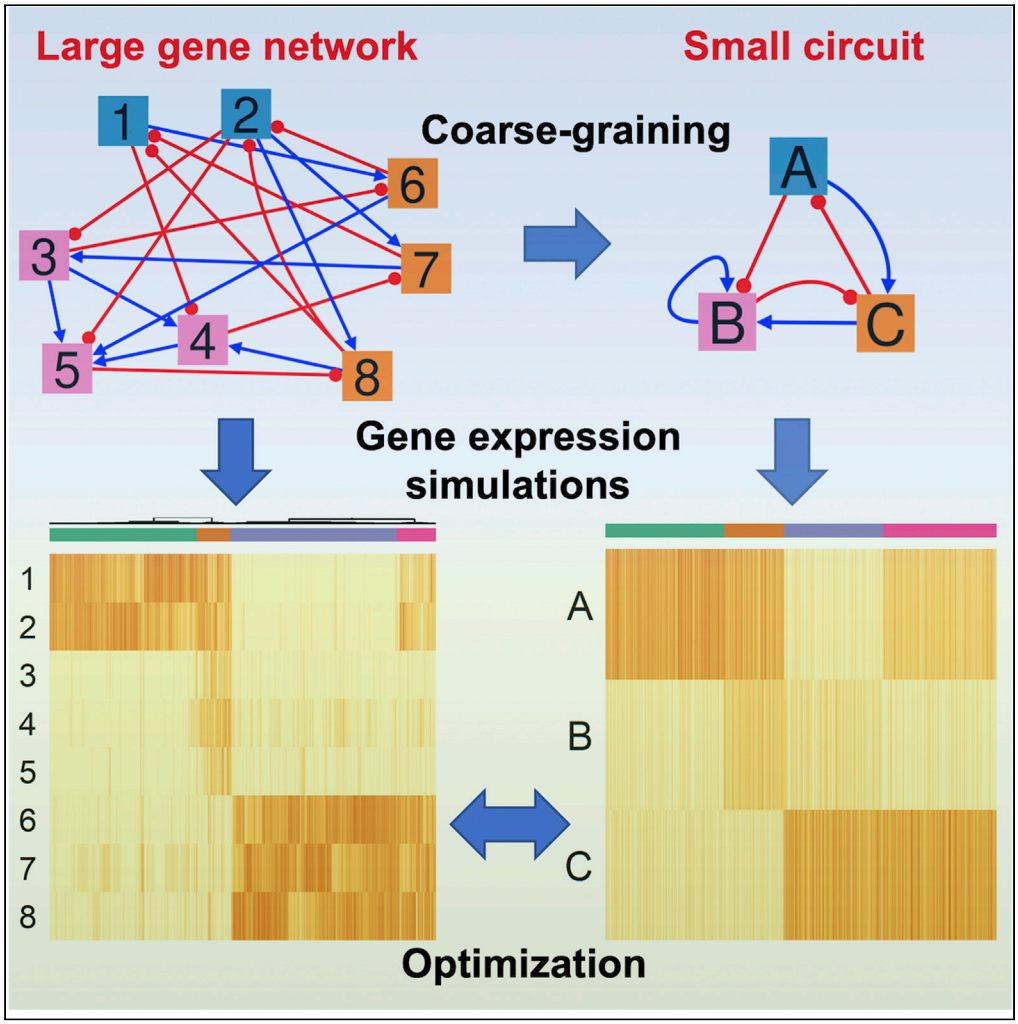
iScience

CellPress
OPEN ACCESS

Article

A data-driven optimization method for coarse-graining gene regulatory networks



Cristian Caranica, Mingyang Lu

m.lu@northeastern.edu

Highlights

Coarse-graining gene regulatory networks by gene grouping and topology optimization

Network coarse-graining reveals important genes and edges from a regulatory network

The method can handle networks with missing, wrong-signed, or redundant edges

Coarse-grained gene circuits recapitulate multiple gene expression states

Caranica & Lu, iScience *26*, 105927
February 17, 2023
https://doi.org/10.1016/j.isci.2023.105927

iScience

CellPress
OPEN ACCESS

Article

A data-driven optimization method for coarse-graining gene regulatory networks

Cristian Caranica[1,2] and Mingyang Lu[1,2,3,4,*]

SUMMARY

One major challenge in systems biology is to understand how various genes in a gene regulatory network (GRN) collectively perform their functions and control network dynamics. This task becomes extremely hard to tackle in the case of large networks with hundreds of genes and edges, many of which have redundant regulatory roles and functions. The existing methods for model reduction usually require the detailed mathematical description of dynamical systems and their corresponding kinetic parameters, which are often not available. Here, we present a data-driven method for coarse-graining large GRNs, named SacoGraci, using ensemble-based mathematical modeling, dimensionality reduction, and gene circuit optimization by Markov Chain Monte Carlo methods. SacoGraci requires network topology as the only input and is robust against errors in GRNs. We benchmark and demonstrate its usage with synthetic, literature-based, and bioinformatics-derived GRNs. We hope SacoGraci will enhance our ability to model the gene regulation of complex biological systems.

INTRODUCTION

One of the major challenges in systems biology is to understand how gene regulatory networks (GRNs) enable the creation and maintenance of cellular states and how they control and drive cellular state transitions in biological processes, such as cell differentiation and disease progression.[1,2] Common ways to construct GRNs[3,4] are either by the bottom-up approach,[5–7] in which gene regulatory interactions are derived from literature data, or by the top-down approach[8] in which bioinformatics methods are applied on genome-wide omics data,[9–13] such as transcriptomics data. The constructed GRNs often contain a large number of genes and edges, making it hard to understand how GRNs operate and control their dynamics.

One strategy to address these issues is network coarse-graining, which constructs small core regulatory circuits that capture the dynamical behavior of large GRNs. Large GRNs often contain many genes and edges with redundant regulatory roles and functions, thus the redundant components can be combined into simple network models.[14–16] Compared to a large GRN, a small circuit model is more likely to capture the most important network functions and to reveal the roles of genes and the relationship between genes. Moreover, it is easier to fix errors or make changes in small GRNs than in those large counterparts because of much reduced searching space.

Coarse-graining, sometimes referred to as model reduction, is a popular technique in physics, chemistry, and engineering disciplines.[17–22] Various coarse-graining methods have been developed in systems biology to model GRNs.[23–25] The most representative ones are based on timescale exploitation,[26] optimization,[27,28] lumping,[29] or singular value decomposition.[30] However, most existing methods require the knowledge of the mathematical rate equations and detailed kinetic parameters of the full models, which is not available in many cases. Moreover, it is often difficult for the existing methods to establish high-quality models from GRNs containing missing or inaccurate regulatory interactions.

To alleviate these limitations, here we present a new GRN coarse-graining method, named *Sampling coarse-Grained circuits* (SacoGraci), to construct optimized small circuit topologies that capture the gene expression states of a large GRN. SacoGraci first clusters the genes and models describing the network's behavior, followed by an optimization process that samples coarse-grained circuits (CGCs)

[1]Department of Bioengineering, Northeastern University, Boston, MA 02115, USA

[2]Center for Theoretical Biological Physics, Northeastern University, Boston, MA 02115, USA

[3]The Jackson Laboratory, Bar Harbor, ME 04609, USA

[4]Lead contact

*Correspondence: m.lu@northeastern.edu

https://doi.org/10.1016/j.isci.2023.105927

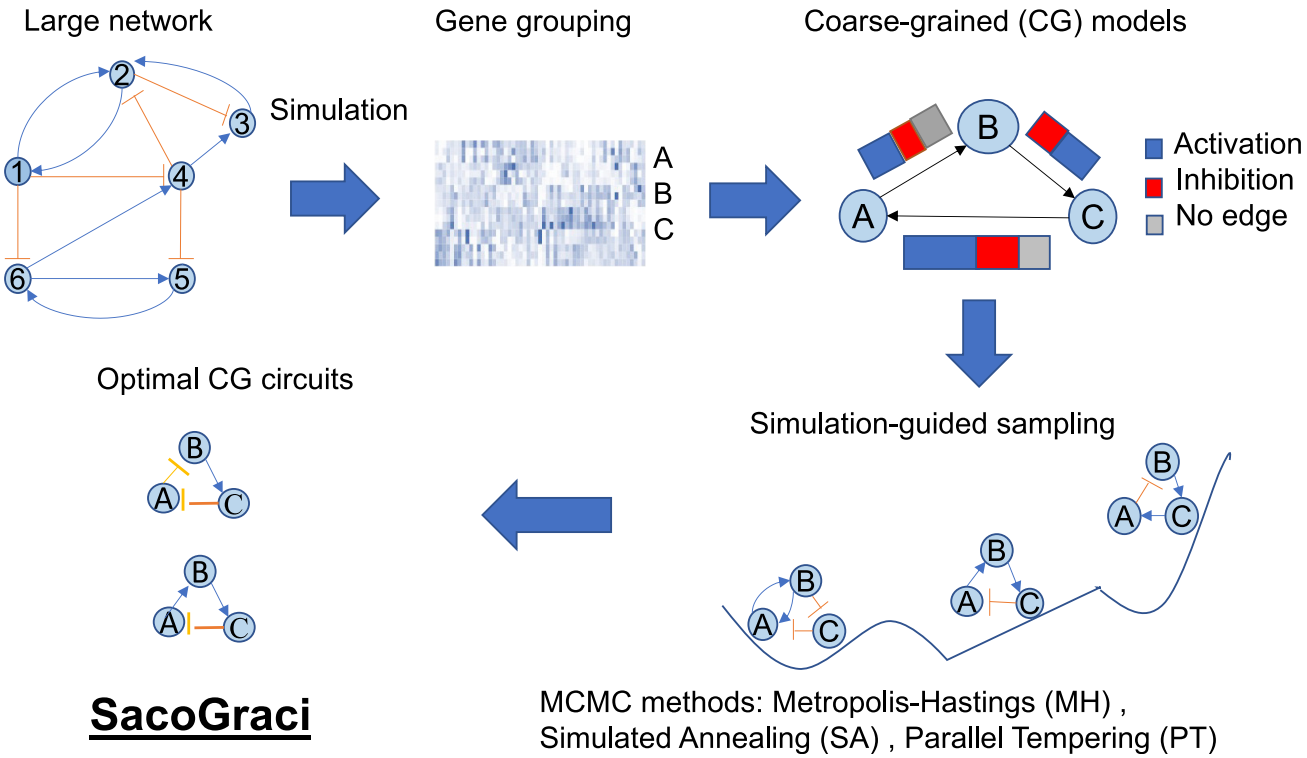


Figure 1. Workflow of the gene network coarse-graining algorithm

The goal of algorithm is to construct, from a large gene regulatory network, coarse-grained circuits (CGCs) models that capture the network's behavior. First, given the topology of the full network, mathematical modeling method RACIPE simulates the steady-state gene expression profiles from an ensemble of randomly generated models. Second, cluster analysis is applied to simulated data to identify clusters of genes, which represent coarse-grained nodes, and clusters of gene expression profiles, which represent network states. Third, CGCs are constructed by Markov chain Monte Carlo (MCMC)-based optimization, where circuit edges are chosen according to the regulatory interactions of the full network. The scoring function of the optimization is based on the similarity of the states of CGC and the states of the full network.

producing gene expression patterns similar to those of the GRN. As the core of SacoGraci, network optimization uses a new scoring function and multiple sampling schemes. Here, the scoring function quantifies the dissimilarity of gene expression states between the small circuit and the large GRN based on the simulations of gene expression by an ensemble-based mathematical modeling method named RACIPE.[31,32] Because RACIPE captures GRN's gene expression states from an ensemble of models with randomly generated kinetic parameters,[33–37] there is no need to specify or optimize kinetic parameters in each score evaluation. Such a feature allows us to focus on the optimization of circuit topology by a Markov chain Monte Carlo (MCMC)-based sampling scheme,[38] where the sampling space is determined by the topology of the large GRN. SacoGraci has the advantage over most existing methods in that it only requires the GRN topology as the input; it is robust against errors in large GRNs; and the optimization is independent to the choice of model parameters.

In the following, we will first provide an overview of SacoGraci, with the detailed algorithms explained in STAR Methods. We will then show the benchmark of SacoGraci on a large series of GRNs expanded from two synthetic gene circuits and perturbed in the network topology by different levels. Finally, we will illustrate its applications on four biological GRNs, including both literature-based and bioinformatics-derived networks.

RESULTS

Overview of the coarse-graining algorithm SacoGraci

SacoGraci takes the topology of a large GRN as the input and identifies the optimal CGCs that best capture the gene expression states of the full GRN. The workflow of SacoGraci is illustrated in Figure 1. First, RACIPE is applied to the full GRN to simulate the stable steady-states gene expression profiles of an

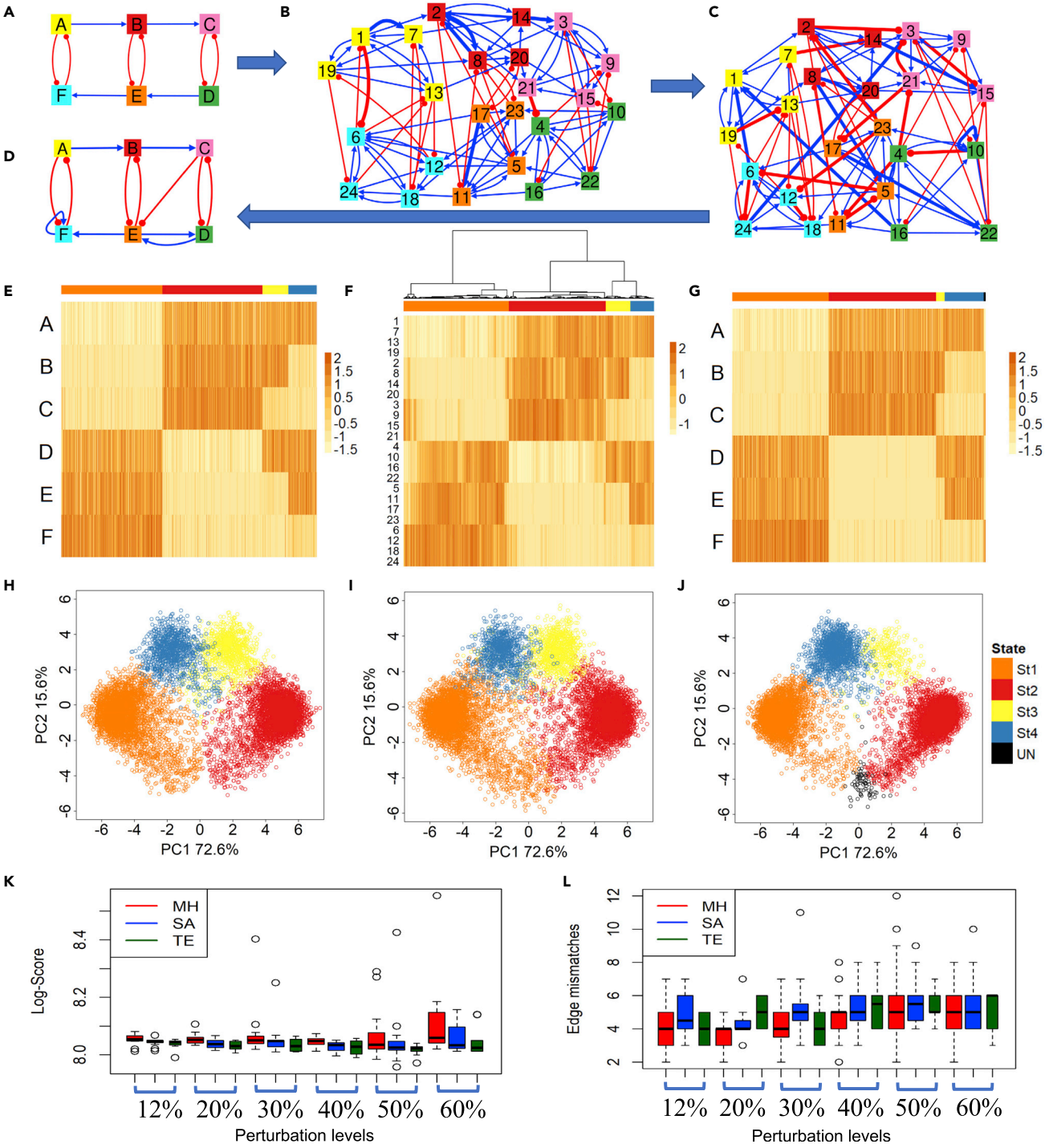


Figure 2. Benchmark tests on a synthetic gene circuit of three coupled toggle switches

(A) Topology of the original circuit.

(B) Topology of an expanded network. Every node of the original circuit is expanded into four nodes of the same color. Edges that will not be present in the perturbed network are shown in bold.

(C) Topology of a perturbed network with a 30% edge perturbation level. Edges that have been added to the expanded network or have changed the sign are shown in bold.

(D) Topology of the best last-iteration CGC obtained after running all MCMC methods for networks with 30% perturbation level.

(E) Heatmap of the RACIPE-simulated data for the original circuit.

Figure 2. *Continued*

(F) Heatmap of simulated gene expression of the expanded network. It shows the hierarchical clustering analysis (HCA) of the RACIPE models using Ward.D2 as linkage method and 1 minus Pearson correlation as distance. Four well-separated clusters are determined based on HCA.

(G) Heatmap of the simulated RACIPE models for the best CGC.

(H) Scatterplot of the simulated gene expression data of the original circuit projected onto the first two principal component (PC) axes of the expanded network's simulated data. RACIPE models belonging to the same cluster are colored with the same color.

(I) Scatterplot of the simulated data of the expanded network projected onto its first two PCs.

(J) Scatterplot of the simulated data of the best CGC projected onto the first two PCs of the simulated data of the expanded network.

(K) Boxplots of the distributions of the last scores (log-scale) obtained by each MCMC method for each network perturbation level.

(L) Boxplots for the distributions of edge mismatches of the last-iteration circuits and the original circuit for each MCMC method and each network perturbation level. For each boxplot, the lower edge, middle edge, and upper edge correspond to 25^{th} percentile, 50^{th} percentile and 75^{th} percentile, respectively. The whiskers expand out to 1.5 times interquartile range. The black models in panels G and J correspond to models unclassified to any cluster (labeled as UN)

ensemble of mathematical models with randomly selected kinetic parameters. Second, clustering analysis is applied to the simulated gene expression profiles to identify model clusters, each of which defines a *network state*, and gene clusters, each of which defines a node in the CGC. The number of network states and the number of circuit nodes can be either determined by biological context or be considered as hyperparameters for optimization. Third, an optimization process is then applied to sample candidate circuits and obtain the optimal CGCs by a MCMC method using a scoring function that quantifies the mismatch between the network states of the CGC and those of the full GRN. The regulatory interactions are sampled according to edge type (*i.e.*, activation, inhibition, or no interaction) distributions determined from the full GRN. During the score evaluation, RACIPE is applied to each candidate circuit to obtain its stable steady-state gene expression profiles. Details of the SacoGraci procedures are described in STAR Methods.

Benchmarking SacoGraci using synthetic circuits

We first benchmarked SacoGraci in identifying high-quality CGCs from a large GRN using two small synthetic circuits. We started from each small circuit and expanded it to a large network with redundant regulatory interactions derived from the small circuit. In the benchmark test, we applied SacoGraci to the large network (the input) and evaluated whether the identified CGC recovers not only the gene expression distribution of the original circuit but also its topology (ground truth). Many biological networks may contain inconsistent edges (*i.e.*, different excitatory or inhibitory edge types) even for genes with similar roles. Due to the availability of biological data on gene regulatory interactions, literature-derived GRNs may also contain inaccurate and missing edges. Thus, to evaluate the robustness of the network coarse-graining, we also tested SacoGraci on large networks with different levels of edge perturbations.

The first test case is a circuit of three coupled toggle-switches (diagram in Figure 2A). To generate the large network, we expanded each of the six nodes of the original circuit to four genes (details in STAR Methods). The final large network contains 24 genes (diagram in Figure 2B), where genes with the same color correspond to the colored node of the original circuit. We clustered the models and the genes using hierarchical clustering analysis (HCA) with Ward.D2[39] and one minus Pearson correlation as the distance function. As shown in the heatmaps of simulated gene expression profiles generated by RACIPE, the gene expression profiles from the large network (Figure 2F) form clusters of expression patterns similar to those obtained from the original circuit (Figure 2E). Genes corresponding to the same node were also clustered together, indicating that they play similar roles in network behavior. Thus, we chose four network states and four gene clusters for network coarse-graining. The network states were clearly observed in the scatterplots of gene expressions projected onto the first two principal components (PCs) of large GRN's simulated data. (Figures 2H and 2I).

From the expanded network, perturbed networks for six different perturbation levels (each level with ten networks) were generated by randomly deleting, adding, and changing the signs of a proportion of edges, as described in STAR Methods. These perturbed networks were used as the inputs to test SacoGraci (i.e., the perturbed network defines the edge type distributions for sampling CGCs). An example of a 30% edge-perturbed network is shown in Figure 2C. For this perturbation level, SacoGraci identified the CGC shown in Figure 2D as the best-scored last-iteration circuit across all sampling methods. The optimized CGC generates similar network states (Figures 2G and 2J) as the large network. The CGC also nicely captures the

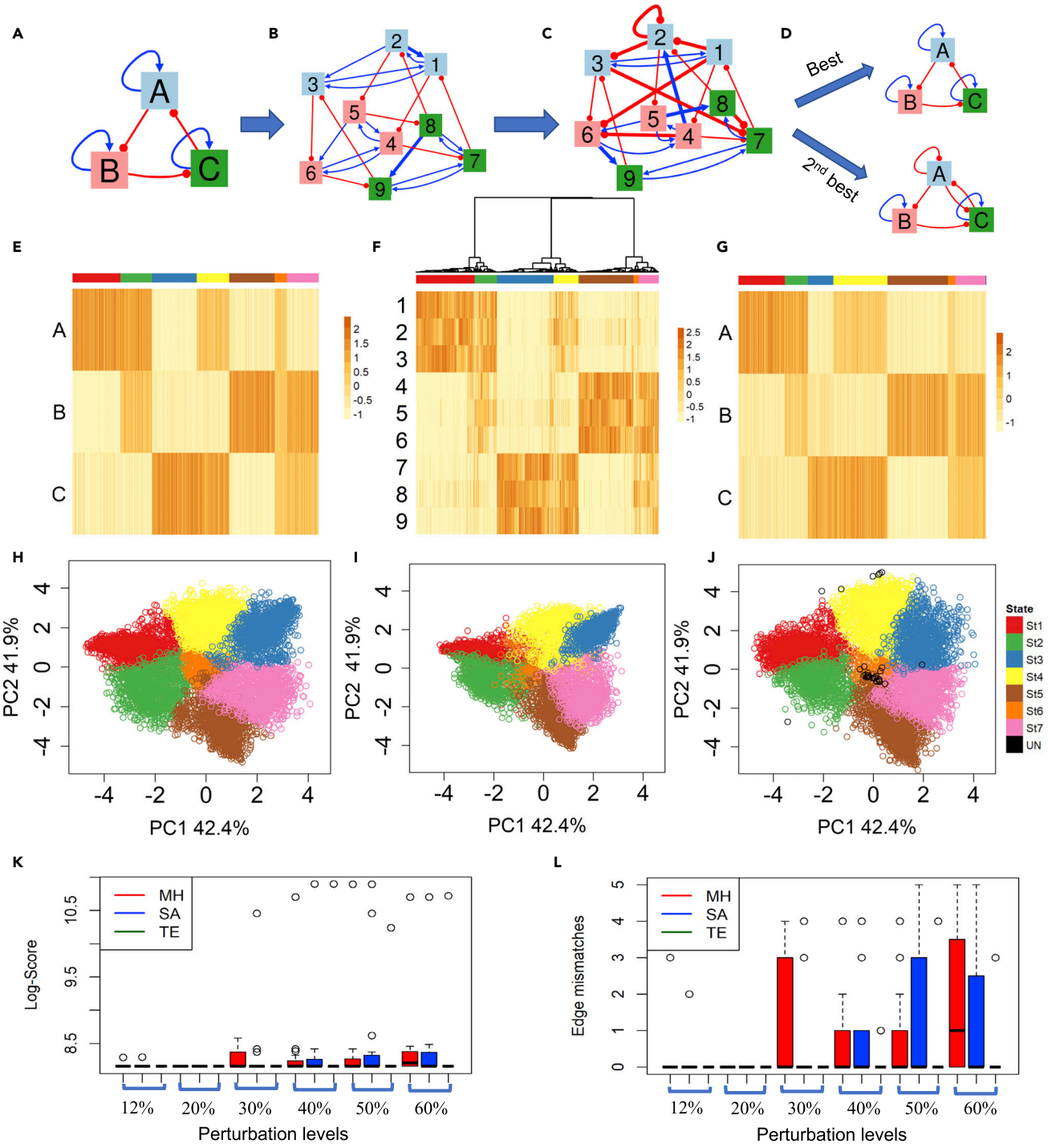


Figure 3. Benchmark tests on a synthetic gene circuit of a repressilator with self-activation loops

(A) Topology of the original circuit.

(B) Topology of an expanded network. Every node of the original circuit is expanded into three nodes of the same color. Edges that will not be present in the perturbed network are shown in bold.

(C) Topology of a perturbed network with a 50% edge perturbation level. Edges that have been added to the expanded network or have changed the sign are shown in bold.

(D) Topology of the second best last-iteration CGC obtained after running all MCMC methods across all perturbation levels considered. The best last-score CGC is the same as the original circuit.

(E) Heatmap of the RACIPE-simulated data for the original circuit.

Figure 3. *Continued*

(F) Heatmap of simulated gene expression of the expanded network. It shows the hierarchical clustering analysis (HCA) of the RACIPE models using Ward.D as linkage method and 1 minus Pearson correlation as distance. Seven well-separated clusters are determined based on HCA. Gene clustering puts the genes of the same color into the same group.

(G) Heatmap of the simulated RACIPE models for the second best CGC.

(H) Scatterplot of the gene expression data of the original circuit projected onto the first two PCs of the expanded network's simulated data. RACIPE models belonging to the same cluster are colored with the same color.

(I) Scatterplot of the simulated data of the expanded network projected onto the first two PCs.

(J) Scatterplot of the simulated data of best CGC projected onto first two PCs of the simulated data of the expanded network.

(K) Boxplots of the distributions of the last scores (log-scale) obtained by each MCMC method for each network perturbation level.

(L) Boxplots for the distributions of edge mismatches of the last-iteration circuits and the original circuit for each MCMC method and each network perturbation level. For each boxplot, the lower edge, middle edge, and upper edge correspond to 25th percentile, 50th percentile and 75th percentile, respectively. The whiskers expand out to 1.5 times interquartile range.The black models in panels G and J correspond to models unclassified to any cluster (labeled as UN).

network states of the original circuit (Figures 2E and 2H), except for a lower proportion of models in the third state (yellow) from the CGC.

Next, we compared the performance of circuit optimization using three different MCMC sampling methods: Metropolis-Hastings (MH), Simulated Annealing (SA), and Parallel Tempering (TE). Figures 2K and 2L summarize the performance of all three MCMC methods for networks perturbed by different edge-perturbation levels. Figure 2K shows the last-iteration scores of the sampling methods for different levels of edge perturbation. The score measures the mismatch between the expression profiles of the CGC and those of the large network—the smaller the score, the better the matching. As expected, for each perturbation level, the median TE score is lower than the median SA score, which is lower than the median MH score. It is remarkable that, for each MCMC method, the scores do not increase significantly as we increased the perturbation level from 12% to 50%. Only for the 60% edge-perturbation level, the median score of MH is slightly higher than the other median MH scores. Even for this high perturbation level, we can obtain low scores. The best overall score was obtained by an SA run from a network with 50% of edge perturbations. For each MCMC sampling method, we calculated the number of edge mismatches (existent vs. nonexistent, or different edge types) between the last-iteration CGC and the original circuit. For instance, the number of edge mismatches between the original circuit (Figure 2A) and the optimal circuit (Figure 2D) is three, because the latter has three edges not existing in the original circuit. As shown in Figure 2L, the number of edge mismatches is on average between four and six for all MCMC methods and perturbation levels. Interestingly, many of these mismatches are self-activation loops due to the mutual activations between genes from the same group in the expanded networks (a median of 66.67%); but these self-activation loops are absent in the original circuit. Otherwise, we observed low levels of mismatches in all sampling methods. Moreover, we found all MCMC runs converged within 1400 iterations (details in the STAR Methods section "detailed implementation of the MCMC sampling methods" and Figure S9). Overall, we found that the circuit optimization procedures are effective in identifying CGCs even for high levels of perturbations.

The second test case is a repressilator circuit,[40] consisting of three self-activating nodes *A*, *B*, and *C*, where *A* inhibits *B*, *B* inhibits *C*, and *C* inhibits *A* (diagram in Figure 3A). To generate the expanded network, we expanded each node to threes genes, *i.e.*, *A* is expanded to genes 1, 2, and 3; *B* to genes 4, 5, and 6; *C* to genes 7, 8, and 9 (Figure 3B). Similar to the previous case, we expanded the circuit to a large GRN by connecting multiple copies of the original circuit with consistent edges. We also generated large networks with different levels of edge perturbations. Figures 3C and 3D show an example of such a perturbed network with 50% of edge perturbations and the last-iteration CGC obtained from SacoGraci. In this example, the original circuit, the expanded network, and the CGC all generate very similar gene expression profiles, as illustrated in Figures 3E–3J. Note that a repressilator is usually modeled to generate oscillatory dynamics.[40] However, from an ensemble of models with randomly generated kinetic parameters, the steady-state gene expression profiles capture the network states along an oscillatory trajectory.[34] This explains why the coarse-graining scheme works very well for an oscillatory system here, where we match the steady-state gene expression profiles. The last-iteration scores of all MCMC methods (Figure 3K) are mostly very low, indicating that all methods obtained CGCs with a very good fit most of the time. In fact, 90% of the MCMC runs produce the same last-iteration circuit as the original circuit. The original circuit also produces the lowest score in this case. The number of edge mismatches between the last-iteration circuits and the original circuit (Figure 3L) is much lower than those from the first example. We observed a

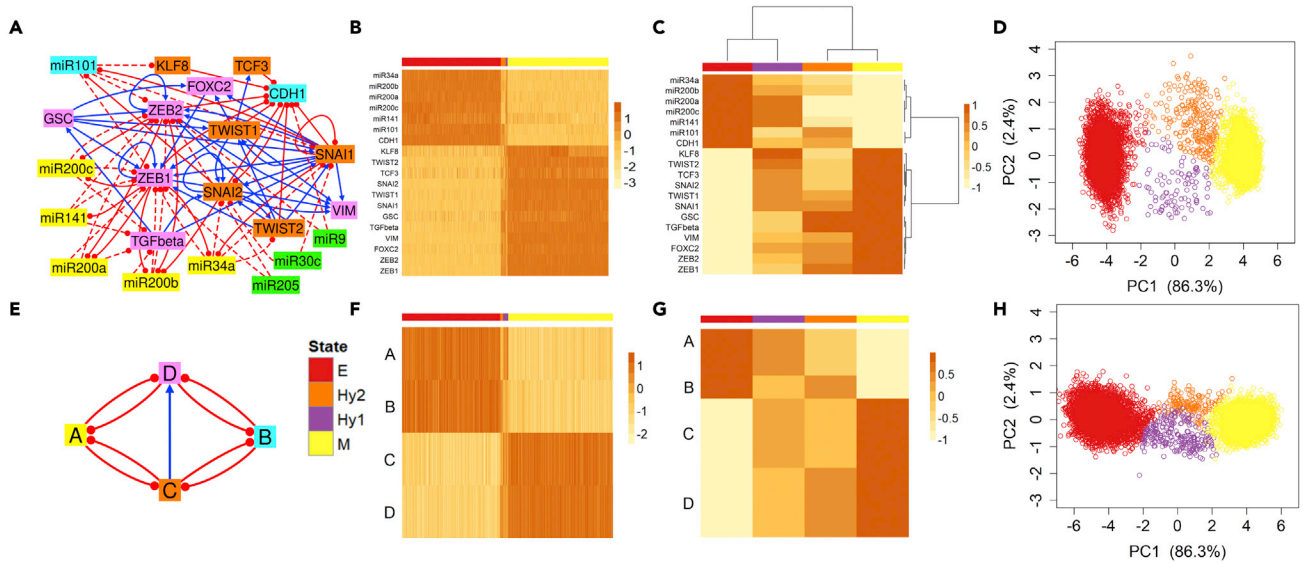


Figure 4. Coarse-graining a gene network regulating epithelial-mesenchymal transition (EMT)

(A) EMT network topology.

(B) Heatmap of the RACIPE-simulated gene expression profiles from the EMT network. Each row represents a network gene, and each column represents a simulated model. Clustering of the simulated RACIPE models was done using k-means clustering (k = 4) on the 2D points obtained from the simulated data projected onto its first two PCs.

(C) Heatmap of the median values of the simulated models for each gene in each cluster. Red and yellow clusters correspond to epithelial and mesenchymal states, respectively. Purple and orange clusters correspond to the hybrid states. Gene clustering was performed by HCA of this heatmap, where four gene clusters were identified. Genes belonging to the same gene cluster are illustrated with the same color as those in the network diagram shown in panel A.

(D) Scatterplot of the simulated data of the full network projected onto its first two PCs.

(E) Network topology of the optimized CGC (second best).

(F) Heatmap of the RACIPE-simulated models from the optimized CGC. Each row represents a CGC node, and each column represents a RACIPE model.

(G) Heatmap of the median expression values of the simulated data of the optimized CGC. Each row represents a CGC node, and each column represents a model cluster.

(H) Scatterplot of the simulated data of the CGC projected onto the first two PCs of the full network (panel D). Models corresponding to the same cluster are illustrated with the same color.

slightly worse performance of circuit optimization for networks with high perturbation levels; yet TE usually has the most stable and best performance for those cases. The performance of SacoGraci was also similar when the circuits were expanded to GRNs with different group sizes and different ways of edge perturbations (Figures S1–S4).

Coarse-graining a GRN of epithelial-to-mesenchymal transition (EMT)

In the previous section, we showed the performance of SacoGraci on a series of large GRNs expanded from two synthetic circuits. Next, we illustrate the application of SacoGraci on four biological GRNs, including three literature-derived networks and a network generated by bioinformatics analysis.

The first example is a GRN responsible for the decision making of EMT. During EMT, epithelial cells undergo a phenotypic transition to mesenchymal cells by losing cell adhesion and gaining high motility.[41] EMT has been found to play crucial roles in embryonic development, wound healing, and cancer metastasis.[42,43] The EMT GRN, which consists of 13 TF, 9 microRNAs, and 82 regulatory links between them (network topology shown in Figure 4A), was previously constructed[31] using the gene regulatory data from the literature and Ingenuity Pathway Analysis.[44]

We first applied RACIPE to the EMT GRN and obtained stable steady states from 10,000 models (one steady state per model). From hierarchical clustering analysis of the simulated gene expression profiles (Figure S5), we observed two major clusters of models, representing the epithelial (E, with high CDH1 levels) and mesenchymal (M, with high VIM levels) states. We also observed a small fraction of models with intermediate levels of gene expression. However, the models do not form distinct clusters in HCA,

thus they are difficult to be separated from the E and M states. But when the same data were projected to its first two PCs (Figure 4D), we can clearly observe these models (colored in purple and orange) away from the E (red) and M (yellow) states. We performed k-means clustering (k = 4) of the data projected to the first two PCs and successfully identified four distinct clusters of models associated with the E, M states and two hybrid EMT states (Hy1 and Hy2). We opted to do the clustering in a supervised manner, by providing the center clusters as the input to the 4-means algorithm. The model clustering is consistent with experimental observations of hybrid EMT phenotypes.[45–47]

Once the model clustering has been determined, we then grouped genes based on the gene expression patterns of different EMT states. We started from the RACIPE-simulated gene expression profiles (Figure 4B), computed the median expression values for each gene in each model cluster, and then performed HCA again (Figure 4C). Here, the median values were computed in Figure 4C to emphasize on the gene expression differences in the rare hybrid EMT states. From the last step, we obtained the gene clustering, which we used to determine the nodes of CGCs. We decided that four gene clusters will be sufficient to capture the expression patterns. The gene grouping, as annotated in Figure 4A, puts miR34a, miR141, miR200a, miR200b, and miR200c in node *A*; miR101 and CDH1 in node *B*; KLF8, TCF3, SNAI1, SNAI2, TWIST1, and TWIST2 in node *C*; and GSC, FOXC2, TGFβ, VIM, ZEB1, and ZEB2 in node *D*. As the genes miR205, miR9, and miR30c act as input nodes, *i.e.*, nodes not regulated by other genes, they were not considered for gene grouping. Genes in node *A* are major miRNAs responsible for inducing the epithelial state.[48] CDH1 (*i.e.*, E-Cadherin) in node *B* is known to be regulated by major EMT master regulators, such as SNAIL and ZEB families, and serves as a readout in previous EMT gene regulatory circuit models.[16,49] MiR101 forms a toggle switch with SNAI1, while CDH1 forms a toggle switch with SNAI2. Also, miR101 inhibits ZEB1 and ZEB2, which are repressors for CDH1. All these interactions lead to a 0.91 Pearson correlation between CDH1 and miR101 in simulated gene expression. Thus, CDH1 and miR101 were grouped together in node *B*. Genes in nodes *C* and *D* are mainly regulators responsible for mesenchymal states.

With the model clustering and gene grouping being determined, SacoGraci can then be applied to identify the optimal CGCs with MCMC sampling. We ran all three MCMC methods and searched for the best-performing CGCs. An example of such CGC (second best last-iteration circuit) is illustrated in Figure 4E. As shown in the heatmaps (Figures 4F and 4G) and PCA results (Figure 4D and 4H), the simulated gene expression profiles from the CGC resemble those from the original EMT network, indicating a successful coarse-graining. While the symmetry of the CGC would suggest that nodes *A* and *B* play similar roles and they might as well be combined into a single node, we note that having both *A* and *B* help to distinguish between the two hybrid states.

In the optimal CGC, node *A* forms two double-negative feedback loops with nodes *C* and *D*, respectively. These circuit motifs are consistent with the double-negative feedback loops formed by the miR34 and SNAIL families (as in the case of the *A*-*C* loop) and those formed by the miR200 and ZEB families (as in the case of the *A*-*D* loop). Both SNAIL and ZEB inhibit CDH1, CDH1 inhibits SNAI2, and miR101 inhibits ZEB; all of these explain the *B*-*C* and *B*-*D* double-negative feedback loops. Interestingly, the CGC has a directed interaction from node *C* to *D*, consistent with the gene regulation from SNAIL to ZEB. It is also supported by the biological observations that the node *C* genes, such as TWIST and SNAIL families, are usually activated prior to the activation of ZEB family.[16,45,50] Moreover, the mono-directional interaction is also largely consistent with the edge-type distributions from the large GRN. Taken together, the CGC captures the major regulatory features of the large EMT network. Note that, in another CGC model (best last-iteration CGC, as shown in Figure S6), *C*-*to*-*D* becomes bidirectional, but *C*-*to*-*A* becomes mono-directional. Presumably, some asymmetry is needed to match well with full-network gene expression. Overall, the former CGC model is more consistent with experimental evidence of EMT.

Coarse-graining a GRN of small-cell lung cancer (SCLC)

The second example is a GRN of SCLC[51] consisting of 33 transcription factors (Figure 5A). Previous studies have shown that the network allows neuroendocrine/epithelial, mesenchymal-like, and hybrid phenotypes.[51] We also observed gene expression clusters associated with these phenotypes from the simulations of the SCLC GRN using RACIPE (Figures 5B and 5C). From the HCA of the simulated gene expression data (Figure 5B), we decided to group the genes into four nodes (grouping scheme illustrated in Figure 5A by gene colors). With the defined model and gene grouping scheme, we performed the circuit optimization with MCMC methods, where the best last-iteration CGC is shown in Figure 5D.

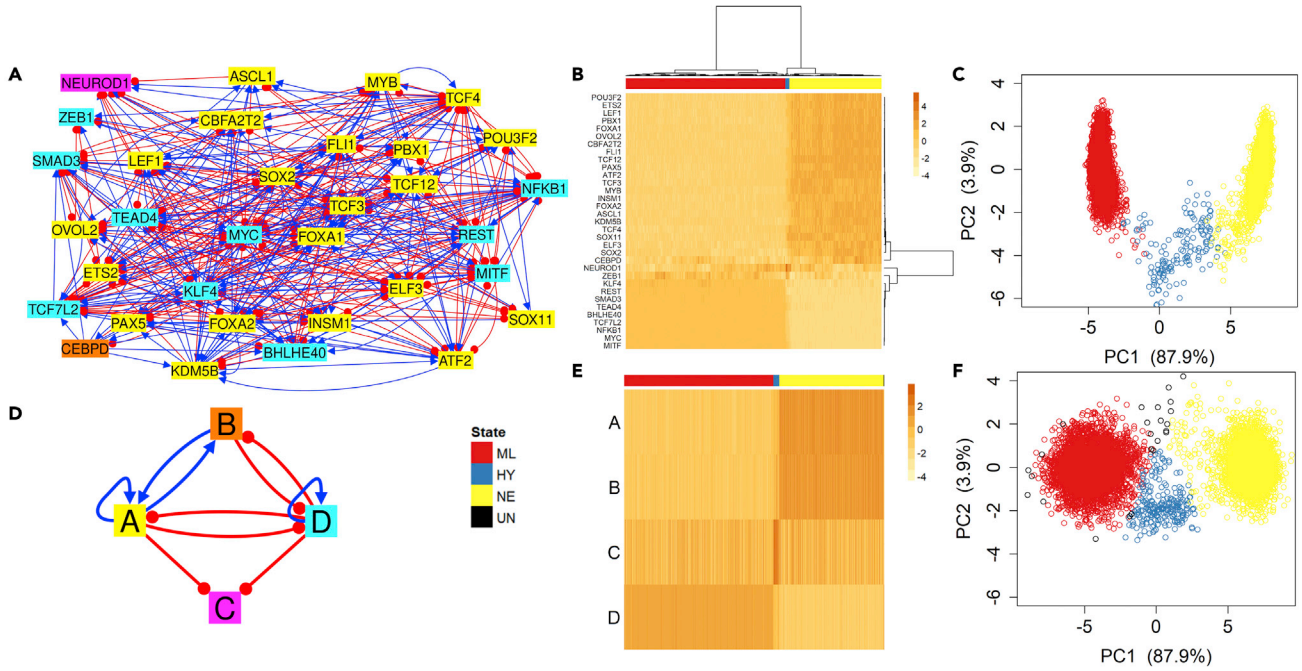


Figure 5. Coarse-graining a gene network involved in small-cell lung cancer (SCLC)

(A) SCLC network topology.

(B) Heatmap of the RACIPE-simulated gene expression profiles from the SCLC network (HCA with Ward.D2 and Pearson correlation). Each row represents a network gene, and each column represents a simulated model. Three model clusters and four gene clusters were determined based on this clustering. Genes belonging to the same gene cluster have the same color in the network diagram shown in panel A.

(C) Scatterplot of the simulated data of the SCLC network projected onto its first two PCs.

(D) Network topology of the best CGC.

(E) Heatmap of the simulated data of the best CGC. Each row represents a CGC node, and each column represents a simulated model.

(F) Scatterplot of the simulated data of the best CGC projected onto the first two PCs of simulated data of the full network. Models corresponding to the same cluster are illustrated with the same color. There is a very small fraction (~0.7%) of models unclassified to any cluster (black points in panels E and F, labeled as UN).

In the optimal CGC, node *A* consists of ASCL1, an SCLC biomarker and a transcription factor involved in neuronal commitment, and some other genes involved in neuronal development and differentiation, such as SOX2, POU3F2, or TCF3.[52–54] Node *B* consists of one single gene CEBPD, which has diverse cellular functions depending on biological contexts. For example, CEBPD can act as a tumor repressor in pancreatic ductal adenocarcinoma[55] or can contribute to proinflammation when activated by IL-6.[56] Node *C* also consists of one single gene NEUROD1, another SCLC biomarker associated with cell migration and metastasis.[57] Node *D* consists of mainly mesenchymal biomarkers, like ZEB1, and oncogenes, like MYC and TCF7L2. The topology of the CGC (Figure 5D) reveals a toggle switch-like topology with an added readout node (node *C*). Nodes *A* and *B* stimulate each other, and both form double-negative feedback loops with node *D*. Note that the nodes like *B* and *C* contain a single gene, which may suggest the importance of these genes in determining the behavior of the GRN. The optimal CGC is slightly larger than a previous reduced model[15] which combines the genes from *A* and *B* into one single node. When we applied SacoGraci to the reduced (three-node) model, we obtained slightly worse results. The best three-node CGC (Figure S7C, yellow cluster has a median PC1 value of 5.61) did not capture the epithelial state in the full network (yellow cluster, with median PC1 value of 7.20 in Figure 5C) as well as the best four-node CGC (Figure 5F, yellow cluster has a median PC1 value of 6.64). Moreover, we modeled CEBPD as a separated node because of its distinct gene expression, as seen in the full GRN (Figure 5B).

Coarse-graining a GRN of gonadal sex determination (GSD) in human

The third example is a GRN established to model the differentiation process of GSD.[58] GSD is an important process during sex development, where bipotential gonadal primordium (BGP) differentiates either into testes or ovaries.[59] Mutations in genes involved in GSD regulation can lead to disorders in sex development.[60] From the simulation of 10,000 RACIPE models of the GSD GRN, we identified six clusters of

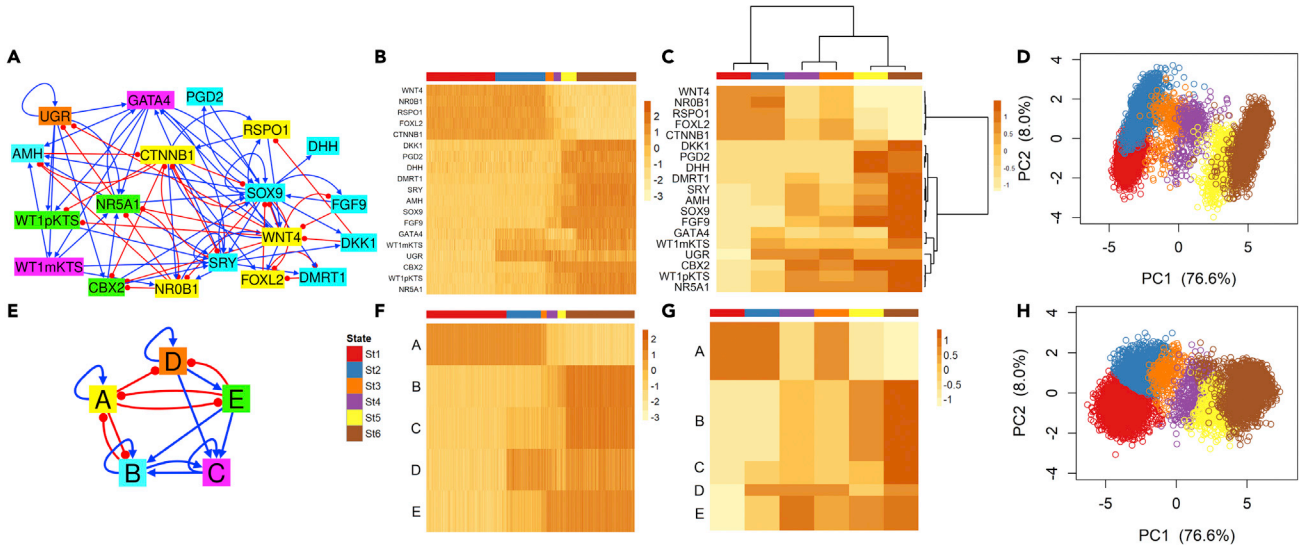


Figure 6. Coarse-graining a gene network in human gonadal sex determination (GSD)

(A) GSD network topology.

(B) The heatmap of the simulated gene expression profiles of the GSD network, where six model clusters were determined.

(C) The heatmap of median values of the simulated models (HCA with Ward.D2 and Pearson correlation). From the HCA, five gene clusters were also identified.

(D) Scatterplot of the simulated data of the GSD network projected onto its first two PCs.

(E) Network topology of the best CGC.

(F) Heatmap of the simulated models of the best CGC.

(G) Heatmap of median expression values of the simulated data of the best CGC.

(H) Scatterplot of the simulated data of the best CGC projected onto the first two PCs of the simulated data of the full network. Models corresponding to the same cluster are illustrated with the same color.

gene expression states (Figures 6B–6D), which can be associated with various cellular states during GSD. In particular, the red and brown clusters correspond to the differentiated granulosa cells and Sertoli cells, respectively. The purple and orange clusters correspond to undifferentiated precursor cells, with the orange one closer to the granulosa lineage and the purple one closer to the Sertoli lineage. The major difference between the differentiated and undifferentiated states is the levels of UGR, the node in the GRN consisting of LHX1, LHX9, EMX2, PAX2, and PAX8 genes and representing the urogenital ridge, an embryonic structure precursor of the gonads. UGR gene expression indicates whether the differentiation process takes place (high UGR levels) or not (low UGR levels).[59] The blue cluster has relatively high UGR, GATA4, and granulosa cell-specific genes, such as WNT4 and FOXL2; thus, it likely corresponds to granulosa cells, but not fully differentiated. The yellow cluster has low levels of UGR and high levels of Sertoli cell-specific genes, likely associated with a genetic disorder due to the low level of GATA4.[61] Note that the RACIPE modeling was able to identify much richer cellular phenotypes (six state clusters) than the previous Boolean network modeling (3 steady states).[58]

Next, from the median gene expression profiles of each model cluster (Figure 6C), we identified five gene clusters from HCA (the grouping scheme illustrated with colors in Figure 6A) for coarse-graining. Figure 6E shows the topology of best last-iteration CGC from SacoGraci. Note that the CGC captures very well all the six states of the original GRN, even for those states with similar gene expression profiles, such as the two undifferentiated states (purple and orange). In the CGC, node *A* consists of granulosa cell-specific genes such as FOXL2, CTNNB1, RSPO1, or WNT4. The other gene in the node, NROB1 (a marker gene of gonadal primordium BGP) is known to act antagonistically to SRY,[62,63] the gene that triggers Sertoli cell differentiation. Network begins differentiation toward ovaries when all the genes in node A are active. When active, CTNNB1 and FOXL2 inhibit Sertoli cell differentiation.[58] Node *B* contains genes specific to Sertoli cells. The Sertoli cell differentiation pathway gets activated by first activating SRY, which activates SOX9 and then the rest of the genes in *B* in an activation cascade. The SOX9 activation is also followed by the repression of the granulosa cell differentiation pathway.[58,64] Node *D* contains one single gene UGR, again confirming its important role in the GRN of GSD. Lastly, the genes from nodes *C* and *E* are specific to non-differentiated BGP.

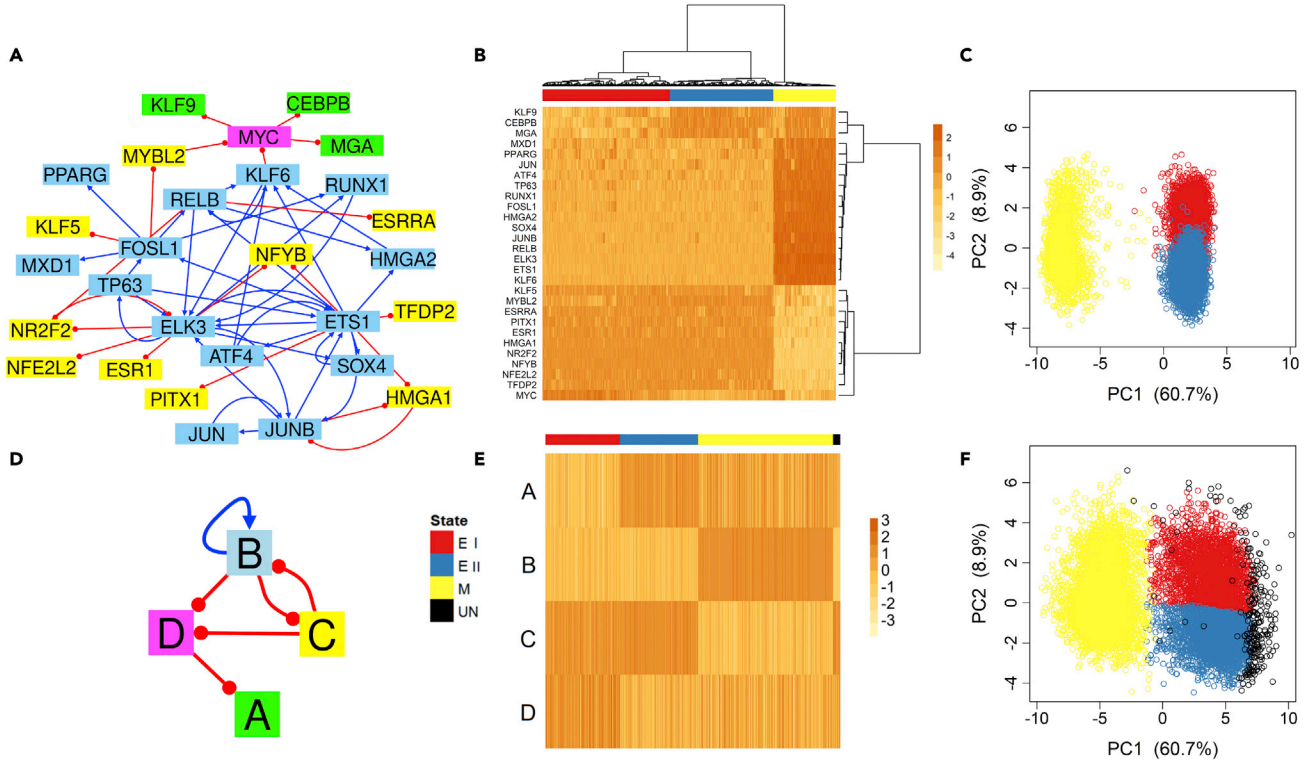


Figure 7. Coarse-graining a network of TGFB1-induced EMT in OVCA420 cancer cell line

(A) OVCA420 network topology.

(B) The heatmap of the simulated gene expression profiles of the network (HCA with Ward.D2 and Pearson correlation). From the HCA, three model clusters and four gene clusters were determined.

(C) Scatterplot of the simulated data of the full network projected onto its first two PCs.

(D) Network topology of the best CGC.

(E) Heatmap of the simulated models of the best CGC.

(F) Scatterplot of the simulated data of the best CGC projected onto the first two PCs of the simulated data of the full network. There is a small fraction (~3%) of models unclassified to any cluster (black points in panels E and F, labeled as UN).

The CGC topology indicates a toggle switch-like behavior with many possible intermediate states. There are two main states of the CGC, one with high levels of node *A* and the other with high levels of nodes *B*, *C*, and *E*. These two states are established and maintained by (1) the toggle switch between nodes *A* and *B*, (2) the toggle switch between nodes *A* and *E*, (3) the activation links between nodes *B*, *C*, and *E*, and (4) node *D* that activates nodes *C* and *E* and is repressed by node *A*. But node *D* is repressed by node *E*, which could explain the existence of the undifferentiated states. Taken together, the optimal CGC not only captures the major cellular states of the GSD network but also sheds light on the logics of gene regulation in the complex system.

Coarse-graining a GRN of TGF-b-induced EMT specific to a cancer cell line

Unlike previous examples where GRNs are literature based, the fourth example is a GRN constructed by a combined bioinformatics and systems biology modeling approach (the diagram of the GRN topology illustrated in Figure 7A)[36] using time-series scRNA-seq data of TGFB1-induced EMT in ovarian cancer cell line OVCA420.[65] From RACIPE simulations, we identified three gene expression clusters that can be associated with the mesenchymal (M) state (yellow), and two epithelial-like states (the first one E I, in red, the second one, E II, in blue) (Figures 7B and 7C). In this case, we found it is sufficient to capture gene expression variations with four gene groups, thus being selected as the grouping scheme for coarse-graining.

Figure 7D shows the topology of best CGC obtained after coarse-graining the network using the MH algorithm as the sampling scheme. The sampling space of possible topologies was very small in this case, so we did not need to apply the more computationally intensive methods, i.e., SA and TE. The CGC topology captures the main features of the full network. In the full network, the genes from node B, JUNB and

ELK3, form toggle switches with the genes from node C, HMGA1 and NR2F2, respectively. This is reflected in the CGC by the toggle switch between nodes C and B. Moreover, MYBL2 from node C and KFL6 from node B repress the MYC gene from node D, which explains the inhibitory edges from C to D and B to D, respectively. These interactions also explain how the E II state is allowed by the CGC.

Performance of coarse-graining of biological GRNs

The optimization process in SacoGraci allows us to generate an ensemble of optimal CGCs, which can be further analyzed to evaluate the robustness of coarse-graining and its relationship with network edge redundancy. To illustrate this utility, from a collection of last-iteration CGCs obtained by SacoGraci, we performed additional analysis to evaluate the consistency of the CGCs and its relationship with network redundancy, as shown in Figure S10. For any possible edge between two nodes in a CGC, we identified the dominant edge type (activation, inhibition, or no edge) with the highest frequency among CGCs. We then compared the percentage of dominant type in the CGCs (representing edge consistency of CGCs, x axis) with the percentage of the dominant type among the edges in the large GRN connecting the corresponding nodes (representing edge redundancy of the large GRN, y axis). Cases were excluded when there is no edge in the large GRN between two gene groups. A low edge consistency and high edge redundancy would indicate the flexibility of the edge choice during circuit optimization and suggest low impact of the edge. While, a high edge consistency and low edge redundancy would indicate a potential correction for errors in the original GRN, such as missing edges or edges with wrong signs. Figure S10 For example, in the case of GSD network, all 23 optimal CGCs have an inhibitory edge from nodes E to A. However, in the full network, there is only one associated inhibitory regulation from NR5A1 to NR0B1 (Figure S10E), suggesting the important role of the E-to-A inhibition to network behavior.

Lastly, we also compared the performance of the three MCMC methods, MH, SA, and TE, in the above-mentioned applications to four biological networks. Figure S8 shows the boxplots of last-iteration scores obtained by these methods for the four networks. Particularly, for the third example where five-node CGCs were optimized (Figure S8C), we observed that on average TE method generates CGCs with lower scores than those from MH and SA, presumably because of larger sampling space allowed by the TE method. When modeling circuits with less than four nodes, however, SacoGraci has similar performance among MH, SA, and TE (Figures S8A, S8B, and S8D). Note that all these biological GRNs contain a high level of redundancy, which explains why the performance of network coarse-graining is consistently high.

DISCUSSION

In this study, we introduced SacoGraci, a new coarse-graining algorithm for gene regulatory networks based on ensemble-based mathematical modeling, dimensionality reduction, and gene circuit optimization. The algorithm requires the topology of a GRN as the only input to produce the CGCs that are best at reproducing the gene expression profiles of the large GRN. We benchmarked the effectiveness and robustness of our method using two small synthetic circuits, each of which was expanded to large networks with different levels of edge perturbation. In both cases, SacoGraci could successfully reproduce the gene expression profiles and recover the topology of the original synthetic circuits, even when the edge-perturbation level was as high as 60%. Furthermore, we successfully applied network coarse-graining to several literature-based and bioinformatics-constructed GRNs. We expect SacoGraci to contribute to a better understanding of the gene regulatory mechanisms of biological networks and the role of regulatory interactions to network dynamics. An ensemble of optimized CGCs also reveals the relationship of the robustness of coarse-graining and network redundancy. We also expect the algorithm to be developed into a generally applicable framework able to coarse grain many other types of biological networks.

From our benchmark tests and biological network applications, we observed the following strengths of our approach. First, in all cases, we could reproduce GRN's expression patterns using CGCs of small sizes. These circuits usually have only three to six nodes, but they are already sufficient in modeling large biological networks with up to six distinct cellular states. Second, the identified CGCs usually can preserve rare populations of cellular states. This is due to the big penalty term in the scoring function that disfavors the CGCs not able to capture all states. Third, the performance of the method was sometimes insensitive to the choice of number of coarse-grained nodes. For instance, in the case of the SCLC network, desired results have been obtained for either three-node circuit or four-node circuit (Figures S7 and 5). Users may wish to rely on biological insights to select the most appropriate size of CGC. Fourth, the method worked well to handle large networks, e.g., those containing as many as 33 genes and 357 edges. The computational cost in general increases exponentially with

the CGC size due to the increase in sampling space of CGC topologies. Restrictions imposed on what topologies can be sampled, e.g., due to edge-type distributions derived from the large GRN (see STAR Methods), can make the computational cost sub-exponential. For the most computationally intensive case we tested, i.e., coarse-graining GSD using five-node circuits, it took 4 h of running time for MH and SA and 5–6 h for TE (for two NVIDIA K80 GPU cards, with 2496 GPU cores per card). Here, the MH and SA methods usually converged after at most 400 iterations, while TE converged after around 200–250 iterations.

Limitations of the study

While SacoGraci proved very effective in finding CGCs that reproduce the behavior of a GRN to a great extent, there could be a few limitations in certain situations. First, in the current approach, we assumed the RACIPE-simulated gene expression profiles form clusters of spherical shape, which we use to approximate the model assignment to each gene expression state. While the current scoring function works very well in our test cases, this may generate assignment errors in some rare cases, where the spherical assumption is not satisfied. One potential solution is to define a new distance function between gene expression distribution of a GRN and that of a CGC using metrics, such as Kullback-Leibler divergence, Kolmogorov-Smirnov statistic test, or earth mover's distance. Second, we chose the number of CG nodes based on the number of gene clusters that can reproduce the gene expression patterns of various model clusters. But there is no guarantee that, for instance, using four-node CGCs would get us much better results than using three-node CGCs. It would be an interesting question to identify the optimal number of coarse-grained nodes through optimization. Third, the quality of network coarse-graining and computational cost might strongly depend on the choice of the sampling method. Here, we used three MCMC methods to sample candidate CGCs. The most sophisticated method we applied, the parallel tempering algorithm, produced the best overall results, but was usually slowest for large GRNs with large sampling space. Some other sampling algorithms might further improve the combined speed and accuracy, such as genetic algorithms or some other heuristic methods, especially for cases where the number of coarse-grained nodes is more than five. For CGCs with less than five nodes, the simulated annealing method would suffice to perform well within a short amount of time. Lastly, the current scoring function was designed to match steady-state gene expression distributions, but not necessarily gene expression dynamics. It would be helpful to incorporate the information of gene expression time dynamics during the circuit optimization.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Simulating GRNs with RACIPE
  - Model clustering and gene grouping
  - Sampling coarse-grained circuits
  - Scoring function for circuit optimization
  - Detailed implementation of the MCMC sampling methods
  - Expanding a small circuit and perturbing a full network

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2023.105927.

ACKNOWLEDGMENTS

This work is supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R35GM128717, and by startup funds from Northeastern University.

AUTHOR CONTRIBUTIONS

Conceptualization, M.L.; Methodology, C.C. and M.L.; Software, C.C. and M.L.; Investigation, C.C.; Formal Analysis, C.C.; Writing-Original Draft, C.C. and M.L.; Writing-Review & Editing, C.C. and M.L.; Funding Acquisition, M.L.; Supervision, M.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

INCLUSION AND DIVERSITY

One or more of the authors of this paper received support from a program designed to increase minority representation in their field of research.

Received: August 9, 2022
Revised: December 19, 2022
Accepted: January 3, 2023
Published: February 17, 2023

REFERENCES

1. Gerstein, M.B., Kundaje, A., Hariharan, M., Landt, S.G., Yan, K.K., Cheng, C., Mu, X.J., Khurana, E., Rozowsky, J., Alexander, R., et al. (2012). Architecture of the human regulatory network derived from ENCODE data. Nature *489*, 91–100.

2. Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein-protein interactions in yeast. Nat. Biotechnol. *18*, 1257–1261.

3. Shahzad, K., and Loor, J.J. (2012). Application of top-down and bottom-up systems approaches in ruminant physiology and metabolism. Curr. Genom. *13*, 379–394.

4. Katebi, A., Ramirez, D., and Lu, M. (2021). Computational systems-biology approaches for modeling gene networks driving epithelial–mesenchymal transitions. Comput. Syst. Oncol. *1*, e1021.

5. Kulkarni, V.V., Arastoo, R., Bhat, A., Subramanian, K., Kothare, M.V., and Riedel, M.C. (2012). Gene regulatory network modeling using literature curated and high throughput data. Syst. Synth. Biol. *6*, 69–77.

6. Thiele, I., and Palsson, B.Ø. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat. Protoc. *5*, 93–121.

7. Feist, A.M., Herrgård, M.J., Thiele, I., Reed, J.L., and Palsson, B.Ø. (2009). Reconstruction of biochemical networks in microorganisms. Nat. Rev. Microbiol. *7*, 129–143.

8. Pratapa, A., Jalihal, A.P., Law, J.N., Bharadwaj, A., and Murali, T.M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nat. Methods *17*, 147–154.

9. Deshpande, A., Chu, L.-F., Stewart, R., and Gitter, A. (2022). Network inference with Granger causality ensembles on single-cell transcriptomics. Cell Rep. *38*, 110333.

10. Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., and Califano, A. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinf. *7*, S7.

11. Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M.S.H., Ko, S.B.H., Gouda, N., Hayashi, T., and Nikaido, I. (2017). SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. Bioinformatics *33*, 2314–2321.

12. Pranzatelli, T.J.F., Michael, D.G., and Chiorini, J.A. (2018). Optimized ATAC-seq and DNase1-seq pipelines for rapid and accurate genome regulatory network inference. BMC Genom. *19*, 563.

13. Aibar, S., González-Blas, C.B., Moerman, T., Huynh-Thu, V.A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.C., Geurts, P., Aerts, J., et al. (2017). SCENIC: single-cell regulatory network inference and clustering. Nat. Methods *14*, 1083–1086.

14. Tripathi, S., Kessler, D.A., and Levine, H. (2022). Minimal frustration underlies the usefulness of incomplete and inexact regulatory network models in biology. Preprint at bioRxiv. https://doi.org/10.1101/2022.06.07.495167.

15. Chauhan, L., Ram, U., Hari, K., and Jolly, M.K. (2021). Topological signatures in regulatory network enable phenotypic heterogeneity in small cell lung cancer. Elife *10*, e64522.

16. Lu, M., Jolly, M.K., Levine, H., Onuchic, J.N., and Ben-Jacob, E. (2013). MicroRNA-based regulation of epithelial–hybrid–mesenchymal fate determination. Proc. Natl. Acad. Sci. USA *110*, 18144–18149.

17. Atilgan, A.R., Durell, S.R., Jernigan, R.L., Demirel, M.C., Keskin, O., and Bahar, I. (2001). Anisotropy of fluctuation dynamics of proteins with an elastic network model. Biophys. J. *80*, 505–515.

18. Ingólfsson, H.I., Lopez, C.A., Uusitalo, J.J., de Jong, D.H., Gopal, S.M., Periole, X., and Marrink, S.J. (2014). The power of coarse graining in biomolecular simulations. Wiley Interdiscip. Rev. Comput. Mol. Sci. *4*, 225–248.

19. Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A.E., and Kolinski, A. (2016). Coarse-grained protein models and their applications. Chem. Rev. *116*, 7898–7936.

20. Lu, M., and Ma, J. (2005). The role of shape in determining molecular motions. Biophys. J. *89*, 2395–2401.

21. Lu, M., Poon, B., and Ma, J. (2006). A new method for coarse-grained elastic normal-mode analysis. J. Chem. Theor. Comput. *2*, 464–471.

22. VanWart, A.T., Eargle, J., Luthey-Schulten, Z., and Amaro, R.E. (2012). Exploring residue component contributions to dynamical network models of allostery. J. Chem. Theor. Comput. *8*, 2949–2961.

23. Snowden, T.J., van der Graaf, P.H., and Tindall, M.J. (2017). Methods of model reduction for large-scale biological systems: a survey of current methods and trends. Bull. Math. Biol. *79*, 1449–1486.

24. Erban, R., Kevrekidis, I.G., Adalsteinsson, D., and Elston, T.C. (2006). Gene regulatory networks: a coarse-grained, equation-free approach to multiscale computation. J. Chem. Phys. *124*, 084106.

25. Sinitsyn, N.A., Hengartner, N., and Nemenman, I. (2009). Adiabatic coarse-graining and simulations of stochastic biochemical networks. Proc. Natl. Acad. Sci. USA *106*, 10546–10551.

26. Prescott, T.P., and Papachristodoulou, A. (2014). Layered decomposition for the model order reduction of timescale separated biochemical reaction networks. J. Theor. Biol. *356*, 113–122.

27. Danø, S., Madsen, M.F., Schmidt, H., and Cedersund, G. (2006). Reduction of a biochemical model with preservation of its basic dynamic properties. FEBS J. *273*, 4862–4877.

28. Maurya, M.R., Bornheimer, S.J., Venkatasubramanian, V., and Subramaniam, S. (2009). Mixed-integer nonlinear optimisation approach to coarse-graining biochemical networks. IET Syst. Biol. *3*, 24–39.

29. Dokoumetzidis, A., and Aarons, L. (2009). Proper lumping in systems biology models. IET Syst. Biol. *3*, 40–51.

30. Meyer-Bäse, A., and Theis, F. (2008). Gene regulatory networks simplified by nonlinear balanced truncation. In Independent Component Analyses, Wavelets, Unsupervised Nano-Biomimetic Sensors, and Neural Networks VI, *vol. 6979*Independent Component Analyses, Wavelets,

Unsupervised Nano-Biomimetic Sensors, and Neural Networks VI (SPIE), pp. 103–110.

31. Huang, B., Lu, M., Jia, D., Ben-Jacob, E., Levine, H., and Onuchic, J.N. (2017). Interrogating the topological robustness of gene regulatory circuits by randomization. PLoS Comput. Biol. *13*, e1005456.

32. Kohar, V., and Lu, M. (2018). Role of noise and parametric variation in the dynamics of gene regulatory circuits. NPJ Syst. Biol. Appl. *4*, 40.

33. Huang, B., Lu, M., Galbraith, M., Levine, H., Onuchic, J.N., and Jia, D. (2020). Decoding the mechanisms underlying cell-fate decision-making during stem cell differentiation by random circuit perturbation. J. R. Soc. Interface *17*, 20200500.

34. Katebi, A., Kohar, V., and Lu, M. (2020). Random parametric perturbations of gene regulatory circuit uncover state transitions in cell cycle. iScience *23*, 101150.

35. Huang, B., Jia, D., Feng, J., Levine, H., Onuchic, J.N., and Lu, M. (2018). RACIPE: a computational tool for modeling gene regulatory circuits using randomization. BMC Syst. Biol. *12*, 74.

36. Ramirez, D., Kohar, V., and Lu, M. (2020). Toward modeling context-specific EMT regulatory networks using temporal single cell RNA-seq data. Front. Mol. Biosci. *7*, 54.

37. Su, K., Katebi, A., Kohar, V., Clauss, B., Gordin, D., Qin, Z.S., Karuturi, R.K.M., Li, S., and Lu, M. (2022). NetAct: a computational platform to construct core transcription factor regulatory networks using gene activity. Preprint at BioRxiv. https://doi.org/10.1101/2022.05.06.487898.

38. Landau, D.P., and Binder, K. (2009). A Guide to Monte Carlo Simulations in Statistical Physics (Cambridge University Press).

39. Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. *58*, 236–244.

40. Elowitz, M.B., and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. Nature *403*, 335–338.

41. Nieto, M.A., Huang, R.Y.-J., Jackson, R.A., and Thiery, J.P. (2016). J. P. Emt: 2016. Cell *166*, 21–45.

42. Gupta, G.P., and Massagué, J. (2006). Cancer metastasis: building a framework. Cell *127*, 679–695.

43. Thiery, J.P., Acloque, H., Huang, R.Y.J., and Nieto, M.A. (2009). Epithelial-mesenchymal transitions in development and disease. Cell *139*, 871–890.

44. Krämer, A., Green, J., Pollard, J., and Tugendreich, S. (2014). Causal analysis approaches in ingenuity pathway analysis. Bioinformatics *30*, 523–530.

45. Zhang, J., Tian, X.J., Zhang, H., Teng, Y., Li, R., Bai, F., Elankumaran, S., and Xing, J. (2014). TGF-β–induced epithelial-to-mesenchymal transition proceeds through stepwise activation of multiple feedback loops. Sci. Signal. *7*, ra91.

46. Karacosta, L.G., Anchang, B., Ignatiadis, N., Kimmey, S.C., Benson, J.A., Shrager, J.B., Tibshirani, R., Bendall, S.C., and Plevritis, S.K. (2019). Mapping lung cancer epithelial-mesenchymal transition states and trajectories with single-cell resolution. Nat. Commun. *10*, 5587.

47. Qiu, X., Zhang, Y., Martin-Rufino, J.D., Weng, C., Hosseinzadeh, S., Yang, D., Pogson, A.N., Hein, M.Y., Hoi Joseph Min, K., Wang, L., et al. (2022). Mapping transcriptomic vector fields of single cells. Cell *185*, 690–711.e45.

48. Zhang, J., and Ma, L. (2012). MicroRNA control of epithelial–mesenchymal transition and metastasis. Cancer Metastasis Rev. *31*, 653–662.

49. Tian, X.-J., Zhang, H., and Xing, J. (2013). Coupled reversible and irreversible bistable switches underlying TGFβ-induced epithelial to mesenchymal transition. Biophys. J. *105*, 1079–1089.

50. Jia, D., Jolly, M.K., Tripathi, S.C., Den Hollander, P., Huang, B., Lu, M., Celiktas, M., Ramirez-Peña, E., Ben-Jacob, E., Onuchic, J.N., et al. (2017). Distinguishing mechanisms underlying EMT tristability. Cancer Converg. *1*, 2.

51. Udyavar, A.R., Wooten, D.J., Hoeksema, M., Bansal, M., Califano, A., Estrada, L., Schnell, S., Irish, J.M., Massion, P.P., and Quaranta, V. (2017). Novel hybrid phenotype revealed in small cell lung cancer by a transcription factor network model that can explain tumor heterogeneity. Cancer Res. *77*, 1063–1074.

52. Graham, V., Khudyakov, J., Ellis, P., and Pevny, L. (2003). SOX2 functions to maintain neural progenitor identity. Neuron *39*, 749–765.

53. Gribble, S.L., Kim, H.-S., Bonner, J., Wang, X., and Dorsky, R.I. (2009). Tcf3 inhibits spinal cord neurogenesis by regulating sox4aexpression. Development *136*, 781–789.

54. Kasher, P.R., Schertz, K.E., Thomas, M., Jackson, A., Annunziata, S., Ballesta-Martinez, M.J., Campeau, P.M., Clayton, P.E., Eaton, J.L., Granata, T., et al. (2016). 6q16.1 deletions encompassing POU3F2 cause susceptibility to obesity and variable developmental delay with intellectual disability. Am. J. Hum. Genet. *98*, 363–372.

55. Hartl, L., Duitman, J., Aberson, H.L., Chen, K., Dijk, F., Roelofs, J.J.T.H., Dings, M.P.G., Hooijer, G.K.J., Hernanda, P.Y., Pan, Q., et al. (2020). CCAAT/enhancer-binding protein delta (C/EBPδ): a previously unrecognized tumor suppressor that limits the oncogenic potential of pancreatic ductal adenocarcinoma cells. Cancers *12*, 2546.

56. Cantwell, C.A., Sterneck, E., and Johnson, P.F. (1998). Interleukin-6-Specific activation of the C/EBPδ gene in hepatocytes is mediated by Stat3 and Sp1. Mol. Cell Biol. *18*, 2108–2117.

57. Ikematsu, Y., Tanaka, K., Toyokawa, G., Ijichi, K., Ando, N., Yoneshima, Y., Iwama, E., Inoue, H., Tagawa, T., Nakanishi, Y., and Okamoto, I. (2020). NEUROD1 is highly expressed in extensive-disease small cell lung cancer and promotes tumor cell migration. Lung Cancer *146*, 97–104.

58. Ríos, O., Frias, S., Rodríguez, A., Kofman, S., Merchant, H., Torres, L., and Mendoza, L. (2015). A Boolean network model of human gonadal sex determination. Theor. Biol. Med. Model. *12*, 26.

59. Yang, Y., Workman, S., and Wilson, M.J. (2019). The molecular pathways underlying early gonadal development. J. Mol. Endocrinol. *62*, R47–R64.

60. Ohnesorg, T., Vilain, E., and Sinclair, A.H. (2014). The genetics of disorders of sex development in humans. Sex Dev. *8*, 262–272.

61. Lourenço, D., Brauner, R., Rybczynska, M., Nihoul-Fékété, C., McElreavey, K., and Bashamboo, A. (2011). Loss-of-function mutation in GATA4 causes anomalies of human testicular development. Proc. Natl. Acad. Sci. USA *108*, 1597–1602.

62. Lalli, E., and Sassone-Corsi, P. (2003). DAX-1, an unusual orphan receptor at the crossroads of steroidogenic function and sexual differentiation. Mol. Endocrinol. *17*, 1445–1453.

63. Swain, A., Zanaria, E., Hacker, A., Lovell-Badge, R., and Camerino, G. (1996). Mouse Dax1 expression is consistent with a role in sex determination as well as in adrenal and hypothalamus function. Nat. Genet. *12*, 404–409.

64. Koopman, P. (1999). Sry and Sox9: mammalian testis-determining genes. Cell. Mol. Life Sci. *55*, 839–856.

65. Cook, D.P., and Vanderhyden, B.C. (2020). Context specificity of the EMT transcriptional response. Nat. Commun. *11*, 2142.

66. Liu, J.S. (2008). Monte Carlo Strategies in Scientific Computing (Springer).

67. Hamze, F., Dickson, N., and Karimi, K. (2010). Robust parameter selection for parallel tempering. Int. J. Mod. Phys. C *21*, 603–615.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
SacoGraci (R package)	This paper	https://github.com/lusystemsbio/SacoGraci
sRACIPE (R package)	(Kohar and Lu, 2018)	https://www.bioconductor.org/packages/release/bioc/html/sRACIPE.html

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Mingyang Lu (m.lu@northeastern.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- The SacoGraci R package, tutorial script, and all other data have been deposited on our GitHub repository https://github.com/lusystemsbio/SacoGraci, and is publicly available as of the date of publication.
- Additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Simulating GRNs with RACIPE

We applied an ordinary differential equation (ODE)-based method, namely random circuit perturbation RACIPE, implemented as an R package sRACIPE[32], to simulate the stable steady states of a GRN. RACIPE takes a network topology as the input and builds a system of ODEs that describes the network's dynamics. By randomly sampling the kinetic parameters of the ODE system from biologically relevant ranges, RACIPE builds an ensemble of models, each of which as an instantiation of ODEs where all kinetic parameters and initial conditions are randomly specified. These kinetic parameters reflect different characteristics of the network, such as production and degradation rates for genes and parameters from shifted Hill function for regulatory interactions. See the original RACIPE study[31] for details of the methodology. Each model is simulated by an ODE solver until convergence to a stable steady state. Finally, we obtained a set of gene expression profiles from the analysis of an ensemble of models. A total of 10000 RACIPE models were typically generated to ensure we captured robust features of the GRN. The steady state gene expressions from these models usually form robust clusters, which can be further associated with distinct cellular states. Thus, we can predict the potential cellular states only from the network topology.

Model clustering and gene grouping

From the RACIPE-simulated gene expression profiles of the large GRN, we aim to determine the distinct patterns of gene expression and distinct groups of genes with shared dynamical behavior in the network. To achieve this, SacoGraci performs hierarchical clustering analysis (HCA) to determine the number of model and gene clusters by default. The other method we used for model clustering was k-means clustering. However, users can also provide their own choices of the grouping scheme. For the HCA, we typically use a distance function of one minus Pearson correlation between the gene expression profiles of two models and the Ward's D2 [39] minimum variance method as the linkage method. The only exception was for the case of the synthetic repressilator circuit with activation loops, where we used the older Ward's D method.

To evaluate the quality of the grouping scheme, we also recommend to visually examine the heatmap of the clustered models and the scatterplot of gene expression profiles with low dimensional projection, using methods such as principal component analysis (PCA). With a high-quality grouping scheme, the clusters obtained from hierarchical clustering should form dense, compact scatterplot regions. Clusters that have similar expression patterns and have very close or overlapping scatterplot regions might be combined. For gene grouping, users can explore different number of gene clusters and/or incorporate prior knowledge (such as the number of gene families involved in the biological context) into the pipeline. Each of the gene group from the clustering analysis will be a node of the CGC. In our test cases we chose different number of gene clusters (e.g., from three to five) to explore CGCs of different sizes.

Sampling coarse-grained circuits

In the previous step, SacoGraci defines the CG nodes from the clustering analysis. Now, we will determine how the edges of CGCs are sampled. For each ordered pair (A, B) of gene clusters, we determine the probability distribution of the edge type (activation, inhibition, no edge) from A to B. To do so, we examine all possible edges from genes in A to genes in B from the full network and calculate the proportion of activation and inhibitory edges. This will give us the probability of an activation edge from A to B and probability of an inhibitory edge from A to B, respectively. The remaining fraction determines the probability of having no edge from A to B.

As mentioned above, the nodes of a CGC correspond to gene groups of the full network. Then, to build a CGC we sample edges for all ordered pairs of nodes. The edge types are sampled according to the above calculated edge type distributions. The only restriction we impose here is to get a connected topology with no input nodes. RACIPE-simulated gene expression of an input gene/node looks like noise because nothing regulates the expression of that input gene. That is why, whenever a large network has input nodes, we delete them and apply SacoGraci to the remaining network. To obtain optimal CGCs, we defined the scoring function (see next section) and utilized three Markov Chain Monte Carlo (MCMC) sampling methods, i.e., Metropolis-Hastings (MH), Simulated Annealing (SA) and Parallel Tempering (TE)[66] (see the section after next).

Scoring function for circuit optimization

A scoring function is defined to compare the simulated gene expression profiles of a CGC with those of a full network. For each CGC, we generate 10000 models with randomly selected kinetic parameters using RACIPE. Each model is simulated by an ODE solver until convergence to a stable steady state, which is represented by a vector of gene expression values: $x_i = (g_i^1, g_i^2, \ldots, g_i^N)$, where N is the number of genes in the network and g_i^j is the expression value of j^{th} gene in the i^{th} model.

For each model cluster C of the full network, we define its center, O_C, as the median of the gene expression vectors x_i for every model i in the cluster C. We prefer medians to other possible choices, such as centroids, as the median of a group of values is more robust to outliers than other statistics. Next, we permute all 10,000 RACIPE-simulated gene expression vectors. The permuted gene expression vectors take the form

$$x_{i,perm} = \left(g_{i_1}^1, g_{i_2}^2, \ldots, g_{i_N}^N\right), \qquad \text{(Equation 1)}$$

where $i_1, i_2, \ldots, i_N$ are values sampled from the set $\{1, 2, \ldots, M\}$, and the total number of models is $M = 10000$. Then, we define the radius of model cluster C, r_C, as the $[\alpha * M]^{th}$ shortest Euclidian distance from any of the permuted gene expression vectors to the center O_C. Here [] is the integer part function and α is a parameter selected from the set $\{0.01, 0.05, 0.10, 0.15, 0.20\}$ to control the cluster radius r_C. Parameter α varies from network to network but is the same for all model clusters corresponding to a given network. We also assume a spherical form of gene expression profiles for each model cluster. If r_C is too small, many of the models initially belonging to cluster C could be left outside of the sphere of radius r_C. If r_C is too big, the sphere could include models with more than one gene expression pattern. In general, it is recommended to choose a rather small r_t. For a given network, to choose α, we start with $\alpha = 0.01$ and find the radius of each cluster. We check how many models were left outside of the spheres. We choose the next value of α if the number of left out models is bigger than 3% of M. The maximum possible value of α is 0.20.

With O_C and r_C computed for every model cluster of the full network, we then define the score of a given CGC as follows. First, an ensemble of M models are simulated with RACIPE for the CGC. Second, the gene expression vectors of the CGC are expanded to the size of the full network by assigning the gene expression value of a CG node A to all the genes belonging to A in the full network. For instance, for a full network of 7 genes and a CGC containing 3 nodes A,B,C, if A is a group of genes $A = \{g^1, g^2, g^3\}$ and $B = \{g^4, g^5\}$ and $C = \{g^6, g^7\}$, then a gene expression vector of a model of CGC, $x_i^{CGC} = (g_i^A, g_i^B, g_i^C)$ can be expanded to a vector of the full network $x_i^{exp} = (g_i^A, g_i^A, g_i^A, g_i^B, g_i^B, g_i^C, g_i^C)$. Third, for each expanded vector x_i^{exp} obtained in the previous step, we assign a scoring term E_i as the minimum ratio of the Euclidian distance from x_i^{exp} to the cluster center O_C and the cluster radius r_C:

$$E_i = \min_C \left\{ \frac{d\left(x_i^{exp}, O_C\right)}{r_C} \right\}. \quad \text{(Equation 2)}$$

When and only when E_i is less than 1, we assign the model i to the cluster that realizes the minimum:

$$C_i = \operatorname*{argmin}_C \left\{ \frac{d\left(x_i^{exp}, O_C\right)}{r_C} \right\}. \quad \text{(Equation 3)}$$

Sometimes, the ratio is less than 1 for multiple model clusters from the full network, indicating the expanded CG model is within the ranges of multiple model clusters. But according to Equation (3), the cluster with the minimum ratio is assigned. Eventually, each expanded CG model has an assigned score, and many of them are also assigned to the model clusters of the full network. For each model cluster C, we calculate the median score

$$E_C = \operatorname*{median}_{C_i = C}(E_i). \quad \text{(Equation 4)}$$

Here, E_C reflects how well models from a CGC represent the gene expression patterns of a given cluster C of the full network. Fourth, we define the total score as

$$E_{tot} = \frac{M}{n_C} \sum_C E_C + \sum_{E_i > 1} E_i + E_p, \quad \text{(Equation 5)}$$

where n_C is the number of model clusters of the full network. The second term in Equation (5) gives penalties to the expanded CG models not assigned to any cluster. In addition, if, for a certain cluster C, the number of assigned CG models is less than half of the full-network models in the same cluster C, then a penalty term of 50000 (E_p) is added to E_{tot}.

Detailed implementation of the MCMC sampling methods

We used three Markov Chain Monte Carlo (MCMC) sampling methods to obtain optimal CGCs: Metropolis-Hastings (MH), Simulated Annealing (SA) and Parallel Tempering (TE).[66] We assume the probability of a CGC having the score E_{tot} is given by a Boltzmann distribution

$$p(E_{tot}, T) \propto e^{-\frac{E_{tot}}{kT}}, \quad \text{(Equation 6)}$$

where T is the temperature, and k is the Boltzmann's constant (set to 1 here).

Since E_{tot} only depends on the topology of the CGC, we use MCMC methods to sample different circuit topologies as follows. During every iteration, we only change an edge of the current CGC. An edge is picked at random from the current circuit and then we sample its type according to its edge type distribution determined by the topology of the full network.

Since the scoring function is based on RACIPE simulations of 10000 models, a new calculation of the score for the same circuit will produce a slightly different result. To improve the efficiency and robustness of the circuit optimization, we chose to calculate the score five times and take the average of the five scores as the final score, when we sample a CGC for the first time. If the circuit is sampled at a later iteration, its score is not calculated again.

For the synthetic circuit cases we chose ten different perturbed networks for each perturbation level (see section "Expanding a small circuit and perturbing a full network"). For each perturbed network, we run MH and SA twice and TE once as follows: 1) We chose two different coarse-grained topologies as the starting circuits for both MH and SA. 2) The same two circuits were also among the initial circuits of the TE method.

The other initial circuits for the TE method were randomly generated according to the edge type distributions determined by the perturbed network. Total number of initial topologies used for TE was 24. 3) We run MH, SA and TE using the initial circuits determined above and the edge type distributions given by the perturbed network. All three MCMC methods were run for 1400 iterations.

To coarse grain the EMT, SCLC and GSD networks we applied a similar simulation scheme. In each case we generated 20 CGCs that were used as starting circuits for both MH and SA. They were chosen in such a way that the number of edge mismatches between them were as big as possible. That was to ensure we cover different regions of the sampling space. The 20 initial circuits were also a part of the initial circuits of the TE simulations. Initial number of topologies for TE method was 24. Both MH and SA were run in parallel using 20 threads, each thread running a simulation. The whole TE procedure was run 10 times for each of these networks. We did not run SA and TE when we coarse-grained the OVAL420 network, as there were only a few CGC topologies to sample in this case. All three MCMC methods were run for 1400 iterations.

Before starting the MCMC methods, we run MH for 50 iterations. The purpose is to evaluate the magnitude of the scores of the CGCs. Knowing this we can decide on the temperature value we need to use for running MH and the temperature grids for SA and TE. We denote by E_{MH} the last score we get from the preliminary MH run. It is recommended that the temperature value for MH, T_{MH} satisfies the condition

$$60 < \frac{E_{MH}}{T_{MH}} < 75. \qquad \text{(Equation 7)}$$

This rationale of the criterion in Equation (7) can be understood as follows. If the score of the current circuit is E_{cur}, then the acceptance probability of a newly sampled circuit with score E_{new} is $e^{\frac{E_{cur}-E_{new}}{T_{MH}}}$. If we assume that E_{cur} has the magnitude of E_{EM}, then, when the score is increased by 3%, the acceptance probability is larger than 10% for T_{MH} satisfying the above constraint. This will ensure a sufficient acceptance of newly sampled circuits with slightly higher scores than the current one. In this study, for the coupled toggle switches example, we chose $T_{MH} = 40$; for the other examples, we chose $T_{MH} = 60$.

For SA method we developed a temperature schedule consisting of 14 decreasing values, with the method running 100 iterations for each temperature value. The first four temperature values ensured we did a deep exploration of the sampling space. Next three values were within the range around T_{MH}. As for reduction rule of the temperature, we chose two geometric rates, one for the first seven values and one for the last seven values. For the coupled toggle switches example, the starting temperature was 120, first decreasing rate was 0.85, and the second decreasing rate was 0.6. For the other examples, the starting temperature was 150, the first decreasing rate was 0.8, and the second decreasing rate was 0.6.

For TE method we created K = 24 replicas of the original Monte Carlo Markov chain. Each replica has its own temperature and is engaged in its own Markov Chain Monte Carlo search for CGCs using MH. A temperature grid $T_1 < T_2 < \ldots < T_K$ is constructed with $T_1 = 1$ corresponding to our target replica (our solution to the problem). The largest temperature was set to $T_K \cong \frac{E_{MH}}{30}$. That gives us a big chance of having an acceptance rate of at least 50% for replica with temperature T_K. Each replica can swap its state with one of the replicas of the neighboring temperature. The acceptance probability of swaps between replicas with temperatures T_i and T_j is

$$\rho_{i,j} = min\left\{1, \frac{p(E_{tot,j}, T_i)p(E_{tot,i}, T_j)}{p(E_{tot,i}, T_i)p(E_{tot,j}, T_j)}\right\}, \qquad \text{(Equation 8)}$$

where $p(E_{tot,i}, T_i)$ is the Boltzmann distribution with temperature T_i and $E_{tot,i}$ is the score of the current circuit sampled by replica with temperature T_i.

To ensure a target swap rate for neighboring replicas around 0.4, we added new temperatures to the grid according to a robust feedback-optimized method.[67] First, we performed an update of sampled circuit topology for each replica using MH. Second, we proposed swaps between neighboring replicas L and L-1, L-1 and L-2, …,2 and 1, where L is the number of replicas. Third, we repeated the first and second steps *m* times. Fourth, for each pair of neighboring replicas (i, i-1), the following quantity was calculated:

$$\mathbb{Q}_{i,i-1} = \frac{1}{N_{swap}^{i,i-1}} \sum_{l=1}^{N_{swap}^{(i,i-1)}} ln\left(\rho_{i,i-1}^{l}\right), \qquad \text{(Equation 9)}$$

$N_{swap}^{(i,i-1)} = m$ is the number of proposed swaps between replicas i and i-1, and $\rho_{i,i-1}^{l}$ is the acceptance probability for swapping i and i-1 at the l^{th} attempt. If $R_{i,i-1} = \left[\sqrt{\frac{\mathbb{Q}_{i,i-1}}{\ln(0.4)}}\right] > 0$, then we add to the grid $R_{i,i-1}$ temperatures, evenly spaced between T_{i-1} and T_i. This temperature grid addition process was performed three times to make sure there were enough temperatures on the grid to prevent isolation of replicas. The values of m for the three runs of the procedure were 50,100 and 150, respectively. After the temperature addition process, the first and second steps described above were performed for another 1100 times. The initial temperature grid for the coupled toggle switches example was {1, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 6, 9, 11, 13, 20, 28, 40, 55, 70, 85, 100}. For the other examples the initial grid was {1, 1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 6, 9, 11, 13, 20, 28, 40, 55, 70, 90, 120}.

We found the optimization simulations usually converged (no more than 5% changes in scores for 100 consecutive iterations) at the end of the 1400-interation runs, as illustrated in Figure S9. For the most computationally intensive case we tested, i.e., coarse-graining GSD using five-node circuits and benchmarking the synthetic circuit with six nodes, it took 4 hours of running time for MH and SA and 5-6 hours for TE (using NVIDIA K80 GPU cards, with 2496 GPU cores per card) to converge. For the other cases it took about 3 hours for MH and SA to converge and 4 hours at most for TE. The only exception was for the EMT case where some MH and SA simulations showed high variation for many iterations (Figure S9C). It took about 8-9 hours for these simulations to converge. The above times are for running simulations using parallel computing (40 computing threads). Note that, in one such parallel job, we were able to run 20 MH simulations or 20 SA simulations, but just one TE simulation within the timeframe described above. In the case of limited computing resources, we recommend using simulated annealing with different initial circuits, while employing the above-mentioned convergence criterion.

Expanding a small circuit and perturbing a full network

For benchmarking the MCMC sampling methods to obtain optimal CGCs, we started with a small gene circuit (the ground truth) and expanded it to a large network that reproduces the similar gene expression patterns of the circuit. To achieve this, we replaced each node in the circuit by three or four genes. We connected the genes corresponding to the same node by many activation links. That ensures that these genes will have a similar gene expression pattern. For two nodes connected by a directed edge in the small circuit, we connected the corresponding genes by 3 to 6 edges of the same type going in the same direction. Here, we created multiple copies of the original circuit (four copies for the coupled toggle-switches, three for the repressilator) and connected them with edges consistent with edges of the original circuit.

To evaluate the robustness of the coarse graining algorithm when some inconsistent interactions are presented in the expanded network, we generated networks with perturbed topologies. To perturb a network by a certain degree, we changed its topology as follows. For a certain perturbation level a and a network of h edges, we would need to make a total of ah edge changes, each of which is either change of the sign (activation or inhibition) of an edge (50% chance), adding an edge (25% chance), or deleting an edge (25% chance). We required that the resulting perturbed network to be connected and have no input nodes. The perturbation levels a were chosen from {12%,20%,30%,40%,50%,60%}. To ensure a good coverage of the sampling space, for each perturbation level we first randomly generated 20 perturbed topologies. Then, for each one of them we calculated average number of edge mismatches with the other 19 topologies. We chose the top ten topologies with the most mismatches as our perturbed networks. This ensures that the chosen perturbed networks are most different from one another.