





A scalable, open-source implementation of a large-scale mechanistic model for single cell proliferation and death signaling

Cemal Erdem ¹✉, Arnab Mutsuddy¹, Ethan M. Bensman², William B. Dodd ¹, Michael M. Saint-Antoine³, Mehdi Bouhaddou⁴, Robert C. Blake⁵, Sean M. Gross⁶, Laura M. Heiser ⁶, F. Alex Feltus ^{7,8,9} & Marc R. Birtwistle^{1,10}✉

Mechanistic models of how single cells respond to different perturbations can help integrate disparate big data sets or predict response to varied drug combinations. However, the construction and simulation of such models have proved challenging. Here, we developed a python-based model creation and simulation pipeline that converts a few structured text files into an SBML standard and is high-performance- and cloud-computing ready. We applied this pipeline to our large-scale, mechanistic pan-cancer signaling model (named SPARCED) and demonstrate it by adding an IFN γ pathway submodel. We then investigated whether a putative crosstalk mechanism could be consistent with experimental observations from the LINCS MCF10A Data Cube that IFN γ acts as an anti-proliferative factor. The analyses suggested this observation can be explained by IFN γ -induced SOCS1 sequestering activated EGF receptors. This work forms a foundational recipe for increased mechanistic model-based data integration on a single-cell level, an important building block for clinically-predictive mechanistic models.

¹Department of Chemical & Biomolecular Engineering, Clemson University, Clemson, SC, USA. ²Computer Science, School of Computing, Clemson University, Clemson, SC, USA. ³Center for Bioinformatics and Computational Biology, University of Delaware, Newark, DE, USA. ⁴Department of Cellular and Molecular Pharmacology, University of California San Francisco, San Francisco, CA, USA. ⁵Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA. ⁶Department of Biomedical Engineering, Oregon Health & Science University, Portland, OR, USA. ⁷Department of Genetics and Biochemistry, Clemson University, Clemson, SC, USA. ⁸Biomedical Data Science and Informatics Program, Clemson University, Clemson, SC, USA. ⁹Center for Human Genetics, Clemson University, Clemson, SC, USA. ¹⁰Department of Bioengineering, Clemson University, Clemson, SC, USA. ✉email: cemalerdem@gmail.com; mbirtwi@clemson.edu

The ever-increasing availability and accumulation of FAIR¹ (findable, accessible, interoperable, and reproducible) and big (omics) datasets requires new computational methods and models to integrate, analyze, and interpret the underlying information^{2–4}. How can we leverage the totality of available information not only to learn more about biology but also to make predictions, especially those that are clinically relevant? Advances in statistical and machine learning approaches enable (mostly) data-driven exploration and hypothesis generation from big datasets^{5–8}. Trained on features of the input dataset(s), such models can be used for, as just a few examples, to predict drug responses^{9–11} or decide tumor type/stage^{12–15}. Although transformative, such machine learning and statistical models have shortcomings. Most notably, they often fail to explain predicted outcomes with detailed mechanistic reasoning^{16–20} – a major scientific gap and a roadblock to reconciling and integrating such models.

Besides such “black-box” modeling approaches, an alternative and complementary vehicle for data integration are so-called “mechanistic models”²⁰. Mechanistic models provide an interpretable integration of different data types, because they have explicitly modeled biophysical correlates, while enabling further exploration for underlying logic behind heterogeneous, nonlinear, and often unintuitive relationships across big datasets²¹. If mechanistic models are available towards the whole-genome or whole-single-cell scale, one can start to predict complex, multi-network, and emergent cellular behaviors^{22,23}, elucidate phenotypic responses to multiple perturbations^{24,25}, tailor and train on patient-specific data for personalized, pharmacologic decision making^{26,27}, or use them as “data integrators” for data consistency checking²⁸. However, most published mechanistic models are “small” scale; built for single pathways with a handful of genes, meant to interpret a single dataset^{29–38}. Such small-scale mechanistic models provided important insights into processes such as yeast response to pheromones³⁵, *lac* operon regulation in *E. coli*³⁴, or phenotypic responses to different ligand stimulations²⁹. However, the limited scope of small-scale models means they inherently will struggle to integrate multiple datasets. Large-scale mechanistic models^{23,39–41}, on the other hand, can provide a more extensive representation of cellular interactions and are thus well-poised for data integration that complement shortcomings of machine learning approaches.

One of the many ways of mechanistic model construction is the use and modification of existing models by inserting new species or interactions to explain new experimental observations^{38,42,43}. Model merging, the act of stitching pre-existing models together, is an extension of this method for creating larger models. However, such an approach requires extensive detail checking and harmonizing species/parameter definitions. Often, unfortunately, sufficient annotation is not provided which makes this task harder. Moreover, while most mechanistic models are comprised of ordinary differential equations (ODEs), many large-scale models require multiple sub-modules of different mathematical formalisms. For example, metabolic processes are usually described by steady-state flux-balance models^{44,45}, gene expression events are stochastic^{46–48}, and protein signaling events are represented by a system of ODEs^{29,30,38}. Thus, sorting out a single platform for different modeling formalisms to create a large-scale model is a daunting task. It is so far only achieved by creating highly custom-structured and custom-coded model-agglomerates that are not well-suited to further alterations or re-use^{23,40}. The latter, Bouhaddou2018 pan-cancer model⁴⁰, is previously published by our group to study single-cell responses to mitogens and drugs.

A second way of constructing models is to build them bottom-up by writing out every reaction one by one. In this regard, rule-based modeling (RBM) provides an innovative approach⁴⁹. RBM software, such as BioNetGen^{50,51}, Kappa⁵², and PySB⁵³, enables researchers to

write “rules” for repeated reaction events following specific patterns. RBM software then creates the reaction network by propagating the rules from the initial set of species. Although RBM revolutionized large-scale model construction by minimizing manual equation scripting (i.e., writing out every differential equation), some limitations exist. First, it can generate a vast (even infinite) number of reactions from a small set of rules (usually called the curse of combinatorial complexity). This makes interpreting, analyzing, and debugging such models cumbersome, if possible. Tools like NFsim⁵⁴ can overcome such problems by simulating events based on the rules rather than a priori generating the entire reaction network. Thus, such software becomes advantageous when a small number of rules create a very large number of reactions, e.g., polymerization, aggregation, or multi-site phosphorylation⁵⁵. However, such network-free simulators typically require an explicit representation of every molecule in the system, which dramatically increases the computational cost and renders such methods inefficient for large-scale mechanistic models. Secondly, current RBM implementations dictate that reactions taking place via the same rule have the same rate constant parameter values. Often, allostery or site cooperativity precludes this simplifying assumption, leading to manually writing out every such reaction in the model (or writing one rule for each reaction), which then obviates the advantages of RBM. Finally, with its capability of capturing biological complexity via simple rules, the RBM concept is quite powerful but additional efforts are needed to enable merging of existing non-rule-based models, creating a mixture of different modeling formats (i.e., mixed-grain modeling), and defining different simulation settings (i.e., hybrid modeling = deterministic + stochastic parts).

Regardless of how a large-scale model is constructed, it should have certain properties for FAIRness (findable, accessible, interoperable, and reproducible) and re-useability^{56–58}. Porubsky et al.⁵⁷ recently summarized the *best modeling practices* and reinforced: providing metadata/annotations and model creation steps/files (Practices 1–5), using standard and cross-platform model files (Practice 3), and open-source, license-free, version-controlled, and reproducible model dissemination (Practices 8–9). As the size of the model increases, conforming to modeling standards (e.g., simulation type, simulation speed, software to use, scripting package to use, algorithm to use) gets harder. That is why most of the large-scale (many genes or whole-cell) models are necessarily custom-structured, are composed of multiple submodules, or are lacking sufficient annotations and metadata (e.g., ENSEMBL or HGNC identifiers)^{23,39–41}. These custom-made models also do not yet follow a single standard format, a key property for easy distribution, re-use, and model merging and expansion with other models. The SBML (Systems Biology Markup Language) format^{59,60} offers a long-established and well-defined way of specifying annotated model structures, with an explicit and structured definition of each element of a mechanistic model (species, reactions, volumes, initial concentrations, parameters, rules, events, equations). SBML is an extensible, machine-readable markup language and not a simple text file. SBML has interfaces and packages in most programming languages (like Python, C++, Perl) and can be imported by most software (Python, MATLAB, COPASI⁶¹, Virtual Cell⁶², and another ~300 packages). However, it is non-trivial to write thousands of reactions in SBML standards, directly or with available GUI-based software. To circumvent this problem, there are efforts to convert other model formats to SBML, like Antimony⁶³. The Antimony format is defined in simple text format and is human readable and interpretable. Regardless, any constructed mechanistic model, in SBML format or not, must be simulated with reasonable CPU time. Although simulating models on local machines is often done, High Performance (HPC) or Cloud Computing (CC) platforms are suitable for larger tasks such as parameter

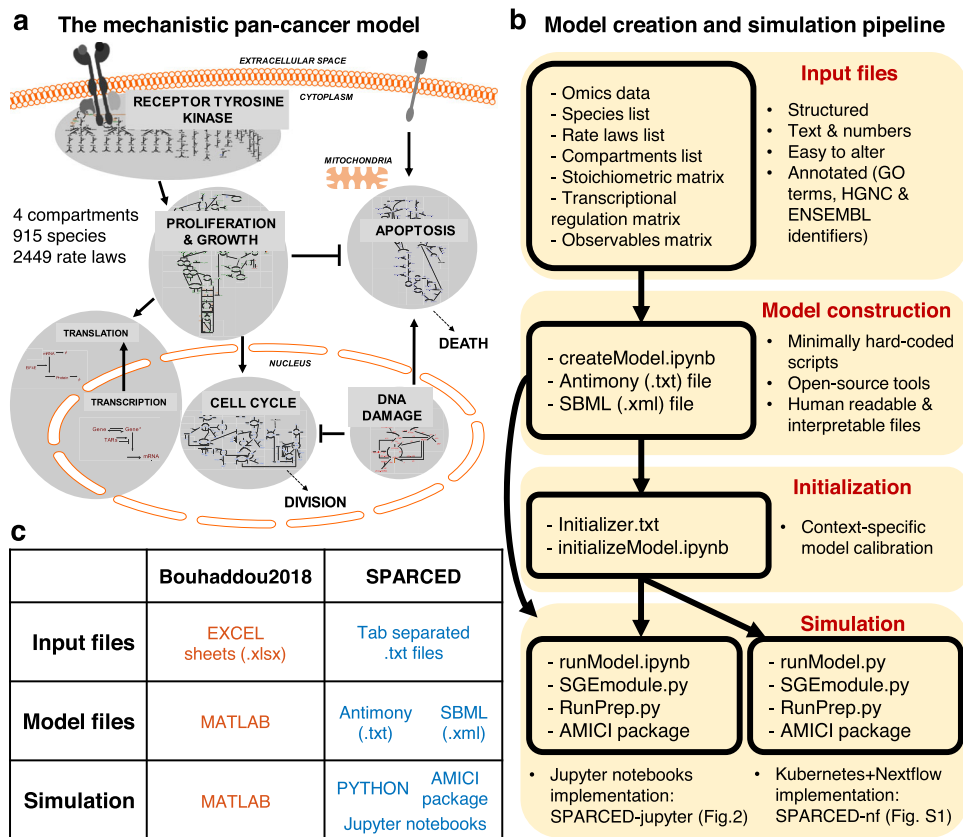


Fig. 1 SPARCED is a structured, human interpretable, and easy to modify big mechanistic model. **a** The schematic of the underlying model for SPARCED. Image adapted from⁴⁰. **b** The pan-cancer mechanistic model Bouhaddou2018 is re-written in open-source and structured file format. The steps of model construction include input file creation and conversion into an SBML file. The optional initialization step calibrates model parameters for new cellular contexts and phenotypic behaviors. The annotated SBML model file and stochastic module are simulated together at single-cell level locally or by using cloud-computing. The benefits of the new SPARCED model include easy alteration and expansion capabilities through text file editing, human-readable annotated input files, and use of Jupyter notebooks for model creation and simulation. The modeling pipeline introduced here are inline with good practices of re-usable big mechanistic models⁵⁷. **c** The Bouhaddou2018 model file types are simplified and converted into open-source platforms.

sensitivity/estimation or multiple single-cell simulations^{64–67}. Therefore, another milestone for large-scale mechanistic models is inherent HPC/CC compatibility, especially for single-cell simulations and heterogeneous data integration.

Here, we provide a framework and model construction recipe for large-scale mechanistic modeling that converts our lab’s previous large-scale pan-cancer model into a format that conveys several crucial properties noted above. First, we define a simple set of structured and annotated input text files that set model specifics: genes, species, reactions, reaction stoichiometry, cellular compartments, transcriptional regulations, input omics data, and parameter values (Fig. 1). These text files enable easy creation or alteration of the model network, with minimal coding or software usage requirements (but they are easily amenable to such things if desired). We then use Jupyter notebooks⁶⁸ to process the input files and to create a human-interpretable Antimony file, which is then converted into an SBML (community gold-standard) model file. We simulate the model using SBML compatible Python packages including AMICI, specifically designed for efficient simulation of large-scale models^{67,69}, and our own Python submodule for stochastic gene expression that enables single-cell simulations. We also develop an HPC/CC (Kubernetes) compatible version of the pipeline that enables simulating large number of single cells and/or stimulation conditions. To apply our work, we re-create and extend our previous single mammalian cell mechanistic model of proliferation and death

signaling and regulation⁴⁰, which we call SPARCED (SBML, Proliferation, Apoptosis, Receptor Tyrosine Kinases, Cell cycle, Expression, DNA damage). The pipeline and model are available on GitHub (github.com/birtwistlelab/SPARCED)⁷⁰. Using the newly created model format, we investigate if a putative mechanism could explain the experimental observation that interferon-gamma (IFN γ) inhibits epidermal growth factor (EGF)-induced cell proliferation. The model analysis suggests that IFN γ could inhibit cell proliferation through SOCS1-induction and reduced AKT and MAPK activity. This large-scale mechanistic model construction recipe and the SPARCED model outlined here is an important step towards creating and testing large-scale mechanistic models as data integration and clinical decision-making tools.

Results

SPARCED model construction and unit testing. Current large-scale mechanistic models are agglomerates of smaller models and tools, used mainly within the same research lab. Most such models also lack clear and satisfactory annotation and metadata, making them harder to understand and alter^{23,40}. The goals of this work were (i) to build tools that help large-scale mechanistic model construction and alteration, that is simple, efficient, open-source, and cloud computing compatible; (ii) to provide a scalable and re-usable big mechanistic model for a single mammalian

cell; and (iii) demonstrate the work through application to a biological question.

We first created a set of simple input files and scalable processing scripts for one of the broadest cancer signaling models in the literature⁴⁰, called the Bouhaddou2018 model here (Fig. 1a). The input files (Supplementary Data 1–7) are simple tab-separated text files (Fig. 1b, c), unlike licensed file formats with a mixture of hard-coded information in multiple interconnected scripts commonly used in modeling literature. A Jupyter notebook (Supplementary Data 8) processes the input files into an Antimony text file (Supplementary Data 9). The model creation code generates the SPARCED model file in SBML format (Supplementary Data 10) using the Antimony text file and annotations from the model input files (Supplementary Data 2 and 6). When the model construction step is complete and the SBML file is created, it is imported and simulated using a Python package called AMICI^{67,69}. For every new cell line model, a pre-calibration step called *Initialization* is employed to tune parameter values. Here, we ensure total protein levels match experimental observations and particular phenotypic criteria are met; for example, we specify that serum and growth factor starved cells on average do not traverse the cell cycle and do not die by apoptosis within 48 h. The resulting *initialized* parameter values and species concentrations are saved in a new SBML file, and the model is compiled for model testing and other simulations.

The result is what should be a replica of the Bouhaddou2018 model, which we call SPARCED. Like the Bouhaddou2018 model, the initial SPARCED model is based on non-transformed breast epithelial MCF10A cell line data. We annotated all the species in the model with HGNC gene identifiers, providing easier programmatic filtering and curation of species list, while keeping the user defined simpler names for complicated species structures. However, the extent to which the models are congruent was not yet clear, and thus we next set out to examine agreement between the two. We verified that the previous Bouhaddou2018 model simulations are reproducible and match expected experimental observations through the same unit test concept (Table 1) introduced for the original model⁴⁰. Each unit test has a dedicated Jupyter notebook on GitHub repository (github.com/birtwistle-lab/SPARCED/SPARCED_Brep). We illustrate select unit testing examples below, but all results are presented in supplementary figures (Supplementary Figs. 2–11).

SPARCED model simulation. Before presenting particular unit test applications, we wanted to provide an overview of model simulation. We built a Jupyter notebook called `runModel.ipynb` (Supplementary Data 11) to simulate the SPARCED model (Fig. 2). This notebook requires the model SBML (from `createModel.ipynb`, Fig. 2a), along with the simulation duration (th), the ligand concentrations (if desired), the name for the output files, and whether the simulation should be deterministic only or hybrid mode (flagD). The “Initialization” calibration step is employed only when the model is being trained for a new omics data or for different phenotypic criteria (Fig. 2b). The rest of the `runModel.ipynb` notebook imports necessary packages and model files and runs the simulation (Fig. 2c).

As mentioned, the SPARCED model consists of two modules: deterministic and stochastic. The SBML file forms the basis of the deterministic module whereas the stochastic module describes gene states (active/inactive) and mRNA birth/death events for the genes (Fig. 2c). When run in the hybrid simulation mode, the deterministic and stochastic modules exchange information every 30 simulated seconds (Fig. 2d, e). The current levels of select protein states can induce changes in gene activation/deactivation or mRNA transcription/decay rates. The newly updated mRNA copy numbers change

nascent protein translation rates in the deterministic module (Fig. 2d). When run deterministically, the model does not stochastically sample gene activation or mRNA transcription events, and such simulations correspond to an average cell state.

Individual cells (in vitro on a dish or in vivo) exhibit mRNA and protein expression variability, in part due to stochastic gene expression processes^{47,48}. To capture this phenomenon in silico, we ran simulations in hybrid mode. In this mode, each simulation has different initial mRNA and protein levels that are dictated by burst like expression processes, and the expression throughout the simulated time course follows suit. This leads to a natural and typically observed amount of variation in total protein levels. We hereafter refer to such settings and resulting trajectories as single-cell simulations. Virtual cell population responses are sets of multiple independent single-cell simulations, usually 100 cells. So, when the `runModel.ipynb` notebook is run multiple times in hybrid mode, different single-cell responses are simulated (Fig. 2f). For instance, the activation and phosphorylation of ERK (Fig. 2f left, red lines) and AKT (Fig. 2f right, blue lines) proteins in response to growth factor treatment will show variability across three example cells. Although the amplitude of initial response is similar for all three cells, the longer-term responses are quite different. Our previous analyses showed that such single-cell heterogeneity in the initial concentrations of these proteins could help predict cellular fate, namely cell division⁴⁰. These Jupyter notebooks provide a simple interface to interact with the SPARCED model.

SPARCED model unit testing: deterministic. We first tested agreement between deterministic Bouhaddou2018 and SPARCED model simulations. The SPARCED model simulations recapitulated the response of an average (deterministic) cell under different stimulation conditions, to within simulation error (Fig. 3a). As an example, we highlight SPARCED model simulations of the cell response (MCF10A cells) to treatment with EGF alone or EGF + insulin (Fig. 3b and Supplementary Fig. 12). Treating growth factor and serum-starved MCF10A cells with EGF and insulin induces activation of ERK, AKT, and their downstream signaling partners, which together influence cell proliferation^{40,71,72}. The Bouhaddou2018 model showed that compared to single ligand treatments, EGF + insulin stimulation increases and prolongs AKT and its downstream EIF4EBP1 phosphorylation (Fig. 3b). The simulation results from the Bouhaddou2018 model (the solid lines) and SPARCED model (circles) are indistinguishable. The SPARCED-*nf* implementation, which runs on a high-performance cloud computing infrastructure, similarly reproduces the original simulation data (Fig. 3b, triangles). These results, together with all other deterministic tests in Table 1 (Supplementary Figs. 3–8 and 11), confirm that the SPARCED model recapitulates the Bouhaddou2018 model simulations and unit tests in deterministic settings. Thus, the simple input file structure combined with automatic model generation is equivalent to the prior MATLAB instantiation in this regard.

SPARCED model unit testing: stochastic (hybrid). Next, we evaluated the SPARCED model for stochastic unit tests in single-cell simulations. Each single simulated cell has different initial protein levels and dynamics due to stochastic gene expression, and thus may respond differently to the same treatment. A simulated cell population is a collection of multiple single cell simulations, usually 100 unless otherwise noted. The SPARCED model stochastic simulations closely matched Bouhaddou2018 model results, to within simulation error (Fig. 3c and Supplementary Fig. 13a). As an example, we highlight here how single cells respond stochastically to DNA damage. Etoposide, a chemotherapy drug, induces double- and single-stranded DNA

Table 1 List of SPARCED model unit testing and comparisons to Bouhaddou2018 model.

Descriptions of unit tests	Simulation type	Figure #	Original paper Figure #
Functional test to ensure the deterministic module is updated every 30 s with mRNA numbers generated by the stochastic module.	Hybrid	Supp. Fig. 2	2B
Simulated ligand-receptor cooperativity coefficients for the receptor tyrosine kinases match experimental observations (negative cooperativity: EGF, FGF, IGF, INS; no cooperativity: HGF, NRG1, and positive cooperativity: PDGF).	Deterministic	Supp. Fig. 3a	3A + S3A
Activated EGF receptors internalize and peak ~30 min after ligand treatment.	Deterministic	Supp. Fig. 3b	S3B
EGF and insulin stimulation activates both ERK and AKT pathways. Dual stimulation with the two ligands induces prolonged AKT activation.	Deterministic	Supp. Fig. 4, 5	3B, C, D + S3C
Double and/or single stranded DNA damage activates p53 and DNA damage repair mechanisms represses its response.	Deterministic	Supp. Fig. 6a	3E
Increasing DNA damage amount in single cells leads to higher number of activated p53 peaks.	Hybrid	Supp. Fig. 6b, c	3F + S3E
Increasing simulated TRAIL dose decreases the time it takes to die for an average cell.	Deterministic	Supp. Fig. 7a, b	3G
The fraction of surviving cells decreases as stimulated TRAIL dose increases.	Hybrid	Supp. Fig. 7c	3H
Increasing ERK and AKT activity levels prolongs TRAIL induced time to death, whereas increasing PUMA and NOXA expression levels decreases the time it takes for cells to die.	Deterministic	Supp. Fig. 7d	3I
Increasing Cyclin D mRNA levels induces proper cyclin-CDK complex progression and oscillations for cell cycle entry and progression.	Deterministic	Supp. Fig. 8a	3J
Etoposide treatment induces cell cycle arrest and cell death. Cycling cells (with prior growth factor stimulation) show increased percentage of death to etoposide treatment, compared to non-cycling cells.	Hybrid	Fig. 3d, e	4A, B, C
Inhibition of AKT and ERK pathways together synergistically increase cell death, in EGF and insulin stimulated cells.	Hybrid	Supp. Fig. 9	5A
ERK and AKT inhibition-induced cell death mechanisms are predominantly BIM dependent, not BAD dependent.	Hybrid	Supp. Fig. 10a	5C
EGF and insulin cooperatively induce cell cycle entry, with insulin inducing very little cell cycle entry alone.	Hybrid	Supp. Fig. 10b	6B
Activation of both ERK and AKT pathways is required for robust cell cycle entry. Time averaged ppERK and ppAKT levels correlate with Cyclin D levels.	Deterministic	Supp. Fig. 11	6E
The number of ribosomes within the cell doubles within 24 h.	Deterministic	Supp. Fig. 8b	S2D

The SPARCED model passed each test depicted and recapitulated experimental and simulation observations reported by the Bouhaddou2018 model. Supp.: Supplementary.

damage, causes cell cycle arrest, and leads to cell death⁷³. Previous experimental data⁴⁰ showed that in the absence of EGF and insulin (to promote cell cycle exit), there is minimal etoposide-induced cell death (Supplementary Fig. 13b, c). However, in the presence of EGF and insulin (to drive cell cycle progression), etoposide-induced cell death increases over time and reaches around 60% of the cells (Fig. 3d, e). Simulating etoposide treatment of cycling cells induces robust p53 pulses, disruption of Cyclin A dynamics/cell cycle arrest (Fig. 3d), and more cell death relative to non-cycling cells (Fig. 3e). The SPARCED simulation results closely match experimental data and Bouhaddou2018 simulations. We conclude that SPARCED model captures DNA damage induced single-cell death percentage and cell cycle state-dependent effect of etoposide. The SPARCED model also passed all other stochastic/hybrid unit tests (Table 1, Supplementary Figs. 2, 6, 7, 9, and 10).

SPARCED model unit testing: context change. Different cell types have different mRNA and protein expression levels, and many mechanistic models assume that it is different expression levels that drive different phenotypes, as opposed to changes in biochemical rate constants. These constants are based on biophysical events like binding, which are based on molecular structures. Here, we tested the ability of the SPARCED model to be re-“initialized” to study different cell types by changing initial levels of total proteins and mRNAs without changing the model topology. Thus, we introduced a protocol to enable SPARCED model context change (Supplementary Fig. 14a and Supplementary Data 12 and 13). In short, *OmicsData*, *Species*, and *RateLaws* input files are updated with new cell line

information, including mRNA levels, protein/species levels, and constitutive translation rate constants. Then, the new model is created by running the “createModel” Jupyter notebook or by submitting a new SPARCED-nf job.

The re-calibration step for context change followed in Bouhaddou2018 model was called *Initialization*, where protein-specific translation rates and key parameters important for cell decision making are estimated to ensure agreement with new omics datasets and expected phenotypic behavior with respect to proliferation and apoptosis. Here we also provide a new, python-based version of the *Initialization* procedure for SPARCED models (see Computational Methods), where the outputs are species concentrations and rate parameter values updated in a new SBML file. Here, to test the drug combination response differences in different cell lines, we changed SPARCED model context (i.e., parameter values and species concentrations) by initializing the model to the U87 glioma cell line. Following the protocol outlined in Supplementary Fig. 14a, we replaced MCF10A cell line values in the input files with values from U87 cell line data.

U87 cells are PTEN-deficient and more sensitive to AKT inhibition compared to MCF10A cells⁴⁰. Both cell lines show minimal sensitivity to MEK inhibition alone and AKT & MEK inhibitors are both needed to kill MCF10A cells. In contrast, AKT inhibition alone is sufficient to kill U87 cells. To simulate the U87 cell response to AKT and MEK inhibitors, we first updated the *OmicsData* input file (Supplementary Data 1) using U87 mRNAseq data from Bouhaddou2018 model (Supplementary Data 14). Here, we did not use U87 cell line proteomic data and estimated the initial total protein levels using the new mRNA levels and gene-level mRNA/protein ratios from MCF10A data

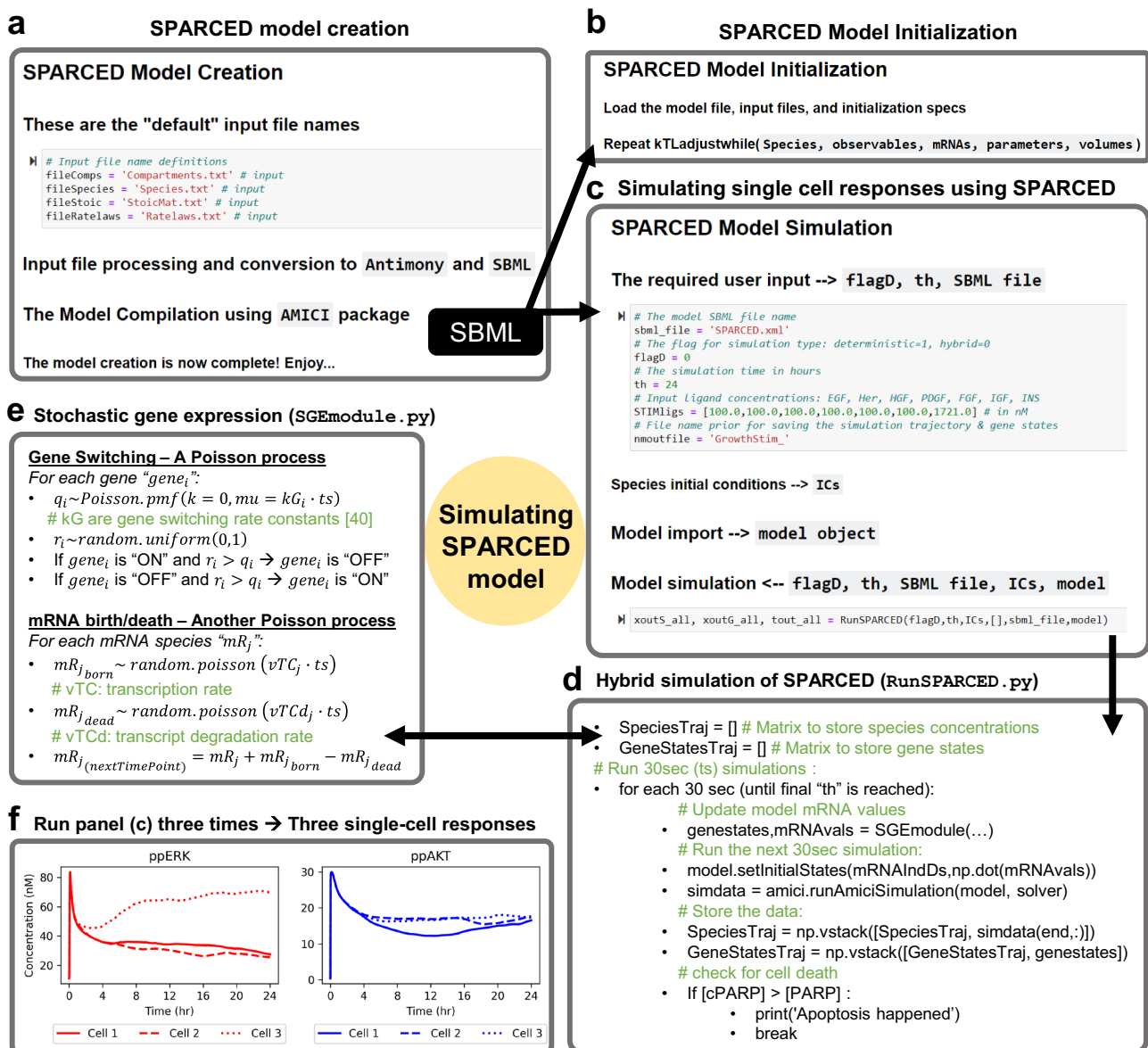


Fig. 2 SPARCED-jupyter enables single-cell response simulations using Jupyter notebooks. **a** The model creation notebook processes the input files and converts them into the model SBML file, which is compiled for simulations using the AMICI python package. **b** When a model is generated for a new cellular context (using new omics input data), next is an initialization step to adjust protein translation rate constants and cell death/DNA damage related parameters. **c** The model simulation starts with specifying and importing the SPARCED model SBML. The user defines the model file name and the sets four additional parameters: (i) The flag (1 or 0) to specify if the model should run in deterministic or in hybrid mode (see **d** and **e**), respectively. (ii) The time duration in hours for which the model should run. (iii) The vector of ligand concentrations (in nM) to stimulate the cells. (iv) The output file name. Next, the species initial conditions are, by default, read-in from the "Species" input file. Then, the model file is imported and simulated according to the specified input. The model outputs three matrices: species concentrations over time, the activation states of genes over time, and the time points of simulation in seconds. **d** The model is simulated iteratively for each 30 s, where the current species concentrations are inputs for the gene expression module, which then outputs new mRNA levels to update the SBML model states. The model is then run for another 30 s, until the total simulation time reaches the user input (th) or until the cell dies. The cell is considered dead when `[cleaved-PARP] > [PARP]`. **e** In the gene expression module, in hybrid mode, the model randomly decides which genes become active or inactive, and which mRNAs are transcribed or degraded. This SGEmodule.py script is called every 30 s with updated species concentrations, simulated using the models SBML with AMICI package. **f** When the model is hybrid-simulated three times, the different cell responses are observed. Shown are serum-starved average cells stimulated with full growth media for 24 h. Plotted are free ppERK and ppAKT species concentrations (nM).

(Supplementary Data 15). We set the PTEN translation rate to zero and set values of rate parameters dictated by *Initialization* in the *RateLaws* input file (Supplementary Fig. 14b and Supplementary Data 12). Additionally, we provide an improved Python based initializer *initializeModel.ipynb* notebook (Supplementary Data 16 and 17), which re-creates (Supplementary

Fig. 15) the un-stimulated steady-state initial conditions for species and adjusts translation rate constants using cell-line specific initialization input file (Supplementary Data 18). We also updated the species initial conditions in the *Species* input file using steady-state values for U87 cells from the Bouhaddou2018 model (Supplementary Data 19). We created a new model SBML

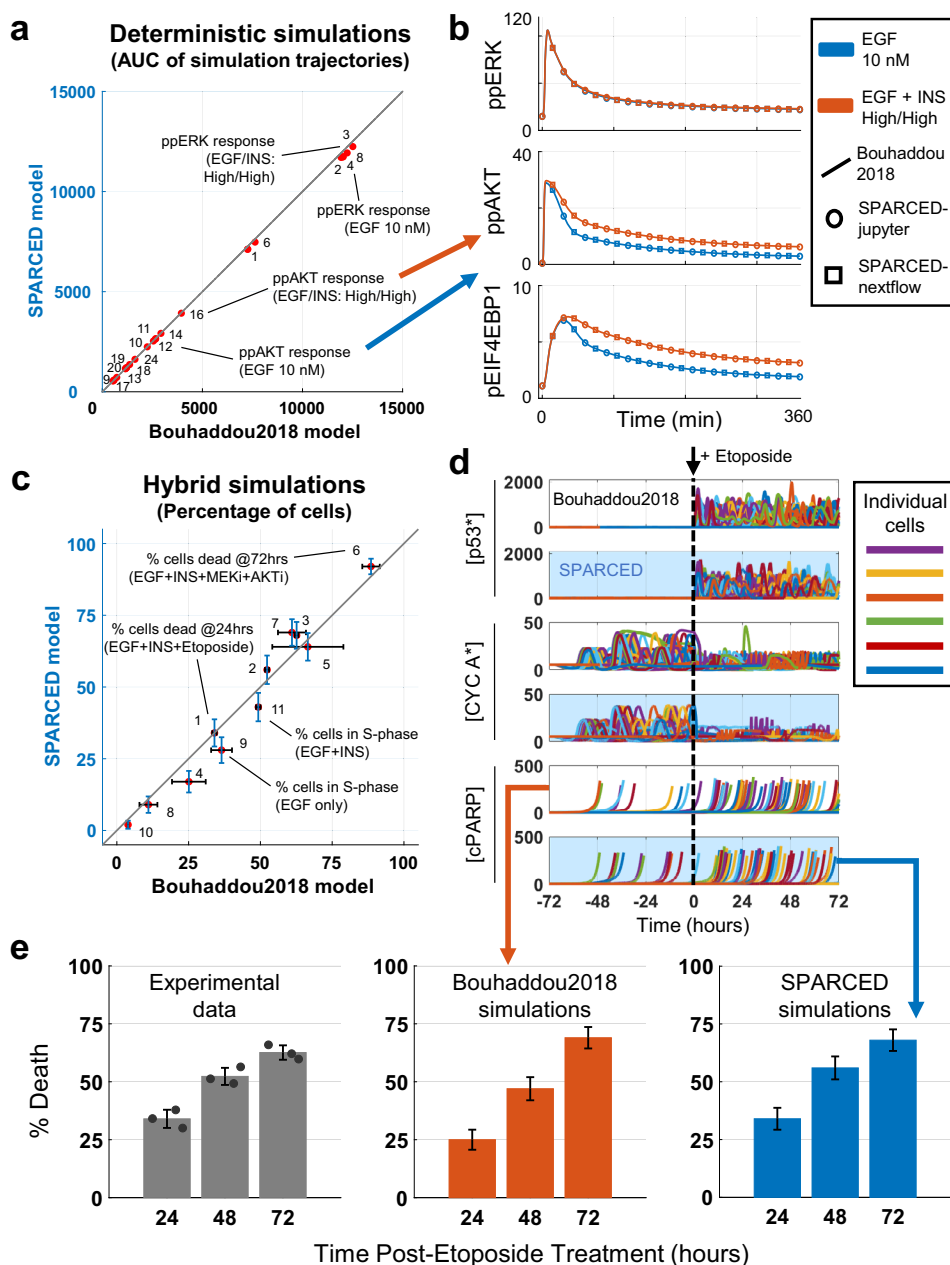


Fig. 3 SPARCED model recapitulates experimental observations and deterministic/hybrid (deterministic + stochastic) simulation results of the Bouhaddou2018 model. **a** Summary of comparisons of SPARCED model deterministic simulations to Bouhaddou2018 model simulations. The area under the curve (AUC) values of each simulation (see Supplementary Fig. 12) are calculated and plotted for the two model results. **b** Simulation results from Bouhaddou2018 model (line) and SPARCED-nf model (triangle) run on Kubernetes cluster workflow are the same as SPARCED model (circle) results. Comparisons of selected panels from **(a)** are shown only. **c** Experimental and stochastic simulation results from Bouhaddou2018 model are reproduced by SPARCED model simulations. Each dot is a different condition, explained in Supplementary Fig. 13a. Error bars show experimental or simulation standard error of the mean. Simulations are of at least 100 cells, and three independent experimental observations where applicable. **d** Stochastic simulation of 100 cells recapture protein level trajectories (active p53, Cyclin A, and cPARP) from older model qualitatively. Panels with blue background are SPARCED simulations and white background panels are from Bouhaddou2018 model. 100 stochastic cells are stimulated with EGF + Insulin for 72 h before Etoposide treatment for another 72 h. Etoposide is stimulated also with EGF + Insulin. Results for Etoposide treatment without prior growth factor stimulations are shown in Supplementary Fig. 13b. **e** Quantification of results in **(d)** shows that SPARCED model simulations coincide with earlier observations in percentage of death induced by etoposide treatment. See Supplementary Fig. 13c for the effect of no growth factor stimulation before Etoposide treatment. Gray bars represent mean \pm s.e.m. of three independent biological replicates (dots). Blue and orange bars represent the percentage of dead cells at specified time \pm s.e.m. Source data are provided in Source Data.

file (SPARCED_U87) using the updated input files. SPARCED_U87 model simulations of response to MEK and AKT inhibitors reproduced the Bouhaddou2018 model results and experimental observations (Supplementary Fig. 14c). We conclude that changing model context by changing input files is

possible and contributes towards the goal of easy model alteration to study of different cell types.

When the cellular context (omics input data) for the SPARCED model is changed, all appropriate Unit Tests should be passed. We expect that addition and alteration of the list

provided (Table 1) will accommodate increasingly different prior knowledge about the new context. Examples of such information include cell line mutations, growth condition differences, or tumor cell behavior.

Illustrating easy model expansion by application to the IFN γ pathway: SPARCED-I. The SPARCED model reproduces original model results and experimental observations, but how can we use the new simple model expansion capabilities? Here, we focused on experimental observations that interferon-gamma (IFN γ) inhibits MCF10A cell proliferation. Specifically, as part of a much larger LINCS consortium effort to deeply profile the MCF10A cell line dynamic response to perturbations (synapse.org/LINCS_MCF10A), we observed that IFN γ inhibits EGF-induced cell proliferation (Fig. 4a). We wanted to use the SPARCED model expansion functionality to evaluate the suitability of candidate mechanisms by which IFN γ might inhibit proliferation.

We thus merged a model for interferon-gamma receptor (IFNGR) signaling into SPARCED, creating the model variant SPARCED-I. The newly added Yamada2003 model⁷⁴ captures how IFN γ binds to pre-JAK-bound receptors, inducing dimerization. The ligand-bound receptor homodimers are activated by JAK, which also phosphorylates STAT1 when bound to the active receptor complex. Activated STAT1 then dimerizes and translocates to the nucleus, inducing transcription of SOCS1. SOCS1 mRNA is exported to the cytoplasm and translated into SOCS1 protein. SOCS1 protein binds to and inhibits activated receptor homodimers. In this model, there are three phosphatases (SHP2, PPN, and PPX) acting on multiple species.

Following the model alteration protocol in Supplementary Fig. 14a and Supplementary Data 13, we added 34 new species, 8 corresponding genes, and ~70 reactions (Fig. 4b) based on the Yamada2003 model⁷⁴. New genes (*IFNG*, *IFNGR*, *JAK2*, *STAT1*, *SHP2*, *SOCS1*, *PPN*, *PPX*) corresponding to the proteins in the model are added as new rows and mRNA levels are inserted into the OmicsData input file. Gene copy numbers are taken as two⁴⁰. The genes are also added as new rows to the GeneReg file (Supplementary Data 20). In the Yamada2003 model, activated nuclear STAT1 dimers (STAT1*Dn) induce SOCS1 mRNA transcription, and this is captured by adding a new column in the GeneReg input file, with the only non-zero element at the STAT1*Dn and SOCS1 gene intersection. Next, each protein, protein complex, and mRNA species are inserted into the Species and StoichiometricMatrix input files as new rows. Each new reaction is inserted into the Ratelaws input files as rows, and into the StoichiometricMatrix input file as new columns. This expansion brought the total number of species of SPARCED-I model to 954, and the number of reactions to 2540 (Fig. 4c, Supplementary Data 21).

SPARCED-I model unit testing. The SPARCED-I model, with parameter values from the Yamada2003 model, should reproduce the original results exactly, which we verified (Fig. 4d, red lines & diamonds, respectively). We then modified the mRNA, protein, and compartment volume values to that of MCF10A cell context (data from⁴⁰). However, the MCF10A data (Supplementary Data 22) had missing values for IFNGR and (arbitrary) phosphatase species PPN and PPX. So, we initialized the concentration of IFNGR as half of JAK2 concentration (the receptor is typically rate limiting⁷⁴). The concentrations of PPN and PPX phosphatases were equal to half of SHP2 concentration in Yamada2003 model, so we updated their values to half of SHP2 concentration in MCF10A cells. In addition to the reactions from the Yamada2003 model, we added new translation (for the new

eight genes) and degradation (for all new species) reactions into the model. The rate constants of these extra reactions are initially assumed to be equal to the average of corresponding reactions of SPARCED model genes. Starting from these parameter values, the SPARCED-I model showed unrealistic (ultrafast) receptor activation and STAT1 phosphorylation/nuclear transport rates. Therefore, we varied six parameters that have high impact on STAT1 activation dynamics (see Methods) to approximate the timing (within the first hour) of STAT1*Dn pulses reported by Yamada (Fig. 4d) and others^{75,76}. Changing rate constants in such a manner accounts for the entangled effects of unmodeled cellular context and mechanisms. Tuning these parameters produced expected pulsing times and response behavior of STAT1*Dn, SOCS1, and SOCS1 mRNA levels (Fig. 4d black lines). The final values are updated in the SPARCED-I model file (Supplementary Data 21).

A key feature of SPARCED-I is its ability to simulate a virtual cell population response and the above-observed reduction in proliferation induced by IFN γ is inherently a population-based property. As a unit test, we simulated 100 single cell trajectories (Fig. 4e) of SPARCED-I model and concluded that SPARCED-I model recapitulates observations from earlier models (i.e., qualitative STAT1 and SOCS1 dynamics) and passes all unit tests (i.e., hybrid mode works).

SPARCED-I model variant analysis: hypotheses testing. Next, we wanted to use the expanded model to help us interpret the experimental observations. *How does IFN γ inhibit EGF-induced cell proliferation?* The SPARCED model captures regulation of cell proliferation via the ERK and AKT pathways. Growth-inducing ligands, like EGF, bind to and activate receptor tyrosine kinases (RTKs), which in turn leads to upregulation of AKT and ERK phosphorylation. The two pathways together induce upregulation of cyclin D through cJUN, cFOS, and cMYC activities^{40,71,77}.

In the literature, there are different mechanisms by which IFN γ was suggested to play a role in cell proliferation^{78–81}. The SPARCED-I model enabled us to evaluate one of these hypotheses for consistency with experimental observations (Fig. 4a), where the simulation steps are matched to the experimental setup (Fig. 5a). The mechanism involves the negative regulator of IFN γ signaling, SOCS1. SOCS1 protein has different binding domains, including SH2 domains^{82,83}. SH2 domains bind to phosphorylated tyrosine residues on other proteins^{84,85}. It is proposed that SOCS1 not only binds to activated IFN γ receptors, but also to many other activated receptor complexes with free phosphorylated tyrosine residues (Fig. 5b)^{83,86,87}. Thus, IFN γ -induced SOCS1 protein can bind to growth factor-activated receptor complexes (or the so-called signaling competent dimers – pSCD) and prevent further downstream signaling by sequestration. This mechanism was modeled by adding SOCS1 binding to activated receptor complexes (pSCDs) reactions in the Ratelaws input file. The SPARCED-I SOCS1 model contained 1302 species (348 new) and 3584 reactions (1044 new) (Supplementary Data 23). GRB2 proteins also contain SH2 domains that bind to tyrosine phosphorylated receptors (pSCDs) and the rate constants of SOCS1 interaction with all these complexes are taken as the average of such parameters of GRB2 complexes.

Before evaluating the SOCS1 crosstalk model responses, the initial conditions must be set. The SPARCED-I model initial conditions are based on serum and growth factor starved MCF10A cells. However, the experiments with IFN γ (Fig. 4a) were done in media with horse serum. Horse serum upregulates ppAKT levels by four-fold (Supplementary Figs. 16 and 17), possibly through IGF/IGF1R pathway^{88–90}. Including 0.02 nM IGF meets this basal

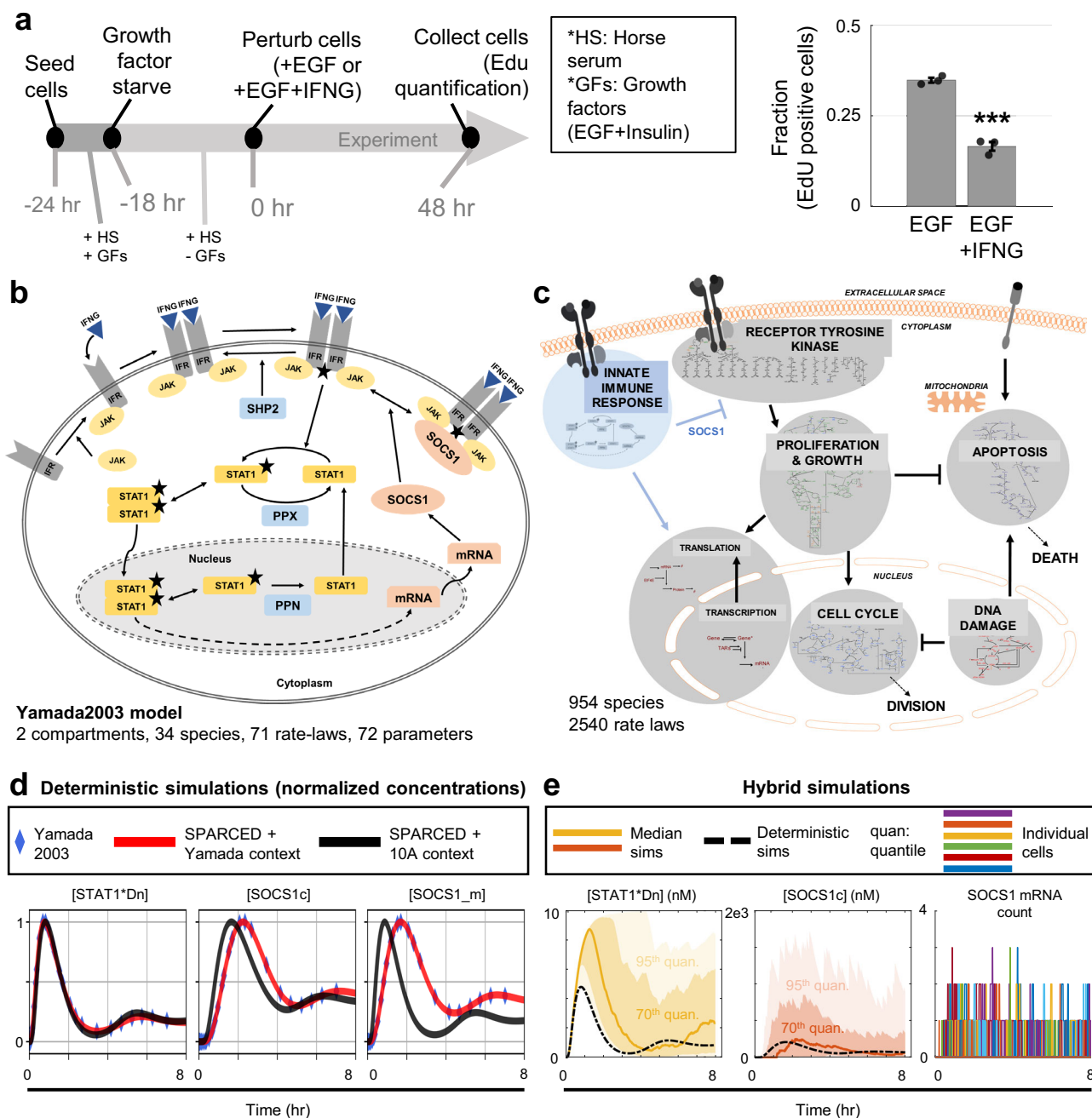


Fig. 4 SPARCED model is enlarged to include interferon-gamma (IFN γ) signaling pathway. **a** Experiments showed that IFN γ treatment significantly decreases cell proliferation induced by EGF alone (p -value = $1.86E-04$). Fraction of EdU positive cells at 48 h after EGF alone or EGF + IFN γ treatment are calculated. Bars represent mean \pm s.e.m. of three independent biological replicates. Significance tested using two-sided two-sample t -test, where *** p -value < 0.001. **b** Yamada2003 model schematic of IFN γ pathway added into SPARCED model. **c** Overview of the added IFN γ -IFN γ R pathway in relation to the Bouhaddou2018 model pathways. The “SOCS1” link is the candidate mechanism tested in the next section. **d** Simulations of the Yamada2003 model, SPARCED-I model, and SPARCED-I model with MCF10A context show qualitative and quantitative agreements. The Yamada2003 model results (blue diamonds) are obtained by running the model file in COPASI⁶¹. **e** 100 stochastic cell (hybrid) simulations (area plots) and the deterministic (dashed-black lines) simulation of 10 nM IFN γ stimulation are shown. Colored dark lines represent median cell trajectories, dark and light-colored regions represent 70th and 95th quantiles, respectively. The right-most plot shows mRNA count of SOCS1 in each cell, colored differently. Source data are provided in Source Data.

activity increase constraint, and therefore is included in simulations prior to and during simulated EGF and IFN γ treatments (see Methods and Supplementary Figs. 16–19).

After creating the SPARCED-I SOCS1variant and defining the simulation conditions, we stochastically simulated the model with IGF treatment for 24 h for 100 different single cells to provide a baseline. Then, either EGF or EGF + IFN γ are added for an

additional 48 h for each cell (Fig. 5a). As a simulation metric for cells in S-phase at 48 h, we counted cells for which the sum of concentrations of Cyclin E, A, and B is greater than 20 nM⁴⁰. We counted such cells for different values of the dissociation constant (K_d) (Fig. 5d). We varied the unbinding rate constant (\log_{10} within the range -4 to 3) and ran different 100 cell simulations for each condition. Looking at the ratio of number of cells in S-phase in IFN γ

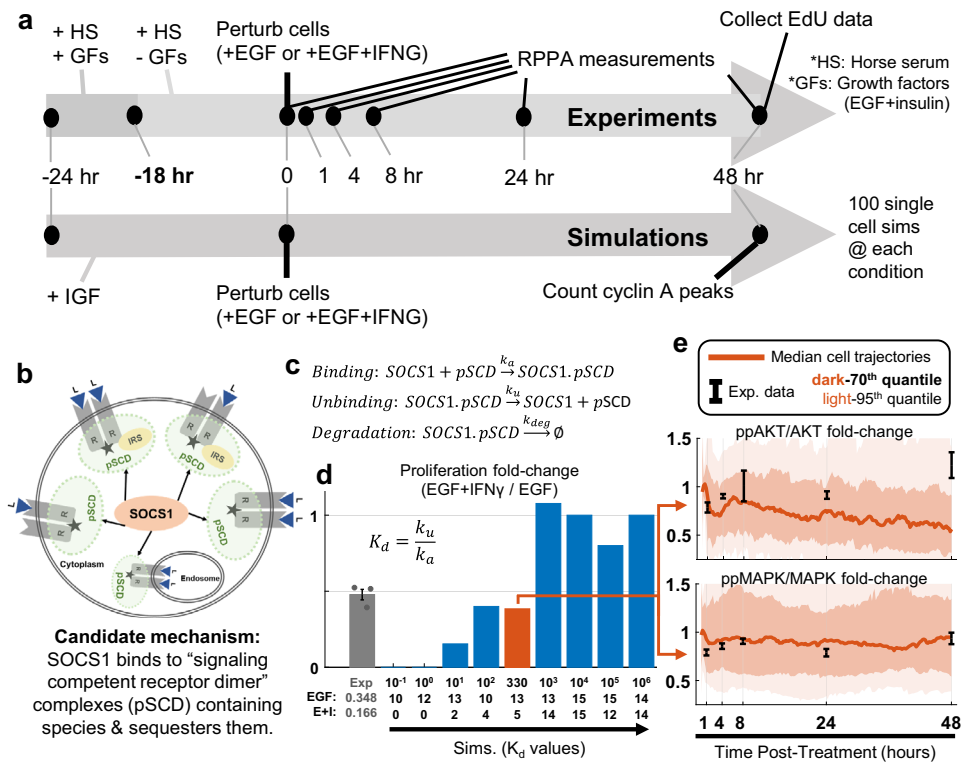


Fig. 5 SPARCED-I model suggests that a SOCS1-based sequestration mechanism can explain IFN γ -induced cell proliferation decreases. **a** Experimental setup and simulation workflow for SPARCED-I model variant analysis. See Methods. **b** Candidate mechanism schematic: SOCS1 sequesters activated ligand-receptor complexes. **c** The reactions show one set of examples added to capture the inhibitory mechanism. pSCD: activated ligand-receptor complexes. K_d : Disassociation constant. **d** Simulation of 100 stochastic cells for SPARCED-I-SOCS1 model at different K_d values (varied uniformly between -1 and 6 in log $_{10}$ scale) for SOCS1 binding (blue bars) showed that SOCS1 sequestration of activated receptor complexes can explain the IFN γ effect on cell proliferation inhibition, compared to experiments (gray bar). The number of cells in S-phase at 48 h of simulation time are shown on the x-axis (first row: K_d values, second row: cells with EGF, third row: cells with EGF + IFN γ). The y-axis is the ratio of number of cells in S-phase with IFN γ to without IFN γ . The experimental data (Exp.) are from three independent replicates (dots) and represent mean \pm s.e.m. The mean of each experimental condition is given in x-axis. The orange bar is K_d set at the estimated initial value (from GRB2-Receptor binding/unbinding reactions). **e** The K_d value used for these trajectories corresponds to the orange bar in **(d)**. Normalized ppAKT and ppMAPK levels show a significant decrease after IFN γ treatment, when EGF + IFN γ SPARCED-I-SOCS1 model simulations are compared to EGF alone case. RPPA data (synapse.org/LINCS_MCF10A) are shown in black, from three independent replicates (error bars are s.e.m.). Colored dark lines represent median cell trajectories from simulations, dark and light-colored regions represent 70th and 95th quantiles, respectively. Source data are provided in Source Data.

treated cells to EGF-only cells, we saw that the model (Fig. 5d, blue and orange bars) can capture the experimental data of \sim 50% decrease (Fig. 5d, gray bar), especially for the lower end of the K_d values. This range of K_d values is consistent with previous experimental observations for such interactions^{91,92}. We conclude that the SOCS1 mechanism model is not inconsistent with decreased proliferation in IFN γ treated simulated cells.

We investigated responses of simulated cells to explore why fewer simulated cells enter the cell cycle (Fig. 5d). We saw a simulated decrease in both AKT and MAPK activation in IFN γ treated cells, supported by experimental RPPA data (synapse.org/LINCS_MCF10A), which was not fitted (Fig. 5e). For the estimated K_d value simulations shown (orange bar in Fig. 5d) the greatest decrease occurs at around one hour after ligand treatment and stays decreased for 48 h. Both simulations and experiments show decreased ppAKT and ppMAPK levels at 1–8 h post-ligand treatment (Fig. 5e, compare experimental black bars to median simulation orange lines). The variability in the 8 h measurement for ppAKT was large, and the 24 h measurement showed still reduced ppAKT. By 48 h, ppAKT is upregulated in the experimental data, which is in part captured by the simulation variance (the 70th quantile), but the model does not fully explain. These results suggest that downregulation of both AKT and

MAPK responses leads to decreased cell proliferation, as previously investigated by Bouhaddou et al.⁴⁰ and others^{72,93–95}.

Is the somewhat subtle decrease in MAPK and AKT activity for \sim 1 day enough to prevent cell cycle entry in MCF10A cells? Bouhaddou et al. showed that a slight change in time integrated MAPK and/or AKT activity dynamics can tilt the cell cycle progression decisions significantly (see Supplementary Fig. 11 and ref. ⁴⁰ for further details) and here we saw both ppMAPK and ppAKT levels show a statistically significant decrease after IFN γ treatment, which may explain fewer proliferative cells. Taken together, these results suggest that: (1) the SPARCED model formalism can be used for simple implementation of large-scale mechanistic model-based hypotheses testing, (2) crosstalk between IFN γ and EGF pathways may occur through SOCS1 sequestration of activated receptor domains, and (3) SOCS1-induced inhibition of AKT and MAPK activation may contribute to proliferation suppression.

Discussion

Here, we have re-created one of the largest mechanistic models in the literature, using our new python-based creation and simulation pipeline. Our modeling pipeline and model creation recipe are based

on structured and easy to modify input text files and uses Jupyter notebooks or scripts (for scaled cloud-computing) to create and simulate model files. It also enables easier model alteration (species/rate law or parameter value changes), omics data integration, and model variant vs. hypotheses testing. Our exemplar model, called SPARCED, is available online on GitHub (github.com/birtwistlelab/SPARCED)⁷⁰. While the pipeline we introduce is a recipe for large-scale model construction, the SPARCED model itself can serve as a basis for creating context-specific (personalized) model variants, studying virtual cell population responses, and as a building block towards whole-cell-scale models.

First, we showcased the use of SPARCED model by changing the cellular context of the model from MCF10A breast epithelial cells to U87 glioblastoma cells, by only replacing parameter values in a few input files. Although we used previously calculated values from the Bouhaddou2018 model to show reproducibility of the subsequent analyses, it is notable that SPARCED pipeline correctly creates and simulates a large-scale mechanistic model file only from an altered set of text-based input files. Additionally, we provided a new version of the Initialization script (Supplementary Data 16) that utilizes another cell-line specific input file (Supplementary Data 18) to calibrate the model initial conditions. The initialization allows distribution of total protein and mRNA level omics data across all model species and estimates data-driven, cell-line specific translation rate constants. As a customizable set of steps, the initialization sustains user defined phenotypic responses, like cells not going into apoptosis or cell cycle without growth factor stimulation. Importantly, the procedure accommodates mRNA input alone (without proteomics data) and calculates total protein levels using gene-level mRNA-to-protein ratios from the default MCF10A values. The outputs of the initialization procedure are species concentrations and parameter values deposited into a new SBML file, which is exported with a new name and re-compiled using AMICI.

Secondly, we investigated a candidate crosstalk mechanism to explain how IFN γ can inhibit EGF-induced cell proliferation. We created the model variant by only changing the input files and hypothesized that IFN γ -induced SOCS1 sequestration of activated receptors is a putative crosstalk mechanism. Besides the one tested here, there are others in the literature, like the positive feedback of STAT1 inducing STAT1 and IRF1 transcription⁷⁹, or the inhibition of Bcl-2 by STAT1⁷⁸. However, here, we only focused on demonstrating the capabilities of SPARCED pipeline to easily create and test model variants to help explain one of our experimental observations, where testing all possible mechanisms was out of our scope.

Many existing big models are constructed in complicated and hard-coded ways and are not available in standard modeling formats, like SBML. For instance, the Bouhaddou2018 model we used as our starting point was custom coded in MATLAB with tens of different script files with thousands of lines. Although the model performance was optimized for its topology, alteration and expansion of the model was extremely difficult. However, models, especially the large-scale and clinically relevant mechanistic models, must become easy to formulate, understand, and disseminate for reproducibility, re-useability, and applicability in clinical decision making. Here, the model construction pipeline and the SPARCED model contributes to this need by being built upon structured and annotated input files, by using open-source packages, and by being available publicly on GitHub.

One key advantage of the SPARCED model format is its potential compatibility with RBM. The reactions and species created by RBM software can be incorporated (manually or programmatically) into the SPARCED model input files. Although existing RBM software can export models in SBML

format and enable multiple features, the SPARCED models enable single rate parameter changes and inclusion/exclusion of individual rate laws at the input file level. Then, the SPARCED-nf pipeline can be used to study large-scale variant analysis or to do parameter scanning. One main goal of the AMICI package⁹⁶ is enabling large-scale parameter estimation, and our choice to use this package was to enable such future endeavors when needed. Combining this idea to test consistency across multiple datasets, users can search for best-fit models or pinpoint discrepant datasets given the model topology²⁸.

Another advantage of the SPARCED model will be its ability to integrate multiple omics datasets into a large-scale mechanistic model, creating a “personalized” model variant reflective of another cellular or patient context^{26,97}. With the *Initialization* procedure linked to our pipeline, users can incorporate mRNA, copy number variation (CNV), and even proteomics data from established databases like CCLE⁹⁸, TCGA^{99–101}, HPA^{102–104}, and Cellosaurus¹⁰⁵ into the input files programmatically (Supplementary Fig. 14) and test changing the initial conditions of the model using the same network structure (Fig. 4).

The SPARCED model encodes intrinsic stochasticity of total protein levels and mRNA numbers in its hybrid simulation mode, making it unique (together with the Bouhaddou2018 model) to offer stochastic as well as deterministic simulation settings within a single model of this biological and time scale. There are other tools such as COPASI that offer hybrid (deterministic + stochastic) simulation settings^{61,106}. Our hybrid simulation approach treats the gene expression module as stochastic (events modelled as Poisson processes) and the protein signaling module as deterministic. COPASI uses next-reaction-method¹⁰⁷ for the part it determines as stochastic based on molecule numbers of the interacting species. However, as the developers stated, such implementations tend to be inefficient and take prolonged simulation wall-times. Indeed, COPASI (v4.25, build 207, on Windows 10 Pro) fails when we try to simulate our model. A next step for the SPARCED model is to combine the gene expression (scripted) and protein signaling (SBML file) modules into a single SBML model file. Such a change would enable broader cross-platform testing and usage of the SPARCED model. As stated above, even the current SPARCED model SBML is too big for most tools available to accurately simulate for the relevant time scales (24–72 h). New numerical/algorithmic methods are required to simulate large-scale hybrid models¹⁰⁸. The single-cell capability of SPARCED allows one to capture some important aspects of cell line and tumor heterogeneity compared to an average cell condition (the way many mechanistic models are built). Users can leverage this feature to simulate virtual populations and study a cell population response to drug treatment, which is often a single-cell readout as are most cellular phenotypes. However, such simulation settings require larger computational resources and thus model compatibility for high performance computing environments. The SPARCED model is built to be compatible with cloud computing, where it can be used to simulate thousands of single cells with single job execution (see Methods).

There are, of course, some shortcomings and remaining challenges. Although we extensively showed that the SPARCED model creation/alteration is much easier compared to previous version, it still is a (careful) stitch-together of other models. There are certainly other models that can be substituted and tested. The tab-separated input files separate model details from the simulation itself and offers multiple advantages mentioned throughout this work. However, it can be seen cumbersome for some modelers. These input files include species and compartment annotations, but there are other recent efforts to standardize such metadata/annotation sharing, which we would conform to when fully developed¹⁰⁹. The hybrid mode of SPARCED includes

Python scripts for stochastic simulation of gene expression module. By defining this module in another SBML file, and using packages like SBML comp¹¹⁰, one can start exploring other methods and tools for performance testing. However, a full comparison to other hybrid and stochastic methods requires computational tools that are yet to be developed that work with models of this size. Additionally, we do not provide scripts to merge new models into SPARCED, and the field of model merging is an active research area^{111–113}. For instance, researchers can write code to insert new reactions and species from other model files into existing input files, which would then also require to update `OmicData.txt` input file with new gene information. Yet, in our experience, model merging often necessitates human interaction to define the mechanisms by which species interact with one another and what rate laws should define those interactions. Finally, if one desires to alter the model at the Antimony file stage, there are currently no automated ways we provide to map the changes onto input files. This would be possible if desired but was not studied here.

SPARCED is a large-scale pan-cancer pathway model that incorporates six major sub-modules, making it one of the state-of-the-art computational models for mammalian signaling. However, it does not yet include one of the hallmarks of cancer, the cellular metabolism mechanisms¹¹⁴. Additionally, the current version of the SPARCED simulation code does not explicitly track cell division events, although the cell cycle itself is modeled. However, this is ongoing work and will enable us to better capture and compare to experimental observations and data, such as traditional drug dose viability response experiments that are fundamentally related to tracking single cells and their division/death events.

One of the challenges is to explore simulations of *spatially aware* single cells. Currently, the SPARCED model captures intrinsic heterogeneity of cells (by having stochastic gene switching and mRNA birth/death events) but these cells cannot “talk” to each other. In the future, by having scenarios where spatial orientation of cells are recorded and the secreted or stimulated molecules are shared between them, we can better capture tissue microenvironment and heterogeneous pharmacokinetics^{115,116}. Related to this first task, the second challenge is to create and simulate scenarios with multiple cell types (i.e., models trained on data from different subtypes of cells) or defining events to capture differentiation of cells. For example, one may be able to use single cell RNAseq data to train SPARCED-like models to enable tissue-level simulations with the critical cell types in the proper geometric locations. This overall vision would enable spatially aware, single-cell level, large-scale mechanistic models trained on individual patient data for *in silico* drug screening. The pipeline presented here is an important step towards this goal.

Another challenge to achieve using large-scale models is a whole-cell level mechanistic model for mammalian cells^{56,117}. With our approach, the SPARCED model can be enlarged using other small-scale models for pathways and mechanisms not currently included in the model. By utilizing the unit testing approach, one can then verify the model performance and get larger, more comprehensive models. The open-source framework presented here increasingly facilitates community contribution for model context-change and parameter tuning based on new experimental conditions.

Our introduced method of large-scale mechanistic model construction, and the SPARCED model as a basis, will enable researchers to more easily create and manipulate new model versions, test different mechanisms of action to interpret experimental observations, and change the model's cellular context. The models created by the SPARCED pipeline can incorporate multiple (omics) datasets, providing non-“black-box” data integration and modeling;

however the extent to which a fixed “initialization” pipeline can be successfully applied to a variety of cell lines remains to be tested. These SPARCED models additionally provide single-cell level simulations, compatibility with cloud computing, and human-interpretable & annotated model files in SBML format (as do other modeling tools, albeit not at this scale). The SPARCED model now can more easily be re-used as one of the largest mammalian-cell mechanistic model in the literature and serves a primer role in creation of context-specific, hypotheses testing, and expandable models. In conclusion, the SPARCED model format contributes towards important foundations of reusable big models, paving the way towards personalized mechanistic models for data integration.

Methods

Experimental methods

Cell culture. MCF10A cells (ATCC #CRL-10317, acquired from LINCS Consortium/Gordon Mills and STR verified internally in March 2019) are cultured in DMEM/F12 (Gibco #11330032) medium supplemented with 5% (by volume) horse serum (Gibco #16050122), 20 ng/mL EGF (PeproTech #AF-100-15), 0.5 mg/mL hydrocortisone (Sigma #H-0888), 10 µg/mL insulin (Sigma #I-1882), 100 ng/mL cholera toxin (Sigma #C-8052), and 2 mM L-Glutamine (Corning #25-005-CI). Cells were cultured at 37 °C in 5% CO₂ in a humidified incubator and passaged every 2–3 days with 0.25% trypsin (Corning #25-053-CI) to maintain subconfluency. Serum starvation medium is DMEM/F12 medium supplemented with 2 mM L-Glutamine. Experimental starvation medium is DMEM/F12 medium supplemented with 5% (by volume) horse serum (Gibco #16050122), 0.5 mg/mL hydrocortisone (Sigma #H-0888), 100 ng/mL cholera toxin (Sigma #C-8052), and 2 mM L-Glutamine (Corning #25-005-CI).

Tissue culture treated, non-collagen coated plates with full serum starvation. The cells were seeded in full growth media at 150,000 cells/well in tissue culture treated six well plates (Corning # 08-772-1B). The next day, cells are washed once with 1X PBS (one phosphate buffered saline tablet (Sigma #P4417-100TAB) in 200 mL milli-Q water, autoclaved) and the media was exchanged to serum starvation media (DMEM/F12 medium, 2 mM L-Glutamine) for 16–24 h. Then, the cells were treated with vehicle control, EGF (10 ng/mL, PeproTech #AF-100-15), and HGF (40 ng/mL, R&D Systems #294-HGN-005) for 0, 5, and 60 min in a humidified, 5% CO₂, 37 °C incubator.

Collagen-coated plates and growth-factor starvation only. Collagen-coating mixture was prepared as follows: 7.5 mL diluent buffer (20% v/v glycerol, 10 mM EDTA, PBS), 1.5 mL Tris-HCl, 0.6 mL COL1 (Cultrex #3442-050-01), and 5.4 mL PBS. 950 µL coating mix was added into each well of a six-well plate. After making sure that the entire well surface was covered, the plates were incubated one hour at room temperature. After incubation, any remaining liquid is aspirated and discarded. The wells were washed twice with sterile PBS and left lid-open under a sterile laminar flow hood until wells were fully dry (~one hour). Upon replacement of the plate lid, the plates were stored in a benchtop desiccator at room temperature for a minimum of 3 days before use. Then, MCF10A cells were seeded in full growth media at 150,000 cells/well. After being allowed to attach for 7–8 h, the wells were washed once with PBS and the media was changed to full growth media without EGF and insulin for 18 h. Then, the cells were treated with vehicle control, EGF (10 ng/mL), and HGF (40 ng/mL) as above for 0, 5, and 60 min.

Cell lysis. After growth factor treatment, the plates were removed from the incubator and put on ice. The media in the wells were aspirated and the wells were washed with PBS. 110 µL of freshly-prepared, ice-cold RIPA buffer (50 mM Tris, pH 7–8 (Acros Organics #14050-0010), 150 mM NaCl (Fluka #71383), 0.1 % SDS (Fisher #46040CI), 0.5% sodium deoxycholate, 1% Triton-X-100, filter sterilized, stored at 4 °C) with protease & phosphatase inhibitors (1 µg/mL aprotinin, 1 µg/mL leupeptin, 1 µg/mL pepstatin A, 10 mM β-glycerophosphate, and 1 mM sodium orthovanadate) was added into each well, while gently rotating the plate to cover the full surface area. The plates were transferred to the cold room for 15–20 min, with slow rocking. The lysate was scraped off from the wells with a cell scraper. 100 µL of cell lysate from each well were transferred into labeled Eppendorf tubes on ice. Each tube was vortexed three times to homogenize cell debris, keeping other tubes on ice. All tubes were then centrifuged at 4 °C for 15 min at 17,135 g. 80 µL of the supernatant from each tube was transferred into new Eppendorf tubes on ice. These cleared lysate samples were stored at –80 °C for long-term storage or used immediately as below.

Protein quantification. Total protein quantification was done using the BCA-Pierce 660 Assay (Thermo Scientific #23225). As the reference, BSA stock (Thermo Scientific #23209) was used according to the manufacturer protocol. In short, 10 µL of samples and BSA standards were loaded into wells, in triplicate, in a 96-well plate

(Corning #3370). 150 μ L BCA Protein Assay Reagent was loaded into each non-empty well. The plate was covered and incubated at room temperature for 5 min. The absorbance at 660 nm was measured on a plate reader (BioTek #Epoch2). The average reading of blank wells was subtracted from all other readings, and then average readings were calculated. The standard curve was fitted by a polynomial using blank-corrected mean values of each standard condition versus its BSA concentration. The fitted curve was used to determine the protein concentration in each sample.

Immunoblotting and quantification. The lysates were put on ice and the amount of sample to load into each well was calculated using the total protein concentrations determined above (3 μ g loaded here). Each sample was mixed with 2X Sample Buffer (950 μ L of Laemmli's Buffer (BioRad #161-0737), 50 μ L beta-mercaptoethanol (Fisher #034461-100)) in a 1:1 ratio and transferred into a new Eppendorf tube. The sample solutions were heated at 95 °C for 5 min on the heating plate and then briefly spun in benchtop microcentrifuge to return any condensation to the bottom of the tube. 10% acrylamide gels were prepared, and samples were loaded into the wells. A pre-stained protein ladder (LI-COR #928-70000) was loaded in the first and last wells. The gel was run in SDS running buffer (100 mL 10X Tris-Glycine-SDS buffer (IBI Scientific #IB01160) + 900 mL milli-Q water) at constant 220 V until the dye front runs off the gel. Then, wet-transfer to nitrocellulose membrane (0.45 μ m pore size, VWR #10063-173) was done using cold transfer buffer (3.03 g Tris-base (Acros Organics #14050-0010), 14.4 g glycine (Acros Organics #220910050), 100 mL methanol (BDH #BDH1135-4LP), volume adjusted to 1 Liter with milli-Q water) and running the cassette with ice block at constant 100 V for one hour. 1X TBST (Tris-Buffered Saline, 0.1% Tween) was prepared: 100 mL of 10X TBS solution (24 g Tris-base, 88 g NaCl (Fluka #71383), adjusted pH to 7.6, adjusted final volume to 1 Liter with milli-Q water, autoclaved), 1 mL Tween-20 (Fisher #BP337-100), and milli-Q water until final volume of 1 Liter (~900 mL). When the transfer was finished, the membrane was blocked using BSA-TBST blocking buffer (2.5 g bovine serum albumin (Fisher# BP1600-100), 50 mL 1X TBST) for 45 min at room temperature. The blocking buffer was discarded, and the membrane is incubated in primary antibody solution (1:1000 dilution, 10 μ L primary antibody in 10 mL 5% BSA-TBST blocking buffer) overnight in cold room. The primary antibodies used were: AKT_pS473 (Cell Signaling #4060; 1:1000), AKT (Cell Signaling #2920; 1:1000), ERK_pT202_pY204 (Cell Signaling #4370; 1:1000), ERK (Cell Signaling #4696; 1:1000), alpha-tubulin (Novus #NB100-690, 1:1000), and β -actin (LI-COR #926-42212, 1:1000). After primary antibody incubation, membranes were washed three times for 15 min each with 1X TBST at room temperature, with gentle rocking. Then, the membranes are incubated with LI-COR secondary antibodies in 10 mL TBST blocking buffer for 45 min (anti-rabbit 800CW, LI-COR #926-32211 or anti-mouse 680LT, LI-COR #925-68070; 1:8000) at room temperature, with gentle rocking. Membranes were washed three times for 15 min each, with 1X TBST, on rocker. The imaging was done with a LI-COR Odyssey Infrared Imager, where bands were quantified using LI-COR Image Studio Lite v5.2 software (Supplementary Figs. 16–19).

Computational methods

The Bouhaddou2018 model. The Bouhaddou2018 model (Fig. 1a) is one of the largest single-cell mechanistic models for mammalian cell signaling regulating proliferation and death. The first version of the model used as a test case in this work was written in MATLAB (The MathWorks, Inc.)⁴⁰. The model is a hybrid of deterministic and stochastic modules. The deterministic module describes the concentration dynamics of 774 proteins, protein complexes, and post-translationally modified species through 2449 reactions using the Sundials CVODEs package for simulation¹¹⁸. The stochastic module describes gene state (active/inactive) and mRNA birth/death dynamics for 141 genes. The deterministic and stochastic modules exchange information every 30 simulated seconds. In short, the current levels of select protein states can induce changes in gene activation/deactivation and/or mRNA transcription/decay rates. The newly updated mRNA copy numbers change nascent protein translation rates in the deterministic module. See⁴⁰ for further details.

The SPARCED model. We converted the Bouhaddou2018 model into a Python + SBML⁵⁹ format (Fig. 1b). The deterministic module is ultimately encoded in an SBML file (.xml) whereas the stochastic module is written in Python. A foundational and important feature of this recoding effort is that the SBML file is generated from a small set of simple structured input text files (Fig. 1c) via Python scripts. Introduction of such structured input files and associated Jupyter notebooks enables simple alteration of model structure and/or parameter values, for example turning on/off certain interactions. The input files also enable rigorous annotation of model features using, for example, ENSEMBL¹¹⁹ and HGNC¹²⁰ identifiers, which is seldom done in such mechanistic modeling.

Input files. There are six SPARCED model input text files (tab separated values), each with a defined structure as detailed below. The user can change these files to create and compile a new model.

- (1) **OmicsData:** This file (Supplementary Data 1) includes the gene copy number, mRNA copy number, and proteomic data. This input file also

contains rate constants for the stochastic module and initialization procedure. Each row of the file corresponds to one gene and the columns are different data types. The first column is gene name (HGNC identifiers), the second column is gene copy number, the third column is mRNA molecule copy number per cell (mpc), the fourth and fifth columns are rate constants of gene inactivation and activation respectively (s^{-1}), the sixth column is constitutive transcription rate constants (molecules per second), the seventh column is maximal transcription rate constants (molecules per second), the eighth column is mRNA degradation rate constants (s^{-1}), the ninth column is protein copy number (mpc), the tenth column is protein half-life parameters (seconds), and finally the eleventh column is the translation rate constants (s^{-1}). These latest set of rate constants are from literature and provided for genes for which our omics input lacked protein level data. All the rate constants are taken from the Bouhaddou2018 model. Users can add new rows to this file, using RNAseq data to estimate mRNA levels for the genes to be added⁴⁰. When adding genes (rows) to the model, a reasonable starting point for rate constants (or other values), in the absence of any other data, is to use median values from the genes/parameters currently in the model.

- (2) **Species:** This file (Supplementary Data 2) contains information about the species in the deterministic module. Each row corresponds to one species (protein, protein complex, post-transcriptionally modified species). Transcripts (in nM) are also included in this file because they are regarded as species with updated concentrations in the stochastic module every 30 s and are used in translation rate laws. The first column is the species name. Names can be arbitrary so long as they are unique in the model. Importantly, the name list needs to match the first column in the `StoichiometricMatrix` file described below. The second column is the species home compartment. The home compartment of a species defines its cellular localization. A species can reside in a compartment defined in the `Compartments` input file: currently Cytoplasm, Mitochondria, Nucleus, or Extracellular. The third column is initial condition in nM units, with respect to the home compartment volume. These values are taken from the Bouhaddou2018 model, post-initialization. The fourth column is a comma separated list of ENSEMBL gene identifiers corresponding to gene products present in the species.
- (3) **RateLaws:** This file (Supplementary Data 3) has a row for each reaction in the deterministic module. The first column is the unique (arbitrary) name of each reaction. Currently, we named each reaction based on the related sub-module (e.g., vA1-87 for Apoptosis and vC1-104 for Cell Cycle). The number and order of rows in this file should match the columns in the `StoichiometricMatrix` input file defined below. The second column in this file contains the home compartments for the reactions. The designated compartments should be one defined in the `Compartments` input file: currently Cytoplasm, Mitochondria, Nucleus, or Extracellular. The home reaction compartments define the effective search volume for each reaction and is used to rescale concentrations when appropriate. Note that both species and reactions have home compartments defined, where a species can participate in a reaction defined in a different compartment. For instance, the EGF binding to EGFR reaction occurs in extracellular space (volume V_e), where EGF's home compartment is the extracellular space and EGFR's home compartment is cytoplasm (V_c). A volumetric correction for EGFR concentration in this rate law is done by multiplying by the ratio of V_e/V_c . The third column can have either a number or a reaction formula. If it is a number, it means the corresponding reaction is mass-action type, and the number is the rate constant for that reaction in units of nM and seconds. Note that the reactants and products are defined in the `StoichiometricMatrix` input file. If the third column is a formula, it means the reaction will follow that rate law, and the next set columns in that row are the values of each parameter defined in the formula in the third column, again in units of nM and seconds. The rate law can include any species name described in the Species input file. The parameter names in the rate law should start with "k" and be unique in that formula. We distinguish multiple parameter names with an underscore and ascending list of numbers (e.g., kA_1, kA_2). During model generation, all parameter names in this file are re-named in an ascending order based on the number of rate laws. The full list of parameter name/value pairs are outputted into a new file (`ParamsAll`) for user reference.
- (4) **StoichiometricMatrix:** This file (Supplementary Data 4) defines the reaction stoichiometry, and therefore the reactants and products of model reactions. The rows correspond to the Species input file and the columns correspond to the rows in the RateLaws input file. Here, the species and rate law names should match the names defined in Supplementary Data 2 and 3. Each element (starting at the second row and second column index) has a stoichiometric coefficient (typically -2, -1, 0, 1, or 2), where negative sign indicates reactants, and positive sign implicates products of a reaction. We also provide an option to not use the stoichiometric matrix as an input file (see github.com/birtwistlelab/SPARCED/tree/noStoicMat)¹²¹. Instead, the reactions are defined within a new column in the updated `RateLawsNew` input file.

- (5) **GeneReg**: This file (Supplementary Data 5) describes transcriptional activation and inhibition interactions, where rows correspond to genes (the same order as the first column of Supplementary Data 1) and columns to species that are defined as activators or repressors of transcriptional activity. The first column is gene name (HGNC format). There are currently seven more columns in this file, each corresponding to one species defined as an activator or a repressor (e.g., p53 induces p21 transcription or AP1 inhibits cFOS transcription). A single value of zero indicates no effect. A non-zero entry in row *i* and column *j* denotes that species *j* regulates gene *i* transcription. The non-zero entries have the form “A; B”, where “A” is the hill coefficient and “B” is the half-maximal concentration of the species “j” effect. To simplify the input file structure, we use positive values of “A” to denote activation, and negative “A” values to denote inhibition. This file is used by the stochastic module script to update mRNA levels. To add additional transcriptional regulators (activators or repressors) into the SPARCED model, users should add as many columns as new regulator species and populate the columns with corresponding rate constants.
- (6) **Compartments**: This file (Supplementary Data 6) contains the names of compartments in the model (first column), the volume of the compartment in liters (second column), and the corresponding GO-term of the compartment (third column). The compartment names should match the compartment names listed in *Species* and *Ratelaws* input files.
- (7) **Observables**: This file (Supplementary Data 7) contains information about model observables. Each observable corresponds to the compartmental-volume-corrected summation of all formats of a protein. There are 102 observables defined (columns) for the model species (rows) in this file. The entries are either 1 (the species in the row is part of the observable in the column) or 0 (otherwise). The “createModel” Jupyter notebook (Supplementary Data 8) uses this file to define an observables variable as an input for the AMICI model compiler.
- (8) **Initializer (Optional)**: This file (Supplementary Data 16) contains information used for model initialization. Species concentrations (columns 1–2), mRNA level adjustments (columns 3–4), parameter values (columns 5–7), observables to exclude from translation rate adjustments (column 8), and single parameter scan range (columns 9–11) are populated for each step of initialization. The steps used here are shown to work to get a good starting point for serum starved MCF10A cells, which do not undergo apoptosis or enter cell cycle without growth factor stimulation, in deterministic simulation mode.

Dependencies.

- (1) **Docker**: All model dependencies and runtime environments are Dockerized into a downloadable image for self-contained model execution. To run the SPARCED model using Jupyter notebooks (SPARCED-jupyter), Docker must be installed. Then, by downloading the docker image, built on the Ubuntu-18.04 operating system with python3 installed (hub.docker.com/repository/docker/birtwistlelab/sparced-notebook), users can run the Jupyter notebooks defined below in any web-browser within the docker container. The Docker image includes system utilities required for the simulation package AMICI^{67,69}.

Jupyter notebook 1: model creation. The input files described above are processed by the “createModel” Jupyter notebook⁶⁸ (Supplementary Data 8 and converted into an Antimony⁶³ text file (Fig. 1b and Supplementary Data 9). This intermediate step and file provide an additional means to model input, fine-tuning, and alteration for experienced users. It can be explored via any text editor and it lists all elements of the model: species, rate laws, parameters, compartments, and corresponding values (Supplementary Data 9). This text file is then converted into an SBML (.xml) file, using *libantimony* in the same script. The Antimony format does not, to our knowledge, support addition of structured annotations, so the annotations (species and compartments) are added to the newly created SBML file using *libsbml* and the model input files (Supplementary Data 2 and 6). Finally, the annotated SPARCED model file (.xml) is generated (Supplementary Data 10).

Deterministic module. The model SBML file forms the basis of the deterministic module. When run deterministically, the model does not account for stochastic gene switching and mRNA transcription events (see next section). The default parameters and concentrations of the SPARCED model correspond to an average, serum-starved cell state in deterministic mode.

Stochastic module. In addition to the deterministic module, the SPARCED model includes a stochastic module. The stochastic module describes gene states (active/inactive) and mRNA birth/death events for genes (currently 141 of them). The deterministic and stochastic modules exchange information every 30 simulated seconds. The current levels of select protein states can induce changes in gene activation/deactivation and/or mRNA transcription/decay rates. The newly

updated mRNA copy numbers change nascent protein translation rates in the deterministic module. See⁴⁰ for further details.

The stochastic module constitutes two short Python scripts. At the start of each simulation, one of the scripts (*RunPrep.py*) reads in the *OmicData* input file, processes parameter values, and sets the initial transcript levels and stochastic module rate constants. The second script (*SGEmodule.py*) uses information from *RunPrep.py* and species concentrations from the deterministic module to simulate mRNA transcription/degradation and gene activation/inactivation events. One output of the second script is the new concentrations of mRNAs, which is updated in the deterministic module to calculate rates of translation for the next 30 s simulation. The second output is the state of all gene copies (active or inactive).

Jupyter notebook 2: model initialization. Making use of the created model file and the *Initializer* input file, total protein abundance data are converted protein and protein complex starting concentrations by adjusting translation rate constants. The “initializeModel” notebook (Supplementary Data 16) also verifies that the simulated cells behave as expected in serum-starved state. It is possible to modify the steps of initialization to confer new basal behavior (such as cycling) or a mutational effect (loss of PTEN in U87 cells), as introduced by *Initialization* protocol in Bouhaddou2018 model. Running this file is optional and recommended only when new models are created for new cell contexts.

Jupyter notebook 3: model simulation. Supplementary Data 11 includes an example of simulation setup and input parameters of the SPARCED model. The notebook “runModel” imports a user specified model SBML file and stochastic module scripts to run simulations using the AMICI package^{69,122}. AMICI is an interface for Sundials CVODEs solvers that converts SBML files into executable C code for fast simulation. Other required input parameters for model simulation include: a flag to specify if the simulations are fully deterministic (flag = 1) or hybrid (flag = 0), the total simulation time in hours (th), input ligand concentrations, and a flag (1 or 0) to indicate if results should be exported.

The SPARCED-I model. We created a new enlarged version of the SPARCED model called SPARCED-I. We merged Yamada2003 model⁷⁴ of interferon-gamma receptor (IFNGR) signaling into SPARCED. The expansion included addition of 34 new species, 8 corresponding genes, and ~70 reactions. New genes (IFNG, IFNGR, JAK2, STAT1, SHP2, SOCS1, PPN, PPIX) corresponding to the proteins in the model are added as new rows and mRNA levels are inserted into the *OmicData* input file. Gene copy numbers are taken as two⁴⁰. Each new protein, protein complex, and mRNA species are inserted into the *Species* and *StoichiometricMatrix* input files as new rows. Each new reaction is inserted into the *Ratelaws* input files as rows, and into the *StoichiometricMatrix* input file as new columns. The new genes are added as rows to the *GeneReg* file. The final SPARCED-I model has 954 species and 2540 reactions.

Parameter estimation of the SPARCED-I model. After expansion of the SPARCED model with IFNGR pathway and setting the initial species levels from MCF10A cells, SPARCED-I model showed ultrafast receptor and STAT1 activation dynamics inconsistent with biological observations. Thus, we selected six rate constant parameters for calibration based on substantial sensitivity for STAT1 activation dynamics. The parameters calibrated are: (i) STAT1 binding to activated receptor complexes, (ii) nuclear translocation rate of STAT1* dimers, (iii) SOCS1 mRNA translation rate constant, (iv) SOCS1 protein degradation rate, (v) STAT1 unbinding rate from active receptor-SOCS1 complexes, and (vi) set the EIF4E-dependent translation rate constant of SOCS1 to zero (see⁴⁰ for details on the EIF4E effect on translation). We fit the chosen parameters individually, while keeping the values of “best fit” at each step: parameters with minimal sum of squared errors of [STAT1*Dn] dynamics between COPASI and SPARCED-I-10A simulations are retained. The range of variation was set at ± 2 in log₁₀ scale. The SPARCED-I model was then run 1000 simulated hours, without any ligand stimulation, for equilibration.

Context estimation of the SPARCED-I model. The SPARCED model initial conditions are based on serum and growth factor starved MCF10A cells, grown in standard tissue culture plates. However, the experiments involving IFN γ were done in media with horse serum and collagen-coated tissue culture plates. The details of both experimental procedures are explained below. “Bridge” experiments showed that the horse serum upregulates active AKT (ppAKT) levels four-fold prior to growth factor stimulation. We modeled increased basal levels of active AKT with IGF1 treatment for 24 h and found that 0.02 nM IGF1 was consistent with the above experimental observations (Supplementary Figs. 16 and 17). Therefore, we simulate the SPARCED-I model with 0.02 nM IGF1 treatment for 24 h prior to EGF or EGF + IFN γ stimulation.

Cell proliferation (S-phase entry) estimation in SPARCED-I model. The Bouhaddou2018 model used sum of concentrations of Cyclin E, A, and B when greater than 20 nM to decide S-phase entry⁴⁰. Here, we used the same condition as the Bouhaddou2018 model to identify the number of cells in S-phase.

SPARCED-nf (nextflow, model version running on Nautilus-Kubernetes cluster). The SPARCED model can be ported to high-performance cloud computing infrastructures for large-scale simulation, in the SPARCED-nf variant. Specifically, we ran SPARCED-nf simulations using the Pacific Research Platform, a distributed network of academic computing resources organized as a Kubernetes cluster^{64,66}. The prevalence of Kubernetes on both democratized and commercial cloud compute networks makes the model portable, allowing users to run large-scale jobs on distributed supercomputers on a wide range of platforms⁶⁵.

Dependencies.

- (1) **Docker:** As is the practice with Kubernetes-compatible workflows, all model dependencies and runtime environments are *Dockerized* into a downloadable image for self-contained model execution. This means when a job for SPARCED-nf is launched on the Kubernetes cluster, it will download the Docker image for SPARCED-nf and execute the model within that container. The Docker image for SPARCED-nf is built on the Ubuntu-18.04 operating system with python3 installed, as well as a few minor system utilities required for AMICI. The image can be found at <https://hub.docker.com/repository/docker/birtwistlelab/sparced>.
- (2) **Nextflow (nf):** In this cloud-scalable version of the model, the Jupyter notebooks have been converted into python source code and re-modularized for greater parallel-simulation efficiency. The process of creating and executing the model is handled entirely by Nextflow, a workflow-management application and language for building resilient pipelines. When SPARCED-nf is launched, Nextflow begins by creating a head pod on the cluster to coordinate each of the jobs needed to run the model (Supplementary Fig. 1). The head pod creates smaller jobs that each download the containerized dependencies from Dockerhub, pull the model source files from the SPARCED-nf GitHub repository, and run the assigned process. Once the model has completed execution, the output files are saved to a section of the Kubernetes cluster called the persistent volume claim (PVC), where they remain stored in the cloud for user download.

SPARCED-nf model simulation set-up. SPARCED-nf uses the same tab-separated-value input files as SPARCED. For SPARCED-nf to build and execute, the files are copied into the aforementioned PVC for workflow access. This is done with `kube-runner` (<https://github.com/SystemsGenetics/kube-runner>), a submodule for automating common PVC tasks with Kubernetes' `kubectl` tool. The `kube-load.sh` file is used to write new input to the PVC, and `kube-login.sh` is used to access and delete old input files from the cluster.

Along with its scalability, SPARCED-nf is also highly customizable. The `nextflow.config` configuration file is used to define the specifics of simulation scenarios.

- (1) `nextflow.config`: This configuration file has two main sections. In the first section (called K8), users define the Kubernetes namespace specifics and folder configurations. In the second section (called params), users customize runtime arguments for simulation settings. The available parameters are `input_dir_name` (the directory name of the input files), `flag_deterministic` (`flag=1` for deterministic or `flag=0` for hybrid simulations), `sim_time` (simulation time in hours), `Vol_nuclear` (volume of nuclear compartment in liters), `Vol_cyto` (volume of cytoplasmic compartment in liters), `speciesVals` (species names + initial concentration values to start from), `ratelawVals` (parameter names + values), and `numCells` (number of single cells if the simulations are hybrid). Importantly, the “speciesVals” and “ratelawVals” parameters allow users to pass in a formatted string to specify parameter sweeps. Using these in conjunction with the “numCells” parameter, the user can simulate thousands of cells in hundreds of different microenvironments in a single execution.
- (2) `SPARCED-nf:model_build`: Analogous to “`createModel.ipynb`” in SPARCED-jupyter model, this phase of the Nextflow pipeline constructs all necessary files for the model simulation.
- (3) `SPARCED-nf:split_from_params`: This is the major parallelizing step of SPARCED-nf. Having received the relevant model files from the last step, the workflow ingests the `speciesVals`, `ratelawVals`, and `numCells` arguments set by the user in the `nextflow.config`. Using the input files, it creates new input files to satisfy the user-specified parameter sweeps. Each new input file permutation is moved into its own new folder, and each such folder is duplicated `numCells` times.
- (4) `SPARCED-nf:model_run`: This final step of the Nextflow workflow is responsible for model execution and output generation. Each folder created in the previous step above serves as the unique runtime environment in this step. The model pulls assigned simulation input files associated with the folder. Each instance of this step is run in parallel across different simulation environments (Supplementary Fig. 1b). Functionally, the code executed is very similar to the “`runModel.ipynb`” notebook and the model outputs are saved to the PVC.

When the models complete execution, each `SPARCED-nf:model_run` instance saves its output to a unique folder on the PVC. To download these folders to the local filesystem, users can employ `kube-save.sh` (from the `kube-runner` module).

Computational standard-error-of-the-mean. We report the s.e.m. for simulations (Fig. 3c, e, Supplementary Figs. 9a–e, 10a, b, 13a, 13c, and 14c) using the ratio of binomial proportions. See Eq¹. below, where the “Percentage of cells” corresponds to the percentage of cells showing the phenotypic readout (i.e., percentage of cells in S-phase, percent cell death) and the “Number of total cells” is the number of starting single-cell simulations, usually 100.

$$s.e.m. = \sqrt{\frac{\text{Percentage_of_cells} \cdot (100 - \text{Percentage_of_cells})}{\text{Number_of_total_cells}}} \quad (1)$$

Data availability

The LINCS datasets analyzed during the current study are available in the Synapse repository, synapse.org/LINCS_MCF10A. Western blot quantifications are in Supplementary Data 24 and raw blot images are in Supplementary Figs. 18 and 19. Source Data are provided with this paper at <https://doi.org/10.6084/m9.figshare.19658802.v1>¹²³.

Code availability

The final model scripts, files, and information are available in Birtwistle Lab GitHub repository, github.com/birtwistlelab/SPARCED70 and github.com/birtwistlelab/SPARCED/tree/noStoicMat121.

Received: 16 July 2021; Accepted: 7 June 2022;

Published online: 21 June 2022

References

1. Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
2. Ma'ayan, A. et al. Lean Big Data integration in systems biology and systems pharmacology. *Trends Pharm. Sci.* **35**, 450–460 (2014).
3. Gomez-Cabrero, D. et al. Data integration in the era of omics: current and future challenges. *BMC Syst. Biol.* **8**, 11 (2014).
4. Stites, E. C. et al. Use of mechanistic models to integrate and analyze multiple proteomic datasets. *Biophys. J.* **108**, 1819–1829 (2015).
5. Mirza, B. et al. Machine learning and integrative analysis of biomedical big data. *Genes* **10**, 87 (2019).
6. Huang, S., Chaudhary, K. & Garmire, L. X. More is better: recent progress in multi-omics data integration methods. *Front. Genet.* **8**, 84 (2017).
7. Zeng, I. S. L. & Lumley, T. Review of Statistical Learning Methods in Integrated Omics Studies (An Integrated Information Science). *Bioinforma. Biol. Insights*. **12**, 117793221875929 (2018).
8. Jensen, K. J. & Janes, K. A. Modeling the latent dimensions of multivariate signaling datasets. *Phys. Biol.* **9**, 045004 (2012).
9. Adam, G. et al. Machine learning approaches to drug response prediction: challenges and recent progress. *Npj Precis. Oncol.* **4**, 19 (2020).
10. Ianevski, A. et al. Prediction of drug combination effects with a minimal set of experiments. *Nat. Mach. Intell.* **1**, 568–577 (2019).
11. Liu, H. et al. Predicting effective drug combinations using gradient tree boosting based on features extracted from drug-protein heterogeneous network. *BMC Bioinforma.* **20**, 645 (2019).
12. Wong, D. & Yip, S. Machine learning classifies cancer. *Nature* **555**, 446–447 (2018).
13. Ehteshami Bejnordi, B. et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA* **318**, 2199 (2017).
14. Kleppe, A. et al. Chromatin organisation and cancer prognosis: a pan-cancer study. *Lancet Oncol.* **19**, 356–369 (2018).
15. Esteve, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
16. Yu, M. K. et al. Visible machine learning for biomedicine. *Cell* **173**, 1562–1565 (2018).
17. Baker, R. E., Peña, J. M., Jayamohan, J. & Jerusalem, A. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biol. Lett.* **14**, 20170660 (2018).

18. Najafabadi, M. M. et al. Deep learning applications and challenges in big data analytics. *J. Big Data*. **2**, 1 (2015).
19. Wang, F., Casalino, L. P. & Khullar, D. Deep learning in medicine—promise, progress, and challenges. *JAMA Intern. Med.* **179**, 293 (2019).
20. Yang, J. H. et al. A white-box machine learning approach for revealing antibiotic mechanisms of action. *Cell* **177**, 1649–1661. (2019).
21. Kholodenko, B. N., Hancock, J. F. & Kolch, W. Signalling ballet in space and time. *Nat. Rev. Mol. Cell Biol.* **11**, 414–426 (2010).
22. Carrera, J. & Covert, M. W. Why build whole-cell models? *Trends Cell Biol.* **25**, 719–722 (2015).
23. Karr, J. R. et al. A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401 (2012).
24. Carrera, J., Elena, S. F. & Jaramillo, A. Computational design of genomic transcriptional networks with adaptation to varying environments. *Proc. Natl Acad. Sci.* **109**, 15277–15282 (2012).
25. Münzner, U., Klipp, E. & Krantz, M. A comprehensive, mechanistically detailed, and executable model of the cell division cycle in *Saccharomyces cerevisiae*. *Nat. Commun.* **10**, 1308 (2019).
26. Saez-Rodriguez J. & Blüthgen N. Personalized signaling models for personalized treatments. *Mol. Syst. Biol.* <https://onlinelibrary.wiley.com/doi/abs/10.15252/msb.20199042> (2020).
27. Halasz, M., Kholodenko, B. N., Kolch, W. & Santra, T. Integrating network reconstruction with mechanistic modeling to predict cancer therapies. *Sci. Signal.* **9**, ra114 (2016).
28. Macklin, D. N. et al. Simultaneous cross-evaluation of heterogeneous *E. coli* datasets via mechanistic simulation. *Science*. **369**, eaav3751 (2020).
29. Santos, S. D. M., Verveer, P. J. & Bastiaens, P. I. H. Growth factor-induced MAPK network topology shapes Erk response determining PC-12 cell fate. *Nat. Cell Biol.* **9**, 324–330 (2007).
30. Kholodenko, B. N., Demin, O. V., Moehren, G. & Hoek, J. B. Quantification of short term signaling by the epidermal growth factor receptor. *J. Biol. Chem.* **274**, 30169–30181 (1999).
31. Tyson, J. J. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc. Natl Acad. Sci.* **88**, 7328–7332 (1991).
32. Nyman, E., Fagerholm, S., Julleson, D., Stralfors, P. & Cedersund, G. Mechanistic explanations for counter-intuitive phosphorylation dynamics of the insulin receptor and insulin receptor substrate-1 in response to insulin in murine adipocytes. *Febs J.* **279**, 987–999 (2012).
33. Schmierer, B., Tournier, A. L., Bates, P. A. & Hill, C. S. Mathematical modeling identifies Smad nucleocytoplasmic shuttling as a dynamic signal-interpreting system. *Proc. Natl Acad. Sci.* **105**, 6608–6613 (2008).
34. Vilar, J. M. G., Guet, C. C. & Leibler, S. Modeling network dynamics. *J. Cell Biol.* **161**, 471–476 (2003).
35. Kofahl, B. & Klipp, E. Modelling the dynamics of the yeast pheromone pathway. *Yeast* **21**, 831–850 (2004).
36. Tyson, J. J., Chen, K. & Novak, B. Network dynamics and cell physiology. *Nat. Rev. Mol. Cell Biol.* **2**, 908–916 (2001).
37. Puszyński, K., Hat, B. & Lipniacki, T. Oscillations and bistability in the stochastic model of p53 regulation. *J. Theor. Biol.* **254**, 452–465 (2008).
38. Sedaghat, A. R., Sherman, A. & Quon, M. J. A mathematical model of metabolic insulin signaling pathways. *Am. J. Physiol. Endocrinol. Metab.* **283**, E1084–E1101 (2002).
39. Carrera, J. et al. An integrative, multi-scale, genome-wide model reveals the phenotypic landscape of *Escherichia coli*. *Mol. Syst. Biol.* **10**, 735 (2014).
40. Bouhaddou, M. et al. A mechanistic pan-cancer pathway model informed by multi-omics data interprets stochastic cell fate responses to drugs and mitogens. *PLoS Comput. Biol.* **14**. <http://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1005985&type=printable> (2018).
41. Fröhlich, F. et al. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell Syst.* **7**, 567–579 (2018).
42. Dalle Pezze, P. et al. A dynamic network model of mTOR signaling reveals TSC-independent mTORC2 regulation. *Sci. Signal.* **5**, ra25 (2012).
43. Capuani, F. et al. Quantitative analysis reveals how EGFR activation and downregulation are coupled in normal but not in cancer cells. *Nat. Commun.* **6**, 7999 (2015).
44. Orth, J. D., Thiele, I. & Palsson, B. Ø. What is flux balance analysis? *Nat. Biotechnol.* **28**, 245–248 (2010).
45. Lee, J. M. Flux balance analysis in the era of metabolomics. *Brief. Bioinform.* **7**, 140–150 (2006).
46. Sherman, M. S. & Cohen, B. A. A computational framework for analyzing stochasticity in gene expression. Morozov AV, editor. *PLoS Comput. Biol.* **10**, e1003596 (2014).
47. Raj, A., Peskin, C. S., Tranchina, D., Vargas, D. Y. & Tyagi, S. Stochastic mRNA synthesis in mammalian cells. Schibler U, editor. *PLoS Biol.* **4**, e309 (2006).
48. Raj, A. & van Oudenaarden, A. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell* **135**, 216–226 (2008).
49. Faeder, J. R., Blinov, M. L., Goldstein, B. & Hlavacek, W. S. Rule-based modeling of biochemical networks. *Complexity* **10**, 22–41 (2005).
50. Harris, L. A. et al. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics* **32**, 3366–3368 (2016).
51. Xu, W., Smith, A. M., Faeder, J. R. & Marai, G. E. RuleBender: a visual interface for rule-based modeling. *Bioinformatics* **27**, 1721–1722 (2011).
52. Boutillier, P. et al. The Kappa platform for rule-based modeling. *Bioinformatics* **34**, i583–i592 (2018).
53. Lopez, C. F., Muhlich, J. L., Bachman, J. A. & Sorger, P. K. Programming biological models in Python using PySB. *Mol. Syst. Biol.* **9**, 646 (2013).
54. Sneddon, M. W., Faeder, J. R. & Emonet, T. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods.* **8**, 177–183 (2011).
55. Hogg, J. S., Harris, L. A., Stover, L. J., Nair, N. S. & Faeder, J. R. Exact hybrid particle/population simulation of rule-based models of biochemical systems. *PLoS Comput Biol.* **10**, e1003544 (2014).
56. Goldberg, A. P. et al. Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* **51**, 97–102 (2017).
57. Porubsky, V. L. et al. Best practices for making reproducible biochemical models. *Cell Syst.* **11**, 109–120 (2020).
58. Azeloglu, E. U. & Iyengar, R. Good practices for building dynamical models in systems biology. *Sci. Signal.* **8**, fs8 (2015).
59. Hucka, M. et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531 (2003).
60. Keating, S. M. et al. SBML Level 3: an extensible format for the exchange and reuse of biological models. *Mol. Syst. Biol.* **16**. <https://onlinelibrary.wiley.com/doi/abs/10.15252/msb.20199110> (2020).
61. Hoops, S. et al. COPASI—a COMplex PATHway Simulator. *Bioinformatics* **22**, 3067–3074 (2006).
62. Loew, L. M. & Schaff, J. C. The Virtual Cell: a software environment for computational cell biology. *Trends Biotechnol.* **19**, 401–406 (2001).
63. Smith, L. P., Bergmann, F. T., Chandran, D. & Sauro, H. M. Antimony: a modular model definition language. *Bioinformatics* **25**, 2452–2454 (2009).
64. Rensin, D. K. *Kubernetes—Scheduling the Future at Cloud Scale*. <http://www.oreilly.com/webops-perf/free/kubernetes.csp>. (OSCON, 2015).
65. Thurgood, B., Lennon, R. G. Cloud Computing With Kubernetes Cluster Elastic Scaling. In: *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems - ICFNDS*. 1–7 (Paris, France: ACM Press, 2020). <http://dl.acm.org/citation.cfm?doid=3341325.3341995> (2019).
66. Smarr L, et al. The Pacific Research Platform: Making High-Speed Networking a Reality for the Scientist. In *Proc. of the Practice and Experience on Advanced Research Computing*. 1–8 (Pittsburgh PA USA: ACM, 2020). <https://doi.org/10.1145/3219104.3219108> (2018).
67. Fröhlich, F., Theis, F. J., Rädler, J. O., Hasenauer, J. Parameter estimation for dynamical systems with discrete events and logical operations. *Bioinformatics*. **33**,1049–1056 (2016).
68. Kluyver, T. et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. In Loizides F., Schmidt B., (ed) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press. 87–90 (2016).
69. Fröhlich F., Kaltenbacher B., Theis F. J., Hasenauer J. Scalable parameter estimation for genome-scale biochemical reaction networks. *PLOS Comput. Biol.* **13**, e1005331 (2017).
70. Erdem C., Bensman, E. M., Mutsuddy, A., Feltus, F. A., Birtwistle M. R. SPARCED (v1.0.0). Zenodo. <https://doi.org/10.5281/zenodo.6482266> (2022)
71. Nakakuki, T. et al. Ligand-specific c-Fos expression emerges from the spatiotemporal control of ErbB network dynamics. *Cell* **141**, 884–896 (2010).
72. von Kriegsheim, A. et al. Cell fate decisions are specified by the dynamic ERK interactome. *Nat. Cell Biol.* **11**, 1458–1464 (2009).
73. Sullivan, D. M., Latham, M. D. & Ross, W. E. Proliferation-dependent topoisomerase II content as a determinant of antineoplastic drug action in human, mouse, and Chinese hamster ovary cells. *Cancer Res.* **47**, 3973–3979 (1987).
74. Yamada, S., Shiono, S., Joo, A. & Yoshimura, A. Control mechanism of JAK/STAT signal transduction pathway. *FEBS Lett.* **534**, 190–196 (2003).
75. Torgerson, T. R., Colosia, A. D., Donahue, J. P., Lin, Y. Z. & Hawiger, J. Regulation of NF-kappa B, AP-1, NFAT, and STAT1 nuclear import in T lymphocytes by noninvasive delivery of peptide carrying the nuclear localization sequence of NF-kappa B p50. *J. Immunol. Balt. Md 1950.* **161**, 6084–6092 (1998).
76. Tsai, C. C. et al. Glycogen synthase Kinase-3β Facilitates IFN-γ-induced STAT1 activation by regulating Src Homology-2 domain-containing phosphatase 2. *J. Immunol.* **183**, 856–864 (2009).
77. Angel, P., Hattori, K., Smeal, T. & Karin, M. The jun proto-oncogene is positively autoregulated by its product. *Jun/AP-1*. *Cell* **55**, 875–885 (1988).
78. Meissl, K., Macho-Maschler, S., Müller, M. & Strobl, B. The good and the bad faces of STAT1 in solid tumours. *Cytokine* **89**, 12–20 (2017).

79. Schroder, K., Hertzog, P. J., Ravasi, T. & Hume, D. A. Interferon- γ : an overview of signals, mechanisms and functions. *J. Leukoc. Biol.* **75**, 163–189 (2004).
80. Parker, B. S., Rautela, J. & Hertzog, P. J. Antitumour actions of interferons: implications for cancer therapy. *Nat. Rev. Cancer* **16**, 131–144 (2016).
81. Townsend, P. A. et al. STAT-1 Interacts with p53 to enhance DNA damage-induced apoptosis. *J. Biol. Chem.* **279**, 5811–5820 (2004).
82. Seif, F. et al. The role of JAK-STAT signaling pathway and its regulators in the fate of T helper cells. *Cell Commun. Signal* **15**, 23 (2017).
83. Hilton, D. J. Negative regulators of cytokine signal transduction. *Cell Mol. Life Sci.* **55**, 1568–1577 (1999).
84. Pawson, T., Gish, G. D. & Nash, P. SH2 domains, interaction modules and cellular wiring. *Trends Cell Biol.* **11**, 504–511 (2001).
85. Huang, H. et al. Defining the specificity space of the human Src homology 2 domain. *Mol. Cell Proteom.* **7**, 768–784 (2008).
86. Böhmer, F. D. & Friedrich, K. Protein tyrosine phosphatases as wardens of STAT signaling. *JAK-STAT*. **3**, e28087 (2014).
87. Tseng, P. C. et al. Regulation of SHP2 by PTEN/AKT/GSK-3 β signaling facilitates IFN- γ resistance in hyperproliferating gastric cancer. *Immunobiology* **217**, 926–934 (2012).
88. Wang, S. et al. Circulating IGF-1 promotes prostate adenocarcinoma via FOXO3A/BIM signaling in a double-transgenic mouse model. *Oncogene* **38**, 6338–6353 (2019).
89. Weeks, K. L., Bernardo, B. C., Ooi, J. Y. Y., Patterson, N. L., McMullen, J. R. The IGF1-PI3K-Akt Signaling Pathway in Mediating Exercise-Induced Cardiac Hypertrophy and Protection. In *Exercise for Cardiovascular Disease Prevention and Treatment: From Molecular to Clinical, Part 2*. (ed. Xiao, J.) 187–210 (Singapore, Springer 2017) https://doi.org/10.1007/978-981-10-4304-8_12.
90. Melnik, B. C., John, S. M. & Schmitz, G. Over-stimulation of insulin/IGF-1 signaling by Western diet may promote diseases of civilization: lessons learnt from Laron syndrome. *Nutr. Metab.* **8**, 41 (2011).
91. Liu, B. A. et al. SH2 domains recognize contextual peptide sequence information to determine selectivity. *Mol. Cell Proteom.* **9**, 2391–2404 (2010).
92. Hause, R. J., Leung, K. K., Barking, J. L., Ciaccio, M. F., Chuu C.P., Jones, R.B. Comprehensive Binary Interaction Mapping of SH2 Domains via Fluorescence Polarization Reveals Novel Functional Diversification of ErbB Receptors. Katz E., editor. *PLoS One*. **7**, e44471 (2012).
93. Lawlor, M. A. & Alessi, D. R. PKB/Akt: a key mediator of cell proliferation, survival and insulin responses? *J. Cell Sci.* **114**, 2903–2910 (2001).
94. Albeck, J. G., Mills, G. B. & Brugge, J. S. Frequency-modulated pulses of ERK activity transmit quantitative proliferation signals. *Mol. Cell.* **49**, 249–261 (2013).
95. Vadlakonda, L., Pasupuleti, M., Pallu, R. Role of PI3K-AKT-mTOR and Wnt signaling pathways in transition of G1-S phase of cell cycle in cancer cells. *Front. Oncol.* <http://journal.frontiersin.org/article/10.3389/fonc.2013.00085/abstract>.
96. Fröhlich, F. et al. AMICI: High-Performance Sensitivity Analysis for Large Ordinary Differential Equation Models. *ArXiv*. <http://arxiv.org/abs/2012.09122>.
97. Barrette, A. M., Bouhaddou, M. & Birtwistle, M. R. Integrating transcriptomic data with mechanistic systems pharmacology models for virtual drug combination trials. *ACS Chem. Neurosci.* **9**, 118–129 (2018).
98. Barretina, J. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603–607 (2012).
99. Atlas Research N Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* **455**, 1061–1068 (2008).
100. Cancer Genome Atlas Research, N. et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).
101. Hutter, C. & Zenklusen, J. C. The Cancer Genome Atlas: creating lasting value beyond its data. *Cell* **173**, 283–285 (2018).
102. Uhlen, M. et al. Tissue-based map of the human proteome. *Science* **347**, 1260419–1260419 (2015).
103. Thul, P. J. et al. A subcellular map of the human proteome. *Science* **356**, eaal3321 (2017).
104. Uhlen, M. et al. A pathology atlas of the human cancer transcriptome. *Science* **357**, eaan2507 (2017).
105. Bairoch, A. The Cellosaurus, a Cell-Line Knowledge Resource. *J. Biomol. Tech. JBT* **29**, 25–38 (2018).
106. Crudu, A., Debussche, A. & Radulescu, O. Hybrid stochastic simplifications for multiscale gene networks. *BMC Syst. Biol.* **3**, 89 (2009).
107. Gibson, M. A. & Bruck, J. Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. *J. Phys. Chem. A* **104**, 1876–1889 (2000).
108. Yeom, J. S., Georgouli, K., Blake, R. & Navid, A. Towards dynamic simulation of a whole cell model. In: *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 1–10 (2021).
109. Neal, M. L., Gennari, J. H., Waltemath, D., Nickerson, D. P. & König, M. Open modeling and exchange (OMEX) metadata specification version 1.0. *J. Integr. Bioinforma.* **17**, 20200020 (2020).
110. Smith, L. P. et al. SBML Level 3 package: Hierarchical Model Composition, Version 1 Release 3. *J. Integr. Bioinforma.* **12**, 268–268 (2015).
111. Krause, F. et al. Annotation and merging of SBML models with semanticSBML. *Bioinformatics* **26**, 421–422 (2010).
112. Goldberg, A. P. et al. Emerging whole-cell modeling principles and methods. *Curr. Opin. Biotechnol.* **51**, 97–102 (2018).
113. Neal, M. L. et al. SemGen: a tool for semantics-based annotation and composition of biosimulation models. *Bioinformatics*. **35**, 1600–1602 (2019).
114. Hanahan, D. & Weinberg, R. A. Hallmarks of cancer: the next generation. *Cell* **144**, 646–674 (2011).
115. Ghaffarizadeh, A., Heiland, R., Friedman, S. H., Mumenthaler, S. M., Macklin, P. PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. *PLOS Comput. Biol.* **14**, e1005991 (2018).
116. Swat, M. H. et al. Multi-scale modeling of tissues using CompuCell3D. *Methods Cell Biol.* <https://linkinghub.elsevier.com/retrieve/pii/B9780123884039000138> (2012).
117. Szigeti, B. et al. A blueprint for human whole-cell modeling. *Curr. Opin. Syst. Biol.* **7**, 8–15 (2018).
118. Hindmarsh, A. C. et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw. TOMS* **31**, 363–396 (2005).
119. Yates, A. D. et al. Ensembl 2020. *Nucleic Acids Res.* <https://doi.org/10.1093/nar/gkz966> (2019).
120. Braschi, B. et al. Genenames.org: the HGNC and VGNC resources in 2019. *Nucleic Acids Res.* **47**, D786–D792 (2019).
121. Erdem, C., Bensusan, E. M., Mutsuddy, A., Feltus, F. A., Birtwistle, M. R. SPARCED_noStoicMat (v1.0.0nsm). Zenodo; <https://doi.org/10.5281/zenodo.6482267> (2022).
122. Weindl, D., et al. ICB-DCM/AMICI: AMICI v0.11.2. Zenodo. <https://zenodo.org/record/3949231> (2020).
123. Erdem, C., et al. Source_Data. figshare. 11286531547 Bytes. https://figshare.com/articles/dataset/Source_Data/19658802/1 (2022).

Acknowledgements

The authors acknowledge funding from the National Institutes of Health Grants R01GM104184 (M.R.B.), 1R35GM141891 (M.R.B.), U54HG008098-LINCS Center (M.R.B.), U54CA209988 (L.M.H.), and U54HG008100-LINCS Center (L.M.H.), the National Science Foundation Grant CC*-1659300 (F.A.F.), and portions of this work were performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (R.C.B.). C.E. was an NIH-LINCS Postdoctoral Fellow.

Author contributions

Conceptualization: C.E. and M.R.B.; Methodology: C.E., L.M.H., F.A.F., and M.R.B.; Software: C.E., E.M.B., A.M., M.M.S., M.B., R.C.B., and W.D.; Validation: C.E., E.M.B., A.M., R.C.B., and M.R.B.; Formal analysis: C.E.; Investigation: C.E., S.M.G., and L.M.H.; Resources: L.M.H., F.A.F., and M.R.B.; Data curation: C.E., A.M., W.D., and S.M.G.; Writing – Original Draft: C.E. and M.R.B.; Writing – Review & Editing: C.E. and M.R.B.; Visualization: C.E., E.M.B., and M.B.; Supervision: C.E. and M.R.B.; Project administration: C.E. and M.R.B.; Funding acquisition: L.M.H., F.A.F., and M.R.B.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-022-31138-1>.

Correspondence and requests for materials should be addressed to Cemal Erdem or Marc R. Birtwistle.

Peer review information *Nature Communications* thanks James Faeder and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Ross Cloney. This article has been peer reviewed as part of Springer Nature's **Guided Open Access** initiative.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022