation. The degradation rate is described by the first-order degradation constant of RNA_i with half-life h_i given by Equation 15b.

$$k_{deg,i} = \ln(2) / h_i \quad (15b)$$

At steady state, the time derivative of concentration is zero, so the steady state concentration $|RNA_i|_{ss}$ can be substituted to produce

$$k_{synth,i} = \ln(2) |RNA_i|_{ss} / h_i \quad (15c)$$

Substituting the relative expression of the RNA species i , designated E_i , multiplied by the total RNA concentration $|RNA|$ gives

$$k_{synth,i} = \ln(2) |RNA|_{Ei} / h_i \quad (15d)$$

Assuming exponential growth with time t and an initial rate of synthesis designated $k_{synth,i,0}$ and initial concentration $|RNAi|_0$ and a cell cycle time of τ ,

$$k_{synth,i} = k_{synth,i,0} \exp(\ln(2) t / \tau) \quad (15e)$$

and, thus

$$|RNAi| = |RNAi|_0 \exp(\ln(2) t / \tau) \quad (15f)$$

$$d|RNAi|/dt = k_{synth,i,0} \exp(\ln(2) t / \tau) - k_{deg,i} |RNAi|_0 \exp(\ln(2) t / \tau) \quad (15g)$$

Assuming that mother and daughter cells are identically distributed:

$$|RNAi|_{\tau} = 2|RNAi|_0 \quad (15h)$$

This means integrating equation 15g from $t=0$ to τ , should generate an increase in concentration equal to the initial concentration. Performing the integration, rearranging the terms and substituting $|RNA|_{Ei}$ for $|RNAi|$ gives

$$k_{synth,i,0} = |RNA|_0 Ei \{k_{deg,i} + \tau/\ln(2)\} \quad (15i)$$

[0379] Genes that are transcribed together in a transcription unit have the same synthesis rate in the transcription sub-model. This is because the sub-model does not include incomplete transcription of a transcription unit, where the RNA polymerase can fall off the unit, after transcribing a subset of its genes. For half-lives that have not been measured the half-lives were adjusted to even out the synthesis rates within a transcription unit. For the other cases, the synthesis rate of the transcription unit was set to be the average of the rates within the transcription unit.

[0380] The probability of an RNA polymerase binding a given transcription unit, i , is based on the synthesis rate of transcription unit i relative to the synthesis rate of all the other transcription units:

$$\text{Probability(transcribing } i) = P_{tu,i} = k_{synth,i} / \sum_j k_{synth,j} \quad (15j)$$

Thus the probability of binding can be derived from the half-lives and the relative concentrations Ei of RNA.

[0381] A Markov model is used to determine the state of each RNA polymerase. An RNA polymerase may exist in any of the four following states: 1. Free (FS, not bound to a

chromosome); 2. Non-specifically bound somewhere on a chromosome (NSB); 3.

Specifically bound to the promoter of a transcription unit (SB); 4. Actively transcribing a transcription unit (AT). Transition probabilities between these states are designed to maintain the occupancy of each state within a narrow window around their expected values. The transition probabilities (P_{trans}) are determined by four logistic control functions, tuned by the RNA polymerase state expectations.

[0382] All newly created or released RNA polymerases are in the free state. From the free state, a polymerase can transition to the non-specifically bound state, transition to the specifically bound state, or remain in the free state. From the non-specifically bound state, a polymerase can transition to the free state, transition to the specifically bound state, or remain in the non-specifically bound state. A random position on a chromosome for the non-specifically bound polymerase is selected from the polymerase accessible sites as determined by the Chromosome state. From the specifically bound state, a polymerase can transition to the free state, non-specifically bound state, or actively transcribing state, or remain in the specifically bound state. A polymerase can only transition into the specifically bound state if a free sigma factor is available. Upon transition a polymerase into this state, a transcription unit to bind to is picked randomly. The probability of binding each transcription unit is based on experimentally measured gene expression and half-life data, as derived above. Once in the actively transcribing state, a polymerase will remain in the actively transcribing state until transcription is terminated, the RNA polymerase is displaced by another protein, or the RNA polymerase falls off due to a stall. In all these cases, the RNA polymerase falls off into the free state. For a polymerase in the actively transcribing state, any bound sigma factors are released and the transcript is elongated according to nucleic acid limits (given the transcript sequence) and the elongation rate *kel*. The transcript and polymerase are released if transcription is complete and the necessary termination factors are available.

[0383] All RNAs are initialized to the mature state. RNA polymerases are initialized as follows: Each RNA polymerase is randomly assigned (with replacement) to one of the actively transcribing, specifically bound, non-specifically bound, or free states weighted by the expected occupancy of each state. Actively transcribing and specifically bound

polymerases are randomly assigned to transcription units weighted by the transcription unit binding probabilities (P_{tu}). Each transcription unit to which one or more actively transcribing polymerases have been assigned is divided into 1 segment for each polymerase. Actively transcribing polymerases are randomly assigned to positions within the assigned segment of their assigned transcription unit (positions near the segment border are not allowed to prevent polymerases from being too close to each other) with uniform probability. Non-specifically bound polymerases are randomly assigned to an accessible region on the chromosome.

[0384] FIG. 30A and FIG. 30B are flowcharts that illustrate an example method 3000 for a Transcription sub-model, according to an embodiment. In step 3000, parameters are set for the Transcription sub-model, including the allowed states and probabilities of transitioning from one state to the next. Markov chain parameters are set to preserve observed distributions of RNA species and states. Protein monomers and complexes that serve as transcription enzymes are identified, including at least one each of sigma factors, elongation factors, termination factors and anti-termination factors. The RNA polymerase elongation rate kel is also set for the organism.

[0385] In step 3003 and 3005 the FS state is increment upon the occurrence of a new RNA polymerase. In step 3011 a Markov model weighted by the transition probabilities is used to change counts in each state. In step 3013 respond to increase in NSB by updating chromosome state for a random binding site.

[0386] In step 3015, a specific transcription unit and promoter p are determined based on transcription unit binding probabilities P_{tu} and binding probability fold changes fp for each promoter. If promoter o is accessible and there is a sigma factor determined in step 3021, then in step 3023 update the chromosome stat to bind RNA polymerase at the promoter and in step 3025, decrement copy count for free sigma factor.

[0387] The next AT polymerase with elongation factor is selected in step 3031, and a sigma factor is released in step 3035 if elongation is in its first time step. If the next position on DNA is accessible and the needed RNA base is available and the elongation rate is not exceeded, as determined in step 3041, then in step 3043 an RNA base is added to the transcript. If there is another non-limited AT polymerase, as determined in step 3045, then

control returns to step 3031 to get the next AT polymerase. If not, then it is determined in step 3051 whether transcription by an RNA polymerase is completed and termination factors are available. If not, the sub-model ends for the current time step. If so, then in step 3053, the RNA transcript is released and its count incremented; and the RNA is released to the free state and the counts of AT and FS states are adjusted.

[0388] FIG. 30C is a list of cell state variables and other sub-models interacting with the Transcription sub-model, according to an embodiment.

[0389] For an *M. genitalium* specific model, it is noted that termination of transcription in some bacteria can require the hexameric ATP-dependent helicase Rho. However, Rho is not essential in *B. subtilis* (gram positive) or *M. genitalium*, and therefore only Rho-independent termination was included in the *M. genitalium* specific transcription sub-model. Rho-independent termination occurs via the intrinsic properties of RNA which disrupt RNA polymerase-DNA binding.

[0390] Gene expression data for *M. genitalium* was unavailable, so data (measured at 37°C) was used from *Mycoplasma pneumoniae*, the closest phylogenetic relative of *M. genitalium*. The microarray data was presented in normalized log form. Since the exact method of normalization was unknown to us, to re-derive expression levels the value $2^{(\text{presented value})}$ was calculated. The *M. genitalium* genome is contained in the larger *M. pneumoniae* genome and we were able to map a *M. pneumoniae* gene to all but six *M. genitalium* genes. The genes were mapped based on matching gene names, matching gene descriptions, and Bio-cyc's Align Multi-Genome Browser.

[0391] The half-lives of mRNA for *M. genitalium* are obtained from measurements in *E. coli* performed at 30 °C in M9 minimal media. Additional sets of *E. coli* half-life data are available, but the set was chosen with the most comprehensive list of genes mapping to homologous *M. genitalium* genes, and whose average half-life was closest to the reported bulk *E. coli* mRNA half-life. The gene mapping between *E. coli* and *M. genitalium* was based on common gene names and annotations, Bio-Cyc's Align Multi-Genome Browser, and UniProt. A half-life estimate for 274 *M. genitalium* genes was thus obtained. For the remaining genes, the average half-life, 4.425 minutes, was assigned. Separate sources were

used to acquire the half-lives of tRNAs (45 minutes), rRNAs (150 minutes), and sRNAs (89 minutes).

[0392] The transcription unit structure (sets of genes that are transcribed together following a single promoter binding event) was compiled from several sources including primary reports and databases of co-transcribed genes, and a computational model that predicts promoter and transcription unit start sites. Transcription units were also predicted using information on the conservation of gene order across multiple species, the related functions of adjacent genes, the similar expression levels of adjacent genes as measured by microarrays, and the strandedness (transcription direction) of adjacent genes. The 525 *M. genitalium* genes were transcribed in 355 transcription units. 294 genes fall in transcription units of more than one gene. The longest transcription unit contains 4 genes.

[0393] FIG. 30D is a list of parameters and values for a Transcription sub-model, according to an embodiment. At each time step, the sub-model for *M. genitalium* performs the following steps.

1. Assign all newly synthesized RNA polymerases to the free state
2. Use the Markov model to randomly transition RNA polymerases among the FS, NSB, SB, and AT states, weighted by the state transition probabilities *Ptrans*.
3. Randomly assign RNA polymerases entering the NSB state to an accessible position on the chromosome (performed by Chromosome state).
4. Randomly assign RNA polymerases entering the SB state to specific transcription units weighted by the product of the transcription unit binding probabilities (*Ptu*) and the binding probability fold changes. (Fold changes arise from other cellular processes and are described in the Transcriptional Regulation and DNA Supercoiling sub-models.) We determine whether the selected promoter site is accessible and bind the polymerase to a chromosome using the Chromosome state class. RNA polymerase specific binding can only occur if there is an available sigma factor, and a free sigma factor is accordingly decremented.
5. Simulate RNA polymerization by actively transcribing RNA polymerases with the aid of elongation factors.

- (a) Release the sigma factor if this is the first second of elongation, and increment the free sigma factor count.
 - (b) If all of the necessary elongation factors are available, elongate the transcript according to nucleic acid limits (given the transcript sequence) and the elongation rate *kel*. We allocate available nucleic acids among the actively polymerizing RNA polymerases by adding a base to each elongating transcript before moving to the next base of a given transcript. Polymerization can also be limited if the new site at which the polymerase will land is not accessible (for example, if another RNA polymerase lies in its way).
6. If transcription is complete and the necessary termination factors are available,
- (a) Release the completed transcript;
 - (b) Transition the RNA polymerase to the free state;
 - (c) Increment the RNA count.

4.10 RNA Processing sub-model

[0394] Cells transcribe genes both individually as well as in groups referred to as transcription units or operons. Operonic transcription has several advantages compared to single-gene transcription. First, operonic transcription allows cells to minimize the number of transcriptional regulators required to control gene expression. Second, operonic transcription increases the likelihood that groups of gene products have similar stoichiometry. At the same time, operonic transcription carries the costs of synthesizing intercistronic RNA and cleaving operonic non-coding transcripts into individual RNAs. Additionally, operonic transcription doesn't provide separate control of the expression of each gene. The RNA Processing sub-model models operonic RNA cleavage into individual RNA gene products.

[0395] The RNA processing sub-model models the cleavage of operonic non-coding RNA into individual gene products and intercistronic RNA fragments. Because the kinetics of RNA cleavage are not well characterized, this process makes several simplifying assumptions. First, this process assumes that each RNA is fully cleaved in a single time step. Consequently, this process collapses the cleavage of each RNA into a single reaction, and only represents uncleaved and fully cleaved RNA. Intermediate cleavage configurations are

not represented. Second, this process assumes that the mean cleavage rate, v_i , of each RNA species i is independently limited by (1) the copy number of unprocessed RNA, r , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of RNA processing enzymes, e_j . Based on these assumptions, the functional form of v_i is given by Equation 16.

$$v_i = \min \left(\overbrace{r_i^+}^{\text{RNA}}, \underbrace{\left[\min_j \frac{m_j}{M_{ji}^+} \right]}_{\text{metabolites}}, \underbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}_{\text{enzymes}} \right), \quad (16) \text{where}$$

M_{ji} is the stoichiometry of metabolite j in the processing of RNA species i including NTP hydrolysis coupled to phosphorolytic cleavage, $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the processing of RNA species i , and Δt is the simulation time step.

[0396] The RNA Processing sub-model implements a stochastic model of the arrival of RNA processing events with relative rates v_i . Until RNA, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected processing reaction, updating the copy numbers of RNA and metabolites and decrementing the available enzymatic capacity.

[0397] After the Mass state initializes the total cell mass, the RNA state initializes the total copy count of each RNA species and initializes all RNA to their mature—processed and modified—configuration. Second, the tRNA Aminoacylation sub-model initializes tRNA to the aminoacylated configuration. Next, the Macromolecular Complexation and Ribosome Assembly sub-models initialize the copy count of each ribonucleoprotein complex. The Translation sub-model initializes the mRNA codon location of each ribosome.

[0398] The expression of the RNA processing enzymes was fit to provide sufficient enzymes to quickly process RNA, or more specifically to prevent sustained accumulation of unprocessed RNA.

[0399] FIG. 31A is a flowchart that illustrates an example method 3100 for a RNA Processing sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is N_m , the count of enzyme index e is N_e , and the reaction index is r for RNA processing reactions. In step 3101, the protein monomers and complexes that serve as enzymes for RNA processing reactions are identified as is the stoichiometry M_{mr} of metabolites in the reactions, and catalytic rate K_{er} of enzymes in the reaction. In step 3103 the enzyme capacity C_e during a single time step is set to the count of that enzyme.

[0400] In step 3111 the next unprocessed RNA species r , if any, is determined; and in step 3113 Equation 16 is evaluated to determine V_r (corresponding to v_i). This is repeated until all metabolites or enzymes or unprocessed RNA are consumed, at which point $V_r = 0$ for all r , and the method ends. When a V_r has been calculated for every unprocessed RNA species, control passes to step 3121 to randomly select a reaction to implement using a multinomial distribution with probabilities based on the arrival rates V_r .

[0401] In step 3123, the RNA processing reaction is run to process one copy of unprocessed RNA species r . The copy count of the RNA is updated (unprocessed RNA count is decremented and processed RNA count incremented). The metabolite counts for metabolites m are decremented by M_{mr} and the enzyme capacity C_e is also decremented. Control passes back to step 3111 to recompute V_r for all r based on the new counts.

[0402] The RNA state variable represents the copy numbers of unprocessed, processed, and intercistronic RNA. The Metabolite state variable represents the copy number of each intracellular metabolite. The Protein Monomer and Protein Complex state variables represent the copy number of each RNA processing enzyme. RNA are synthesized and matured in four steps. This process models the cleavage of RNA transcripts produced by the Transcription sub-model. The RNA Modification sub-model models the modification of transcripts cleaved by this process. The Macromolecular Complexation and Ribosome Assembly sub-models model the formation of macromolecular complexes, including the ribosomal sub-units. The Translation sub-model models the function of m-, r-, s-, and tRNA in translation. The RNA Decay sub-model models the degradation of cleaved intercistronic RNA fragments.

[0403] For the *M. genitalium* example, the transcription unit organization of the *M. genitalium* chromosome was reconstructed by mapping the “suboperon” structure of *M. pneumoniae* chromosome on to that of *M. genitalium* with two modifications. First, non-coding RNA were organized into transcription units according to their “reference operons.” Second, all mRNAs either not associated with sub-operons, without *M. pneumoniae* homologs, or which have been rearranged since divergence from *M. pneumoniae*, were assigned to their own transcription units. The *in silico* *M. genitalium* chromosome is organized into 335 transcription units containing 525 genes.

[0404] Bacteria also transcribe 3’ and 5’ leader sequences before and after each non-coding RNA gene. These leader sequences must be cleaved to produce functional non-coding RNA. *M. genitalium* mRNA cleavage is not well described and is not modeled. The model assumes *M. genitalium* polycistronic mRNA are not cleaved. *E. coli* rRNA have been shown to be transcribed as a single 30S transcript which is cleaved into individual 5S, 16S, and 23S rRNA transcripts by the action of several ribonucleases. The *E. coli* rRNA cleavage scheme was adapted and simplified for the reduced ribonuclease complement of *M. genitalium* by removing cleavage reactions catalyzed by non-homologous enzymes. First, ribonuclease III (MG367) hydrolytically cleaves the 30S rRNA into 5S, 16S, and 23S rRNA precursors. Second, hydrolytic ribonuclease J (MG139) and phosphorolytic ribonucleases RsgA (MG110) and DeaD (MG425) cleave the 3’ and 5’ ends of the 5S and 16S rRNA precursors.

M. genitalium doesn't contain a homolog of the *E. coli* 9S RNA cleavage enzyme ribonuclease E. The mechanisms of 3' and 5' cleavage of the *M. genitalium* 5S rRNA precursor are unknown. 3' and 5' cleavage of the 5S rRNA precursor are modeled as a spontaneous process. The reconstructed *M. genitalium* small non-coding RNA (sRNA) precursor cleavage was based on that of *E. coli*. ffs (MG0001) is cleaved at both its 3' and 5' ends by ribonuclease III. Ribonuclease P cleaves the 5' end of rpnB (MG0003) and ssrA (MG0004). The reconstructed *M. genitalium* tRNA precursor cleavage scheme was based on that of *E. coli*. Ribonucleases III (MG367) and P (MG0003, MG465) hydrolytically cleave the 3' and 5' ends of each pre-tRNA, removing the 3' and 5' leader regions and intercistronic regions to produce individual tRNA. The stoichiometry, kinetics, and energetics of each non-coding RNA cleavage reaction were reconstructed based on extensive review of the primary literature.

[0405] FIG. 31B is a list of instructions that illustrates an example RNA Processing sub-model, according to an embodiment for *M. genitalium*.

4.11 RNA Modification sub-model

[0406] Post-transcriptional base modification occurs in order to degenerately encode amino acid triplet codes using far fewer tRNA species than possible using only Watson-Crick base pairing. Modification of wobble position is believed to be most important for improving codon recognition. Modifications distant from the anticodon are believed to help tRNAs properly fold and stabilize their catalytically active structures. Many cells also post-transcriptionally modify rRNA. rRNA modifications are believed to help stabilize rRNA, participate in protein synthesis, and confer resistance to ribosomal inhibitors. The exact role of rRNA modification is unknown. The RNA Modification sub-model models tRNA and rRNA modifications.

[0407] The RNA Modification sub-model models non-coding RNA modification. Because the kinetics of RNA modification are not well characterized, this sub-model makes several simplifying assumptions. First, this sub-model assumes that each RNA is fully modified in a single time step. Consequently, this process collapses the modification of each RNA into a

single reaction, and only represents unmodified and fully modified RNA. Intermediate modification configurations are not represented. Second, this sub-model assumes that the mean arrival rate, v_i of modification events of each RNA species i is independently limited by (1) the copy count of unmodified RNA, r , (2) the copy counts of intracellular metabolites, m_j , and (3) the copy counts of RNA modification enzymes, e_j . Based on these assumptions, the functional form of v_i is given by Equation 17, which parallels Equation 16.

$$v_i = \min \left(\overbrace{r_i^u}^{\text{RNA}}, \underbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}_{\text{metabolites}}, \underbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}_{\text{enzyme}} \right) \quad (17)$$

where M_{ji} is the stoichiometry of metabolite j in the modification of RNA species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the modification of RNA species i , and Δt is the simulation time step.

[0408] This sub-model implements a stochastic model of the arrival of RNA modification events with relative rates v_i . Until RNA, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each modification event, (2) selects a single modification to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected modification reaction, updating the copy numbers of RNA and metabolites and decrementing the available enzymatic capacity.

[0409] FIG. 32A is a flowchart that illustrates an example method 3200 for a RNA Modification sub-model, according to an embodiment. This flow chart uses the notation of FIG. 31A that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is r for RNA processing reactions, and follows the same flow, with the only change being the steps in FIG. 32A applies to RNA modification instead of RNA processing. For brevity these steps will not be narrated here.

[0410] The RNA state represents the unmodified and modified copy numbers of each RNA species. The Metabolite state represents the copy number of each intracellular metabolite. The Protein Monomer and Protein Complex states represent the copy number of each RNA modification enzyme. RNA are synthesized and matured in four steps. This sub-model models the modification of individual non-coding RNA following RNA. The Macromolecular Complexation and Ribosome Assembly p sub-models model the formation of macromolecular complexes, including the 30S and 50S ribosomal particles. The Translation sub-model models the function of m-, r-, s-, and tRNA in translation. The RNA initial conditions were described above with respect to the RNA Processing sub-model.

[0411] For the *M. genitalium* example whole-cell model, tRNA modification complement was reconstructed based on the observed complement of *E. coli* modifications. First, the *E. coli* modification complement was reconstructed. Second, modifications situated in conserved motifs were transferred to *M. genitalium*. The *M. genitalium* rRNA modification complement was similarly reconstructed. The stoichiometry and kinetics of each RNA modification reaction were reconstructed based on an extensive review of the literature.

[0412] FIG. 32B is a list of instructions that illustrates an example RNA Modification sub-model, according to an embodiment.

4.12 RNA Aminoacylation sub-model

[0413] The tRNA Aminoacylation sub-model simulates the conjugation of amino acids to the tRNAs. tRNAs serve as mediators between the ribosome and the amino acids which form polypeptides. tRNAs are composed of short RNA sequences which recognize specific codons (triplets of bases) on mRNAs. Each tRNA binds to a specific amino acid, and then interacts with a ribosome to deliver this amino acid to an elongating polypeptide chain, according to the mRNA code. tmRNAs are short RNA structures that add a proteolysis tag to the end of incomplete polypeptides upon ribosomal stalling, signaling the polypeptides for degradation. The tRNA Aminoacylation sub-model also simulates the aminoacylation of the tmRNA which delivers the amino acid alanine to stalled ribosomes. These aminoacylation reactions are both enzyme and energy dependent.

[0414] The tRNA Aminoacylation sub-model maximizes the number of aminoacylation reactions (tRNA aminoacylations, tmRNA aminoacylations, and tRNA modifications) up to the limits of available RNAs, enzymes, and metabolites. The enzymatic bounds are calculated from the catalytic turnover constant (k_{cat}) of each enzyme. The required metabolites include, among others, amino acids and ATP. Since multiple reactions require the same metabolites and enzymes, reactions to occur within a given time step are randomly selected using a probability distribution that is weighted by the abundances of the reaction requirements. That is, the limits of each reaction are calculated assuming that all of the available required resources would be allocated to the given reaction. The probability distribution for selecting given reactions is weighted by these calculated limits. Reactions are performed one by one until insufficient resources exist to perform any additional reactions. Intermediate steps in the aminoacylation of tRNAs are not represented.

[0415] The tRNA Aminoacylation sub-model reads from and writes to the RNA state class. It reads in the counts of free and aminoacylated tRNAs and tmRNA, and writes back the updated counts. All of the tRNAs are initialized to an aminoacylated state.

[0416] FIG. 33A is a flowchart that illustrates an example method 3300 for an RNA Aminoacylation sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is N_m , the count of enzyme index e is N_e , and the reaction index is r for tRNA aminoacylation reactions. In step 3301, the protein monomers and complexes that serve as enzymes e for tRNA aminoacylation reactions are identified as is the stoichiometry M_{mr} of metabolites in the reactions, and catalytic rate K_{er} of enzymes in the reaction. In step 3303 the enzyme capacity C_e during a single time step is set to the count of that enzyme, N_e .

[0417] In step 3311 the next free tRNA or tmRNA species r , if any, is determined; and in step 3313 a reaction limit RLr is determined. This is repeated until all metabolite or enzyme or unprocessed RNA are consumed, at which point $RLr = 0$ for all r , and the method ends. When a RLr has been calculated for every free tRNA and tmRNA species, control passes to step 3321 to randomly select a reaction r to implement using a multinomial distribution with probabilities based on the reaction limits RLr .

[0418] In step 3323, the RNA aminoacylation reaction r is run to process one copy of the tRNA or tmRNA species r . The copy count of the RNA is updated (free tRNA count or free tmRNA count is decremented and aminoacylated RNA count incremented). The metabolite counts for metabolites m are decremented by Mmr and the enzyme capacity Ce is also decremented. Control passes back to step 3311 to recompute RLr for all r based on the new counts.

[0419] For the *M. genitalium* example whole-cell model, *M. genitalium* is believed to have 36 tRNA aminoacylation reactions and 1 tmRNA aminoacylation reaction, and all of the reactions require 1 ATP. There are also two tRNA modification reactions that must occur. Since there is no glutamyl tRNA synthetase (to add a glutamine to a tRNA), glutamyl-tRNA synthetase first adds a glutamate to tRNA MG502, and then Glu-tRNA Gln amidotransferase converts the glutamate into a glutamine. Also, a methionylformyl transferase is used to add the formyl group onto the methionine on tRNA MG488 (formylmethionine is used as the start codon).

[0420] FIG. 33B is a list of parameters for an RNA Aminoacylation sub-model, according to an embodiment. The following list of instructions indicates a method for the example embodiment at each time step. For each of the 39 possible reactions (36 tRNA aminoacylations, 1 tmRNA aminoacylation, 1 Glu-Gln amidotransfer, 1 methionylformyltransfer), the sub-model calculates the maximum number of reactions that can occur using Equation 18.

$$\text{Reaction Limit } i = \min \begin{cases} \text{Number of available required metabolites (amino acid, ATP, ...)} \\ \text{Number of available required enzyme} \times k_{\text{cat}} \\ \text{Number of free tRNA or tmRNA} \end{cases} \quad (18)$$

The available ATP, amino acids, and enzyme activities used in the limits calculation above may be double counted as multiple reactions may require the same metabolites or enzymes. Therefore, the selection of which reactions occur, within the calculated limits, is randomly determined. While resources are available:

- (a) Randomly select a reaction to perform, weighing the probability of selecting a given reaction by its calculated reaction limit
- (b) Decrement the used metabolites, enzyme activities, free tRNAs, free tmRNAs
- (c) Increment the produced metabolites, and modified RNA
- (d) Recalculate the reaction limits as in Step 1, based on the new availabilities

4.13 RNA Decay sub-model

[0421] In presence of ribonucleases such as ribonuclease R, RNAs have relatively short half-lives compared to that of other macromolecules (e.g., protein, DNA) and the mass doubling time. The relatively short half-lives of RNAs enables the small cells with very small pool of RNAs and particularly few mRNAs to sample a broader range of configurations of the RNA pool over a shorter period that would be possible with longer half-lives. This helps the cell more finely tune the expression of proteins, more efficiently execute cell-cycle dependent events, and respond to the external environment. However, this enhanced fitness due to short RNA half-lives comes at a large energetic cost. In addition to ribonucleases, aminoacylated RNAs require peptidyl tRNA hydrolase to release their conjugated amino acids. This sub-model decays all species of RNA, and at all maturation states including aminoacylated states.

[0422] Given the availability of the necessary decay enzymes, RNAs are randomly selected for degradation by a Poisson probability distribution based on the RNA half-life. This sub-model contains no intermediate representation of RNA degradation. RNA degradation is

treated as an all-or-nothing event that proceeds to completion in a single time step. Upon degradation, water is used to break the nucleotide-nucleotide bonds and the nucleotides are recycled. All aborted transcripts (due to stalled RNA polymerases or RNA polymerases that have been knocked off of the DNA) are also degraded by this process. No initialization steps are required for this sub-model.

[0423] FIG. 34A is a flowchart that illustrates an example method 3400 for an RNA Decay sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is r for decay reactions of RNA species r . In step 3401, the protein monomers and complexes that serve as enzymes e for RNA decay reactions are identified as is the stoichiometry Mmr of metabolites in the reactions, and catalytic rate Ker of enzyme e in the reaction r .

[0424] In step 3411, the limits DR and DT on decay of RNA and tRNA, respectively, during one time step are determined, limited independently by copy count Ne of enzymes. For example, if enzyme e_1 works on RNA in reaction r_1 and enzyme e_2 works on tRNA in reaction r_2 , then, in some embodiments, $DR = 0$ if $Ne_1 = 0$ and $DT = Ne_2 * Ke_2r_2$. In step 3421, values of Nr , the copy count for RNA species r , $kdecay r$ and reactants and products of reaction r from one or more cell state variables are retrieved.

[0425] In steps 3431 and 3433, the next RNA species is selected that has associated decay enzymes, if any. If none, the method ends. Otherwise step 3435 is executed to determine if a copy of that RNA species is aborted. If so, the aborted RNA copy is slated for decay in step 3441, described below. Otherwise, the next RNA species is selected at random in step 3437. In step 3437 and 3439, it is determined whether a Poisson distribution with rate parameter equal to $Nr * kdecay r$ provides a random number $X \geq 1$. If not, no copy of that species is decayed or degraded and control passes back to step 3431 and following to select the next RNA species. Otherwise, a copy of that species is decayed or degraded in step 3441. Step 3441 includes decrement metabolite copy count Nm by Mmr for all m ; increment copy count of produced metabolites; decrement copy count of RNA species r ; decrement number of decay events DR (and DT if species is tRNA).

[0426] FIG. 34B is a list of cell state variables and other sub-models interacting with the RNA Decay sub-model, according to an embodiment.

[0427] For the *M. genitalium* example whole-cell model, RNA decay is modeled as requiring ribonuclease R (MG104). The decay of tRNAs also requires peptidyl tRNA hydrolase (Pth; MG083). Half-lives of all the RNA species are largely based on experimental measurements of homologous *E. coli* genes. The *E. coli* genes are mapped to the *M. genitalium* genes by homology.

[0428] FIG. 34C is a list of parameters and values for an RNA Decay sub-model, according to an embodiment applied in the *M. genitalium* example. For each time step, do the following.

1. Determine the counts of free ribonuclease R(NR) and peptidyl tRNA hydrolase (NT) from the Protein Monomer State Class
2. Determine the limits on the number of RNA decay events, DR: (The number of tRNA decay events is designated DT)
 - (a) If NR =0, then DR =0 (All RNAs require ribonuclease R in order to decay, but ribonuclease R has a high rate of activity, so as long as NR > 0, RNA decay can occur)
 - (b) DT = NTkhyd (The tRNA decay limit is dependent on the peptidyl tRNA hydrolase availability and kinetic rate)
3. Considering each RNA independently, decide whether to decay an RNA by random number selection from a Poisson distribution with a rate parameter, λ calculated as in Equation 19.

$$\lambda = RNA_i * k_{decay\ i} \quad (19)$$

where RNA_i is the count of a given RNA species i

4. If aborted transcripts exist in the cell due to RNA polymerase stalling or RNA polymerase displacement on the DNA, decay all aborted sequences. Calculate the total number of metabolites used and produced from the decay of the given RNA species, using the matrix Mdecay, and update the counts of existing RNAs and metabolites.

4.14 Translation sub-model

[0429] Translation is the process whereby the ribosome, accessory enzymes, and tRNAs transcode mRNAs and produce amino acid polymers, also called polypeptides, which when completed form protein monomers. Translation begins with the recruitment of the 30S and 50S ribosomal particles (also called subunits) and an initiation factor (e.g., initiation factor 3, IF3) to an mRNA molecule. Next, the ribosomal particles assemble into a ribosome (e.g., 70S ribosome) on the mRNA molecule with the help of initiation factors. Third, the ribosome polymerizes amino acids presented by aminoacylated tRNAs with the help of elongation factors. Finally, a release factor recognizes the stop codon UAG or UAA, hydrolyzes the peptidyl tRNA bond, and dissociates. A ribosome recycling factor dissociates the E-site tRNA, an elongation factor G releases the release factor and 50S ribosome, and an initiation factor 3 dissociates the 30S ribosome, P-site tRNA, and mRNA.

[0430] A ribosome may stall for various reasons including collisions with other proteins and limited resources required of translation. When a ribosome stalls, its last tRNA is expelled and replaced by the tRNA-like domain of a tmRNA molecule. Next, the mRNA-like domain of the tmRNA expels the bound mRNA. Third, the ribosome resumes polymerization, now using the tmRNA's mRNA-like domain as its template. This results in the production of an amino acid polymer containing a C-terminal proteolysis tag. Finally, the proteolysis tag will be recognized by the protein degradation machinery, and the amino acid polymer will be degraded into its individual component amino acids. The probability that a ribosome stalled is an uncharacterized value, and therefore this probability is set to be very low, such that ribosome stalling is a rare event which does not typically limit the simulation.

[0431] The translation sub-model simulates protein translation by ribosomes and accessory initiation, elongation, and termination factors. Ribosomes transition from a free to active state if the necessary factors are present, and bind to an mRNA. The selection of a specific mRNA to bind to a ribosome is random and weighted by mRNA abundances. As the sequence of each gene is known, the codons presented by the mRNAs are translated using aminoacylated tRNAs and elongation factors at a rate of up to 16 amino acids per second (measured by radioactive labeling in *E. coli*). Once a stop codon has been reached,

translation is terminated and the polypeptide will undergo further modifications by other processes in the whole model simulation. The sub-model also simulates the identification of stalled ribosomes by the tmRNA, the replacement of the tRNA and mRNA with the tmRNA, and the synthesis of the proteolysis tag encoded by the tmRNA's mRNA-like domain.

[0432] The simulation begins in a state in which ribosomes are already bound to mRNAs and in the process of elongating. Each ribosome is randomly assigned (without replacement) to an mRNA species, weighted by the current expression of the mRNAs. Then, each ribosome is assigned to positions within the assigned mRNA with uniform probability. No ribosomes are initialized to the stalled state, since the expected probability of stalling is negligible. No tmRNAs are initiated to the bound state.

[0433] FIG. 35A and FIG. 35B are flowcharts that illustrate an example method 3500 for a Translation sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is r for translation reactions of mRNA species r . In step 3501, the protein monomers and complexes that serve as enzymes e for translation reactions (including ribosomes, initiation factors, IF, elongation factors, EF, and termination factors, TF, including release factors and recycle factors) are identified, as is the stoichiometry Mmr of metabolites in the reactions.

[0434] In steps 3503 and 3505 the next free ribosome is selected at random for which there is both an initiation factor and available energy. If none, control passes to step 3521 to select the next bound ribosome. Otherwise, in step 3511 a random mRNA species r is selected based on a multinomial distribution with probability of each species based on the copy count of free members of that species. In step 3513, the free ribosome is bound to a copy of the selected mRNA species r . the free counts of both ribosomes and RNA species r are decremented and their bound counts are incremented. Control passes back to step 3503 to select the next free ribosome.

[0435] In steps 3521 and 3523 the next bound ribosome is selected at random for which there is both an elongation factor and needed metabolites. If none, control passes to step 3545 to test for stalling. Otherwise, elongation occurs. In step 3525 it is determined whether the

ribosome was first bound in the current time step. If so, then, in step 3631, the initiation factor (IF) is released, and copy counts for the free and bound IF updated; and, in step 3532 a start amino acid, f-methionine, is associated as a first element of the polypeptide to be produced. Control passes back to step 3521 to select the next bound ribosome at random. If this is not the first time step with the ribosome, then, in step 354, the elongation limit for the current time step is set to the elongation rate *kelong* for the time step. Control then passes to step 3533 in the loop for adding amino acids one by one to the polypeptide.

[0436] In step 3533, the next amino acid to be added is determined from the mRNA sequence. If the sequence is finished, then the polypeptide-ribosome complex is marked completed; and, next bound ribosome is inspected at step 3521. If the amino acid sequence is not yet finished, then if there are enough resources (elongation factors and metabolites, and aminoacylated tRNA with the correct amino acid, and elongation limit is not exceeded), as determined in step 3541, add the amino acid in step 3543, update the counts for the affected molecules, and return to step 3533 for the next amino acid in the sequence.

[0437] If there are not enough resources, then in step 3545 determine if there has been advance during current time step. If so, select the next bound ribosome, if any in step 3521. However, if there is no advance during the time step, then go to step 3551 to determine if the complex is to be considered stalled, based on a random number from a Poisson distribution with a small occurrence rate, e.g. 0.001 or less. The model would certainly work with $p \leq 0.00001$ and probably with $p \leq 0.01$. If the random number is one or more, consider the ribosome stalled and in step 3553 update the state variables and counts to reflect that status.

[0438] After every bound ribosome has advanced as much as resources allow, control passes to step 3561 in a loop for terminating completed polypeptides. If enough resources are available as determined in step 3561, and there is a completed polypeptide-ribosome complex, as determined in step 3563, then in step 3565 change ribosome and mRNA to free, and add protein monomer or RNA product counts to the state variables. Return to step 3561 and repeat the process until there are no more completed complexes or resources to free them. Then the method ends.

[0439] FIG. 35C is a list of cell state variables and other sub-models interacting with the Translation sub-model, according to an embodiment.

[0440] For the *M. genitalium* example embodiment, most of the parameters required for this sub-model were obtained from the literature or derived from the *M. genitalium* genome sequence. FIG. 35D is a list of parameters and values for a Translation sub-model, according to an embodiment. The example translation sub-model embodiment performs the following steps at each time step.

[0441] Initialization. As described in FIG. 35E, the Translation sub-model forms 70S ribosomes equal to the minimum of the 30S and 50S subunit copy numbers, and randomly positions each 70S ribosome on mRNA weighted by the copy number of each mRNA. FIG. 35E is a list of instructions that illustrates an example method for initializing ribosomes for a Translation sub-model, according to an embodiment. Each ribosome exists in one of two states, free or actively transcribing. Ribosomes are created in the free state and may transition to the actively transcribing state as follows: If ribosome factor A is present, initiation factors (IF-1, IF-2, and IF-3) are present, and one unit of energy (GTP) is available, then randomly select free ribosomes to initiate up to the limits of ribosome factor A, initiation factors, and GTP. Randomly select mRNA species for each initiating ribosome to bind to, weighted by the counts of each mRNA species. Update the state of each ribosome and mRNA, and the amount of available substrates.

[0442] Elongation. Next, the polypeptides are elongated. It is assumed that one of each elongation factor (EF-tu, TS, and G) is sufficient for each ribosome, but a separate set of factors may be useful for each ribosome. Elongation proceeds as follows: If elongation factors (EF-tu, TS, and G), aminoacylated tRNAs, and energy (GTP) are available, then randomly select actively transcribing ribosomes to elongate up to the limits of elongation factors. Available amino acids and energy are allocated among actively translating ribosomes. If the ribosome is newly initiated, then release the initiation factors; associate a tRNA for f-methionine to bind the first amino acid, and release a free tRNA. Otherwise, derive the amino acid sequence of the translating gene by converting the genome sequence into a tRNA sequence using the amino acid code; associate aminoacylated tRNAs to bind

amino acids to the growing polypeptide up to the aminoacylated tRNA limit, energy limit, and elongation limit, *kelong*. Release free tRNAs, then update the state of each ribosome, the progress of all actively translating ribosomes, and the amount of available substrates. Note, for cases in which translation of a peptide finishes partway through the time step, the elongation factors are released, but only available at the time step. For simplicity, the transition between P and E sites is not explicitly modeled in this embodiment.

[0443] Termination. Once all amino acids of a protein have been translated, one release factor, one recycling factor, one elongation factor G, and one GTP molecule are sufficient to terminate the polypeptide, as follows: If all release factors, recycling factors, elongation factors G, and energy (GTP) are available, then randomly select completed polypeptide-mRNA-ribosome complexes to dissociate; and update the state of each ribosome and mRNA, monomer counts, and the amount of available substrates. The ribosome will be available to bind mRNA at the following iteration

[0444] Translation Stalling. The stalling of ribosomes is dealt with as follows. If in a time step an elongating ribosome hasn't advanced, then with a small probability, *Pstalled*, the ribosome is transitioned to the stalled state. The mRNA is then expelled and replaced with a tmRNA. A proteolysis tag is encoded and the amino acid chain is marked for degradation by the Protein Decay sub-model.

4.15 Protein Processing I sub-model

[0445] Following translation, some cells, such as bacteria, deformylate the N-terminal methionine of most proteins and cleave the N-terminal methionine of approximately 7% of proteins. The function of peptide deformylation and methionine cleavage is not well understood. Some authors believe that bacteria employ these reactions to recycle formylmethionine which is a metabolically expensive amino acid. Other authors, citing the N-end rule, believe that bacteria cleave the N-terminal methionine of specific proteins in order to regulate protein half-lives by exposing the second-most N-terminal amino acid.

[0446] The Protein Processing I sub-model simulates N-terminal formylmethionine deformylation and N-terminal methionine cleavage, the first step of post-translational

processing. Because the mechanisms of early protein processing are not well characterized on the genomic scale, this sub-model makes several simplifying assumptions. First, this process assumes that each protein is both deformed and cleaved in a single time step. Consequently, this process collapses the processing of each protein into a single reaction, and only represents nascent and fully processed proteins. Intermediate protein configurations are not represented. Second, this sub-model assumes that the mean arrival rate, ν_i , of processing events of each protein species i is independently limited by (1) the copy number of unprocessed protein, p_i , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of processing enzymes, e_j . Based on these assumptions, the functional form of ν_i is given by Equation 20.

$$\nu_i = \min \left(\overbrace{p_i}^{\text{Protein}}, \underbrace{\left[\min_j \frac{m_j}{M_{ji}^*} \right]}_{\text{metabolism}}, \underbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}_{\text{enzyme}} \right) \quad (20)$$

Where the symbols are as defined previously, e.g., with respect to Equation 16 and Equation 17.

[0447] The Protein Processing I sub-model implements a stochastic model of the arrival of early protein monomer processing events with relative rates ν_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, ν_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by ν_i , and (3) executes the selected processing reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity.

[0448] The Protein Monomer and Protein Complex states represent the copy number of each protein processing enzyme. The Protein Monomer state also represents the nascent and processed copy numbers of each protein monomers species. The Metabolite state represents the copy number of each intracellular metabolite. Proteins are synthesized and matured in six steps. This sub-model simulates the first step in post-translational processing following

translation (see Translation sub-model) and preceding membrane and extracellular protein translocation and cytosolic protein folding (see Protein Translocation and Protein Folding sub-models). The nascent and processed configurations of each protein monomer species are initialized with zero copy number because the typical occupancy these configurations is small compared to that of the mature configuration. The expression of the protein processing enzymes was fit to provide sufficient enzymes to quickly process proteins, or more specifically to prevent sustained accumulation of unprocessed proteins.

[0449] FIG. 36A is a flowchart that illustrates an example method 3600 for a Protein Processing I sub-model, according to an embodiment. This flowchart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is b for protein processing reactions. A protein not yet processed by processing I is called a nascent protein. The flow is similar to the flow of the stochastic models for RNA processing (FIG. 31A) and RNA modification (FIG. 32A). In step 3601, the protein monomers and complexes that serve as enzymes for protein processing I reactions are identified as is the stoichiometry Mmb of metabolites in the reactions, and catalytic rate Keb of enzymes in the reaction. In step 3603 the enzyme capacity Ce during a single time step is set to the count of that enzyme.

[0450] In step 3611 the next nascent protein species b , if any, is determined; and in step 3613 Equation 20 is evaluated to determine Vb (corresponding to vi). This is repeated until all metabolite or enzyme or nascent protein is consumed, at which point $Vb = 0$ for all b , and the method ends. When a Vb has been calculated for every nascent protein species, control passes to step 3621 to randomly select a reaction to implement using a multinomial distribution with probabilities based on the arrival rates Vb .

[0451] In step 3623, the RNA processing reaction is run to process one copy of nascent protein species b . The copy count of the protein species is updated (nascent protein count is decremented and processed protein count incremented). The metabolite counts for metabolites m are decremented by Mmb and the enzyme capacity Ce is also decremented. Control passes back to step 3611 to recompute Vb for all b based on the new counts.

[0452] *M. genitalium* peptide deformylase (MG106) was assumed to deformylate all protein monomers. The protein substrates of *M. genitalium* methionine aminopeptidase (MG172) were reconstructed based on specific N-terminal methionine cleavages of *M. genitalium* homologs observed. The kinetics of peptide deformylation and N-terminal methionine cleavage were reconstructed from the primary literature.

[0453] FIG. 36B is a list of instructions that illustrates an example Protein Processing I sub-model, according to an embodiment for the *M. genitalium* example.

4.16 Protein Translocation sub-model

[0454] Cell proteins are translated in the cytosol. However, many functionally important proteins including nutrient transporters, ATP synthase, metabolic enzymes, adhesins, receptors, transducers, virulence factors, and the protein translocation machinery itself must be embedded in the cell membrane or secreted in order to properly function. Cells employ short N-terminal and C-terminal signal sequences to selectively translocate proteins. This sub-model simulates membrane and extracellular protein localization, the second step in post-translational processing.

[0455] The subcellular localization—cytosol, integral membrane, lipoprotein, terminal organelle cytosol, terminal organelle lipoprotein, or extracellular space—of each protein monomer and the N-terminal signal sequence of each lipoprotein and secreted protein was reconstructed based on an extensive review of the primary literature, several proteomic databases, and several computational predictions. A result of the reconstruction process is a list of the localization and signal sequence of each protein monomer.

[0456] The Protein Translocation sub-model simulates integral membrane, lipoprotein, and secreted protein translocation into the cell membrane by a translocation enzyme (such as the SecA preprotein translocase), the second step of post-translational processing. Proteins are synthesized and matured in six steps. The Protein Translocation sub-model simulates protein translocation following deformylation and N-terminal methionine cleavage and preceding lipoprotein diacylglycerol transfer and lipoprotein and secreted protein signal sequence cleavage. Because the mechanisms of protein translocation are not well characterized on the

genomic scale, this sub-model makes several simplifying assumptions, similar to those made for other sub-models. First, this process decouples translation, N-terminal formylmethionine processing, and translocation by assuming that translocation does not begin until after translation termination and N-terminal formylmethionine processing. Second, this sub-model assumes that each protein is fully translocated in a single time step. Consequently, this process collapses the translocation of each protein into a single reaction, and only represents untranslocated and fully translocated proteins. Intermediate protein localizations are not represented. Third, this process assumes that the mean arrival rate, v_i , of translocation events of each protein species i is independently limited by (1) the copy number of untranslocated protein, p , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of the translocation enzymes, e_j . Based on these assumptions, the functional form of v_i is given by Equation 21,

$$v_i = \min \left(\overbrace{p_i^u}^{\text{Protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^*} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (21)$$

where the notation is as described above. The metabolites for this reaction include ATP and GTP hydrolysis.

[0457] The Protein Translocation sub-model implements a stochastic model of the arrival of protein monomer translocation events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each translocation event, (2) selects a single translocation event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected translocation reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity.

[0458] The untranslocated and translocated configurations of each protein monomer species are initialized with zero copy counts because the typical occupancy of these configurations is small compared to that of the mature configuration. The expression of the

protein translocation enzymes was fit to provide sufficient enzymes to quickly translocate proteins, or more specifically to prevent sustained accumulation of untranslocated proteins.

[0459] FIG. 37A is a flowchart that illustrates an example method 3700 for a Protein Translocation sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is b for protein processing reactions. A protein not yet translocated is called an untranslocated protein. The flow is similar to the flow of the stochastic models for protein processing I (FIG. 36A) and will not be repeated here. In step 3713, Equation 21 is evaluated as expressed in the revised notation.

[0460] *M. genitalium* employs two convergent SecA-dependent pathways—co-translational and post-translational—to translocate approximately 35-45% of all protein monomers into the plasma membrane. *M. genitalium* does not contain a type I Tat transporter or sortase. The co-translational SecA pathway translocates integral membrane proteins into the cell membrane. First, GTP-dependent signal recognition particles (SRP; MG0001, MG048) and molecular chaperones co-translationally recognize the type I signal sequence of nascent integral membrane proteins. Second, SRPs deliver nascent proteins to their cognate receptor FtsY (MG297) which is associated with the SecA type II preprotein translocon. The preprotein translocase SecA (MG072) iteratively pushes nascent proteins through the SecYEG (MG170, MG055, MG476) translocase pore by an ATP-dependent, step-wise mechanism. Several studies have shown that SecDF (MG277) and YidC (MG464) also participate in the translocase pore. SecDF and YidC are believed to increase the efficiency of protein translocation. The exact functions of SecDF and YidC are not known.

[0461] The *M. genitalium* post-translational SecA pathway translocates lipoproteins and secretes proteins. First, nascent proteins complete translation. Second, SecA translocons directly recognize the type II signal sequence of nascent proteins. Type II signal sequences are short, positively charged N-terminal sequences. Type II signal sequences are composed of three regions: n, h, and c. The n region is composed of 1-5 positively charged amino acids. The h region is composed of 7-15 hydrophobic amino acids and forms an α -helix. The c region is composed of 3-7 polar amino acids and forms a β -strand. Signal peptidase II cleaves

type II signal sequences at lipoboxes distinguished by the sequence L[ASI][GA]C inside the c region. Similarly to the co-translational pathway, SecA iteratively translocates nascent proteins into the plasma membrane.

[0462] Following translocation *M. genitalium* anchors lipoproteins to the outer leaflet through the covalent ligation of diacylglycerol by diacylglycerol transferase and extracellularly cleaves the type II signal sequence of each lipoprotein and secreted protein at conserved lipoboxes. *M. genitalium* does not have an I signal sequence protease and does not cleave the signal sequences of integral membrane proteins. Integral membrane protein signal peptides are believed to help anchor proteins to the membrane.

[0463] Following insertion into the plasma membrane and signal peptide cleavage, terminal organelle-localized lipoproteins are recruited into the terminal organelle. The stoichiometry, energetics, and kinetics of each protein translocation reaction were reconstructed from the primary literature. It has been reported that SecA translocates 270 picoMoles (pmol, 1 pmol = 10⁻¹² Moles) amino acid per minute. It has also been reported that SecA translocates 20-30 amino acids per ATP.

[0464] FIG. 37B is a list of instructions that illustrates an example Protein Translocation sub-model, according to an embodiment for the *M. genitalium* example.

4.17 Protein Processing II sub-model

[0465] This sub-model addresses the third step of post-translational processing: lipoprotein diacylglycerol adduction and lipoprotein and secreted protein signal peptide cleavage. The localization and N-terminal signal sequence length of each protein monomer was reconstructed based on an extensive review of the primary literature, several proteomic database, and several computational predictions. The stoichiometry and kinetics of each protein processing reaction was reconstructed from the primary literature.

[0466] Because the mechanisms of protein processing are not well characterized on the genomic scale, this sub-model makes several simplifying assumptions. First, this process assumes that each protein is both covalently modified and cleaved in a single time step. Consequently, this process collapses the processing of each protein into a single reaction and

only represents unprocessed and fully processed proteins. Intermediate protein configurations are not represented. Second, this process assumes that the mean arrival rate, v_i , of processing events of each protein species i is independently limited by (1) the copy number of unprocessed protein, p , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of processing enzymes, e_j . Based on these assumptions, the functional form of v_i is given by Equation 22,

$$v_i = \min \left(\overbrace{p_i^u}^{\text{Protein}}, \underbrace{\left[\min_j \frac{m_j}{M_{ji}^*} \right]}_{\text{metabolites}}, \underbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}_{\text{enzyme}} \right) \quad (22)$$

where the notation is as described above.

[0467] This sub-model implements a stochastic model of the arrival of protein monomer processing events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected processing reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity. The unprocessed and processed configurations of each protein monomer species are initialized with zero copy number because the typical occupancy these configurations is small compared to that of the mature configuration.

[0468] FIG. 38A is a flowchart that illustrates an example method 3800 for a Protein Processing II sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is b for protein processing reactions. The flow is similar to the flow of the stochastic models for protein processing I (FIG. 36A) and will not be repeated here. In step 3813, Equation 22 is evaluated as expressed in the revised notation.

[0469] *M. genitalium* lipoproteins are translated in the cytosol and subsequently targeted to the plasma membrane by type II N-terminal signal sequences. Following membrane

insertion, *M. genitalium* lipoproteins are first anchored to the outer leaflet of the plasma membrane. This is achieved via covalent adduction of diacylglyceryl to the sulfhydryl group of the lipobox cysteine by diacylglyceryl transferase (MG086). Many bacteria further anchor lipoproteins through phospholipidation. However, *M. genitalium* does not have an apolipoprotein transacylase, and is not believed to phospholipidate lipoproteins. Second, lipoprotein N-terminal signal sequences are cleaved immediately C-terminal to the lipobox cysteine by type II signal peptidase (MG210).

[0470] Extracellular *M. genitalium* proteins are transcribed in the cytosol and targeted to the plasma membrane by type II signal sequences. In contrast to lipoproteins, extracellular proteins do not undergo diacylglyceryl transfer, and instead are cleaved immediately C-terminal to the cysteines of their lipoboxes by type II signal peptidase, releasing the resultant protein and signal peptide into the extracellular space.

[0471] *M. genitalium* does not have a type I signal sequence protease and does not cleave the signal sequences of integral membrane proteins. Integral membrane protein signal peptides are believed to help anchor proteins to the membrane.

[0472] FIG. 38B is a list of instructions that illustrates an example Protein Processing II sub-model, according to an embodiment for the *M. genitalium* example.

4.18 Protein Folding sub-model

[0473] Proteins are synthesized as long, catalytically inactive linear chains of amino acids. Subsequently, proteins fold into energetically favorable, compact, and enzymatically competent three-dimensional structures. While some protein species are believed to fold spontaneously, other proteins are believed to require helper chaperone proteins to properly fold. In addition, some protein species require metal ions and other small molecule prosthetic groups to fold. This process models chaperone-mediated protein folding. The prosthetic group requirements for protein folding were reconstructed based on an extensive review of the literature and several databases.

[0474] The Protein Folding sub-model implements a model of the chaperone-mediated folding of processed, translocated polypeptides. Because the kinetics and energetics of

protein folding are not well understood, this process represents the three-dimensional configuration of each protein as a two-state (folded, unfolded) Boolean variable. Furthermore, this process makes the simplifying assumption that the folding rate, r_i , of protein species i is a Boolean function of the copy numbers of metabolites and chaperones.

$$r_i = \min \left(\overbrace{p_i^u}^{\text{protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}} \right]}^{\text{metabolites}}, \overbrace{\min_j \frac{e_j}{C_{ji}}}^{\text{enzymes}} \right) \quad (23)$$

where P_{iu} is the unfolded copy number of protein species i , m_i is the copy number of metabolite i , e_i is the copy number of chaperone i , M_{ji} is the stoichiometry of prosthetic group j in protein i , and C_{ji} is true if protein species i requires chaperone j to fold. p_{if} is the folded copy number of protein species i .

[0475] Protein folding is one of the last steps in protein maturation following protein processing and translocation and preceding protein modification. The Protein Monomer and Protein Complex states initialize the total copy number of each protein species, and initially set all protein monomers and complexes to their mature—folded and modified—configuration.

[0476] FIG. 39A is a flowchart that illustrates an example method 3900 for a Protein Folding sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is b for protein processing reactions. The flow is similar to the flow of other stochastic models for protein processing. In step 3901, the protein monomers and complexes that serve as enzymes (chaperones) for protein folding reactions are identified, as is the stoichiometry Mmb of metabolites in the reactions, and Boolean chaperone requirement $CREb$ for enzyme e to fold protein b in the reaction.

[0477] In step 3911 the next unfolded protein species b , if any, is determined; and in step 3913 Equation 23 is evaluated to determine V_b (corresponding to v_i) using the revised notation. $V_b = \text{minimum of } \{N_b; \text{minimum over all } m \text{ of } N_m / \text{MAX} [0, -M_{mb}]\}$; random number X from a Poisson distribution with occurrence rate = minimum over all e of $N_e / C_{Reb}\}$. This is repeated until all metabolite or chaperone or unfolded protein is consumed, at which point $V_b = 0$ for all b , and the method ends. When a V_b has been calculated for every nascent protein species, control passes to step 3921 to randomly select a reaction to implement using a multinomial distribution with probabilities based on the arrival rates V_b , normalized here by the count N_b of unfolded proteins of species b .

[0478] In step 3923, the protein folding reaction is run to fold one copy of unfolded protein species b . The copy count of the protein species is updated (unfolded protein count is decremented and folded protein count incremented). The metabolite counts for metabolites m are decremented by M_{mb} . Control passes back to step 3911 to recompute V_b for all b based on the new counts.

[0479] *M. genitalium* is believed to employ three chaperones to assist protein folding. First, trigger factor Tig (MG238) co-translationally binds all nascent polypeptides at the ribosome exit site (L23) and assists in early protein folding. Second, chaperones GroEL (MG392) and DnaK (MG305) and their co-chaperones are believed to assist in late folding. GroEL and DnaK are believed to fold 10-15% and 5-18% of all proteins, respectively. GroEL and its co-chaperone GroES (MG393) are believed to help fold intermediate sized proteins (20-60 kDa). GroEL is believed to bind each protein for 30 s to 10 min, and to couple folding to ATP hydrolysis. DnaK and its co-chaperones DnaJ (MG019) and GrpE (MG201) are believed to help fold large proteins (> 30 kDa). DnaK is a monomeric protein which transiently (< 2 min) binds to the backbones of short, linear, unfolded peptide segments. GrpE is believed to couple peptide release to ATP hydrolysis. DnaJ is believed to regulate the activity of DnaK, as well as bind the side chains of hydrophobic and aromatic residues and directly assist protein folding. FtsH and the lipids phosphatidyl ethanolamine and phosphatidyl glycerol have been suggested to assist membrane protein folding. However, the role of FtsH as a molecular chaperone is not well established, and FtsH has been associated with several

additional functions. Consequently, it was decided not to model FtsH as a chaperone. The contributions of phosphatidyl ethanolamine and phosphatidyl glycerol to protein folding are also poorly understood, and are not modeled in the example embodiments.

[0480] *M. genitalium* GroEL substrates were reconstructed based on two proteome-scale studies of *E. coli* and *B. subtilis* (see Table S3AG). *M. genitalium* DnaK substrates were reconstructed based on a proteome scale study of *E. coli*.

[0481] FIG. 39B is a list of instructions that illustrates an example Protein Folding sub-model, according to an embodiment for the *M. genitalium* example.

4.19 Protein Modification sub-model

[0482] Post-translational protein modification serves several important functions. First, post-translation modification can increase the structural and chemical diversity of the proteome by stabilizing alternative conformations and providing catalytic cofactors. Second, post-translational modification, and in particular phosphorylation, provides a mechanism to regulate protein activity. Third, post-translational modification can be used to regulate protein expression through proteasome recruitment. This process models protein covalent modification including phosphorylation, lipoyl transfer, and α -glutamate ligation.

[0483] The Protein Modification sub-model simulates protein covalent modification, the fifth step of post-translational processing. In particular, this process models protein phosphorylation, lipoyl transfer, and α -glutamate ligation. Because the mechanisms of protein modification are not well characterized on the genomic scale, this sub-model makes several simplifying assumptions. First, this sub-model assumes that each protein is fully modified in a single time step, collapses the modification of each protein into a single reaction, and only represents unmodified and fully modified proteins. Intermediate protein configurations are not represented. Second, this sub-model assumes that the mean arrival rate, v_i , of modification events of each protein species i is independently limited by (1) the copy number of unmodified protein, p , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of the protein modification enzymes, e_j . Based on these assumptions, the functional form of v_i is given by Equation 24.

$$v_i = \min \left(\overbrace{P_i^s}^{\text{Protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzyme}} \right) \quad (24)$$

where M_{ji} is the stoichiometry of metabolite j in the modification of protein species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the modification of protein species i , and Δt is the simulation time step.

[0484] This sub-model implements a stochastic model of the arrival of protein monomer and macromolecular complex modification events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each modification event, (2) selects a single modification event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected modification reaction, updating the copy numbers of proteins and metabolites and decrementing the available enzymatic capacity.

[0485] The Protein Monomer state and Protein Complex state represent the copy number of each protein modification enzyme. The Protein Monomer state also represents the unmodified and modified copy numbers of each protein monomers species. The Metabolite state represents the copy number of each intracellular metabolite.

[0486] The unmodified configurations of each protein monomer species is initialized with zero copy number because the typical occupancy of this configuration is small compared to that of the modified/mature configuration. The expression of the protein modification enzymes was fit to provide sufficient enzymes to quickly modify newly synthesized proteins, or more specifically to prevent sustained accumulation of unmodified proteins.

[0487] FIG. 40A is a flowchart that illustrates an example method 4000 for a Protein Modification sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of enzyme index e is Ne , and the reaction index is b for protein processing reactions. A protein not yet modified is called an unmodified protein. The flow is similar to the flow of the stochastic models for Protein Processing I, among others. In step 4001, the protein monomers and complexes that serve as enzymes for protein modification reactions are identified as is the stoichiometry Mmb of metabolites in the reactions, and catalytic rate Keb of enzymes in the reaction. In step 4003 the enzyme capacity Ce during a single time step is set to the count of that enzyme.

[0488] In step 4011 the next unmodified protein species b , if any, is determined; and in step 4013 Equation 24 is evaluated to determine Vb (corresponding to vi) using the revised notation. This is repeated until all metabolite or enzyme or unmodified protein are consumed, at which point $Vb = 0$ for all b , and the method ends. When a Vb has been calculated for every unmodified protein species, control passes to step 4021 to randomly select a reaction to implement using a multinomial distribution with probabilities based on the arrival rates Vb .

[0489] In step 4023, the RNA processing reaction is run to process one copy of unmodified protein species b . The copy count of the protein species is updated (unmodified protein count is decremented and modified protein count incremented). The metabolite counts for metabolites m are decremented by Mmb and the enzyme capacity Ce is also decremented. Control passes back to step 4011 to recompute Vb for all b based on the new counts.

[0490] The *M. genitalium* protein modification complement was reconstructed in three steps. First, protein modifications that have been observed in *M. genitalium*, *M. pneumoniae*, or other related bacteria or which have been computationally predicted were curated. Second, curated protein modifications were mapped to *M. genitalium* homologs using sequence alignment. Third, based on the genome annotation, *M. genitalium* was determined to have three protein modification pathways: (1) phosphorylation catalyzed by serine/threonine kinase PrkC (MG109), (2) lipoyl transfer to lysine catalyzed by LplA (MG270), and (3) C-terminal α -glutamate ligation catalyzed by RimK (MG012). Fourth, modifications were

rejected which do not belong to one of these three pathways, or for which a specific locus was not reported. The reconstructed protein modification network contains one kinase that modifies 16 proteins, one lipoyl transferase that modifies the E2 subunit of pyruvate dehydrogenase (PdhC, MG272), providing an important organosulfur cofactor which contains a catalytic disulfide bond, and one α -glutamate ligase that modifies 50S ribosomal protein L6 (RplF, MG166). The stoichiometry and kinetics of all three modeled protein modification reactions were based on a review of the primary literature. Catalytic disulfide bonds were separately reconstructed and are modeled by several chemical reactions in the Metabolism sub-model.

[0491] FIG. 40B is a list of instructions that illustrates an example Protein Modification sub-model, according to an embodiment for the *M. genitalium* example.

4.20 Protein Complexation sub-model

[0492] Many enzymes are functional as multimeric proteins or ribonucleoproteins. Macromolecular complexes are believed to form quickly ($k_{on} \approx 10^3 - 10^6 \text{ M}^{-1} \text{ s}^{-1}$), energetically favorably ($\Delta G \approx -12 \text{ kcal mol}^{-1}$), and stably ($KD \approx 10^{-4} - 10^{-10} \text{ M}$). This process models the formation of all macromolecular complexes except the 30S and 50S ribosomal particles, the 70S ribosome, the FtsZ ring, and the oriC DnaA complex. The Ribosome Assembly, Translation, FtsZ Polymerization, and Replication Initiation processes model the more complex and better characterized formation of these complexes.

[0493] The Protein Complexation sub-model simulates the formation of macromolecular complexes as a stochastic process. Because macromolecular Complexation is poorly understood, this sub-model makes several simplifying assumptions. First, this process assumes that macromolecular complexes form spontaneously, uncoupled to other chemical reactions and without assistance from chaperones or other proteins. The contribution of chaperones to the three-dimensional folding of individual protein monomers is modeled by the Protein Folding process. Second, this sub-model assumes that macromolecular complexation is fast and proceeds to completion within the 1s time step of the simulation. Third, this process makes the simplifying assumption that each macromolecule complex

forms with the same specific rate. Fourth, this sub-model assumes that complexes form by simultaneous collision of each subunit. Together these assumptions imply that the relative formation rate, r_i , of each complex, i , is described by mass-action kinetics, as given by Equation 25.

$$r_i = \prod_j \left(\frac{m_j}{V} \right)^{S_{ij}} \quad (25)$$

where m_j is the copy number of gene product j , V is the cell volume, and S_{ij} is the stoichiometry of subunit j in complex i . This enables the sub-model to simulate complex formation by (1) calculating the relative formation rate of each complex, r_i , (2) randomly forming one complex according to a multinomial distribution defined by the relative formation rates, (3) updating the copy numbers of RNA and protein subunits and complexes, and (4) repeating until insufficient protein and RNA subunits are available to form additional complexes. Importantly, this algorithm resolves the order of macromolecular complex formation before several subunits participate in multiple complexes.

[0494] After the cell volume and the total copy number of each RNA and protein gene product are initialized (see Geometry, RNA, and Protein Monomer states), the copy count of each macromolecular complex is initialized by iteratively evaluating the dynamic model until convergence, or more specifically until insufficient subunits are available to form additional complexes.

[0495] The RNA and Protein Monomer states represent the copy count of each RNA and protein subunit synthesized by the RNA and protein synthesis pathways. The Protein Complex state represents the copy count of each complex, ci . The Geometry state represents the cell volume.

[0496] FIG. 41A is a flowchart that illustrates an example method 4100 for a Protein Complexation sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of gene product index g is Ng , where gene products include both protein monomers produced by translation and RNA produced by RNA polymerase reactions, and the reaction index is x for complex processing reactions to produce complex x . The gene products to be combined in the complex are also called subunits. The flow is similar to the flow of the stochastic models for Protein Processing I, among others. In step 4101, the protein monomers and RNA that serve as subunits of complexation reactions are identified, as is the stoichiometry S_{xg} of gene products in the reaction x . There are no enzymes in the modeled complexation reactions.

[0497] In step 4111, the next complex species x for assembly, if any, is determined; and in step 4113 Equation 25 is evaluated to determine V_x (corresponding to ri) using the revised notation. This is repeated until all subunits are consumed, at which point $V_x = 0$ for all x , and the method ends. When a V_x has been calculated for every complex species, control passes to step 4121 to randomly select a reaction to implement using a multinomial distribution with probabilities based on the arrival rates V_x .

[0498] In step 4123, the Complexation reaction is run to assemble the subunits of one complex x . The copy count of the complex and gene products are updated (free gene products counts are decremented by S_{xg} and complex copy count incremented). Control passes back to step 4111 to recompute V_x for all x based on the new counts.

[0499] An extensive review of the primary literature and several databases suggested that *M. genitalium* forms 201 distinct macromolecular complexes containing 269 protein and 5 RNA gene products.

[0500] FIG. 41B is a list of instructions that illustrates an example Protein Complexation sub-model, according to an embodiment.

4.21 Ribosome Assembly sub-model

[0501] Ribosomes are large ribonucleoproteins which synthesize polypeptides. Thus ribosomes are a special case of protein complexes.

[0502] The Ribosome Assembly sub-model makes several simplifying assumptions to model ribosomal particle assembly. First, the process only represents individual rRNA transcripts and protein monomers and fully formed ribosomal particles. Intermediate states of ribosomal particle formation are not represented. Second, this process assumes that ribosomal particle formation is kinetically fast and energetically favorable such that ribosomal particle formation proceeds to completion within the simulation time step. Finally, the process assumes that each GTPase hydrolyzes 1 GTP molecule per ribosomal particle. With these assumptions, the rate of formation of ribosomal particle i is given by Equation 26.

$$\Delta c_i = \begin{cases} \min \left(\overbrace{\min_j \frac{r_j}{R_{ji}}}^{\text{rRNA}}, \overbrace{\min_j \frac{p_j}{P_{ji}}}^{\text{protein}}, \overbrace{\frac{m_{GTP}}{\eta_i}}^{\text{GTP}}, \overbrace{\left(\min_j \frac{e_j}{E_{ji}} \right)}^{\text{GTPase}} > 0 \right) \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

where m , r , p , c , and e represent the copy number of each metabolite, rRNA transcript, protein monomer, ribosomal particle, and GTPase, R_{ij} and P_{ij} represent the stoichiometry of rRNA transcript or protein monomer i in ribosomal particle j , E_{ij} is true if ribosomal particle j formation requires GTPase i and false otherwise, and $\eta_i = E_{ij}$ is the GTP cost of forming ribosomal particle i .

[0503] The sub-model implements a stochastic model of the arrival of ribosomal particle assembly events with relative rates Δc_i . Until RNA, protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, Δc_i , of each assembly event, (2) selects an event to execute according to a binomial distribution parameterized by Δc_i , and (3) executes the selected assembly reaction, updating the copy numbers of RNA, protein, and metabolites and decrementing the available enzymatic capacity.

[0504] The RNA and Protein Monomer states represents the free copy numbers of each RNA and protein monomer species. The Protein Complex state represents the copy numbers

of ribosomal particles and the ribosome. The Ribosome state represents the (t)mRNA location of each ribosome.

[0505] After the Mass state initializes the total cell mass, the RNA and Protein Monomer states initialize the total copy number of each rRNA and protein monomer species. Next, the Ribosome Assembly sub-model initializes the total copy counts of ribosomal particles to a steady-state of the ribosome assembly sub-model by evaluating the sub-model with excess GTP. The Translation sub-model initializes the copy number of ribosomes, and randomly positions each ribosome on mRNA weighted by the expressed copy number of each codon. The expression of each ribosomal RNA and protein monomer was fit to provide sufficient ribosomes for translation and prevent sustained amino acid accumulation.

[0506] FIG. 42A is a flowchart that illustrates an example method 4200 for a Ribosome Assembly sub-model, according to an embodiment. This flow chart uses the notation that the count of metabolite index m is Nm , the count of gene product index g is Ng , where a gene products include both protein monomers produced by translation and RNA produced by RNA polymerase reactions, and the reaction index is x for complex processing reactions to produce complex x . The gene products to be combined in the complex are also called subunits.

[0507] In step 4201 the parameter values are set: protein IDs of ribosome subunits; RNA IDs of ribosome subunits; stoichiometry Rxr of RNA species r in reaction x for assembling ribosome species x ; stoichiometry Pxb of protein monomer species b in reaction x for assembling ribosome species x ; Emx is true if ribosome species x requires metabolite m (e.g., GTPases and H₂O) and $\eta_x = \sum Emx$ over all GTPase m is the GTP (energy) cost of assembling a copy of ribosome species x .

[0508] In steps 4211 and 4213 the next ribosome is identified with enough resources (metabolites and GTP); and in step 4217 the rate of formation ΔNx of copies of ribosome x is determined using Equation 25 with the revised notation. If not enough resources are detected in step 4213, then in step 4215 ΔNx is set to 0. In step 4221 the reaction x is run to produce ΔNx copies of the ribosome and update the copy counts of consumed and produced

molecules. The next random ribosome to produce is selected from a binomial distribution with parameter = ΔN_x , and control returns to step 4211, until there are no ribosomes left to be assembled.

[0509] The *M. genitalium* 70S ribosome is composed of two subunits – the 30S and 50S ribosomal particles – which assemble at the mRNA Shine Dalgarno sequence with assistance from initiation factors. The *M. genitalium* example embodiment of this sub-model simulates the enzyme-catalyzed formation of 30S and 50S ribosomal particles. The 30S particle is composed of 1 RNA and 20 protein monomer subunits. The 50S particle is composed of 2 RNA and 32 protein monomer subunits. The 30S and 50S particles have both been shown to assemble in stereotyped patterns with assistance from several GTPases. Era (MG387) and RbfA (MG143) have been associated with 30S particle formation. EngA (MG329), EngB (MG335), Obg (MG384), and RbgA (MG442) have been associated with 50S particle formation. The exact functions, kinetics, and energetics of the six GTPases are unknown.

[0510] FIG. 42B is a list of instructions that illustrates an example Ribosome Assembly sub-model, according to an embodiment.

4.22 Organelle Assembly sub-model

[0511] Various organisms have cells with one or more special organelles, such as Golgi bodies. These organelles are described with a hierarchical assembly of protein content. The Organelle Assembly sub-model implements a Boolean model of the observed hierarchical assembly of the protein content, for example of the *M. genitalium* terminal organelle. After the total cell mass is initialized, the Protein Monomer state initializes the organelle copy number of each organelle protein.

[0512] FIG. 43A is a flowchart that illustrates an example method for a Organelle Assembly sub-model, according to an embodiment. In step 4301, set parameters for special organelle assembly: protein IDs of protein species found in special organelle (mitochondria, golgi bodies, terminal organelle...); protein IDs of organelle assembly enzymes; stoichiometry Tmr of metabolite m in reaction b for translocating protein monomer species b

to organelle; stoichiometry Amr of metabolite m in reaction b for assembling protein monomer species b in organelle.

[0513] In step 4311 it is determined whether there is a next protein species b for translocation. If so, then in step 4313 it is determined if there are sufficient resources to translocate at least one protein, in terms of translocation enzymes and metabolites. If so, then in step 4315 copies of the protein species are localized in organelle; copy count of protein in organelle compartment is incremented; copy count of protein in cytosol is decremented; consumed metabolite counts are decrement and released metabolite counts are incremented. Control returns to step 4311 to select the next protein species. In some embodiments, the protein species are retrieved randomly.

[0514] If there are no more protein species to translocate or resources to move them, then control passes to step 4321. In step 4321 it is determined whether there is a next protein species b for assembly in the organelle compartment. If so, then in step 4323 it is determined if there are sufficient resources to assemble at least one protein, in terms of assembling enzymes and metabolites. If so, then in step 4325, copies of the protein species are assembled in organelle; copy counts of protein and metabolites are updated. Control returns to step 4321 to select the next protein species. In some embodiments, the protein species are retrieved randomly. If there are no more protein species to assemble or resources to assemble them, then the method ends.

[0515] *M. genitalium* maintains a flask shape with a single 300 × 80 nm membrane-bound bleb or terminal attachment organelle throughout most of its life cycle. The *M. genitalium* terminal organelle divides during cell division, producing a daughter organelle which subsequently migrates to the opposite pole. The *M. genitalium* terminal organelle has been associated with several cellular processes including motility, adhesion, replication, and cytokinesis. In an example embodiment, a Terminal Organelle sub-model models the assembly of the protein content of the terminal organelle.

[0516] The terminal organelle is composed of eight proteins – high molecular weight cytodherence accessory proteins (HMW) 1-3 (MG312, MG218, MG317), adhesins MgPa (MG191), P32 (MG318) and P65 (MG217), and proteins P110 (MG192) and P200

(MG386)406–409. The terminal organelle protein content is believed to assemble in the stereotyped hierarchical pattern illustrated in FIG. 43B. FIG. 43B is a block diagram of hierarchical protein organization of a Terminal Organelle, according to an embodiment. First, HMW1 and HMW2 mutually recruit each other into the terminal organelle. Second, HMW1 recruits MgPa, HMW3, and P200 into the terminal organelle. Third, HMW3 recruits P32 which in turn recruits P65. P110 independently localizes to the terminal organelle. The kinetics of terminal organelle assembly are unknown.

[0517] The Terminal Organelle sub-model implements a Boolean model of the observed hierarchical assembly of the protein content of the *M. genitalium* terminal organelle. After the total cell mass is initialized, the Protein Monomer state initializes the terminal organelle copy number of each terminal organelle protein. FIG. 43C is a list of instructions that illustrates an example Terminal Organelle Assembly sub-model, according to an embodiment.

4.23 Protein Activation sub-model

[0518] The activity of mature protein monomers and complexes is not fixed, but rather can be modulated by small molecules, DNA, RNA, and other proteins, as well as by temperature and pH. Furthermore, cells often purposefully modulate protein activity to respond to external signals and maintain homeostasis.

[0519] Protein chemical regulation was reconstructed based on extensive review of the primary literature and several databases, with particular emphasis on antibiotics. The reconstructed protein chemical regulation network contains 16 metabolites, three Boolean-valued pseudometabolites or stimuli, and temperature which regulate 10 proteins including four transcription factors, Topoisomerases II and IV, the 30S and 50S ribosomal particles. The 10 putative chemically regulated proteins are critical for transcription, supercoiling, and translation. The effects of physical properties such as temperature and pH on protein activity were curated from BRENDA, BioCyc, and UniProt. The prosthetic groups and coenzymes required for maturation and catalysis were also curated.

[0520] Because the exact effects of small molecules, temperature, and pH on the functional activity of proteins are not well characterized, this process implements a Boolean model of their effects on the functional state – enzymatically competent or incompetent – of mature proteins. Less well characterized effects of other physical properties such as pH are not modeled in some embodiments. The functional state, I_i , of every copy of protein species i is governed by a single independent Boolean function, f_i , as shown in Equation 26.

$$I_i = f_i \left(\frac{m}{V}, s, T \right) \quad (26)$$

where m_j (an element of the vector m in Equation 26) is the concentration of metabolite j , and s_j (an element of the vector s in Equation 26) is the value of stimulus j , and T is the temperature. Proteins with no known chemical regulation were assumed to be constitutively competent.

[0521] The Protein Folding sub-model separately models the role of chaperones in protein folding and accounts for the small molecule prosthetic groups required for protein folding. Several processes including Metabolism model the coenzymes required for catalytic activity. After the temperature, cell volume, pseudometabolite values, and metabolite and total protein copy numbers are initialized, the competency state of each of the regulated proteins is initialized to the steady state of the regulatory network. Because the regulatory network is acyclic, the network converges in one iteration.

[0522] FIG. 44A is a flowchart that illustrates an example method 4400 for a Protein Activation sub-model, according to an embodiment. In FIG. 44A, the index b is used for proteins, so f_b corresponds to f_i in Equation 26. In step 4401 the sub-model sets parameters for protein activation: Boolean function f_b determining whether concentration of metabolites M_m (based on counts N_m of metabolites divided by volume v of cell), temperature T and other stimuli S_s are within ranges for a protein of species b to be competent.

[0523] In step 4411, the next protein species b is selected for activation, until all proteins that are not inherently competent have been selected and the method ends. When one is selected, then it is determined in step 4413, whether Equation 26 evaluates to true. If so, then

in step 4415 all copies of protein species b are marked competent (i.e., all activated → none inactivated). If not, then in step 4417 no copies of protein species b are marked competent (i.e., none activated → all inactivated). In either case, control returns to step 4411 to select the next protein species.

[0524] FIG. 44B is a list of instructions that illustrates an example Protein Activation sub-model, according to an embodiment for an *M. genitalium* example.

4.24 Protein Decay sub-model

[0525] Protein decay includes degradation, disassembly and refolding. Protein degradation serves two critically important functions. First, protein degradation eliminates abnormal and potentially toxic proteins including prematurely aborted polypeptides tagged with SsrA degradation signals, and salvages their valuable metabolic resources to support protein synthesis. Second, degradation controls protein expression, enabling cells to direct valuable amino acids away from ineffective or even harmful proteins toward productive proteins, regulate protein activity, and respond to external signals. The degradation rates of *E. coli* proteins are correlated with their N-terminal residue, suggesting that some cells target proteins for degradation by manipulating their N-termini. This relationship is known as the N-end rule.

[0526] Protein refolding is an efficient mechanism for eliminating abnormally folded proteins, requiring less energy than protein degradation and avoiding the large cost of re-synthesis. The kinetics and energetics of protein refolding are not well characterized. This process models the degradation of protein monomers, macromolecular complexes, cleaved signal sequences, and prematurely aborted polypeptides as well as the mis-folding and re-folding of protein monomers and complexes.

[0527] The half-life of each protein monomer was predicted using the N-end rule. The half-life of each macromolecular complex was set equal to the weighted mean half-life of its RNA and protein subunits. The half-lives of damaged and misfolded configurations of each protein species, cleaved signal sequences, and prematurely aborted polypeptides were set to zero to reflect their rapid degradation.

[0528] Under conditions of excess proteases, peptidases, and ATP, the Protein Decay sub-model simulates the configuration of each protein as a five-state system – aborted, nascent, mature, misfolded, and damaged – and models the degradation of protein *i* as Poisson process with rate parameter $k_{d,i} = \ln(2)$ divided by the half-life of the protein species, $\tau_{1/2,i}$, where $\tau_{1/2,i} = 20$ h for the nascent and mature states and zero for the aborted, misfolded, and damaged states. Under conditions of limited proteases, peptidases, or ATP, the sub-model stochastically degrades proteins as a function of the copy number and catalytic rate of each enzyme and the number of cytosolic ATP molecules. The sub-model assumes (1) cytosol-localized nascent, mature, misfolded, and damaged proteins are degraded by protease La and six peptidases, (2) membrane and extracellular-localized proteins are not vulnerable to degradation by the cytosol localized protease La, (3) aborted polypeptides are degraded by protease FtsH and six peptidases, (4) protein degradation is kinetically fast and therefore this sub-model doesn't represent intermediate degradation states, and (5) aborted, misfolded, and damaged proteins are immediately degraded if proteases are expressed and energy is available.

[0529] The Protein Decay sub-model simulates protein unfolding as a Poisson process with unfolding rate parameter k_u set to the nominal rate 10^{-6} per second per molecule. Cytosol-localized protein re-folding is modeled as a single chemical event catalyzed by the cytosol-localized chaperone ClpB and driven by the hydrolysis of a single ATP molecule. Because protein refolding is not well characterized, this process assumes that if ClpB is expressed and excess ATP is available, protein refolding proceeds to completion within the simulation time step. If ClpB is expressed, but ATP is not present in excess, the model stochastically re-folds proteins equal to the intracellular copy number of ATP. This model makes several simplifying assumptions. First, the model assumes that all protein species misfold and re-fold at the same rate. Second, the model assumes that the kinetics of protein misfolding and re-folding are fast and therefore intermediate mis-folded states can be ignored.

[0530] The Protein Monomer and Protein Complex states represent the copy number of each protein monomer and macromolecular complex including the signal sequence,

misfolded, and damaged copy numbers of each protein species and the copy number of each proteolytic enzyme in each of five compartments: cytosol, membrane, terminal organelle cytosol, terminal organelle membrane, and extracellular space. The Polypeptide state represents the amino acid sequence of each prematurely aborted polypeptide. The Chromosome, FtsZ Ring, Ribosome, and RNA Polymerase states represent the detailed configurations of DNA-bound proteins, the FtsZ septal ring, RNA polymerases, and ribosomes, respectively. These detailed configurations are updated when DNA-bound proteins, FtsZ, RNA polymerase, or ribosomes are degraded to reflect decreased enzyme copy numbers. The Transcript and Polypeptide states are also updated upon RNA polymerase or 70S ribosome degradation to reflect transcript and polypeptide abortion. The RNA state represents the damaged copy number of each RNA species. The Metabolite state represents the copy number of each intracellular metabolite.

[0531] The Protein Decay sub-model marks RNA subunits of degraded macromolecule complexes as “damaged” for immediate degradation by the RNA Decay process. Initially, misfolded and damaged configurations of each RNA and protein species are set with zero copy counts because the typical occupancies of these configurations are small compared to that of the mature configuration. Similarly, the cell is initialized with no prematurely aborted polypeptides. The RNA, Protein Monomer, and Protein Complex states initialize the total copy counts of each RNA and protein species. The expression of the proteases and peptidases was fit to provide sufficient enzymes to quickly degrade damaged proteins, or more specifically to prevent sustained accumulation of damaged proteins.

[0532] FIG. 45A through FIG. 45D are flowcharts that illustrate an example method 4500 for a Protein Decay sub-model, according to an embodiment. Here the index for target protein monomer species (gene product) is b , for target polypeptide (partial gene product) species is p , the index for enzyme (protein or complex) species is e , the index for target complex species is x , the index for target RNA species is r , the index for metabolites is m . In step 4501 parameter values are set, including values for: unfolding rates KUb , KUx ; kinetic rate Ke in units of amino acids released per second by enzyme e ; degradation rates of proteins and complexes $KDPb$ and $KDCx$, respectively, based on their half-lives;

stoichiometry Mmb of metabolite m in reaction b to degrade protein monomer; stoichiometry Mmx of metabolite m in reaction x to degrade complex; stoichiometry Rrx of RNA species r in reaction x to degrade complex; stoichiometry Pbx of protein monomer species b in reaction x to degrade complex; stoichiometry Mmp of metabolite m (including metabolites of ATP hydrolysis) in reaction p to degrade polypeptide.

[0533] In steps 4503 to 4509, mature proteins, which are folded, become randomly unfolded based on a Poisson distribution with rate = copy count of mature (folded) proteins of species b (NMb) multiplied by KU . In some embodiments, KU is KUb a function of the protein species (or some other factor or factors); but, in the illustrated embodiment, KU is a constant for gene product protein monomers. The count of unfolded proteins is designated NUb and is incremented by the random number pulled from the Poisson distribution.

[0534] In steps 4511 through 4517, complex species become randomly unfolded based on a Poisson distribution with rate = $NMx * KU$, where NMx is the copy count of mature (folded) complex species x . The count of unfolded complexes is designated NUx and is incremented by the random number pulled from the Poisson distribution.

[0535] In steps 4521 through 4535, unfolded proteins or complexes are randomly folded while folding resources are available (e.g., folding enzyme present and metabolites are sufficient for ATP hydrolysis, and unfolded proteins or complexes remain), as determined in step 4521 and 4535, respectively. Copy counts of affected state variables are updated. The selected unfolded protein or complex is based on a multinomial distribution with probability for each species based on the copy count of unfolded members of the species.

[0536] In steps 4541 through 4555, a random number of copies of a randomly selected complex species are broken down to their gene product parts, based on a Poisson distribution with occurrence rate = $Nx * KDCx$ —provided resources are sufficient, as determined in step 4521 and step 4551. Each RNA and protein subunit produced by degrading a complex is counted as damaged.

[0537] When the randomly selected number of complex copies have all been degraded to their damage gene product parts, then in steps 4561 through 4575, polypeptides are broken

down (degraded) to amino acids while resources allow (enough enzymes, metabolites and enzymatic capacity).

[0538] When all polypeptides have been degraded that can be in the current time step, then in steps 4581 through 4599, a random number of protein monomers are broken down in each species while resources allow (e.g., peptidases available, randomly selected enzymatic capacity remains, metabolites sufficient). (RNA parts of complexes are degraded in the RNA Decay sub-unit described above.)

[0539] The *M. genitalium* chromosome contains one protein chaperone—the cytosol-localized and ATP-dependent protease ClpB (MG355)—dedicated to protein refolding. The *M. genitalium* genome contains homologs of proteases FtsH (MG457) and La (MG239) and six peptidases: aminopeptidase (MG324), cytosol aminopeptidase (MG391), glycoprotease (MG208), metalloendopeptidase (MG046), oligoendopeptidase F (MG183), and proline iminopeptidase (MG020)182,634 . The membrane-localized protease FtsH is believed to cleave prematurely aborted polypeptides tagged with an N-terminal SsrA tag into approximately 15 amino long fragments by an ATP-dependent mechanism. The kinetics and energetics of FtsH proteolysis have been reported. The cytosol-localized protease La is believed to processively cleave most other proteins into fragments 10-20 amino acids in length by an ATP-dependent mechanism. The kinetics and energetics of La proteolysis have also been reported. Peptidases are believed to hydrolytically degrade the polypeptide fragments produced by peptidases FtsH and La. The specific function of each of the six *M. genitalium* peptidases is unknown.

[0540] FIG. 45E through FIG.45G are lists of instructions that illustrate an example Protein Decay sub-model, according to an embodiment for an *M. genitalium* example.

4.25 FtsZ polymerization sub-model

[0541] Cell division in many bacterial species requires the assembly of an FtsZ ring at the cell membrane around the midplane of the cell. FtsZ is a homologue of eukaryotic tubulin that assembles into long polymers. These polymers are typically localized to the center of the cell, forming a membrane-bound ring. FtsZ is a GTPase, and GTP hydrolysis to GDP causes

the FtsZ filaments to bend. This bending serves as one of the forces enabling cell division. GTP only hydrolyzes to GDP at an active site formed at the interface of two FtsZ molecules, meaning that FtsZ polymerization is required for GTPase activity. The FtsZ polymerization sub-model addresses the assembly FtsZ monomers into long polymers.

[0542] Typically, FtsZ polymers of variable length are able to bind the cell membrane to form contractile rings. The mean filament length in *E. coli* has been measured to be 100 nm (23 FtsZ monomers). The minimum fragment length observed in *C. crescentus* is 40 nm (9 FtsZ monomers). Being such a small organism, it is hypothesized that *M. genitalium* would have filaments on the lower end of what is sufficient for contractile force generation. For simplicity of calculating the Cytokinesis progress, the sub-model uses polymers of a fixed length, 9 subunits. This sub-model simulates polymerization of FtsZ-GTP monomers to 9mers.

[0543] The sub-model uses a differential equation model to assemble the polymers . During polymerization, FtsZ-GTP can exist in any intermediate state: 1mer, 2mer, ... 8mer, 9mer. The set of differential equations models the following reactions: Activation of FtsZ monomers(F) to FtsZ-GTP(FT); Deactivation of FtsZ-GTP(FT) to FtsZ(F) Exchange of FtsZ-GDP(FD) to FtsZ-GTP(FT); Reverse exchange of FtsZ-GTP(FT) to FtsZ-GDP(FD); Nucleation of two FtsZ-GTP molecules(FT) to form a dimer(FT₂); Reverse nucleation of a dimer(FT₂) into two FtsZ-GTPs(FT); Polymer elongation adding an FtsZ-GTP(FT) to an existing polymer of length 2-8 subunit s(FT₂ -FT₈); Reverse elongation removing an FtsZ-GTP(FT) from an existing polymer of length 3-9subunits(FT₃ -FT₉). FIG. 46 is a list of parameters and values for an FtsZ polymerization sub-model useful for modeling cell division, according to an embodiment.

[0544] The count of full-length activated FtsZ (9mer) filaments is used by the Cytokinesis sub-model for cell pinching. The FtsZ polymerization differential equations are run a number of times (up to 50 iterations) before the simulation starts to ensure starting at a steady state distribution of polymer lengths. Iterations are run until the change in the solution across consecutive iterations is less than a set tolerance (e.g., about 0.2% of the enzymatic counts).

[0545] FtsZ can exist in one of multiple states: inactivated monomer, activated monomer (GTPbound), nucleated (dimer of two activated FtsZ molecules), elongated polymer of three or more GTPbound FtsZ molecules. The hydrolyzed FtsZ polymers are only considered during cell membrane pinching in the Cytokinesis sub-model. The FtsZ molecules can move to states of higher and lower polymerization at known rates. FtsZ polymerization is modeled using a set of differential equations modified involving the activation, nucleation, and elongation of FtsZ polymers. The main modifications were that the equations were simplified to not include annealing and cyclization of FtsZ polymers. The following differential equation model given by equations 27 is evaluated at each time step.

$$\begin{aligned}
 \frac{dF}{dt} &= k_{act2}F_T - k_{act1}F \\
 \frac{dF_D}{dt} &= k_{act2}F_T [GDP] - k_{act1}F_D [GTP] \\
 \frac{dF_T}{dt} &= k_{act1}F - k_{act2}F_T + k_{act1}F_D [GTP] - k_{act2}F_T [GDP] - 2k_{nuc1}F_T^2 + \dots \\
 &\quad + \dots - 2k_{nuc2}F_{T2} - k_{el1}F_T \left(\sum_{i=2}^8 F_{Ti} \right) + k_{el2} \left(\sum_{i=3}^8 F_{Ti} \right) \\
 \frac{dF_{T2}}{dt} &= k_{nuc1}F_T^2 - k_{nuc2}F_{T2} - k_{el1}F_T F_{T2} + k_{el2}F_{T3} \\
 \frac{dF_{Ti}}{dt} &= k_{el1}F_T F_{Ti-1} - k_{el2}F_{Ti} - k_{el1}F_T F_{Ti} + k_{el2}F_{Ti+1}, \text{ for } i \in 3..8 \\
 \frac{dF_{T8}}{dt} &= k_{el1}F_T F_{T8} - k_{el2}F_{T8}
 \end{aligned} \tag{27}$$

Solving these equations results in a real-value distribution of monomers and filament lengths at each time step. The model framework involves the storage of the count of each molecular entity rather than the concentration. This sub-model discretizes the distribution at each time step for compatibility with the rest of the simulation.

4.26 Metabolism sub-model

[0546] To grow and replicate, cells transform external nutrients into cellular mass. The first step in this process is to import nutrients from the external environment and metabolize those nutrients primarily into macromolecule building blocks. Furthermore, cells must recycle and/or export byproducts. The Metabolism sub-model models the import of extracellular nutrients and their conversion into macromolecule building blocks.

[0547] *M. genitalium* is believed to have adapted to the rich environment provided by its host human urogenital epithelium by massive degenerative evolution from low G+C content Gram positive bacteria, eliminating non-essential genes involved in oxidative phosphorylation, ATP synthesis via the pentosephosphate pathway, and amino acid, nucleotide, lipid, and cofactor biosynthesis. Several studies have also shown that *M. genitalium* has evolved metabolic enzymes with relaxed substrate specificity. *M. genitalium* is generally cultured on SP-4 medium, a complex and undefined medium based on CMRL-1066.

[0548] *Extracellular Medium* The *in silico* medium was based on SP-4 medium with supplementary metabolites added to facilitate *in silico* growth. First, the chemical composition of SP-4 medium was estimated from the characterized composition of each medium component. Second, additional metabolites were added until the flux-balance analysis (FBA) metabolic model predicted non-zero growth. Finally, the concentrations of several metabolites were increased to support sustained cell growth.

[0549] *Cellular Composition* The *in silico* chemical composition of *M. genitalium* was based on its reconstructed cellular mass composition, and fit to match the 27 other modeled cellular processes.

[0550] *Energy* In addition to metabolic building blocks such as nucleic and amino acids, cellular growth and maintenance requires energy. The *in silico* *M. genitalium* growth-associated maintenance energy (GAM) was based on the characterized energetic requirements of *E. coli*.

[0551] *Reactions* FIG. 47A and FIG. 47B are block diagrams that illustrate part of a metabolic network of *M. genitalium*, according to an embodiment. The *M. genitalium* metabolic network was reconstructed with guidance from the *M. genitalium* flux-balance analysis (FBA) metabolic model. Briefly, the metabolic network was reconstructed based on homology to previously modeled organisms and an extensive review of the primary literature. In particular, the metabolic network was reconstructed to support the metabolic demands of all 27 other sub-models.

[0552] *Kinetics* Reaction kinetics were curated primarily from the proteomic databases SABIO-RK, BRENDA, and BioCyc. The maximum exchange rates of carbon and non-carbon-containing metabolites were set to 12 and 20 milliMoles (mmol) per grams dry cell weight (gDCW) per hour, comparable to recent FBA models of *M. genitalium* and *E. coli* metabolism.

[0553] The Metabolism sub-model models several critical cellular functions including (1) the uptake of external nutrients, (2) the synthesis of the metabolic building blocks and intermediate energy carriers, (3) lipid assembly, membrane insertion, and maturation, and (4) the recycling and export of the metabolic byproducts of other cellular processes. This process makes two key assumptions and uses FBA to model *M. genitalium* metabolism. In particular, this process calculates the flux, v , of each metabolic reaction. First, the sub-model assumes that the internal dynamics of the metabolome are fast compared to the time step of the *M. genitalium* simulation, or, equivalently, that the metabolic network can be considered to be at steady-state on the 1s simulation time scale. Second, the sub-model assumes that the *M. genitalium* metabolic network maximizes cellular metabolite production given the available extracellular nutrients and metabolic enzymes.

[0554] To be compatible with the 27 other sub-models, the Metabolism sub-model has four differences from most FBA metabolic models. First, the metabolic network, S , and the

cellular mass production pseudoreaction are expanded to produce all of the metabolites required by the 27 other sub-model, where S_{ij} is the stoichiometry of metabolite i in reaction j and S_{ib} is the stoichiometry of metabolite i in the cellular mass production pseudoreaction, b . Second, internal exchange reactions are added to recycle the metabolic byproducts of the other processes (e.g. recycle ADP and Pi to ATP), that is to exchange metabolites between the cytosol and the metabolic network. Third, the optimization objective, g , is expanded to include the construction of new cell mass as well as the recycling and export of the metabolic byproducts of the 27 other sub-models. The objective g represents the relative fitness gains of intracellular metabolite recycling and novel cell mass production, that is, g represents how heavily the metabolic network favors novel cell mass production over metabolite recycling. g was set assuming that the production/exchange of each molecule contributes equally to cellular fitness. Specifically g was set such that the production/exchange of each individual molecule is weighted equally. Fourth, the flux bounds, v_l are v_u , are functions of the cell dry mass, m , the copy count and maximum exchange rate of each extracellular metabolite, n_m and $k_{\pm ex}$, the thermodynamics, ΔE , of each reaction, and the copy number and maximum catalytic rate of each metabolic enzyme, n_e and $k_{\pm cat}$. Specifically, external exchange flux bounds are functions of the cell dry mass and the extracellular copy counts and maximum exchange rates of the exchanged metabolites; internal exchange flux bounds are functions of the intracellular copy numbers of the exchanged metabolites; transport and chemical reaction flux bounds are functions of the copy numbers and kinetics of the enzymes and the reaction thermodynamics. Fifth, because the average energy used by the 27 other sub-models is significantly less than the observed GAM, this process also represents the turnover of the otherwise unmodeled energy consumption. FIG. 47C and FIG. 47D are block diagrams that illustrate an integrated flux balance analysis for *M. genitalium*, according to an embodiment.

[0555] Equation set 28 summarizes the FBA metabolic model.

$$\begin{aligned}
v^* &= \underset{v}{\operatorname{argmax}} g^T v \text{ subject to} \\
Sv &= 0 \text{ and} \\
v_l &\leq v \leq v_u \text{ where} \\
g_i &= \begin{cases} \sum_{j \neq bm} S_{j,b} & \text{pseudoreaction } i = b \text{ represents the synthesis of biomass (metabolite } bm) \\ -1 & \text{reaction } i \text{ represents an internal exchange} \\ 0 & \text{otherwise} \end{cases} \\
v_l &= f_l(m, n_m, k_{-ex}, -\Delta E, n_e, k_{-cat}) \\
v_u &= f_u(m, n_m, k_{-ex}, \Delta E, n_e, k_{-cat})
\end{aligned} \tag{28}$$

[0556] *Integration* The Metabolism sub-model imports extracellular nutrients and converts them into the building blocks of macro-molecular synthesis. The Metabolite state represents both the extracellular and cellular copy numbers of each metabolite. The rates of extracellular nutrient exchange are functions of the total cell mass represented by the Mass state and the extracellular metabolite copy numbers. Cytosolic and membrane metabolite copy counts limit the rates of internal exchange/recycling. The rates of chemical and transport reactions are limited by the copy numbers and kinetics of metabolic enzymes synthesized by the protein synthesis and maturation pathway and represented by the Protein Monomer and Protein Complex states. The Metabolic Reaction state records the calculated flux of each metabolic reaction. The Simulation object allocates metabolites produced by the Metabolism sub-model to the other 27 sub-models to support several functions including DNA, RNA, and protein synthesis and maturation. Additionally, the other 27 sub-models generate byproducts which the Metabolism sub-model either recycles or exports from the cell.

[0557] *Initial Conditions* The Metabolite state initializes the copy count of each metabolite. The Protein Monomer state initializes the total copy count of each protein monomer. The Macromolecular Complexation sub-model initializes the copy count of each macromolecular complex. This process initializes the cellular growth rate and reaction fluxes

to the steady-state of the metabolic network using the same FBA simulation used at each time step.

[0558] *Fitting* The metabolic model was fit to match the observed *M. genitalium* mass doubling time, $\tau = 9$ h. Specifically, metabolic enzyme expression was fit using a modified version of minimization of metabolic adjustment. FIG. 47E and FIG. 47F are lists of instructions that illustrate an example Metabolism sub-model, according to an embodiment for the *M. genitalium* example.

4.27 Cytokinesis sub-model

[0559] Cytokinesis, the division of the cell into two separate daughter cells, is the final step in the cell cycle. The major step of cytokinesis is the pinching of the cell membrane in a certain location (typically the midline of the cell) until the membrane is separated and division is complete. Typically, a membrane-bound ring of a tubulin homologue forms at the cell membrane around the pinching site, and the localization and assembly of this ring requires a set of accessory proteins. Both contractile forces of the ring filaments and external forces by the accessory proteins contribute to the pinching of the cell membrane.

[0560] *M. genitalium* has the tubulin homologue, FtsZ (MG224), but does not contain the accessory proteins required for cell division in other bacterial species. Therefore, cytokinesis in *M. genitalium* is not fully understood. However, an “iterative pinching” has been presented by which a cell can divide using only FtsZ. The proposed mechanism invokes the formation of a contractile Z ring along the interior surface of the midline of the cell membrane. The ring is comprised of GTP-activated FtsZ polymer filaments that bend when their GTP is hydrolyzed. The bending draws the membrane-bound ends of the filaments together, thereby constricting the cell membrane. Many such pinching events result in a complete constriction of the cell. *M. genitalium* cytokinesis is thus a cycle of filament binding, bending, and dissociation. It has been suggested that *M. genitalium* cell division can occur in the absence of FtsZ through motility. The Cytokinesis sub-model uses the iterative pinching model because *M. genitalium* motility is not accessible, and FtsZ is known to be an essential gene in *M. genitalium*. FIG. 48A is a list of parameters and values for the Cytokinesis sub-model

useful for modeling cell division, according to an embodiment. All of the parameters used in the Cytokinesis sub-model are described in FIG. 48A.

[0561] *Parameter Assignment.* The kinetic rates of FtsZ filaments binding and dissociating from the membrane, *kbind* and *kunbind*, were uncharacterized. Therefore, the rates were set to be rapid such that their values under typical protein expression do not significantly limit the progression of cytokinesis.

[0562] FIG. 48B is a block diagram that illustrates an example cycle of filament bending during cytokinesis in the Cytokinesis sub-model, according to an embodiment. There are four main steps to the implemented iterative pinching model, as seen in FIG. 48B. First straight FtsZ-GTP filaments bind inside the cell membrane, such that they form an inscribed polygon (stage A). Next the GTP molecules in these filaments hydrolyze to GDP, causing the filaments to bend. Since the filaments are cell membrane bound, this bending pinches the membrane inwards. The filaments bend just enough to approximately form a new circle. The new circumference of the pinched region is now smaller, and equal to the perimeter of the inscribed polygon (stage C). The precise timing of the binding, hydrolysis, and dissociation of the contractile rings allows the sub-model to maintain the contraction progress across iterative pinching steps. After hydrolysis, one of the two bent filament rings is dissociated. The other remains, to maintain the progress of cell pinching (stage D). Now a new set of straight filaments can bind and form a new polygon that will be smaller than the previous (stage A). Upon binding, the residual ring of bent filaments can dissociate (stage B). The cycle repeats, resulting in smaller and smaller pinched circumferences, until the cell has divided.

[0563] Cytokinesis was implemented as a process that is only initiated upon chromosomal segregation. Next, the iterative pinching is modified such that two sets of filaments bind (stage A). Multiple filament sets are involved to maintain the progress of membrane constriction. Indeed, it was previously proposed that the FtsZ filaments are more abundant and that a new ring of filaments attaches to stabilize the membrane and maintain progress before the depolymerization event. Others report that the FtsZ ring is 3-9 filaments thick depending on the bacterial strain. Using the lower bound, since *M. genitalium* is a very small

bacterium, the sub-model assumes that the FtsZ maintains a thickness of up to 3 filaments. One embodiment of the sub-model involves two sets of straight FtsZ filaments to inscribe the diameter of the cell. Then up to one set of filaments can depolymerize while two new sets of straight filaments bind the diameter of the cell. Only once the two new sets of straight filaments have bound, can the second bent set of filaments depolymerize.

[0564] The output of the Cytokinesis sub-model is an indication of the progress of cell pinching, including the determination of when the cell has successfully split into two daughter cells. Complete cell division is also the trigger to end the entire simulation. FIG. 48C is a list of state variables and sub-models that interact with the Cytokinesis sub-model, according to an embodiment. Initialization steps are not required for this process, as Cytokinesis does not take place until late in the cell cycle. Initialization of FtsZ polymers is handled by the FtsZ Polymerization sub-model.

[0565] At each time step, the sub-model bends a pair of full length (9mer) FtsZ polymers along the circumference of the midplane of the cell. When these filaments hydrolyze and bend, circle and arc formulas are used to calculate the new smaller circumference of the cell. Depending on the state of the FtsZ ring, one or more of the following five steps is performed.

1. Straight filament binding (stage A): If cytokinesis has just initiated (triggered by Chromosome Segregation) or if the previous FtsZ ring has hydrolyzed and undergone dissociation, the sub-model attempts to bind two straight filaments at each polygon edge, forming a regular polygon that inscribes the pinched cell circumference.

(a) Determine the number of filaments to bind, based on the number of edges on the polygon that inscribes the pinched cell circumference. The number of edges (numEdges) is determined using the pinched diameter of the cell calculated in the previous time step (D_{prev}), and the length of the FtsZ filament (L) as given by Equation 29a.

$$\text{numEdges} = \left\lceil \left(\frac{\pi}{\sin^{-1} \left(\frac{L}{D_{prev}} \right)} \right) \right\rceil \quad (29a)$$

- (b) Bind straight filaments such that each edge has a random chance of being bound and the probability of binding each of two filaments is equal to k_{bind}
 - (c) Upon each binding event, update the state of the FtsZ ring.
2. Bent filament dissociation II (stage B): If there was a previous cycle and at least one straight filament has bound each polygon edge, then dissociate the bent filament from each edge.
- (a) Randomly dissociate bent FtsZ-GDP filaments such that the probability of dissociating each filament is equal to $k_{unbound}$.
 - (b) Dissociate unbound FtsZ-GDP filaments into FtsZ-GDP monomers.
 - (c) Upon each dissociation event, update the state of the FtsZ ring.
3. Filament bending (Schematic S6C): If all polygon edges have two filaments bound and all residual bent filaments have dissociated, bend the edges of the newly completed polygon. When the filaments bend, their length does not change, and the amount of bending is only sufficient to form a new circle. The new circumference is therefore equal to the old polygon's perimeter. Each fragment is now an arc. For simplicity, all of the GTP molecules in the pair of filaments at a particular edge hydrolyze at the same time. In this version of the model, all filaments are of a fixed length. When they are joined end-to-end to form a regular polygon, the polygon may not fully inscribe the entire circumference. This remaining portion of the circumference does not bend when the polygon does. It is preserved and accounted for in the next iteration.
- (a) Randomly determine whether to bend the filaments, such that the probability of bending each edge is k_{hyd} . Bending at an edge can only take place if there are sufficient water molecules available to hydrolyze all of the GTP molecules in the two filaments.
 - (b) Update the state of the the FtsZ ring and change the bent filament from FtsZ-GTP to FtsZ-GDP.
 - (c) If all the edges have been bent, calculate the new pinched diameter of the cell using Equation 29b.

$$\begin{aligned} \text{Diameter}_{\text{AfterHydrolysis}} &= \frac{\text{circumference}}{\pi} \\ &= \frac{\text{numEdges} \times L}{\pi} \end{aligned} \quad (29b)$$

Adjust this diameter to account for the polygon not perfectly inscribing the cell circumference by adding back the length of the old perimeter not accounted for by the inscribed polygon.

4. Bent filament dissociation I (stage D): If all the filaments have been bent, dissociate one bent filament from each polygon edge. The other ring must remain to preserve pinching progress by maintaining the new smaller circumference.

(a) Randomly dissociate bent FtsZ-GDP filaments such that the probability of dissociating each filament is equal to k_{unbind} .

(b) Dissociate unbound FtsZ-GDP filaments into FtsZ-GDP monomers.

(c) Upon each dissociation event, update the state of the FtsZ ring.

5. Termination: If the pinched diameter is smaller than the length of one filament, conclude cytokinesis.

4.28 Host interaction sub-model

[0566] Besides passive external conditions that affect but do not really respond to a cell, in some circumstances the cell affects its surroundings, either as one cell in a colony or tissue or as one (or a few) competing with a population of different cells or for example, a virus that responds to the cell's signaling, Such interactions are accommodated in the Host interaction sub-model.

[0567] In an example embodiment, a host of the parasite *M. genitalium* responds to the presence of that bacterium. *M. genitalium* is a common extracellular urogenital epithelial parasite. Although *M. genitalium* is estimated to colonize the urogenital tract of 3% of all women, most infections are asymptomatic, and most researchers regard *M. genitalium* as an "ideal parasite" which lives in harmony with its human host. Only a small fraction of *M. genitalium* infections manifest clinically as urethritis, cervicitis, or pelvic inflammatory

disease. Several lipoproteins (MG191, MG192, MG217, MG318, and MG386) are involved in host adhesion and activation of the host immune response. 3 additional lipoproteins -- MG149, MG309, and MG412 -- activate the host immune response by stimulating Toll-like (TLR) receptors 1, 2, and 6, which in turn activate the host transcriptional regulator NF- κ B. Additionally, it has been suggested that MG075 is immunoreactive and that *Mycoplasma* also stimulates the host immune response by secreting hydrogen peroxide and superoxide radicals which cause membrane oxidative damage and trigger inflammation.

[0568] Because the interaction between *M. genitalium* and its human host is not well understood, the Host Interactions sub-model implements a Boolean model of the effect of the *M. genitalium* terminal organelle and accessory lipoproteins on six properties of the host human urogenital epithelium: *Adherence* – *Mycoplasma* adheres to its host if its terminal organelle is properly formed and all of its adhesion proteins are expressed; *TLR 1, 2, and 6 activation* – Host TLR 1 is activated if the bacterium is adherent and MG149 or MG412 is expressed. TLR2 is activated if the bacterium is adherent and MG149, MG309, or MG412 is expressed. TLR 6 is activated if the bacterium is adherent and MG309 is expressed; *NF- κ B activation* – NF- κ B is activated if TLRs 2 and 1 or 6 are active; *Inflammatory response activation* – The host inflammatory response is activated if either NF- κ B is active or the bacterium is adherent and at least one of the MG075, MG149, MG309, or MG412 antigens is expressed.

[0569] The Host state uses six Boolean variables to represent the status of the human urogenital epithelium. The Protein Monomer state represents the copy number of each terminal organelle and accessory lipoprotein synthesized and matured by the protein maturation sub-models and translocated into the terminal organelle.

[0570] After the Protein Monomer and Terminal Organelle Assembly states initialize the copy number of each protein, the Host state is initialized to the steady-state of the Host Interaction Boolean model. Because the Boolean model is acyclic, the model converges in one iteration.

[0571] FIG. 49 is a list of instructions that illustrates an example Host Interaction sub-model, according to an embodiment. At each time step, this process evaluates the six Boolean rules to determine the current state of the host.

5. Validation and experimental procedures

5.1 Validation

[0572] The model was validated against a broad range of independent datasets that were not used to construct the model and which encompass multiple biological functions – metabolomics, transcriptomics, and proteomics – and scales, from single cells to populations. FIG. 50 is a graph that shows example comparisons between measured and predicted growth rates for wild-type and 12 single-gene disrupted strains, according to an embodiment. Model predictions that fall within the shaded region were considered consistent with experimental observations; the region has a width of four times the SD of the wild-type strain growth measurement. Horizontal and vertical bars indicate predicted and observed SD. This plot shows positive results for whole-cell modeling.

[0573] Furthermore, in agreement with earlier reports (Yus et al., 2009), the model predicts that the flux through glycolysis is >100-fold more than that through the pentose phosphate and lipid biosynthesis pathways. Furthermore, the predicted metabolite concentrations are within an order of magnitude of concentrations measured in *Escherichia coli* for 100% of the metabolites in one compilation of data (Sundararaj et al., 2004) and for 70% in a more recent high-throughput study (Bennett et al., 2009; Figure 2F). Applicant's model also predicts "burst-like" protein synthesis due to the local effect of intermittent mRNA expression and the global effect of stochastic protein degradation on the availability of free amino acids for translation, comparable to recent reports by Yu et al., 2006 and So et al., 2011. The mRNA and protein level distributions predicted by Applicant's model are also consistent with recently reported single-cell measurements. Taking all of these specific tests of the model's predictions together, it is concluded that the whole-cell model presented here recapitulates experimental data across multiple biological functions and scales as well as presents a comparison to gene knockout phenotypes.

5.2 Experimental procedures

[0574] Media Composition. Each liter of SP-4 is comprised of broth base 600ml (Mycoplasma Broth Base 3.5g (BD 211458), Bacto Tryptone 10g (BD 211705), BactoPeptone 5.3g (BD 211677), distilled water 598ml, pH 7.5 by KOH and autoclaved), 20% glucose 25ml (CalBioChem 346351), CMRL 1066 10X 50ml (ATCC 20-2207), 7.5% sodium bicarbonate 5ml (EMDSX-0320-1), 200mM L-glutamine 5ml, 2% yeast extract solution 35ml (BD 210933), 2% autoclaved TC Yeastolate 100 ml (BD 255772), fetal bovine serum albumin (heat inactivated at 55°C for 2 hours) 170ml (Gibco 26140-079), penicillin G (100,000 U ml⁻¹) 2.5 ml (Sigma P7794), and 0.5% phenol red 4ml (Sigma P0290).

[0575] Frozen Stocks. Cells were harvested for storage as a frozen stock when the media in the 10 cm petri dish cultures was yellow (pH 6.3-6.7). The media from the 10 cm plate cultures was aspirated. Cells were collected by scraping the bottom of the plates, resuspended in 3 ml of FBS, and serially filtered through 1.2, 0.8, 0.45, and 0.2 µm polyethersulfone filters to sterilize and de-clump the cells. Stocks were stored at -80°C.

[0576] Colorimetric Growth Assay Serial Dilutions. Cells were harvested for serial dilutions when the media in the 10 cm petri dish culture was yellow (pH 6.3-6.7), and were harvested as described above for frozen stocks. The filtered suspension was used to make a serial dilution plate. The initial solution in the serial dilution plate is comprised of 50 µl of culture in 450 µl of SP-4. (For strains that were especially difficult to culture, 100 µl culture was added to 400 µl SP-4.) From this concentration were made three 5-fold serial dilutions. Dilutions were performed in triplicate for each culture, and 200 µl of each diluted sample was plated in a 96-well plate. The 96-well plates were stored at 37°C and 5% CO₂. Optical density readings were taken at 550 nm twice a day and used to make growth curves.

[0577] Colorimetric Growth Assay Calculations. The growth rate constant and doubling time for each strain was calculated using the growth curves of serially diluted bacterial cultures measured by the colorimetric assay and the serial dilution factor.

[0578] Quantitative PCR to Measure Cell Growth. Some strains grew so slowly that the colorimetric assay was inadequate to determine a growth rate. In these cases, a DNA

quantification method was used to estimate the growth rate. This method calculates the number of chromosomes in a sample, and assumes that changes in chromosome count are correlated with changes in the *M. genitalium* cell count. The strains were cultured in 4mL SP-4 and incubated at 37°C with 5% CO₂. Three replicate cultures were made for each of 14 days of harvest, resulting in 42 cultures per strain. Each day, cells were harvested by scraping the bottom of the plate, spun down at 4575 xg for 15 minutes, and resuspended in 20-400 µl of TE with 1% SDS. Samples were run on a 0.8% electrophoresis gel, and the DNA band was quantified using a Typhoon scanner and ImageQuant. The slope of the exponential region of the resulting growth curves was used to calculate a growth rate constant and doubling time (see Tab. S30). Comparing the relative growth of wild-type and the tkt gene deletion strain, we see that the results of the PCR method match those of the colorimetric assay.

6. Hardware Overview

[0579] FIG. 51 is a block diagram that illustrates a computer system 5100 upon which an embodiment of the invention may be implemented. Computer system 5100 includes a communication mechanism such as a bus 5110 for passing information between other internal and external components of the computer system 5100. Information is represented as physical signals of a measurable phenomenon, typically electric voltages, but including, in other embodiments, such phenomena as magnetic, electromagnetic, pressure, chemical, molecular atomic and quantum interactions. For example, north and south magnetic fields, or a zero and non-zero electric voltage, represent two states (0, 1) of a binary digit (bit).). Other phenomena can represent digits of a higher base. A superposition of multiple simultaneous quantum states before measurement represents a quantum bit (qubit). A sequence of one or more digits constitutes digital data that is used to represent a number or code for a character. In some embodiments, information called analog data is represented by a near continuum of measurable values within a particular range. Computer system 5100, or a portion thereof, constitutes a means for performing one or more steps of one or more methods described herein.

[0580] A sequence of binary digits constitutes digital data that is used to represent a number or code for a character. A bus 5110 includes many parallel conductors of information so that information is transferred quickly among devices coupled to the bus 5110. One or more processors 5102 for processing information are coupled with the bus 5110. A processor 5102 performs a set of operations on information. The set of operations include bringing information in from the bus 5110 and placing information on the bus 5110. The set of operations also typically include comparing two or more units of information, shifting positions of units of information, and combining two or more units of information, such as by addition or multiplication. A sequence of operations to be executed by the processor 5102 constitutes computer instructions.

[0581] Computer system 5100 also includes a memory 5104 coupled to bus 5110. The memory 5104, such as a random access memory (RAM) or other dynamic storage device, stores information including computer instructions. Dynamic memory allows information stored therein to be changed by the computer system 5100. RAM allows a unit of information stored at a location called a memory address to be stored and retrieved independently of information at neighboring addresses. The memory 5104 is also used by the processor 5102 to store temporary values during execution of computer instructions. The computer system 5100 also includes a read only memory (ROM) 5106 or other static storage device coupled to the bus 5110 for storing static information, including instructions, that is not changed by the computer system 5100. Also coupled to bus 5110 is a non-volatile (persistent) storage device 5108, such as a magnetic disk or optical disk, for storing information, including instructions, that persists even when the computer system 5100 is turned off or otherwise loses power.

[0582] Information, including instructions, is provided to the bus 5110 for use by the processor from an external input device 5112, such as a keyboard containing alphanumeric keys operated by a human user, or a sensor. A sensor detects conditions in its vicinity and transforms those detections into signals compatible with the signals used to represent information in computer system 5100. Other external devices coupled to bus 5110, used primarily for interacting with humans, include a display device 5114, such as a cathode ray

tube (CRT) or a liquid crystal display (LCD), for presenting images, and a pointing device 5116, such as a mouse or a trackball or cursor direction keys, for controlling a position of a small cursor image presented on the display 5114 and issuing commands associated with graphical elements presented on the display 5114.

[0583] In the illustrated embodiment, special purpose hardware, such as an application specific integrated circuit (IC) 5120, is coupled to bus 5110. The special purpose hardware is configured to perform operations not performed by processor 5102 quickly enough for special purposes. Examples of application specific ICs include graphics accelerator cards for generating images for display 5114, cryptographic boards for encrypting and decrypting messages sent over a network, speech recognition, and interfaces to special external devices, such as robotic arms and medical scanning equipment that repeatedly perform some complex sequence of operations that are more efficiently implemented in hardware.

[0584] Computer system 5100 also includes one or more instances of a communications interface 5170 coupled to bus 5110. Communication interface 5170 provides a two-way communication coupling to a variety of external devices that operate with their own processors, such as printers, scanners and external disks. In general the coupling is with a network link 5178 that is connected to a local network 5180 to which a variety of external devices with their own processors are connected. For example, communication interface 5170 may be a parallel port or a serial port or a universal serial bus (USB) port on a personal computer. In some embodiments, communications interface 5170 is an integrated services digital network (ISDN) card or a digital subscriber line (DSL) card or a telephone modem that provides an information communication connection to a corresponding type of telephone line. In some embodiments, a communication interface 5170 is a cable modem that converts signals on bus 5110 into signals for a communication connection over a coaxial cable or into optical signals for a communication connection over a fiber optic cable. As another example, communications interface 5170 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN, such as Ethernet. Wireless links may also be implemented. Carrier waves, such as acoustic waves and electromagnetic waves, including radio, optical and infrared waves travel through space without wires or cables.

Signals include man-made variations in amplitude, frequency, phase, polarization or other physical properties of carrier waves. For wireless links, the communications interface 5170 sends and receives electrical, acoustic or electromagnetic signals, including infrared and optical signals, that carry information streams, such as digital data.

[0585] The term computer-readable medium is used herein to refer to any medium that participates in providing information to processor 5102, including instructions for execution. Such a medium may take many forms, including, but not limited to, non-volatile media, volatile media and transmission media. Non-volatile media include, for example, optical or magnetic disks, such as storage device 5108. Volatile media include, for example, dynamic memory 5104. Transmission media include, for example, coaxial cables, copper wire, fiber optic cables, and waves that travel through space without wires or cables, such as acoustic waves and electromagnetic waves, including radio, optical and infrared waves. The term computer-readable storage medium is used herein to refer to any medium that participates in providing information to processor 5102, except for transmission media.

[0586] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, a hard disk, a magnetic tape, or any other magnetic medium, a compact disk ROM (CD-ROM), a digital video disk (DVD) or any other optical medium, punch cards, paper tape, or any other physical medium with patterns of holes, a RAM, a programmable ROM (PROM), an erasable PROM (EPROM), a FLASH-EPROM, or any other memory chip or cartridge, a carrier wave, or any other medium from which a computer can read. The term non-transitory computer-readable storage medium is used herein to refer to any medium that participates in providing information to processor 5102, except for carrier waves and other signals.

[0587] Logic encoded in one or more tangible media includes one or both of processor instructions on a computer-readable storage media and special purpose hardware, such as ASIC 5120.

[0588] Network link 5178 typically provides information communication through one or more networks to other devices that use or process the information. For example, network link 5178 may provide a connection through local network 5180 to a host computer 5182 or

to equipment 5184 operated by an Internet Service Provider (ISP). ISP equipment 5184 in turn provides data communication services through the public, world-wide packet-switching communication network of networks now commonly referred to as the Internet 5190. A computer called a server 5192 connected to the Internet provides a service in response to information received over the Internet. For example, server 5192 provides information representing video data for presentation at display 5114.

[0589] The invention is related to the use of computer system 5100 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 5100 in response to processor 5102 executing one or more sequences of one or more instructions contained in memory 5104. Such instructions, also called software and program code, may be read into memory 5104 from another computer-readable medium such as storage device 5108. Execution of the sequences of instructions contained in memory 5104 causes processor 5102 to perform the method steps described herein. In alternative embodiments, hardware, such as application specific integrated circuit 5120, may be used in place of or in combination with software to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware and software.

[0590] The signals transmitted over network link 5178 and other networks through communications interface 5170, carry information to and from computer system 5100. Computer system 5100 can send and receive information, including program code, through the networks 5180, 5190 among others, through network link 5178 and communications interface 5170. In an example using the Internet 5190, a server 5192 transmits program code for a particular application, requested by a message sent from computer 5100, through Internet 5190, ISP equipment 5184, local network 5180 and communications interface 5170. The received code may be executed by processor 5102 as it is received, or may be stored in storage device 5108 or other non-volatile storage for later execution, or both. In this manner, computer system 5100 may obtain application program code in the form of a signal on a carrier wave.

[0591] Various forms of computer readable media may be involved in carrying one or more sequence of instructions or data or both to processor 5102 for execution. For example, instructions and data may initially be carried on a magnetic disk of a remote computer such as host 5182. The remote computer loads the instructions and data into its dynamic memory and sends the instructions and data over a telephone line using a modem. A modem local to the computer system 5100 receives the instructions and data on a telephone line and uses an infra-red transmitter to convert the instructions and data to a signal on an infra-red carrier wave serving as the network link 5178. An infrared detector serving as communications interface 5170 receives the instructions and data carried in the infrared signal and places information representing the instructions and data onto bus 5110. Bus 5110 carries the information to memory 5104 from which processor 5102 retrieves and executes the instructions using some of the data sent with the instructions. The instructions and data received in memory 5104 may optionally be stored on storage device 5108, either before or after execution by the processor 5102.

7. Extensions, modifications and alternatives.

[0592] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. Throughout this specification and the claims, unless the context requires otherwise, the word “comprise” and its variations, such as “comprises” and “comprising,” will be understood to imply the inclusion of a stated item, element or step or group of items, elements or steps but not the exclusion of any other item, element or step or group of items, elements or steps. Furthermore, the indefinite article “a” or “an” is meant to indicate one or more of the item, element or step modified by the article.

8. References

- [0593] Atlas, J.C., Nikolaev, E.V., Browning, S.T., and Shuler, M.L. (2008). Incorporating genome-wide DNA sequence information into a dynamic whole-cell model of *Escherichia coli*: application to DNA replication. *IET Syst. Biol.* 2, 369-382.
- Bennett, B.D., Kimball, E.H., Gao, M., Osterhout, R., Van Dien, S.J., and Rabinowitz, J.D. (2009). Absolute metabolite concentrations and implied enzyme active site occupancy in *Escherichia coli*. *Nat. Chem. Biol.* 5, 593-599.
- Bratton, B.P., Mooney, R.A., and Weisshaar, J.C. (2011). Spatial Distribution and Diffusive Motion of RNA Polymerase in Live *Escherichia coli*. *J. Bacteriol.* 193, 5138-5146.
- Brenner, S. (2010). Sequences and consequences. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 365, 207-12.
- Browning, S.T., Castellanos, M., and Shuler, M.L. (2004). Robust control of initiation of prokaryotic chromosome replication: essential considerations for a minimal cell. *Biotechnol. Bioeng.* 88, 575-584.
- Chandrasekaran, S., and Price, N.D. (2010). Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and *Mycobacterium tuberculosis*. *Proc. Natl. Acad. Sci. U S A.* 107, 17845-50.
- Covert, M.W., Schilling, C.H., and Palsson, B.O. (2001). Regulation of gene expression in flux balance models of metabolism. *J. Theor. Biol.* 213, 73-88.
- Covert, M.W., Knight, E.M., Reed, J.L., Herrgard, M.J., and Palsson, B.O. (2004). Integrating high-throughput and computational data elucidates bacterial networks. *Nature* 429, 92-96.
- Covert, M.W., Xiao, N., Chen, T.J., and Karr, J.R. (2008). Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. *Bioinformatics* 24, 2044-2050.
- Castellanos, M., Kushiro, K., Lai, S.K., and Shuler, M.L. (2007). A genomically/chemically complete module for synthesis of lipid membrane in a minimal cell. *Biotechnol. Bioeng.* 97, 397-409.
- Castellanos, M., Wilson, D.B., and Shuler, M.L. (2004). A modular minimal cell model:

- purine and pyrimidine transport and metabolism. *Proc. Natl. Acad. Sci. USA* *101*, 6681-6686.
- Cordwell, S.J., Basseal, D.J., Pollack, J.D., and Humphery-Smith, I. (1997). Malate/lactate dehydrogenase in mollicutes: evidence for a multienzyme protein. *Gene* *195*, 113-120.
- Davidson, E.H., Rast, J.P., Oliveri, P., Ransick, A., Calestani, C., Yuh, C.H., Minokawa, T., Amore, G., Hinman, V., Arenas-Mena, C., et al. (2002). A genomic regulatory network for development. *Science* *295*, 1669-1678.
- de Kok, A., Hengeveld, A.F., Martin, A., and Westphal, A.H. (1998). The pyruvate dehydrogenase multi-enzyme complex from Gram-negative bacteria. *Biochim. Biophys. Acta* *1385*, 353-366.
- Di Ventura, B., Lemerle, C., Michalodimitrakis, K. and Serrano, L. (2006). From *in vivo* to *in silico* biology and back. *Nature* *443*, 527-33.
- Domach, M.M., Leung, S.K., Cahn, R.E., Cocks, G.G., and Shuler, M.L. (2004). Computer model for glucose-limited growth of a single cell of *Escherichia coli* B/r-A. *Biotechnol. Bioeng.* *26*, 1140.
- Fraser, C.M., Gocayne, J.D., White, O., Adams, M.D., Clayton, R.A., Fleishmann, R.D., Bult, C.J., Kerlavage, A.R., Sutton, G., Kelley, J.M., et al. (1995). The minimal gene complement of *Mycoplasma genitalium*. *Science* *270*, 397-403.
- Gibson, D.G., Benders, G.A., Andrews-Pfannkoch, C., Denisova, E.A., Baden-Tillson, H., Zaveri, J., Stockwell, T.B., Brownley, A., Thomas, D.W., Algire, M.A., et al. (2008). Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome. *Science* *319*, 1215-1220.
- Gibson, D.G., Glass, J.I., Lartigue, C., Noskov, V.N., Chuang, R.Y., Algire, M.A., Benders, G.A., Montague, M.G., Ma, L., Moodie, M.M., et al. (2010). Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* *329*, 52-56.
- Glass, J.I., Assad-Garcia, N., Alperovich, N., Yooseph, S., Lewis, M.R., Maruf, M., Hutchison C.A. 3rd, Smith, H.O., and Venter, J.C. (2006). Essential genes of a

- minimal bacterium. *Proc. Natl. Acad. Sci. USA* 103, 425-430.
- Güell, M., van Noort, V., Yus, E., Chen, W.H., Leigh-Bell, J., Michaelodimitrakis, K., Yamada, T., Arumugam, M., Doerks, T., Kuhner, S., et al. (2009). Transcriptome complexity in a genome-reduced bacterium. *Science* 326, 1268-1271.
- Hanawalt, P.C., Donahue, B.A., and Sweder, K.S. (2004). Repair and Transcription: Collision or collusion? *Curr. Biol.* 4, 518-521.
- Harada, Y., Funatsu, T., Murakami, K., Nonoyama, Y., Ishihama, A., and Yanagida, T. (1999). Single-molecule imaging of RNA polymerase-DNA interactions in real time. *Biophys. J.* 76, 709-715.
- Lartigue, C., Glass, J.I., Alperovich, N., Pieper, R., Parmar, P.P., Hutchison C.A. 3rd, Smith, H.O., and Venter, J.C. (2007). Genome transplantation in bacteria: changing one species to another. *Science* 317, 632-638.
- Lartigue, C., Vashee, S., Algire, M.A., Chuang, R.Y., Benders, G.A., Ma, L., Noskov, V.N., Denisova, E.A., Gibson, D.G., Assad-Garcia, N., et al. (2009). Creating bacterial strains from genomes that have been cloned and engineered in yeast. *Science* 325, 1693-1696.
- Kitano, H. (2002). Systems biology: a brief overview. *Science* 295, 1662-4.
- Kuhner, S., van Noort, V., Betts, M.J., Leo-Macias, A., Batisse, C., Rode, M., Yamada, T., Maier, T., Bader, S., Beltran-Alvarez, P., et al. (2009). Proteome organization in a genome-reduced bacterium. *Science* 326, 1235-1240.
- Morowitz H.J., Tourtellotte M.E., Guild W.R., Castro E., and Woese C. (1962). The chemical composition and submicroscopic morphology of *Mycoplasma gallisepticum*, Avian PPLO 5969. *J Mol. Biol.* 4, 93-103.
- Morowitz H.J. (1992). *Beginnings of cellular life: metabolism recapitulates biogenesis* (New Haven and London: Yale University Press).
- Orth, J.D., Thiele, I., and Palsson, B.O. (2010). What is flux balance analysis? *Nat. Biotechnol.* 28, 245-248.
- Orth, J.D., Conrad, T.M., Na, J., Lerman, J.A., Nam, H., Feist, A.M., and Palsson, B.O.

- (2011). A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism--2011. *Mol. Syst. Biol.* 7, 535.
- Pollack, J.D., Myers, M.A., Dandekar, T., and Herrmann, R. (2002). Suspected utility of enzymes with multiple activities in the small genome *Mycoplasma* species: the replacement of the missing "household" nucleoside diphosphate kinase gene and activity by glycolytic kinases. *OMICS* 6, 247-258.
- Pomerantz, R.T., and O'Donnell, M. (2010). Direct Restart of a Replication Fork Stalled by a Head-On RNA Polymerase. *Science* 327, 590-592.
- Russell, J.B., and Cook, G.M. (1995). Energetics of bacterial growth: balance of anabolic and catabolic reactions. *Microbiol. Rev.* 59, 48-62.
- Saeki, H. and Svejstrup, J.Q. (2009). Stability, Flexibility, and Dynamic Interactions of Colliding RNA Polymerase II Elongation Complexes. *Mol. Cell* 35, 191-205.
- Schmidt, H.L., Stocklein, W., Danzer, J., Kirch, P., and Limbach, B. (1986). Isolation and properties of an H₂O-forming NADH oxidase from *Streptococcus faecalis*. *Eur. J. Biochem.* 156, 149.
- So, L.H., Ghosh, A., Zong, C., Sepulveda, L.A., Segev, R., and Golding, I. (2011). General properties of transcriptional time series in *Escherichia coli*. *Nat. Genet.* 43, 554-560.
- Sundararaj, S., Guo, A., Habibi-Nazhad, B., Rouani, M., Stothard, P., Ellison, M., and Wishart, D.S. (2004). The CyberCell Database (CCDB): a comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of *Escherichia coli*. *Nucleic Acids Res.* 32, D293-295.
- Suthers, P.F., Dasika, M.S., Kumar, V.S., Denisov, G., Glass, J.I., and Maranas, C.D. (2009). A genome-scale metabolic model of *Mycoplasma genitalium*, iPS189. *PLoS Comput. Bio.* 5, e1000285.
- Taniguchi, Y., Choi, P.J., Li, G.W., Chen, H., Babu, M., Hearn, J., Emili, A., and Xie, X.S. (2010). Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science* 329, 533-538.
- Thiele, I., Jamshidi, N., Fleming, R.M., and Palsson, B.O. (2009). Genome-scale reconstruction of *Escherichia coli*'s transcriptional and translational machinery: a

- knowledge base, its mathematical formulation, and its functional characterization. *PLoS Comput. Biol.* 5, e1000312.
- Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T.S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J.C., et al. (1999). E-CELL: software environment for whole-cell simulation. *Bioinformatics* 15, 72-84.
- Vora, T., Hottes, A.K., and Tavazoie, S. (2009). Protein occupancy landscape of a bacterial genome. *Mol. Cell* 35, 247-253.
- Yu, J., Xiao, J., Ren, X., Lao, K. and Xie, X.S. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science* 311, 1600-1603.
- Yus, E., Maier, T., Michalodimitrakis, K., van Noort, V., Yamada, T., Chen, W.H., Wodke, J.A., Guell, M., Martinez, S., Bourgeois, R., et al. (2009). Impact of genome reduction on bacterial metabolism and its regulation. *Science* 326, 1263-1268.
-

CLAIMS

What is claimed is:

- 1 1. A method comprising:
 - 2 retrieving on a processor cell state data that indicates chromosome data that represents
 - 3 over 40 percent of genes in a single cell of an organism, and which also indicates
 - 4 populations of gene products in multiple functional compartments, wherein the
 - 5 functional compartments include at least an external conditions compartment, a
 - 6 membrane compartment, a cytosol compartment, and a DNA interactions
 - 7 compartment;
 - 8 executing on a processor a plurality of sub-models that each simulate, for a same time
 - 9 duration called a time step, a different cell process selected from a group of cell
 - 10 processes comprising at least transcription, translation, ribosome assembly, protein
 - 11 translocation, and metabolism;
 - 12 updating on a processor values for the cell state data based on results from executing the
 - 13 plurality of sub-models; and
 - 14 repeating for a plurality of time steps, the steps of retrieving cell state data, executing the
 - 15 plurality of sub-models and updating the values for the cell state data; and
 - 16 storing on a computer-readable medium values of the cell state data after the plurality of
 - 17 time steps.
- 1 2. A method as recited in claim 1 wherein the chromosome data represents all genes in a
- 2 single cell of an organism.
- 1 3. A method as recited in claim 1 wherein the cell state data includes data that represents in
- 2 the DNA interactions compartment populations of transcript species and RNA species.

1 4. A method as recited in claim 1 wherein the cell state data includes data that represents in
2 the cytosol compartment populations of: ribosomes; polypeptides connected to ribosomes;
3 RNA polymerase; protein monomer species; protein complex species; and metabolite
4 species.

1 5. A method as recited in claim 1 wherein the cell state data includes data that represents in
2 the membrane compartment populations of protein monomer species; protein complex
3 species; and metabolite species.

1 6. A method as recited in claim 1 wherein the cell state data includes data that represents in
2 the external conditions compartment populations of protein monomer species; protein
3 complex species; and metabolite species.

1 7. A method as recited in claim 1 wherein the cell state data includes data that represents cell
2 geometry, cell mass, elapsed time, external stimuli.

1 8. A method as recited in claim 1 wherein the cell state data includes data that represents
2 state of a host organism with which the cell resides.

1 9. A method as recited in claim 1 wherein the group of cell processes further comprises at
2 least one of chromosome replication initiation, chromosome replication or segregation in the
3 DNA interactions compartment and protein polymerization for cell division or cytokinesis in
4 the cytosol compartment.

1 10. A method as recited in claim 1 wherein the group of cell processes further comprises at
2 least one of chromosome condensation, chromosome supercoiling, chromosome damage, or
3 chromosome repair in the DNA interactions compartment.

1 11. A method as recited in claim 1 wherein the group of cell processes further comprises at
2 least one of transcriptional regulation, RNA processing, RNA modification, RNA
3 aminoacylation, or RNA decay in the cytosol compartment.

1 12. A method as recited in claim 1 wherein the group of cell processes further comprises at
2 least one of protein processing, protein folding, protein covalent modification, protein
3 activation, assembly of protein complexes, or protein decay in the cytosol compartment.

1 13. A method as recited in claim 1 wherein the group of cell processes further comprises at
2 least one of further protein processing for lipoprotein, protein folding, protein covalent
3 modification, protein activation, assembly of protein complexes, or protein decay in the
4 membrane compartment.

1 14. A method as recited in claim 1 wherein the group of cell processes further comprises
2 protein folding, protein covalent modification, protein activation, assembly of protein
3 complexes, further protein processing for lipoprotein, or protein decay in each of one or more
4 special organelles.

1 15. A method as recited in claim 1 wherein the group of cell processes further comprises
2 interactions with a host organism.

1 16. A method as recited in claim 1, further comprising determining initial values for the cell
2 state data before first executing the plurality of sub-models.

1 17. A method as recited in claim 1, further comprising dividing a value for cell state data that
2 represents a cell resource shared by multiple processes among the multiple processes based
3 on demand by the processes.

1 18. A method as recited in claim 1, further comprising determining a first cell phenotype,
2 such as cell mass doubling time or cell division time, based on the values of the cell state
3 data after the plurality of time steps.

- 1 19. A method as recited in claim 18, further comprising:
2 determining whether the first cell phenotype matches a target phenotype;
3 if the first cell phenotype does not match a target phenotype, then receiving data that
4 indicates a change to one or more genes in the chromosome data and changing the
5 chromosome data based on the data that indicates the change to one or more genes;
6 repeating, for a second plurality of time steps, the steps of executing the plurality of sub-
7 models and updating the values for the cell state data; and
8 determining a second cell phenotype based on the values of the cell state data after the
9 second plurality of time steps.
- 1 20. A method as recited in claim 18, further comprising:
2 determining whether the first cell phenotype matches a target phenotype;
3 if the first cell phenotype does not match the target phenotype, then genetically
4 modifying an actual cell of the organism to correspond to the chromosome data.

1 21. An apparatus comprising:

2 at least one processor; and

3 at least one memory including one or more sequences of instructions,

4 the at least one memory and the one or more sequences of instructions configured to, with

5 the at least one processor, cause the apparatus to perform at least the following:

6 retrieving cell state data that indicates chromosome data that represents over 40

7 percent of genes in a single cell of an organism, and which also indicates

8 populations of gene products in multiple functional compartments, wherein the

9 functional compartments include at least an external conditions compartment, a

10 membrane compartment, a cytosol compartment, and a DNA interactions

11 compartment;

12 executing a plurality of sub-models that each simulate, for a same time duration

13 called a time step, a different cell process selected from a group of cell processes

14 comprising at least transcription, translation, ribosome assembly, protein

15 translocation, and metabolism;

16 updating values for the cell state data based on results from executing the plurality of

17 sub-models; and

18 repeating for a plurality of time steps, the steps of retrieving cell state data, executing

19 the plurality of sub-models and updating the values for the cell state data; and

20 storing values of the cell state data after the plurality of time step.

1 22. A system comprising:
2 at least one processor; and
3 at least one memory including one or more sequences of instructions,
4 the at least one memory and the one or more sequences of instructions configured to, with
5 the at least one processor, cause at least one apparatus to perform at least the
6 following,
7 retrieving cell state data that indicates chromosome data that represents over 40
8 percent of genes in a single cell of an organism, and which also indicates
9 populations of gene products in multiple functional compartments, wherein the
10 functional compartments include at least an external conditions compartment, a
11 membrane compartment, a cytosol compartment, and a DNA interactions
12 compartment;
13 executing a plurality of sub-models that each simulate, for a same time duration
14 called a time step, a different cell process selected from a group of cell processes
15 comprising at least transcription, translation, ribosome assembly, protein
16 translocation, and metabolism;
17 updating values for the cell state data based on results from executing the plurality of
18 sub-models; and
19 repeating for a plurality of time steps, the steps of retrieving cell state data, executing
20 the plurality of sub-models and updating the values for the cell state data; and
21 storing values of the cell state data after the plurality of time step.

1 23. A non-transitory computer-readable medium carrying one or more sequences of
2 instructions, wherein execution of the one or more sequences of instructions by one or more
3 processors causes an apparatus to perform the steps of:
4 retrieving cell state data that indicates chromosome data that represents over 40 percent
5 of genes in a single cell of an organism, and which also indicates populations of gene
6 products in multiple functional compartments, wherein the functional compartments
7 include at least an external conditions compartment, a membrane compartment, a
8 cytosol compartment, and a DNA interactions compartment;
9 executing a plurality of sub-models that each simulate, for a same time duration called a
10 time step, a different cell process selected from a group of cell processes comprising
11 at least transcription, translation, ribosome assembly, protein translocation, and
12 metabolism;
13 updating values for the cell state data based on results from executing the plurality of
14 sub-models; and
15 repeating, for a plurality of time steps, the steps of retrieving cell state data executing the
16 plurality of sub-models and updating the values for the cell state data; and
17 storing on a computer-readable medium values of the cell state data after the plurality of
18 time steps.

FIG. 1

FEATURES OF WHOLE CELL MODEL
100

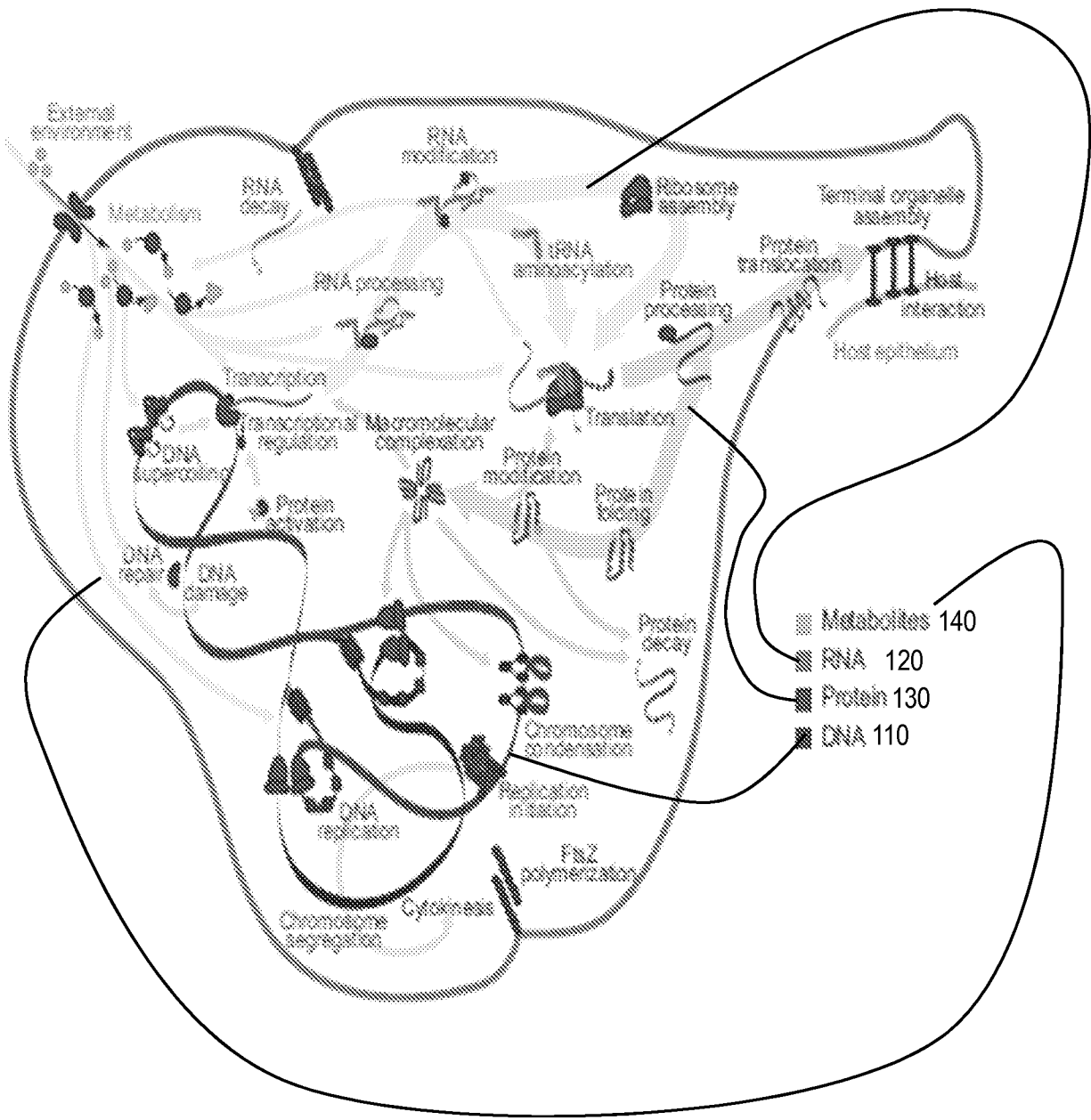


FIG. 2

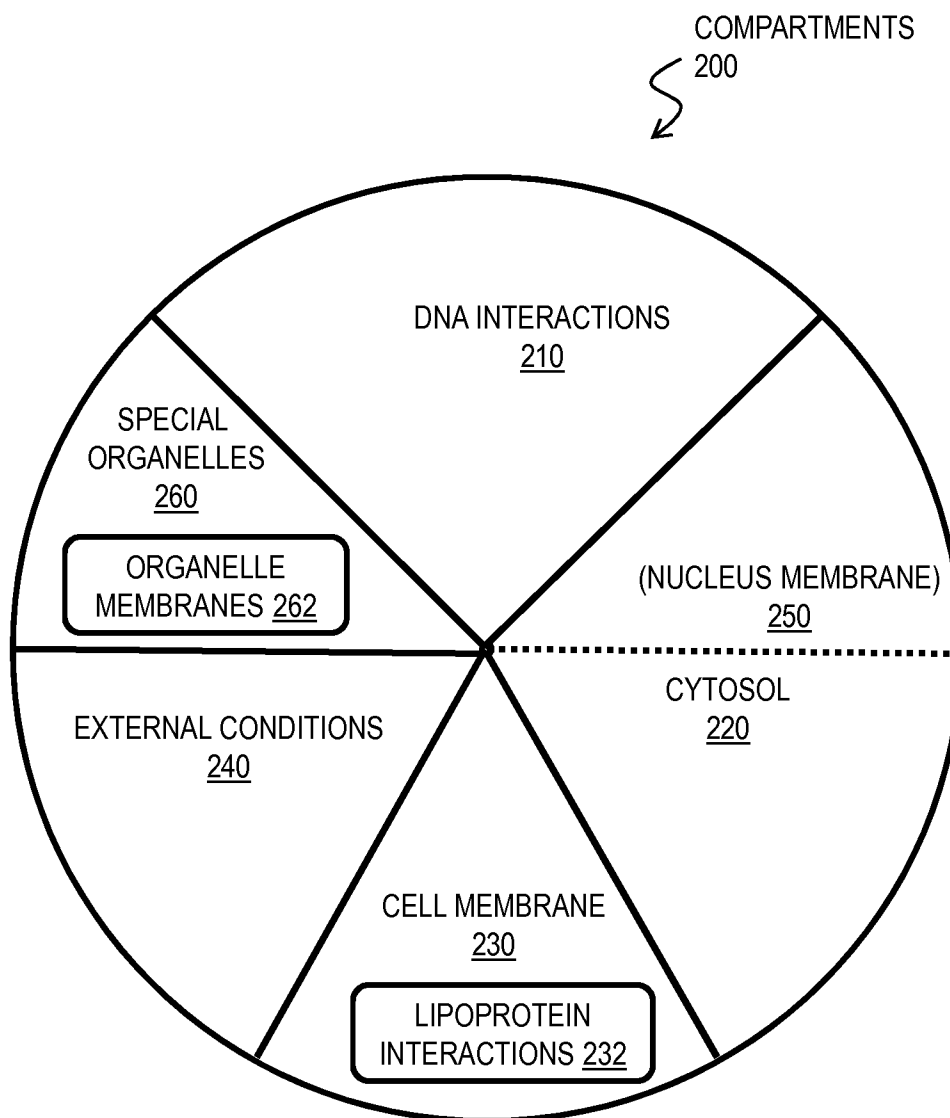


FIG. 3

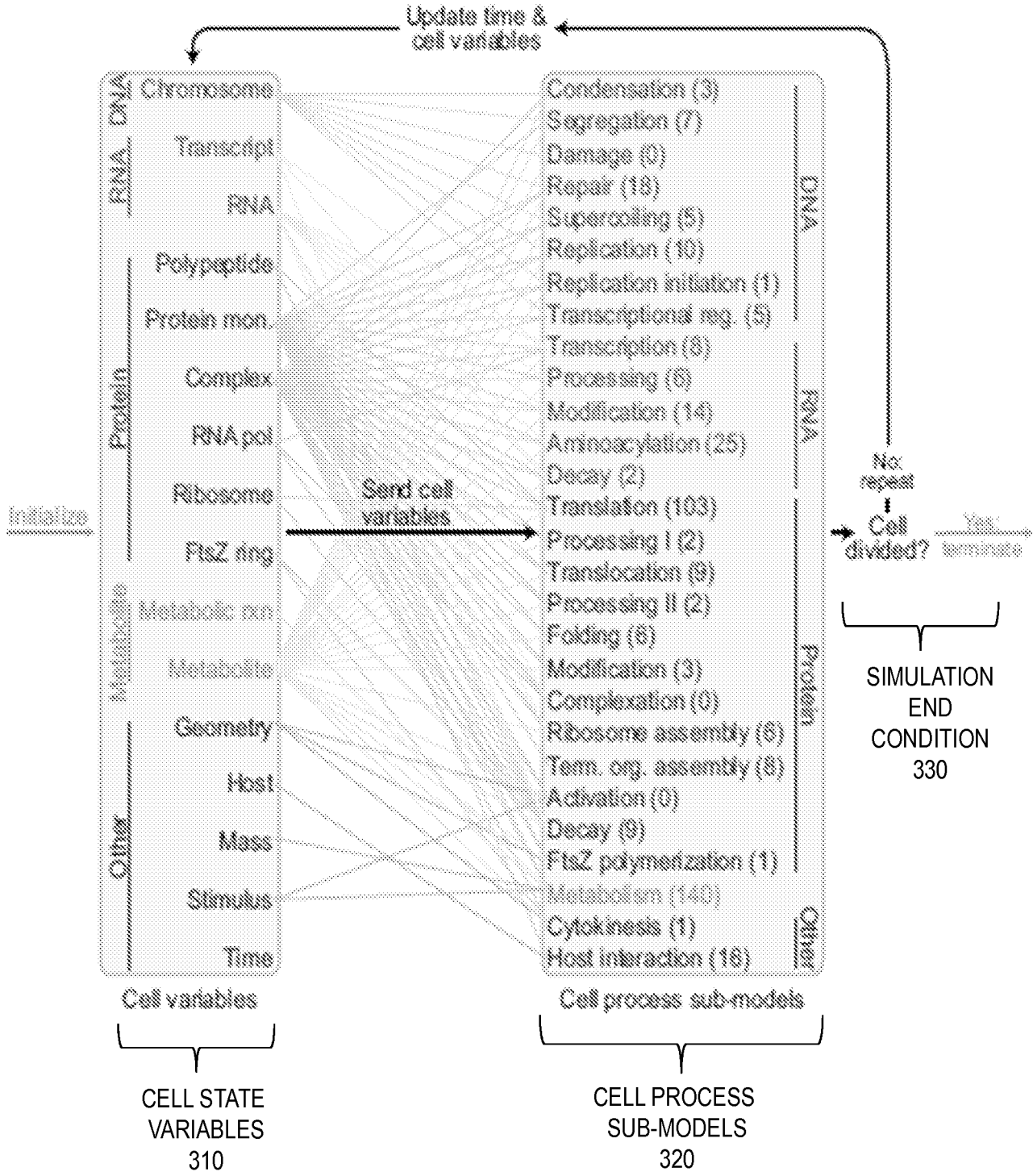


FIG. 4

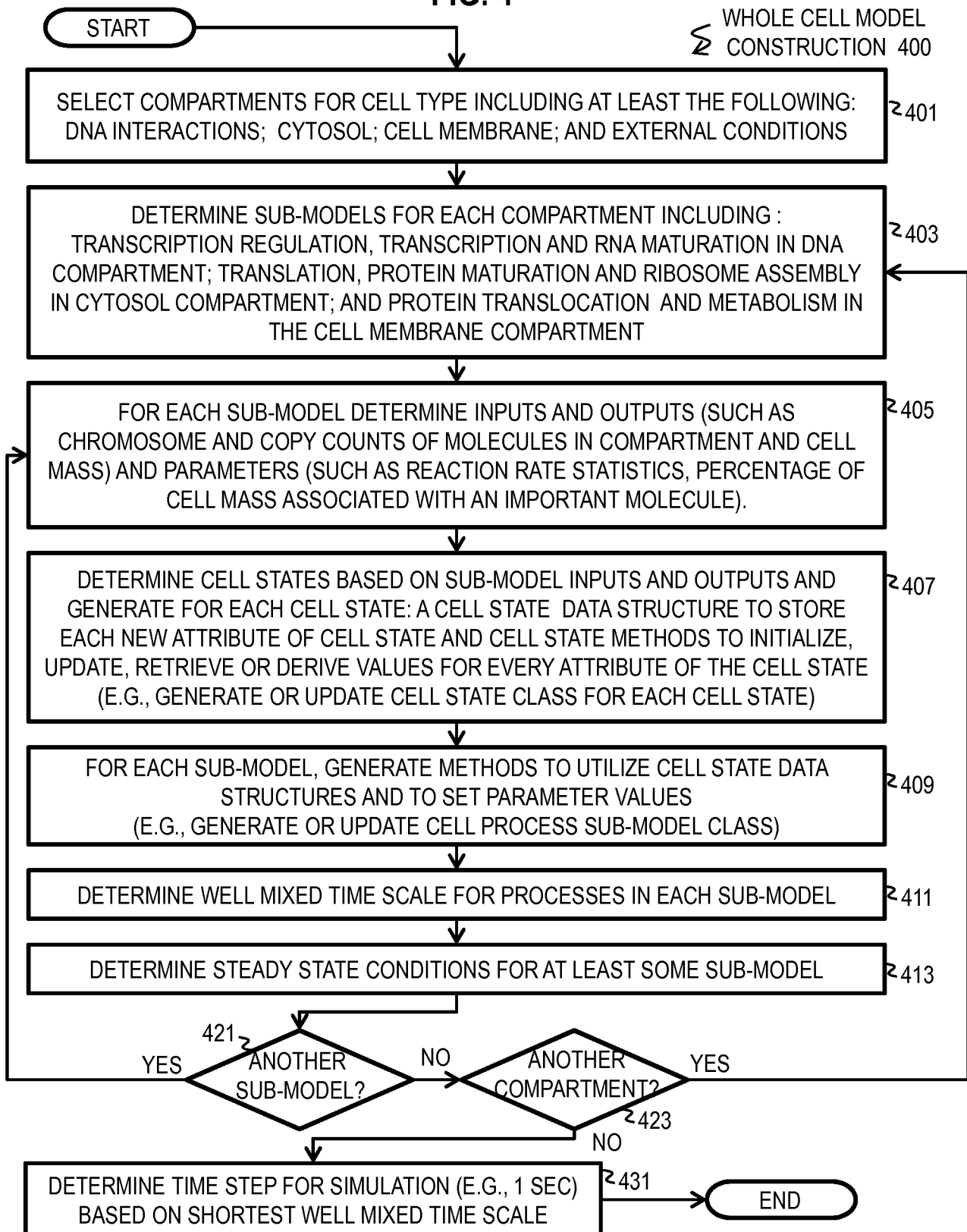


FIG. 5A

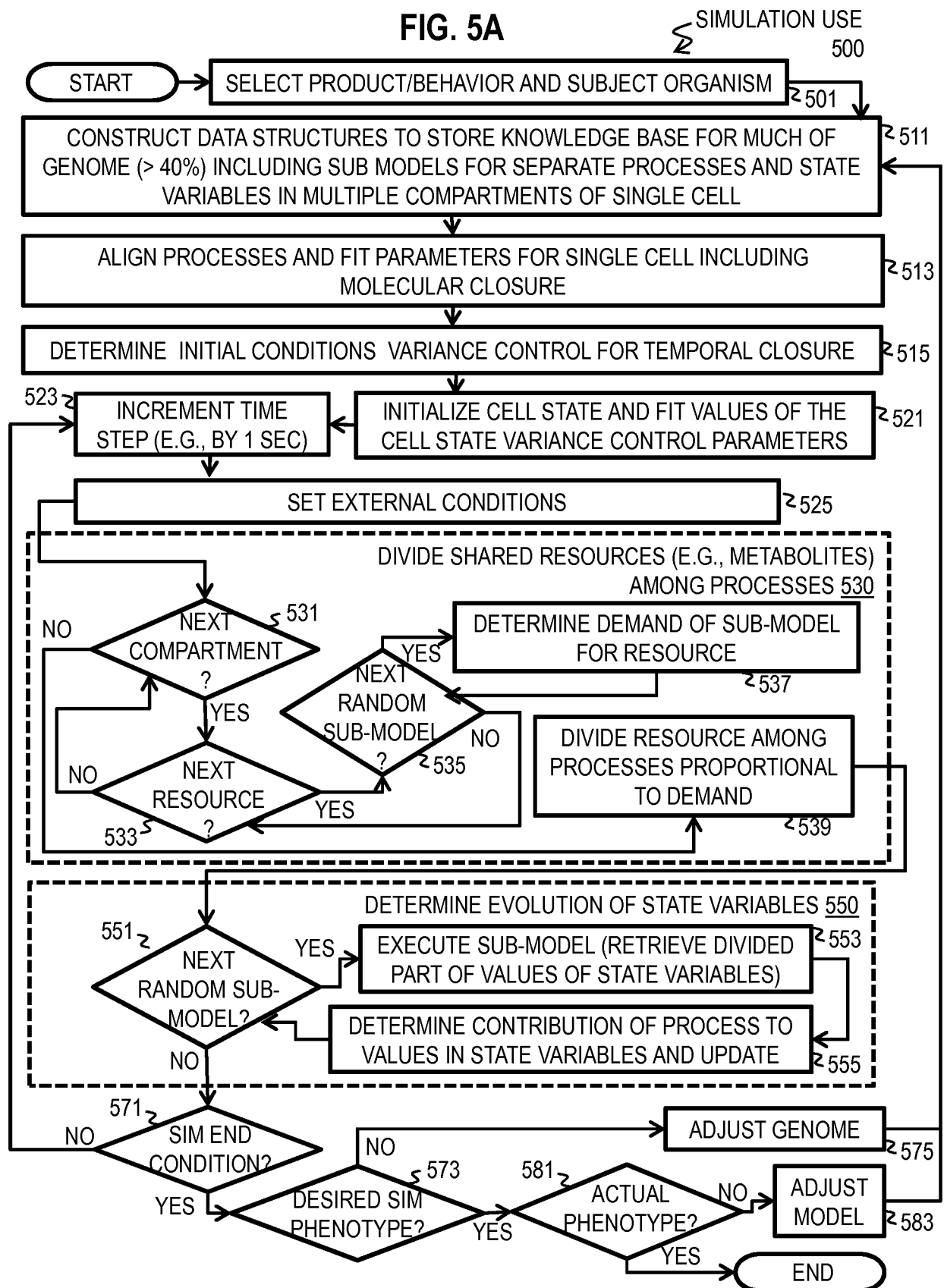


FIG. 5B

Input: $\sigma_m \leftarrow$ standard deviation of the initial total cell mass distribution
 Input: $\eta_r, \eta_p \leftarrow$ RNA and protein copy number distribution initial variance control parameters
 Input: $f_r, f_p \leftarrow$ reconstructed fractional cell RNA and protein composition
 Input: $e_r, e_p \leftarrow$ expected relative expression of each RNA and protein species
 Input: $w_r, w_p \leftarrow$ molecular weight of each RNA and protein species
 Input: $N_i (f_i, e_i, w_i) \leftarrow f_i m / (e_i^* + w_i)$ total initial RNA ($i = r$) and protein ($i = p$) copy number functions
 Set time $\leftarrow 0$ s
 Set values of external stimuli and metabolites according to observations
 Set total cell mass, $m \leftarrow \sim N(\mu, \sigma_m)$, and calculate cell volume and shape
 Set the metabolite counts according to the total cell mass and reconstructed cell composition (see Tab. S42)
 Initialize the Chromosome state with one methylated chromosome; decrement dNMP counts to maintain cell mass
 Set mature RNA copy numbers according to multinomialRand(η_r, N_r, e_r)/ η_r ; decrement NMP counts
 Set mature protein monomer copy numbers according to multinomialRand(η_p, N_p, e_p)/ η_p ; decrement amino acids
 Form macromolecules by calculating the steady-state of the Macromolecular Complexation process
 Set the RNA Polymerase and Transcript states to a steady-state of the Transcription submodel
 Set the Ribosome and Polypeptide states to a steady-state of the Translation submodel
 Set the FtsZ state to a steady-state of the FtsZ Polymerization submodel with no septal rings
 Set the growth rate and metabolic reaction fluxes to a steady-state of the Metabolism submodel
 Set the Host state to a steady-state of the Host Interaction submodel
 Set the chromosome protein occupancy to a steady-state the chromosome-interacting submodels

FIG. 5C

Initialize the initial cell state variance control parameters: $\sigma_m \leftarrow 0$, $\eta_r \leftarrow 0$, $\eta_p \leftarrow 0$

repeat

- Simulate the life cycle of a population of wild type cells using **FIG. 5B** to initialize the value of each cell state variable
- Randomly segregate the cellular content into two daughter cells
- Calculate the variances of the total cell mass, RNA copy number, and protein copy number states
- Set the values of the initial distribution control parameters of each state equal to that of the final distribution
- $\sigma_m \leftarrow$ standard deviation of the final cell mass distribution
- $\eta_r \leftarrow \sigma_r^2/N_r$, where N_r and σ_r^2 are the mean and variance of the final RNA copy number distribution
- $\eta_p \leftarrow \sigma_p^2/N_p$, where N_p and σ_p^2 are the mean and variance of the final **protein** copy number distribution

until the initial variance control parameters (σ_m , η_r , and η_p) converge

FIG. 6A

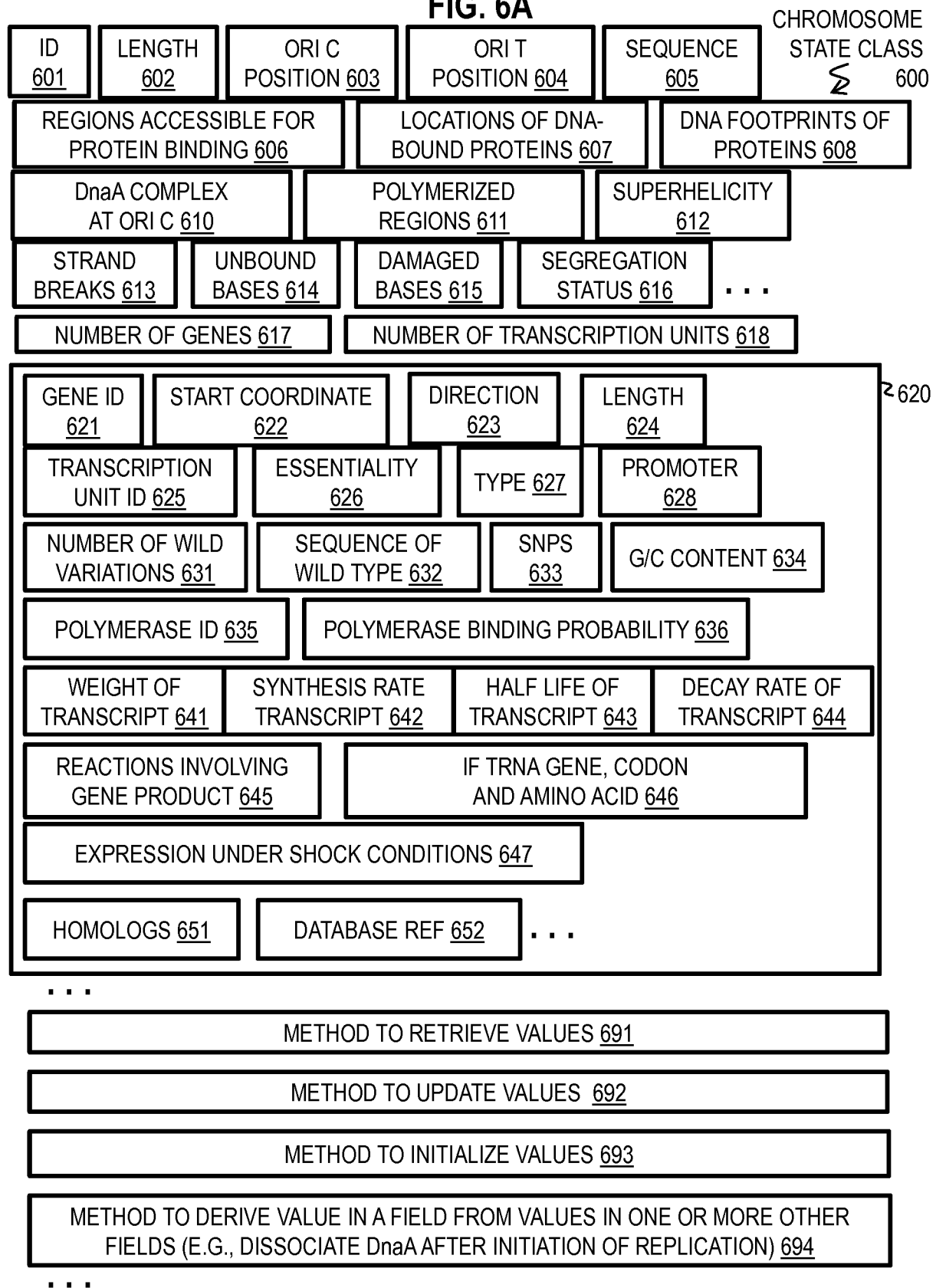


FIG. 6B

671	672	673	674
Physical Property	Symbol	Size	Type
Polymerization	p_{ijk}	$L \times 2 \times 2$	Boolean
Winding	w_{ijk}	$L \times 2 \times 2$	Real
Modification			
Gap site	m_{ijk}^g	$L \times 2 \times 2$	Boolean
Abasic site	m_{ijk}^a	$L \times 2 \times 2$	Boolean
Sugar-phosphate	m_{ijkl}^{sp}	$L \times 2 \times 2 \times M$	Boolean
Base	m_{ijkl}^b	$L \times 2 \times 2 \times M$	Boolean
Intrastrand cross link	m_{ijk}^c	$L \times 2 \times 2$	Boolean
Strand break	m_{ijk}^s	$L \times 2 \times 2$	Boolean
Holliday junction	m_{ijk}^h	$L \times 2 \times 2$	Boolean
Protein occupancy			
Monomer	b_{ijkl}^m	$L \times 2 \times 2 \times B^m$	Boolean
Complex	b_{ijkl}^c	$L \times 2 \times 2 \times B^c$	Boolean
Catenation	s	1×1	Boolean

FIG. 6C

$$\begin{array}{c}
 \begin{array}{cc}
 672 & 679 \\
 \hline
 \end{array} \\
 \begin{array}{l}
 p_{ij1} = 1 \quad \forall i, j \\
 p_{ij2} = 0 \quad \forall i, j \\
 w_{ij1} = (1 + \sigma_{\text{un}}) \frac{L}{h} \quad \forall i, j \\
 w_{ij2} = 0 \quad \forall i, j \\
 m_{ij,k}^a = 0 \quad \forall i, j, k \\
 m_{ij,k}^b = 0 \quad \forall i, j, k \\
 m_{ij,k}^p = 0 \quad \forall i, j, k \\
 m_{ij1}^b = \begin{cases} 1 & i \in \text{Mun1 methylation sites}, \forall j \\ 0 & \text{otherwise} \end{cases} \\
 m_{ij2}^b = 0 \quad \forall i, j \\
 m_{ij,k}^c = 0 \quad \forall i, j, k \\
 m_{ij,k}^d = 0 \quad \forall i, j, k \\
 m_{ij,k}^h = 0 \quad \forall i, j, k \\
 s = 0,
 \end{array}
 \end{array}$$

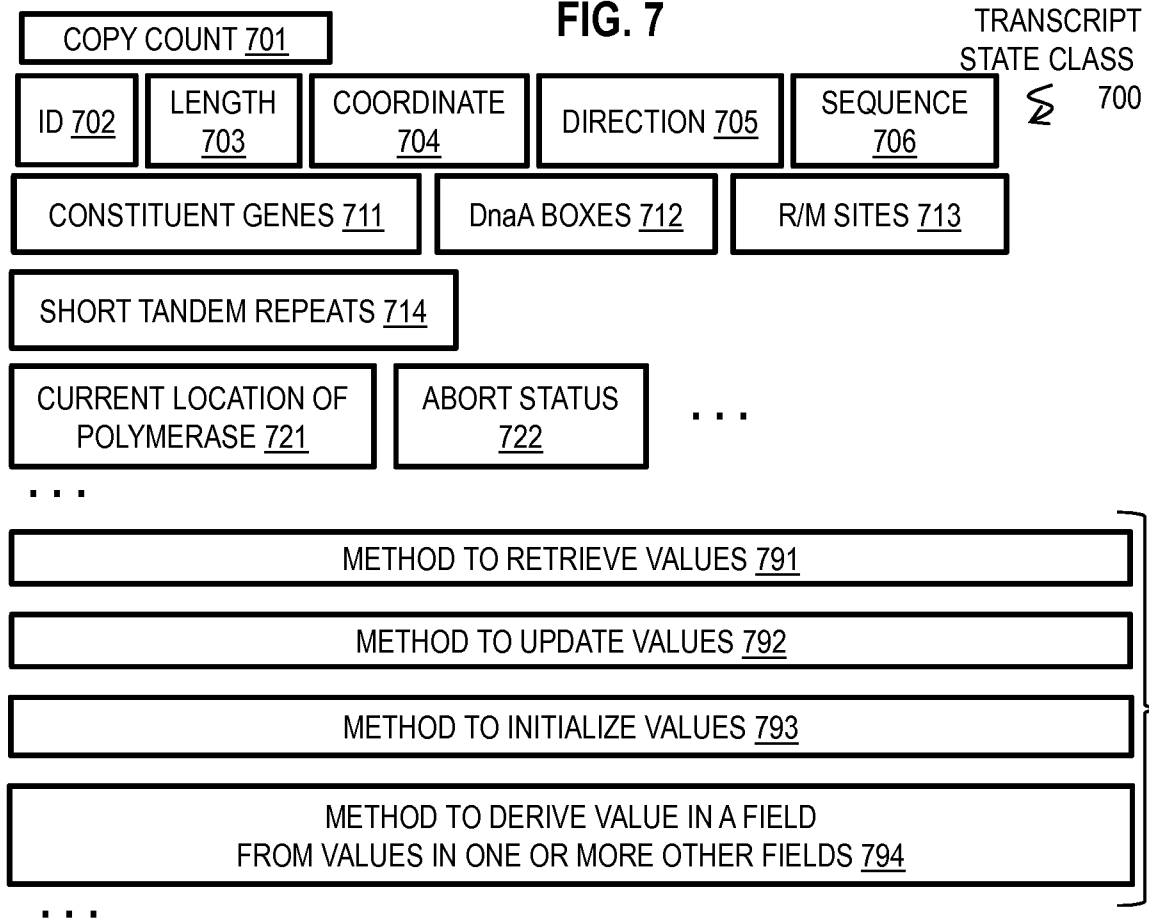
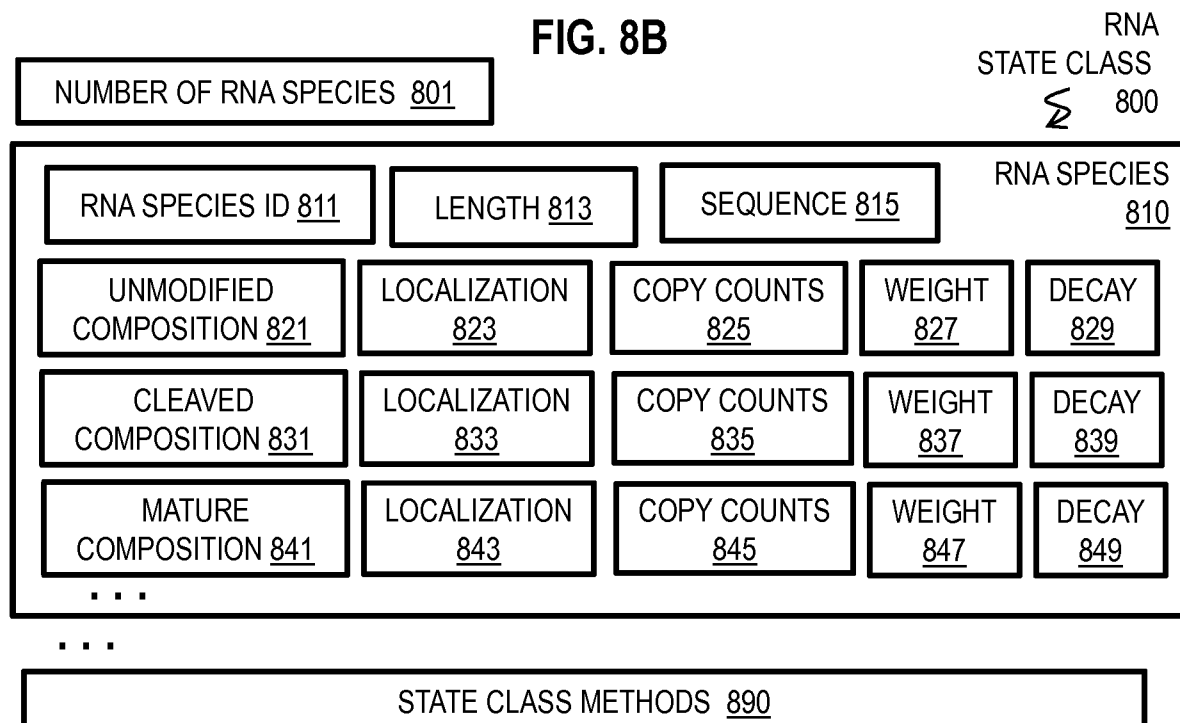
FIG. 7**FIG. 8B**

FIG. 8A

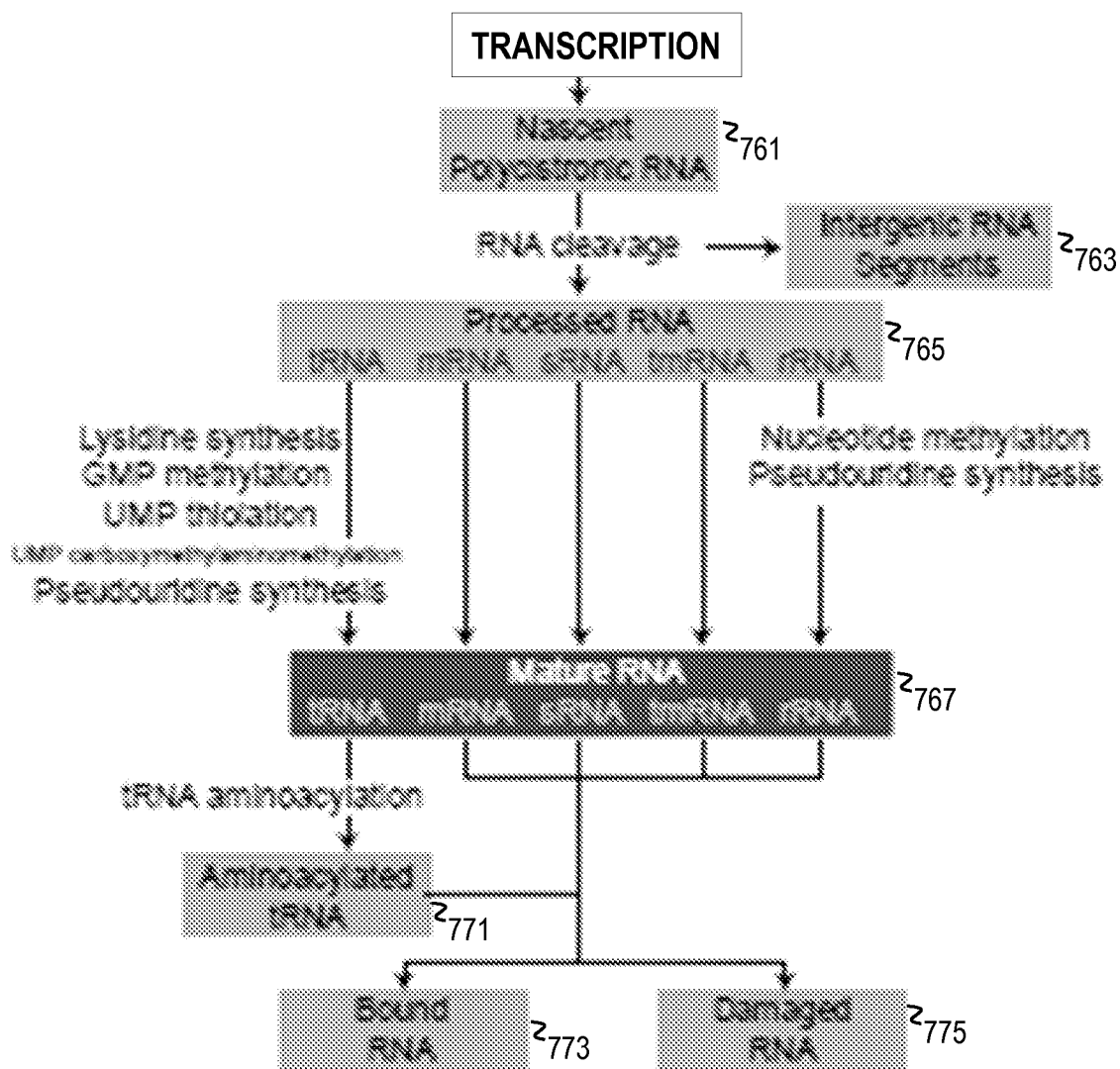


FIG. 8C

Connected Processes	Read from Raa	Written to Raa
Transcription	• Counts of nascent RNAs	• Updated counts of nascent RNAs
RNA Processing	• Counts of nascent, processed, and intergenic RNAs	• Updated counts of nascent, processed, and intergenic RNAs
RNA Modification	• Counts of processed and mature RNA	• Updated counts of processed and mature RNA
tRNA Aminoacylation	• Counts of mature and aminoacylated tRNA	• Updated counts of mature and aminoacylated tRNA
RNA Decay	• Counts of mature, misfolded, and damaged RNA	• Updated counts of mature, misfolded, and damaged RNA
Various Processes	• Counts of mature and bound RNAs	• Updated counts of mature and bound RNAs

FIG. 9

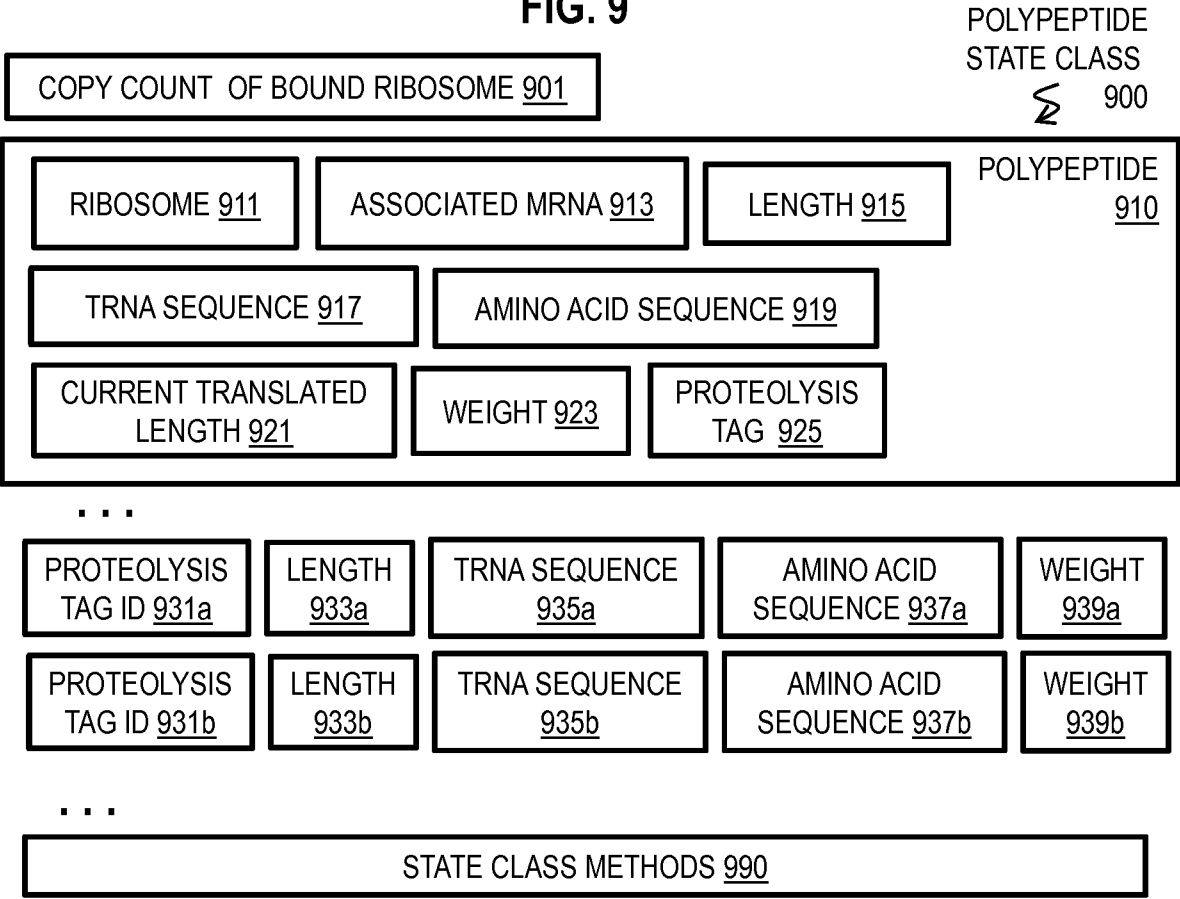


FIG. 10A

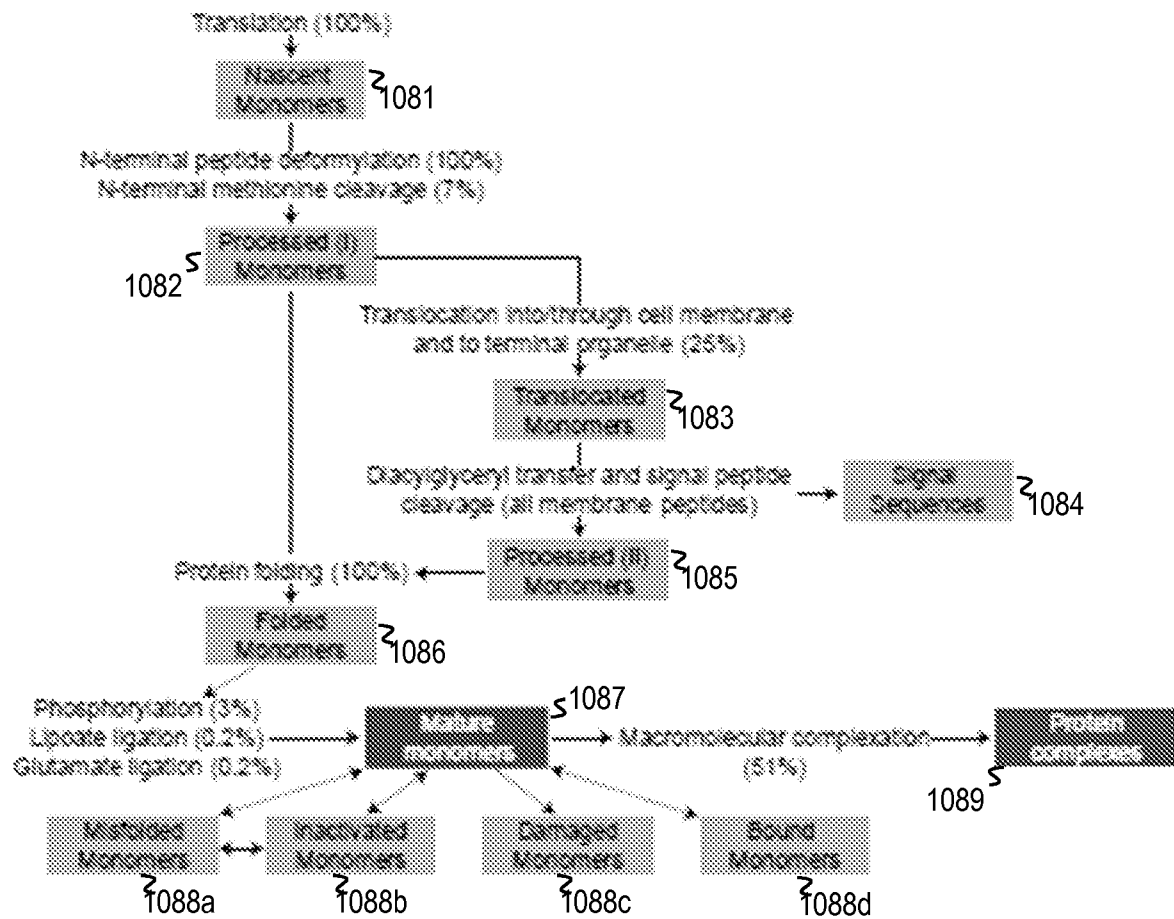


FIG. 10B

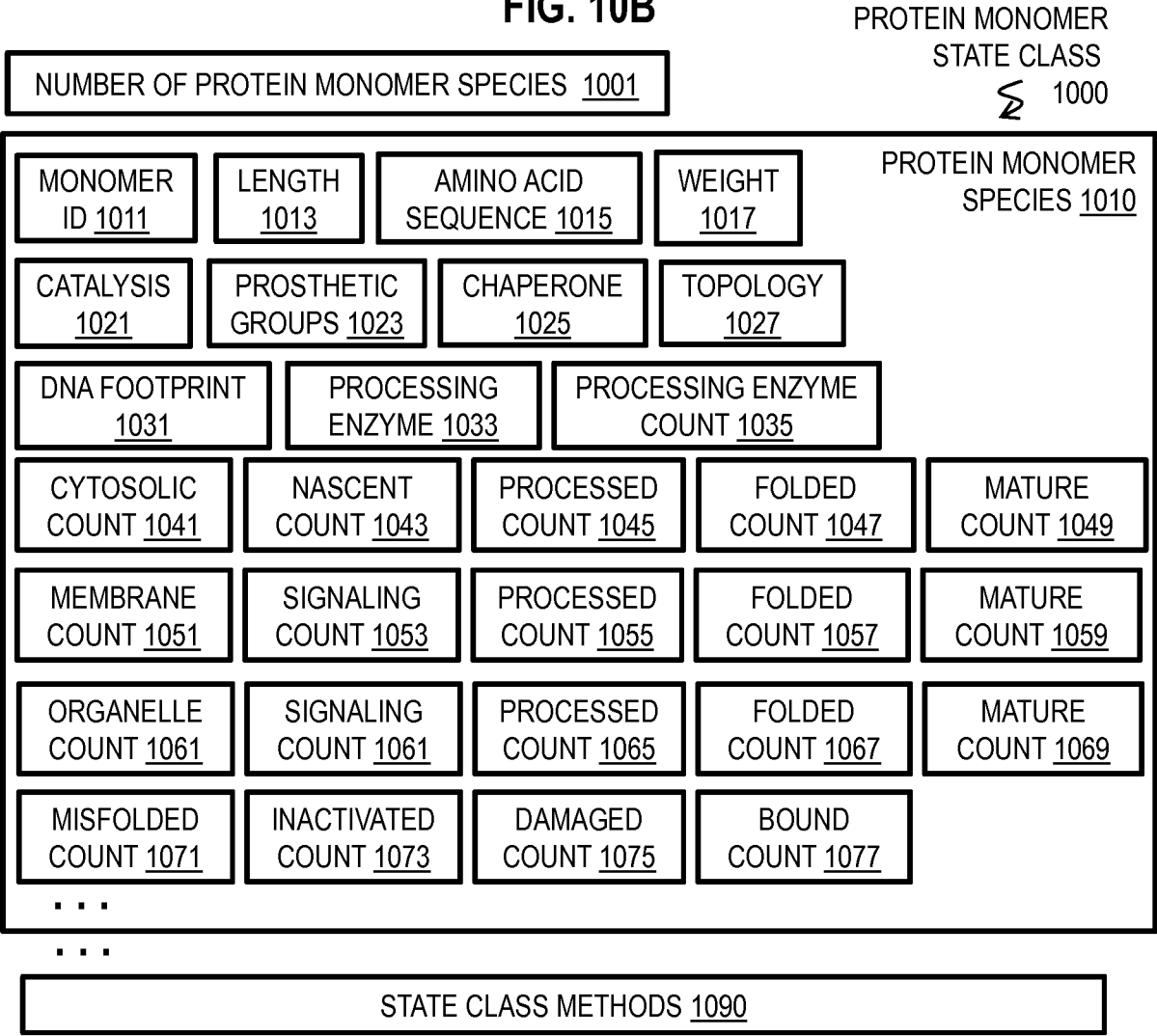


FIG. 10C

Connected Processes	Read from Protein Monomer	Written to Protein Monomer
Translation	• Counts of nascent monomers	• Updated counts of nascent monomers
Protein Processing I	• Counts of nascent and processed (I) monomers	• Updated counts of nascent and processed (I) monomers
Protein Translocation	• Counts of processed (I) and translocated monomers	• Updated counts of processed (I) and translocated monomers
Protein Processing II	• Counts of translocated and processed (II) monomers and signal sequences	• Updated counts of translocated and processed (II) monomers and signal sequences
Protein Folding	• Counts of processed (II) and folded monomers	• Updated counts of processed (II) and folded monomers
Protein Modification	• Counts of folded and mature monomers	• Updated counts of folded and mature monomers
Macromolecular Complexation	• Counts of mature monomers	• Updated counts of mature monomers
Protein Decay	• Counts of monomers • Monomer half lives • Monomer amino acid composition	• Updated counts of monomers • Assignment of damaged monomers • Assignment of misfolded monomers
Protein Activation	• Counts of mature and inactivated monomers	• Updated counts of mature and inactivated monomers
Various other processes	• Counts of mature and bound monomers • Protein monomer counts to determine maximal enzyme activity	• Updated counts of mature and bound monomers

FIG. 11A

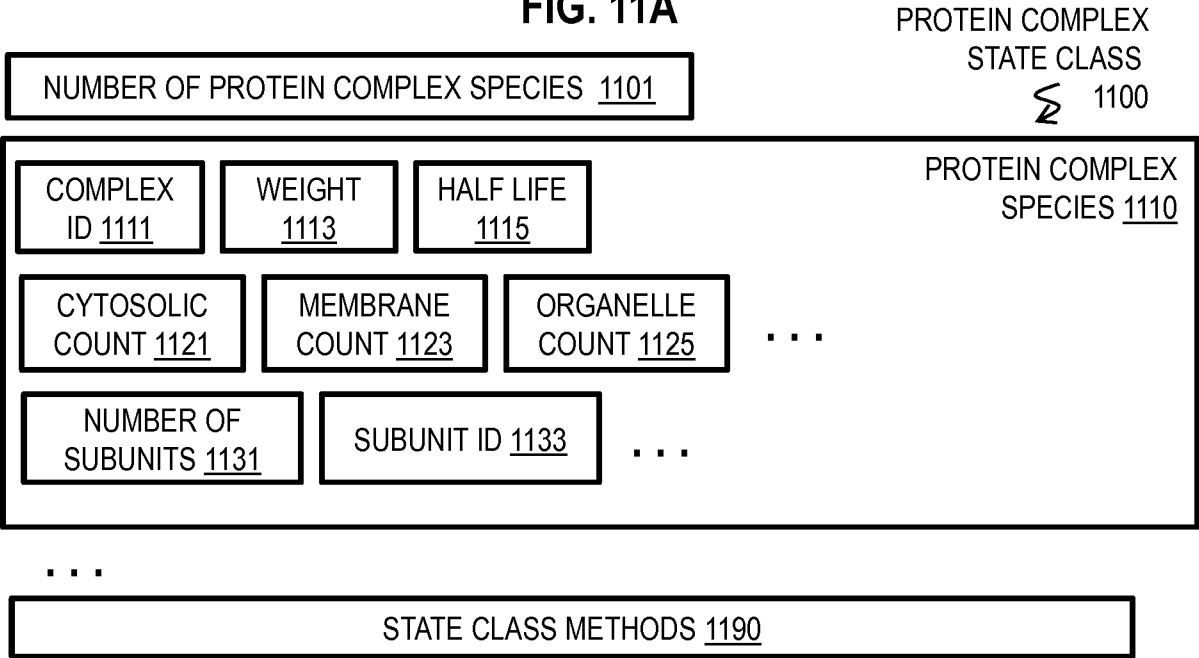


FIG. 11B

Connected Processes	Read from Protein Complex	Written to Protein Complex
Macromolecular Complexation	* Counts of protein complexes * Complex subunit composition	* Updated counts of protein complexes
Ribosome Assembly	* Counts of ribosomes	* Updated counts of ribosomes
Replication	* Counts of replication complexes	* Updated counts of replication complexes
Replication Initiation	* Counts of DnaA complexes	* Updated counts of DnaA complexes
Protein Decay	* Counts of protein complexes * Complex half lives * Complex amino acid composition	* Updated counts of protein complexes
Transcription	* Minimum average expression	
Various other processes	* Protein complex counts to determine maximal enzyme activity	

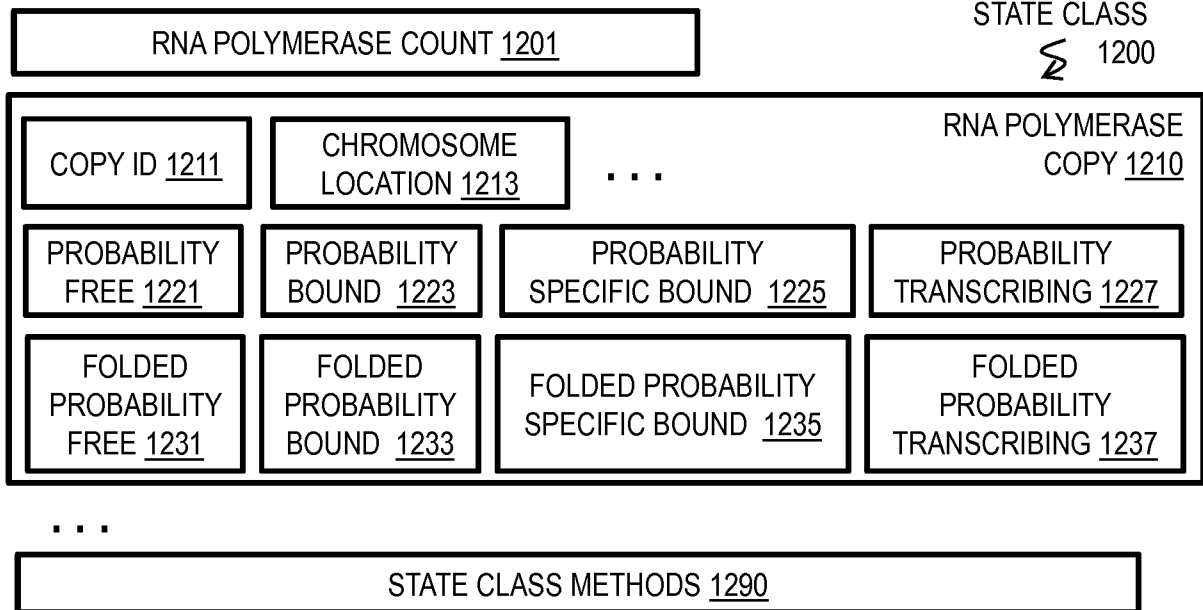
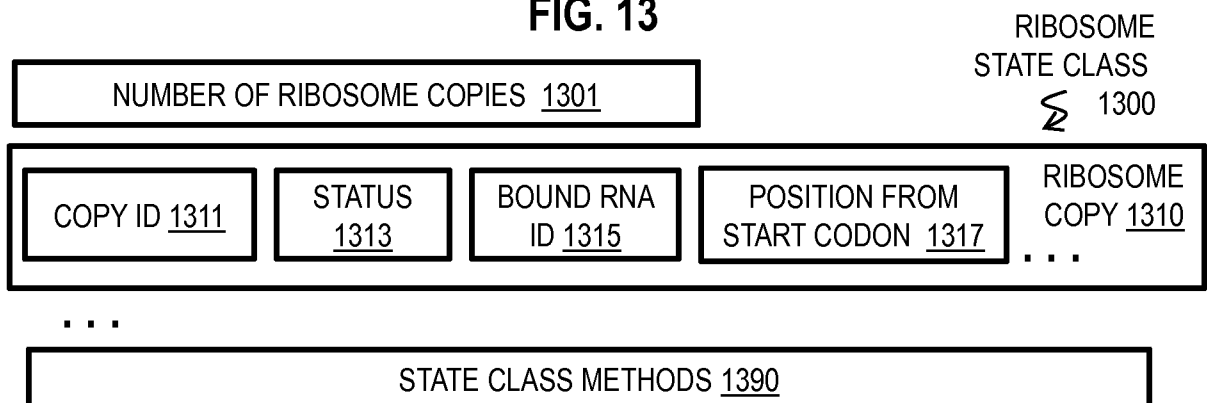
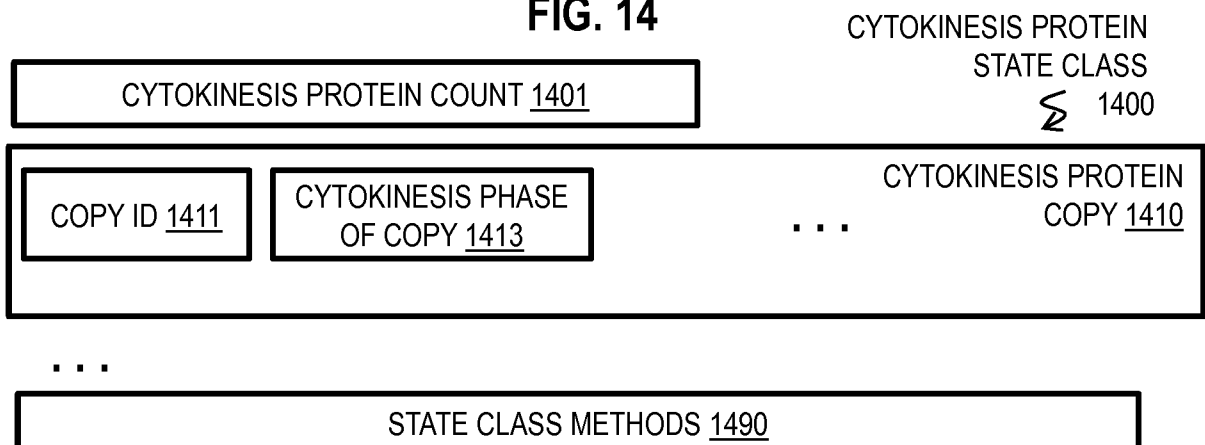
FIG. 12**FIG. 13****FIG. 14**

FIG. 15

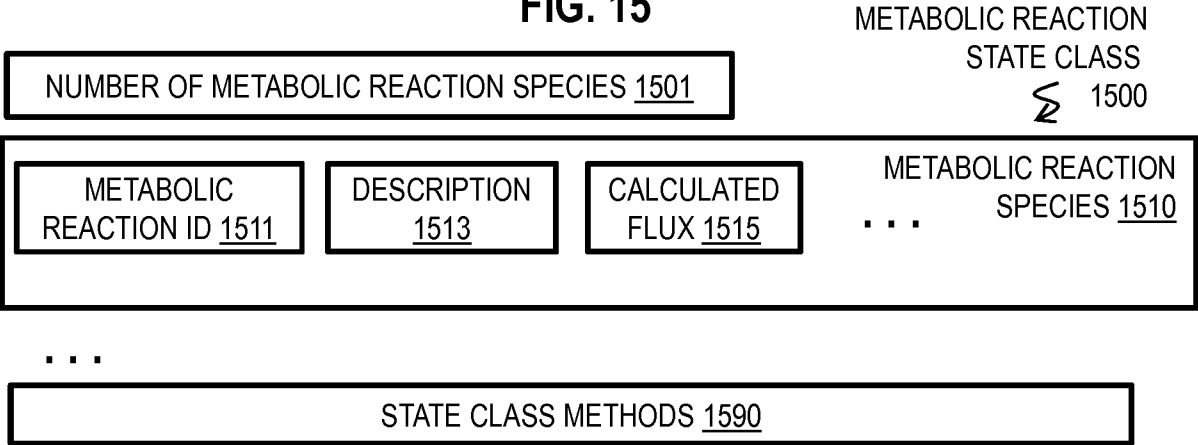


FIG. 16

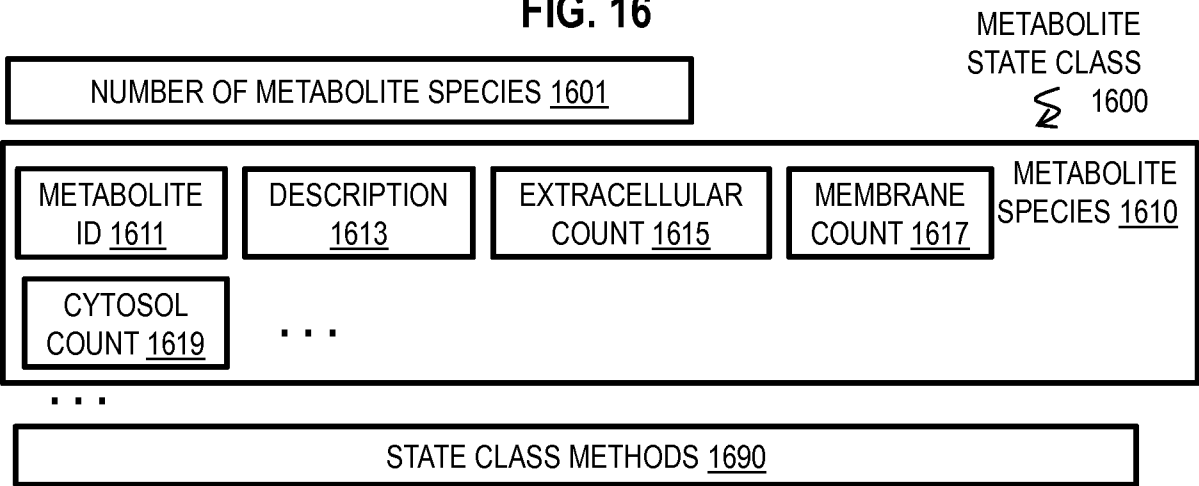


FIG. 17A

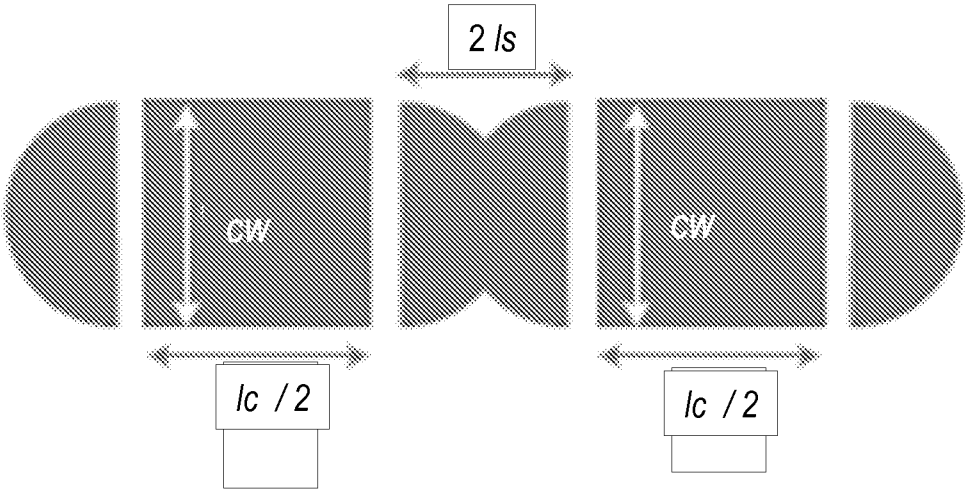


FIG. 17B

GEOMETRY
STATE CLASS
 $\lesssim$ 1700

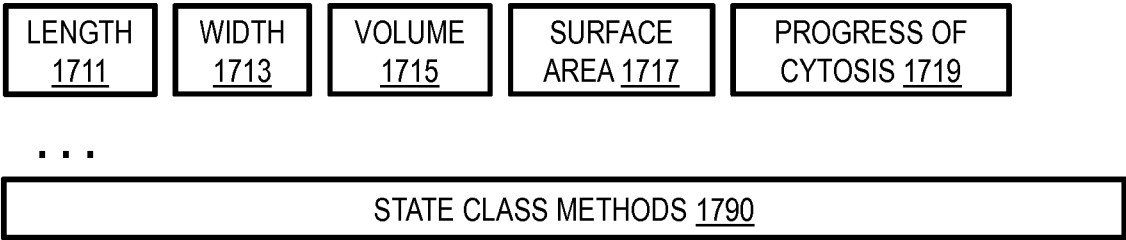


FIG. 18

HOST
STATE CLASS
 $\lesssim$ 1800

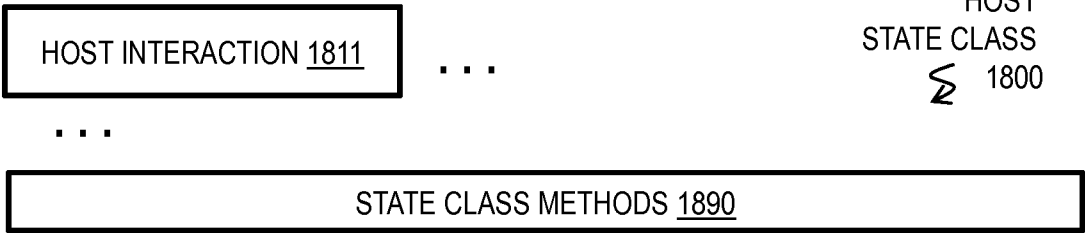


FIG. 19

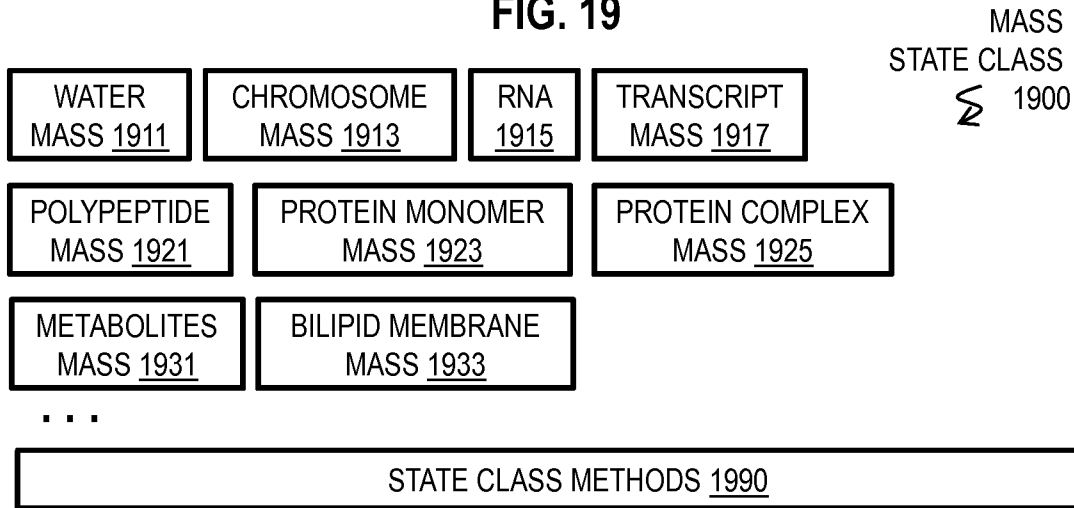


FIG. 20

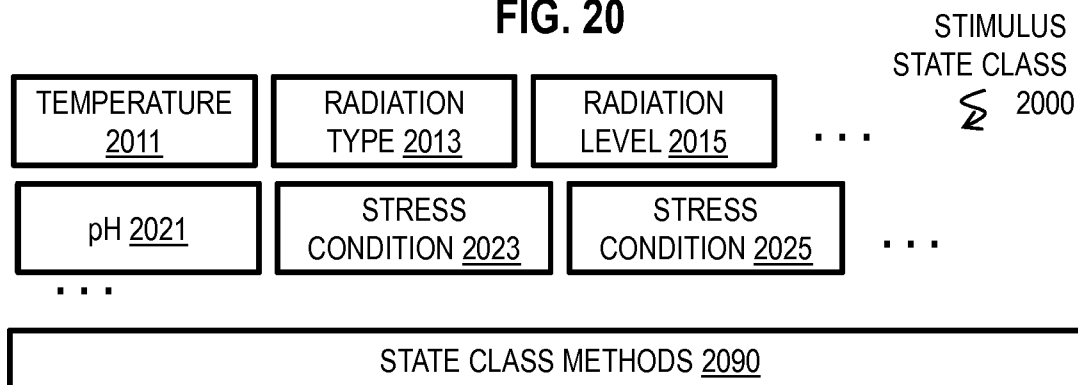


FIG. 21

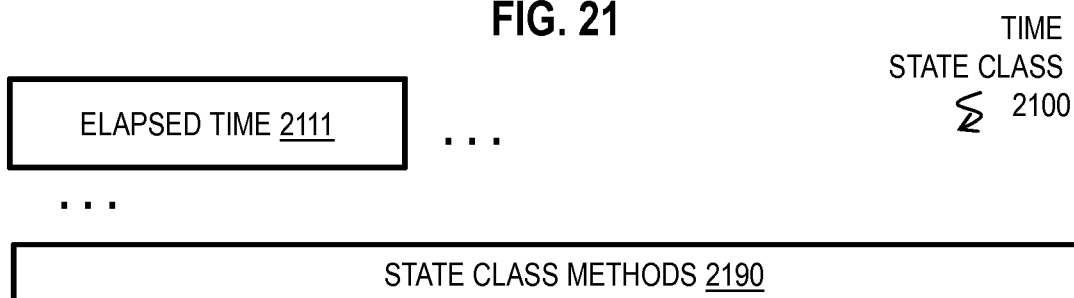


FIG. 22A

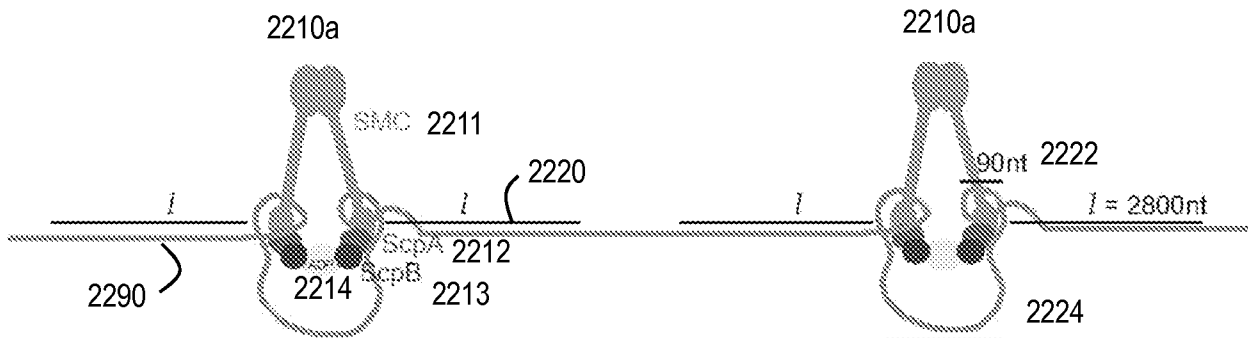
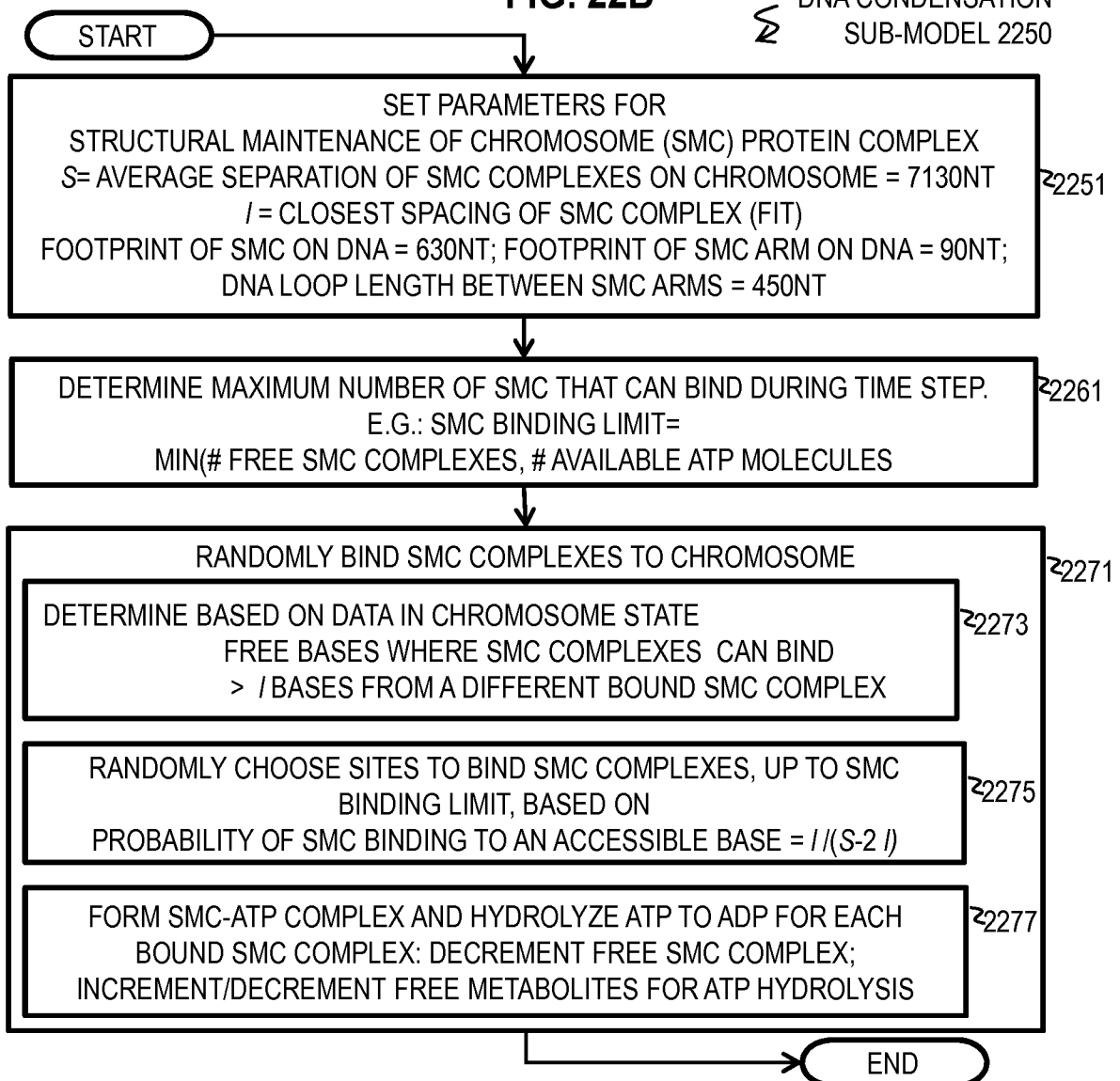


FIG. 22B

Σ DNA CONDENSATION
SUB-MODEL 2250



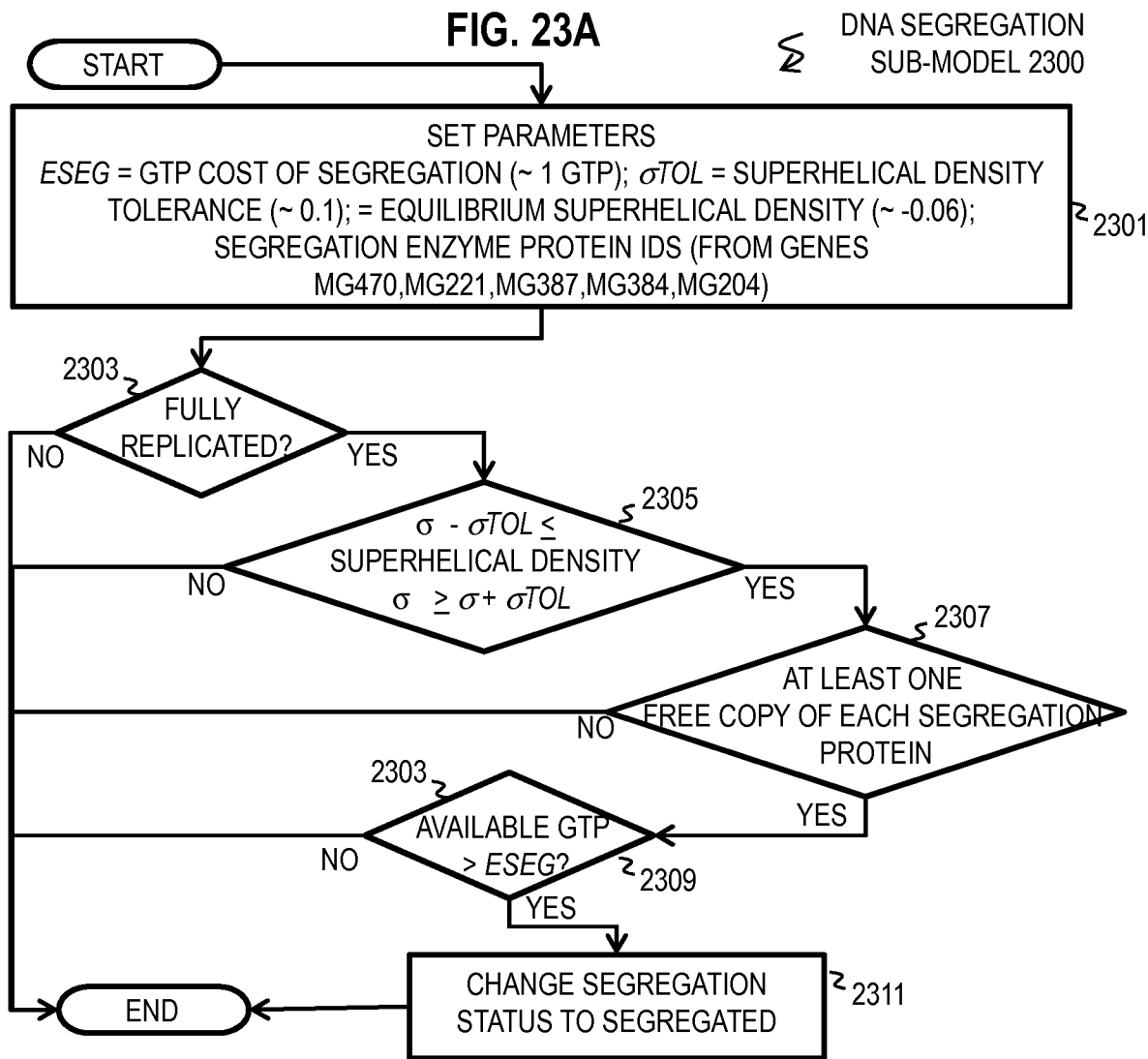


FIG. 23B

Connected States	Read from state	Written to state
Chromosome	• Is chromosome replicated?	• Chromosomes are segregated
	• Are chromosome superhelical densities within tolerance?	
	• Are chromosomes segregated?	

FIG. 24A

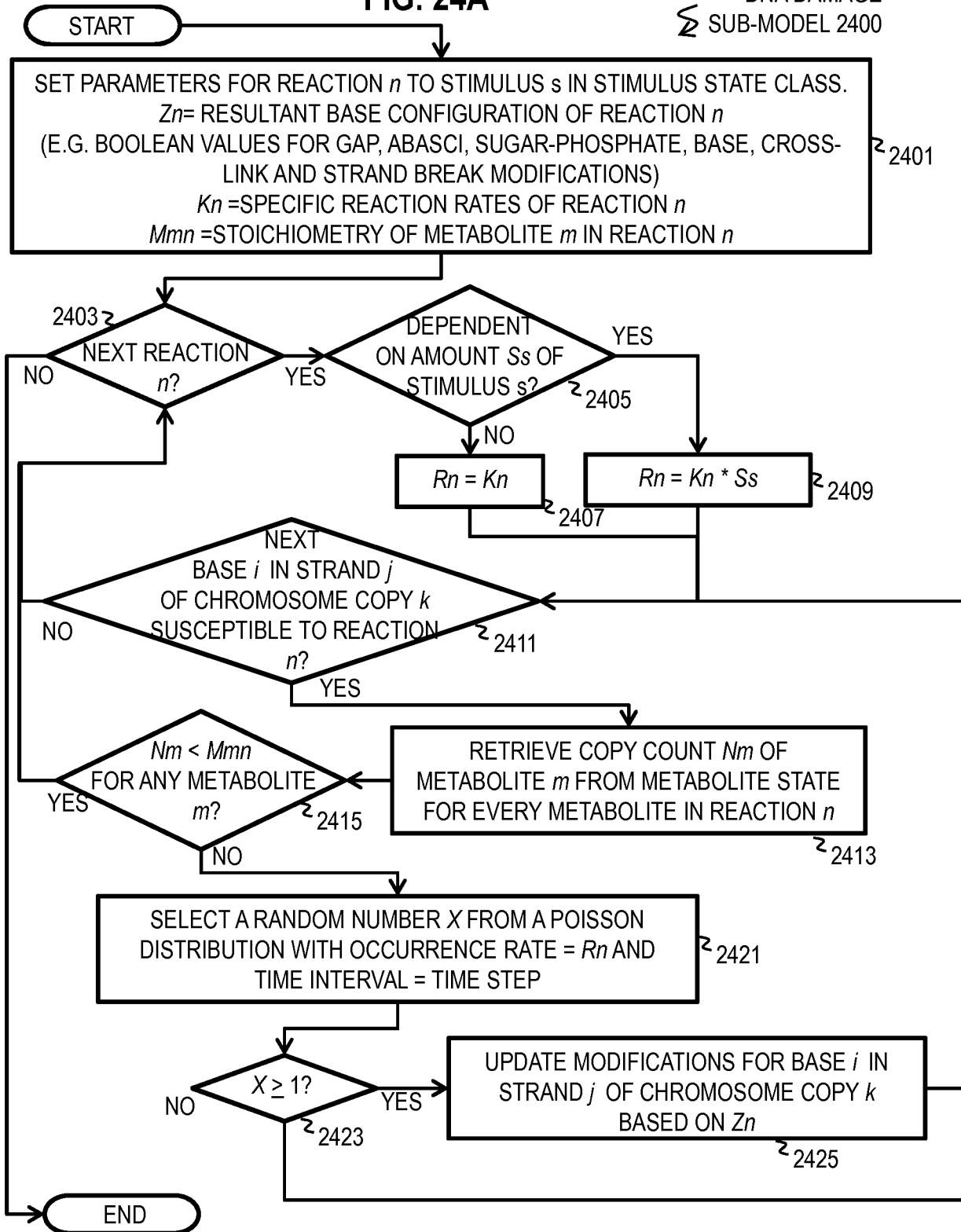
DNA DAMAGE
Σ SUB-MODEL 2400

FIG. 24B

Input: m_i copy number of metabolite i

Input: M_{ij} stoichiometry of metabolite i in reaction j

Input: $z_i^a, z_i^b, Z_{\bullet i}^a, Z_{\bullet i}^b, z_i^c$, and z_i^d : final base configuration resulting from reaction i

foreach DNA modification reaction i do

 Calculate base modification rate

 switch trigger of reaction i do

 case spontaneous

$r_i \leftarrow k_i$

 case radiation

$j \leftarrow$ index of radiation trigger

$r_i \leftarrow k_i s_j$

 foreach base j in strand k of chromosome l susceptible to reaction i in a random order do

 If insufficient metabolic resources to support reaction i ($\exists j$ s.t. $m_j < -M_{ji}$) then

 break

 If $\text{poissonRand}(r_i) \geq 1$ then

 Update the configuration of base j of strand k of chromosome l

$m_{jkl}^a \leftarrow z_i^a$

$m_{jkl}^b \leftarrow z_i^b$

$m_{jkl\bullet}^a \leftarrow Z_{\bullet i}^a$

$m_{jkl\bullet}^b \leftarrow Z_{\bullet i}^b$

$m_{jkl}^c \leftarrow z_i^c$

$m_{jkl}^d \leftarrow z_i^d$

 Update metabolite copy numbers: $m \leftarrow m + M_{\bullet i}$

FIG. 25A

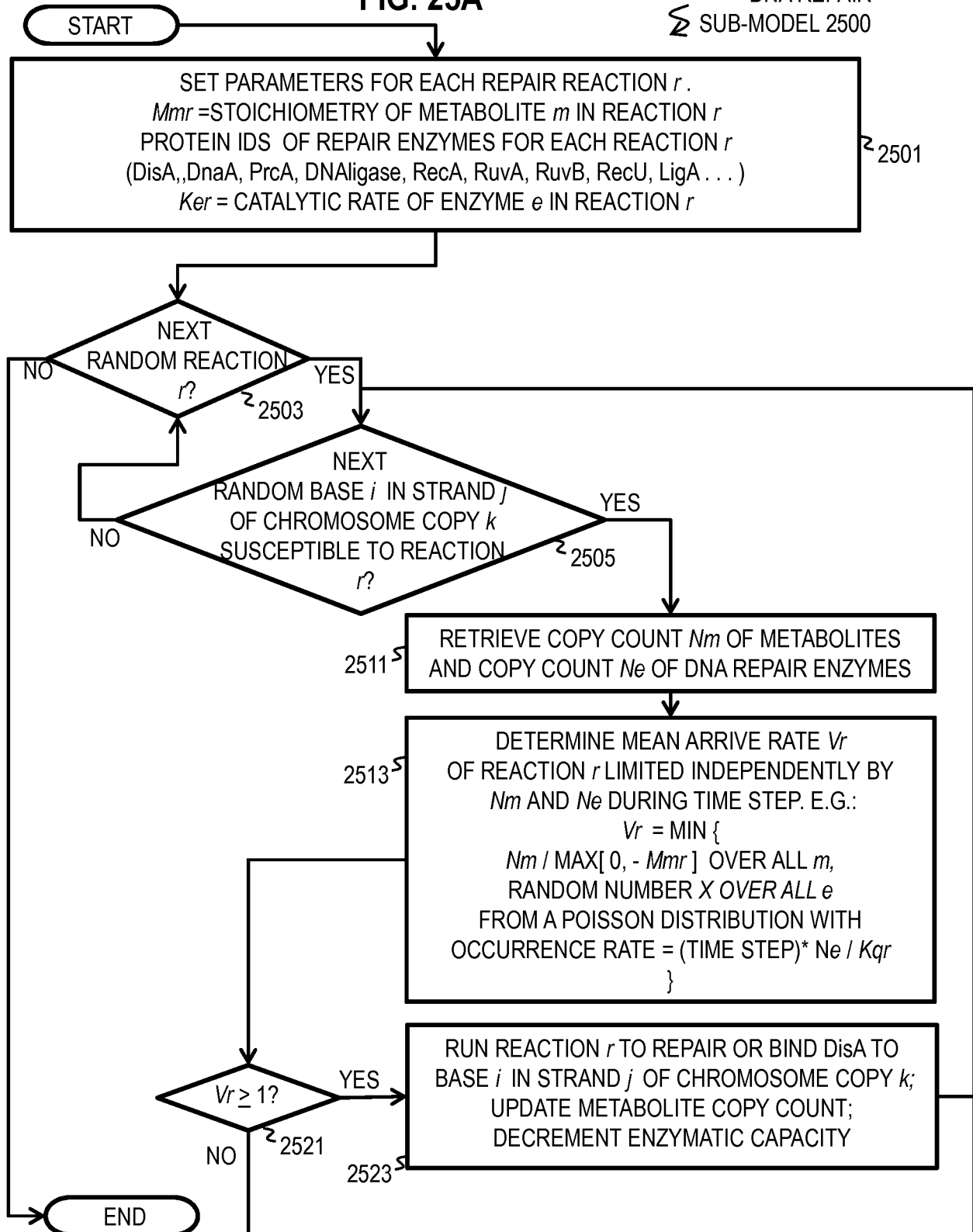
DNA REPAIR
Σ SUB-MODEL 2500

FIG. 25B

Repair, methylate, and restrict DNA

foreach reaction *i* in a random order **do**

```
┌   foreach base j of strand k of chromosome l susceptible to reaction i in a random order do  
│   if sufficient enzymatic and metabolic resources to support reaction i then  
│   │   Execute reaction i  
│   │   Update DNA configuration  
│   │   Update metabolite copy numbers  
│   │   Decrement enzymatic capacity  
└
```

Bind DisA to damaged DNA

while there is free DisA and at least 1 DisA-accessible DNA lesion **do**

```
┌   Let i, j, k ← represent the base, strand, and chromosome of a DisA-accessible DNA lesion  
└   Bind DisA to base i of strand j of chromosome k
```

FIG. 26A

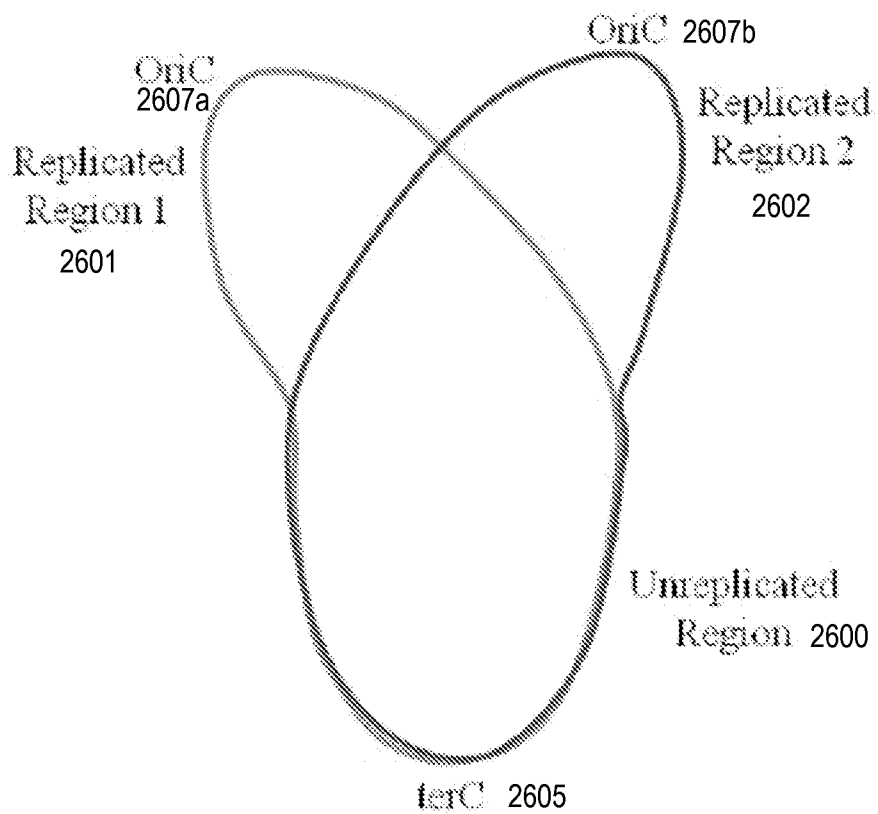


FIG. 26B

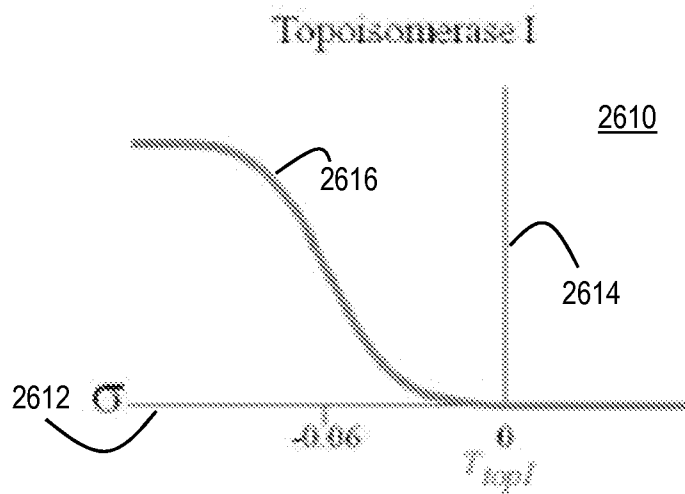


FIG. 26C

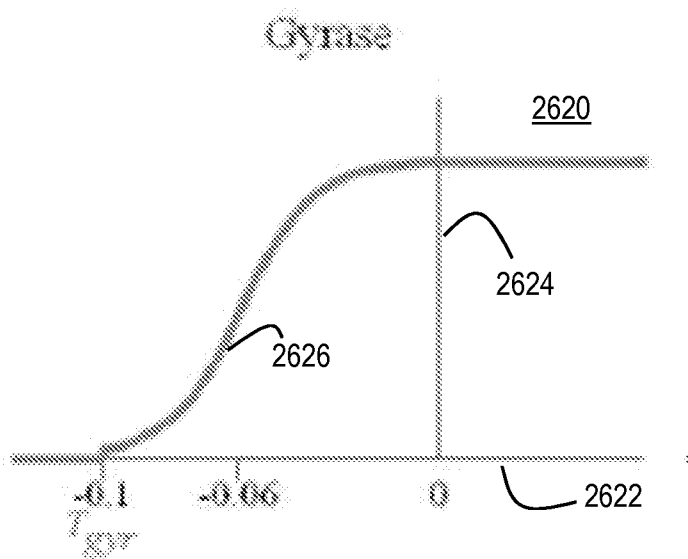


FIG. 26D

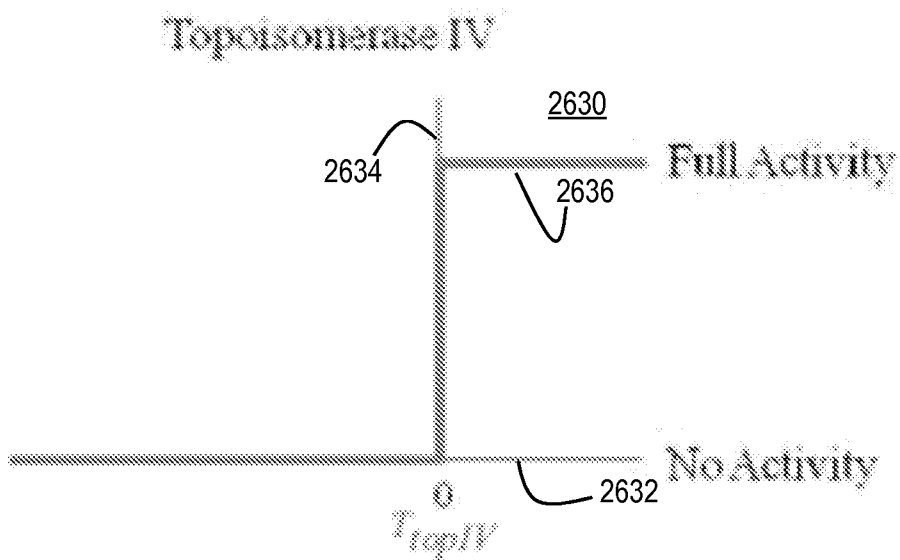


FIG. 26E

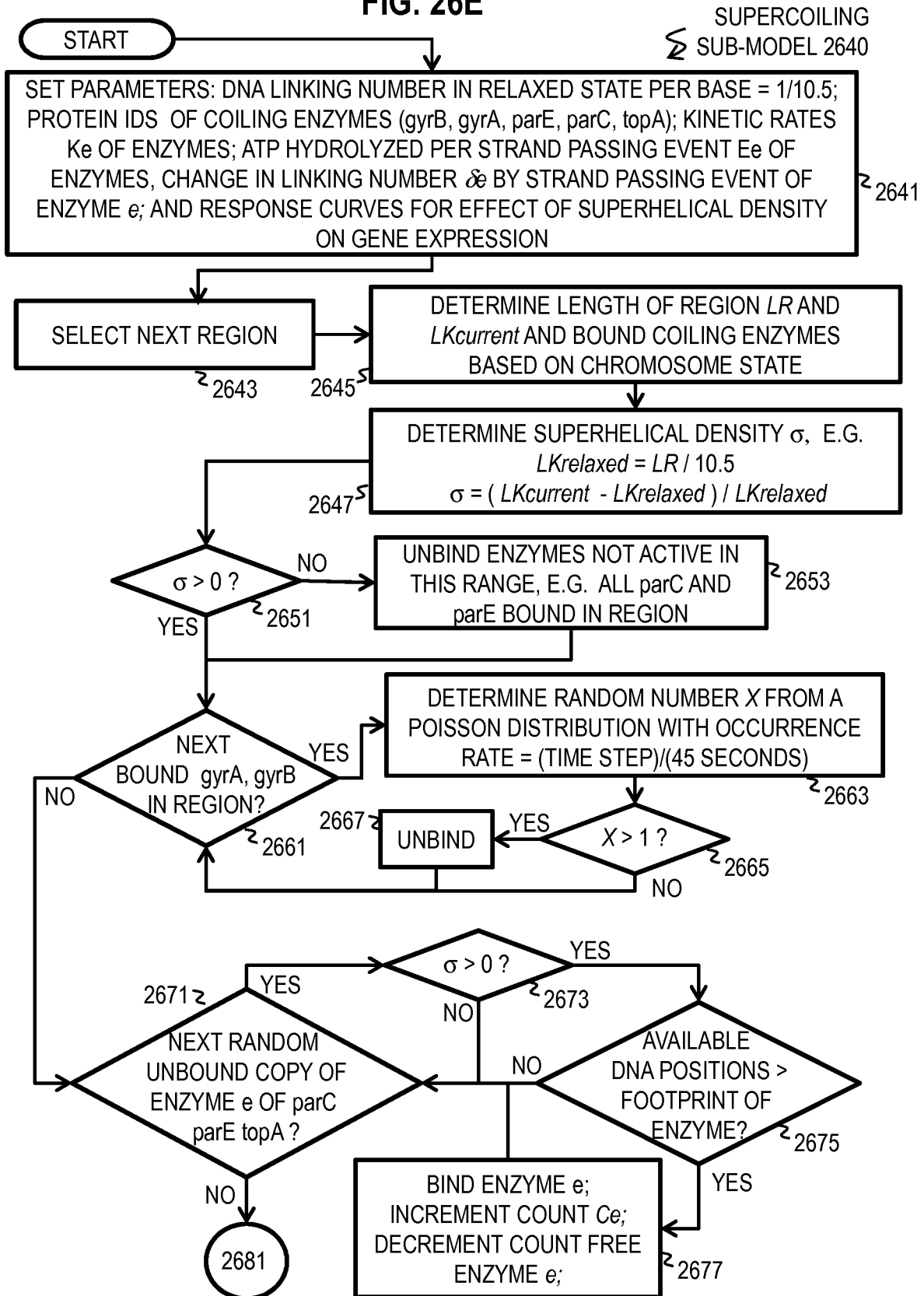


FIG. 26F

SUPERCOILING
 SUB-MODEL 2640
 (CONT)

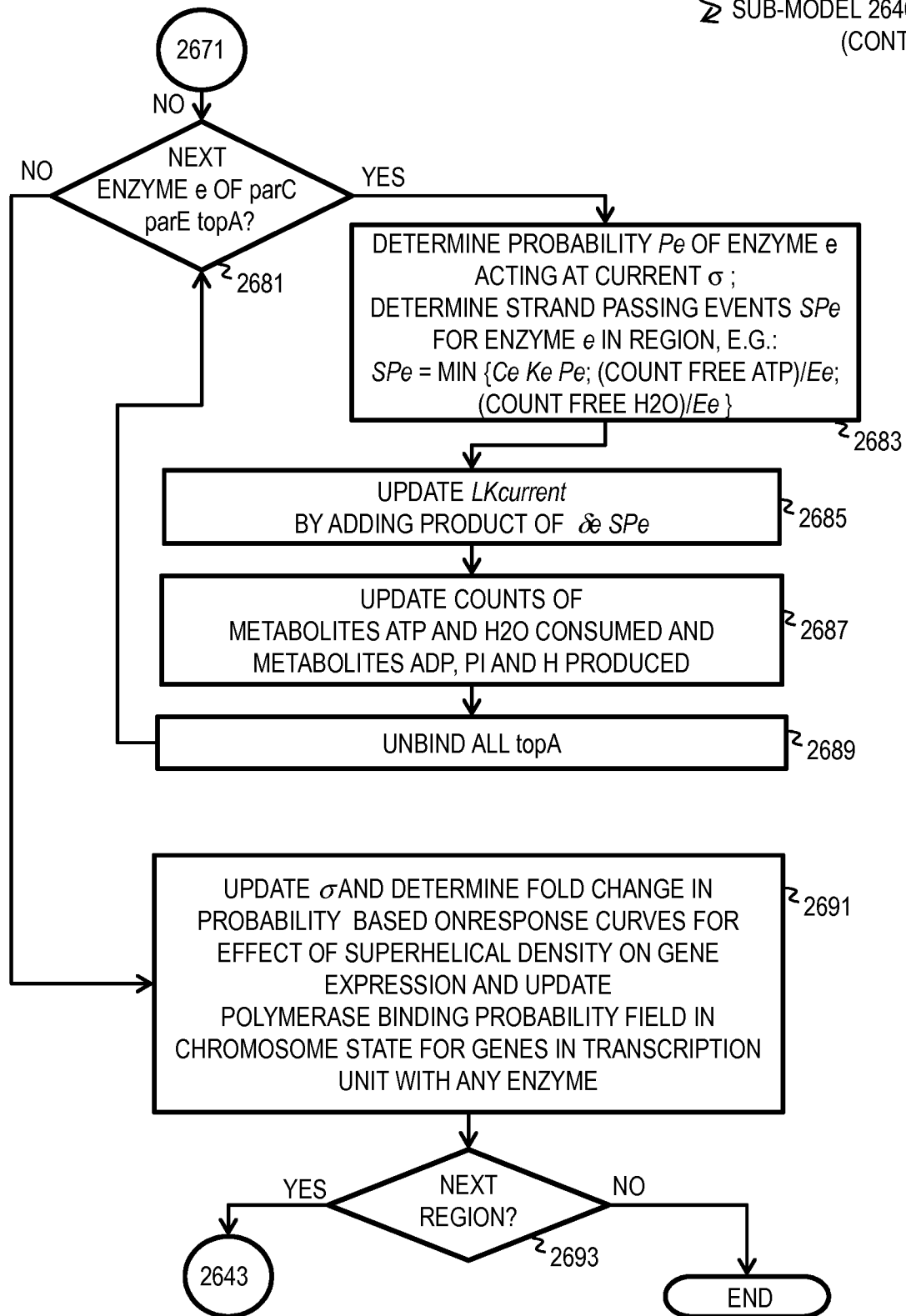


FIG. 26F

Σ SUPERCOILING
SUB-MODEL
PARAMETERS

Enzyme/Complex	Composition	Gene Name(s)	DNA Footprint (nt)
DNA gyrase	(2) MG003, (2) MG004	<i>gyrB</i> , <i>gyrA</i>	140
Topoisomerase IV	(2) MG203, (2) MG204	<i>parE</i> , <i>parC</i>	34
Topoisomerase I	(1) MG122	<i>topA</i>	19

Parameter	Value	Symbol
Supercoiling		
Bases per turn in relaxed DNA	10.5	
Equilibrium superhelical density	-0.06	σ_0
Supercoils introduced by 1 gyrase strand passing event	-2	δ_{gr}
Supercoils introduced by 1 topoisomerase I strand passing event	1	δ_{topI}
Supercoils introduced by 1 topoisomerase IV strand passing event	-2	δ_{topIV}
Strand passing events rate per gyrase	1.2 s^{-1}	k_{gr}
Strand passing events rate per topoisomerase I	1 s^{-1}	k_{topI}
Strand passing events rate per topoisomerase IV	2.5 s^{-1}	k_{topIV}
ATP hydrolyzed per gyrase-catalyzed strand passing event	2	E_{gr}
ATP hydrolyzed per topoisomerase I-catalyzed strand passing event	0	E_{topI}
ATP hydrolyzed per topoisomerase IV-catalyzed strand passing event	2	E_{topIV}
Superhelical density above which gyrase acts	-0.1	T_{gr}
Superhelical density above which topoisomerase I acts	0	T_{topI}
Superhelical density below which topoisomerase IV acts	0	T_{topIV}
Gyrase logistic function steepness parameter	100	L_{gr}
Topoisomerase I logistic function steepness parameter	-100	L_{topI}
Number of bases in the chromosome	580076	
Enzyme DNA footprints	See Tab. S11	
Processivity		
Mean time gyrase stays bound to the DNA	45 s	
Supercoiling's Effect on Gene Expression		
Slopes of linear fit of gyrase σ -gene expression data	42.8	
Slopes of linear fit of topoisomerase I σ -gene expression data	-7.4	
Slopes of linear fit of topoisomerase IV σ -gene expression data	1.1	
y-intercepts of linear fit of gyrase σ -gene expression data	3.57	
y-intercepts of linear fit of topoisomerase I σ -gene expression data	0.56	
y-intercepts of linear fit of topoisomerase IV σ -gene expression data	1.07	
σ bounds for applying linear fit of gene expression data	-0.08, 0.07	σ_{lower} , σ_{upper}

FIG. 27A

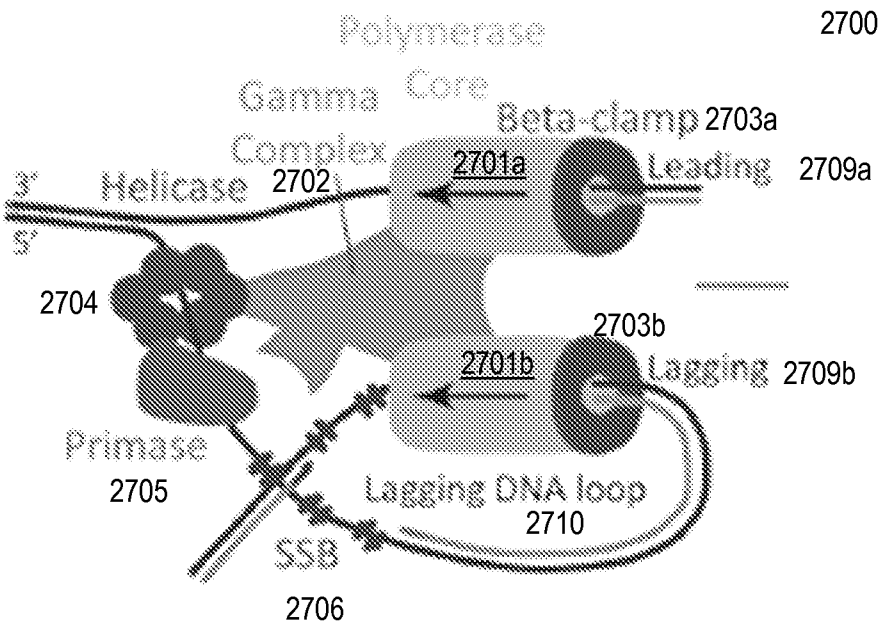


FIG. 27B

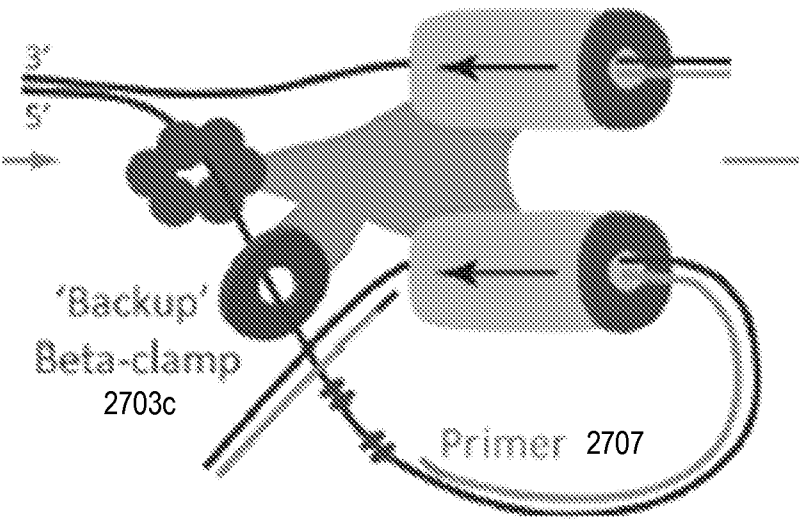


FIG. 27C

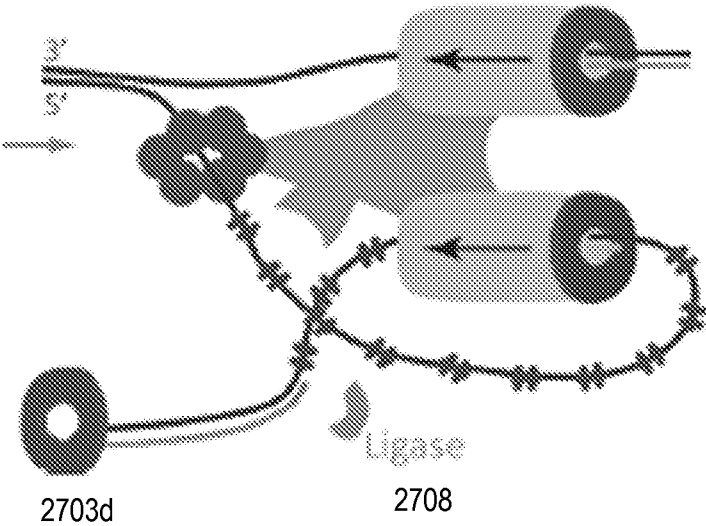
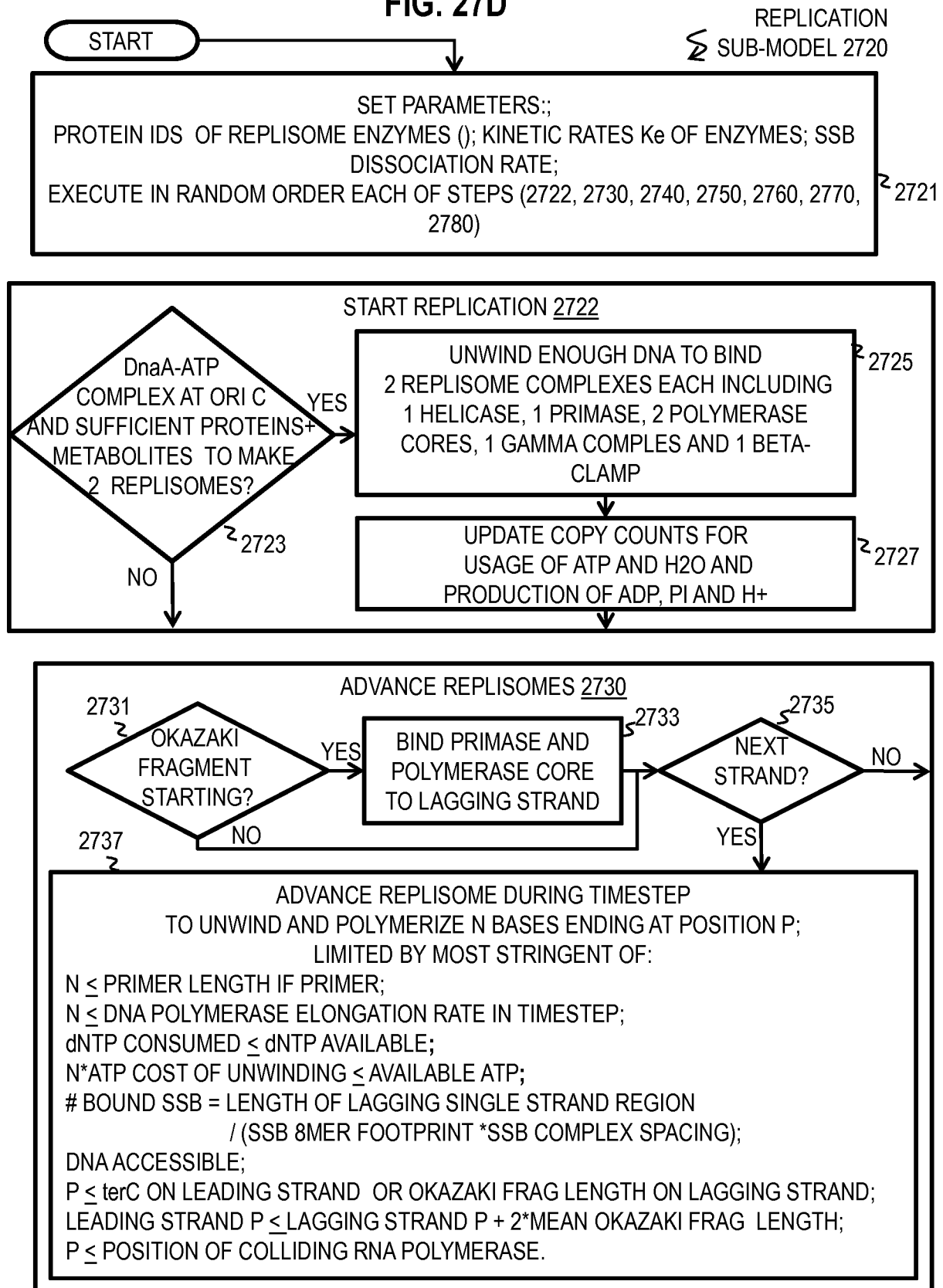


FIG. 27D



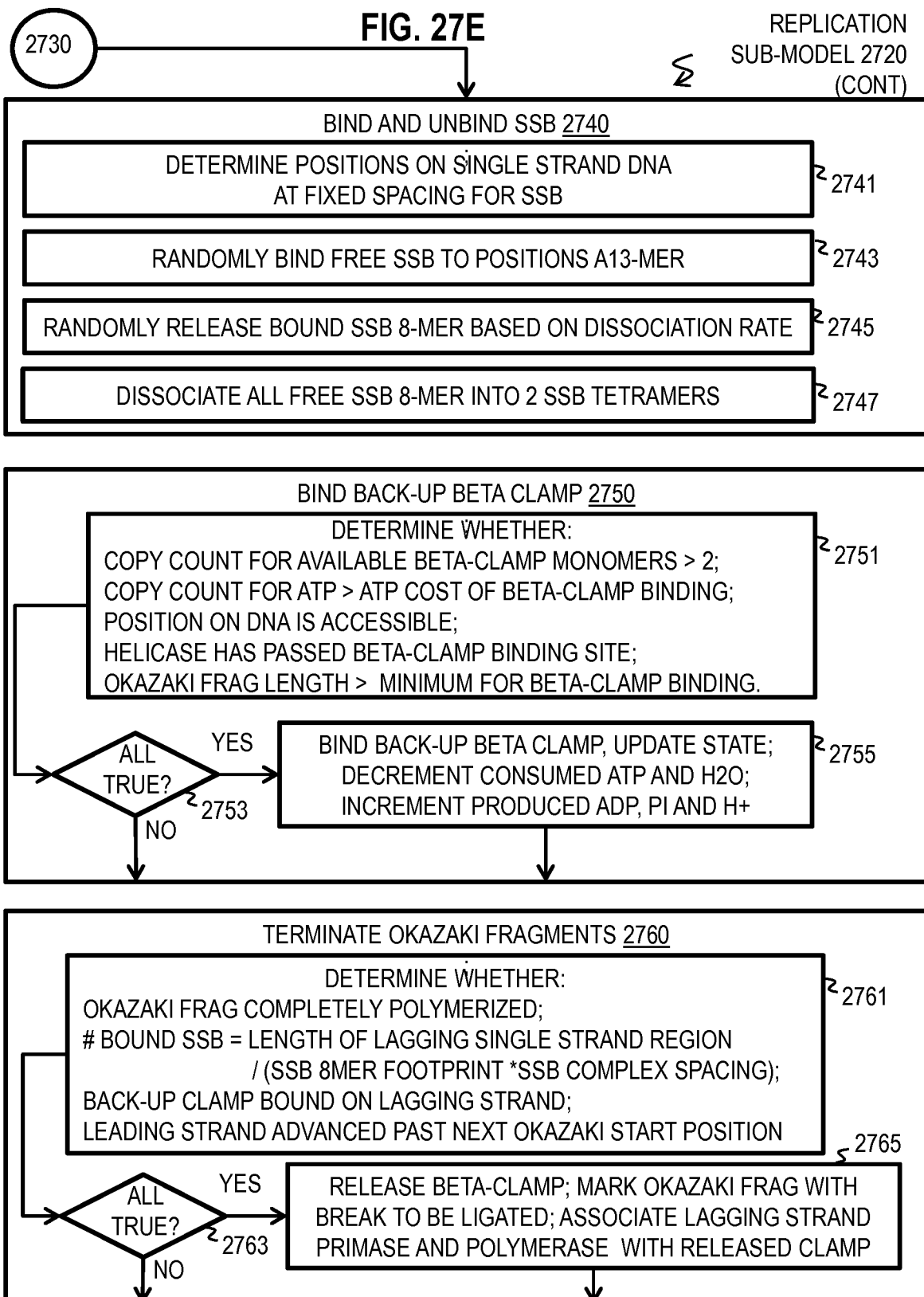


FIG. 27F

REPLICATION
SUB-MODEL 2720
(CONT)

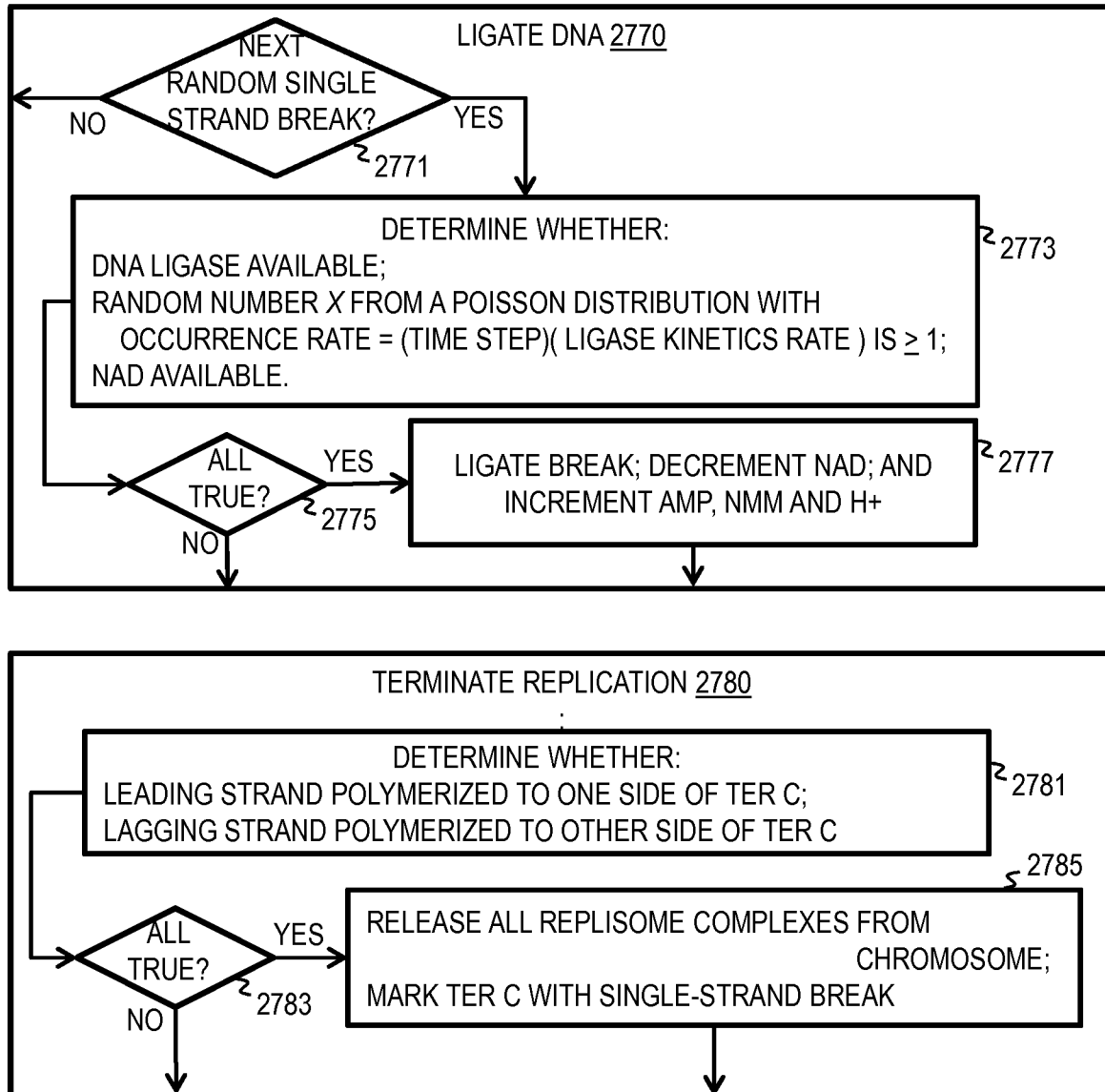


FIG. 27G

Enzymes/Complexes	Composition	Gene Name(s)	DNA Footprints (nt)
DNA helicase	(6) MG094	dnaB	20
DNA primase	(1) MG250	dnaG	14
β -clamp	(2) MG001	dnaN	25
DNA polymerase core	(1) MG031, (1) MG261	polC, polC-2	24
γ -complex	(1) MG007, (1) MG351, (4) MG419	holB, holA, dnaX	26
DNA ligase	(1) MG254	ligA	19
Single stranded binding protein (SSB) dimer	(8) MG091	ssb	145

Parameter	Value	Symbol
oriC position	Base: 1	oriC
TerC position	Base: 290038	terC
Primer length	11 nt	l_{pm}
DNA polymerase elongation rate	100 nt s^{-1}	k_{el}
SSB complex spacing	30 nt	n_{ssb}
Okazaki fragment mean length	1500 nt	l_{or}
Ligase kinetic rate	0.04 s^{-1}	k_{lg}
Current Okazaki fragment length before "back-up" beta-clamp can bind	750 nucleotide	l_{back}
Length of lagging strand loop at the start of Okazaki fragment polymerization	50 nt	l_{loop}
DNA polymerase stall time upon anti-direction collision with RNA polymerase	1.7 s	t_{stall}
SSB dissociation rate	0.3 s^{-1}	k_{dssb}
Chromosome length	580076	
Chromosome sequence	See [182]	
DNA footprints of Proteins	See 'Enzymes/Complexes'	
ATP cost of beta-clamp binding	1 ATP molecule	c_{ssb}
ATP cost of 1 base pair unwinding	1 ATP molecule	c_{el}
NAD requirement per ligation event	1 NAD molecule	c_{lg}

FIG. 27H

Connected States	Read from state	Written to state
Chromosome	• Whether a DnaA complex has formed at oriC • DNA-bound protein locations • Superhelicity • DNA strand breaks to be ligated • DNA sequence • DNA footprints of proteins • Chromosome regions accessible for protein binding • Damaged DNA bases • OriC position • TerC position • Sequence Length	• Polymerized regions of DNA • DNA-bound protein locations • Unwound bases (Effect on superhelicity) • DNA strand breaks to be ligated

FIG. 27I

1. Initiate Replication

- If a 29-mer DnaA-ATP complex exists at the oriC,
 - If sufficient proteins and metabolites exist to make 2 replisomes,
 - Unwind enough DNA bases (b_{unwound}) to bind 2 sets of replication machinery. One set binds on either side of the oriC. All proteins are bound to the leading strand of DNA. Each set includes:
 - 1 helicase
 - 1 primase
 - 2 polymerase cores
 - 1 gamma complex
 - 1 beta-clamp
 - Account for total ATP usage (as well as H_2O usage, and ADP, Pi, and H^+ production):

$$\text{ATP usage} = 2c_{\text{dnaa}} + c_{\text{hel}}b_{\text{unwound}}$$

- Note: following the initiation of replication, the dissociation of the DnaA complex is handled by the Chromosome state class.

2. Advance replisomes

- If an Okazaki fragment is starting at the current timestep,
 - Bind a primase and a polymerase core to the lagging strand
- Model the unwinding and polymerizing of the DNA. The leading and lagging polymerases and helicases are advanced only so far as the following conditions are met:
 Let N = the number of polymerized/unwound bases on a given strand in the given timestep.
 Let P be the last polymerized position. P varies from 0 (at the oriC) to 290038 (at the terC).

FIG. 27J

- $N \leq l_{\text{prim}}$ if a primer is being synthesized
- $N \leq k_{\text{el}}$
- Available dNTPs $\leq$ dNTP requirements
- Available ATP $\leq N e_{\text{hel}}$
- # Bound SSBs $= \frac{\text{Length of lagging single strand region}}{\text{SSB 8mer footprint} \times s_{\text{ssb}}}$
- Upstream DNA regions accessible to replisome proteins (no DNA damage, non-displacable proteins)
- Leading strand $P \leq \text{terC}$, Lagging strand $P \leq \text{Okazaki fragment length}$
- Leading strand $P \leq 2L_{\text{OV}} + \text{lagging strand } P$
- $P \leq$ position of colliding RNA polymerase

3. Bind and Unbind SSBs

SSBs nucleate as a tetramer, binding single-stranded DNA around them. These tetramers generally bind in sets of two.

- For all single-stranded stretches of DNA,
 - Randomly bind SSBs (as 8mers) to deterministically selected position (with fixed spacing: s_{ssb})
- Randomly release SSB 8mers according to the dissociation rate (k_{dssb})
- Dissociate all free 8mer SSBs into two tetramer SSBs

4. Bind "Back-up" Beta-Clamp

The formation of a new Okazaki fragment requires the formation of a new lagging strand DNA loop. To prepare for this event, while the current Okazaki fragment is being polymerized, a "back-up" beta-clamp is bound just downstream of the start position of the next Okazaki fragment. The beta-clamp assembles as a dimer on the chromosome, and the binding event requires hydrolysis of one ATP molecule to ADP (e_{beta}).

- If $\left\{ \begin{array}{l} \# \text{ available beta-clamp monomers} > 2 \\ \# \text{ available ATP} > e_{\text{beta}} \end{array} \right.$
 - Position on the DNA is accessible
 - Helicase has passed beta-clamp binding site
 - Okazaki fragment length $> l_{\text{beta}}$
 - Bind "back-up" beta-clamp
 - Decrement ATP, H_2O , and increment ADP, Pi, and H^+

FIG. 27K

5. Terminate Okazaki fragments

When an Okazaki fragment has been completely polymerized, its beta-clamp is released, and the end of the Okazaki fragment is marked as having a single-strand break to be ligated by DNA ligase. The lagging strand primase and polymerase associate with the “back-up” beta clamp that was previous bound. This forms a new lagging strand DNA loop.

- If
 - Okazaki fragment has been completely polymerized
 - Number of bound SSBs = $\frac{\text{Length of lagging single strand region}}{\text{SSB 8mer footprint} \times s_{ssb}}$
 - A “back-up” clamp has been bound on the lagging strand
 - The replication machinery on the leading strand has advanced beyond the Okazaki start site
- Release beta-clamp
- Mark Okazaki fragment as having a single-strand break to be ligated by DNA ligase
- Associate lagging strand primase and polymerase with the “back-up” beta clamp that was previously bound (This forms a new lagging strand DNA loop.)

6. Ligate DNA

- If single-strand breaks exist (usually between polymerized Okazaki fragments),
 - Ligate these breaks in random order up to the limits of:
 - DNA ligase availability
 - Ligase kinetics (k_{lig})
 - NAD availability
 - Decrement NAD, and increment AMP, NMM, and H^+

7. Terminate Replication

- If leading and lagging strands to both sides of the *terC* have been completely polymerized,
 - Release all replication machinery from the chromosome
 - Mark *terC* as having a single-strand break to be ligated by DNA ligase

FIG. 28A

Parameter	Value	Symbol
Positions of all the DnaA binding sites on the chromosome(s)	(see Tab. S45)	
Factor by which DnaA-ATP to oriC site binding probability increases when other sites are bound	85.6909	C_{site}
Factor by which DnaA-ATP to oriC site binding probability increases when x*4 sized DnaA complex has formed at oriC	1	C_{state}
Rate for DnaA-ATP binding high affinity DnaA boxes	$25\text{ nM}^{-1}\text{ h}^{-1}$	k_{b1ATP}
Rate for DnaA-ATP binding medium affinity DnaA boxes	$0.6\text{ nM}^{-1}\text{ h}^{-1}$	k_{b2ATP}
Rate for DnaA-ATP dissociating from the DNA	20 h^{-1}	k_{d1ATP}
Rate for DnaA-ADP binding high affinity DnaA boxes	$2.5\text{ nM}^{-1}\text{ h}^{-1}$	k_{b1ADP}
Rate for DnaA-ADP binding medium affinity DnaA boxes	$0.61\text{ nM}^{-1}\text{ h}^{-1}$	k_{b2ADP}
Rate for DnaA-ADP dissociating from the DNA	20 h^{-1}	k_{d1ADP}
Rate for DnaA-ADP to DnaA-ATP regeneration	2.3 h^{-1}	k_{Regen}
Rate for DnaA-ADP to DnaA-ATP regeneration catalyzed by membrane lipids	0.018 g L^{-1}	$K_{\text{Regen_P4}}$

FIG. 28B

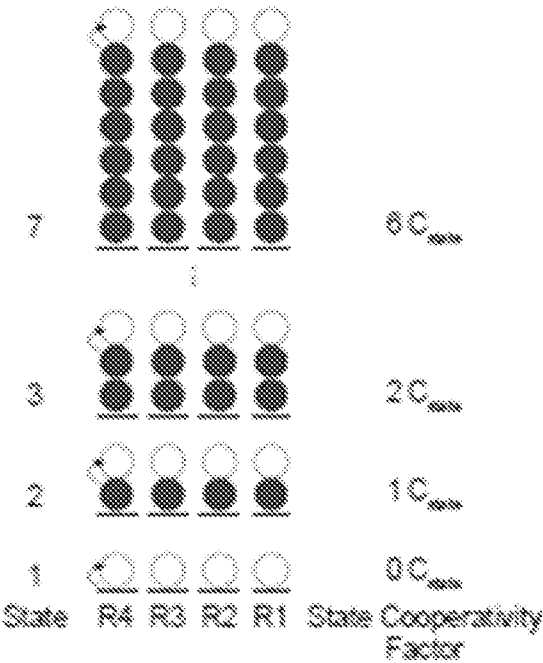


FIG. 28C

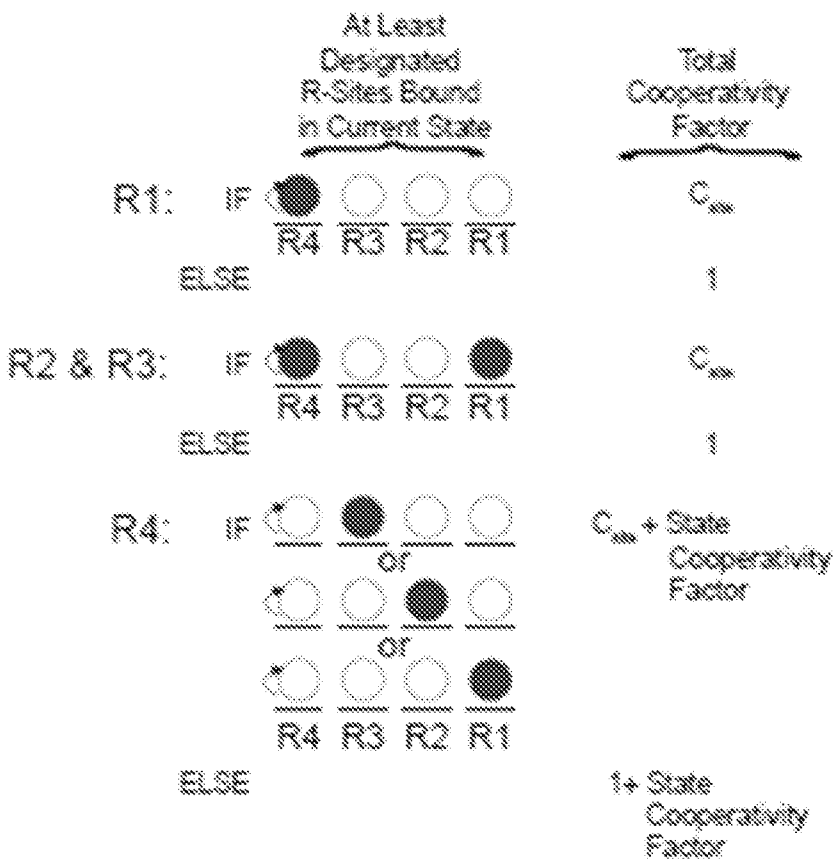


FIG. 28D

Connected States	Read from State	Written to State
Chromosome	• Positions on chromosome(s) where DnaA molecules are bound • Accessible DnaA binding sites • Copy numbers of DnaA binding sites	• Updated positions on chromosome(s) where DnaA molecules are bound
Mass	• Membrane mass (used to calculate membrane concentration)	
Geometry	• Cell volume (used to calculate membrane concentration)	

FIG. 29A

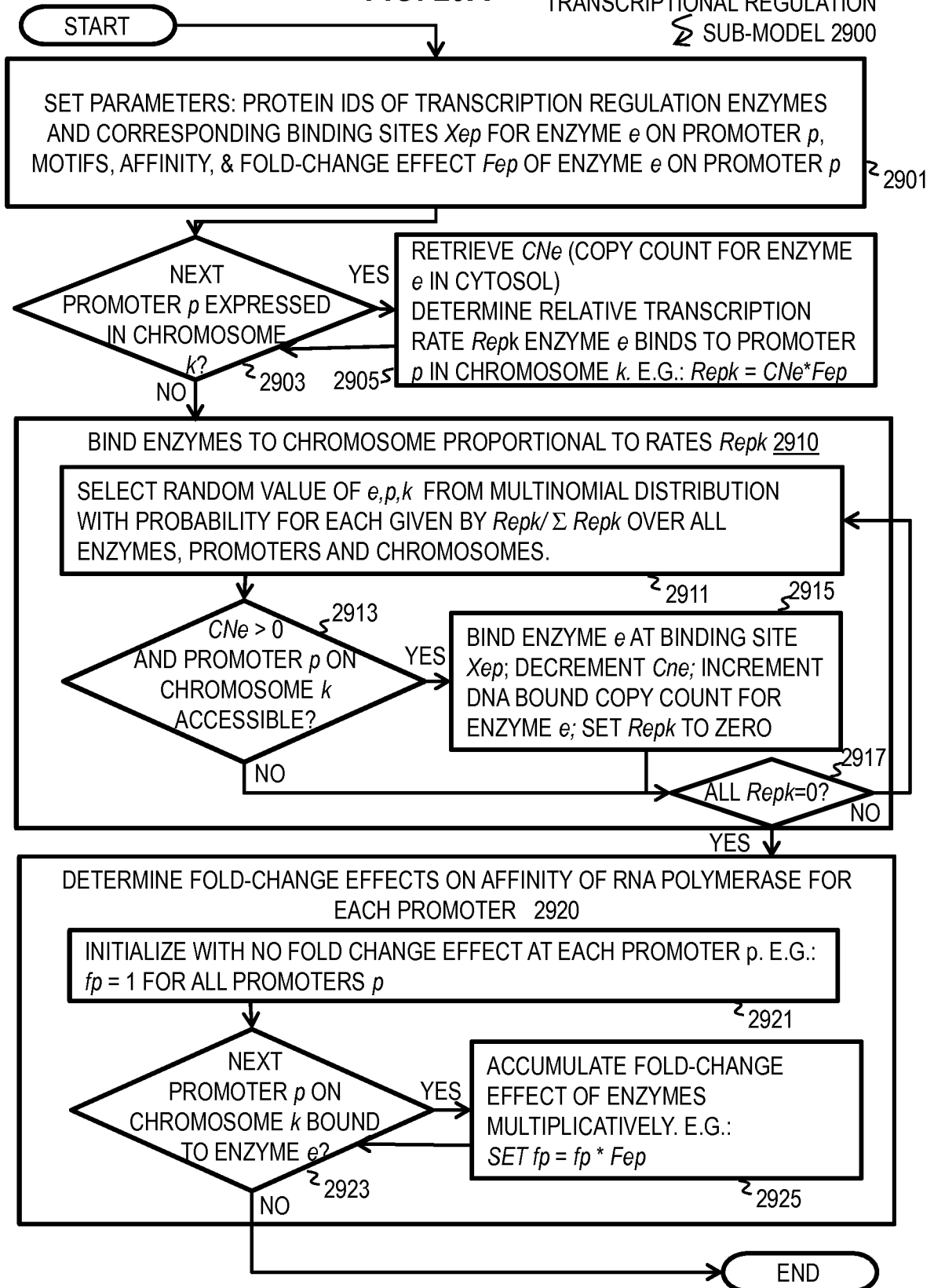
TRANSCRIPTIONAL REGULATION
Σ SUB-MODEL 2900

FIG. 29B

Input: n_{ik} is true if promoter i is expressed in chromosome k
 Input: $p_{c,i}$ free cytosolic copy number of transcriptional regulator i
 Input: $p_{b,i}$ DNA-bound copy number of transcriptional regulator i
 Input: x_{ij} Binding site of transcriptional regulator i at promoter j
 Input: F_{ij} fold-change effect of transcriptional regulator i on promoter j
 Input: b_{ijk}^m, b_{ijk}^c chromosomal protein occupancy as defined in Tab. S2
 Output: f_i fold-change effect of transcriptional regulation on RNA polymerase affinity for promoter i

Calculate the relative rate, r_{ijk} , transcriptional regulator i binds promoter j of chromosome k :

foreach DNA-binding transcriptional regulator i in promoter j of chromosome $k = \{1,2\}$ do
 $r_{ijk} \leftarrow n_{ik} p_{c,i} F_{ij}$

Bind transcriptional regulators to the chromosome:

repeat
 Select regulator i , promoter j , and chromosome $k \sim \text{multinomialRand}(1, r_{ijk} / \sum_{ijk} r_{ijk})$
 If regulator i expressed ($p_{c,i} > 0$) and isRegionAccessible(promoter j of chromosome k to regulator i) then
 Bind protein to chromosome: $b_{ijk}^m = 1 \forall y \in \{x_{ij}, x_{ij} + l_i - 1\}$, where $z = m$ for monomers and c for complexes
 Update free and bound copy numbers: $p_{c,i} \leftarrow p_{c,i} - 1, p_{b,i} \leftarrow p_{b,i} + 1$
 Update binding rate: $r_{ijk} \leftarrow 0$

until no additional transcriptional regulator can bind DNA ($r_{ij} = 0 \forall i, j$)

Calculate the fold-change effect of transcriptional regulators on the affinity of RNA polymerase for each promoter

Initialize fold-change effects: $f_i \leftarrow 1 \forall i$

foreach promoter j of chromosome $k = \{1,2\}$ bound by DNA-binding transcriptional regulator i do
 Add fold-change effects multiplicatively: $f_j \leftarrow f_j F_{ij}$

foreach promoter j regulated by an expressed non-DNA-binding transcriptional regulator i ($p_{c,i} > 0$) do
 $f_j \leftarrow f_j F_{ij}$

FIG. 30A

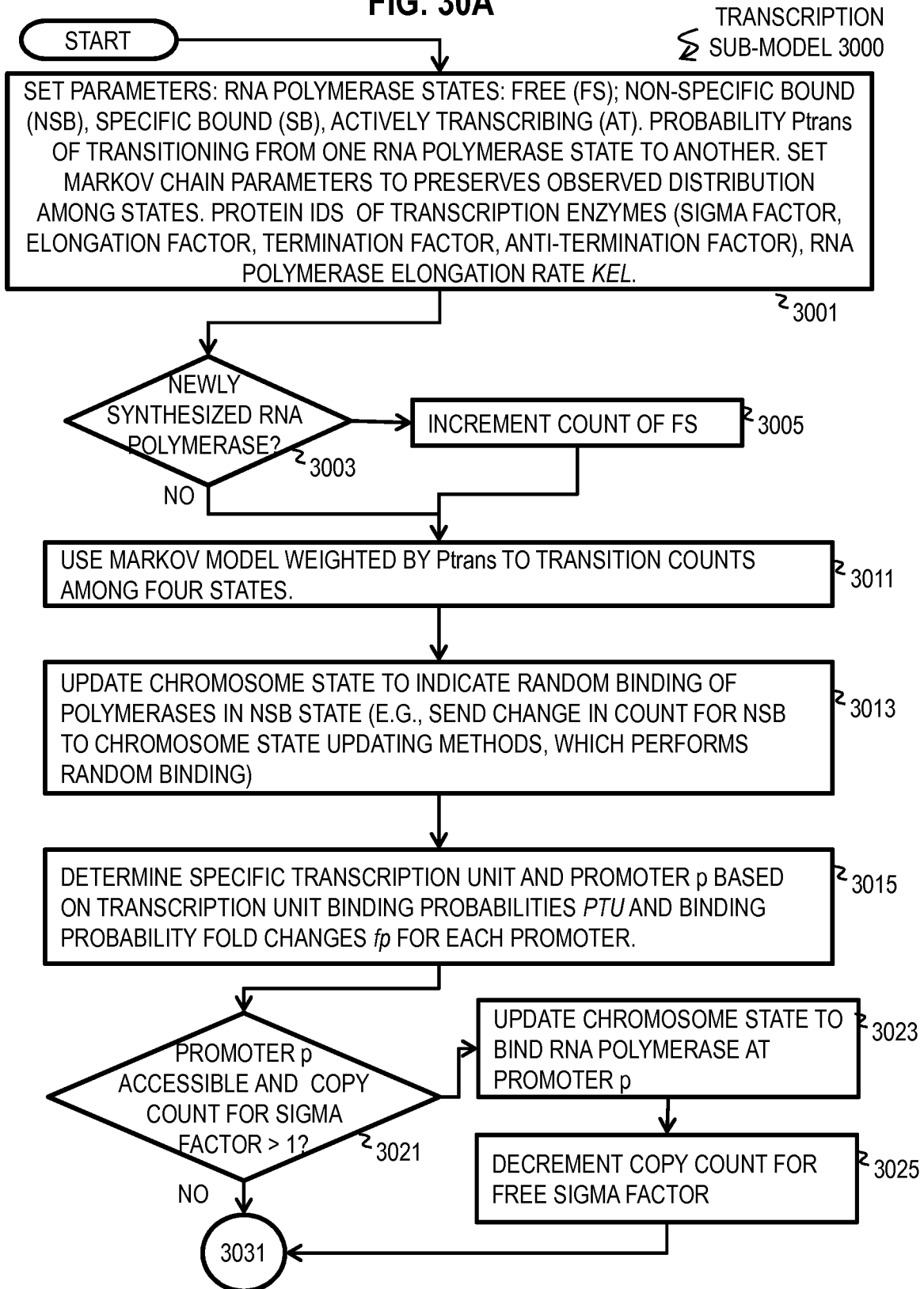


FIG. 30B

TRANSCRIPTION
Σ SUB-MODEL 3000
(CONT)

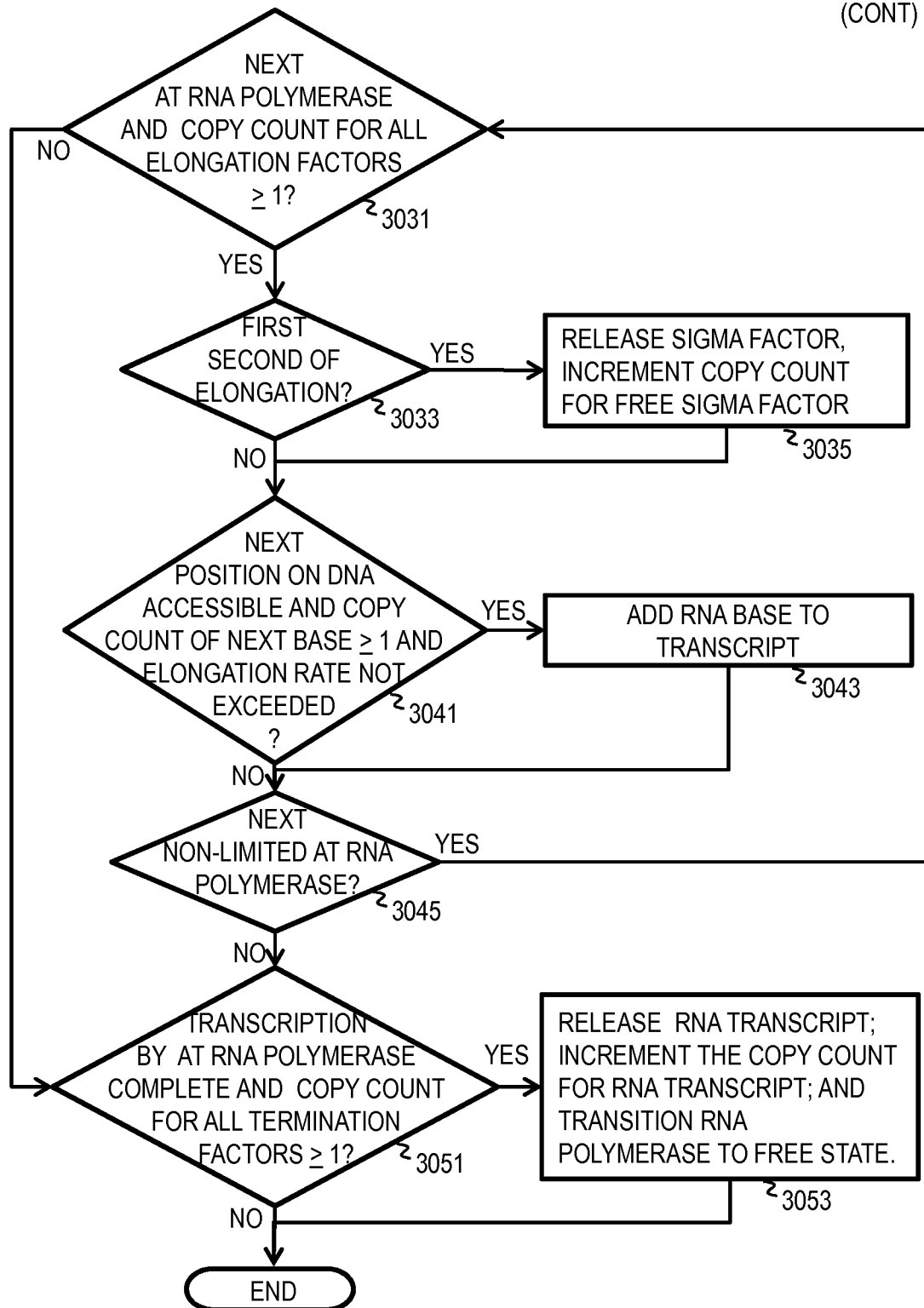


FIG. 30C

Connected States	Read from state	Written to state
Chromosome	- Regions accessible for transcription machinery to bind - Polymerized regions of DNA	Positions and strands on the DNA where transcription machinery is bound
Rna	Counts of RNA species	Updated counts of RNA species
RNA Polymerase	- RNA polymerase states (FS, NSB, SB, AT) - DNA positions and strands of DNA bound RNA polymerases - Expectations of RNA polymerases in FS, NSB, SB, and AT states (used to derive transition Probabilities P_{trans})	- Updated states of RNA polymerases - Updated positions and strands of DNA bound RNA polymerases
Transcript	- Transcription unit indexes of bound RNA polymerases - RNA polymerase position within transcript - Chromosome on which RNA polymerase resides (1 or 2) - Transcription unit sequences, directions, lengths, and 5' start coordinates	- Updated transcription unit indexes of bound RNA polymerases - Updated RNA polymerase position

FIG. 30D

Enzymes/Complexes	Composition	Gene Name(s)	DNA (bases)
RNA polymerase sigma factor	(1) MG249	rpoD	58
Transcription elongation factor	(1) MG282	greA	
Transcription termination factor	(1) MG141	nusA	
Transcription termination/antitermination protein	(1) MG027	nusB	
DNA-directed RNA Polymerase	(1) MG022, (2) MG177, (1) MG340, (1) MG341	rpoE, rpoA, rpoC, rpoB	75

Parameters	Value	Symbol
RNA polymerase elongation rate	50 nt s ⁻¹	k _d
Transcription unit binding probabilities		P _{1n}
RNA polymerase state transition probabilities		P _{trans}
Enzyme DNA footprints	See 'Enzymes/complexes'	
Transcription unit sequences	See Tab. S44, [182]	
Transcription unit directions	See Tab. S44	
Transcription unit lengths	See Tab. S44	
Transcription unit 5' coordinates	See Tab. S44	
Cell Cycle Length	8.9 h	τ

FIG. 31A

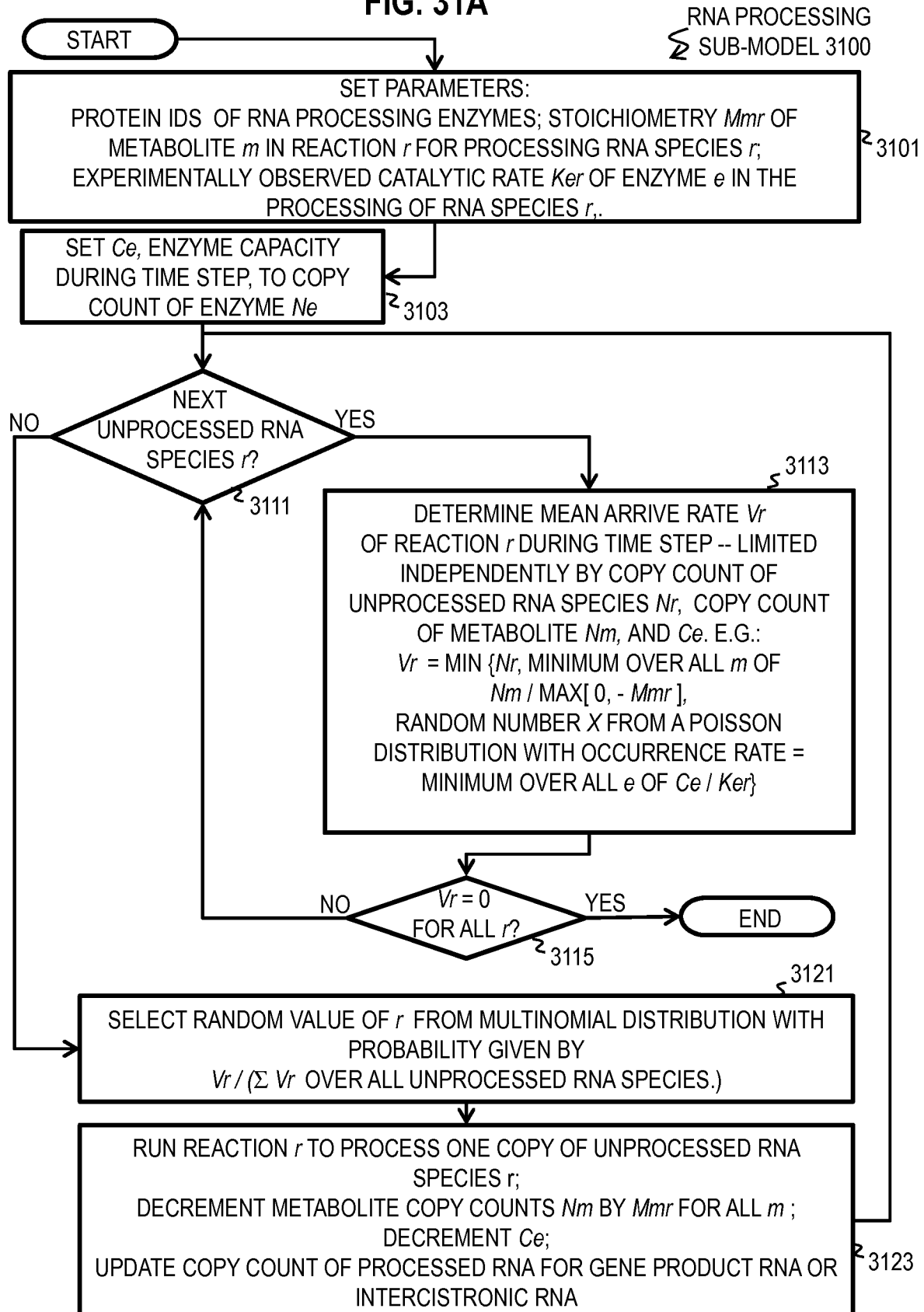


FIG. 31B

Input: r_i^p copy number of processed RNA species i
 Input: r_i^f copy number of intercistronic fragment i
 Input: R_{ij}^g is one if operonic RNA i contains gene j , and zero otherwise
 Input: R_{ij}^f is one if operonic RNA i contains intercistronic fragment j , and zero otherwise
 Let $k_i \leftarrow c_i \Delta t$ be the capacity of enzyme i for RNA processing
 repeat
 Calculate cleavage rates
 foreach operonic non-coding RNA species i do
 ┌ Calculate v_i according to Eq. 16
 Select operonic non-coding RNA species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
 Update RNA copy numbers: $r_i^g \leftarrow r_i^g - 1$, $r^p \leftarrow r^p + R_{\bullet, i}^g$, $r^f \leftarrow r^f + R_{\bullet, i}^f$
 Update metabolites: $m \leftarrow m - M_{\bullet, i}$
 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet, i}$
 until no further cleavage possible ($v_i = 0 \forall i$)

FIG. 32A

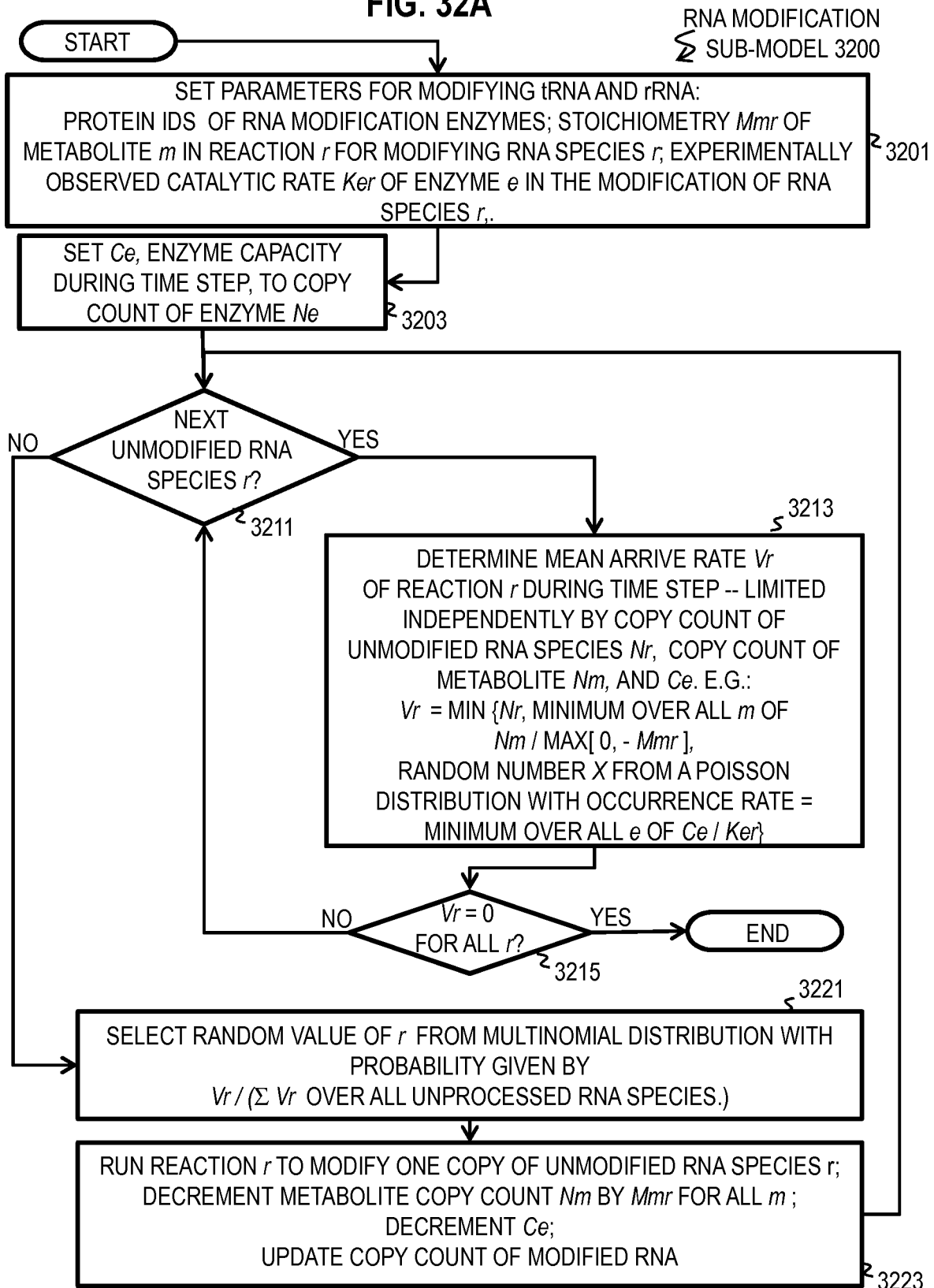


FIG. 32A

Input: r_i^u copy number of modified RNA species i

Let $k_i \leftarrow c_i \Delta t$ be the capacity of enzyme i for RNA modification

repeat

- Calculate modification rates
- foreach non-coding RNA species i do
 - Calculate v_i according to Eq. 17
- Select non-coding RNA species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
- Update RNA copy numbers: $r_i^u \leftarrow r_i^u - 1$, $r_i^m \leftarrow r_i^m + 1$
- Update metabolites: $m \leftarrow m - M_{\bullet, i}$
- Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet, i}$

until no further modification possible ($v_i = 0 \forall i$)

FIG. 33A

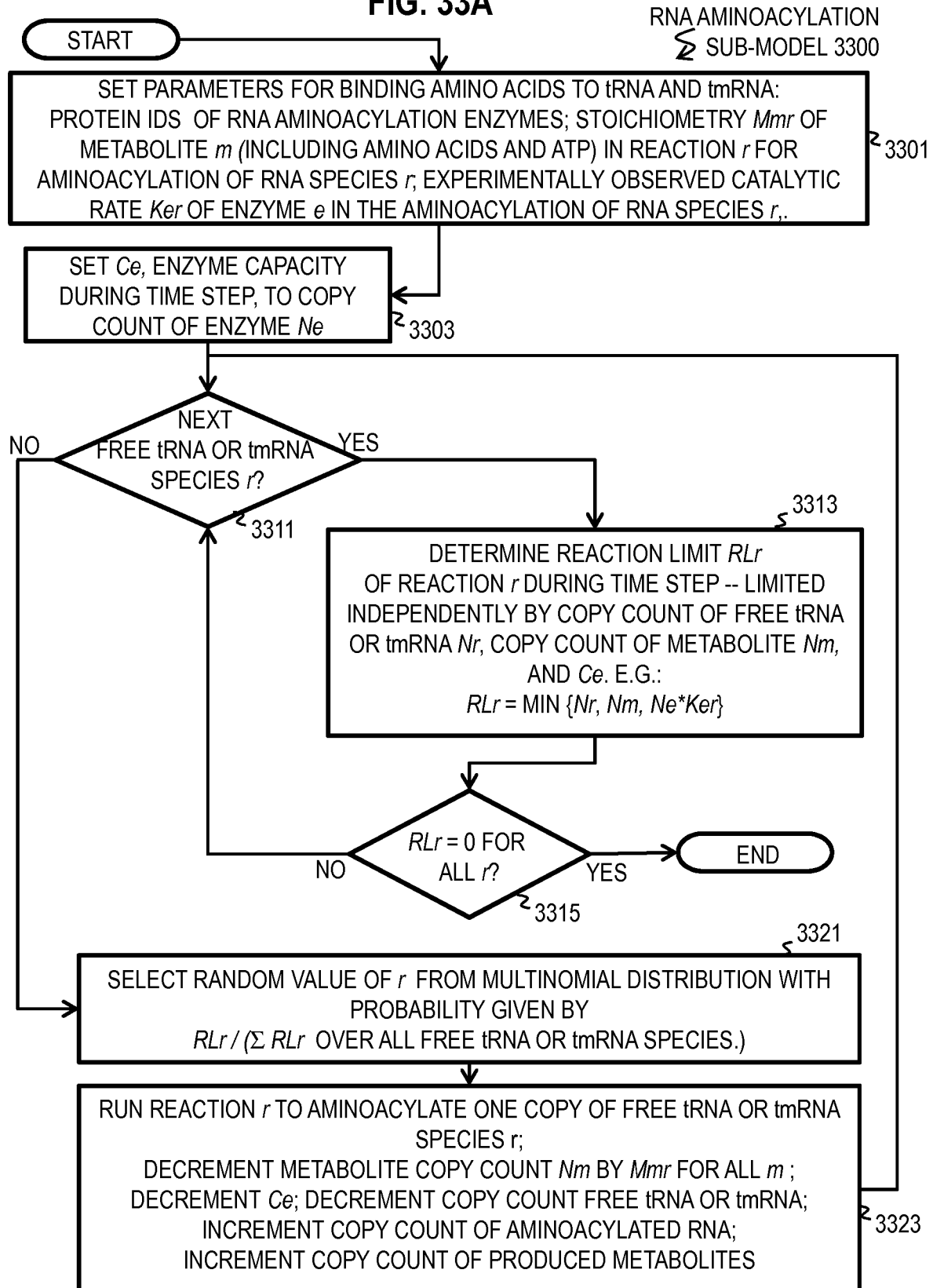


FIG. 33B

Enzymes/Complexes	Composition	Gene Name(s)
Alanyl-tRNA synthetase	(4) MG292	alaS
Arginyl-tRNA synthetase	(1) MG378	argS
Aspartyl-tRNA synthetase	(2) MG036	aspS
Asparaginyl-tRNA synthetase	(2) MG113	asnS
Cysteiny-tRNA synthetase	(1) MG253	cysS
Glutamyl-tRNA synthetase	(1) MG462	gltX
Glycyl-tRNA synthetase	(2) MG251	glyS
Histidyl-tRNA synthetase	(2) MG035	hisS
Isoleucyl-tRNA synthetase	(1) MG345	ileS
Leucyl-tRNA synthetase	(1) MG266	leuS
Lysyl-tRNA synthetase	(2) MG136	lysS
Methionyl-tRNA synthetase	(2) MG021	metG
Phenylalanyl-tRNA synthetase	(2) MG194, (2) MG195	pheS, -
Prolyl-tRNA synthetase	(2) MG283	proS
Seryl-tRNA synthetase	(2) MG005	serS
Threonyl-tRNA synthetase	(2) MG375	thrS
Tryptophanyl-tRNA synthetase	(2) MG126	trpS
Tyrosyl-tRNA synthetase	(2) MG455	tyrS
Valyl-tRNA synthetase	(1) MG334	valS
Glutamyl-tRNA(Gln) amidotransferase	(1) MG098, (1) MG099, (1) MG100	-, -, gatB
Methionyl-tRNA formyltransferase	(1) MG365	

Parameters
Free metabolites required for each reaction
tRNA aminoacylated by each reaction
Enzymes required to aminoacylate each tRNA
k_{cat} of the catalyzing enzyme of each aminoacylation reaction (s^{-1})

FIG. 34A

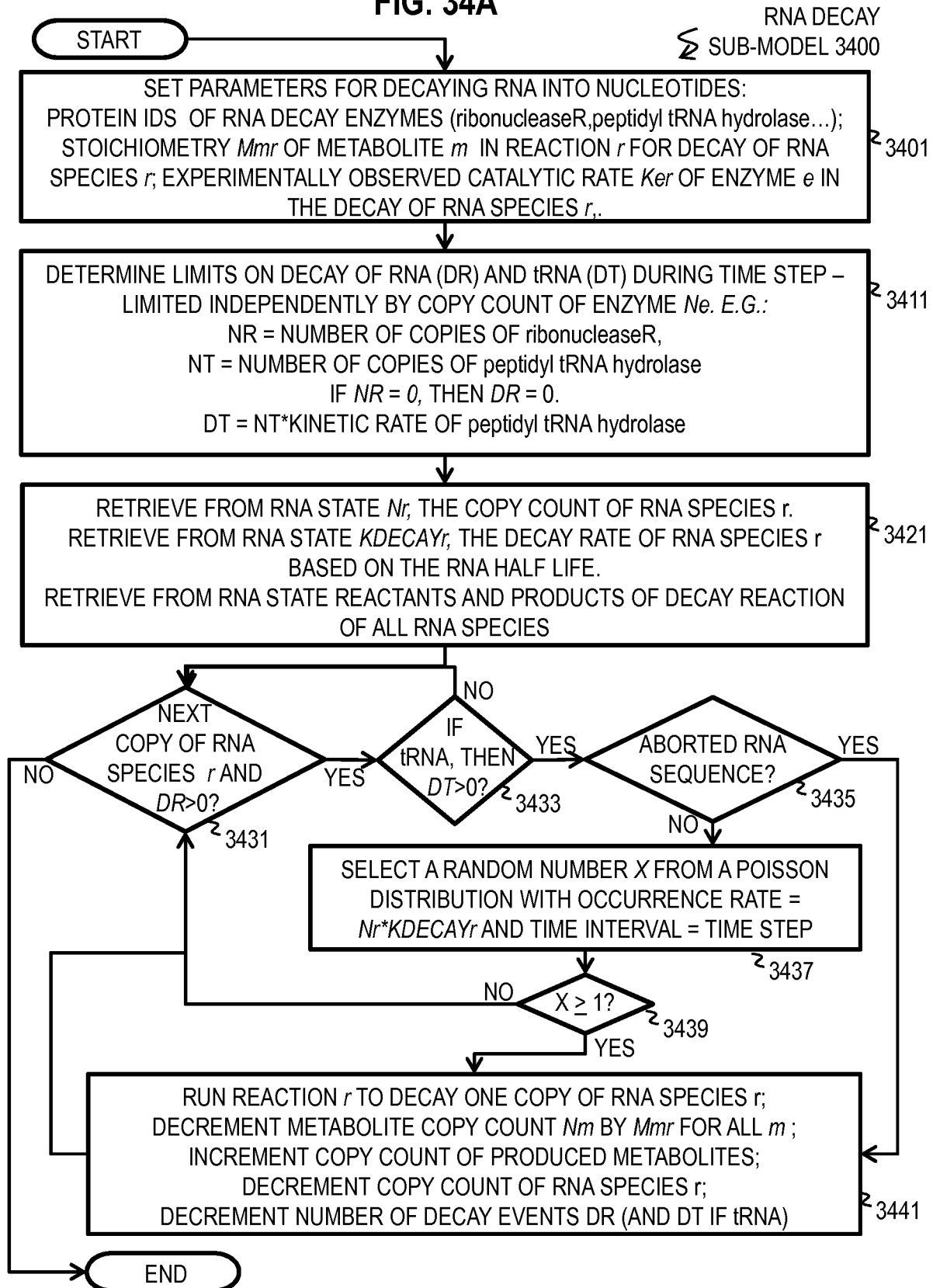


FIG. 34B

Connected State	Read from state	Written to state
Rna	* Count of each RNA species	* Updated count of each RNA species
	* Decay Rates of all RNAs	
Transcript	* Aborted transcript sequences	

FIG. 34C

Parameter	Value	Symbol
Rate of peptidyl tRNA hydrolase activity	0.7 s ⁻¹	k _{hyd}
Half lives of all RNA species	See Tab. S58	t _{1/2}
Decay rates of all RNA species		k _{decay}
Reactants and products of decay reactions of all RNAs		M _{decay}

FIG. 35A

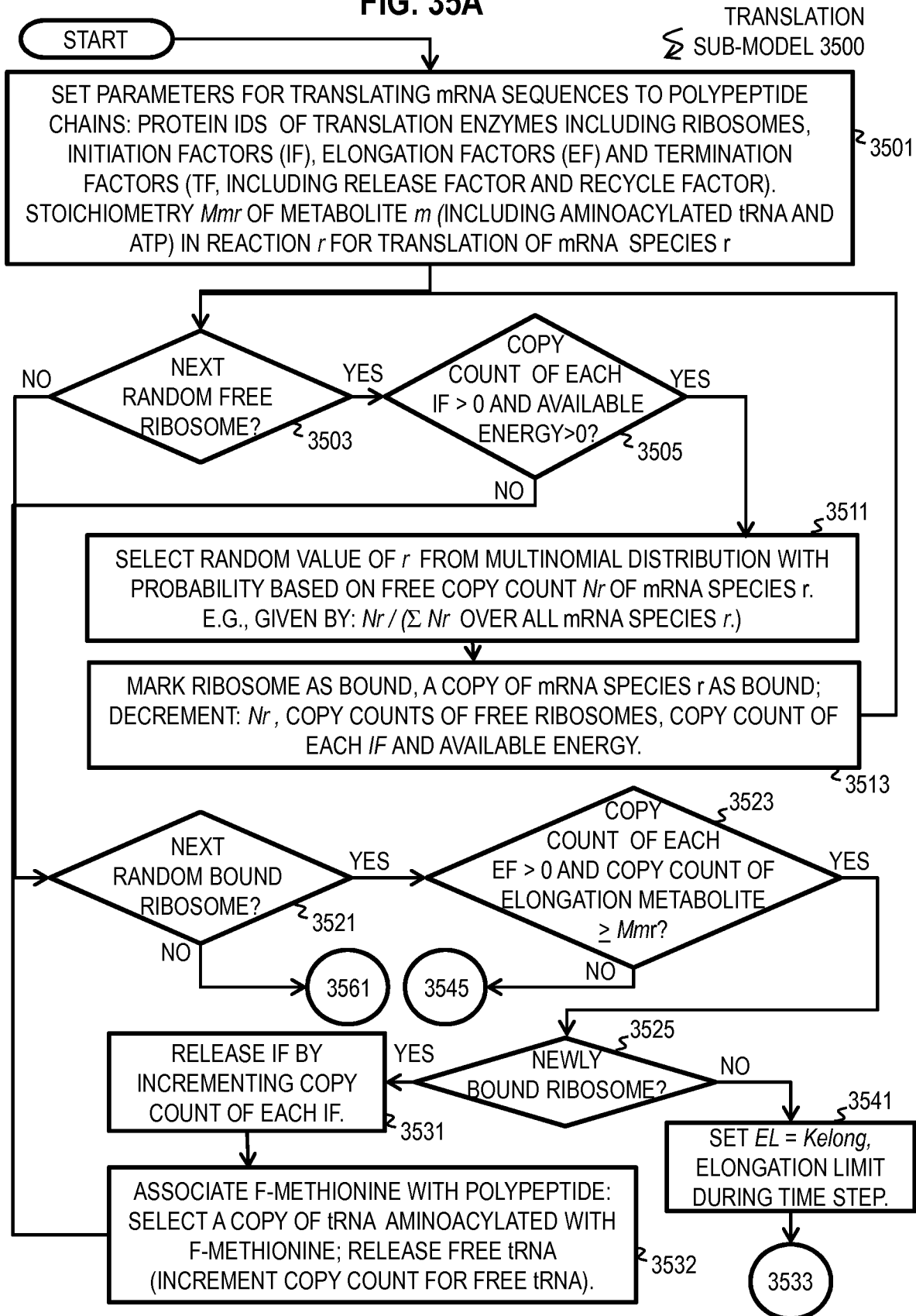


FIG. 35B

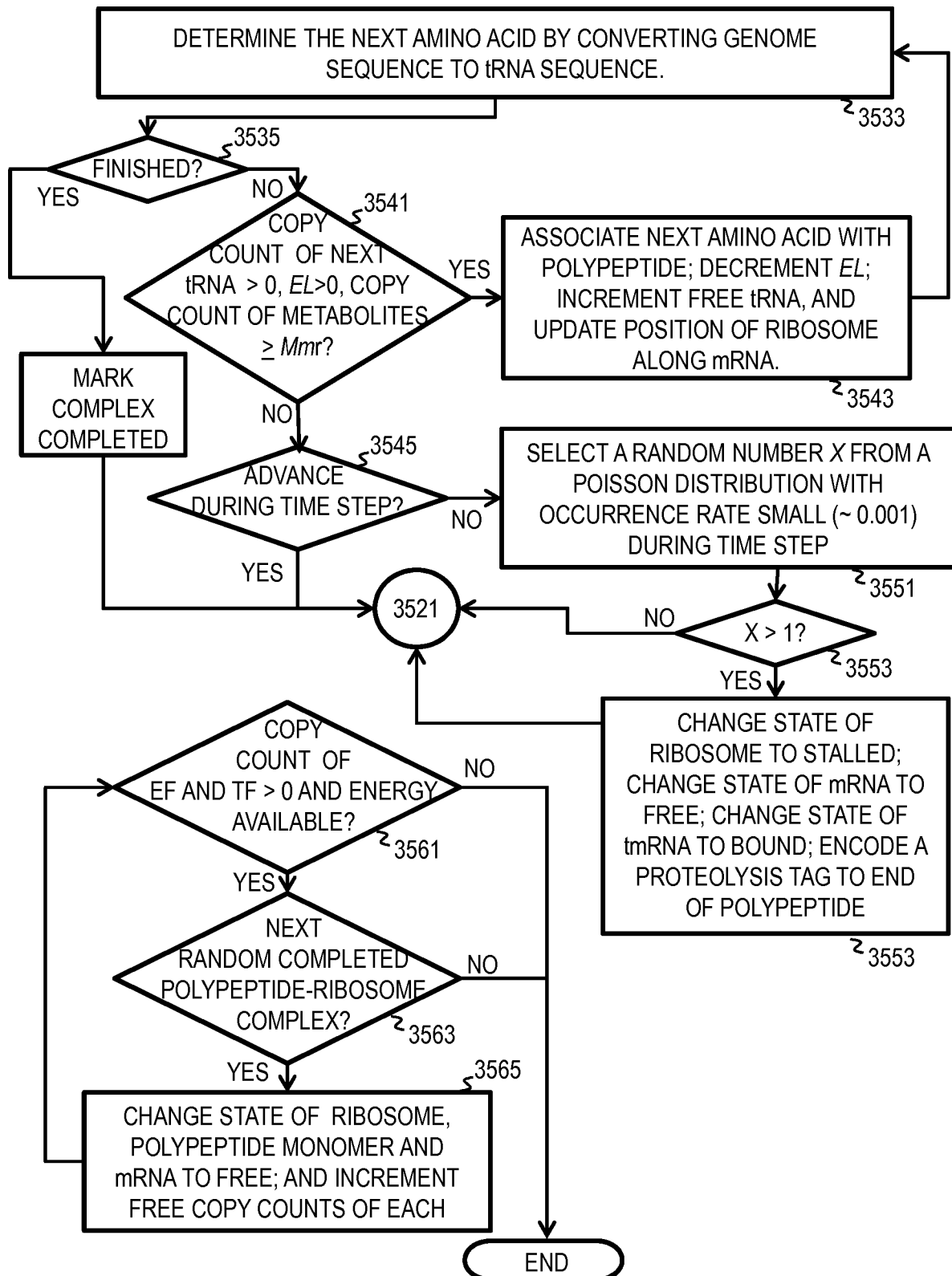


FIG. 35C

Connected State	Read from state	Written to state
Polypeptide	* List of ribosome-bound mRNAs * Lengths of nascent polypeptides (proteolysis tags) * tRNA sequences of protein monomers and proteolysis tags	* Updated list of ribosome-bound mRNAs * Updated lengths of nascent polypeptides (proteolysis tags)
Ribosome	* List of ribosome-bound mRNAs * Lengths of nascent polypeptides (proteolysis tags) * State of each ribosome: free, actively translating, or stalled	* Updated list of ribosome-bound mRNAs * Updated lengths of nascent polypeptides (proteolysis tags) * Updated state of each ribosome: free, actively translating, or stalled
Rna	* Counts of mRNA, aminoacylated tRNA, and aminoacylated tmRNA species	* Updated counts of aminoacylated tRNA and aminoacylated tmRNA species

FIG. 35D

Enzymes/Complexes	Composition	Gene Name(s)
Initiation factor IF-1	(1) MG173	infA
Initiation factor IF-2	(1) MG142	infB
Initiation factor IF-3	(1) MG196	infC
Elongation factor G	(2) MG069	fusA
Elongation factor P	(1) MG026	elp
Elongation factor Tu	(2) MG451	tuf
Elongation factor Ts	(2) MG433	tsf
Peptide chain release factor 1	(1) MG258	prfA
Ribosome recycling factor	(1) MG435	frr
30S ribosomal subunit	(1 each) MGrra16S, MG070, MG087, MG088, MG090, MG092, MG150, MG155, MG157, MG160, MG164, MG165, MG168, MG175, MG176, MG311, MG417, MG424, MG446, MG481, MCS22	16S rRNA, rpsB, rpsL, rpsG, -, rpsR, rpsJ, rpsS, rpsC, rpsQ, rpsN, rpsH, rpsE, rpsM, rpsK, rpsD, rpsI, rpsO, rpsP, -, -
30S ribosomal subunit - initiation factor IF-3 complex	(1) Ribosome_30S, (1) MG196	30S ribosomal subunit, infC
50S ribosomal subunit	(1 each) MGrra23S, MGrra5S, MG081, MG082, MG093, MG151, MG152, MG153, MG154, MG156, MG158, MG159, MG161, MG162, MG163, MG166, MG1676, MG169, MG174, MG178, MG197, MG198, MG257, MG325, MG361, MG362, MG363, MG418, MG426, MG444, MG466, MG473	23S rRNA, 5S rRNA, rplK, rplA, -, rplC, rplD, rplW, rplB, rplV, rplP, rplC, rplN, rplX, rplE, rplF, rplR, rplO, rpmJ, rplQ, rpmI, rplT, rpmE, rpmG, -, rplL, rpmF, rplM, rpmB, rplS, rplJ4, rpmG-2
70S ribosome	(1) Ribosome_50S, (1) Ribosome_30S	50S ribosomal subunit, 30S ribosomal subunit
tmRNA, MCS6 (10aa RNA)	(1) MG0004	ssrA
SsrA binding protein	(1) MG050	ampB
peptidyl-tRNA hydrolase	(1) MG083	pth

Table S26 . Fixed parameters used in the Translation process class.

Parameter	Value	Symbol	Source
Ribosome elongation rate (amino acids/second)	16	k_{elong}	[564]
Probability that stationary ribosome is moved to a stalled state (mRNA replaced by tmRNA)	1×10^{-6}	$P_{stalled}$	See "Parameter Fitting"
tRNA sequences of the protein monomers			Determined from genome sequence, tRNA code [247]
tRNA sequences of the proteolysis tags			Determined from genome sequence, tRNA code [247]

FIG. 35E

```
Free 70S ribosomes ← min(free 30S, 50S ribosomal particles)
Decrement the copy numbers of free 30S and 50S ribosomal particles
foreach 70S ribosome i do
    Select the mRNA of 70S ribosome i weighted by the product of mRNA copy number and length
    Set the bound mRNA species of 70S ribosome i
    Select the position of 70S ribosome i along the bound mRNA with uniform probability
    Set the status of 70S ribosome i to actively translating
    Decrement the copy number of free 70S ribosomes
    Increment the copy number of bound 70S ribosomes
    Set the sequence of the nascent polypeptide corresponding to 70S ribosome i
```

FIG. 36A

PROTEIN PROCESSING I
 Σ SUB-MODEL 3600

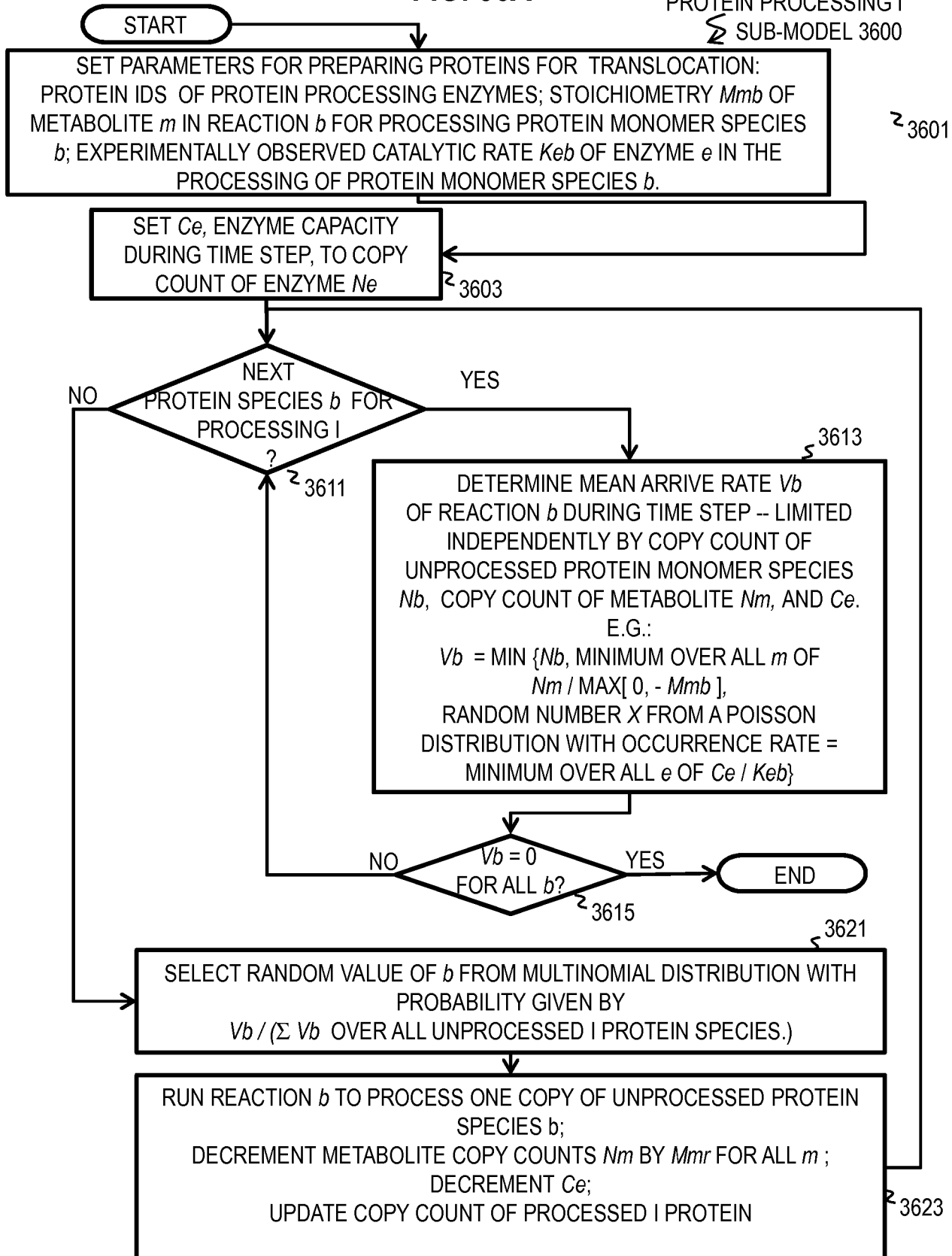


FIG. 36B

Input: p_i^p copy number of processing protein monomer species i

Let $k_i \leftarrow c_i \Delta t$ be the capacity of enzyme i for protein processing

repeat

- Calculate processing rates
- foreach** protein monomer species i **do**
 - └ Calculate v_i according to Eq. 20
- Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
- Update protein monomer copy numbers: $p_i^p \leftarrow p_i^p - 1$, $p_i^p \leftarrow p_i^p + 1$
- Update metabolites: $m \leftarrow m - M_{\bullet, i}$
- Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet, i}$

until no further processing possible ($v_i = 0 \forall i$)

FIG. 37A

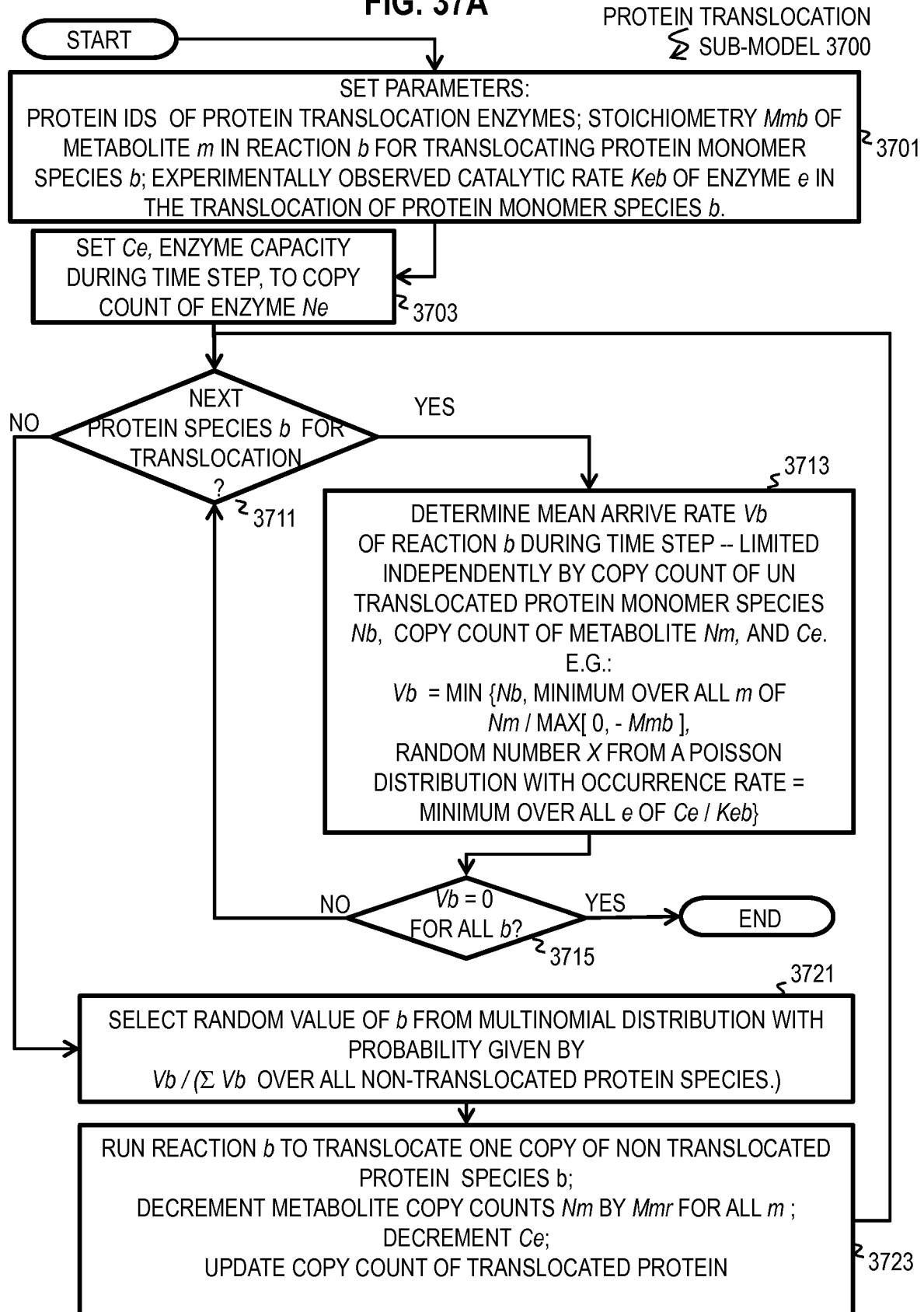


FIG. 37B

Input: p_i^l copy number of translocated (membrane- or extracellular-localized) protein monomer species i

Let $k_i \leftarrow c_i \Delta t$ be the capacity of enzyme i for protein translocation

repeat

- Calculate translocation rates
- foreach** protein monomer species i **do**
 - Calculate v_i according to Eq. 21
- Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
- Update protein monomer copy numbers: $p_i^s \leftarrow p_i^s - 1$, $p_i^l \leftarrow p_i^l + 1$
- Update metabolites: $m \leftarrow m - M_{s,i}$
- Update enzyme catalytic capacity: $k \leftarrow k - K_{s,i}$

until no further translocation possible ($v_i = 0 \forall i$)

FIG. 38A

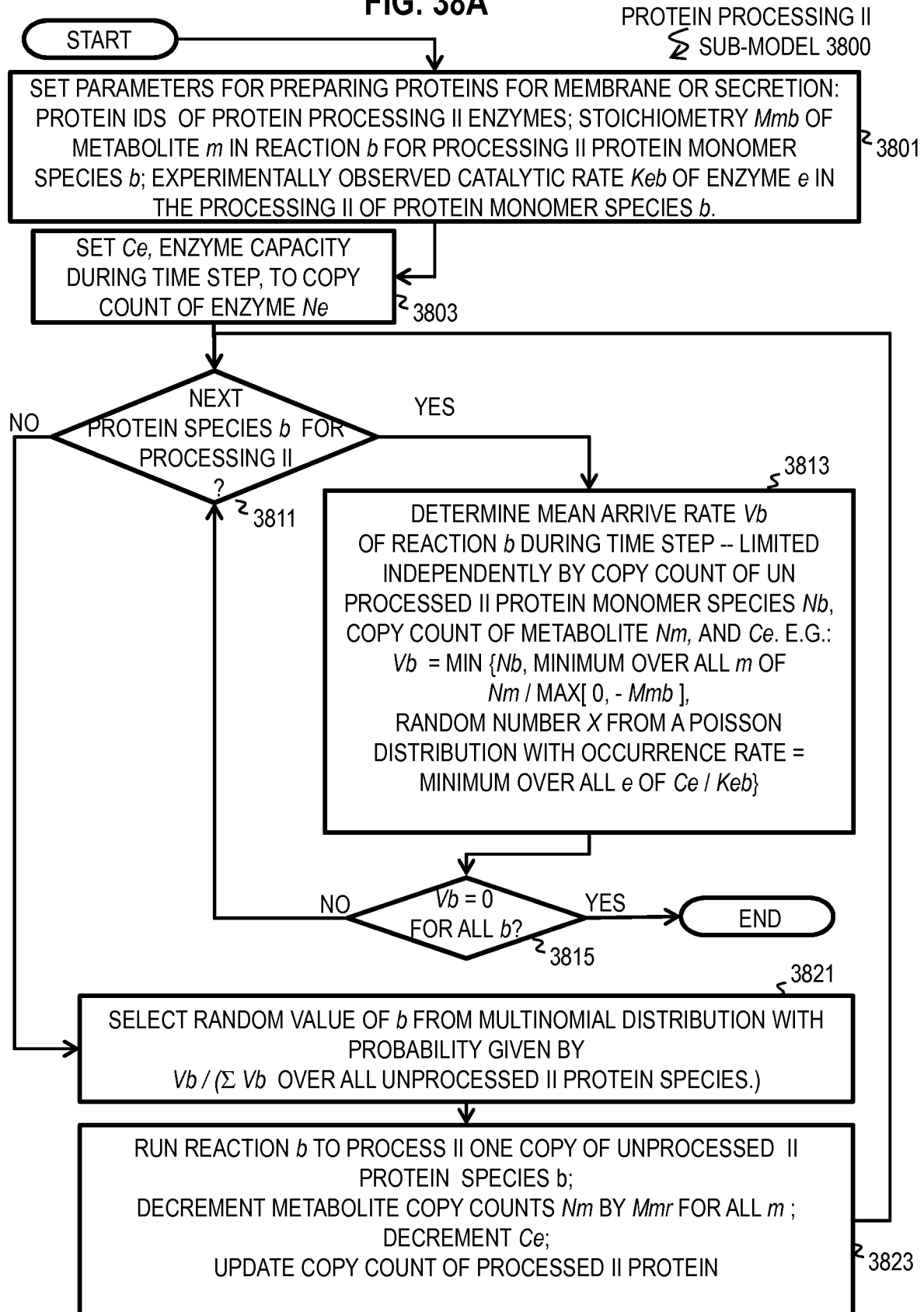


FIG. 38B

Input: p_i^P copy number of processed protein monomer species i
 Input: p_i^S copy number of cleaved signal sequence of protein monomer species i
 Let $k_i \leftarrow c_i \Delta t$ be the capacity of enzyme i for protein processing
 repeat
 Calculate processing rates
 foreach protein monomer species i do
 Calculate v_i according to Eq. 22
 Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
 Update protein monomer copy numbers: $p_i^S \leftarrow p_i^S - 1$, $p_i^P \leftarrow p_i^P + 1$, $p_i^C \leftarrow p_i^C + 1$
 Update metabolites: $m \leftarrow m - M_{\bullet, i}$
 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet, i}$
 until no further processing possible ($v_i = 0 \forall i$)

FIG. 39A

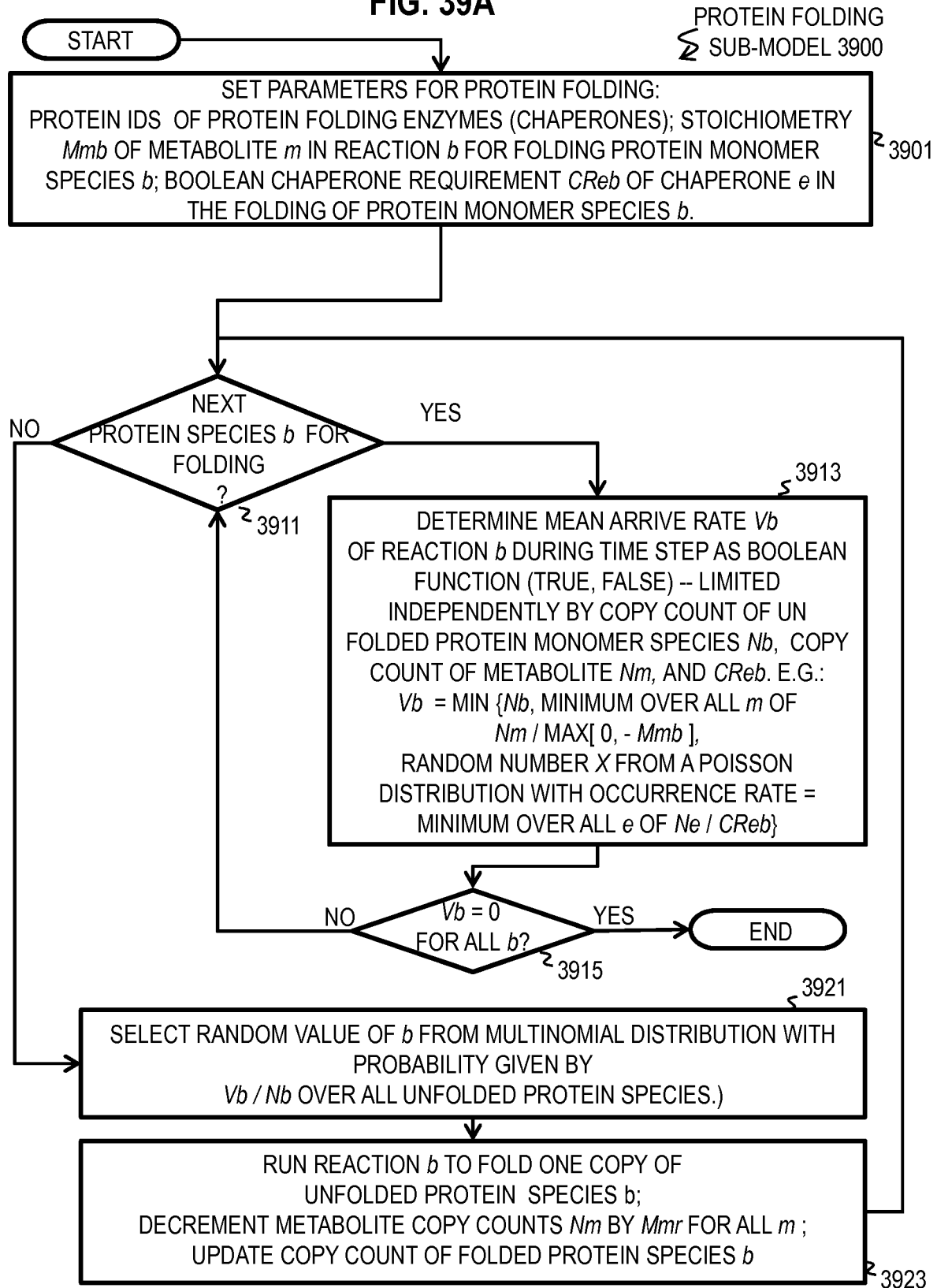


FIG. 39B

```

repeat
  foreach protein species  $i$  do
    Calculate the relative Boolean-valued folding rate,  $r_i$ , of each protein species
     $r_i \leftarrow \min \{p_i^u, m_j/M_{ij}, c_k \geq C_{ik}\}$ 
    Select a protein species  $i$  to fold according to  $\text{multinomialRand}(1, \vec{r})$ 
    Increment the folded copy number of protein species  $i$ ,  $p_i^f \leftarrow p_i^f + 1$ 
    Decrement the unfolded copy number of protein species  $i$ ,  $p_i^u \leftarrow p_i^u - 1$ 
    Decrement the copy numbers of the prosthetic groups of protein species  $i$ ,  $m_j \leftarrow m_j - M_{ij}$ 
until no additional proteins can fold ( $\vec{r} = 0$ )

```

FIG. 40A

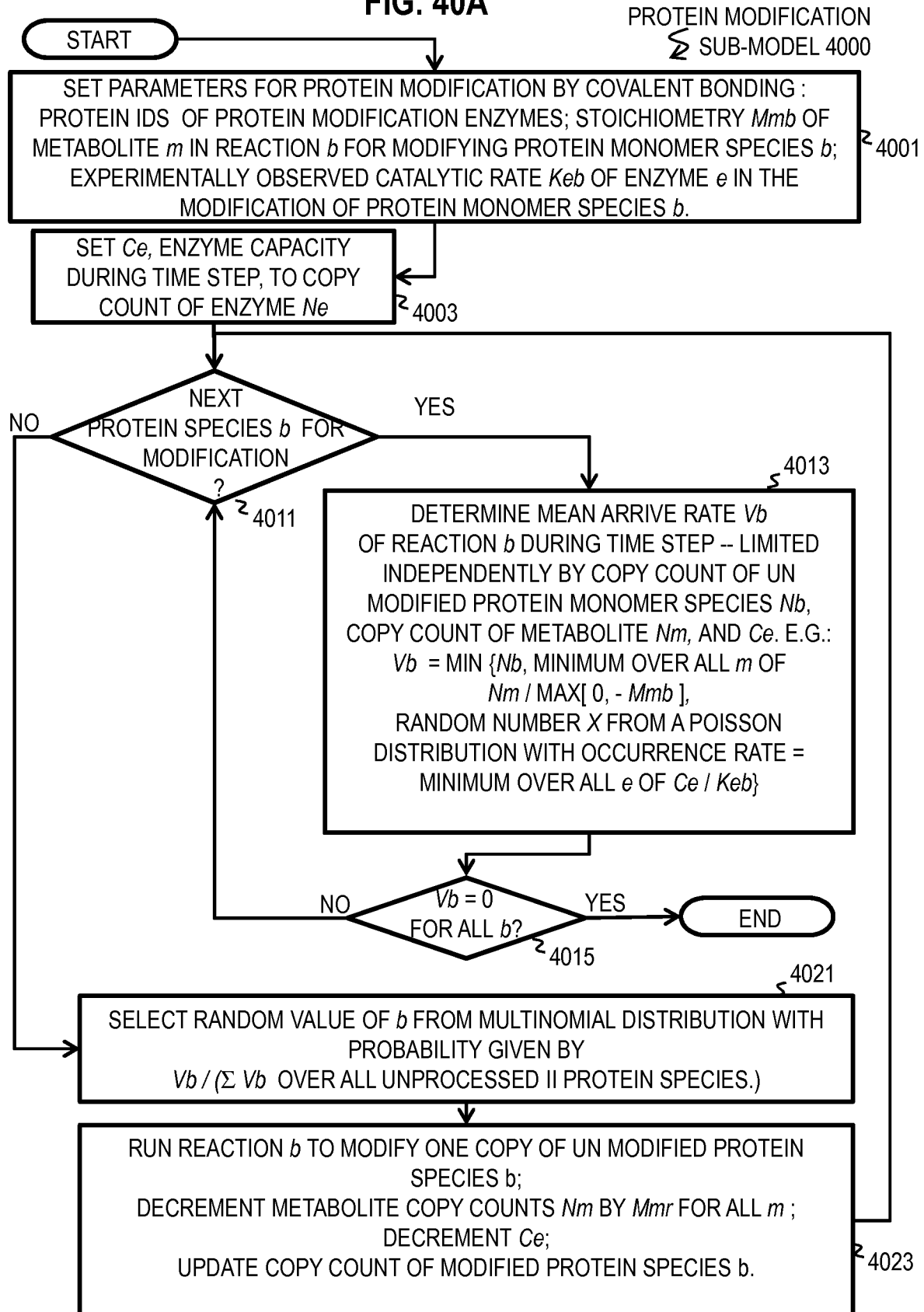


FIG. 40B

Input: p_i^m copy number of modified protein species i

Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for protein modification

repeat

Calculate modification rates

foreach *protein species* i **do**

Calculate v_i according to Eq. 24

Select protein species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$

Update protein copy numbers: $p_i^u \leftarrow p_i^u - 1, p_i^m \leftarrow p_i^m + 1$

Update metabolites: $m \leftarrow m - M_{\bullet, i}$

Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet, i}$

until *no further modification possible* ($v_i = 0 \forall i$)

FIG. 41A

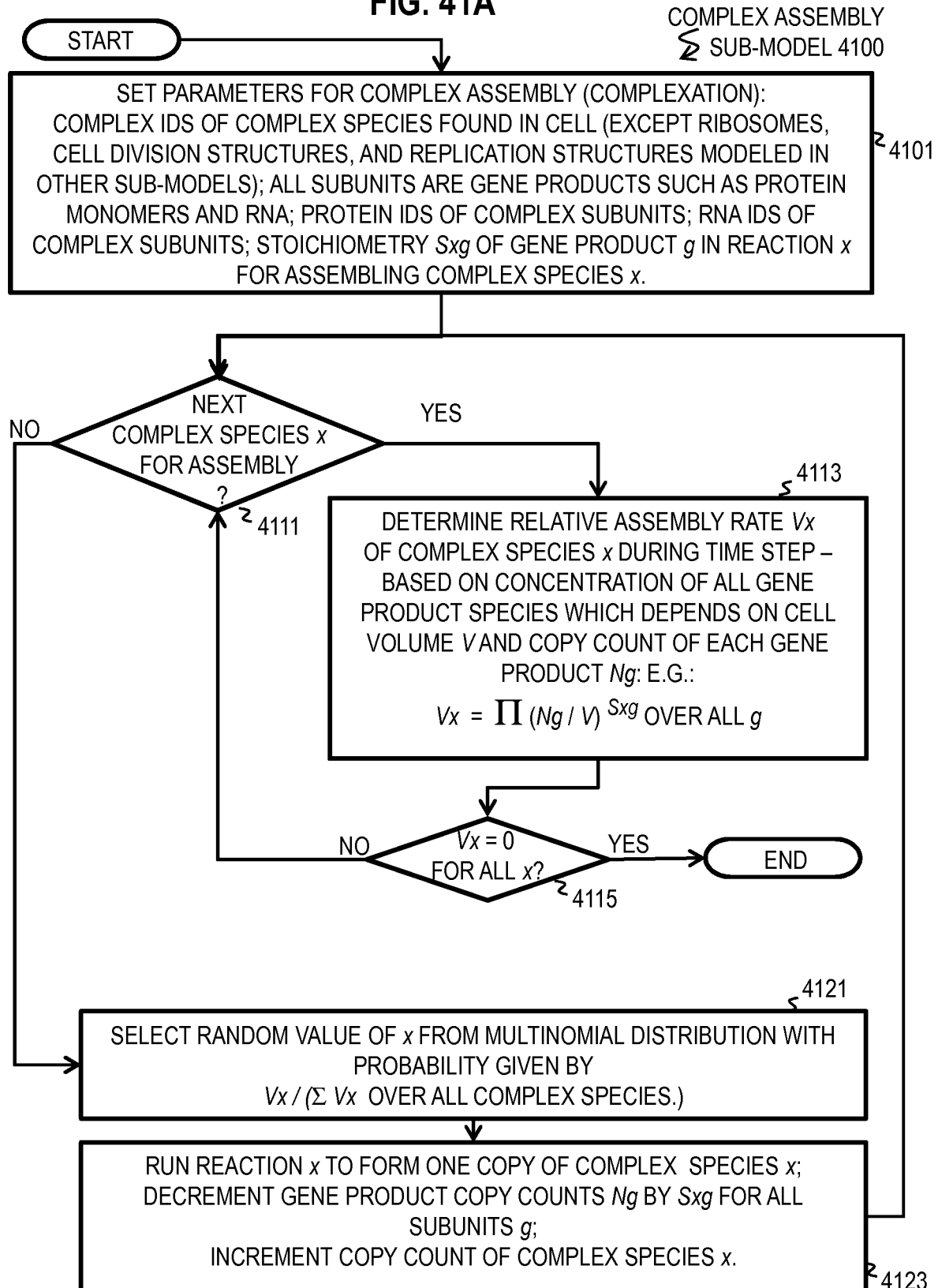


FIG. 41B

```

repeat
  foreach protein complex  $i$  do
    [ Calculate relative formation rate,  $r_i \leftarrow \prod_j \left(\frac{m_j}{V}\right)^{S_{i,j}}$ 
    Select a complex  $k$  to form according to  $\text{multinomialRand}(1, r_i / \sum_i r_i)$ 
    Increment copy number of complex  $k$ ,  $c_k \leftarrow c_k + 1$ 
    Decrement copy numbers of complex  $k$  subunits,  $m_j \leftarrow m_j - S_{k,j}$ 
until Insufficient subunits to form additional complexes

```

FIG. 42A

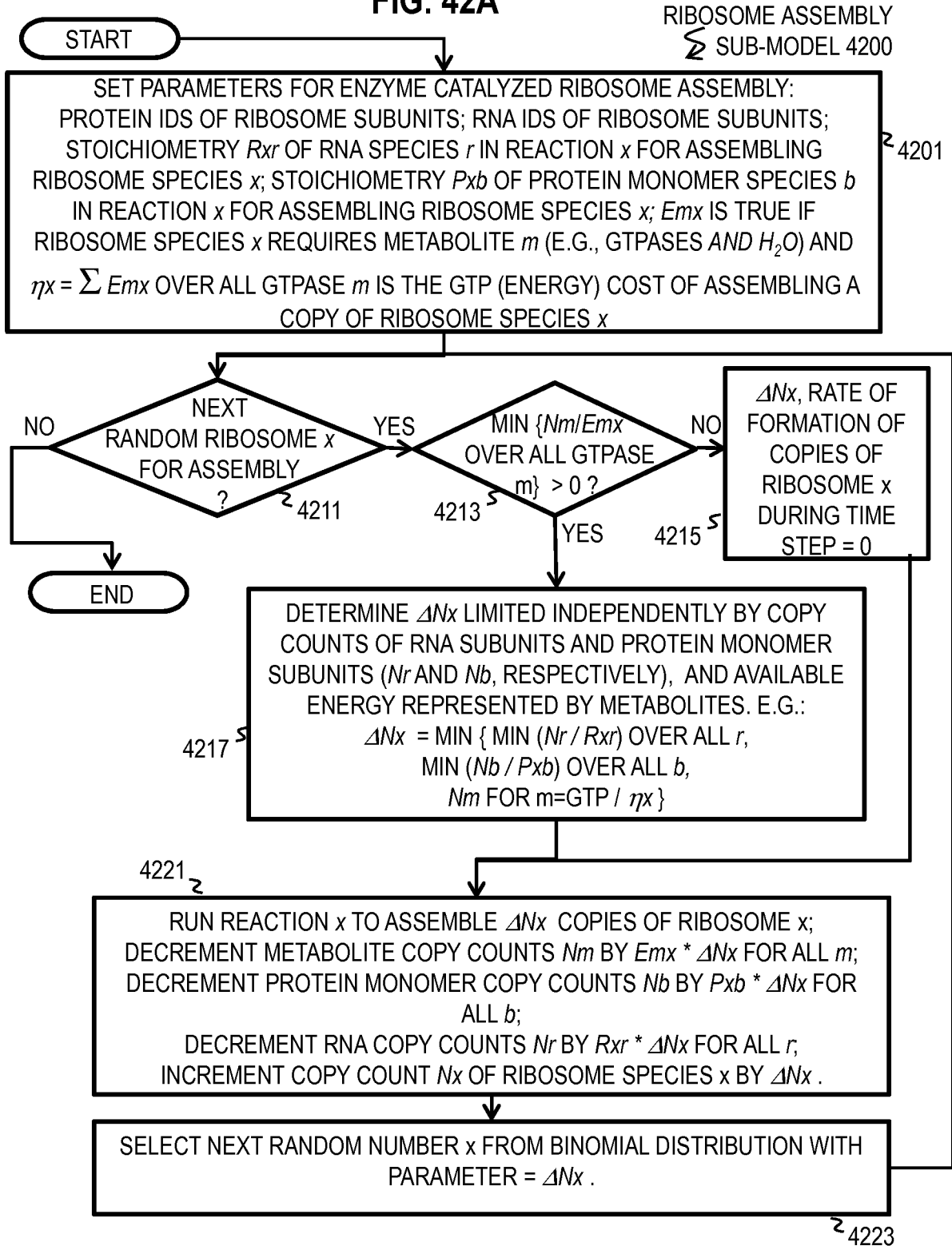


FIG. 42B

```

foreach ribosomal particle  $i$  in random order do
    Calculate the extent of ribosomal particle formation,  $\Delta c_i$ , using Eq. S26
    Form ribosomal subunits and decrement the copy numbers of rRNA transcripts and protein monomers.
     $r \leftarrow r - R_{\bullet i} \Delta c_i$ 
     $p \leftarrow p - P_{\bullet i} \Delta c_i$ 
     $c_i \leftarrow c_i + \Delta c_i$ 

    Hydrolyze GTP
     $m_{GTP} \leftarrow m_{GTP} - \eta_i \Delta c_i$ 
     $m_{CDP} \leftarrow m_{CDP} - \eta_i \Delta c_i$ 
     $m_{P_i} \leftarrow m_{P_i} - \eta_i \Delta c_i$ 
     $m_{H_2O} \leftarrow m_{H_2O} - \eta_i \Delta c_i$ 
     $m_{H^+} \leftarrow m_{H^+} - \eta_i \Delta c_i$ 

```

FIG. 43A

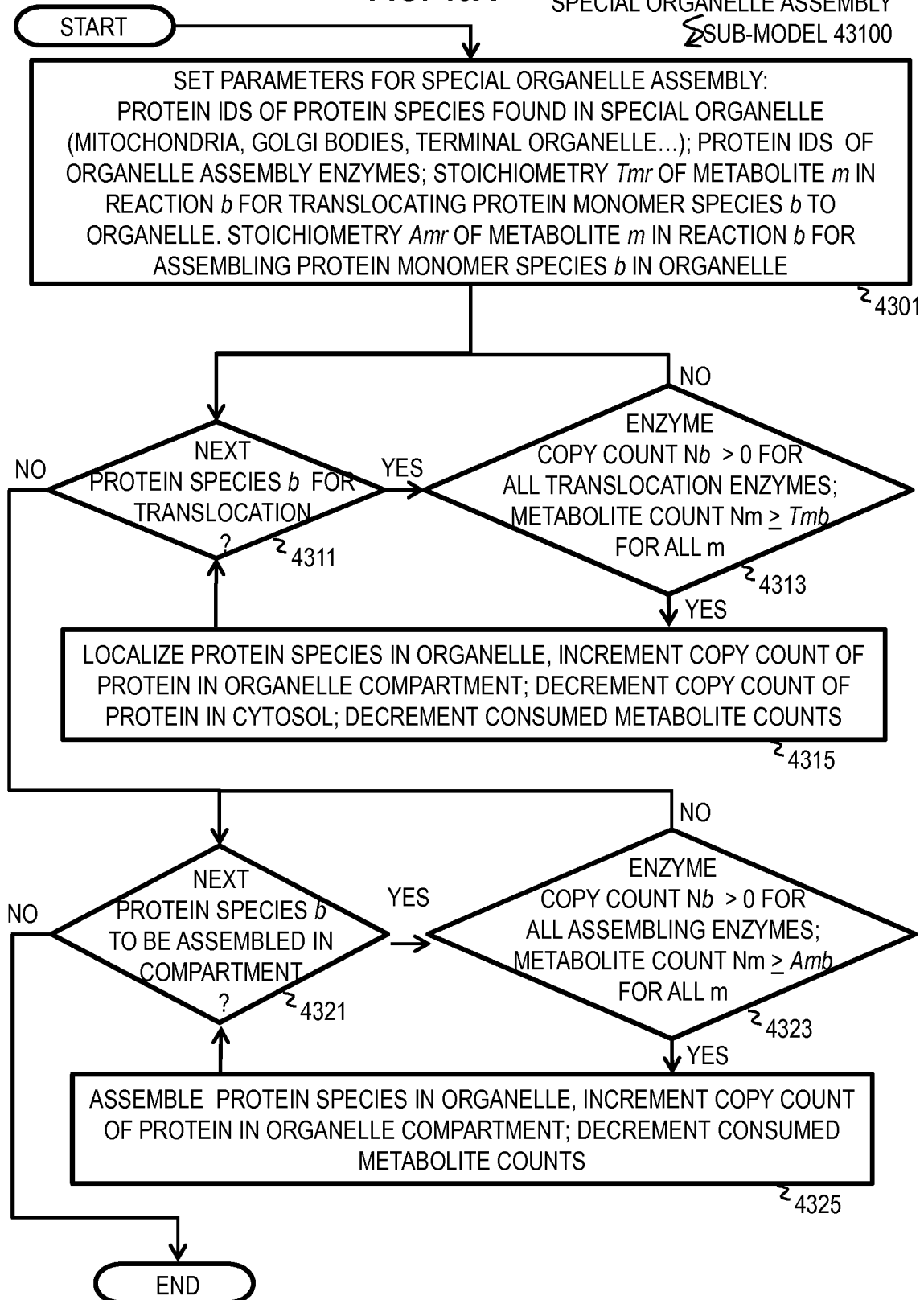
SPECIAL ORGANELLE ASSEMBLY
SUB-MODEL 43100

FIG. 43B

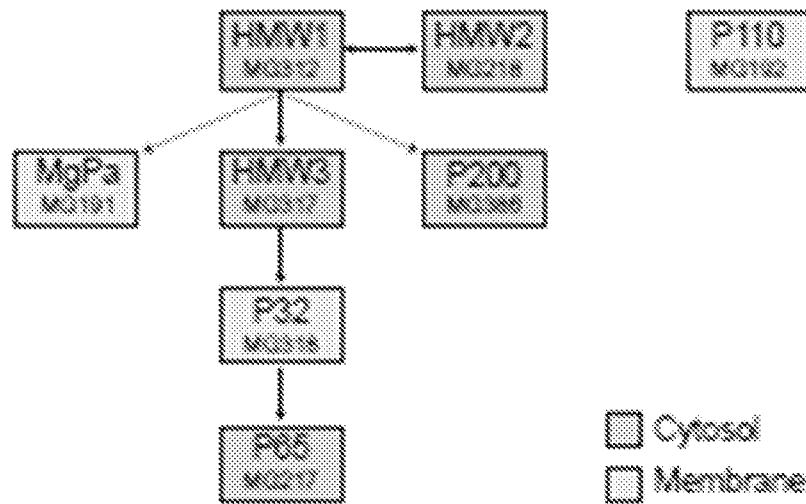


FIG. 43C

If HMW1 and HMW2 expressed ($HMW1_c + HMW1_m > 0$ and $HMW2_c + HMW2_m > 0$) then

 Localize HMW1 and HMW2 to the terminal organelle

$HMW1_{tc} \leftarrow HMW1_{tc} + HMW1_c$

$HMW2_{tc} \leftarrow HMW2_{tc} + HMW2_c$

$HMW1_c \leftarrow 0$

$HMW2_c \leftarrow 0$

If HMW1 localized to terminal organelle ($HMW1_{tc} > 0$) then

 Localize HMW3, MgPa, P200 to terminal organelle

$HMW3_{tc} \leftarrow HMW3_{tc} + HMW3_c$

$MgPa_{tm} \leftarrow MgPa_{tm} + MgPa_m$

$P200_{tc} \leftarrow P200_{tc} + P200_c$

$HMW3_c \leftarrow 0$

$MgPa_m \leftarrow 0$

$P200_c \leftarrow 0$

If HMW3 localized to terminal organelle ($HMW3_{tc} > 0$) then

 Localize P32 to terminal organelle

$P32_{tm} \leftarrow P32_{tm} + P32_m$

$P32_m \leftarrow 0$

If P32 localized to terminal organelle ($P32_{tm} > 0$) then

 Localize P65 to terminal organelle

$P65_{tc} \leftarrow P65_{tc} + P65_c$

$P65_c \leftarrow 0$

Localize P110 to terminal organelle

$P110_{tm} \leftarrow P110_{tm} + P110_m$

$P110_m \leftarrow 0$

FIG. 44A

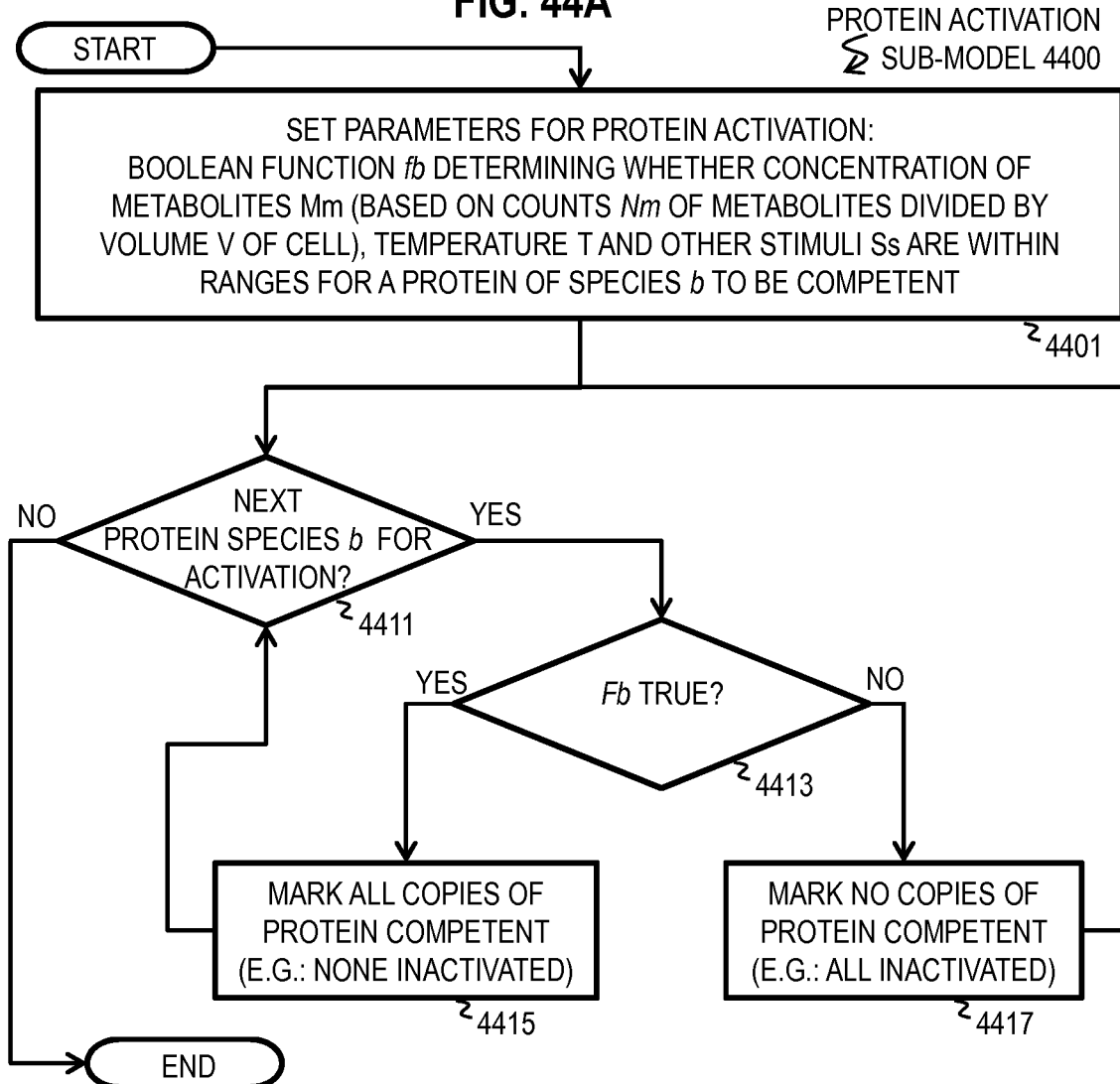


FIG. 44B

```

Input:  $p_i^c$  competent copy number of protein  $i$ 
Input:  $p_i^i$  incompetent copy number of protein  $i$ 
foreach regulated protein  $i$  do
  if regulatory rule  $f_i(m/V, T)$  then
    Mark all copies of protein  $i$  competent
     $p_i^c = p_i^c + p_i^i$ 
     $p_i^i = 0$ 
  else
    Mark all copies of protein  $i$  incompetent
     $p_i^i = p_i^i + p_i^c$ 
     $p_i^c = 0$ 
  
```

FIG. 45A

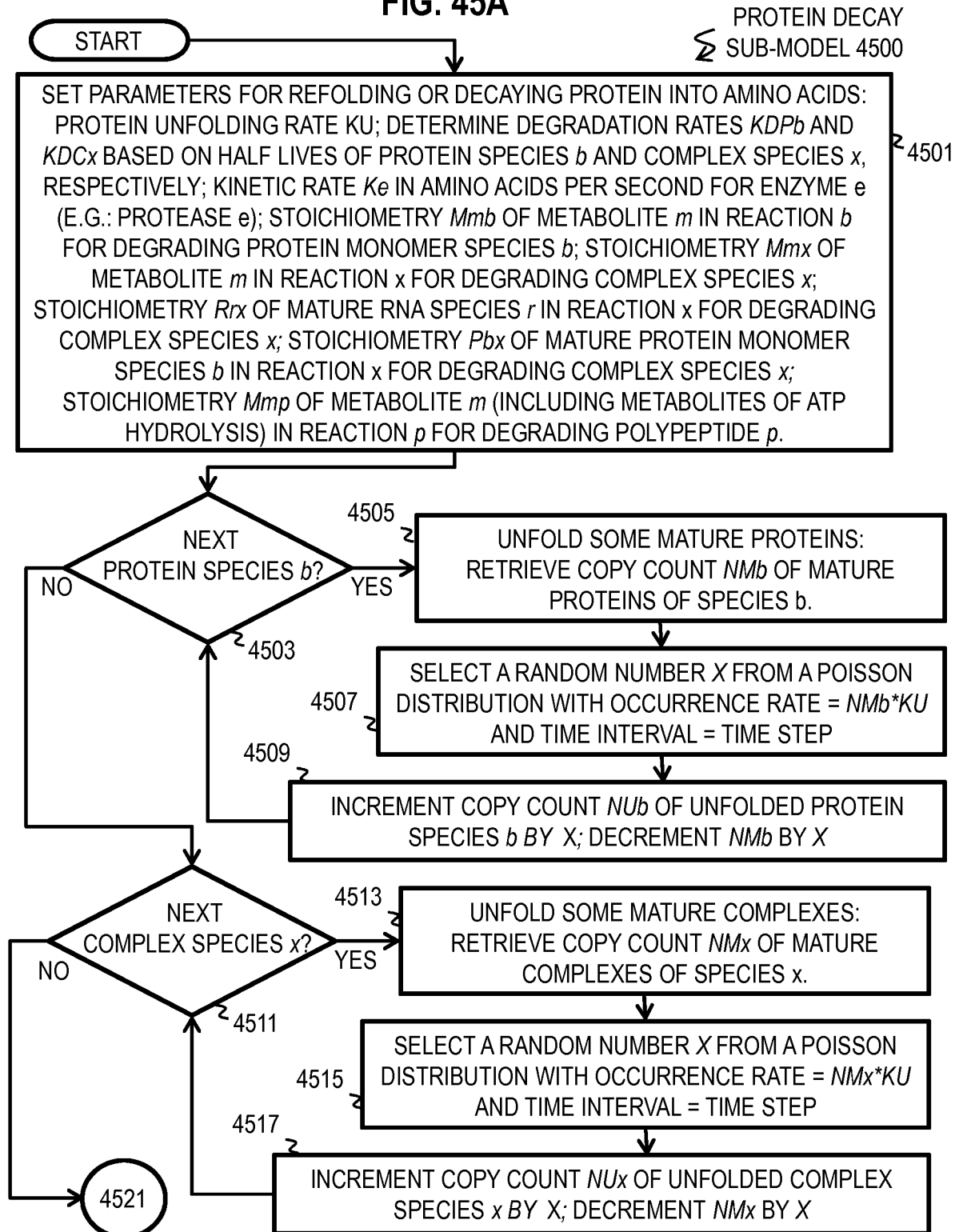


FIG. 45B

PROTEIN DECAY
 SUB-MODEL 4500
 (CONT)

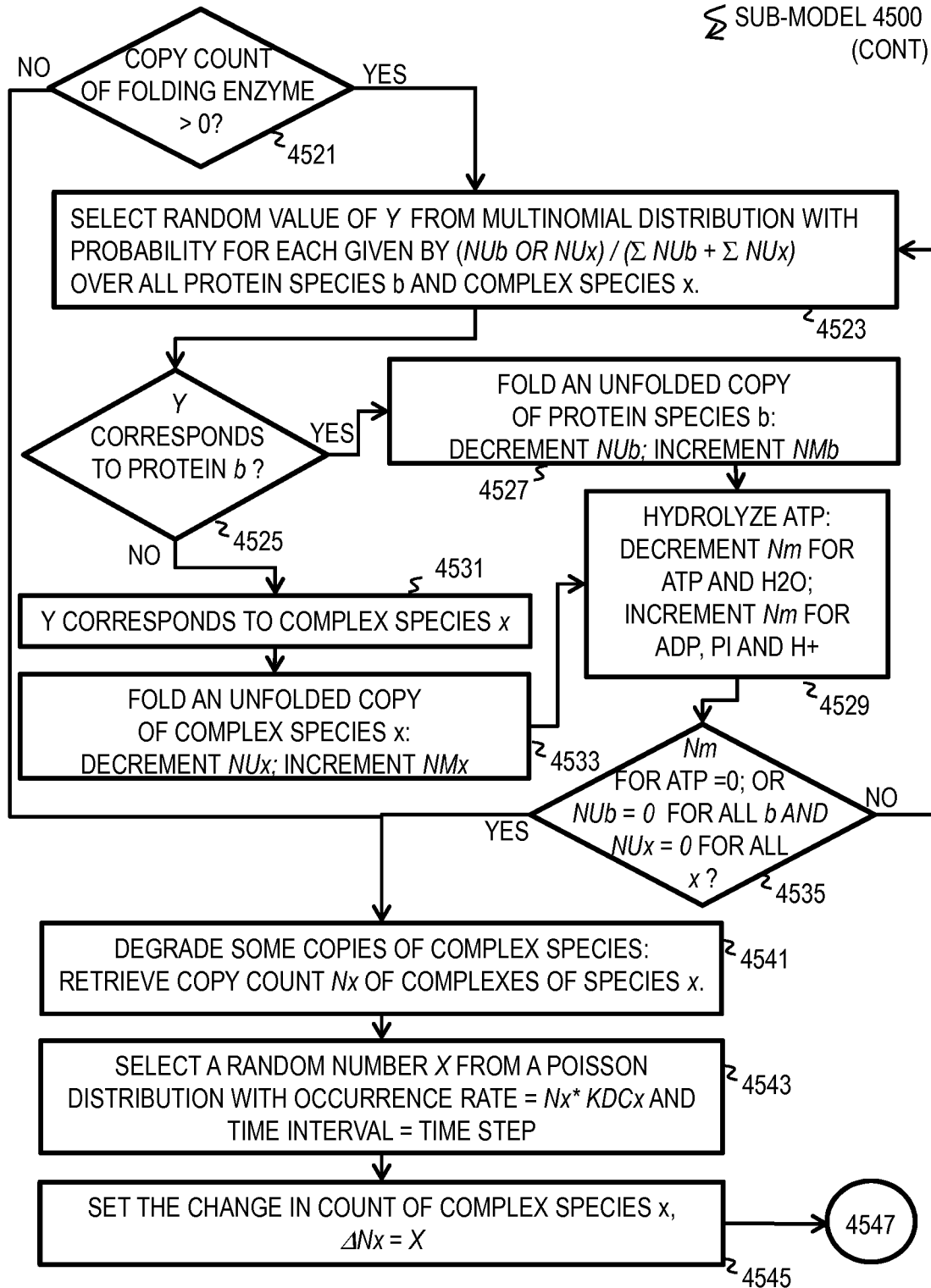


FIG. 45C

PROTEIN DECAY
SUB-MODEL 4500
Σ (CONT)

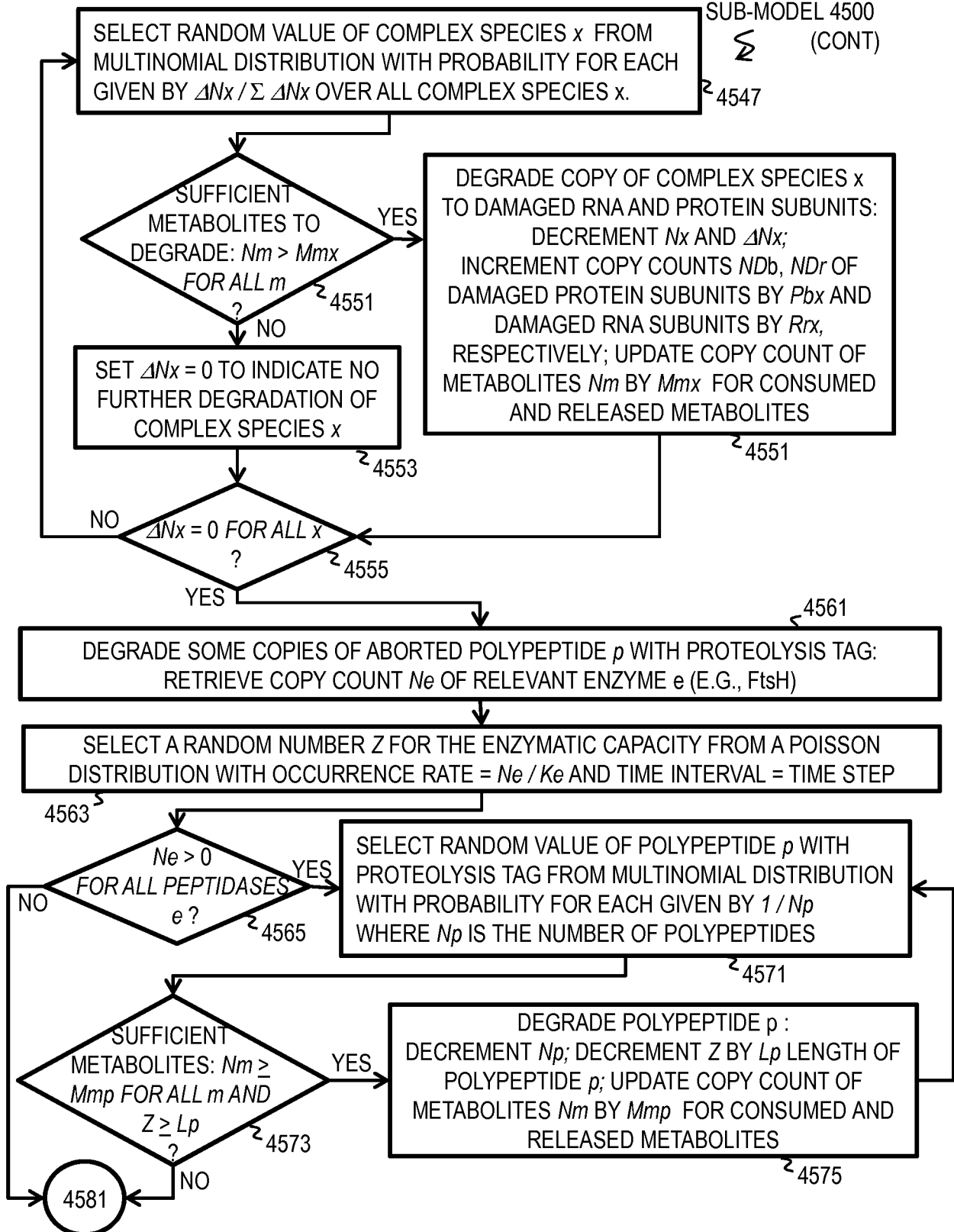


FIG. 45D

PROTEIN DECAY
SUB-MODEL 4500
Σ (CONT)

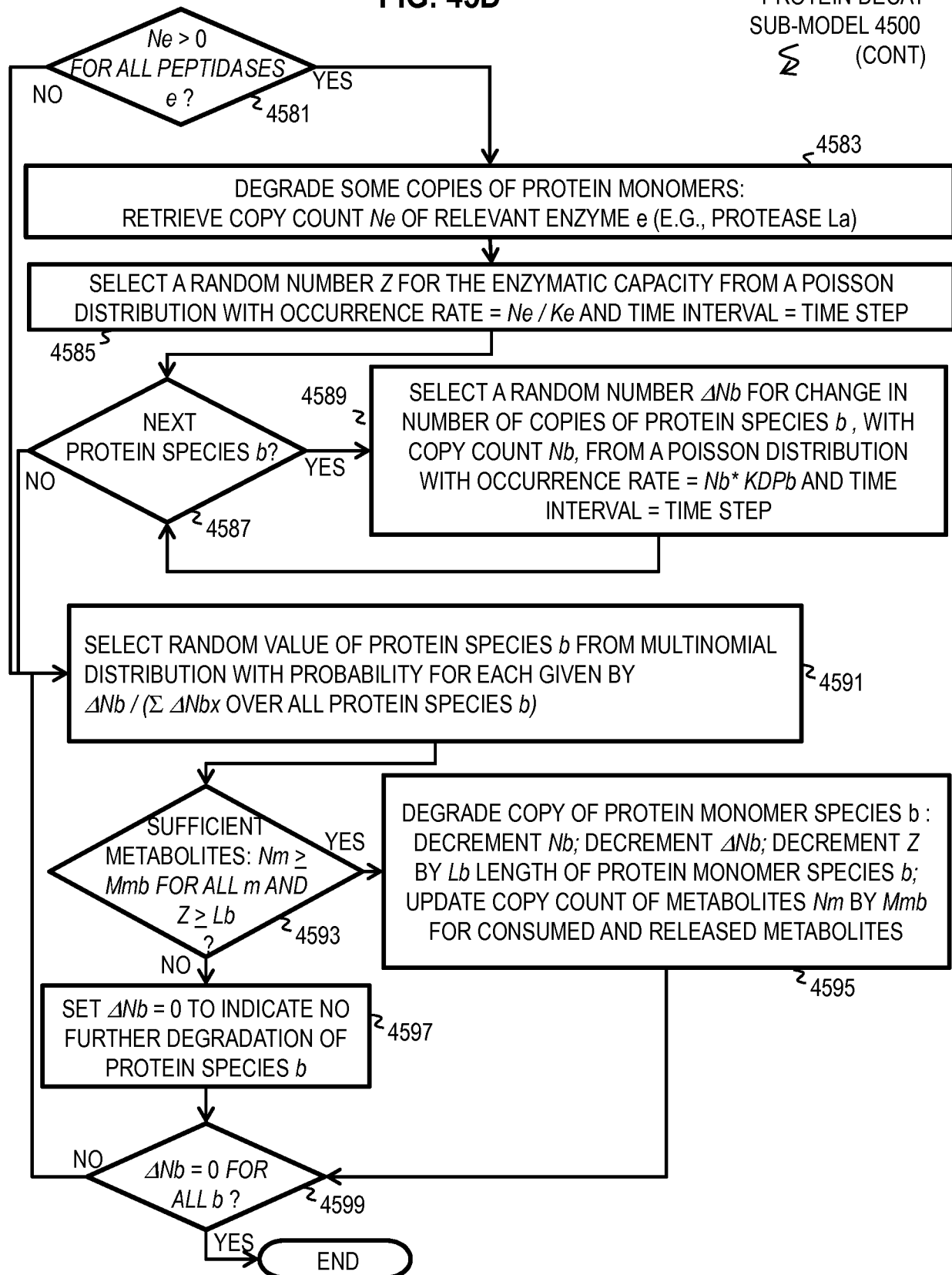


FIG. 45E

Input: $k_u = 10^{-6}$ s protein unfolding rate
 Input: $k_{d,i}^m = \ln 2/\tau_{1/2,i}$ degradation rate of protein monomer species i
 Input: $k_{d,i}^c = \ln 2/\tau_{1/2,i}$ degradation rate of macromolecular complex species i
 Input: k_{cat} , k_{catH} protease kinetic rates in amino acids per second
 Input: m_i copy number of metabolite i
 Input: r_i^d damaged copy number of RNA species i
 Input: p_i^m , p_i^u , p_i^d mature, unfolded, and damaged copy numbers of protein monomer species i
 Input: c_i^m , c_i^u , c_i^d mature, unfolded, and damaged copy numbers of macromolecular complex species i
 Input: $p \leftarrow \{p_i^m, p_i^u, p_i^d\}$ copy number of all forms of protein monomer species i
 Input: $c \leftarrow \{c_i^m, c_i^u, c_i^d\}$ copy number of all forms of macromolecular complex species i
 Input: e_{cat} , e_{catH} copy numbers of proteases La and FtsH
 Input: a_i sequence of prematurely aborted polypeptide i
 Input: l_i^p length of protein monomer species i
 Input: l_i^a length of prematurely aborted polypeptide i
 Input: P_{ij}^m reaction stoichiometry of metabolite i in the degradation of protein monomer j including ATP hydrolysis
 Input: C_{ij}^m stoichiometry of metabolite species i in macromolecular complex j
 Input: C_{ij}^r subunit stoichiometry of mature RNA species i in macromolecular complex j
 Input: C_{ij}^p subunit stoichiometry of mature protein monomer i in macromolecular complex j
 Input: S_{ij}^m reaction stoichiometry of metabolite i in the degradation of polypeptide j including ATP hydrolysis
 Input: $\Delta t = 1$ s is the simulation time step

Misfold proteins (see Alg. S13)
 Refold proteins (see Alg. S12)
 Degrade macromolecular complexes (see Alg. S14)
 Degrade prematurely aborted polypeptides (see Alg. S15)
 Degrade protein monomers (see Alg. S16)

FIG. 45F

```

if ClpB chaperone is expressed ( $cc_{clpB} > 0$ ) then
  repeat
    Select protein species  $i \sim \text{multinomialRand}(1, \{p_i^s, c_i^s\} / (\sum_j p_j^s + \sum_j c_j^s))$ 

    Update protein copy numbers
    if species  $i$  is a protein monomer then  $p_i^s \leftarrow p_i^s - 1, p_i^s \leftarrow p_i^s + 1$ 
    else  $c_i^s \leftarrow c_i^s - 1, c_i^s \leftarrow c_i^s + 1$ 

    Hydrolyze ATP
     $m_{\text{ATP}} \leftarrow m_{\text{ATP}} - 1$ 
     $m_{\text{ADP}} \leftarrow m_{\text{ADP}} + 1$ 
     $m_{\text{P}_i} \leftarrow m_{\text{P}_i} + 1$ 
     $m_{\text{H}_2\text{O}} \leftarrow m_{\text{H}_2\text{O}} - 1$ 
     $m_{\text{H}^+} \leftarrow m_{\text{H}^+} + 1$ 

  until ( $p_i^s = 0 \forall i$  and  $c_i^s = 0 \forall i$ ) or  $m_{\text{ATP}} = 0$ 

```

```

foreach protein monomer  $i$  do
  Calculate protein unfolding extent:  $\Delta p_i^s \leftarrow \text{poissonRand}(k_u p_i^s)$ 
  Update protein copy numbers:  $p_i^s \leftarrow p_i^s - \Delta p_i^s, p_i^s \leftarrow p_i^s + \Delta p_i^s$ 

foreach macromolecular complex  $i$  do
  Calculate protein unfolding extent:  $\Delta c_i^s \leftarrow \text{poissonRand}(k_u c_i^s)$ 
  Update protein copy numbers:  $c_i^s \leftarrow c_i^s - \Delta c_i^s, c_i^s \leftarrow c_i^s + \Delta c_i^s$ 

```

Calculate rates of protein degradation

```

foreach macromolecular complex  $i$  do
   $\Delta c_i \leftarrow \text{poissonRand}(k_{d,i}^c c_i)$ 

```

Degrade specific complexes

```

repeat
  Select protein species  $i \sim \text{multinomialRand}(\Delta c_i / \sum_j \Delta c_j)$ 
  if sufficient metabolic resources available ( $m_j \geq -C_{ji}^m \forall j$ ) then
    Breakdown complex into metabolites and damaged RNA and protein subunits
     $\Delta c_i \leftarrow \Delta c_i - 1$ 
     $c_i \leftarrow c_i - 1$ 
     $p_i^d \leftarrow p_i^d + C_{*i}^p$ 
     $r_i^d \leftarrow r_i^d + C_{*i}^r$ 
     $m \leftarrow m C_{*i}^m$ 
  else
     $\Delta c_i \leftarrow 0$ 

```

```

until  $\Delta c_i = 0 \forall i$ 

```

FIG. 45G

```

Let  $z \leftarrow \text{poissonRand}(c_{FtsH}/k_{FtsH}\Delta t)$  be the FtsH enzymatic capacity
while all peptidases expressed ( $c_i \forall \text{ peptidases } i$ ) do
    Select an aborted polypeptides  $i \sim \text{multinomialRand}(1, 1/n_{\text{app}})$ 
    If insufficient resources to degrade polypeptide ( $\beta_j s_i - S_{*i}^m > m_j$  and  $z \geq \text{length}(l_i^p) \geq 1$ ) then
        break
    Degrade polypeptide:  $s_i \leftarrow \emptyset$ ,  $m \leftarrow m + S_{*i}^m$ ,  $z \leftarrow z - \text{length}(l_i^p)$ 

```

```

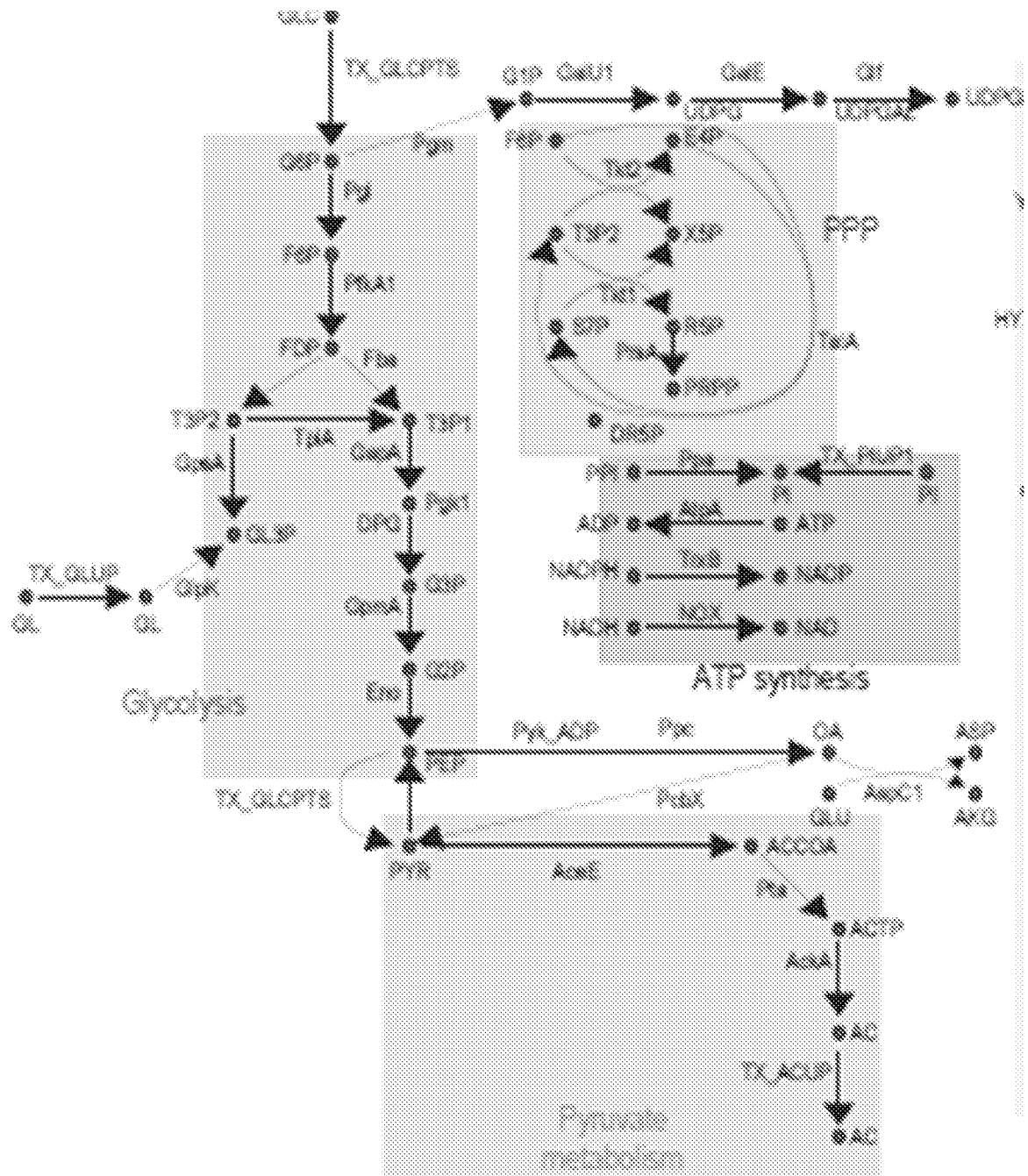
If all peptidases expressed ( $c_i \forall \text{ peptidases } i$ ) then
    Let  $z \leftarrow \text{poissonRand}(c_{Lon}/k_{Lon}\Delta t)$  be the Lon enzymatic capacity
    Calculate rates of protein degradation
    foreach protein monomer  $i$  do
         $\Delta p_i \leftarrow \text{poissonRand}(k_{d,i}^p p_i)$ 
    Degrade specific monomers
    repeat
        Select protein species  $i \sim \text{multinomialRand}(\Delta p_i / \sum_j \Delta p_j)$ 
        If sufficient metabolic resources available ( $m_j \geq -P_{ji}^m \forall j$  and  $z \geq l_i^p \geq 1$ ) then
            Breakdown protein into metabolites
             $\Delta p_i \leftarrow \Delta p_i - 1$ 
             $p_i \leftarrow p_i - 1$ 
             $m \leftarrow m + P_{*i}^m$ 
             $z \leftarrow z - l_i^p$ 
        else
             $\Delta p_i \leftarrow 0$ 
    until  $\Delta p_i = 0 \forall i$ 

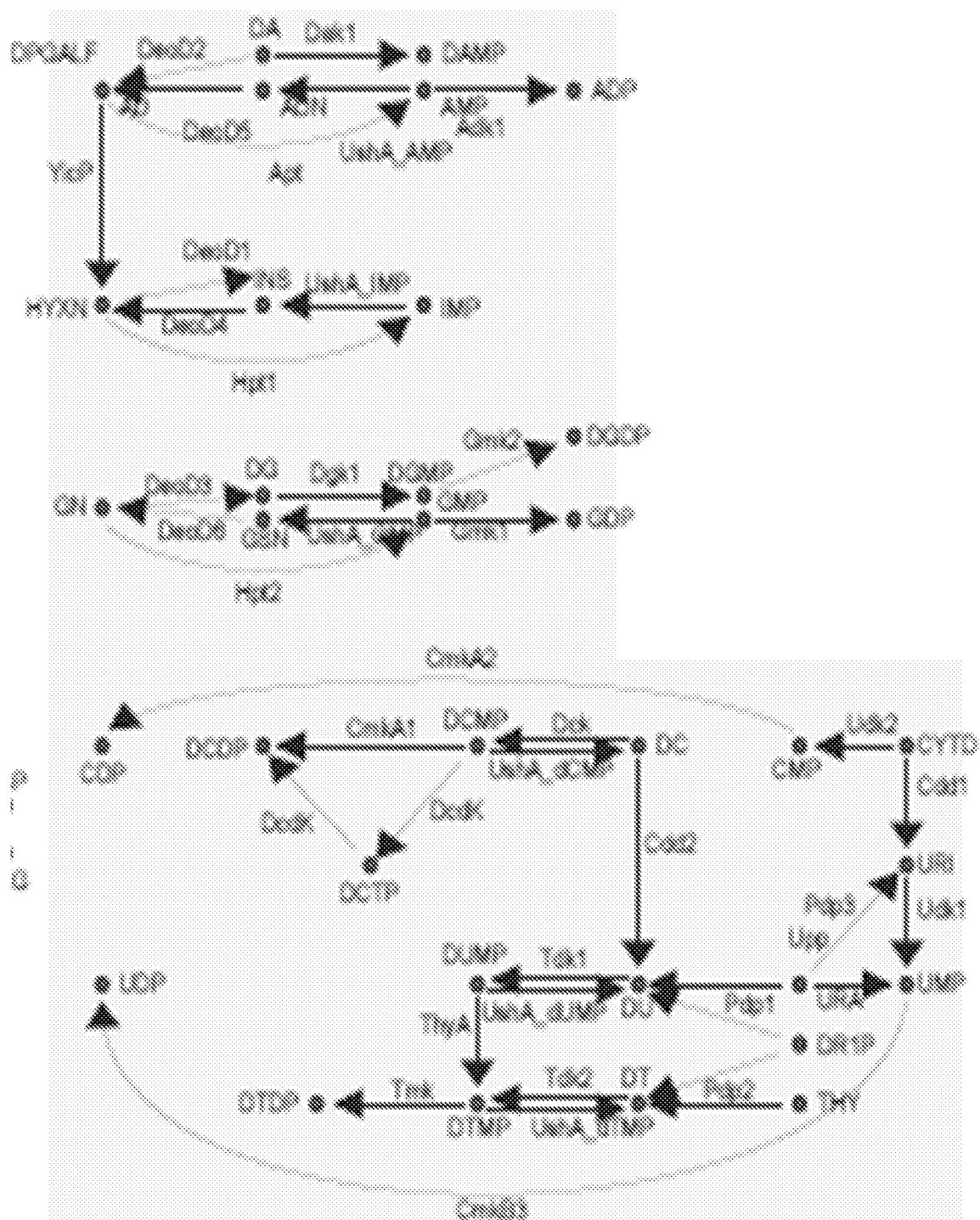
```

FIG. 46

Parameter	Value	Symbol
Maximum length of an FtsZ filament	9 subunits	n
Rate of forward activation of FtsZ by GTP	1.1 s^{-1}	k_{act1}
Rate of reverse activation of FtsZ by GTP	0.01 s^{-1}	k_{act2}
Rate of forward exchange of FtsZ-GTP from FtsZ-GDP	$1 \times 10^4 \text{ M}^{-1} \text{ s}^{-1}$	k_{ex1}
Rate of reverse exchange of FtsZ-GTP to FtsZ-GDP	$5 \times 10^3 \text{ M}^{-1} \text{ s}^{-1}$	k_{ex2}
Rate of forward nucleation of FtsZ-GTP to form a dimer	$4.2 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	k_{nuc1}
Rate of reverse nucleation of FtsZ-GTP to dissociate a dimer	40 s^{-1}	k_{nuc2}
Rates of forward elongation of FtsZ-GTP into polymers of length 3 to 9	$5.1 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	k_{el1}
Rates of reverse elongation of FtsZ-GTP from polymers of length 3 to 9	2.9 s^{-1}	k_{el2}

FIG. 47A





Nucleotide metabolism

FIG. 47C

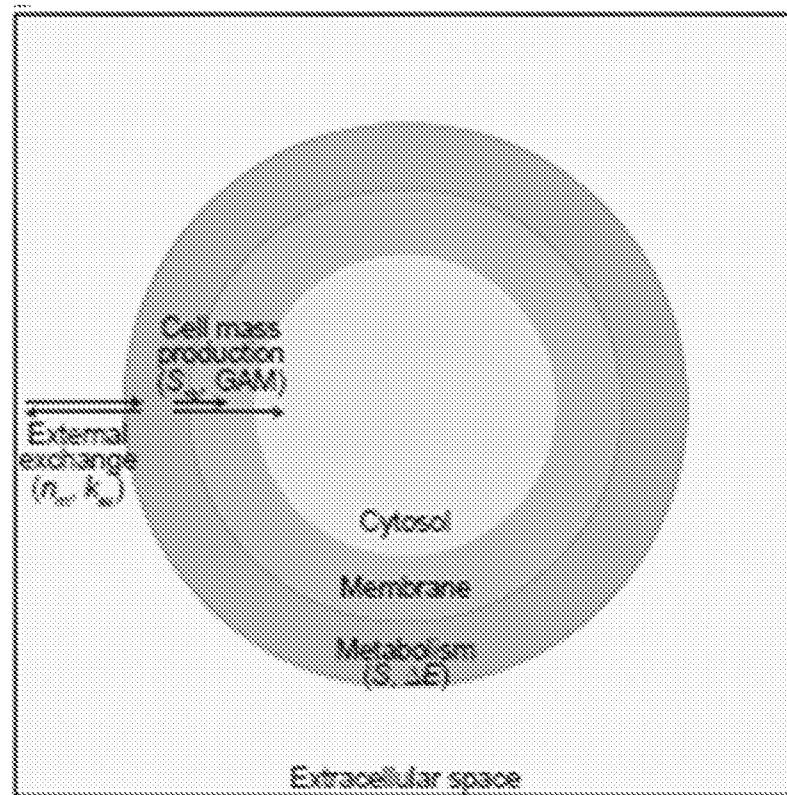


FIG. 47D

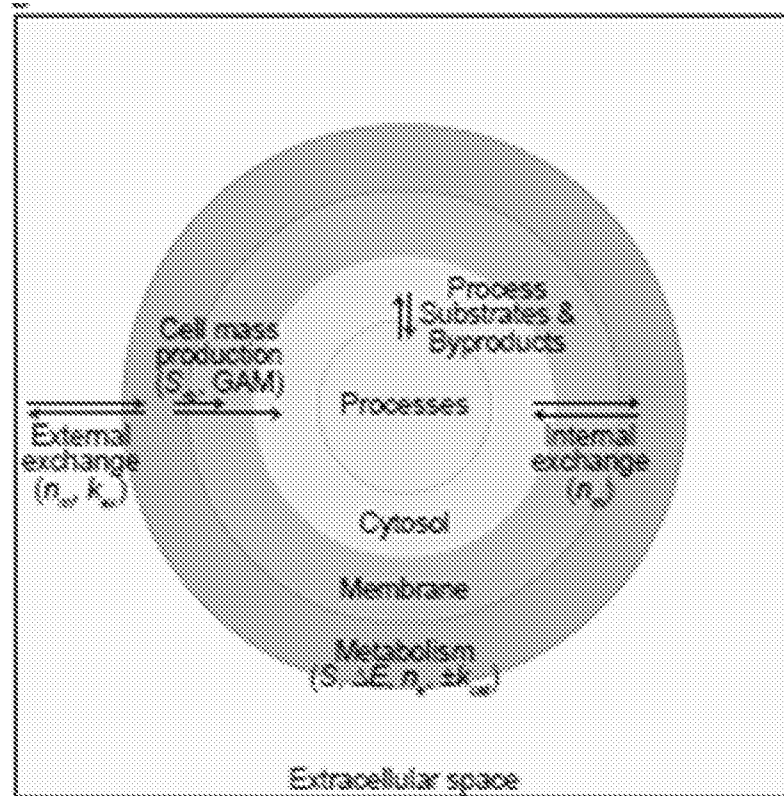


FIG. 47E

Calculate flux bounds

begin

Initialize bounds: $v_{l,i} \leftarrow -\text{inf}$, $v_{u,i} \leftarrow +\text{inf}$

foreach thermodynamically irreversible reaction i do

└ Constrain reverse flux to zero: $v_{r,i} \leftarrow 0$

foreach chemically catalyzed reaction i do

└ $j \leftarrow$ index of enzyme which catalyzes reaction i

└ If $k_{-i,j}$ is known then

└ bound flux by enzyme kinetics and expression: $v_{l,i} \leftarrow \max(v_{l,i}, k_{-cat,i} m_{e,j})$

└ else bound flux by enzyme expression: $v_{l,i} \leftarrow v_{l,i} \cdot (n_{e,j} > 0)$

└ If $k_{+i,j}$ is known then

└ bound flux by enzyme kinetics and expression: $v_{u,i} \leftarrow \min(v_{u,i}, k_{+cat,i} n_{e,j})$

└ else bound flux by enzyme expression: $v_{u,i} \leftarrow v_{u,i} \cdot (n_{e,j} > 0)$

foreach chemical reaction i do

└ foreach protein substrate j of reaction i do

└ └ If protein substrate j is not expressed then constrain flux to zero: $v_{l,i} \leftarrow 0$, $v_{u,i} \leftarrow 0$

foreach internal exchange reaction i do

└ $j \leftarrow$ index of metabolite exchanged by reaction i

└ Bound internal metabolite exchange by copy number: $v_{l,i} \leftarrow \max(v_{l,i}, -n_{m,j})$

foreach external exchange reaction i do

└ $j \leftarrow$ index of metabolite exchanged by reaction i

└ Bound external metabolite exchange by copy number and maximum exchange rate: $v_{l,i} \leftarrow \max(v_{l,i}, mk_{-ex,i})$

└ $v_{u,i} \leftarrow \min(v_{u,i}, mk_{+ex,i}, n_{m,j})$

FIG. 47F

Calculate the growth rate and flux of each reaction

begin

```

┌  $v^* \leftarrow \underset{v}{\operatorname{argmax}} \mu = v_5$  subject to  $Sv = 0$  and  $v_l \leq v \leq v_u$ 
└  $\mu^* \leftarrow v_5^*$ 

```

Update the copy number of each metabolite species

begin

```

    foreach extracellular metabolite  $i$  do
        ┌  $j \leftarrow$  index of reaction which exchanges metabolite  $i$ 
        └  $n_{m,i} \leftarrow n_{m,i} + v_j^*$ 

    foreach intracellular metabolite  $i$  do
        ┌  $j \leftarrow$  index of reaction which exchanges/recycle metabolite  $i$ 
        └  $n_{m,i} \leftarrow n_{m,i} + v_j^*$ 

    foreach metabolic objective component  $i$  do
        ┌  $n_{m,i} \leftarrow n_{m,i} - S_{i,5}\mu^*$ 

    Turnover extra energy produced beyond the demands of the 27 other modeled cellular processes.
     $\Delta GAM = GAM - \text{used energy}$ 
     $n_{m,ATP} \leftarrow n_{m,ATP} - \Delta GAM\mu^*$ 
     $n_{m,ADP} \leftarrow n_{m,ADP} + \Delta GAM\mu^*$ 
     $n_{m,P_i} \leftarrow n_{m,P_i} + \Delta GAM\mu^*$ 
     $n_{m,H_2O} \leftarrow n_{m,H_2O} - \Delta GAM\mu^*$ 
     $n_{m,H^+} \leftarrow n_{m,H^+} + \Delta GAM\mu^*$ 

```

FIG. 48A

Parameters	Value	Symbol
Length of an FtsZ filament	40 nm	L
Number of FtsZ subunits in an FtsZ filament	9 subunits	S
Rate of FtsZ-GTP hydrolyzing to FtsZ-GDP	0.15 s^{-1}	k_{hyd}
Rate of FtsZ polymers binding to the membrane	0.7 s^{-1}	k_{bind}
Rate of FtsZ polymers dissociating from the membrane	0.7 s^{-1}	k_{unbind}

FIG. 48B

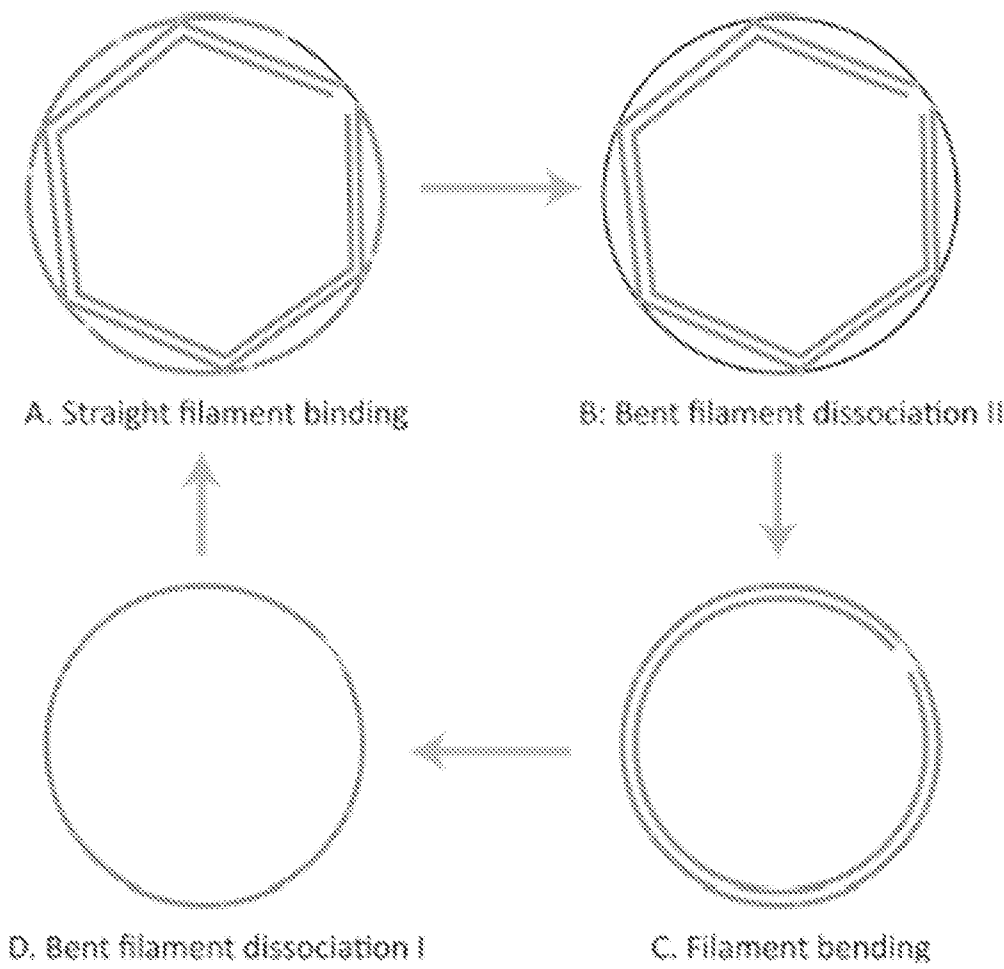


FIG. 48C

Connected State	Read from state	Written to state
FtsZ Ring	* Number of filaments bound to each edge, and whether they are bent or straight * Calculation of number of polygon edges in current cell diameter	* Updated number of filaments bound to each edge, and whether they are bent or straight
Geometry	* Pinched diameter of the cell * Filament length parameters (L, S)	* Updated pinched diameter of the cell
Chromosome	* Has chromosome segregated?	

FIG. 49

adherent $\leftarrow$ organelle proteins *MG191*, *MG192*, *MG217*, *MG218*, *MG312*, *MG317*, *MG318*, and *MG386* expressed
tlr1 $\leftarrow$ *adherent* AND (lipoproteins *MG149* and *MG412* expressed)
tlr2 $\leftarrow$ *adherent* AND (lipoproteins *MG309* expressed)
tlr6 $\leftarrow$ *adherent* AND (lipoproteins *MG149*, *MG309*, and *MG412* expressed)
nfbk $\leftarrow$ *tlr2* AND (*tlr1* OR *tlr6*)
inflammation $\leftarrow$ *nfbk* OR (*adherent* AND *MG075*, *MG149*, *MG309*, or *MG412* antigen expressed)

FIG. 50

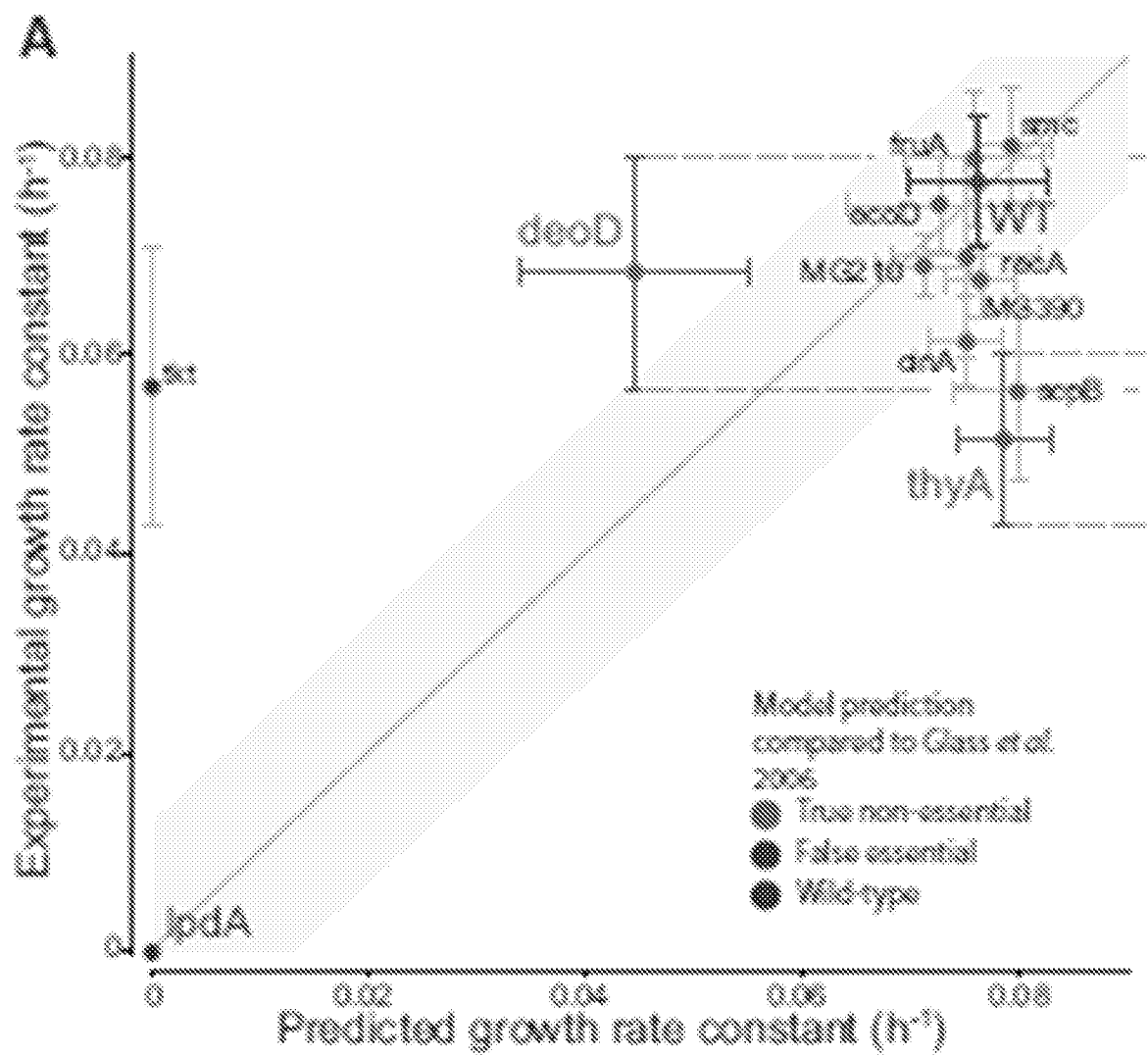


FIG. 51

