Revealing the impact of genomic alterations on cancer cell signaling with a partially transparent deep learning model

Jonathan D. Young^{1*}, Xinghua Lu^{1,2*},

1 Intelligent Systems Program, School of Computing and Information, University of Pittsburgh, Pittsburgh, Pennsylvania, United States of America

2 Department of Biomedical Informatics, School of Medicine, University of Pittsburgh, Pittsburgh, Pennsylvania, United States of America

* jdy10@pitt.edu, xinghua@pitt.edu

Abstract

Cancer is a disease of aberrant cellular signaling and tumor-specific aberrations in signaling systems determine the aggressiveness of a cancer and response to therapy. Identifying such abnormal signaling pathways causing a patient's cancer would enable more patient-specific and effective treatments. We interpret the cellular signaling system as a causal graphical model, where it is known that genomic alterations cause changes in the functions of signaling proteins, and the propagation of signals among proteins eventually leads to changed gene expression. To represent such a system, we developed a deep learning model, referred to as a redundant input neural network (RINN), with a redundant input architecture and an L_1 regularized objective function to find causal relationships between input, latent, and output variables—when it is known *a priori* that input variables cause output variables. We hypothesize that training RINN on cancer omics data will enable us to map the functional impacts of genomic alterations to latent variables in a deep learning model, allowing us to discover the hierarchical causal relationships between variables perturbed by different genomic alterations. Importantly, the direct connections between all input and *all* latent variables in RINN make the latent variables partially interpretable, as they can be easily mapped to input space. We show that gene expression can be predicted from genomic alterations with reasonable accuracy when measured as the area under ROC curves (AUROCs). We also show that RINN is able to discover the shared functional impact of genomic alterations that perturb a common cancer signaling pathway, especially relationships in the PI3K, Nrf2, and TGF β pathways, including some causal relationships. However, despite high regularization, the learned causal relationships were somewhat too dense to be easily and directly interpretable as causal graphs. We suggest promising future directions for RINN, including differential regularization, autoencoder pretrained representations, and constrained evolutionary strategies.

Author summary

A modified deep learning model (RINN with L_1 regularization) can be used to capture cancer signaling pathway relationships within its hidden variables and weights. We found that genomic alterations impacting the same known cancer pathway had interactions with a similar set of RINN latent variables. Having genomic alterations (input variables) directly connected to all latent variables in the RINN model allowed us to label the latent variables with a set of genomic alterations, making the latent

variables partially interpretable. With this labeling, we were able to visualize RINNs as causal graphs and capture at least some of the causal relationships in known cancer signaling pathways. However, the graphs learned by RINN were somewhat too dense (despite large amounts of regularization) to compare directly to known cancer signaling pathways. We also found that differential expression can be predicted from genomic alterations by a RINN with reasonably high AUROCs, especially considering the very high dimensionality of the prediction task relative to the number of input variables and instances in the dataset. These are encouraging results for the future of deep learning models trained on cancer genomic data.

Introduction

The cellular mechanisms leading to cancer in an individual are heterogeneous, nuanced, and not well understood. It is well appreciated that cancer is a disease of aberrant signaling, and the state of a cancer cell can be described in terms of abnormally functioning cellular signaling pathways. Precision oncology depends on the ability to identify the abnormal cellular signaling pathways causing a patient’s cancer, so that patient-specific effective treatments can be prescribed—including targeting multiple abnormal pathways during a treatment regime. Aberrant signaling in cancer cells usually results from somatic genomic alterations (SGAs) that perturb the function of signaling proteins. Although large-scale cancer genomic data are available, such as those from The Cancer Genome Atlas (TCGA) and the International Cancer Genome Consortium (ICGC), it remains a very difficult and unsolved task to reliably infer how the SGAs in a cancer cell cause aberrations in cellular signaling pathways based on the genomics data of a tumor. One challenge is that the majority of the genomic alterations observed in a tumor are non-consequential (*passenger* genomic alterations) with respect to cancer development and only a few are *driver* genomic alterations, i.e., genomic alterations that are causing cancer. Furthermore, even if the driver genomic alterations of a tumor are known, it remains challenging to infer how aberrant signals of perturbed proteins affect the cellular system of cancer cells because the states of signaling proteins (or pathways) in the signaling system are not measured (latent). This requires one to study the causal relationships among latent (i.e., hidden or unobserved) variables, which represent the state of individual signaling proteins, protein complexes, or certain biological processes within a cell, in addition to the observed variables in order to understand the disease mechanisms of an individual tumor and identify drug targets. Most causal discovery algorithms have been developed to find causal structure and parameterizations of causal structure relative to the *observed* variables of a dataset [1–6]. Only a small number of causal discovery algorithms also find latent causal structure, especially in high-dimensional data [7–9].

Deep learning represents a group of machine learning strategies, based on neural networks, that learn a function mapping input to output. The signals of input variables are processed and transformed through many hidden layers of latent variables (i.e., hidden nodes) [10–12]. These hidden layers learn hierarchical or compositional statistical structures, meaning that different hidden layers capture structure of different degrees of complexity [9, 13–15]. Researchers have previously shown that deep learning models can represent the hierarchical organization of signaling molecules in a cell [16–20], with latent variables as natural representations of unobserved activation states of signaling molecules (e.g., membrane receptors or transcription factors). However, deep learning models have not been broadly used as a tool to infer *causal relationships* in the computational biology setting, partly due to deep learning’s “black box” nature.

Relevant to the work presented in this paper, here we briefly describe some of the

studies that used neural network-based approaches to discover gene regulatory networks (GRNs). A study in 1999 by Weaver and Stormo [21] modeled the relationships in gene regulatory networks as coefficients in weight matrices (using the familiar concepts of weighted-sums and an activation function in their model). However, they only tested their algorithm on simulated time series data as large gene expression datasets were not yet available and performing these calculations on large amounts of data was quite challenging at the time. Similarly, Vohradsky [22] and Keedwell et al. [23] also interpreted the weights of a quasi-recurrent neural network as relationships in gene regulatory networks. Again, time series simulated data was used in these studies with reasonable results. A more recent study from 2015 [24], used a linear classifier with one input “layer” and one output “layer” (which they called a neural network) to infer regulatory relationships (as represented by the weights of the weight matrix in the linear classifier) between genes in lung adenocarcinoma gene expression data. However, they did not evaluate their learned regulatory network, did not use DNA mutation data (as we do in this work), and did not use neural networks. There have also been studies that used genetic algorithms (evolving a weight matrix of regulatory pathways) to infer gene regulatory networks [25]. Like the studies above, this work was ahead of its time and they were only able to test their algorithm on simulated expression data. Improving upon Ando and Iba [25], Keedwell and Narayanan [26] used the weights representing a single-layer artificial neural network (ANN) (trained with gradient descent) to represent regulatory relationships between genes in gene expression data. Interestingly, they also used a genetic algorithm for a type of feature selection to make the task more tractable. They achieved good results on temporal simulated data and also tested on real temporal expression data (112 genes over nine time points), but did not have a ground truth to which to compare. In another study, Narayanan et al. [27] used single-layer ANNs to evaluate real, non-temporal gene expression data in a classification setting (i.e., they did not recover GRNs). However, the high-dimensionality of the problem was again a major limiting factor.

Many of the studies discussed above used the weights of a neural network to represent relationships in gene regulatory networks. However, it does not appear that these studies attempted to use a neural network to identify latent causal structure at different hierarchical levels (i.e., cellular signaling system), as we do in this work. In general, the studies above used very basic versions of a neural network (without regularization limiting the magnitude of the weights and thereby the complexity of the learned function) and due to computing constraints and the absence of large genomic datasets, were unable to train on high-dimensional data. Also, most of the methods above required temporal data, as opposed to the static genomic data that we utilize here. Overall, none of the studies discussed above, used a deep neural network (DNN) to predict expression data from genomic alteration data and then recover causal relationships in the weights of a DNN, as we do in this paper.

More recent work from our group used deep learning to simulate cellular signaling systems that were shared by human and rat cells [16] and to recover components of the yeast cellular signaling system, including transcription factors [17]. These studies utilized unsupervised learning methods, in contrast to the supervised methods used in this study. Also, the previous studies from our group did not attempt to find causal relationships representing the cellular signaling system.

In recent work [9], we developed a deep learning algorithm, named redundant input neural network (RINN), to learn causal relationships among latent variables from data inspired by cellular signaling pathways. RINN solves a problem, where a set of input variables *cause* the change in another set of output variables, and this causal interaction is mediated by a set of an unknown number of latent variables. The constraint of inputs

causing outputs is necessary to interpret the latent structure as causal relationships (see [9] for more details regarding the causal assumptions of the RINN). A key innovation of the RINN model is that it is a partially transparent model, allowing input variables to directly interact with all latent variables in its hierarchy. This allows the RINN to constrain (in conjunction with L_1 regularization) an input variable to be connected to a set of latent variables that can sufficiently encode the impact of the input variable on the output variables. In Young et al. [9], we showed that the RINN outperformed other algorithms, including neural network-based algorithms and a causal discovery algorithm known as DM (Detect MIMIC (Multiple Indicators Multiple Input Causes)), at identifying latent causal structure in various types of simulated data.

In the current study, we take advantage of the partially transparent nature of the RINN model and use the model to learn a representation of the cancer cellular signaling system. In this setting, we interpret the cellular signaling system as a hierarchical causal model of interactions between the activation states of proteins or protein complexes within a cell. Based on the assumption that somatic genome alterations (SGAs) that drive the development of a cancer often influence gene expression, we trained the RINN on a large number of tumors from TCGA using tumor SGAs as input to predict cancer differentially expressed genes (DEGs) (outputs). We then evaluated the latent structure learned in the hidden layers of the RINN in an attempt to learn components of the cancer cellular signaling system. We show that the model is capable of detecting the shared functional impact of SGAs affecting members of a common pathway. We also show that a RINN can capture cancer signaling pathway relationships within its hidden variables.

Methods

Data

The data used in this paper were originally downloaded from TCGA [28, 29]. RNA Seq, mutation, and copy number variation (CNV) data over multiple cancer types were used to generate two binary datasets. A binary differentially expressed gene (DEG) dataset was created by comparing the expression value of a gene in a tumor against a distribution of the expression values of the gene across normal samples from the same tissue of origin. A gene is deemed a DEG in a tumor if its value is outside the 2.5% percentile on either side of the normal sample distribution, and then that gene's value was set to 1. Otherwise, the gene's value was set to 0. A somatic genome alteration (SGA) dataset was created by using mutation and CNV data. A gene was deemed to be perturbed by an SGA event if it hosts a non-synonymous mutation, small insert/deletion, or somatic copy number alteration (deletion or amplification). If perturbed, the value in the tumor for that gene was set to 1, otherwise the value was set to 0.

We applied the tumor-specific causal inference (TCI) algorithm [29, 30] to these two matrices to identify the SGAs that causally influence gene expression in tumors and the union of their target DEGs. TCI is an algorithm that finds causal relationships between SGAs and DEGs in each individual tumor, without determining how the signal from the SGA is propagated in the cellular signaling system [29, 30]. We identified 372 SGAs that were deemed driver SGAs, as well as 5,259 DEGs that were deemed as target DEGs of the 372 SGAs using TCI. Overall, combining data from 5,097 tumors led to two data matrices (with dimensions of $5,097 \times 372$ and $5,097 \times 5,259$) as inputs and outputs for RINN, where the SGAs (inputs) are used to predict DEGs (outputs). The binarized data used in this study is available at

(https://github.com/young-jon/plos_comp_bio_2020).

Deep learning strategies: RINN and DNN

In this study, we used two deep learning strategies: RINN and DNN. A DNN is a conventional supervised feed-forward deep neural network. A DNN learns a function mapping inputs ($\mathbf{x}$) to outputs ($\mathbf{y}$) according to

$$f(\mathbf{x}) = \phi(\dots\phi(\phi(\mathbf{x} \cdot \mathbf{W}_1) \cdot \mathbf{W}_2) \dots \cdot \mathbf{W}_n) = \hat{\mathbf{y}} \quad (1)$$

where $\mathbf{W}_i$ represent the weight matrices between layers of a neural network, ϕ is some nonlinear function (i.e., activation function such as ReLU, softplus, or sigmoid), $\cdot$ represents vector-matrix multiplication, and $\hat{\mathbf{y}}$ is the predicted output. This function represents our predicted value for $\mathbf{y}$ and, in words, represents vector-matrix multiplication followed by application of a nonlinear function, repeated multiple times. An iterative procedure (stochastic gradient descent) is used to slowly change the values of all $\mathbf{W}_i$ to bring $\hat{\mathbf{y}}$ closer and closer to $\mathbf{y}$ (hopefully). The left side of Fig 1, without the redundant input nodes and corresponding redundant input weights, represents a DNN. DNNs also have bias vectors added at each layer (i.e., at each $\mathbf{h}_{i-1} \mathbf{W}_i$, where $\mathbf{h}_{i-1}$ is the previous layer's output), which have been omitted from the above equation for clarity. For a more detailed explanation of DNNs, please see [11].

We introduced the RINN for latent causal structure discovery in [9]. A RINN is similar to a DNN with a modification of the architecture. A RINN not only has the input fully connected to the first hidden layer, but also has copies of the input fully connected to each additional hidden layer (Fig 1). This structure allows a RINN to learn direct causal relationships between an input SGA and any hidden node in the RINN. Forward propagation through a RINN is performed as it is for a DNN with multiple vector-matrix multiplications and nonlinear functions, except that a RINN has hidden layers concatenated to copies of the input (Fig 1). Each hidden layer of a RINN with redundant inputs is calculated according to

$$\mathbf{h}^{(i)} = \phi([\mathbf{h}^{(i-1)}, \mathbf{x}] \cdot \mathbf{W}_i + \mathbf{b}_i) \quad (2)$$

where $\mathbf{h}^{(i-1)}$ represents the previous layer's output vector, $\mathbf{x}$ is the vector input to the neural network, $[\mathbf{h}^{(i-1)}, \mathbf{x}]$ represents concatenation into a single vector, $\mathbf{W}_i$ represent the weight matrix between hidden and redundant nodes in layer $i - 1$ and hidden layer i , ϕ is a nonlinear function (e.g., ReLU), $\cdot$ represents vector-matrix multiplication, and $\mathbf{b}_i$ represents the bias vector for layer i . In contrast to a RINN, a plain DNN calculates each hidden layer as

$$\mathbf{h}^{(i)} = \phi(\mathbf{h}^{(i-1)} \cdot \mathbf{W}_i + \mathbf{b}_i) \quad (3)$$

Backpropagation of errors and stochastic gradient descent for a RINN are the same as for a DNN, but with additional weights to optimize. For much more detailed information about the RINN, we recommend that you see our paper where we introduce it [9]. In addition to the architecture modification, all RINNs in this study importantly included L_1 regularization of the weights as part of the objective function. The other component of the objective function was binary cross-entropy error. All DNNs also used

Fig 1. Redundant input neural network (RINN) for TCGA data. A RINN with eight hidden layers ($h^{(1)}, \dots, h^{(8)}$, each with n nodes), m inputs ($sga_1, \dots, sga_m$ on bottom), p outputs ($deg_1, \dots, deg_p$), and seven sets of redundant inputs ($sga_1, \dots, sga_m$ on right side). Each node represents a scalar value and each edge represents a scalar weight. The weights between layers are collected in weight matrices $W_1, \dots, W_9$. Blue edges represent the weights used to create SGA weight signatures.

L_1 regularization of weights plus binary cross-entropy error as the objective function to be optimized. All RINNs used in this study had eight hidden layers (with varying sizes of the hidden layers), as we hypothesized that cancer cellular signaling pathways do not have more than eight levels of hierarchy. However, this assumption can easily be updated if found to be false.

Model selection

We hypothesized that since the RINN mimics the signaling processes through which the SGAs of a tumor exert their impact on gene expression (DEGs); the better a RINN model captures the relationships between SGAs and DEGs, the more closely the RINN will represent the signaling processes of the cancer cells, i.e., the causal network connecting SGAs and DEGs. We performed an extensive amount of model selection to search for the structures and parameterizations that best balance loss and sparsity, or in other words the models that fit the data well. To this end, we performed model selection over 23,000 different sets of hyperparameters (23,000 for RINN and 23,000 for DNN).

Model selection was performed using 3-folds of a 10-fold cross-validation. Using 3-folds of a 10-fold cross-validation gives us multiple validation datasets so we avoid chance overfitting that can occur with a single validation set. This setup also allows us to train on 90% of the data (and validate on 10%) for each split of the data, which is important considering the small number of instances relative to the number of output DEGs that we were trying to predict. Importantly, 3-folds of a 10-fold cross-validation takes significantly less time than training on all 10-folds. Running time is especially important for this study because of the large number of hyperparameter sets to be evaluated. All metrics recorded in this study (i.e., AUROC, cross-entropy error, etc.) represent the mean across the three validation sets.

The hyperparameters searched over in this study include: learning rate, regularization rate, the number of training epochs, activation function, size of hidden layers, number of hidden layers (DNN only; RINN set to eight), and batch size. We used a combined random and grid search approach [31,32] to find the best sets of hyperparameters with the main objective of finding the optimal balance between sparsity and cross-entropy error as explained in the next section and in [9].

Ranking models based on a balance between sparsity and prediction error

Model selection in this study is more complex than standard DNN model selection as we needed to find a balance between the sparsity of the model (i.e., sets of weight matrices for each trained network) and prediction error. In contrast, many DNNs are trained by simply finding the model with the lowest prediction error on a hold-out dataset. As was shown in previous work by our group [9], RINN models trained on simulated data with relatively high sparsity and low cross-entropy error were able to recover much of the latent causal structure contained within the data. We followed the same procedure as in our previous work [9] for selecting the best models (i.e., the models with the highest chance of containing correct causal structure). In brief, we plotted prediction error versus sparsity and measured the Euclidean distance from the origin to a set of hyperparameters, i.e., a unique trained neural network (blue circles in the model selection figure). This distance was measured according to

$$d_x = \sqrt{\left(\sum_{i=1}^m \sum_{j=1}^{r_i} \sum_{k=1}^{c_i} |w_{j,k}^{(i)}|\right)^2 + L^2} \quad (4)$$

where L is the validation set cross-entropy loss for neural network x , m is the number of weight matrices in neural network x , r_i and c_i are the number of rows and columns in matrix i respectively, and w is a scalar weight. The sets of hyperparameters were then ranked according to shortest distance to the origin (smallest d_x) and then we retrained on all data for further analysis of the learned weights. Please see [9] for a more detailed explanation.

AUROC and other metrics

Area under the receiver operating characteristics (AUROC) were calculated for each DEG and then averaged over the three validation sets, leading to 5,259 AUROCs for each classifier when plotted as a histogram of AUROCs. k -nearest neighbors (k NN) was performed using sklearn’s KNeighborsClassifier [33]). Both Euclidean and Jaccard distance metrics were evaluated and the best k values were 21 and 45, respectively. Other distance metrics were evaluated with results worse than those using the Jaccard distance metric. For the random control, we sampled predictions from a uniform distribution over the interval $[0, 1)$ for each of the validation sets. Then used these predictions to calculate AUROCs for each validation set. As with the other AUROC calculations, we took the mean over the three validation sets.

When displaying a single value for a classifier’s AUROC, the AUROCs were calculated as described above and then the mean over all DEGs was calculated. The same procedure was followed for calculating cross-entropy error, area under the precision-recall (AUPR) curve, and the sum of the absolute values of all weights. The same random predictions described above were also used to calculate cross-entropy error and AUPR.

Evaluation of learned relationships among SGAs

Throughout this work, the results from [34] were used as ground truth for comparison purposes. Specifically, Figures S1 and 2 were used as ground truth causal relationships that we hypothesized the RINN may be able to find. Figure S1 is reproduced in the Supporting information section, with a minor formatting modification for convenience (Fig S1). Of the 372 genes in our SGA dataset there were 35 genes that overlapped with the genes in Fig S1. Therefore, the causal relationships that we can find are limited to the relationships between these 35 genes (Table 1).

SGA weight signature

To better understand how a RINN learns to connect SGAs to hidden nodes, we analyzed only the weights going from SGA nodes to hidden nodes. To accomplish this, we generated an “SGA weight signature” for each SGA, which is the concatenation, into a single vector, of all weights for a single SGA going from that SGA to all hidden nodes in all hidden layers. This can be visualized as the concatenation of the blue weights in Fig 1 into one long vector for each SGA. The end result is an SGA weight signature matrix of SGAs by hidden nodes. The weight signatures for a DNN were generated using only a single weight matrix, the weight matrix between the inputs and the first

Table 1. Genes in SGA dataset that overlap with Sanchez-Vega et al., 2018.

Pathway from Fig S1	Genes in SGA Dataset and Fig S1
RTK/RAS	EGFR, FGFR1, ERBB2, FGFR2, FGFR3, PDGFRA, KRAS, HRAS
Nrf2	KEAP1, NFE2L2
TGF β	ACVR2A, SMAD4
PI3K	PTEN, PIK3CA, PIK3R1, AKT1, STK11
p53	MDM2, CDKN2A, TP53
Cell cycle	CDKN2A, CDKN2B, CCNE1, CCND1, CDK4, RB1, E2F3
Notch	NOTCH1, FBXW7, EP300
Hippo	DCHS2, FAT1, NF2
Wnt	RNF43, APC, CTNNB1

hidden layer (i.e., $\mathbf{W}_1$), as these are the only weights in a DNN that are specific to individual SGAs.

Cosine similarity between SGA weight signatures was measured using the sklearn function *cosine_similarity* [33].

Hierarchical clustering using cosine similarity and average linkage was performed on the SGA weight signatures using the seaborn python module function *clustermap*.

Community detection

Cosine similarity community detection figures were generated using Gephi [35] and the cosine similarity between SGA weight signatures. Edges represented the highest or three highest cosine similarities for a given SGA. After generating a graph where nodes were SGAs and edges were highest cosine similarities, we ran the *Modularity* function in Gephi to perform community detection. *Modularity* runs an algorithm based on [36,37]. Next, we partitioned and colored the graph based on community.

Visualizing a RINN as a causal graph

Let $G_{i,j} = (V, E)$, where $G_{i,j}$ is a causal directed acyclic graph with vertices V and edges E for neural network i and set of SGAs j . For this work, V represents SGAs and hidden nodes, and E represents directed weighted edges with weight values corresponding to the weights of a RINN. The edges are directed from SGAs (input) to DEGs (output), as we know from biology that SGAs cause changes in expression. Please see [9] for further explanation of interpreting a RINN in a causal framework.

If we simply interpreted any nonzero weight in a trained RINN as an edge in a $G_{i,j}$, there would be hundreds of thousands to millions of edges in the causal graph (depending on the size of the hidden layers) as L_1 regularization encourages weight values toward zero, but weights are often not actually zero. This means that some thresholding of the weights was required [9].

Even after selecting models based on the best balance between sparsity and error (smallest d_x), our weight matrices were still very dense in terms of what can be readily interpreted visually (even after rounding weights to zero decimals). Therefore, a threshold weight value was needed to limit weight visualizations to only the largest (and we suspect most causally important) weights. To accomplish this, we first limited the weights to be visualized to only those that are descendants (i.e., downstream) of any of the 35 SGAs from Fig S1. Next, for each of the top ten RINN models (ten shortest d_x), we found the absolute value weight threshold that led to 300 edges in total, including all SGA to hidden and hidden to hidden edges. This threshold varied slightly from model

to model, ranging from 0.55 to 0.71. This threshold was chosen as it seemed to give a biologically reasonable density of edges, allowed us to recover some of the relationships in Fig S1, and was low enough to allow at least some of the causal paths to proceed from input all the way to output.

After finding a threshold for each model, any weight whose absolute value was greater than the threshold was added as an edge to the causal graph $G_{i,j}$ for that model. The causal graphs were then plotted as modified bipartite graphs with SGAs on one side and all hidden nodes on the other. Hidden to hidden edges were included as arcing edges on the outside of the bipartite graph. We labeled hidden nodes using a recursive algorithm that found all ancestor or upstream SGAs (i.e., on a path to that hidden node) for a given hidden node and graph $G_{i,j}$.

Finding hidden nodes encoding similar information with respect to SGAs

To determine if hidden nodes with similar connectivity patterns with respect to the input SGAs were shared across the best models, we needed a method to map hidden nodes to some meaningful label. To accomplish this, we used the same recursive algorithm (described at the end of the previous section) to map each hidden node in a causal graph $G_{i,j}$ to the set of SGAs that were ancestors of that hidden node. For this mapping, only the SGAs in the set j could be used to label a hidden node as these are the only SGAs in $G_{i,j}$. We performed this mapping using graphs generated with j set to only the SGAs in the individual pathways from Fig S1 (e.g., $j = \{AKT1, PIK3CA, PIK3R1, PTEN, STK11\}$ for the PI3K pathway). For each model in the best ten models and each pathway in Fig S1, we mapped hidden nodes to a set of SGAs. Next, we compared the labeled hidden nodes across models and determined the number of models that shared identically labeled hidden nodes.

We compared the number of models that shared specific hidden nodes with random controls. The random controls were performed the same as described above for the experimental results, except that j was not set to the SGAs in one of the pathways in Fig S1—rather j was selected randomly from the set of SGAs including all 372 SGAs minus the 35 SGAs in Fig S1. Then ten $G_{i,j}$, one for each of the top ten RINN models, were generated using the random SGAs. The number of models with shared hidden nodes was recorded. This procedure was repeated 30 times for each possible number of SGAs in j . For example, the PI3K pathway (in Fig S1) has five SGAs in it that are also in our SGA dataset. To perform the random control for PI3K, we performed 30 replicates of randomly selecting five SGAs (from the set of SGAs described above) and then recorded the mean number of models sharing an n -SGA labeled hidden node, where n is the number of SGAs a hidden node mapped to using our recursive algorithm for finding ancestors.

Results

Model selection

We trained approximately 23,000 RINN and 23,000 DNN models on the TCGA training dataset with distinct sets of hyperparameters (e.g., number of hidden nodes, activation function, regularization rate, etc.). We evaluated each trained model on multiple validation datasets to evaluate how well it performed. We hypothesized that the models with the most parsimonious weight structure, while still maintaining the ability to accurately capture the statistical relationship between SGAs and DEGs, likely have

learned optimal representations of the impact of SGAs in cancer cells. To reflect the balance between these two objectives, we visualized the performance of all models on a scatter plot with validation set error and the sum of the absolute value of all weights as axes (Fig 2). Each blue dot in this figure represents a neural network trained on a unique set of hyperparameters. We ranked models based on their Euclidean distance to the origin (d_x), and models with the shortest d_x were selected as the models with the best balance between sparsity and validation set error.

The ten best RINN models (i.e., models with lowest d_x) are shown in Table 2. Among multiple activation functions studied, the softplus activation function overwhelmingly provided the best results. This was also seen with the best DNN models. Overall, the number of hidden nodes in each hidden layer for the RINN were relatively small (~ 100) compared to the dimensionality of output space (5,259). Despite being trained with eight hidden layers, the best RINN models utilized only three or four of the eight hidden layers. All top ten DNN models had two hidden layers, with sizes ranging from 50 to 644 hidden nodes.

Table 2. Best RINN hyperparameters.

Rank	Train Epochs	Nodes in Hidden	LR	RR	BS	Activation	CEL	$\sum_1^8 \mathbf{W}_i $
1	303	50	1E-03	5E-06	135	softplus	0.5073	4248
2	144	100	1E-03	6E-06	45	softplus	0.5081	3874
3	738	319	1.64E-04	6.37E-06	30	softplus	0.5089	3373
4	428	326	2.37E-04	5.59E-06	30	softplus	0.5079	4042
5	136	50	1E-03	6E-06	30	softplus	0.5082	4021
6	311	50	1E-03	6E-06	135	softplus	0.5090	3596
7	713	148	1.1E-04	5.17E-06	30	softplus	0.5083	4038
8	326	100	1E-03	6E-06	170	softplus	0.5088	3726
9	149	101	5.23E-04	4.49E-06	30	softplus	0.5057	5030
10	345	180	3.02E-04	5.66E-06	30	softplus	0.5085	3925

Hyperparameters for the ten RINN models with lowest Euclidean distance to origin. Each row in the table represents a set of hyperparameters that was used to fully train a RINN model. Also included are the cross-entropy errors and sum of the absolute values of the weights. ($\mathbf{W}_i$ is a weight matrix between hidden layers. LR: learning rate, RR: regularization rate, BS: batch size, CEL: cross-entropy loss).

Predicting DEG status given SGAs

We examined how well deep learning models can predict DEGs given SGAs as inputs, using different metrics, including cross-entropy loss, AUROC, and AUPR. Cross-validation model selection metrics for RINN and DNN are compared in Table 3. Table 3 shows the mean and standard deviation of the metrics on all 5,259 DEGs. In general, RINN and DNN performed similar across all metrics. RINN and DNN performed significantly better than k -nearest neighbors or random controls. The best RINN and DNN models according to shortest d_x (meaning more regularized) performed similarly, but slightly worse, than the much less regularized RINNs and DNNs. $\sum_1^8 |\mathbf{W}_i|$ is a surrogate measure of the density of a neural network, with higher values indicating higher density of edges, in general.

To get a better idea of how well the models performed at predicting individual DEGs from SGAs, we plotted DEG AUROC histograms for individual models and relevant control models (Fig 3). Fig 3 shows AUROCs for all 5,259 DEGs for the best RINNs and DNNs. RINN and DNN vastly outperformed the control models with much higher AUROCs. Interestingly, for many DEGs, the models can achieve AUROC greater than 0.8. Again, DNNs and RINNs performed similarly when compared to each other.

Fig 2. Model selection results. A: RINN. B: DNN.

Table 3. Cross-validation model selection results.

	Description	CEL	AUROC	AUPR	$\sum_1^8 W_i $
RINN 1	lowest CEL	0.5032 $\pm$ 0.0048	0.7316 $\pm$ 0.0051	0.5678 $\pm$ 0.0039	18899 $\pm$ 250
RINN 2	2nd lowest CEL	0.5032 $\pm$ 0.0047	0.7317 $\pm$ 0.0048	0.5677 $\pm$ 0.0036	16971 $\pm$ 53
DNN 1	lowest CEL	0.5033 $\pm$ 0.0052	0.7318 $\pm$ 0.0058	0.5676 $\pm$ 0.0041	24503 $\pm$ 124
DNN 2	2nd lowest CEL	0.5033 $\pm$ 0.0049	0.7307 $\pm$ 0.0056	0.5663 $\pm$ 0.0032	12739 $\pm$ 75
RINN ED 1	shortest ED	0.5073 $\pm$ 0.0054	0.7258 $\pm$ 0.0054	0.5607 $\pm$ 0.0040	4248 $\pm$ 18
RINN ED 2	2nd shortest ED	0.5081 $\pm$ 0.0058	0.7231 $\pm$ 0.0057	0.5569 $\pm$ 0.0038	3874 $\pm$ 48
DNN ED 1	shortest ED	0.5075 $\pm$ 0.0055	0.7238 $\pm$ 0.0057	0.5570 $\pm$ 0.0029	3491 $\pm$ 18
DNN ED 2	2nd shortest ED	0.5073 $\pm$ 0.0054	0.7240 $\pm$ 0.0054	0.5576 $\pm$ 0.0031	3704 $\pm$ 45
<i>k</i> -NN	<i>k</i> = 21, euclidean	9.6289 $\pm$ 0.1889	0.5599 $\pm$ 0.0034	0.5396 $\pm$ 0.0022	NA
<i>k</i> -NN	<i>k</i> = 45, jaccard	9.0740 $\pm$ 0.1579	0.5799 $\pm$ 0.0028	0.5719 $\pm$ 0.0027	NA
Random	preds $\sim U(0, 1)$	0.9994 $\pm$ 0.0003	0.5003 $\pm$ 0.0003	0.3257 $\pm$ 0.0052	NA

Comparing the two RINN and DNN models with the lowest mean cross-entropy validation set loss and shortest euclidean distance to the origin after evaluating $\sim 23,000$ different sets of hyperparameters. Values represent the mean across three validation sets. The four best models for each metric in bold. (W_i is a weight matrix between hidden layers. CEL: cross-entropy loss, ED: Euclidean distance).

Fig 3. AUROC for RINN and DNN.

Importantly, there wasn't a large difference in the AUROC curves between RINN (and DNN) models selected according to lowest CEL (cross-entropy loss) and those selected using shortest ED (euclidean distance), despite the models selected according to shortest ED being much more regularized.

Learning shared functional impacts of SGAs using the RINN

RINN allows an SGA to interact with all hidden nodes in the latent hierarchy. After training with regularization, the majority of the weights from SGAs to hidden nodes were set close to 0.0, thus the remaining weights reflect how the impact of the SGAs is propagated through the latent variables and eventually influence gene expression. In other words, these remaining weights (edges) between the SGAs and hidden nodes serve as a signature (or a vector embedding) representing an SGA's functional impact. We set out to assess whether the weight signatures learned by RINN truly reflect the functional impact of SGAs. Here, one can measure the closeness of the functional impacts of two SGAs by using the cosine angle similarity. We hypothesized that if RINN correctly captures the functional impact of SGAs, then the weight signature of a gene (SGA) should be more similar to genes belonging to a common pathway than to genes from other pathways. To this end, we used the well documented cancer pathways recently reported by the TCGA pan-cancer analysis project [34] (Fig S1) to determine which genes shared a common pathway.

For each of the 35 driver SGAs that are in the known pathways from the TCGA pan-cancer study [34] (and also are in our SGA dataset), we calculated the cosine similarity of its weight signature with respect to all other SGA weight signatures, and listed the three most similar SGAs in Table 4. Clearly, many of the SGAs that had high cosine similarities relative to the query SGA were in the same pathway as the query SGA according to [34]. Most of the SGAs in the PI3K pathway had high cosine similarities with the other SGAs in the PI3K pathway. *KEAP1* and *NFE2L2* (the only members of the Nrf2 pathway in our SGA dataset) were both found to be each other's SGA with highest cosine similarity (i.e., each other's closest neighbor). Also, *SMAD4* and *ACVR2A* (only members of the TGF β pathway in the SGA dataset) were also found to be each other's SGA with highest cosine similarity. Additional cosine similarity relationships were found between members of the same pathway for the remaining pathways, but these relationships were less frequent than the results for PI3K, Nrf2, and TGF β mentioned above. These cosine similarity results suggest that the hidden nodes in our trained RINNs may represent biological entities or components of a cellular signaling system involved in propagating signals of SGA events, indicating that the connectivity found in our trained RINNs are related to cellular signaling pathways.

To compare the RINN with a DNN, we also performed the same cosine similarity experiment from Table 4, but with weights from the best three DNN models (Table 5). For this experiment, we only used weights from the input SGAs to the first hidden layer, as these are the only weights in a DNN that are specific to individual SGAs. In Table 5, many of the SGAs that had high cosine similarities relative to the query SGA were in the same pathway as the query SGA according to [34]. Overall, there were 28 genes in bold in Table 4 (RINN) and 27 genes in bold in Table 5 (DNN). The DNN performed marginally worse than the RINN at capturing the relationships in the pathways, mostly because the DNN failed to robustly capture as much of the shared function of the SGAs affecting the PI3K pathway as the RINN. However, the DNNs performed slightly better at capturing pathway relationships in the RTK/RAS pathway (especially with cosine similarities relative to *FGFR3*). Just as with the RINN, *KEAP1* and *NFE2L2* were both found to be each other's SGA with highest cosine similarity with the DNN. *SMAD4* and *ACVR2A* were also found to be similar to one another, but not as highly similar as with the RINN. These results indicate that when a DNN can accurately predict DEGs (outputs) from SGAs (inputs), the DNN weight signature cosine similarities can also capture the functional similarity of SGAs.

Table 4. RINN top three SGA weight signature cosine similarity (CS).

x	Highest CS (Mean $\pm$ SD)	2nd Highest	3rd Highest
PTEN	PIK3R1 (0.8 $\pm$ 0.02)	PDGFRA (0.4 $\pm$ 0.01)	PIK3CA (0.4 $\pm$ 0.05)
PIK3CA	PIK3R1 (0.6 $\pm$ 0.04)	AKT1 (0.4 $\pm$ 0.06)	PTEN (0.5 $\pm$ 0.05)
PIK3R1	PTEN (0.8 $\pm$ 0.02)	PIK3CA (0.6 $\pm$ 0.04)	CTNNB1 (0.4 $\pm$ 0.05)
AKT1	PIK3CA (0.4 $\pm$ 0.06)	FGFR2 (0.4 $\pm$ 0.03)	DCHS2 (0.3 $\pm$ 0.09)
STK11	KEAP1 (0.5 $\pm$ 0.05)	FGFR1 (0.5 $\pm$ 0.05)	NFE2L2 (0.4 $\pm$ 0.10)
KEAP1	NFE2L2 (0.6 $\pm$ 0.09)	STK11 (0.5 $\pm$ 0.05)	KRAS (0.4 $\pm$ 0.08)
NFE2L2	KEAP1 (0.6 $\pm$ 0.09)	STK11 (0.4 $\pm$ 0.10)	FGFR1 (0.3 $\pm$ 0.11)
ACVR2A	SMAD4 (0.5 $\pm$ 0.04)	RNF43 (0.4 $\pm$ 0.20)	APC (0.2 $\pm$ 0.10)
SMAD4	ACVR2A (0.5 $\pm$ 0.04)	APC (0.4 $\pm$ 0.06)	DCHS2 (0.4 $\pm$ 0.02)
CDKN2A	NOTCH1 (0.7 $\pm$ 0.09)	HRAS (0.5 $\pm$ 0.07)	FAT1 (0.4 $\pm$ 0.10)
CDKN2B	RB1 (0.4 $\pm$ 0.14)	EGFR (0.4 $\pm$ 0.18)	CDK4 (0.4 $\pm$ 0.18)
CCNE1	E2F3 (0.5 $\pm$ 0.23)	RB1 (0.3 $\pm$ 0.20)	MDM4 (0.3 $\pm$ 0.20)
CCND1	FGFR2 (0.7 $\pm$ 0.05)	DCHS2 (0.3 $\pm$ 0.11)	FBXW7 (0.3 $\pm$ 0.14)
CDK4	PDGFRA (0.5 $\pm$ 0.10)	EGFR (0.4 $\pm$ 0.11)	CDKN2B (0.4 $\pm$ 0.18)
RB1	FGFR3 (0.5 $\pm$ 0.07)	E2F3 (0.4 $\pm$ 0.04)	CDKN2B (0.4 $\pm$ 0.14)
E2F3	CCNE1 (0.5 $\pm$ 0.23)	FGFR3 (0.4 $\pm$ 0.10)	RB1 (0.4 $\pm$ 0.04)
EGFR	PDGFRA (0.5 $\pm$ 0.10)	CDK4 (0.4 $\pm$ 0.11)	CDKN2B (0.4 $\pm$ 0.18)
FGFR1	DCHS2 (0.6 $\pm$ 0.11)	STK11 (0.5 $\pm$ 0.05)	FBXW7 (0.5 $\pm$ 0.07)
ERBB2	EP300 (0.6 $\pm$ 0.08)	FGFR3 (0.5 $\pm$ 0.21)	RB1 (0.4 $\pm$ 0.08)
FGFR2	CCND1 (0.7 $\pm$ 0.05)	FBXW7 (0.4 $\pm$ 0.09)	DCHS2 (0.4 $\pm$ 0.11)
FGFR3	RB1 (0.5 $\pm$ 0.07)	EP300 (0.5 $\pm$ 0.02)	ERBB2 (0.5 $\pm$ 0.21)
PDGFRA	CDK4 (0.5 $\pm$ 0.10)	EGFR (0.5 $\pm$ 0.10)	PTEN (0.4 $\pm$ 0.10)
KRAS	CTNNB1 (0.5 $\pm$ 0.06)	KEAP1 (0.4 $\pm$ 0.08)	RNF43 (0.3 $\pm$ 0.15)
HRAS	NOTCH1 (0.7 $\pm$ 0.05)	FAT1 (0.6 $\pm$ 0.05)	CDKN2A (0.5 $\pm$ 0.07)
MDM4	PDGFRA (0.3 $\pm$ 0.14)	EGFR (0.3 $\pm$ 0.16)	CCNE1 (0.3 $\pm$ 0.10)
TP53	E2F3 (0.4 $\pm$ 0.03)	RB1 (0.3 $\pm$ 0.08)	CCNE1 (0.3 $\pm$ 0.09)
RNF43	APC (0.4 $\pm$ 0.03)	ACVR2A (0.4 $\pm$ 0.20)	KRAS (0.3 $\pm$ 0.15)
APC	FBXW7 (0.5 $\pm$ 0.06)	DCHS2 (0.5 $\pm$ 0.04)	SMAD4 (0.4 $\pm$ 0.06)
CTNNB1	KRAS (0.5 $\pm$ 0.06)	PIK3R1 (0.4 $\pm$ 0.08)	FBXW7 (0.4 $\pm$ 0.07)
DCHS2	FBXW7 (0.8 $\pm$ 0.04)	EP300 (0.6 $\pm$ 0.04)	FGFR1 (0.6 $\pm$ 0.11)
FAT1	NOTCH1 (0.8 $\pm$ 0.07)	HRAS (0.6 $\pm$ 0.05)	CDKN2A (0.4 $\pm$ 0.10)
NF2	EP300 (0.3 $\pm$ 0.16)	FBXW7 (0.3 $\pm$ 0.10)	NOTCH1 (0.3 $\pm$ 0.03)
FBXW7	DCHS2 (0.8 $\pm$ 0.04)	EP300 (0.6 $\pm$ 0.08)	APC (0.5 $\pm$ 0.06)
NOTCH1	FAT1 (0.8 $\pm$ 0.07)	HRAS (0.7 $\pm$ 0.05)	CDKN2A (0.7 $\pm$ 0.09)
EP300	FBXW7 (0.6 $\pm$ 0.08)	DCHS2 (0.6 $\pm$ 0.04)	ERBB2 (0.6 $\pm$ 0.08)

SGAs from Fig S1 with highest cosine similarity relative to SGA x (mean CS across three best models $\pm$ st. dev.). Bolded SGAs are in the same pathway (according to Sanchez-Vega et al., 2018) as SGA x .

An alternative approach to illustrate the information provided by SGA weight signatures is to visualize the highest cosine similarities between weight signatures as edges in a graph and then apply a community discovery algorithm to identify closely related SGAs (Fig 4). In Fig 4, the discovered communities are shown as nodes and edges with the same color. In this figure, a directed edge connects an SGA (at the tail) to another SGA (at the head) that has high cosine similarity relative to the SGA at the tail of the edge; a bi-directed edge indicates that two SGAs are mutually highly similar; the thickness of an edge represents the amount of cosine similarity; and the size of the SGA node represents the degree of that node. Fig 4A is a visualization of the highest cosine similarities for each SGA and Fig 4B is a visualization of the three highest cosine

Table 5. DNN top three SGA weight signature cosine similarity (CS).

x	Highest CS (Mean $\pm$ SD)	2nd Highest	3rd Highest
PTEN	PIK3R1 (0.7 $\pm$ 0.06)	FGFR2 (0.5 $\pm$ 0.05)	PDGFRA (0.4 $\pm$ 0.03)
PIK3CA	AKT1 (0.7 $\pm$ 0.06)	FGFR2 (0.5 $\pm$ 0.02)	CCND1 (0.5 $\pm$ 0.10)
PIK3R1	FGFR2 (0.7 $\pm$ 0.07)	PTEN (0.7 $\pm$ 0.06)	CTNNB1 (0.6 $\pm$ 0.02)
AKT1	PIK3CA (0.7 $\pm$ 0.06)	FGFR2 (0.5 $\pm$ 0.02)	FGFR1 (0.4 $\pm$ 0.14)
STK11	KEAP1 (0.6 $\pm$ 0.02)	FGFR1 (0.4 $\pm$ 0.12)	NFE2L2 (0.4 $\pm$ 0.01)
KEAP1	NFE2L2 (0.7 $\pm$ 0.03)	STK11 (0.6 $\pm$ 0.02)	KRAS (0.4 $\pm$ 0.03)
NFE2L2	KEAP1 (0.7 $\pm$ 0.03)	STK11 (0.4 $\pm$ 0.01)	NOTCH1 (0.4 $\pm$ 0.04)
ACVR2A	RNF43 (0.6 $\pm$ 0.03)	SMAD4 (0.5 $\pm$ 0.03)	CCNE1 (0.4 $\pm$ 0.15)
SMAD4	APC (0.5 $\pm$ 0.04)	FBXW7 (0.5 $\pm$ 0.02)	ACVR2A (0.5 $\pm$ 0.03)
CDKN2A	NOTCH1 (0.8 $\pm$ 0.06)	CDK4 (0.7 $\pm$ 0.02)	HRAS (0.6 $\pm$ 0.02)
CDKN2B	CDK4 (0.6 $\pm$ 0.04)	RB1 (0.5 $\pm$ 0.07)	EGFR (0.4 $\pm$ 0.03)
CCNE1	E2F3 (0.6 $\pm$ 0.08)	TP53 (0.4 $\pm$ 0.07)	ACVR2A (0.4 $\pm$ 0.15)
CCND1	FGFR2 (0.6 $\pm$ 0.06)	PIK3CA (0.5 $\pm$ 0.10)	CTNNB1 (0.5 $\pm$ 0.12)
CDK4	CDKN2A (0.7 $\pm$ 0.02)	CDKN2B (0.6 $\pm$ 0.04)	EGFR (0.5 $\pm$ 0.03)
RB1	FGFR3 (0.5 $\pm$ 0.05)	TP53 (0.5 $\pm$ 0.02)	CDKN2B (0.5 $\pm$ 0.07)
E2F3	CCNE1 (0.6 $\pm$ 0.08)	FGFR3 (0.5 $\pm$ 0.06)	RB1 (0.4 $\pm$ 0.06)
EGFR	PDGFRA (0.6 $\pm$ 0.05)	CDK4 (0.5 $\pm$ 0.03)	MDM4 (0.5 $\pm$ 0.06)
FGFR1	FBXW7 (0.5 $\pm$ 0.10)	DCHS2 (0.5 $\pm$ 0.18)	STK11 (0.4 $\pm$ 0.12)
ERBB2	EP300 (0.6 $\pm$ 0.03)	FGFR3 (0.5 $\pm$ 0.03)	DCHS2 (0.5 $\pm$ 0.10)
FGFR2	PIK3R1 (0.7 $\pm$ 0.07)	CCND1 (0.6 $\pm$ 0.06)	CTNNB1 (0.6 $\pm$ 0.04)
FGFR3	ERBB2 (0.5 $\pm$ 0.03)	HRAS (0.5 $\pm$ 0.07)	EP300 (0.5 $\pm$ 0.08)
PDGFRA	EGFR (0.6 $\pm$ 0.05)	MDM4 (0.5 $\pm$ 0.14)	PTEN (0.4 $\pm$ 0.03)
KRAS	CTNNB1 (0.6 $\pm$ 0.07)	APC (0.5 $\pm$ 0.04)	SMAD4 (0.5 $\pm$ 0.02)
HRAS	NOTCH1 (0.7 $\pm$ 0.03)	CDKN2A (0.6 $\pm$ 0.02)	EP300 (0.6 $\pm$ 0.10)
MDM4	PDGFRA (0.5 $\pm$ 0.14)	EGFR (0.5 $\pm$ 0.06)	CDKN2B (0.3 $\pm$ 0.17)
TP53	RB1 (0.5 $\pm$ 0.02)	CCNE1 (0.4 $\pm$ 0.07)	KRAS (0.4 $\pm$ 0.06)
RNF43	ACVR2A (0.6 $\pm$ 0.03)	APC (0.4 $\pm$ 0.02)	KRAS (0.3 $\pm$ 0.03)
APC	FBXW7 (0.6 $\pm$ 0.02)	SMAD4 (0.5 $\pm$ 0.04)	KRAS (0.5 $\pm$ 0.04)
CTNNB1	PIK3R1 (0.6 $\pm$ 0.02)	FGFR2 (0.6 $\pm$ 0.04)	KRAS (0.6 $\pm$ 0.07)
DCHS2	FBXW7 (0.7 $\pm$ 0.08)	EP300 (0.6 $\pm$ 0.06)	FGFR1 (0.5 $\pm$ 0.18)
FAT1	NOTCH1 (0.8 $\pm$ 0.04)	CDKN2A (0.6 $\pm$ 0.09)	HRAS (0.5 $\pm$ 0.05)
NF2	EP300 (0.4 $\pm$ 0.05)	FAT1 (0.4 $\pm$ 0.05)	NOTCH1 (0.3 $\pm$ 0.04)
FBXW7	DCHS2 (0.7 $\pm$ 0.08)	EP300 (0.7 $\pm$ 0.09)	APC (0.6 $\pm$ 0.02)
NOTCH1	CDKN2A (0.8 $\pm$ 0.06)	FAT1 (0.8 $\pm$ 0.04)	HRAS (0.7 $\pm$ 0.03)
EP300	FBXW7 (0.7 $\pm$ 0.09)	ERBB2 (0.6 $\pm$ 0.03)	DCHS2 (0.6 $\pm$ 0.05)

SGAs from Fig S1 with highest cosine similarity relative to SGA x (mean CS across three best models $\pm$ st. dev.). Bolded SGAs are in the same pathway (according to Sanchez-Vega et al., 2018) as SGA x .

similarities for each SGA. The nodes and edges are colored according to community after running a community detection algorithm. Many of the communities that were discovered with this procedure correspond to the pathways in Fig S1. For example, the green community in Fig 4A captures the majority of the PI3K pathway, the darker blue community and brown community captures the TGF β and Nrf2 pathways respectively, and the purple community captures many of the genes (SGAs) in the Wnt, Hippo, and Notch pathways. In Fig 4B the majority of the PI3K pathway is part of the green community and a blue community emerges with many members of the cell cycle pathway. Many other additional pathway relationships within the ground truth pathways are seen in Fig 4, reinforcing the hypothesis that the latent structure of

trained RINNs contains cellular signaling pathway relationships.

We also visualized the weights from SGAs to hidden nodes (i.e., weight signatures) as a heatmap, where each column represents an SGA and each row represents a hidden node, and the weight from an SGA to a hidden node is represented as an element in the color-coded heatmap. We performed hierarchical clustering on the weight signatures derived from a single RINN model (the RINN with the shortest d_x) as shown in Fig 5. Within this clustering and dendrogram, we observe many relationships that are present in the ground truth pathways. For example, the green cluster contains much of the PI3K pathway, the orange cluster captures the TGF β pathway, the purple cluster contains the two members of the Nrf2 pathway, and the gray cluster contains the members of the Notch pathway and two of the three members of the Hippo pathway.

In addition to the above clustering relationships, the heatmap also illustrates the ability of the RINN to automatically determine the “optimal” number of hidden layers that are needed to encode information from SGAs to DEGs. Here, the hidden layers are numbered starting from input (SGAs) to output (DEGs). This heatmap shows that all the weights before hidden layer 5 have a value of zero, and only one hidden node in layer 5 has incoming weights with nonzero values. All of the top ten models (including this one) utilized only three or four hidden layers similar to what is seen in Fig 5. Hidden layer 7 seems to have the most nonzero incoming weights. Interestingly, many of the most important cancer related SGAs (e.g., *EGFR*, *TP53*, *CDKN2A*, *APC*, *PIK3CA*) have a larger number of weights and larger valued weights than other SGAs in Fig 5, suggesting that this model is capturing relevant biological information.

Inferring causal relationships using RINN

One of the goals of this work was to discover latent causal structure relevant to cancer cellular signaling pathways. We hypothesized that the RINN can reveal the causal relationships among the proteins perturbed by SGAs. For example, if an SGA (intervention) i_1 is connected to a latent variable h_1 and another SGA i_2 is connected to another latent variable h_2 ; if h_1 is connected directly upstream of h_2 with a nonzero weighted edge in the latent hierarchy, it would suggest that the signal of i_1 is upstream of that of i_2 and that the protein affected by i_1 causally regulates the protein affected by i_2 in the cellular system. To examine this type of relationship, we visualized the relationships among the SGAs from four TCGA pathways via the latent nodes they are connected to, as a means to search for causal structure revealed by the RINN models (Fig 6). Overall, the learned causal graphs were more complex and dense than the graphs from the pan-cancer TCGA analysis (Fig S1) in that an SGA in the RINN was often connected to multiple latent variables (depending on the cutoff threshold). This suggests that there is still some difficulty in directly interpreting the weights of a RINN as causal relationships with the current version of the algorithm. It may also suggest that the ground truth we used here is not detailed enough for our purposes, or may have some inaccuracies, and thus is more of a “silver standard”. In general, the learned causal graphs had a large number of false positive edges (when compared to Fig S1) and multiple instances of redundant causal edges—where a single SGA will act multiple times on the same path.

However, some true causal relationships are seen within these graphs. For example, *both* models 0 and 1 have a directed path as follows: $i_{KEAP1} \rightarrow h_1 \rightarrow h_2 \leftarrow i_{NFE2L2}$, where i represents an intervention (i.e., SGA). If we assume that h_1 is a representation of the *KEAP1* protein and h_2 is a representation of *NFE2L2* protein, then we can interpret the above path as, *KEAP1* $\rightarrow$ *NFE2L2*, i.e., *KEAP1* causing or changing the state of *NFE2L2*. This is consistent with the Nrf2 pathway shown in Fig S1. Model 2 has the

Fig 4. Cosine similarity of SGA weight signatures and community detection. These edges do not represent causal relationships, rather cosine similarity (CS) relationships between SGAs, where the head of an edge indicates an SGA with high CS relative to the SGA at the tail. A: Edges represent highest cosine similarity for each SGA. B: Edges represent three highest cosine similarities for each SGA.

Fig 5. Hierarchical clustering of RINN with shortest euclidean distance. The heatmap shows all SGA to hidden node weight values for our best trained RINN model. The vertical axis represents hidden node number (numbered in ascending order starting from hidden layer one—closest to SGAs) and has horizontal lines representing different hidden layer boundaries. The horizontal axis represents the 35 SGAs from Fig S1. The dendrogram and coloring on top show the clustering of the SGAs for one particular clustering cutoff.

following directed path: $KEAP1 \rightarrow h_1 \leftarrow NFE2L2$, which has a more ambiguous interpretation but does emphasize the correlation between these two SGAs. Also, it is plausible that if the weight threshold were decreased slightly for Model 2, there would be an edge between the green hidden node and the blue hidden node, meaning that all three models would then capture the $KEAP1 \rightarrow NFE2L2$ causal relationship. This emphasizes the importance of finding a more robust means of thresholding for future work.

All the RINNs represented in Fig 6 (and Fig 5) only utilized three or four hidden layers to learn the function mapping SGAs to DEGs. Because we used L_1 regularization,

Fig 6. Visualizing the weights of a RINN as causal graphs. Each graph represents the causal relationships for only the SGAs on the left of the graph.

RINNs learned to only use the number of hidden layers necessary to perform well on the prediction task. For example, if only three hidden layers were needed to perform well on prediction, the regularized objective function would set all the weights before the last three hidden layers to 0.0. We set the number of hidden layers to eight as we hypothesized that eight hidden layers was enough hierarchical complexity to capture cancer cellular signalling pathways in a meaningful way. Fig 5 and Fig 6 suggest that this was a correct assumption in this setting for the data we used. As with other model selection techniques, if our trained RINNs had utilized all eight layers during model selection, we could easily increase the number of hidden layers to more than eight and perform another round of model selection.

Shared hidden nodes across top ten models

As shown in Fig 6, SGAs perturbing members of a common pathway are closely connected to a similar set of hidden variables, and often many SGAs are directly connected to a common hidden node. This suggests that such hidden nodes encode a common signal that is shared by the SGAs. We hypothesize that if SGAs from a pathway are connected to a similar set of shared hidden nodes in multiple different RINN models, this would indicate that the RINN can repeatedly detect the common impact of the SGAs, and therefore use a common set of hidden nodes to encode their impact with respect to DEGs. In other words, the RINN consistently encodes the shared functional impact of SGAs perturbing a common pathway and the function of these hidden nodes in a specific RINN model become partially transparent. We labeled hidden nodes based on their SGA ancestors (see section “Visualizing a RINN as a causal graph”) and then examined whether such hidden nodes were conserved across models (Table 6). Indeed, many hidden nodes were conserved across the top ten models. More specifically, Table 6 shows the number of models that shared a specific hidden node labeling and whether or not this would be expected by chance. For example, a hidden node mapping to all five members of the PI3K pathway, $\{AKT1, PIK3CA, PIK3R1, PTEN, STK11\}$, was found in all of our top ten models. Given five random SGAs, the expected number of the top ten models to share a hidden node mapped to all five random SGAs was 0.0 models. This means that in our 30 replicates of the random control with five random SGAs, none of the top ten models ever shared a hidden node mapped to all five random SGAs. This also means that the result above for all five members of the PI3K pathway is a very strong result and a definite pathway relationship discovered with the RINN setup in this work. Many of the 4-SGA and 3-SGA labeled hidden nodes were also found in many more models than the corresponding random control, such as the $\{AKT1, PIK3CA, PTEN, STK11\}$, $\{PIK3CA, PIK3R1, PTEN, STK11\}$, $\{AKT1, PIK3CA, STK11\}$, $\{PIK3R1, PTEN, STK11\}$, and $\{PIK3CA, PIK3R1, PTEN\}$ labeled hidden nodes that were found.

A hidden node mapped to both members of the Nrf2 and TGF β pathways was also found in all ten models, which was much higher than the number of models expected given two random SGAs. Given random SGAs, only 0.7 ± 1.2 models of the top ten would be expected to share the same 2-SGA labeled hidden node. Also, a hidden node mapped to all three members of the Notch pathway was shared by nine of the top ten models, which was also well beyond the number of models expected given random SGAs.

Table 6. Cancer pathway hidden nodes shared across top ten models.

Pathway	Hidden Node x Mapping	Models (out of 10; numbered 0 to 9) that Share Hidden Node x	Expected Number of Models that Share an n -SGA Hidden Node Given m Random SGAs
PI3K	AKT1, PIK3CA, PIK3R1, PTEN, STK11	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0.0 ± 0.0 ($n = 5, m = 5$)
	AKT1, PIK3CA, PTEN, STK11	0, 2, 5, 6, 7, 9	0.4 ± 1.4 ($n = 4, m = 5$)
	PIK3CA, PIK3R1, PTEN, STK11	0, 4, 5, 7, 8	0.4 ± 1.4 ($n = 4, m = 5$)
	AKT1, PIK3CA, PIK3R1, PTEN	1, 3, 6	0.4 ± 1.4 ($n = 4, m = 5$)
	AKT1, PIK3R1, PTEN, STK11	2, 3	0.4 ± 1.4 ($n = 4, m = 5$)
	AKT1, PIK3CA, STK11	1, 2, 3, 5, 6, 8	1.0 ± 1.4 ($n = 3, m = 5$)
	PIK3R1, PTEN, STK11	2, 3, 6, 7, 8, 9	1.0 ± 1.4 ($n = 3, m = 5$)
	PIK3CA, PIK3R1, PTEN	0, 2, 6, 7, 9	1.0 ± 1.4 ($n = 3, m = 5$)
	AKT1, PIK3R1, PTEN	3, 9	1.0 ± 1.4 ($n = 3, m = 5$)
	AKT1, PTEN, STK11	2, 3	1.0 ± 1.4 ($n = 3, m = 5$)
	PTEN, STK11	0, 2, 3, 5, 6, 7, 8	3.4 ± 2.0 ($n = 2, m = 5$)
	PIK3R1, PTEN	0, 2, 3, 6, 8, 9	3.4 ± 2.0 ($n = 2, m = 5$)
	AKT1, PIK3CA	0, 3, 4, 6, 8	3.4 ± 2.0 ($n = 2, m = 5$)
	PIK3CA, STK11	0, 1, 2, 4, 8	3.4 ± 2.0 ($n = 2, m = 5$)
	AKT1, STK11	1, 2, 3, 4, 5	3.4 ± 2.0 ($n = 2, m = 5$)
	PIK3CA, PTEN	5, 6, 8, 9	3.4 ± 2.0 ($n = 2, m = 5$)
	AKT1, PIK3R1	2, 5	3.4 ± 2.0 ($n = 2, m = 5$)
	PIK3R1, STK11	3, 9	3.4 ± 2.0 ($n = 2, m = 5$)
	AKT1, PTEN	5, 9	3.4 ± 2.0 ($n = 2, m = 5$)
Nrf2	KEAP1, NFE2L2	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0.7 ± 1.2 ($n = 2, m = 2$)
TGFβ	ACVR2A, SMAD4	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0.7 ± 1.2 ($n = 2, m = 2$)
Notch	EP300, FBXW7, NOTCH1	0, 1, 2, 3, 4, 5, 6, 7, 9	0.2 ± 0.6 ($n = 3, m = 3$)
	FBXW7, NOTCH1	0, 2, 3, 6, 7, 8, 9	2.1 ± 2.2 ($n = 2, m = 3$)
	EP300, FBXW7	0, 2, 3, 4, 6	2.1 ± 2.2 ($n = 2, m = 3$)

Hidden Node x Mapping represents SGAs that directly act on hidden node x or are ancestors of hidden node x . Models are numbered 0 to 9. (n : number of SGAs mapping to a hidden node, m : number of SGAs in a pathway).

Discussion

In this study, we show that deep learning models, RINN and DNN, can capture the statistical relationships between genomic alterations and transcriptomic events in tumor cells with reasonably high accuracy, despite the small number of training cases relative to the high dimensionality of the data. Our findings further indicate that a regularized deep learning model with redundant inputs (i.e., the RINN) can capture cancer signaling pathway relationships within its hidden variables and weights. The RINN model correctly captured much of the functional similarity between SGAs that perturb a common signaling pathway as reflected by the SGAs' similar interactions with the hidden nodes of the RINN model (i.e., cosine similarity of SGA weight signatures). This shows that SGAs on the same pathway share similar interactions (in terms of connection and weights) with a set of latent variables. These are very encouraging results for eventually using a future version of the RINN to find signaling pathways robustly. Many of the most well known cancer driver genes (*EGFR*, *TP53*, *CDKN2A*, *APC*, and *PIK3CA*) were found to have dense SGA weight signatures and weights with larger values relative to the other genes we analyzed, reinforcing the importance of these genes in driving cancer gene expression and the validity of our models. Our results indicate that RINN consistently employs certain hidden nodes to represent the shared functional

impact of SGAs perturbing a common pathway, although different instantiations of RINN could use totally different hidden nodes. The ability of RINN to explicitly connect SGAs and hidden nodes throughout the latent hierarchy essentially makes RINN a partially transparent deep learning model, so that one can interpret which hidden nodes encode and transmit the signals (i.e., functional impact) of SGAs in cancer cells. Finally, we show that RINN is capable of capturing some causal relationships (given our interpretation of the hidden nodes) between the signaling proteins perturbed by SGAs. All these results indicate that by allowing SGAs to directly interact with hidden nodes in a deep learning model, the RINN model provides constraint, information, and flexibility to enable certain hidden nodes to encode the impact of specific SGAs.

Overall, both RINN and DNN are capable of capturing statistical relationships between SGAs and DEGs. However, latent variables in a DNN model (except those directly connected to SGAs) are less interpretable because the latent variable information deeper in the network is more convoluted. In a DNN model, all SGAs have to interact with the first layer of hidden variables and their information is then propagated through the whole hierarchy of the model. In such a model, it is difficult to pinpoint how the signal of each SGA is propagated. Hidden layers in a DNN are alternate representations of all the information in the input necessary to calculate the output, whereas each hidden layer of a RINN does not need to capture all the information in the input necessary to predict the output because there are multiple chances to learn what is needed from the input (i.e., redundant inputs). This difference gives the RINN more freedom in how it chooses to use the information in the input. The redundant inputs of the RINN are an attempt to deconvolute the signal of each SGA by giving the model more freedom to take advantage of the hierarchical structure and choose latent variables at the right level of granularity to encode the signal of an SGA, e.g., early in the network at the first hidden layer or at later layers in the network. This approach is biologically sensible because different SGAs do affect proteins at different levels in the hierarchy of the cellular signaling system. It is expected that an SGA perturbing a transcription factor (e.g., *STAT3*) should impact a relatively small number of genes in comparison to an SGA that perturbs at a high level in the signaling system (e.g., *EGFR*). Refined granularity enables RINN to search for "optimal" structure in order to encode the signaling between SGAs and DEGs while satisfying our sparsity constraints, leading to RINNs with three to four relatively sparsely connected layers of hidden variables; whereas DNNs tend to use two layers of relatively densely connected latent variables.

A DNN cannot capture the same causal relationships that a RINN can. By nature of its architecture and design, a DNN can capture *direct* causal relationships (i.e., edges starting from an observed variable) between the input and the first hidden layer only—DNNs cannot capture direct causal relationships between the input and any other hidden layers. This means that a DNN cannot be used to generate causal graphs like the ones shown in Fig 6. In addition, a DNN cannot capture causal relationships among *input SGAs* as was described in the section "Inferring causal relationships using RINN", meaning that one cannot infer the causal relationships among SGAs with a DNN. This is because DNNs do not have edges between hidden nodes in the same layer (or redundant inputs). Consider $path_{KEAP1}$ and $path_{NFE2L2}$ as the paths with *KEAP1* or *NFE2L2* as the source node, respectively, in a DNN. In a DNN, to determine that there is a dependency between *KEAP1* and *NFE2L2*, eventually these two paths would need to collide on a hidden node. When these two paths collide on a hidden node, the number of edges in each path will be the same, meaning that the direction of the causal relationship is ambiguous. This limitation of a DNN can be remedied by adding redundant inputs (i.e., RINN). Using the RINN architecture allows us to infer order to

the causal relationships among SGAs—this design difference and extension of causal interpretability are what set it apart from a DNN.

It is intriguing to examine further whether the hierarchy of hidden nodes can capture causal relationships among the signals encoded by SGA-affected proteins. We have shown that many of the pathway relationships and some known causal relationships were present in the hierarchy reflected by the weight matrices of our trained models. However, we also noticed that in our RINN models an SGA is often connected to a large number of hidden nodes, which are in turn connected to a large number of other hidden nodes—meaning that the current RINN model learns relatively dense causal graphs. While one can infer the relationships between the signal perturbed by distinct SGAs of a pathway, our current model cannot directly output a causal network that looks like those commonly shown in the literature. We plan to develop RINN into an algorithm that is able to find more easily interpretable cellular signaling pathways when trained on SGA and DEG data. The following algorithm modifications potentially will lead to better results in the future: 1. Incorporating differential regularization of the weights, 2. Using constrained and parallelized versions of evolutionary algorithms to optimize the weights and avoid the need to threshold weights, and 3. Training an autoencoder with a bottleneck layer to encourage hidden nodes to more easily represent biological entities and then using these weights (and architecture) to initialize a RINN.

In order to interpret the weights of a neural network as causal relationships between biological entities, we assume that the causal relationships between biological entities can be approximated by a linear function combined with a simple nonlinear function (e.g., $activation(wx + b)$), where all variables have scalar values and *activation* represents a simple nonlinear function such as ReLU or softplus). This is a necessary assumption in order to interpret all nonzero hidden nodes as biological entities; however, it could also be the case that some hidden nodes are not biological entities, but rather some intermediate calculation required to compute a relationship between biological entities that cannot be modeled with $activation(wx + b)$. Given the high density of the models learned with TCGA data, it is possible that the relationships between some biological entities cannot be modeled with $activation(wx + b)$, suggesting that more complex activation functions are needed or that biological entities may be present at every other hidden layer. It would be interesting to explore using more complex activation functions, and specifically using an unregularized one hidden layer neural network as an activation function for each hidden node in a RINN. This setup would account for even quite complex relationships between biological entities captured as latent variables. See [9] for additional discussion of this topic.

A cellular signaling system is a complex information encoding/processing machine that processes signals arising from extrinsic environment changes or perturbations (genetic or pharmacological) affecting the intrinsic state of a cell. The relationships of cellular signals are hierarchical and nonlinear by nature, and deep learning models are particularly suitable to modeling such a system [16–20]. However, conventional deep learning models behave like “black boxes”, such that it is almost impossible to determine what signal a hidden node encodes, with few exceptions in image analysis where human interpretable image patterns can be represented by hidden nodes [13–15]. Here, we took advantage of our knowledge of cancer biology that SGAs causally influence the transcriptomic programs of cells and we adopted a new approach that allows SGAs to directly interact with hidden nodes in RINN. We conjecture that this approach forces hidden nodes to explicitly and thus more effectively encode the impact of SGAs on transcriptomic systems. This hypothesis is supported by the discoveries of this paper that SGAs in a common pathway share similar connection patterns to hidden nodes, and that there are hidden nodes that are connected to multiple members of a

pathway in different instances of the model. Essentially, our approach also allows certain hidden nodes to be “labeled” and “partially interpretable”. An interpretable deep learning model provides a unique opportunity to study how cellular signals are encoded and perturbed under pathological conditions. Understanding and representing the state of a cellular system further opens directions for translational applications of such information, such as predicting the drug sensitivity of cancer cells based on the states of their signaling systems. To our knowledge, this is the first time that a partially interpretable deep learning model has been developed and applied to study cancer signaling, and we anticipate this approach laying a foundation for developing future explainable deep learning models in this domain.

Acknowledgments

We thank Bryan Andrews and Gregory Cooper for helpful discussions related to this work, and Chunhui Cai for running TCI and curating the data.

References

1. Cooper G. An Overview of the Representation and Discovery of Causal Relationships Using Bayesian Networks in Causation, Prediction, and Search. In: Computation, Causation, and Discovery. Menlo Park, Calif. : AAAI Press ; Cambridge, Mass. : MIT Press; 1999. p. 3–62.
2. Spirtes P, Glymour CN, Scheines R. Causation, prediction, and search. MIT press Cambridge; 2000.
3. Heinze-Deml C, Maathuis MH, Meinshausen N. Causal structure learning. Annual Review of Statistics and Its Application. 2018;5:371–391.
4. Maathuis MH, Nandy P. A Review of Some Recent Advances in Causal Inference. In: Bühlmann P, Drineas P, Kane M, van der Laan M, editors. Handbook of Big Data. Chapman and Hall/CRC, Boca Raton, FL; 2016. p. 387–407.
5. Peters J, Janzing D, Schölkopf B. Elements of causal inference: foundations and learning algorithms. MIT Press; 2017.
6. Lagani V, Triantafillou S, Ball G, Tegner J, Tsamardinos I. Probabilistic computational causal discovery for systems biology. In: Uncertainty in Biology. Springer; 2016. p. 33–73.
7. Murray-Watters A, Glymour C. What is going on inside the arrows? Discovering the hidden springs in causal models. Philosophy of Science. 2015;82(4):556–586.
8. Frot B, Nandy P, Maathuis MH. Robust causal structure learning with some hidden variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2019.
9. Young JD, Andrews B, Cooper GF, Lu X. Learning Latent Causal Structures with a Redundant Input Neural Network. arXiv preprint arXiv:200313135. 2020.
10. Deng L, Yu D, et al. Deep learning: methods and applications. Foundations and Trends in Signal Processing. 2014;7(3–4):197–387.
11. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. MIT press Cambridge; 2016.
12. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436.

13. Lee H, Ekanadham C, Ng AY. Sparse deep belief net model for visual area V2. In: *Advances in Neural Information Processing Systems*; 2008. p. 873–880.
14. Lee H, Grosse R, Ranganath R, Ng AY. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*. 2011;54(10):95–103.
15. Le QV, Ranzato M, Monga R, Devin M, Chen K, Corrado GS, et al. Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:11126209*. 2011.
16. Chen L, Cai C, Chen V, Lu X. Trans-species learning of cellular signaling systems with bimodal deep belief networks. *Bioinformatics*. 2015;31(18):3008–3015.
17. Chen L, Cai C, Chen V, Lu X; BioMed Central. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*. 2016;17(1):S9.
18. Young JD, Cai C, Lu X. Unsupervised deep learning reveals prognostically relevant subtypes of glioblastoma. *BMC Bioinformatics*. 2017;18(11):381.
19. Lu S, Fan X, Chen L, Lu X. A novel method of using Deep Belief Networks and genetic perturbation data to search for yeast signaling pathways. *PloS One*. 2018;13(9):e0203871.
20. Tao Y, Cai C, Cohen WW, Lu X. From genome to phenome: Predicting multiple cancer phenotypes based on somatic genomic alterations via the genomic impact transformer. In: *Pacific Symposium on Biocomputing*. vol. 25; 2020. p. 79–90.
21. Weaver DC, Workman CT, Stormo GD. Modeling regulatory networks with weight matrices. In: *Pacific Symposium on Biocomputing*; 1999. p. 112–123.
22. Vohradsky J. Neural network model of gene expression. *The FASEB Journal*. 2001;15(3):846–854.
23. Keedwell E, Narayanan A, Savic D. Modelling gene regulatory data using artificial neural networks. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. vol. 1. IEEE; 2002. p. 183–188.
24. Mandal S, Saha G, Pal RK. Neural network based gene regulatory network reconstruction. In: *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*. IEEE; 2015. p. 1–5.
25. Ando S, Iba H. Inference of gene regulatory model by genetic algorithms. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*. vol. 1. IEEE; 2001. p. 712–719.
26. Keedwell E, Narayanan A. Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*. 2005;2(3):231–242.
27. Narayanan A, Keedwell EC, Gamalielsson J, Tatineni S. Single-layer artificial neural networks for gene expression analysis. *Neurocomputing*. 2004;61:217–240.
28. Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*. 2013;45(10):1113.

29. Cai C, Cooper G, Lu K, Ma X, Xu S, Zhao Z, et al. Systematic discovery of the functional impact of somatic genome alterations in individual tumors through tumor-specific causal inference. *PLoS Computational Biology*. 2019;15(7):e1007088.
30. Cooper G, Cai C, Lu X. Tumor-specific causal inference (tci): A bayesian method for identifying causative genome alterations within individual tumors. *bioRxiv*. 2018; p. 225631.
31. Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: *Neural Networks: Tricks of the Trade*. Springer; 2012. p. 437–478.
32. Hinton GE. A practical guide to training restricted Boltzmann machines. In: *Neural Networks: Tricks of the Trade*. Springer; 2012. p. 599–619.
33. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.
34. Sanchez-Vega F, Mina M, Armenia J, Chatila WK, Luna A, La KC, et al. Oncogenic signaling pathways in the cancer genome atlas. *Cell*. 2018;173(2):321–337.
35. Bastian M, Heymann S, Jacomy M. Gephi: an open source software for exploring and manipulating networks. In: *Third International AAAI Conference on Weblogs and Social Media*; 2009. p. 361–362.
36. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008;2008(10):P10008.
37. Lambiotte R, Delvenne JC, Barahona M. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:08121770*. 2008.

Supporting information

Fig S1. Genes in SGA Dataset that Overlap with Sanchez-Vega et al., 2018. This is Figure S1 from Sanchez et al., 2018: “Curated Pathways Including Cross-Cross Pathway Interactions” [34]. Circled in cyan are all genes from these pathways that are in our SGA dataset. We are treating these pathways as ground truth for comparison purposes. The cyan circles have been added to the original figure.