




RESEARCH ARTICLE

Inferring tumor progression in large datasets

Mohammadreza Mohaghegh Neyshabouri ^{1,2}, Seong-Hwan Jun ^{1,2},
Jens Lagergren ^{1,2*}**1** Department of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, **2** Science for Life Laboratory, Stockholm, Sweden* jens.lagergren@scilifelab.se

Abstract

Identification of mutations of the genes that give cancer a selective advantage is an important step towards research and clinical objectives. As such, there has been a growing interest in developing methods for identification of driver genes and their temporal order within a single patient (intra-tumor) as well as across a cohort of patients (inter-tumor). In this paper, we develop a probabilistic model for tumor progression, in which the driver genes are clustered into several ordered driver pathways. We develop an efficient inference algorithm that exhibits favorable scalability to the number of genes and samples compared to a previously introduced ILP-based method. Adopting a probabilistic approach also allows principled approaches to model selection and uncertainty quantification. Using a large set of experiments on synthetic datasets, we demonstrate our superior performance compared to the ILP-based method. We also analyze two biological datasets of colorectal and glioblastoma cancers. We emphasize that while the ILP-based method puts many seemingly passenger genes in the driver pathways, our algorithm keeps focused on truly driver genes and outputs more accurate models for cancer progression.

 OPEN ACCESS

Citation: Mohaghegh Neyshabouri M, Jun S-H, Lagergren J (2020) Inferring tumor progression in large datasets. *PLoS Comput Biol* 16(10): e1008183. <https://doi.org/10.1371/journal.pcbi.1008183>

Editor: Teresa M. Przytycka, National Center for Biotechnology Information (NCBI), UNITED STATES

Received: April 29, 2020

Accepted: July 22, 2020

Published: October 9, 2020

Copyright: © 2020 Mohaghegh Neyshabouri et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All analyzed biological data are available on intogen website (<https://www.intogen.org/>).

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement MSCA-ITN-2017-766030 and from the Swedish Foundation for Strategic Research grant BD15-0043.

Author summary

Cancer is a disease caused by the accumulation of somatic mutations in the genome. This process is mainly driven by mutations in certain genes that give the harboring cells some selective advantage. The rather few driver genes are usually masked amongst an abundance of so-called passenger mutations. Identification of the driver genes and the temporal order in which the mutations occur is of great importance towards research and clinical objectives. In this paper, we introduce a probabilistic model for cancer progression and devise an efficient inference algorithm to train the model. We show that our method scales favorably to large datasets and provides superior performance compared to an ILP-based counterpart on a wide set of synthetic data simulations. Our Bayesian approach also allows for systematic model selection and confidence quantification procedures in contrast to the previous non-probabilistic progression models. We also study two large datasets on colorectal and glioblastoma cancers and validate our inferred model in comparison to the ILP-based method.

Competing interests: The authors have declared that no competing interests exist.

This is a *PLOS Computational Biology* Methods paper.

Introduction

Tumor progression is caused by somatic evolution in which genes are randomly mutated and so-called driver mutations confer the tumor a selective advantage [1]. Several properties of somatic evolution of tumors have been studied intensively and are today more approachable than ever before, for instance,

1. What is the number of driver mutations in a tumor or the average across a collection?
2. Which genes acquire driver mutation, i.e., are so-called driver genes, in contrast to passenger genes?
3. What are the dependencies among driver mutations and, in particular, in which order do they occur?

Our main interest is in developing methods to resolve these questions as they have immense potential towards the identification of targets for new drugs and the development of patient-specific treatment plans.

The fundamental difficulty of answering these questions lies in the successful identification of driver mutations amongst an abundance of passenger mutations. Without prior information, there is no way to identify driver mutations in a single tumor. Over-representation of a gene among those mutated across a large tumor collection is, however, a useful driver gene identification approach. This approach is now feasible with the availability of mutation data through several large scale cancer sequencing efforts, e.g., The Cancer Genome Atlas (TCGA) as well as the International Cancer Genome Consortium (ICGC).

Cancer progression has been extensively studied through the application of various models capturing dependencies between mutated genes, such as oncogenetic trees [2–6] and conjunctive Bayesian networks [7, 8]. More complex models are also used to study the cancer progression [9–11]. However, these models are computationally hard to train, and hence, not applicable for big datasets composed of large numbers of samples and genes, and in the abundance of passenger mutations. On the other hand, studying the cancer progression using basic models at the gene level seems to be insufficient to fully model somatic evolution in cancer [12]. The effects of a driver gene mutation are often mediated by a biological pathway or a protein complex that the gene belongs to. Consequently, a set of genes associated with the same biological pathway or protein complex may affect a tumor in the very same way when mutated, and the selective advantage provided to the tumor by one may exhaust that of any other, which would make the genes be mutated in a mutually exclusive manner. These mechanisms can also provide a partial explanation for the observed heterogeneity of cancer mutations. The mutual exclusivity in driver genes has been studied using various models including [13, 14]. These methods have later been extended to the cancer progression context, where the aim is to identify mutually exclusive driver pathways and their temporal ordering, simultaneously. In [15], the objective is to learn a conjunctive Bayesian network with modules of mutually exclusive genes at the nodes of the network. However, the training procedure is a heuristic algorithm that suffers from computational complexity and scalability issues. In [16], the aim is to find a set of linearly ordered mutually exclusive driver pathways using integer linear programming (ILP). This method can potentially be applied in the presence of passenger mutations and on datasets composed of tens of genes and hundreds of patients.

In this paper, we develop a probabilistic model of mutually exclusive linearly ordered driver pathways. We design a sampling based inference algorithm to train our model. Using an extensive set of experiments we demonstrate our method’s superior performance and scalability to large datasets in comparison to the ILP-based algorithm in [16]. We also analyze two biological datasets on colorectal adenocarcinoma and glioblastoma and demonstrate our superior performance on these datasets in comparison to the ILP-based counterpart [16].

Linear progression model

In this section, we start by illustrating linear pathway progression in cancer using the example model in Fig 1. We then introduce our probabilistic model for this process for which we subsequently develop sampling-based inference algorithms in the next section.

An illustration of the model

A linear pathway progression model of a specific cancer type (or sub-type) is defined as an ordered set of several sets of driver genes. We call these sets of genes *driver pathways*. We refer to the set of genes not included in the driver pathways as the *set of passengers*. Based on this model, cancer starts with a mutation in one of the genes in the first driver pathway. This mutation provides the harboring cells with some selective advantages and the tumor progresses to stage 1. As time goes on, the tumor may progress to stage 2 by acquiring some mutation in one of the genes in the second pathway adding more selective advantage to the tumor. The tumor can continue its progression further in the same way.

Our goal is to infer the driver pathways using a set of binary vectors showing the observed mutation status of a set of genes in various tumors. However, this task is not as simple as it seems to be. The vast majority of genes (belonging to the set of passengers) do not play an

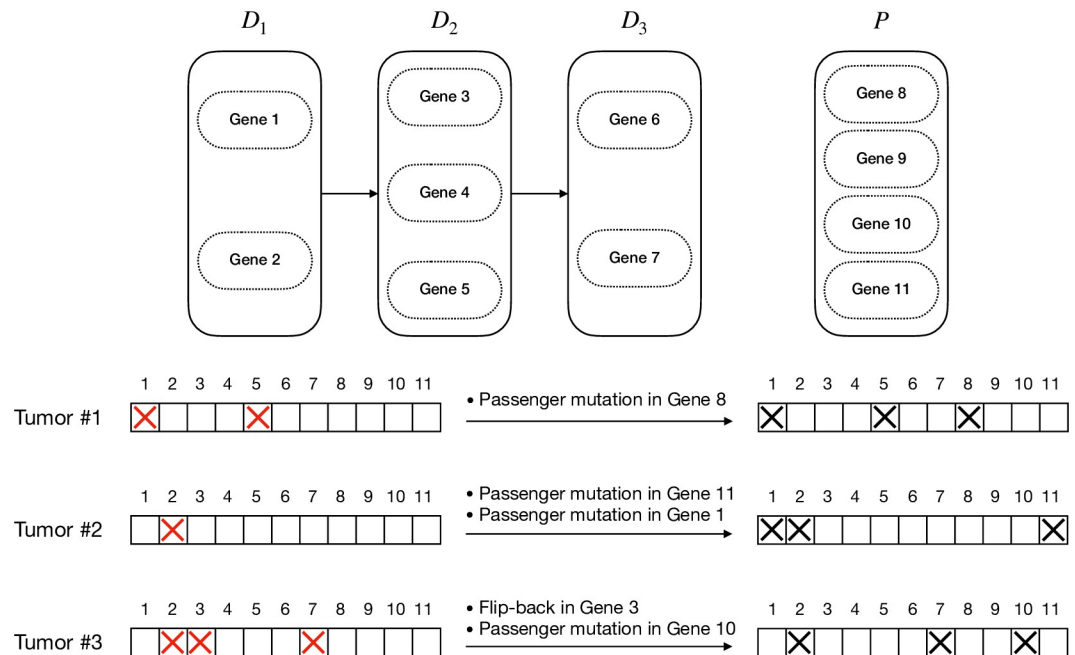


Fig 1. An example of a linear pathway progression model and three tumors generated following the model. The left-hand side gene status bars show the underlying mutational status of the genes with the red cross signs showing the driver mutations. The right-hand side gene status bars show the observed mutation status of genes, where passenger mutations and flip-back events make it hard to infer the true progression model from the observations.

<https://doi.org/10.1371/journal.pcbi.1008183.g001>

important role in the progression of the tumor. Hence, even though they may get mutated, these mutations are passenger mutations. Moreover, according to this model, if a pathway is already mutated due to a mutation in one of its constituting genes, no mutation in the other genes in the pathway can give the cells any more selective advantage. Therefore, such mutations will be considered as passenger mutations as well. Working with bulk data, we have a preprocessing step outputting the binary mutation status for each gene (1 if mutated, 0 otherwise). As a result, some actual driver mutations may get lost because of their small cellular prevalence for instance. We refer to these kinds of errors as flip-back events. For example, consider the progression model in Fig 1. As shown in the left-hand side mutation status vectors, tumors 1, 2, and 3 have the cancer stages of 2, 1, and 3, respectively. However, the passenger mutations and flip-back events lead to our observations shown in the right-hand side mutation status vectors.

Notation

Consider a set of N genes indexed from 1 to N . Let $\mathcal{P} = (D_1, D_2, \dots, D_L, P)$ be an ordered partition of the set of indices $\{1, \dots, N\}$. If each gene index is assigned to exactly one of the sets in \mathcal{P} and D_1 to D_L are not empty, then \mathcal{P} is a *linear pathway progression model* of length L . We refer to D_1, \dots, D_L as the driver pathways and P as the set of passengers.

The observed data consist of a mutation matrix $Y \in \{0, 1\}^{M \times N}$ for M tumors, from equally many patients. We denote the m -th row of this matrix by Y_m . This is a binary vector of length N , representing the mutation status of all genes (mutated/normal) in the m -th tumor with $Y_{m,g} = 1$ indicating that the gene g is mutated and $Y_{m,g} = 0$ otherwise. To allow reference to the status of only a subset of genes $S \subset \{1, \dots, N\}$ in the m -th tumor, we introduce a similar notation $Y_{m,S}$. We use $Y_m^* \in \{0, 1\}^N$ to denote a latent (noise free) gene status vector, where $Y_{m,g}^* = 1$ indicates that gene g is a driver mutation for the m -th sample. Finally, we model the biological noises that arise during DNA sequencing and data processing using two parameters $\delta, \epsilon \in [0, 1]$. Specifically, ϵ denotes probability of a passenger mutation (i.e., false positive in the point of view of recovering the driver mutations) and δ denotes flip-back probability which models error sources such as dropout during sequencing (i.e., false negative).

Probabilistic generative process

A graphical model representation of our generative probabilistic model is shown in Fig 2. We assume that given the pathway progression model, the tumors are independent. Hence, it suffices to describe the generative model for a single tumor $Y_m, m \in \{1, \dots, M\}$.

First, the latent progression stage $\sigma_m \in \{1, \dots, L\}$ is sampled from a categorical distribution with parameter $\alpha = (\alpha_1, \dots, \alpha_L)$, i.e., $p(\sigma_m = l) = \alpha_l$ for $l \in \{1, \dots, L\}$. We use a fixed $\alpha = (1/L, \dots, 1/L)$. However, the parameter α can be arbitrarily chosen based on domain knowledge. We can even straightforwardly extend the graphical model to have a prior distribution on α , say a Dirichlet distribution, and infer the posterior belief on alpha given the data. For each $l \in \{1, \dots, \sigma_m\}$, exactly one gene $g \in D_l$ is mutated to construct $Y_m^* \in \{0, 1\}^N$. This procedure ensures mutual exclusivity of the genes: Y_{m,D_l}^* is a one-hot binary vector of dimension $|D_l|$ for $l \in \{1, \dots, \sigma_m\}$ and a zero vector everywhere else. We refer to the subsequent sub-process acting on Y_m^* as corruption, since without this process, the driver mutations would be easily read off from the tumor. When generating the observed data Y_m given Y_m^* , the ones may be flipped back to zero with probability δ . Alternatively, the zeros may turn into ones in our observed mutation status vectors with probability ϵ . The latter partly models technical problems such as mapping, misalignment and so on, but in particular that in a tumor any gene may acquire

Algorithm 1 Generative process

```

1: Input:  $\alpha, \mathcal{P}, \epsilon, \delta$ 
2: Output:  $Y$ 
3: for  $m \in \{1, \dots, M\}$  do
4:   Let  $Y_m$  be a vector of  $N$  zeros
5:   Draw  $\sigma_m \sim \text{Categorical}(\alpha)$ 
6:   for  $k \in \{1, \dots, \sigma_m\}$  do
7:     Draw  $g_k$  (uniformly) from  $D_k$ 
8:     Set  $Y_m[g_k] = 1$ 
9:   for  $n = 1$  to  $N$  do
10:    if  $Y_m[n] == 0$  then
11:      Set  $Y_m[n] = 1$  with prob.  $\epsilon$ 
12:    else
13:      Set  $Y_m[n] = 0$  with prob.  $\delta$ 

```

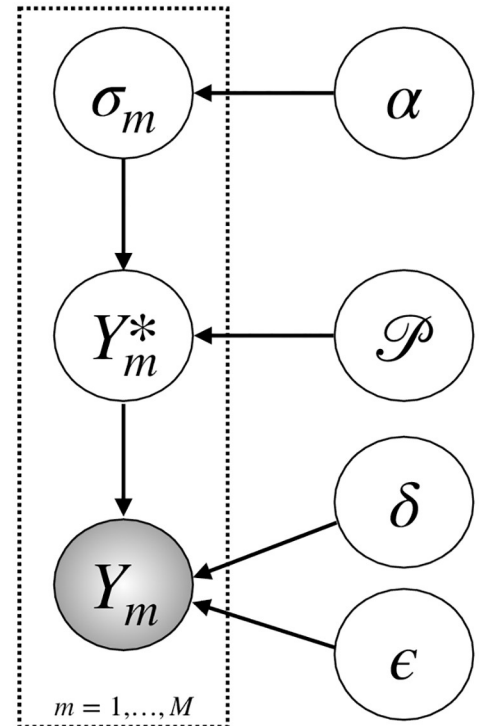


Fig 2. Generative process and the underlying graphical model for the observed mutation matrix.

<https://doi.org/10.1371/journal.pcbi.1008183.g002>

passenger mutations. In particular, even a so-called driver gene that belongs to a driver pathway in which another gene has been previously mutated may acquire a passenger mutation. Here, since the previous mutation has already affected the corresponding biological pathway, the mutation in the second gene does not confer any selective advantage to the tumor, and in this sense, the second mutation is a passenger mutation. The graphical model for the generative process can be found in Fig 2. The procedure for generating synthetic data, i.e., mutated tumor genomes based on our generative model, is also provided in Fig 2.

Methods

We desire to estimate the posterior probability distribution for a given collection of tumors as well as to perform model selection, i.e., determine the number of driver pathways L by computing the marginal likelihood given L . An algorithm for computing the likelihood constitutes a crucial part of both these tasks.

Computing the likelihood

Algorithm 2 Calculation of the likelihood $p(Y|\mathcal{P}, \alpha, \epsilon, \delta)$

```

1: for all  $m \in \{1, \dots, M\}$  do
2:   for  $\sigma_m \in \{1, \dots, L\}$  do
3:      $R \leftarrow 1$ 
4:     for  $S \in \{D_1, \dots, D_L, P\}$  do
5:        $r = \|Y_{m,S}\|_1$ 
6:       if  $S \in \{D_1, \dots, D_{\sigma_m}\}$  then
7:          $A = \frac{r}{|S|} (1 - \delta) \epsilon^{r-1} (1 - \epsilon)^{|S|-r} + \frac{|S|-r}{|S|} \delta \epsilon^r (1 - \epsilon)^{|S|-r-1}$ 
8:       else

```

▷ Calculate $p(Y_m|\mathcal{P}, \alpha, \epsilon, \delta)$
 ▷ Calculate $p(Y_m, \sigma_m|\mathcal{P}, \alpha, \epsilon, \delta)$

9: $A = \epsilon^r (1 - \epsilon)^{|S| - r}$
 10: $R \leftarrow R^* A$
 11: $p(Y_m | \mathcal{P}, \sigma_m, \epsilon, \delta) = R$
 12: $p(Y_m, \sigma_m | \mathcal{P}, \alpha, \epsilon, \delta) = p(\sigma_m | \alpha) p(Y_m | \mathcal{P}, \sigma_m, \epsilon, \delta)$
 13: $p(Y_m | \mathcal{P}, \alpha, \epsilon, \delta) = \sum_{\sigma_m=1}^L p(Y_m, \sigma_m | \mathcal{P}, \alpha, \epsilon, \delta)$
 14: $p(Y | \mathcal{P}, \alpha, \epsilon, \delta) = \prod_{m=1}^M p(Y_m | \mathcal{P}, \alpha, \epsilon, \delta)$

Let $Y = \{Y_m: m \in \{1, \dots, M\}\}$ be a mutation matrix for a collection of tumors, $\mathcal{P} = (D_1, D_2, \dots, D_L, P)$ be a pathway progression model, and α be our prior distribution for the progression stages of the tumors. Denoting the bits in Y_m corresponding to pathway S by $Y_{m,S}$, we can calculate $p(Y | \mathcal{P}, \alpha, \epsilon, \delta)$ using Algorithm 2. The derivation steps of the algorithm and a faster implementation using look-up tables are presented in [S1 Text](#).

Markov Chain Monte Carlo algorithm to train the model

In this section, we describe an MCMC algorithm to generate samples from the posterior distribution of the latent quantities given the observations Y for a fixed model length L :

$\pi(\mathcal{P}, \epsilon, \delta | Y, L, \alpha)$. We apply Gibbs steps to sample the progression model and the error parameters iteratively for T MCMC iterations:

- Initialize $\mathcal{P}^0, \epsilon^0, \delta^0$.
- For $t = 1, \dots, T$:
 - Sample $\mathcal{P}^t \sim \pi(\mathcal{P} | L, \epsilon^{t-1}, \delta^{t-1}, Y, \alpha)$,
 - Sample $\epsilon^t \sim \pi(\epsilon | \mathcal{P}^t, \delta^{t-1}, Y, \alpha)$,
 - Sample $\delta^t \sim \pi(\delta | \mathcal{P}^t, \epsilon^t, Y, \alpha)$.

As sampling directly from the conditional posteriors is challenging, we use the Metropolis-Hastings algorithm to sample each of these latent variables, which renders our algorithm as a type of *Metropolis-within-Gibbs sampler* (see e.g., chapter 10.3.3 of [17]).

Sampling the progression model. Considering a uniform prior distribution for the progression structure given the model length L (in the space of all progression models of length L), we have:

$$\pi(\mathcal{P} | \epsilon, \delta, Y, L, \alpha) = \frac{p(Y | \epsilon, \delta, \mathcal{P}, \alpha) p(\mathcal{P} | L)}{p(Y | \epsilon, \delta, L, \alpha)} \propto p(Y | \epsilon, \delta, \mathcal{P}, \alpha) \tag{1}$$

We use a Metropolis-Hasting sampler to generate progression model samples from this distribution. Our proposal function consists of two types of moves that we call *gene-move* and *pathway-swap* (see [Fig 3](#)). We choose the type of move randomly using a Bernoulli distribution. For the case of *gene-move*, we select a gene uniformly at random and move it to a driver pathway or the set of passengers with a uniform distribution. For the case of *pathway-swap*, we select two driver pathways uniformly at random and swap their positions in the progression structure. According to our experiences, these types of moves can help the model to get out of locally optimal points, where the pathways are placed in a non-optimal order and the mutual exclusivity and the progression conditions are acting against each other, preventing the algorithm from moving towards more likely progression structures by moving one gene at a time.

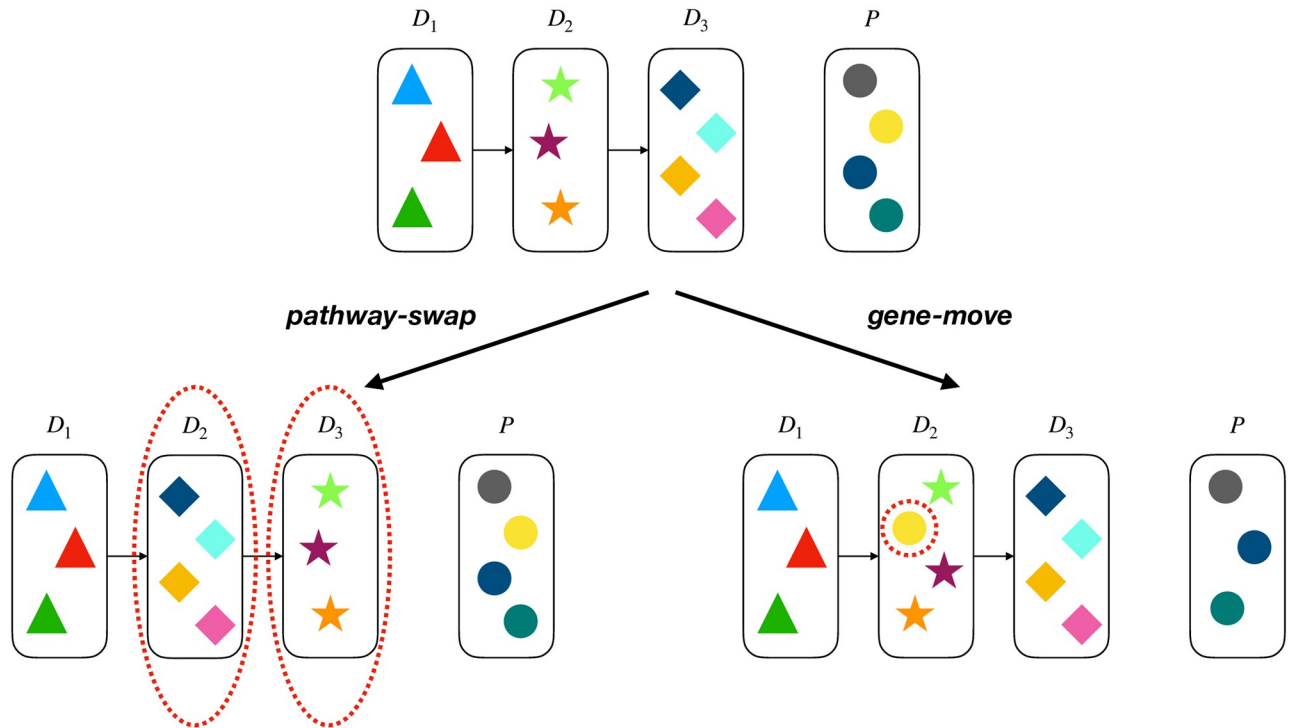


Fig 3. Two types of moves we use for progression model proposal.

<https://doi.org/10.1371/journal.pcbi.1008183.g003>

The acceptance ratio is given by,

$$a(\mathcal{P}^*|\mathcal{P}^t) = \min \left\{ 1, \frac{\pi(\mathcal{P}^*|\epsilon, \delta, Y, L, \alpha)}{\pi(\mathcal{P}^t|\epsilon, \delta, Y, L, \alpha)} \times \frac{q(\mathcal{P}^t|\mathcal{P}^*)}{q(\mathcal{P}^*|\mathcal{P}^t)} \right\} = \min \left\{ 1, \frac{p(Y|\epsilon, \delta, \mathcal{P}^*, \alpha)}{p(Y|\epsilon, \delta, \mathcal{P}^t, \alpha)} \right\}.$$

Sampling the error parameters. The conditional distribution for ϵ is given by,

$$\pi(\epsilon|\mathcal{P}, Y, \alpha, \delta) = \frac{p(Y|\mathcal{P}, \alpha, \epsilon, \delta)p(\epsilon)}{p(Y|\mathcal{P}, \alpha, \delta)} \propto p(Y|\mathcal{P}, \alpha, \epsilon, \delta)p(\epsilon). \tag{2}$$

We use Gaussian random walk proposal, that is, we sample $\epsilon^* \sim \text{Normal}(\epsilon^t, \sigma_\epsilon^2)$. This leads to the following as the acceptance ratio,

$$a(\epsilon^*|\epsilon^t) = \min \left\{ 1, \frac{p(Y|\mathcal{P}, \alpha, \epsilon^*, \delta)}{p(Y|\mathcal{P}, \alpha, \epsilon^t, \delta)} \right\}.$$

We use a similar Metropolis-Hasting sampler to sample our flip-back probability parameter δ . The variance parameters are chosen to be small, for example we use $\sigma_\delta^2 = \sigma_\epsilon^2 = 0.05$ in our experiments. These variance parameters can also be adaptively selected as outlined in [18].

Model selection

The MCMC sampler can be utilized for model selection. Model selection for pathway progression model involves finding a suitable value L for the number of pathways, called model length. We assume that we have a set \mathcal{L} of candidate model lengths, and that we are interested in computing the posterior probability of a model length $L \in \mathcal{L}$ given the observation. Assuming a

uniform prior on the model length, it suffices to compute the *model evidence* $p(Y|L)$. One approach to estimate this quantity is to consider

$$\mathbb{E}_{p(\mathcal{P}, \epsilon, \delta | Y, L)} \left[\frac{1}{p(Y|\mathcal{P}, \epsilon, \delta, L)p(\epsilon, \delta)p(\mathcal{P}|L)} \right] = \frac{1}{p(Y|L)}, \quad (3)$$

where the expectation can be estimated using the MCMC samples, leading to an estimate for $p(Y|L)$. More details on the model selection procedure can be found in [S2 Text](#).

Synthetic data simulations

In this section, we use an extensive set of experiments on synthetic datasets to demonstrate the accuracy and efficiency of our method and, in particular, its superior performance compared to the earlier ILP-based approach [16]. For the synthetic data simulations, having the generative model \mathcal{P} , we calculate a performance metric called *POCO* (Percentage Of Correct Ordering of genes). To this end, considering an inferred model $\tilde{\mathcal{P}}$, we go over all pairs of genes and for each pair, we check if their position with respect to each other (gene 1 before gene 2/gene 1 after gene 2/two genes in the same pathway) in \mathcal{P} is preserved in $\tilde{\mathcal{P}}$. *POCO* is the percentage of the gene pairs with their relative position preserved. A more detailed discussion on this performance metric can be found in [S3 Text](#).

Experiment 1: Known driver genes

In this experiment, we have generated a set of synthetic datasets with fixed flip-back probability $\delta = 0.3$, and various back-ground mutation rate ϵ and number of patients. We have distributed 25 genes in 5 driver pathways with 4 scenarios for the pathway sizes, i.e., number of genes in the pathways:

- *uniform*, where the sizes are (5, 5, 5, 5, 5),
- *increasing-by-2*, where the pathway sizes are (1, 3, 5, 7, 9)
- *decreasing-by-2*, where the pathway sizes are (9, 7, 5, 3, 1)
- *random*, where we randomly order the genes in a row and put 4 separating borders uniformly at random in 24 possible spots between genes.

[Fig 4](#) shows the experiment results. As shown in this figure, while the ILP has difficulties with handling a large number of patients (leading to drop in its performance for large datasets due to inability to converge), our method effectively takes advantage of the statistical power arising from the increasing number of patients to improve its performance. This figure also shows the robust performance of our algorithm in all background mutation rates and pathway size scenarios compared to the ILP algorithm. Moreover, our MCMC samples can be used to estimate the error parameters. A comprehensive analysis of our error estimate performance can be found in [S3 Text](#).

Experiment 2: Unknown driver genes

In this experiment, we demonstrate the effect of adding passengers to the pool of genes. To this end, we generated datasets with 5, 25 and 100 passenger genes. For each case, we have constructed 10 datasets with 500 patients, flip-back probability of 0.3, background mutation rate of 0.05 and 25 genes uniformly distributed in 5 driver pathways.

While our MCMC algorithm does not require any information on the error parameter, the ILP algorithm requires the background mutation rate as an input to the algorithm, if we want

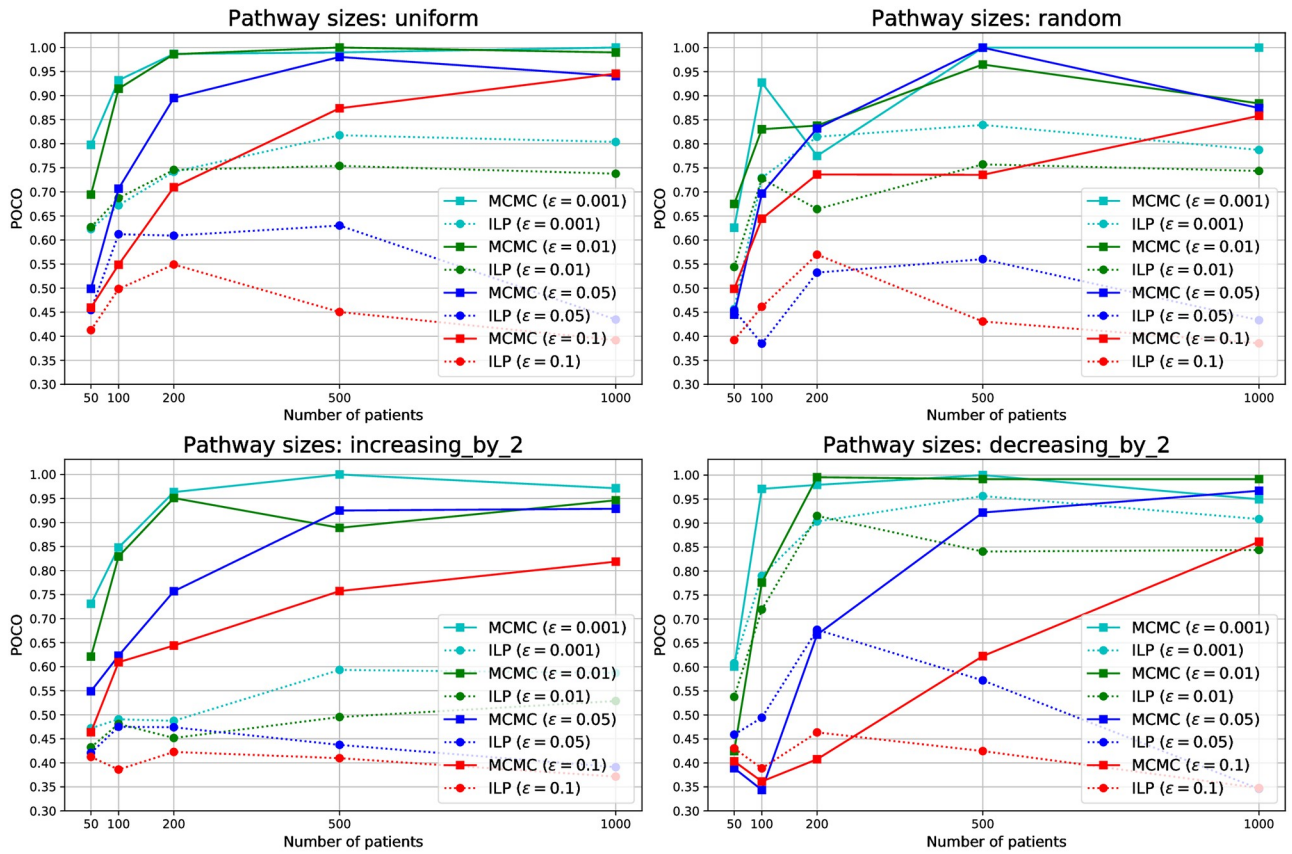


Fig 4. POCO, averaged over 10 datasets, for various ϵ values and pathway size scenarios. The flip-back probability is set to $\delta = 0.3$ in all the datasets. <https://doi.org/10.1371/journal.pcbi.1008183.g004>

to allow for assigning genes to the set of passengers. We have used 3 versions of the ILP algorithm with background mutation rates of 0.01, 0.05 and 0.1, to investigate the likely disruptive effect of incorrect background mutation rate input on the ILP method. We emphasize that the optimal background mutation rate parameter is not typically known in biological data. Hence, having a strong dependency on this parameter is a significant drawback for the ILP-based method.

In Fig 5, we have shown the performance of the competitors using POCO measure, driver detection F1 score, and specific pathway detection F1 score. As shown in this table, the ILP given the true generative background mutation rate of 0.05 performs best among the ILPs, as expected. However, our algorithm significantly outperforms all the ILP competitors, even the one with the extra knowledge of the true generative error parameter. More detailed results on the detection of the genes in specific pathways can be found in S3 Text.

Experiment 3: Model length selection

In this experiment, we demonstrate our model length selection performance. To this end, we have generated datasets using model length parameters from 2 to 9 with 50, 100, 200, 500 and 1000 patients, 25 driver genes uniformly distributed in the driver pathways, 175 passenger genes, background mutation rate of 0.01, and flip-back probability of 0.3. We have constructed 10 datasets using each model length and used our method with model length candidates from 2 to 20.

Method	POCO			F1 score (driver detection)			F1 score (pathway detection)		
	5 passengers	25 passengers	100 passengers	5 passengers	25 passengers	100 passengers	5 passengers	25 passengers	100 passengers
MCMC	0.942	0.936	0.951	0.974	0.960	0.935	0.916	0.877	0.868
ILP($\epsilon = 0.01$)	0.568	0.423	0.285	0.909	0.667	0.343	0.392	0.238	0.086
ILP($\epsilon = 0.05$)	0.772	0.808	0.856	0.882	0.864	0.779	0.389	0.379	0.304
ILP($\epsilon = 0.1$)	0.551	0.494	0.713	0.493	0.428	0.400	0.380	0.333	0.276

Fig 5. POCO, F1 scores for driver detection and F1 scores for specific pathway detection (averaged over pathway 1 to 5) in Experiment 2.

<https://doi.org/10.1371/journal.pcbi.1008183.g005>

Fig 6 shows the confusion matrices for various numbers of patients. As shown in this figure our performance gets better as the number of patients in the dataset increases. However, we emphasize that as shown in the POCO tables in the second column of Fig 6, even for the cases that we have not correctly identified the model length, the genes ordering with respect to each other is recovered up to a significant level, which is of great importance. Consider the case of 100 patients, for instance. We can see from the corresponding confusion matrix in the first column of Fig 6 that the element (6, 2) equals 4. This means that in 4 simulations with the generative model length of 6, our algorithm has mistakenly inferred the model length equal to 2. However, looking at the corresponding element in the averaged poco matrix (in the second column of Fig 6), we see that in these 4 simulations, our inferred model has achieved a POCO score of 0.88 in average, which seems a quite satisfying result for such a small number of patients.

Biological data analysis

We analyzed two large biological datasets of colorectal adenocarcinoma (COADREAD) and glioblastoma multiforme (GBM) from IntOGen-mutations [19] and compared our algorithm against our implementation of the ILP-based method in [16]. To this end, we first filtered out all the silent mutations as a preprocessing step. Alongside the datasets, IntoGen has published a list of potential driver genes for each cancer type. We call the genes in these lists the *potential driver genes* and use only those genes in our input matrices for both our algorithm and the ILP-based competitor. These input matrices can be found in Figs 7A and 8A. In these matrices, the rows and columns represent the genes and the tumors, respectively. A black rectangle in position (i, j) means that tumor j has a mutation in gene i . After the preprocessing steps, we have 9465 genes and 290 patients with 24 genes in the list of potential driver genes for GBM, and 9169 genes and 193 patients with 37 potential driver genes for COADREAD.

For each dataset, we performed model selection routines with candidate lengths from 2 to 30 and chose the best model length. The output models for both our algorithm and the ILP-based method are shown in Figs 7 and 8. The MCMC output models shown in these figures are consensus-like models, constructed using our samples from the posterior distributions of the progression models. A pseudo-code description of the algorithm we used to construct our consensus-like output models is provided in S4 Text. This algorithm takes the MCMC model samples as input and outputs a progression model that can be considered as *the average model*. We emphasize that in our analyses, the consensus-like models were pretty close to the MCMC samples with the highest likelihoods in both datasets.

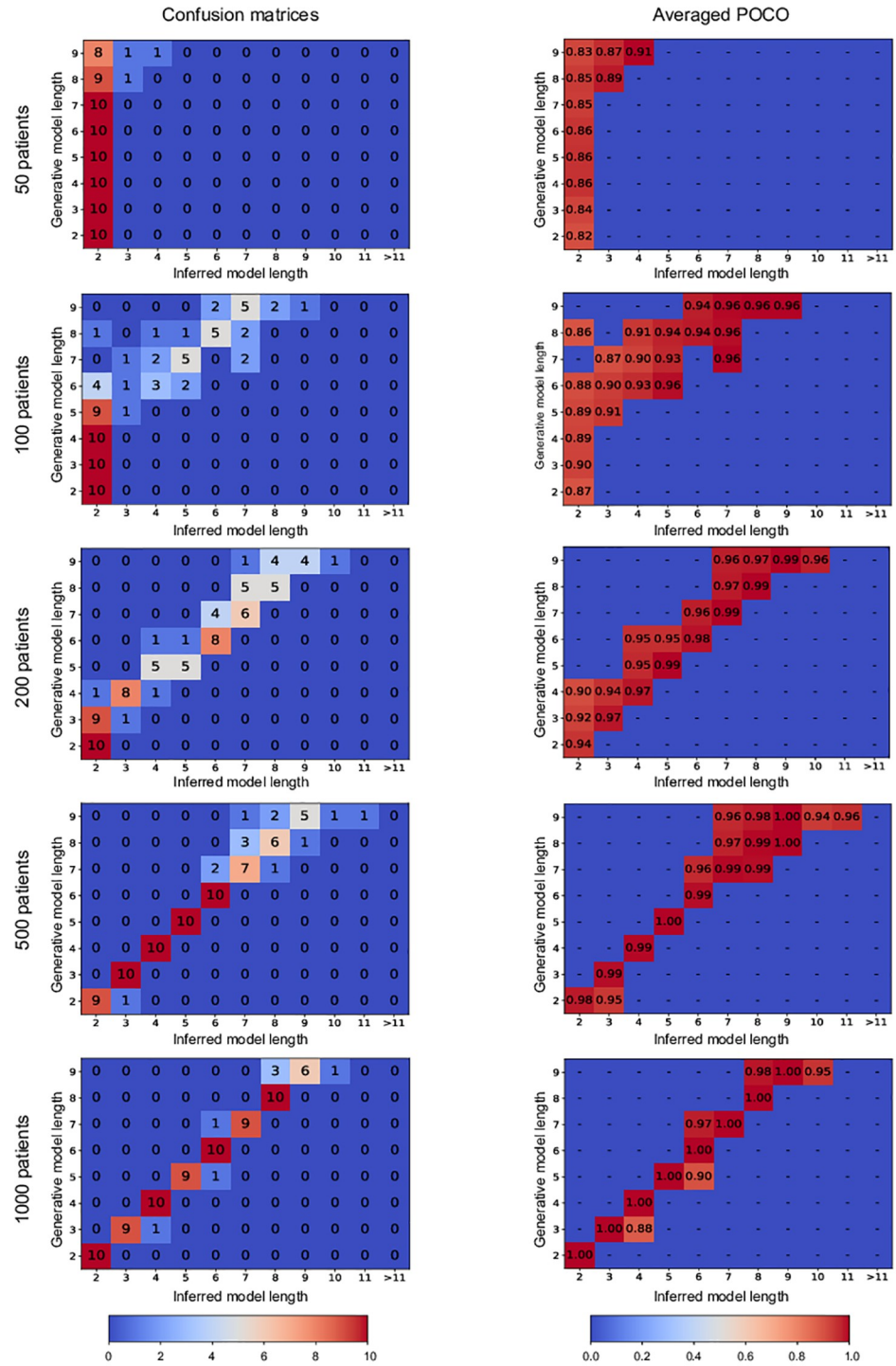


Fig 6. Confusion matrices and averaged POCO scores of our MCMC algorithm in Experiment 3. The matrices in the first column show the confusion matrices for simulations with various number of patients. The element (i, j) in each one of the matrices in the first column shows the number of experiments in which the generative model length was i and the inferred model length was j . The matrices in the second column show the averaged POCO scores of the inferred models in each scenario. Here, the element (i, j) shows the average POCO score for the cases in which the generative model length was i and the inferred model length was j .

<https://doi.org/10.1371/journal.pcbi.1008183.g006>

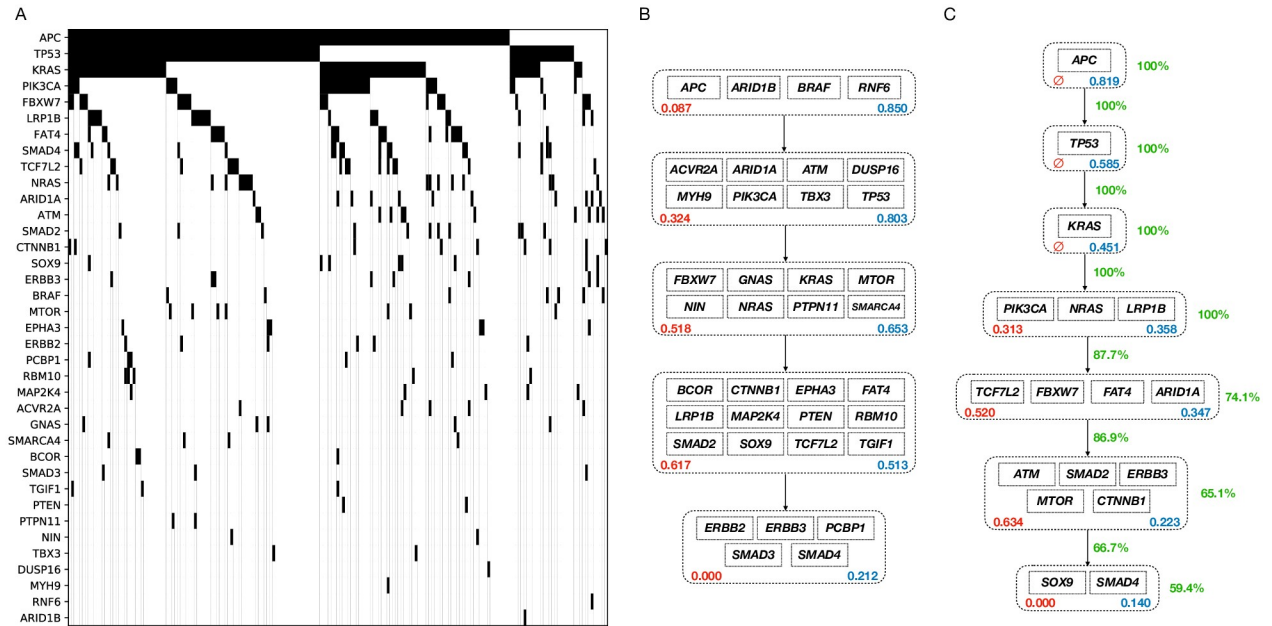


Fig 7. The COADREAD dataset analysis. A: The dataset representation. The genes are sorted based on their mutation frequencies. B: ILP-based method inferred model. C: Our MCMC method inferred model.

<https://doi.org/10.1371/journal.pcbi.1008183.g007>

After we constructed the average model, we calculate our confidence on the pathways and their respective position in the progression model as follows. For each pathway, we go over all pairs of genes in the pathway. Our confidence in the pathway is the averaged percentage of our MCMC samples with these pairs of genes in the same pathway. To calculate our confidence on

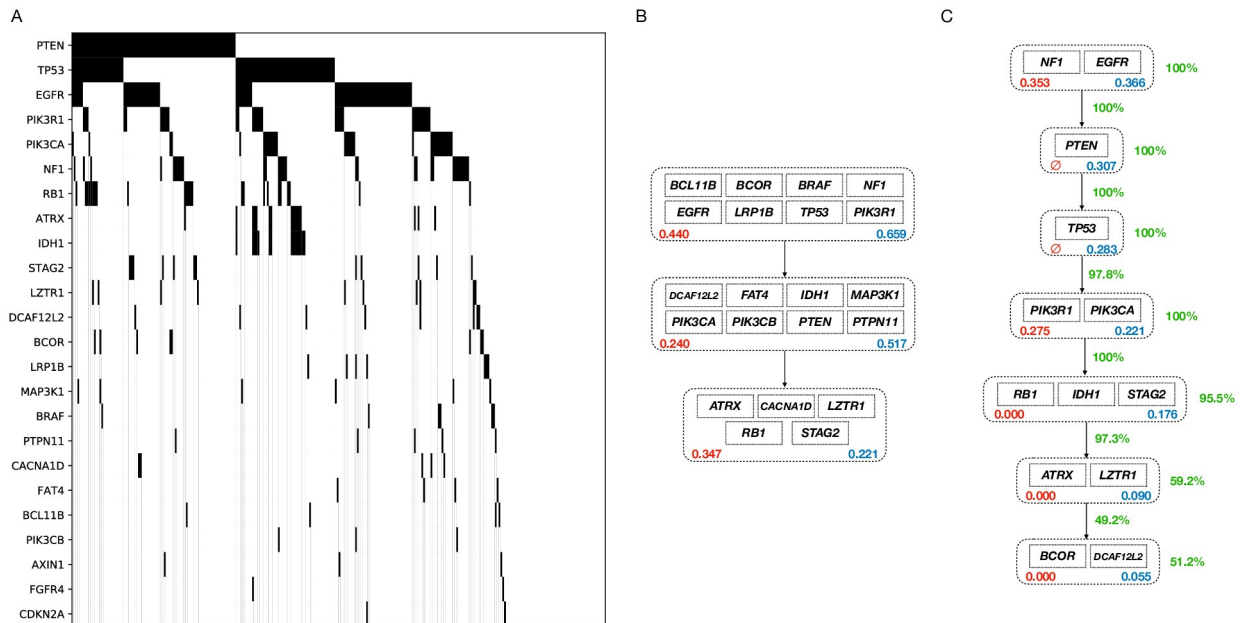


Fig 8. The GBM dataset analysis. A: The dataset representation. The genes are sorted based on their mutation frequencies. B: ILP-based method inferred model. C: Our MCMC method inferred model.

<https://doi.org/10.1371/journal.pcbi.1008183.g008>

an arrow connecting pathway 1 to pathway 2, we go over all pairs of genes ($gene_1$, $gene_2$) with $gene_1$ in pathway 1 and $gene_2$ in pathway 2. Our confidence on the arrow is the percentage of our MCMC samples that have put $gene_1$ before $gene_2$. We have shown these confidence metrics using green numbers beside the arrows and pathways in the depicted models.

To assess the performance of our algorithm against the ILP-based competitor, we define a metric named ME (Mutual Exclusivity) score for each pair of genes, denoted by S_{ME} :

$$S_{ME}(g_i, g_j) = S_{ME}(g_j, g_i) = \frac{\text{mutation rate of } g_i \text{ in patients with mutated } g_j}{\text{mutation rate of } g_i \text{ in all patients}} \quad (4)$$

As the average ME score of the pairs of genes in the identified pathways is smaller, we have a higher level of mutual exclusivity in the pathway, and hence a better model. We have shown the ME scores of the pathways identified by both algorithms using red numbers inside the depicted pathways in Figs 7 and 8. Moreover, the fraction of the tumors having at least one mutation in each pathway is shown using the blue numbers inside the pathways. The first driver pathway having the highest fraction of mutations and a decreasing trend afterward implies that the progression condition holds in the inferred model.

As shown in Fig 7, our algorithm results in a progression model of length 7 for the COAD-READ dataset. The first pathway includes *APC*, a tumor suppressor gene. Mutation and inactivation of this gene is known to be an early event playing a key role in colorectal cancer tumorigenesis [20]. The second pathway includes *TP53*, another tumor suppressor, which is highly mutated in colorectal cancer. Mutant p53 is also shown to have an oncogenetic role in colorectal cancer through gain-of-function mechanisms [21]. The third pathway includes *KRAS*. Mutated *KRAS* is known to be highly associated with colorectal cancer [22]. *KRAS* mutation in colorectal cancer is also known to be a subsequent event after a mutation in *APC* [23], a temporal order of mutations that is recovered by our algorithm as well.

As shown in Fig 8, our algorithm inferred a progression model of length 7 for the GBM dataset. The first pathway includes *EGFR* and *NF1*, two genes known to be the main drivers of the *classical* [24] and *Mesenchymal* [25] subtypes, respectively. *PTEN* in the second pathway is a well-known tumor suppressor known to be involved in regulation of glioblastoma oncogenesis [26]. The third pathway includes *TP53*, another well-known tumor suppressor gene, which is known to play an important role in various cancer types. In glioblastoma in particular, the *p53-ARF-MDM2* pathway is reported to be deregulated in 84% of the patients and 94% of the cell lines [27]. The fourth pathway includes two class IA PI3K subunit genes *PIK3CA* and *PIK3R1*. Mutations in *PI3K* survival cascade is known to be highly associated with glioblastoma [28]. The genes in the fifth pathway *IDH1*, *RB1* and *STAG2* are also known to be associated with the cancer progression in brain glioblastoma [29, 30, 31].

Comparing our results against the ILP-based method, we see that the ILP-based method puts a lot of genes with low mutation rates in the driver pathways, while our algorithm keeps its focus on the highly mutated genes. This happens since the ILP is trying to minimize the number of bits that are needed to get flipped to make the dataset perfectly following the errorless linear progression model. Hence, the ILP tends to put some not necessarily driver genes with low mutation rates into the driver pathways, as it does not affect the ILP cost that much. As an extreme example, if a gene is not mutated in the dataset, the model cost does not depend on where the ILP puts the gene. Our algorithm on the other hand implicitly considers equal importance for the genes in a pathway, which prevents us from putting less highly mutated genes in the driver pathways. Moreover, while our MCMC samples provide us with proper confidence metrics on the inferred models, the ILP-based method has to use bootstrapping techniques and re-run the algorithm over and over to provide some confidence metrics.

Discussion

We investigated the progression patterns in cancer. To this end, we developed a probabilistic model that tries to capture the patterns of progression and mutual exclusivity among the genes involved in cancer. We designed an efficient MCMC algorithm to make inferences on the progression model. We demonstrated the superior performance of our algorithm compared to a previously introduced ILP-based method on a wide set of synthetic data simulations. We also analyzed two biological datasets on colorectal cancer and glioblastoma and showed that our inferred progression models can be better validated compared to the models suggested by the ILP-based counterpart.

Supporting information

S1 Text. Likelihood calculation details.

(PDF)

S2 Text. Model selection details.

(PDF)

S3 Text. Details of synthetic data simulations.

(PDF)

S4 Text. Details of biological data analysis.

(PDF)

Author Contributions

Conceptualization: Jens Lagergren.

Data curation: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun.

Formal analysis: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun, Jens Lagergren.

Funding acquisition: Jens Lagergren.

Investigation: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun, Jens Lagergren.

Methodology: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun, Jens Lagergren.

Project administration: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun.

Resources: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun.

Software: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun.

Supervision: Jens Lagergren.

Validation: Mohammadreza Mohaghegh Neyshabouri.

Visualization: Mohammadreza Mohaghegh Neyshabouri.

Writing original draft: Mohammadreza Mohaghegh Neyshabouri.

Writing review & editing: Mohammadreza Mohaghegh Neyshabouri, Seong-Hwan Jun, Jens Lagergren.

References

1. Beerenwinkel N, Greenman CD, Lagergren J. Computational cancer biology: an evolutionary perspective. *PLoS computational biology*. 2016; 12(2). <https://doi.org/10.1371/journal.pcbi.1004717> PMID: 26845763
2. Szabo A, Boucher KM. Oncogenetic trees. In: *Handbook of cancer models with applications*. World Scientific; 2008. p. 1–24.
3. Desper R, Jiang F, Kallioniemi OP, Moch H, Papadimitriou CH, Schäffer AA. Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of computational biology*. 1999; 6(1):37–51. <https://doi.org/10.1089/cmb.1999.6.37> PMID: 10223663
4. Beerenwinkel N, Rahnenführer J, Kaiser R, Hoffmann D, Selbig J, Lengauer T. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. *Bioinformatics*. 2005; 21(9):2106–2107. <https://doi.org/10.1093/bioinformatics/bti274> PMID: 15657098
5. Farahani HS, Lagergren J. Learning oncogenetic networks by reducing to mixed integer linear programming. *PloS one*. 2013; 8(6). <https://doi.org/10.1371/journal.pone.0065773> PMID: 23799047
6. Tofigh A, Sjölund E, Höglund M, Lagergren J. A Global Structural EM Algorithm for a Model of Cancer Progression. In: *NIPS: Annual Conference on Neural Information Processing Systems*; 2011. p. 163–171.
7. Gerstung M, Baudis M, Moch H, Beerenwinkel N. Quantifying cancer progression with conjunctive Bayesian networks. *Bioinformatics*. 2009; 25(21):2809–2815. <https://doi.org/10.1093/bioinformatics/btp505> PMID: 19692554
8. Parviainen P, Farahani HS, Lagergren J. Learning bounded tree-width Bayesian networks using integer linear programming. In: *Artificial Intelligence and Statistics*; 2014. p. 751–759.
9. Hjelm M, Höglund M, Lagergren J. New probabilistic network models and algorithms for oncogenesis. *Journal of Computational Biology*. 2006; 13(4):853–865. <https://doi.org/10.1089/cmb.2006.13.853> PMID: 16761915
10. Dao P, Kim YA, Wojtowicz D, Madan S, Sharan R, Przytycka TM. BeWith: A Between-Within method to discover relationships between cancer modules via integrated analysis of mutual exclusivity, co-occurrence and functional interactions. *PLoS computational biology*. 2017; 13(10):e1005695. <https://doi.org/10.1371/journal.pcbi.1005695> PMID: 29023534
11. Schill R, Solbrig S, Wettig T, Spang R. Modelling cancer progression using Mutual Hazard Networks. *Bioinformatics*. 2020; 36(1):241–249. <https://doi.org/10.1093/bioinformatics/btz513> PMID: 31250881
12. Diaz-Uriarte R, Vasallo C. Every which way? On predicting tumor evolution using cancer progression models. *BioRxiv*. 2019; p. 371039. <https://doi.org/10.1371/journal.pcbi.1007246> PMID: 31374072
13. Vandin F, Upfal E, Raphael BJ. De novo discovery of mutated driver pathways in cancer. *Genome research*. 2012; 22(2):375–385. <https://doi.org/10.1101/gr.120477.111> PMID: 21653252
14. Constantinescu S, Szczurek E, Mohammadi P, Rahnenführer J, Beerenwinkel N. TiME: a waiting time model for mutually exclusive cancer alterations. *Bioinformatics*. 2016; 32(7):968–975. <https://doi.org/10.1093/bioinformatics/btv400> PMID: 26163509
15. Cristea S, Kuipers J, Beerenwinkel N. pathTiME: joint inference of mutually exclusive cancer pathways and their progression dynamics. *Journal of Computational Biology*. 2017; 24(6):603–615. <https://doi.org/10.1089/cmb.2016.0171> PMID: 27936934
16. Raphael BJ, Vandin F. Simultaneous inference of cancer pathways and tumor progression from cross-sectional mutation data. *Journal of Computational Biology*. 2015; 22(6):510–527. <https://doi.org/10.1089/cmb.2014.0161> PMID: 25785493
17. Robert C, Casella G. *Monte Carlo statistical methods*. Springer Science & Business Media; 2013.
18. Andrieu C, Thoms J. A tutorial on adaptive MCMC. *Statistics and computing*. 2008; 18(4):343–373. <https://doi.org/10.1007/s11222-008-9110-y>
19. Gonzalez-Perez A, Perez-Llamas C, Deu-Pons J, Tamborero D, Schroeder MP, Jene-Sanz A, et al. IntOGen-mutations identifies cancer drivers across tumor types. *Nature methods*. 2013; 10(11):1081. <https://doi.org/10.1038/nmeth.2642> PMID: 24037244
20. Zhang L, Shay JW. Multiple roles of APC and its therapeutic implications in colorectal cancer. *JNCI: Journal of the National Cancer Institute*. 2017; 109(8). <https://doi.org/10.1093/jnci/djw332> PMID: 28423402
21. Nakayama M, Oshima M. Mutant p53 in colon cancer. *Journal of molecular cell biology*. 2018; 11(4):267–276. <https://doi.org/10.1093/jmcb/mjy075>
22. Tan C, Du X. KRAS mutation testing in metastatic colorectal cancer. *World journal of gastroenterology: WJG*. 2012; 18(37):5171. PMID: 23066310

23. Boutin AT, Liao WT, Wang M, Hwang SS, Karpinets TV, Cheung H, et al. Oncogenic Kras drives invasion and maintains metastases in colorectal cancer. *Genes & development*. 2017; 31(4):370–382. <https://doi.org/10.1101/gad.293449.116> PMID: 28289141
24. Xu H, Zong H, Ma C, Ming X, Shang M, Li K, et al. Epidermal growth factor receptor in glioblastoma. *Oncology letters*. 2017; 14(1):512–516. <https://doi.org/10.3892/ol.2017.6221> PMID: 28693199
25. Behnan J, Finocchiaro G, Hanna G. The landscape of the mesenchymal signature in brain tumours. *Brain*. 2019; 142(4):847–866. <https://doi.org/10.1093/brain/awz044> PMID: [The request was aborted: Could not create SSL/TLS secure channel.](#)
26. Benitez JA, Ma J, D'Antonio M, Boyer A, Camargo MF, Zanca C, et al. PTEN regulates glioblastoma oncogenesis through chromatin-associated complexes of DAXX and histone H3. 3. *Nature communications*. 2017; 8:15223. <https://doi.org/10.1038/ncomms15223> PMID: 28497778
27. Zhang Y, Dube C, Gibert M, Cruickshanks N, Wang B, Coughlan M, et al. The p53 pathway in glioblastoma. *Cancers*. 2018; 10(9):297. <https://doi.org/10.3390/cancers10090297> PMID: 30200436
28. Hasslacher S, Schneele L, Stroh S, Langhans J, Zeiler K, Kattner P, et al. Inhibition of PI3K signalling increases the efficiency of radiotherapy in glioblastoma cells. *International journal of oncology*. 2018; 53(5):1881–1896. <https://doi.org/10.3892/ijo.2018.4528> PMID: 30132519
29. Deng L, Xiong P, Luo Y, Bu X, Qian S, Zhong W, et al. Association between IDH1/2 mutations and brain glioma grade. *Oncology letters*. 2018; 16(4):5405–5409. <https://doi.org/10.3892/ol.2018.9317> PMID: 30250611
30. Ichimura K, Pearson DM, Kocalkowski S, Bäcklund LM, Chan R, Jones DT, et al. IDH1 mutations are present in the majority of common adult gliomas but rare in primary glioblastomas. *Neuro-oncology*. 2009; 11(4):341–347. <https://doi.org/10.1215/15228517-2009-025> PMID: 19435942
31. Mondal G, Stevers M, Goode B, Ashworth A, Solomon DA. A requirement for STAG2 in replication fork progression creates a targetable synthetic lethality in cohesin-mutant cancers. *Nature communications*. 2019; 10(1):1686. <https://doi.org/10.1038/s41467-019-09659-z> PMID: 30975996