Bioinformatics, 36(22-23), 2020, 5473–5480
doi: 10.1093/bioinformatics/btaa1040
Advance Access Publication Date: 26 December 2020
Original Paper

Systems biology

Atlas: automatic modeling of regulation of bacterial gene expression and metabolism using rule-based languages

Rodrigo Santibáñez [1,2,*], Daniel Garrido [2] and Alberto J. M. Martin [1,*]

[1]Laboratorio de Biología de Redes, Centro de Genómica y Bioinformática, Universidad Mayor, Santiago 8580745, Chile and [2]Department of Chemical and Bioprocess Engineering, School of Engineering, Pontificia Universidad Católica de Chile, Santiago 7820436, Chile

*To whom correspondence should be addressed.
Associate Editor: Pier Luigi Martelli

Received on May 9, 2020; revised on November 19, 2020; editorial decision on December 5, 2020; accepted on December 12, 2020

Abstract

Motivation: Cells are complex systems composed of hundreds of genes whose products interact to produce elaborated behaviors. To control such behaviors, cells rely on transcription factors to regulate gene expression, and gene regulatory networks (GRNs) are employed to describe and understand such behavior. However, GRNs are static models, and dynamic models are difficult to obtain due to their size, complexity, stochastic dynamics and interactions with other cell processes.

Results: We developed *Atlas*, a Python software that converts genome graphs and gene regulatory, interaction and metabolic networks into dynamic models. The software employs these biological networks to write rule-based models for the PySB framework. The underlying method is a divide-and-conquer strategy to obtain sub-models and combine them later into an ensemble model. To exemplify the utility of Atlas, we used networks of varying size and complexity of *Escherichia coli* and evaluated *in silico* modifications, such as gene knockouts and the insertion of promoters and terminators. Moreover, the methodology could be applied to the dynamic modeling of natural and synthetic networks of any bacteria.

Availability and implementation: Code, models and tutorials are available online (https://github.com/networkbiolab/atlas).

Contact: rlsantibanez@uc.cl or alberto.martin@umayor.cl

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Recent technological advances have enabled the inquiry into, and understanding of, biological systems at an unprecedented level of detail (e.g. Regev *et al.*, 2017). From such developments, the impact of stochastic dynamics in living systems has been corroborated, measured and modeled (Raj and van Oudenaarden, 2008). To date, most of the available models look for the reproduction of cell metabolism at the genome scale using constraint-based models (Szigeti *et al.*, 2018). However, constraint-based models disregard the dynamic and stochastic nature of metabolism (Costa *et al.*, 2016) and the prediction of the impact of genetic modifications remains challenging [e.g. Foster *et al.* (2019) or Long and Antoniewicz (2019)]. Dynamic modeling of metabolism has been proposed to circumvent the drawbacks of constraint-based models (e.g. Hädicke and Klamt, 2017) despite specific issues, such as the need for calibration, extensive validation, and showing a time-consuming development. Also, it is necessary to consider that metabolism is only one aspect of cellular behavior, and models are desired that encompass all cellular processes (Karr *et al.*, 2012, 2015; Sanghvi *et al.*, 2013). If available, those models would help create an understanding of complex cell dynamics, with applications in biotechnology or biomedicine (Carrera and Covert, 2015), e.g. designing minimal cells (Rees-Garbutt *et al.*, 2020) or synthetic genomes (Fredens *et al.*, 2019). Although there are available whole-cell models for *Mycoplasma genitalium* (Karr *et al.*, 2012) and recently for *Escherichia coli* (Macklin *et al.*, 2020) and there is a proposed pathway to develop integrative and larger models (Covert *et al.*, 2008; Szigeti *et al.*, 2018), whole-cell models are still not widely developed or adopted (Szigeti *et al.*, 2018).

2 Approach

Gene expression regulates metabolism, which, in turn, modulates transcription, translation and degradation rates as well as the

activity of transcription factors (Covert *et al.*, 2008). These processes interplay in networks of molecular interactions between DNA, RNA, proteins and metabolites (Grimbs *et al.*, 2019; Hernández-Prieto *et al.*, 2014). Here, we aimed to perform the integrative modeling of transcription, translation, regulation of gene expression, metabolism and genome architecture, which is considered a prototype whole-cell model (Szigeti *et al.*, 2018). We developed *Atlas*, a software that facilitates the dynamic modeling of gene regulation and bacterial metabolism by using biological networks to develop rule-based models (RBMs) employing the PySB framework (Lopez *et al.*, 2013) for later simulation, curation and analysis. The developed software takes inspiration from available tools that automate the reconstruction of draft constraint-based models, such as Merlin (Dias *et al.*, 2015), RAVEN (Agren *et al.*, 2013; Wang *et al.*, 2018), ModelSEED (Henry *et al.*, 2010), KBase (Arkin *et al.*, 2018) and other software (reviewed in Faria *et al.*, 2018).

An RBM employs an abstract language very similar to chemical equations capable of encoding millions of individual reactions (Danos *et al.*, 2007) depending on the strictness of rule definitions. Further, we chose to develop RBMs because of their modularity; also, they allow deterministic simulations through network generation (Blinov *et al.*, 2004, 2006; Hlavacek *et al.*, 2006), do not require the modeling of mass balances for all molecular species (network-free simulations) (Sneddon *et al.*, 2011), and permit stochastic simulations employing Gillespie's Stochastic Simulation Algorithm (Gillespie, 1976) or modifications (Danos *et al.*, 2007, 2008; Sneddon *et al.*, 2011). In addition to the mentioned features, RBMs are more readable than their counterparts, such as ODE-based models, which makes RBMs easier to review, inspect and correct collaboratively using version control tools, e.g. Git (Perez-Riverol *et al.*, 2016). Finally, rule-based languages were used previously to model automatically signaling pathways, e.g. with the software INDRA (Gyori *et al.*, 2017) and KAMI (Harmer *et al.*, 2019).

3 Materials and methods

3.1 Biological networks

Primary data employed were obtained from the EcoCyc database (Karp *et al.*, 2018b; Keseler *et al.*, 2017) with the help of an updated version of PythonCyc (https://github.com/latendre/PythonCyc) and PathwayTools version 24 (Karp *et al.*, 2019). The modified API is distributed freely from https://github.com/networkbiolab/PythonCyc and the Python Package Index with examples of use at https://github.com/networkbiolab/pythoncyc-notebooks.

Data were formatted as biological networks. For instance, genes are connected to their regulators in a canonical gene regulatory network (GRN) and we modified the network to connect transcription factors to DNA-binding sites and RNA polymerase-sigma factors (RNAP-σ) to promoters to obtain a sigma-specificity network. In the case of metabolic networks, we employed tripartite networks in which a reaction connects to the associated enzyme and metabolite(s) instead of the more common bipartite representation of reactions and metabolites. Finally, three types of interaction networks [protein–protein, protein–DNA binding sites (a GRN) and protein–metabolite] were formatted as collapsed hyper-graphs (Klamt *et al.*, 2009) to encode complexes, i.e. networks where (a group of) nodes connect to a group of nodes. We employed brackets to denote complexes, e.g. '[crp, crp]' representing the CRP homodimer and '[crp, CAMP, crp, CAMP]' to define the CRP-cyclic AMP dimer. The software *Atlas* disregards the order of components: '[crp, CAMP, crp, CAMP]' and '[crp, crp, CAMP, CAMP]' are equivalents. The networks employed in this work are in the Supplementary Material online and in the *examples* directory at https://github.com/networkbiolab/atlas.

3.2 Natural and synthetic GRNs used as examples

Natural GRNs representing data from *E.coli* were employed as examples and include the lactose, arabinose and fucose degradation operons (LacI, AraC and FucR regulons), the central carbon metabolism (Millard *et al.*, 2017) and all *E.coli* transporters and enzymes from the BioCyc database (Karp *et al.*, 2018b; Keseler *et al.*, 2017). In addition, we employed the regulation of gene expression for the *E.coli* sigma factors (*sigma factors model*) (Perez-Acle *et al.*, 2018). Primary data were completed with available information on genome architecture from Cho *et al.* (2009) and sigma factor specificity from Cho *et al.* (2014).

We modified the *sigma factors model* (Perez-Acle *et al.*, 2018) to exemplify the modeling of synthetic designs prior to experimentation. These modifications include the knockout of each sigma factor modeled and the incorporation of a promoter and/or a terminator to modify the rpoBC operon (Cho *et al.*, 2014). The two types of *in silico* modifications were made modifying the genome graph used as input for *Atlas*, adding a promoter or a terminator between the rpoB coding DNA sequence (CDS) and the rpoC ribosome binding site (RBS) or removing the CDS of each sigma factor preserving the natural promoters, RBS and terminators. In the case of the insertion of an rpoC promoter, the GRN was modified to incorporate the RNAP-σ specificity of the rpoB promoter.

3.3 Draft, simulation, curation and analysis of RBMs

Draft sub-models were obtained using biological networks as input and combined later in a divide-and-conquer modeling strategy. These ensemble models were employed for simulation, curation and calibration.

Models were simulated with the PySB interfaces for the SciPy ODE integrator (Virtanen *et al.*, 2020) and the Kappa Simulator v4.0 (KaSim) (Boutillier *et al.*, 2018). The ODE integration requires the enumeration of all components and individual reactions (network generation) (Blinov *et al.*, 2004, 2006; Hlavacek *et al.*, 2006). In any situation in which the network generation procedure took excessive time to finalize (set as a 5-min threshold), network-based simulations were replaced by network-free simulations employing KaSim.

Models were exported to the *kappa* language and analyzed with the Kappa Static Analyzer (KaSA) from the Kappa platform (Boutillier *et al.*, 2018) to perform reachability analysis (Danos *et al.*, 2008; Feret, 2007) after their reconstruction or any manual curation. In brief, RBMs describe a network of reactions, some of which could be dead rules due to the unavailability of preceding rules that synthesize reactants in the required form. Curation of the data was carried out manually, for instance, to remove duplications (e.g. gene products with two identical reactions but different metabolite names), ambiguities (e.g. names referring to a family of metabolites), lack of compartmental information [e.g. transport reactions in which substrate(s) and product(s) are the same metabolite but are located in different compartments], incorrect stoichiometry of reaction per enzymatic complex, missing gene regulations and others.

In the case of the *sigma factor model* and its *in silico* genetic modifications, we performed the following analysis. The dynamics of 1000 stochastic simulations for 100 units of time performed with KaSim were contrasted by employing the software edgeR (Chen *et al.*, 2014; Robinson *et al.*, 2010). Simulations were carried out with arbitrary rates at one event per unit time (also arbitrary). In the case of the addition of a promoter and/or a terminator to modify the rpoBC operon, the three resulting models were subject to calibration with the transcriptomics data of cold stress from Jozefczuk *et al.* (2010), GEO accession GSE20305, assuming the new networks describe the correct genome architecture. We calibrated the binding and unbinding rates of the RNAP-σ complexes to promoters and the RNA decay rates of the new models and the reference model by employing the software Pleione and the described strategy 3 with the χ^2 fitness function (Santibáñez *et al.*, 2019): 100 iterations, 100 models per iteration, selecting two models to recombine with a probability inverse to the ranking (see the Supplementary Material online for more details). After calibration, co-expression networks were constructed with ExpressionCorrelation (http://www.baderlab.org/Software/ExpressionCorrelation) for the average values of 1000 simulations. The ExpressionCorrelation employs Pearson's correlation coefficient and we selected absolute values higher than 0.95 for visualization.

The co-expression and other networks were visualized with the software Cytoscape v3.7.2 (Shannon *et al.*, 2003; Su *et al.*, 2014)

and models were visualized within Jupyter notebooks with the software pyViPR (Ortega and Lopez, 2020).

4 Results and discussion

4.1 Software overview and basic workflow

An overview of the *Atlas* software is depicted in Figure 1. The *atlas* module has functions to reconstruct independent models from genome graphs and protein–protein, protein–metabolites and protein–DNA interaction and metabolic networks. In addition, a specialized function could simultaneously employ data from the genome graph and from a sigma-specificity network (RNAPσ-promoters interaction network) to produce a model of bacterial regulation of gene expression. As models are independent, the module also provides a function to combine them, and functions to add regulatory relationships to gene expression rules; to get, remove, modify and add rules; and remove and get the current value of a parameter. After reconstruction, models require to set their parameters (if they were not provided as metadata in the networks) and to define the initial condition. The user of *Atlas* could choose from a variety of simulators and, finally, plot the results of simulations. In the case of stochastic simulations, the results include every simulation along with the mean and standard deviation. The user could export the model at

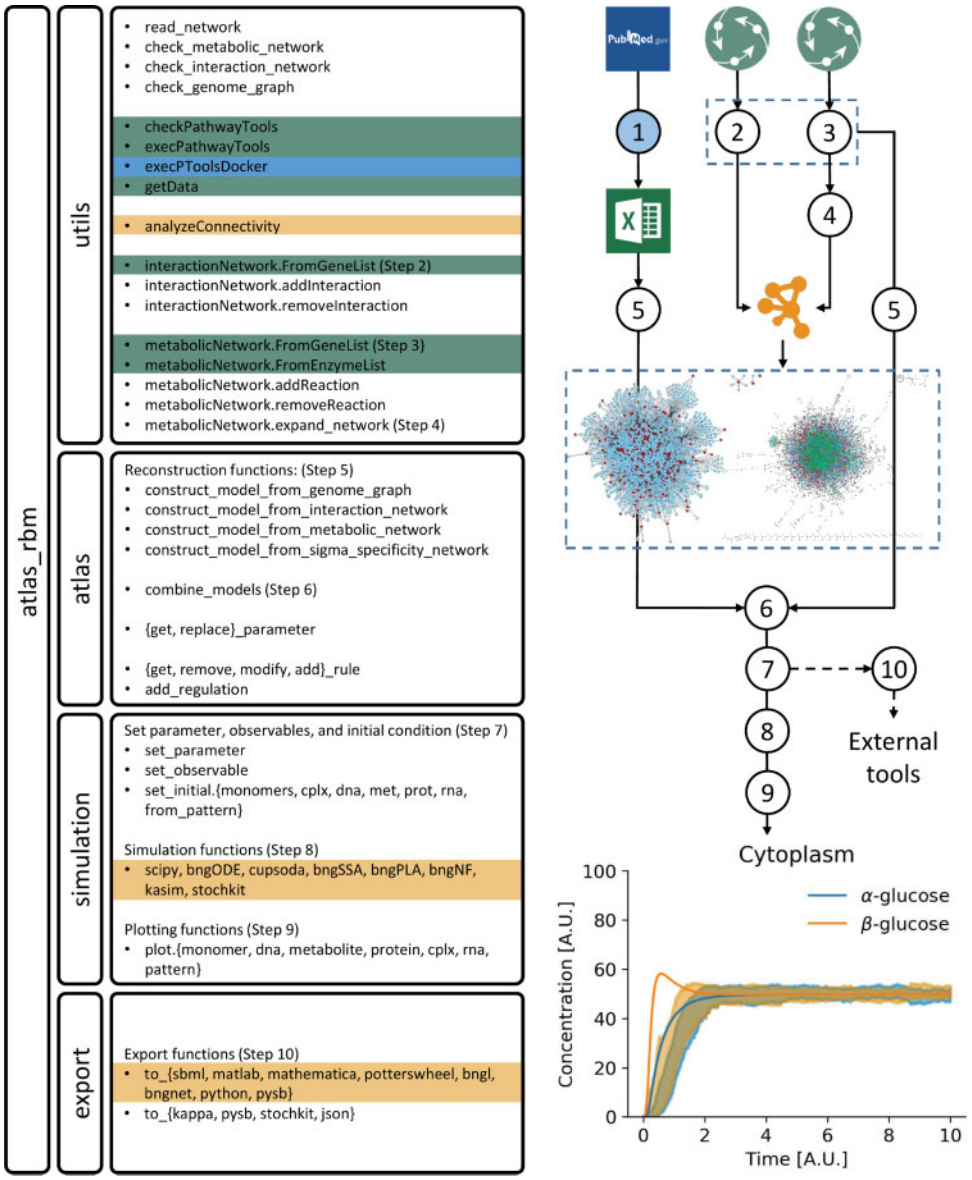

Fig. 1. Overview of the *Atlas* software and a typical workflow from gathering data to plot simulation results. (Left) *Atlas* is a python3 software divided into four modules. The *atlas* module has functions to reconstruct RBMs from biological networks in plain text. The *utils* module has functions to read and check networks (uniqueness of reactions and uniqueness of interactions), analyze the models produced by Atlas with KaSA and get information from locally installed BioCyc databases with the help of the PathwayTools software. The *simulation* module has functions to set parameters, observables and initials, simulate the model with a variety of software and plot the results. Finally, the *export* module has functions to export the model through any supported format in the PySB framework. Functions that require external software are highlighted: PathwayTools (green), Docker (blue) and BioNetGen/KappaTools (yellow). (Right) A typical workflow is divided into the following steps: (i) review and compile data from the literature using a spreadsheet software, (ii) obtain protein complex composition from PathwayTools with 'utils.interactionNetwork.FromGeneList', (iii) obtain metabolic data from PathwayTools with 'utils.metabolicNetwork.FromGeneList' or 'utils.metabolicNetwork.FromEnzymeList', (iv) expand the metabolic network (enzymes, substrates, products) to a source–target format for visualization, (v) reconstruct models matching the type of network and, optionally, add regulatory interactions of protein–DNA interactions to gene expression rules or correct rules if necessary, (vi) combine sub-models, (vii) set parameters, observables and the initial condition of model components, (viii) simulate the model and (ix) plot variables. Optionally (Step 10), export the model for simulation, analysis, or curation with external tools. Light blue denotes a manual step. A.U., arbitrary unit

any stage in a variety of formats, and employ external tools to simulate, curate or analyze the reconstructed model. Complementarily, *Atlas* provides utilitarian functions that are able to read and check the different networks, analyze the connectivity of the model and obtain data from the BioCyc databases (Caspi *et al.*, 2016; Karp *et al.*, 2018a). Data could be transformed and exported for visualization with Cytoscape (Shannon *et al.*, 2003; Su *et al.*, 2014) and models could be visualized within Jupyter notebooks with pyViPR (Ortega and Lopez, 2020).

An important note is the definition of the different components of the model (or agents). We defined five distinct *agents*: Proteins ('prot'), Metabolites ('met'), DNA ('dna'), RNA ('rna') and Complexes ('cplx'). A 'Complex' *agent* is an alias for complexes, such as the RNAP or the bacterial ribosome. All *agents* have a 'name' and a 'location' site for identification purposes. In addition, all components have interaction sites named 'dna', 'met', 'prot' and 'rna' that allow interaction with another *agent* of the matching type. The DNA and RNA *agents* have an additional identification site called 'type' to define their nature: promoter, RBS, CDS, terminator or binding site. Finally, proteins, DNA and RNA *agents* have two sites, named 'up' and 'dw', that enable the automated description of complexes of the same type. Two proteins interact in their 'up' and 'dw' sites because the unique 'prot' site per *agent* allows only dimerization. Following the definition of agents, *Atlas* is capable of writing complexes of any size and determining the correct internal links of components. See the Supplementary Material online for more details.

4.2 The lactose operon: modeling regulation of gene expression, transcription, translation and metabolism

We modeled a variety of metabolic networks of different sizes and complexities. The lactose model is composed of three enzyme-coding genes and one regulator, the arabinose-fucose model is composed of 13 enzyme-coding genes and two regulators, the *E.coli* central carbon metabolism model is composed of 200 enzyme-coding genes and the genome-scale metabolic model of *E.coli* is composed of 3596 transport and enzymatic reactions. To highlight the capabilities of *Atlas*, we describe in detail the modeling of the lactose metabolism because it is a common model of gene regulation with more than 50 years of biochemical information (Lewis, 2011).

The lactose operon from *E.coli* consists of three genes: the *β*-galactosidase gene lacZ, the lactose permease gene lacY (also known as lactose-proton symporter) and the galactoside O-acetyltransferase gene lacA. The EcoCyc database indicates that LacY can incorporate α-lactose, melibiose, lactulose, 3-O-galactosylarabinose and melibionate into the cell cytoplasm. Interestingly, the common colorimetric substrate ONPG (o-nitrophenyl-*β*-galactoside) is mentioned in the description for the lactose transport, but there is no inclusion of the reaction for LacY. Next, LacZ could metabolize lactose into *β*-galactose and glucose, lactulose into *β*-galactose and fructofuranose, and 3-O-galactosylarabinose into *β*-galactose and arabinose. Data from the literature (e.g. Huber *et al.*, 1981; Juers *et al.*, 2012) were used to complete the data derived from the EcoCyc database and were added manually to the network (Supplementary Tables S1 and S2 and Fig. S1) and the final network is depicted in Figure 2A (labels were omitted for visualization purposes). The modeling of similar corrections for other enzymes could be useful to understand the dynamic properties of metabolic pathways before experimental validation of the kinetics properties of each enzyme. In the case of lactose degradation, simulations of the curated metabolic network are shown in Figure 2B for the two anomers of glucose, galactose and allolactose produced from a source of 100 molecules (or an arbitrary concentration unit) of *β*-lactose. As expected, the degradation of lactose into glucose and galactose is complete, while mutarotation allows for the equilibrium of anomers. Although sugar mutarotations are very slow reactions, they are spontaneous and we included, in *Atlas*, the capacity to model non-enzymatic reactions as EcoCyc reports 145 'spontaneous' and three transport reactions without an identified gene.

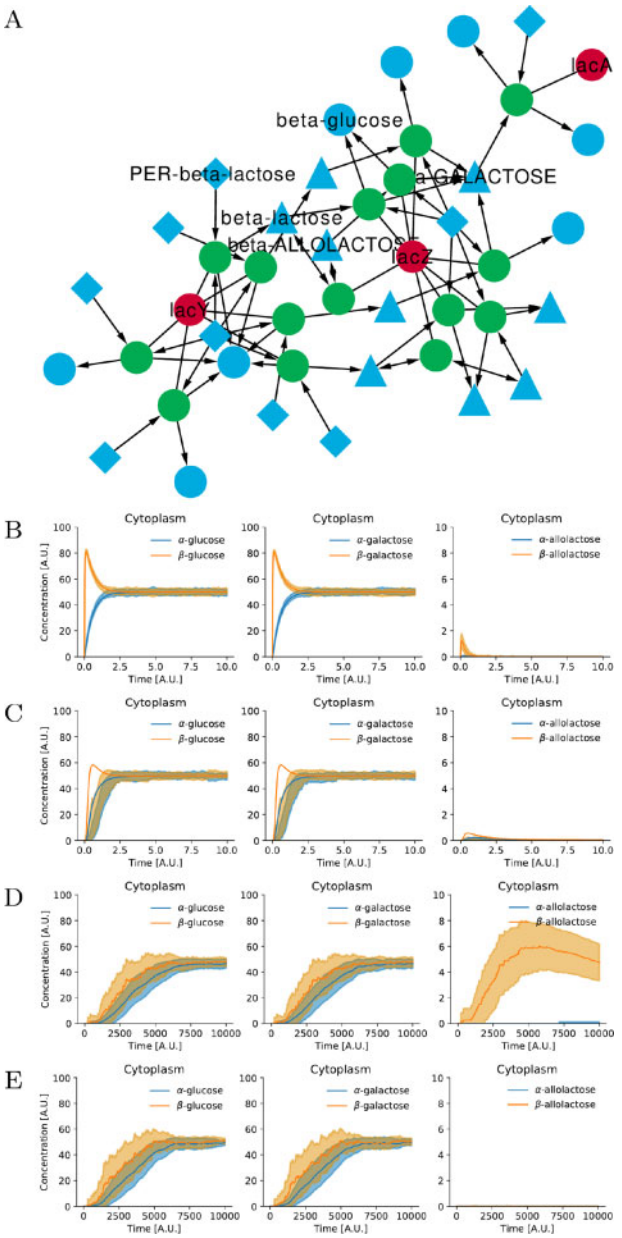

Fig. 2. Simulation of RBMs for the lactose degradation pathway. (**A**) Visualization of the curated metabolic network from the EcoCyc database. Nodes represent enzymes (red), reactions (green) and metabolites (cyan). Shapes represent substrates (diamonds), intermediates (triangles) and products (circles). Arrows show the reaction reversibility. (**B–E**) Total concentration of glucose, galactose, and allolactose produced from 100 molecules of lactose with hypothetical parameters. The continuous lines represent a deterministic simulation (SciPy, B and C) or the mean of 100 stochastic simulations (KaSim, D and E) with the area showing 1 SD. (B) Simulation of the metabolic network reconstructed from the network in (A). (C) Simulation of the metabolic and protein–protein interaction networks. (D and E) Simulation of the metabolism, protein–protein interactions, transcription, translation and gene expression regulation: (D) depicts the natural situation in which allolactose binds a lacI protein and is protected from degradation, and (**E**) shows a hypothetical situation in which allolactose cannot bind the lacI protein. Models at https://github.com/networkbiolab/atlas/tree/master/examples/lactose. A.U., arbitrary unit

Once we curated the metabolic network, we modeled the protein–protein interaction network that connects gene expression to the metabolism for reactions performed by protein complexes (Supplementary Fig. S2 and Table S4). For the *E.coli* lactose metabolism, the *β*-galactosidase is a homotetramer, the galactoside O-acetyltransferase is a homotrimer and the lactose-proton symporter acts as a monomer. We employed the collapsed hyper-network

representation to describe the protein–protein interactions from the literature or assumptions and automated the modeling of assembly processes. For instance, the assembly process for the β-galactosidase tetramer comes from dimers (Matsuura *et al.*, 2011), and we supposed the existence of a galactoside O-acetyltransferase dimer as pre-complex (Fowler *et al.*, 1985). Additionally, we took into consideration the reaction stoichiometry for each enzymatic complex. In this curation step, we identified whether reactions could happen independent of complex assembly (i.e. monomers are catalytically active) or if the protein complex is necessary for the catalytic activity *in vivo* (i.e. monomers are inactive). For the β-galactosidase complex, each subunit is catalytically active only when the tetramer is assembled (Li *et al.*, 2018). Similarly, the galactoside O-acetyltransferase active sites act independently of each other and because they are formed with residues from two adjacent monomers (Lewendon *et al.*, 1995; Wang *et al.*, 2002), the trimer was assumed to be the only active catalytic form. Figure 2C shows deterministic and stochastic simulations for an RBM including the assembly of protein complexes and the metabolic reactions. Interestingly, the deterministic and stochastic simulations disagree at the beginning of the dynamics, although they reached a similar steady-state. Because the stochastic simulation requires the assembly of enzyme complexes before performing any metabolic reaction, simulations showed a lag-phase, which is in contrast to the deterministic simulation.

Next, the model was coupled to a representation for transcription and translation in addition to the activity of transcription factors. We employed the Kappa BioBrick Framework (KBF) (Stewart and Wilson-Kanamori, 2011) and automated the modeling of rules describing bacterial transcription and translation. The KBF describes transcription and translation as a succession of rules. These rules describe the reversible docking of RNAP (ribosome) to a promoter (RBS), the sliding of the RNAP through the DNA and sliding of the ribosome through the RNA, and fall off from the terminator (RNAP) or the stop codon (ribosome). *Atlas* considers all promoters and terminators to write the rules described in the KBF. The transcription from the lactose operon is initiated at four promoters and terminated by two Rho-independent terminators (Fig. 3). Moreover, we modeled an internal promoter that drives transcription from the lacYA operon, although its importance *in vivo* is not clear (Zaslaver *et al.*, 2006). Employing the rules defined in KBF, we reconstructed a model for transcription and translation that considers the genomic architecture of the lactose operons and coupled it to the metabolic model presented previously (including protein assembly). Therefore, the resulting ensemble model requires only DNA (and the transcription/translation machinery) to produce the necessary proteins for metabolic activity.

Finally, the RBM representing the lactose metabolism was completed with a representation of transcriptional control. To model gene regulation, we added to the interaction network the LacI-allolactose and LacI-DNA binding site complexes. In the case of LacI, each dimer binds in tandem to one DNA-binding site, and two dimers could dimerize, forming a DNA loop that impedes the binding of RNAP-σ to promoters or initiates transcription (Rutkauskas *et al.*, 2009). To inactivate LacI, free proteins bind allolactose, which seems to impede the binding of LacI-allolactose to DNA-binding sites (Lewis, 2005). In principle, the modeling of DNA-binding protein interactions requires one rule per transcription factor, as we could ignore differences in the rates of DNA-protein kinetics. However, the different affinities, the genomic architecture and the transcription factor mechanisms [reviewed in van *Hijum et al.* (2009)] encouraged the development of another approach. To do so, overlapping DNA-binding sites and other genomic features (represented in Fig. 3) were defined as a collapsed hyper-network similar to what was done for protein complexes (Supplementary Table S5). Figure 2D shows the results of simulations in which allolactose binds free lacI proteins, while Fig. 2E shows the simulation from a hypothetical situation in which free lacI proteins cannot bind allolactose. The difference between both situations was modest and showed an earlier rise of the glucose and galactose concentration around 100 units of time when allolactose could bind lacI proteins (Supplementary Fig. S3). Because allolactose binds free lacI proteins, the release of lacI proteins freeing the promoter occurred in both models.

Although system parameters could be found in databases or calibrated [e.g. with pyBioNetFit (Mitra *et al.*, 2019) or *Pleione* (Santibáñez *et al.*, 2019)], the results show that modeling of RBMs for metabolism, protein complex assembly, transcription, translation and regulation of gene expression can be done in an automated manner, facilitating deterministic and stochastic simulation. Parameters employed for simulation are detailed in Supplementary Tables S2–S5 and a benchmark is detailed in Supplementary Table S6.

4.3 Modeling natural and synthetic transcriptional control: the sigma factors model

We later addressed the modeling of RNAP-σ assembly and transcriptional control of its expression mediated by the activity of sigma factors. Compared to eukaryotes, bacteria have only one RNAP and different sigma factors that confer promoter specificity (Mauri and Klumpp, 2014). The bacterium *E.coli* has seven sigma factors that interact physically with the core RNAP to form holoenzymes. The purpose here is to present how to model transcription control as the RBM is presented and calibrated elsewhere (Perez-Acle *et al.*, 2018; Santibáñez *et al.*, 2019) and to employ it to model synthetic transcriptional control. Also, *Atlas* models a molecular step in bacterial transcription disregarded in KBF (Stewart and Wilson-Kanamori, 2011): the sigma factor is released from holoenzymes when transcription is initiated (Mauri and Klumpp, 2014).

We modeled the holoenzymes binding to promoters as if those interactions were the binding of any transcription factor to their cognate DNA-binding sites. To do so, we considered the RNAP-σ specificity (Supplementary Table S7) and the genome architecture (Supplementary Table S8) simultaneously, two features that we employed separately for the modeling of DNA–protein interactions and transcription. Both networks are represented in Figure 4A (a canonical GRN) and B (an extended network to show the considered genome architecture). The resulting model describes holoenzymes explicitly as a complex of five proteins instead of a unique agent modeling the RNAP complex employed in the lactose model. Results for the dynamics of the described GRN are shown in Figure 4C and D for a hypothetical case of only RNA synthesis without mRNA degradation. It can be seen that gene expression shows

Fig. 3. Genomic organization of the *E.coli* lactose operon. The lactose operon shows four promoters controlled independently by the repressor LacI, the activator–repressor CRP, the repressor H-NS and the repressor MarA. An internal promoter and two terminators contribute to the expression dynamics of enzymes and the transporter. Image from the EcoCyc website (Keseler *et al.*, 2017, https://ecocyc.org/gene? orgid=ECOLI&id=EG10527#tab=TU)

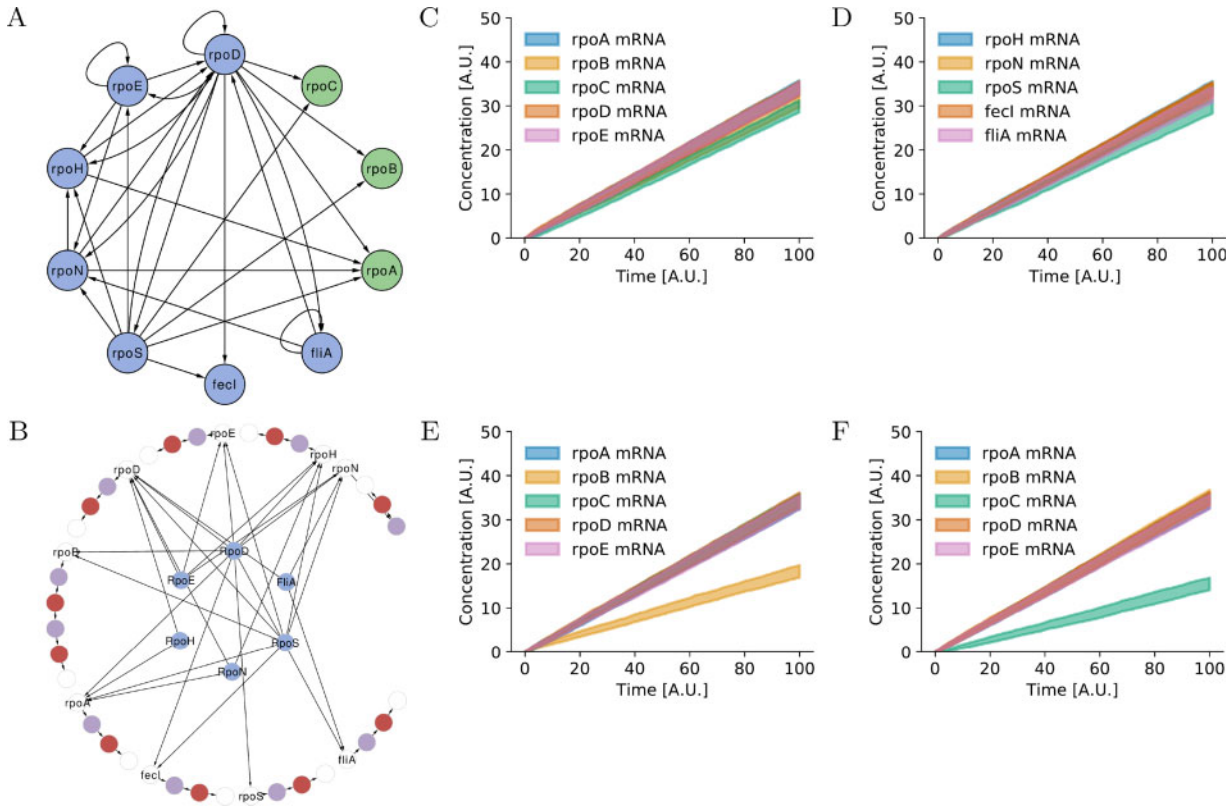

Fig. 4. Stochastic simulation of the *E.coli* sigma factor GRN. (**A**) Visualization of the curated GRN from the EcoCyc database. The light blue nodes represent the seven sigma factor and the green nodes represent the three RNAP subunits encoding genes. Arrows represent the positive regulation of transcription determined from sigma factor specificity for promoters. (**B**) Extension of the GRN to encode the genomic architecture of the 10 considered genes. The rpoB and rpoC (left side of the outer ring) form a single operon. Labeled white nodes are the promoters, purple nodes are the RBSs, red nodes are the CDS and unlabeled white nodes are the terminators. (**C–F**) Mean of 100 stochastic simulations (KaSim) and 1 SD from the mean. (C and D) Stochastic simulation of the natural genomic architecture and regulatory interactions. (E) Stochastic simulation for the network modified with an *in silico* internal rpoC promoter. (F) Stochastic simulation for the network modified with an *in silico* internal rpoB terminator. Models at https://github.com/networkbiolab/atlas/tree/master/examples/sigma-model. A.U., arbitrary unit

similar rates, though the results are influenced by the parameter values and the initial condition for proteins.

The use of *Atlas* is not restricted to natural networks and allows for the modeling of a different genomic arrangement of genes. One purpose of such a procedure is to assess differences in mRNA and other cell component dynamics with the final goal of computational-aided design before experimental evaluation. For the sigma model, we modeled three variants that modified the rpoBC operon architecture. Those variants are: (i) the incorporation of an internal promoter between rpoB and rpoC genes allowing for the interaction of an RNAP-σ complex, (ii) the incorporation of an internal terminator allowing for the falloff of the RNAP and (iii) the incorporation of both. Our simulations showed that the incorporation of a promoter for rpoC reduced the synthesis rate for rpoB due to reduced RNAP availability for its promoter (Fig. 4E). This, in turn, is determined by the model parameters and initial condition. On the other hand, the addition of the internal terminator reduced the synthesis rate for rpoC (Fig. 4F) due to a reduced probability of continuing RNA elongation from rpoB into rpoC. Finally, simultaneous modifications showed no changes in RNA synthesis rates due to compensation of falling off RNAPs from the rpoB terminator and the interaction of RNAP-σ holoenzymes to the synthetic rpoC promoter (Supplementary Fig. S4B), and showed similar expression rates as the situation of independent rpoB and rpoC operons (Supplementary Fig. S4D).

More realistic stochastic simulations were performed with the *sigma factors model* extended to model RNA degradation as unimolecular decay (Perez-Acle *et al.*, 2018). In contrast to the simulation results shown in Figure 4E and F, and Supplementary Figure S4, the *in silico* variants were calibrated as if the new models represent the natural genomic architectures. Also, we performed an indirect comparison of mRNA quantities using Pearson's correlation coefficient to compare mRNA dynamics for the average of simulations. We report correlations higher than 0.95 as the absolute value in a co-expression network (Supplementary Fig. S5). The expression profiles for rpoB and rpoC remained correlated in all variants, in contrast to a correlation coefficient of 0.57 determined from the original data. A complete explanation is tailored to the ability of the performed calibration to find parameter values that most closely resemble the experimental data for an unnatural transcriptional network and delineates the need for (cell-free) experiments to accurately measure RNA synthesis rates in modified genomic contexts.

Finally, we performed *in silico* knockout experiments. Comparisons employing the edgeR software (Chen *et al.*, 2014; Robinson *et al.*, 2010) and a threshold for the false discovery rate (FDR) of 0.05 showed that the deletion of rpoD and rpoS had the most impact in mRNA synthesis, while the other deletions did not show differential expressed genes. The knockout of rpoD impedes the expression of rpoS (Fig. 4B) while we observed lower expression for fliA and fecI and higher expression for rpoA, rpoE, rpoH and rpoN compared to the reference model. In turn, the knockout of rpoS showed lower expression for rpoB and fecI and higher expression for fliA. The determined fold change and FDR values are presented in Supplementary Tables S9 and S10, respectively. However, simulations were done to highlight the capability of *Atlas* to model different genetic modifications and parameters did not reflect any experimentally determined rate. Also, the models did not incorporate degradation rules for mRNAs, and an extension of the model to synthesize and degrade proteins will allow for the detailed modeling of *in silico* designs and the comparison of simulations to

experimental data from synthetic constructs employing cell-free translation-transcription technologies (Borkowski *et al.*, 2018).

5 Conclusion

Mathematical and computational modeling is often viewed as a specialized task. To facilitate modeling, we automated the development of RBMs, as these types of models show simulation flexibility, a reasonable degree of readability, modularity for integrative modeling and good simulation scalability.

Atlas produces sub-models from genome graphs, and protein–protein, protein–metabolites and protein–DNA interaction and metabolic networks. We developed, in this work, a divide-and-conquer strategy supported by the modularity of RBMs, as it is the pathway for the development of whole-cell models (Szigeti *et al.*, 2018). The software produces RBMs for the PySB framework (Lopez *et al.*, 2013) and *rules* can be added in any order while PySB checks on whether new rules are compatible with the current model. In addition, PySB could export to *kappa* language and we employed the KaSA software (Boutillier *et al.*, 2018) to further assess the coherence of the developed RBMs. Simulation of RBMs could be done within PySB and calibration of exported models could be performed with pyBioNetFit (Mitra *et al.*, 2019, only BNGL models) or *Pleione* (Santibáñez *et al.*, 2019, BNGL and *kappa* models) to compare the reconstructed models with experimental data or available models.

Atlas contrasts with available software because it lacks a graphical interface [e.g. RuleBender (Smith *et al.*, 2012) and VirtualCell (Blinov *et al.*, 2017)], although the user could employ *Atlas* within a Jupyter notebook and use pyViPR (Ortega and Lopez, 2020) to visualize the model structure. Also, *Atlas* relies on the user to obtain formatted data to model interactions, in contrast to INDRA (Gyori *et al.*, 2017), which can use natural language processing to read information and reconstruct models. In turn, *Atlas* can model metabolism, transcription and translation, as well as widespread protein–protein interactions found in signaling pathways that INDRA (Gyori *et al.*, 2017) and KAMI (Harmer *et al.*, 2019) can model.

Finally, the models and the *Atlas* software are extensible, for instance, to model cooperative behavior not currently supported. The utilization of the law of mass action for the metabolic network (and other reactions) limits the utility of the resulting RBMs in the current form, but exporting to BNGL or *kappa* leverages this imposition, as they support mathematical expressions as reaction rates. However, we expect to extend *Atlas* to consider enzyme–metabolite interactions and describe the detailed mechanisms of enzyme reactions (Saa and Nielsen, 2017) and allosteric regulations of metabolic activity, as well as to model the assembly of ribosomes (Davis *et al.*, 2016; Gupta and Culver, 2014; Shajani *et al.*, 2011). Notably, *Atlas* is already compatible with metabolic and interaction data from eukaryotes and we obtained a model from data for 1991 metabolic reactions of the yeast *Saccharomyces cerevisiae* from BioCyc. In addition, we expect further interoperability with INDRA models of signaling pathways to model protein modifications, such as phosphorylation and the collaboration from researchers. With collaboration in mind, we shared the developed models in this work at https://github.com/networkbiolab/atlas/tree/master/examples.

Acknowledgements

HPC@CGB-UM: this research was partially supported by the computing infrastructure of the Centro de Genómica y Bioinformática, Universidad Mayor, Chile. We acknowledge to Nicole Betti, subdirector of postgrade student affairs of the School of Engineering, Pontificia Universidad Católica de Chile, that without her support this manuscript would not been possible.

Funding

This work was supported by Agencia Nacional de Investigación y Desarrollo, Ministerio de Ciencia, Tecnología, Conocimiento e Innovación, Chile [PCHA-2014-21140377 to R.S., 1190074 to D.G. and 1181089 to A.J.M.M.].

Conflict of Interest: none declared.

Data availability

The data underlying this article is available in Zenodo, at http://doi.org/10.5281/zenodo.4362673. The datasets were derived from sources in the public domain: Gene Expression Omnibus, https://www.ncbi.nlm.nih.gov/bioproject/PRJNA125345; and EcoCyc, https://ecocyc.org/.

References

Agren,R. *et al.* (2013) The RAVEN toolbox and its use for generating a genome-scale metabolic model for *Penicillium chrysognum*. *PLoS Comput. Biol.*, **9**, e1002980.

Arkin,A.P. *et al.* (2018) KBase: the United States department of energy systems biology knowledgebase. *Nat. Biotechnol.*, **36**, 566–569.

Blinov,M.L. *et al.* (2004) BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, **20**, 3289–3291.

Blinov,M.L. *et al.* (2006) A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. *Biosystems*, **83**, 136–151.

Blinov,M.L. *et al.* (2017) Compartmental and spatial rule-based modeling with virtual cell. *Biophys. J.*, **113**, 1365–1372.

Borkowski,O. *et al.* (2018) Cell-free prediction of protein expression costs for growing cells. *Nat. Commun.*, **9**, 1–11.

Boutillier,P. *et al.* (2018) The Kappa platform for rule-based modeling. *Bioinformatics*, **34**, i583–i592.

Carrera,J. and Covert,M.W. (2015) Why build whole-cell models? *Trends Cell Biol.*, **25**, 719–722.

Caspi,R. *et al.* (2016) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.*, **44**, D471–D480.

Chen,Y. *et al.* (2014) Differential expression analysis of complex RNA-seq experiments using edgeR. In: Datta, S. and Nettleton, D. (ed.) *Statistical Analysis of Next Generation Sequencing Data*. Springer, Cham, Switzerland, pp. 51–74.

Cho,B.K. *et al.* (2009) The transcription unit architecture of the *Escherichia coli* genome. *Nat. Biotechnol.*, **27**, 1043–1049.

Cho,B.K. *et al.* (2014) Genome-scale reconstruction of the sigma factor network in *Escherichia coli*: topology and functional states. *BMC Biol.*, **12**, 4–11.

Costa,R.S. *et al.* (2016) Kinetic modeling of cell metabolism for microbial production. *J. Biotechnol.*, **219**, 126–141.

Covert,M.W. *et al.* (2008) Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. *Bioinformatics*, **24**, 2044–2050.

Danos,V. *et al.* (2007) *Rule-Based Modelling of Cellular Signalling*. Lecture Notes in Computer Science. LNCS. Vol. **4703**, pp. 17–41.

Danos,V. *et al.* (2008) *Abstract Interpretation of Cellular Signalling Networks*. Lecture Notes in Computer Science. LNCS. Vol. **4905**, pp. 83–97.

Davis,J.H. *et al.* (2016) Modular assembly of the bacterial large ribosomal subunit. *Cell*, **167**, 1610–1622.E15.

Dias,O. *et al.* (2015) Reconstructing genome-scale metabolic models with merlin. *Nucleic Acids Res.*, **43**, 3899–3910.

Faria,J.P. *et al.* (2018) Methods for automated genome-scale metabolic model reconstruction. *Biochem. Soc. Trans.*, **46**, 931–936.

Feret,J. (2007) Reachability analysis of biological signalling pathways by abstract interpretation. In: *AIP Conference Proceedings*, Vol. **963**, Corfu, Greece, pp. 619–622.

Foster,C.J. *et al.* (2019) From *Escherichia coli* mutant 13C labeling data to a core kinetic model: a kinetic model parameterization pipeline. *PLoS Comput. Biol.*, **15**, e1007319.

Fowler,A.V. *et al.* (1985) The amino acid sequence of thiogalactoside transacetylase of *Escherichia coli*. *Biochimie*, **67**, 101–108.

Fredens,J. *et al.* (2019) Total synthesis of *Escherichia coli* with a recoded genome. *Nature*, **569**, 514–518.

Gillespie,D.T. (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, **22**, 403–434.

Grimbs,A. *et al.* (2019) A system-wide network reconstruction of gene regulation and metabolism in *Escherichia coli*. *PLoS Comput. Biol.*, **15**, 1–29.
Gupta,N. and Culver,G.M. (2014) Multiple in vivo pathways for *Escherichia coli* small ribosomal subunit assembly occur on one pre-rRNA. *Nat. Struct. Mol. Biol.*, **21**, 937–943.
Gyori,B.M. *et al.* (2017) From word models to executable models of signaling networks using automated assembly. *Mol. Syst. Biol.*, **13**, 954.
Hädicke,O. and Klamt,S. (2017) EColiCore2: a reference network model of the central metabolism of *Escherichia coli* and relationships to its genome-scale parent model. *Sci. Rep.*, **7**, 39647.
Harmer,R. *et al.* (2019) Bio-curation for cellular signalling: the KAMI project. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **16**, 1562–1573.
Henry,C.S. *et al.* (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat. Biotechnol.*, **28**, 977–982.
Hernández-Prieto,M.A. *et al.* (2014) Toward a systems-level understanding of gene regulatory, protein interaction, and metabolic networks in cyanobacteria. *Front. Genet.*, **5**, 1–18.
Hlavacek,W.S. *et al.* (2006) Rules for modeling signal-transduction systems. *Sci. STKE*, **2006**, re6.
Huber,R.E. *et al.* (1981) The anomeric specificity of β-galactosidase and lac permease from *Escherichia coli*. *Can. J. Biochem.*, **59**, 100–105.
Jozefczuk,S. *et al.* (2010) Metabolomic and transcriptomic stress response of *Escherichia coli*. *Mol. Syst. Biol.*, **6**, 1–16.
Juers,D.H. *et al.* (2012) LacZ β-galactosidase: structure and function of an enzyme of historical and molecular biological importance. *Protein Sci.*, **21**, 1792–1807.
Karp,P.D. *et al.* (2018a) The BioCyc collection of microbial genomes and metabolic pathways. *Brief. Bioinform.*, **20**, 1085–1093.
Karp,P.D. *et al.* (2018b) The EcoCyc database. *EcoSal Plus*, **8**, 1–19.
Karp,P.D. *et al.* (2019) Pathway Tools version 23.0 update: software for pathway/genome informatics and systems biology. *Brief. Bioinform.*, bbz104.
Karr,J.R. *et al.* (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, **150**, 389–401.
Karr,J.R. *et al.* (2015) The principles of whole-cell modeling. *Curr. Opin. Microbiol.*, **27**, 18–24.
Keseler,I.M. *et al.* (2017) The EcoCyc database: reflecting new knowledge about *Escherichia coli* K-12. *Nucleic Acids Res.*, **45**, D543–D550.
Klamt,S. *et al.* (2009) Hypergraphs and cellular networks. *PLoS Comput. Biol.*, **5**, e1000385.
Lewendon,A. *et al.* (1995) Structural and mechanistic studies of galactoside acetyltransferase, the *Escherichia coli* LacA gene product. *J. Biol. Chem.*, **270**, 26326–26331.
Lewis,M. (2005) The lac repressor. *C R Biol.*, **328**, 521–548.
Lewis,M. (2011) A tale of two repressors. *J. Mol. Biol.*, **409**, 14–27.
Li,X. *et al.* (2018) Bottom-up single-molecule strategy for understanding subunit function of tetrameric β-galactosidase. *Proc. Natl. Acad. Sci. USA*, **115**, 8346–8351.
Long,C.P. and Antoniewicz,M.R. (2019) Metabolic flux responses to deletion of 20 core enzymes reveal flexibility and limits of *E. coli* metabolism. *Metab. Eng.*, **55**, 249–257.
Lopez,C.F. *et al.* (2013) Programming biological models in Python using PySB. *Mol. Syst. Biol.*, **9**, 646.
Macklin,D.N. *et al.* (2020) Simultaneous cross-evaluation of heterogeneous *E. coli* datasets via mechanistic simulation. *Science*, **369**, eaav3751.
Matsuura,T. *et al.* (2011) Kinetic analysis of β-galactosidase and β-glucuronidase tetramerization coupled with protein translation. *J. Biol. Chem.*, **286**, 22028–22034.
Mauri,M. and Klumpp,S. (2014) A model for sigma factor competition in bacterial cells. *PLoS Comput. Biol.*, **10**, e1003845.
Millard,P. *et al.* (2017) Metabolic regulation is sufficient for global and robust coordination of glucose uptake, catabolism, energy production and growth in *Escherichia coli*. *PLoS Comput. Biol.*, **13**, e1005396.
Mitra,E.D. *et al.* (2019) PyBioNetFit and the biological property specification language. *iScience*, **19**, 1012–1036.
Ortega,O.O. and Lopez,C.F. (2020) Interactive multiresolution visualization of cellular network processes. *iScience*, **23**, 100748.
Perez-Acle,T. *et al.* (2018) Stochastic simulation of multiscale complex systems with PISKaS: a rule-based approach. *Biochem. Biophys. Res. Commun.*, **498**, 342–351.
Perez-Riverol,Y. *et al.* (2016) Ten simple rules for taking advantage of Git and GitHub. *PLoS Comput. Biol.*, **12**, e1004947.
Raj,A. and van Oudenaarden,A. (2008) Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell*, **135**, 216–226.
Rees-Garbutt,J. *et al.* (2020) Designing minimal genomes using whole-cell models. *Nat. Commun.*, **11**, 836.
Regev,A. *et al.*; Human Cell Atlas Meeting Participants. (2017) The human cell atlas. *Elife*, **6**, e27041.
Robinson,M.D. *et al.* (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.
Rutkauskas,D. *et al.* (2009) Tetramer opening in LacI-mediated DNA looping. *Proc. Natl. Acad. Sci. USA*, **106**, 16627–16632.
Saa,P.A. and Nielsen,L.K. (2017) Formulation, construction and analysis of kinetic models of metabolism: a review of modelling frameworks. *Biotechnol. Adv.*, **35**, 981–1003.
Sanghvi,J.C. *et al.* (2013) Accelerated discovery via a whole-cell model. *Nat. Methods*, **10**, 1192–1195.
Santibáñez,R. *et al.* (2019) Pleione: a tool for statistical and multi-objective calibration of Rule-based models. *Sci. Rep.*, **9**, 15104.
Shajani,Z. *et al.* (2011) Assembly of bacterial ribosomes. *Annu. Rev. Biochem.*, **80**, 501–526.
Shannon,P. *et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
Smith,A.M. *et al.* (2012) RuleBender: integrated modeling, simulation and visualization for rule-based intracellular biochemistry. *BMC Bioinformatics*, **13**, S3.
Sneddon,M.W. *et al.* (2011) Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods*, **8**, 177–183.
Stewart,D. and Wilson-Kanamori,J.R. (2011) Modular modelling in synthetic biology: light-based communication in *E. coli*. *Electron. Notes Theor. Comput. Sci.*, **277**, 77–87.
Su,G. *et al.* (2014) Biological network exploration with Cytoscape 3. *Curr. Protoc. Bioinformatics*, **47**, 8.13.1–8.13.24.
Szigeti,B. *et al.* (2018) A blueprint for human whole-cell modeling. *Curr. Opin. Syst. Biol.*, **7**, 8–15.
van Hijum,S.A.F.T. *et al.* (2009) Mechanisms and evolution of control logic in prokaryotic transcriptional regulation. *Microbiol. Mol. Biol. Rev.*, **73**, 481–509.
Virtanen,P. *et al.*; SciPy 1.0 Contributors. (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, **17**, 261–272.
Wang,H. *et al.* (2018) RAVEN 2.0: a versatile toolbox for metabolic network reconstruction and a case study on Streptomyces coelicolor. *PLoS Comput. Biol.*, **14**, e1006541.
Wang,X.G. *et al.* (2002) Structure of the lac operon galactoside acetyltransferase. *Structure*, **10**, 581–588.
Zaslaver,A. *et al.* (2006) A comprehensive library of fluorescent transcriptional reporters for *Escherichia coli*. *Nat. Methods*, **3**, 623–628.