

Programming assignment, R: Simulating Pathways, Mutual Exclusivity, et al. in Cancer Progression Models

Raquel Blanco Martínez-Illescas*, Daniel Peñas Utrilla*, Henry Secaira Morocho*

2021-01-21

Contents

1	Introduction	1
1.1	Cancer Progression Models	1
1.2	Evolutionary Models	2
1.3	Order of effects	2
1.4	Epistatic interactions	2
1.4.1	Synthtetic Viability	3
1.4.2	Mutual exclusivity	3
1.5	Frequency dependent fitness	3
2	Methods	3
3	PathTiMEx, a generative probabilistic graphical model of cancer progression	4
3.1	Simplified cancer progression model	7
3.2	Simulating data from a simplified model	11
3.3	Order effects	15
4	References	18

1 Introduction

1.1 Cancer Progression Models

Cancer is an heterogeneous disease caused by the continuous accumulation of different somatic mutations during lifetime of an individual (1–3). Identifying mutations leading to cancer progression becomes key to understand cancer development and treatment (4,5). Somatic mutations that affect the cells are classified in two main groups: passenger and driver mutations. Passenger mutations are silent mutations which do not lead to any pathogenic situation (6). On the other hand, driver mutations provide the cells with

*Universidad Autónoma de Madrid, Bionformatics and Computational Biology Master

morphological and metabolic alterations that ultimately lead to a selective growth advantage (7). Driver mutations can affect both oncogenes and tumor suppressor genes, but conversely (8).

Although cancer progression is a dynamic process, tumor data is usually obtained as cross-sectional samples. This sort of data is a combination of single-time snapshots taken from different tumors at different stages of cancer progression (2). However, a longitudinal dataset consisting of same individual tumor samples from different time points, is preferred. Not all possible mutations' order seem to be equally responsible for cancer progression. Therefore, it is necessary to know which are the restrictions leading to cancer development. Models explaining those dependencies are called Cancer Progression Models (CPMs) (9). CPMs are depicted as Directed Acyclic Graphs (DAG), where nodes represent genes and arrows dependencies between them (5).

1.2 Evolutionary Models

Previous studies have inferred the alterations (passenger and driver genes) and the order in which they occur during cancer progression using generative probabilistic models (1,10,11). Those methods use Oncogenic Trees (OT) and Conjunctive Bayesian Networks (CBN) to impose restrictions in the occurrence of mutations. As discussed in (9), in such methods a mutation in a driver gene can occur only if the preceding parent mutations have occurred, this is known as *monotonicity*. Nevertheless, it is not realistic to have a single set of restrictions for all genotypes, since genotypes can follow different paths during disease progression (9). Thus, OTs and CBNs cannot be used to address deviations from monotonicity (9). However, Evolutionary tumor progression models can incorporate the order restrictions from OTs and CBNs and allow us to analyze the consequences of deviations from monotonicity and the genetic context in which a mutation appears (9). Moreover, fitness landscapes can be used to understand the consequences of different evolutionary scenarios in CPMs, such as the possible paths of tumor progression and identification of genes that can block those paths (5).

1.3 Order of effects

The order in which somatic mutations are acquired influence clonal evolution, since mutations may behave as driver or passenger depending on the genetic context (9,12). Three mechanisms may contribute to the influence of order of effects (12). First, the initial mutation can alter the cellular environment of a neoplastic clone. Then, as a consequence, the second mutation will arise in a cellular environment determined by the first mutation (12). Second, the initial mutation can alter cellular pathways as targets for subsequent mutations (12). Third, the initial mutation can modify the epigenetic program of cells and thus alter the consequences of the second mutation (12). Therefore, the fitness of a mutations depends in which mutations were acquired previously. It is important to mention that order of effects from the restrictions imposed in a DAG, since in restrictions the fitness of a double mutant does not depend on which mutation was acquired first (13).

1.4 Epistatic interactions

Epistasis is defined as a deviation from the expected phenotype when combining two alleles (14). Cancer progression is driven by the accumulation of somatic mutations that interact epistatically, that is their effect is non-additive to the tumor fitness as a phenotype (14,15). For example, combinations of mutations that show positive epistasis result in a stronger fitness increase (stronger than the additive effects of individual mutations) (15). On the other hand, mutations that show negative epistasis result in fitness decrease (less than expected from their additive effects) (15). Therefore, mutations that show positive fitness are more likely to co-occur, whereas mutations that show negative fitness are rarely observed together resulting in mutual exclusivity (15). Moreover, reciprocal sign epistasis (see below) affect the ruggedness and lead to multiple peaks (a signature of epistasis) in the fitness landscape (5,16).

1.4.1 Synthtetic Viability

Synthetic viability is the combination of two mutations that rescue the lethal effects of a each single mutation (17). The idea of synthetic viability has been recently applied to identify genomic markers for drug resistance prediction and drug-combination for anti-cancer therapy (17).

1.4.2 Mutual exclusivity

Mutual exclusivity is a common phenomenon in cancer progression (2,15) and occurs by synthetic lethality (described below) and null effect. This phenomenon is common in cancer signaling pathways (2,15). Synthetic lethality (or reciprocal sign epistasis) occurs when combination of two mutations is detrimental for the viability of the cell, whereas each individual mutation is not (5). On the other hand, the null effect states that a mutation that occurs first involves most of the selective advantage and thus decrease the selective pressure for others mutations to arise (2,15).

1.5 Frequency dependent fitness

(Just and idea) Describe frequency dependent fitness

In this work, we mapped DAGs inferred from three different generative probabilistic models to actual tumor evolutionary models by allowing deviations from monotonicity using functions of the **OncoSimulR** package. Moreover, we simulated other relevant scenarios for cancer progression, such as order of effects, epistatic interactions (syntehtic lethality, synthetic viability), and frequency-dependent fitness. In addition, we mapped the genotypes of our evolutionary models to fitness landscape in order to gain a better knowledge of mutational paths during tumor progression. Similarly, we did simulations of tumor progression to understand the effect of fitness associated to each genotype.

2 Methods

Complete fitness specification of each model is obtained from the function `allFitnessEffects`. The fitness-genotype mapping is specified by a data frame. Each evolutionary models is coded setting this data frame where dependencies between genes inferred in each model are indicated as parent-child relationships. Parent gene mutations are mandatory to child gene mutations to occur (monotonicity). Parents and children gene mutations are introduced in the model by two different columns, “parent” and “child”, respectively. Mutations do not requiring a previous mutation derive from a “Root” node (non-altered genotype). Moreover, two additional columns are set to specify fitness effects associated to each genotype (evolutionary model, not just a generative model). Those columns are “s” and “sh”. In the column “s” is specified the fitness effect when the restrictions defined in the CPMs model are satisfied. On the other hand, in column “sh” is set the fitness effect when restrictions are not satisfied. Mutations that are against inferred constraints can be set to a 0 value in fitness effect. However, we want to allow deviations from the monotonicity, setting a penalization when this situations occurs (negative value in “sh”).

Additionally, type of dependency between mutations can also be specified in the data frame. There are three different possible dependencies in **OncoSimulR**: monotone relationship, where relationship between specific genes is fully respected; semiminotone relationship, where two or more genes are connected to the same gene, but if just one of the parent mutations occurs is enough to child gene mutation; and XOR relationship, similar to the previous one, but child gene mutation will occur only if one parent gene is already mutated. Dependency is set in a new column called “typeDep” in the data frame. Nomenclature for the three possible genes relationship is “MN”, “SM” and “XMPN” for monotone, semimonotone and XOR relationship, respectively.

Restrictions’ data frame just defined is used as argument of the function `allFitnessEffects`. Restrictions considered in the data frame can apply not to one gene, but to a set of genes (module) (1,10,11). This situation

is defined by the argument “geneToModule”. Furthermore, OncoSimulR allows to specify driver or passenger genes in the models. In the model mapped, all genes/modules implemented are driver genes/modules, but they are explicitly indicated. A fitnessEffects object is returned from the function allFitnessEffects, and is used as input for the function plot and the DAG is visualized. It is possible because OncoSimulR package implement the method plot.fitnessEffects for the fitnessEffects objects. When modules are used, they can be expanded with the argument “expandModules = TRUE”. In addition, Wild type fitness is shown setting “addwt = TRUE”. To obtain a table with all the fitness effects of all genotypes, the previous fitnessEffects object is used as input in the function evalAllGenotypes. Finally, fitness landscape of each evolutionary model is visualize with both plot or PlotFitnessLandscape function.

Fitness effects associated to each genotype can be used to simulate tumor progression with the function oncoSimulIndiv or oncoSimulPop. These functions simulates a single evolutionary trajectory or a set of evolutionary trajectories in the same conditions, respectively. McFarland model (continuous-time, logistic-like, and death rate depends on population size) is used for simulation of cancer progression, since it leads to a better performance (9). Initial population size, the mutation rate and final time of simulation is also set. The argument “sampleEvery” informs about how often the whole population is sampled. As we are using McFarland model, a very small value is set. The argument “keepEvery” is set larger than the argument “sampleEvery” to obtain nice plots. “keepPhylog = TRUE” is set to plot the parent-child relationships occurring in the simulation as well as its frequency, using the “plotClonePhylog” function. The argument “onlyCancer” is set to TRUE when simulation is returned if cancer is reached. Otherwise, it is set to FALSE. Simulations are plotted using the function “plot”. Plot styles is set to “stacked” or “line” with the argument “type”.

From the canonical models, simplified derived models are constructed. Those models maintain dependencies between genes, but they just focus on the important genes in cancer onset (18). These simplified versions are used to properly show and discuss different utilities available in OncoSimulR, such as order effects, epistasis, synthetic viability, synthetic lethality or frequency-dependent fitness.

3 PathTiMEx, a generative probabilistic graphical model of cancer progression

Cristea et al., 2017 (10) introduce a generative probabilistic graphical model of cancer progression called pathTiMEx. It is both, a waiting time model for independent mutually exclusive pathways, and a waiting time model for cancer progression among single genes. The colorectal cancer model depicted in Figure 3.A (10) is composed of 8 genes (APC, KRAS, TP53, EVC2, PIK3CA, EPHA, FBXW7 and TCF7L2), and is organized in 3 mutually exclusive modules and 2 individual genes. Mutually exclusive modules represent a set of genes participating in the same pathway that are not mutated at the same time (19). If one of the genes of the module is mutated, other genes will not mutate. The colorectal cancer dataset used to built that model is obtained from (20). Generative model inferred is mapped into an evolutionary model, where deviations from monotonicity are allowed.

The poset restrictions proposed (10) can be coded using the function allFitnessEffects. Some parameters are mandatory when the function is used (restriction table). s and sh values are not specified in (10) since they are not interested in fitness. Values given to both parameters derive from the waiting time rate parameter λ defined in the model. Early events in cancer progression are important to cancer onset and therefore, will get greater λ values, while late events will have a lower one (values for all genes or modules are showed in Table 1). It has been previously proposed that alterations associated with the onset of colorectal cancer (such as APC, KRAS and TP53) may provide a larger fitness advantage than alterations associated with tumor progression (21). On the other hand, sh is given a constant value for all possible deviations from restrictions. Dependency between genes is set as monotonic (MN). Model will be represented as a DAG.

Table 1: Waiting time rate parameter (λ) for each gene/module

Gene/module	Waiting time rate parameter (λ)
APC	9.5
KRAS	2.89
TP53, EVC2	1.92
PIK3CA, EPHA3	0.17
FBXW7, TCF7L2	0.08

```
## First, it is necessary to load OncoSimulR and igraph package
library(OncoSimulR)

## Restriction table (extended version of the poset)
colcancer <- data.frame(
  parent = c(rep("Root",3), "A", "B", "C"), # Parent nodes
  child = c("A", "B", "D", "C", "E", "E"), ## Child nodes
  s = c(0.5, 0.2, 0.05, 0.1, rep(0.05, 2)),

  sh = -0.3,

  typeDep = "MN" ## Type of dependency
)

## Fitness specification of the poset
colcancer_efec <- allFitnessEffects(
  colcancer, # Poset

  geneToModule = c( ## Specification of the modules
    "Root" = "Root",
    "A" = "APC",
    "B" = "TP53, EVC2",
    "C" = "KRAS",
    "D" = "PI3KCA, EPHA",
    "E" = "FBXW7, TCF7L2"),

  drvNames = c( ## Specification of drivers
    "APC", "TP53", "EVC2", "KRAS",
    "PI3KCA", "EPHA", "FBXW7", "TCF7L2")
)

## DAG representation
plot(colcancer_efec, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

Figure 1 shows the DAG derived from the generative model proposed by (10). Different limbs branch out from a non-altered genotype (“Root” node), each of them mimicking dependencies inferred in (10). The function `evalAllGenotypes` is used to map genotypes to fitness values. Figure 2 shows the fitness landscape

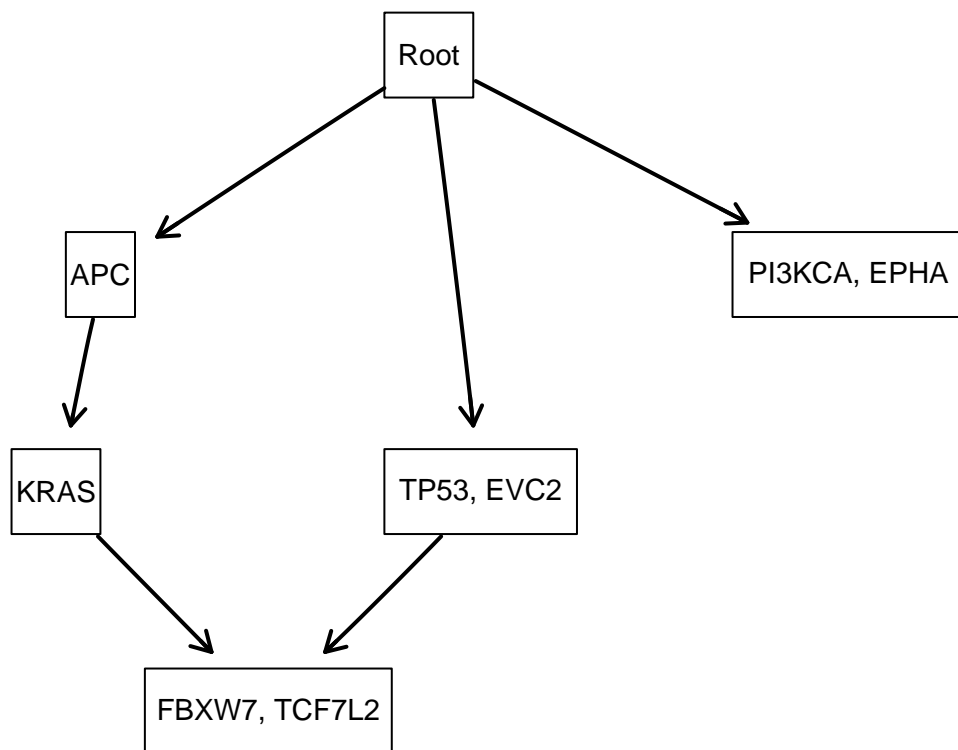


Figure 1: DAG from colorecta cancer

derived from the DAG (see Figure 1). All possible genotypes obtained from the DAG are labeled in the fitness landscape. A busy combination of mountains and valleys can be observed in the fitness landscape due to the huge amount of possible genotypes combinations. The highest peaks represent local maximum (green box), while the lowest peaks represent local minimum (red box). Genotypes climb or go down hills depending on whether mutation satisfies or no restrictions between them. Evolutionary model allows to map the fitness associated to each genotype going beyond a simple set of restrictions.

```
colcancer_efec_FL <- evalAllGenotypes(colcancer_efec, max = 110000)
## Output is not shown due to size of the table.

## Plot of fitness landscape
plotFitnessLandscape(colcancer_efec_FL)
```

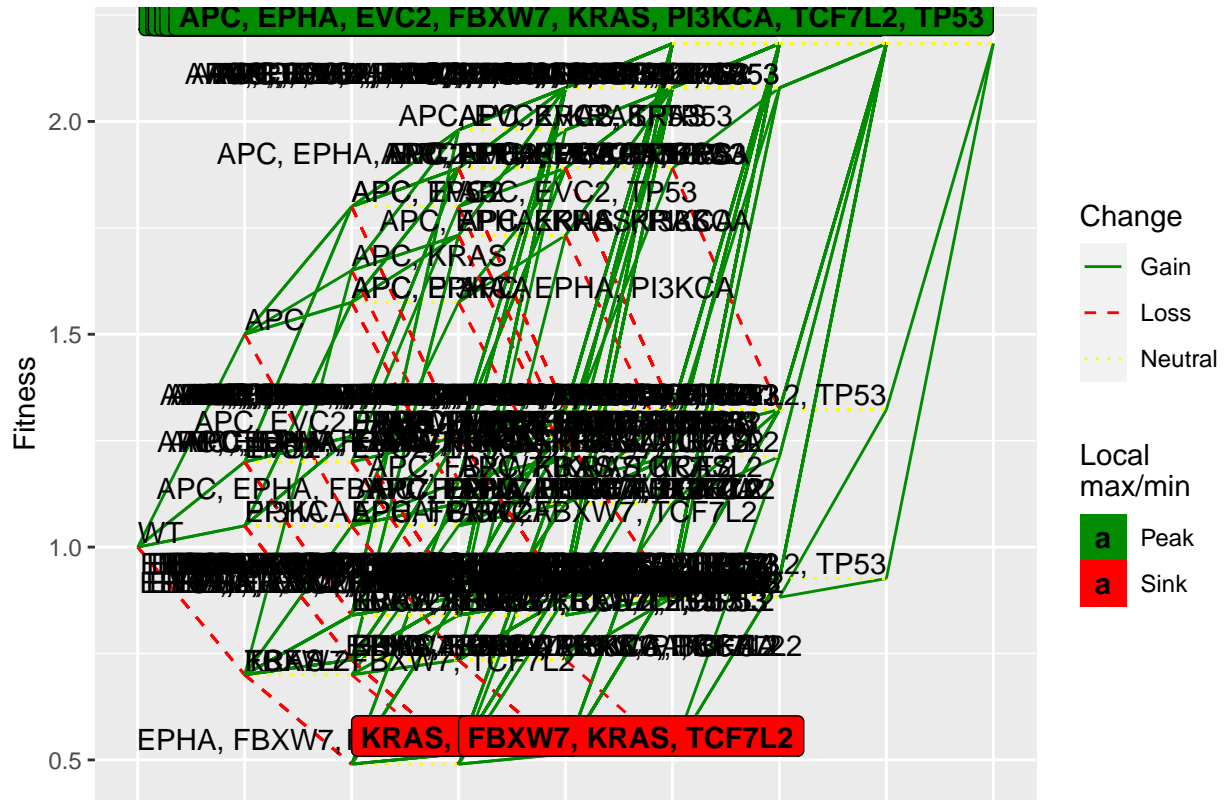


Figure 2: Fitness landscape from colorectal cancer

3.1 Simplified cancer progression model

In order to properly visualize the fitness landscape, a simplified version of the model coded in section 3 is generated. This model does not use modules, just individual genes. This approach will lead to clear fitness landscape and to properly identification of events that may occur.

There is a phenomena of mutual exclusivity between genes of the modules (10). Mutations of certain genes may not be present in the genotype if another gene of the same pathway is already mutated. Dependencies between genes of the same module are defined by default as semimonotone (OR relationship) in OncosimulR.

Hence, mutations of one of the genes mutually excluded of the specific pathway is enough to provide all fitness contribution to the genotype. Mutations in other genes of the same module will constitute a null effect to genotype fitness.

```
## Fitness specification of the simplified poset
Scolcancer <- allFitnessEffects(colcancer,

    geneToModule = c( ## Specification of the modules
        "Root" = "Root",
        "A" = "APC",
        "B" = "TP53",
        "C" = "KRAS",
        "D" = "PI3KCA",
        "E" = "FBXW7"),

    drvNames = c( ## Specification of drivers
        "APC", "TP53", "KRAS",
        "PI3KCA", "FBXW7")

    )
```

```
plot(Scolcancer, expandModules = TRUE, autofit = TRUE)
```

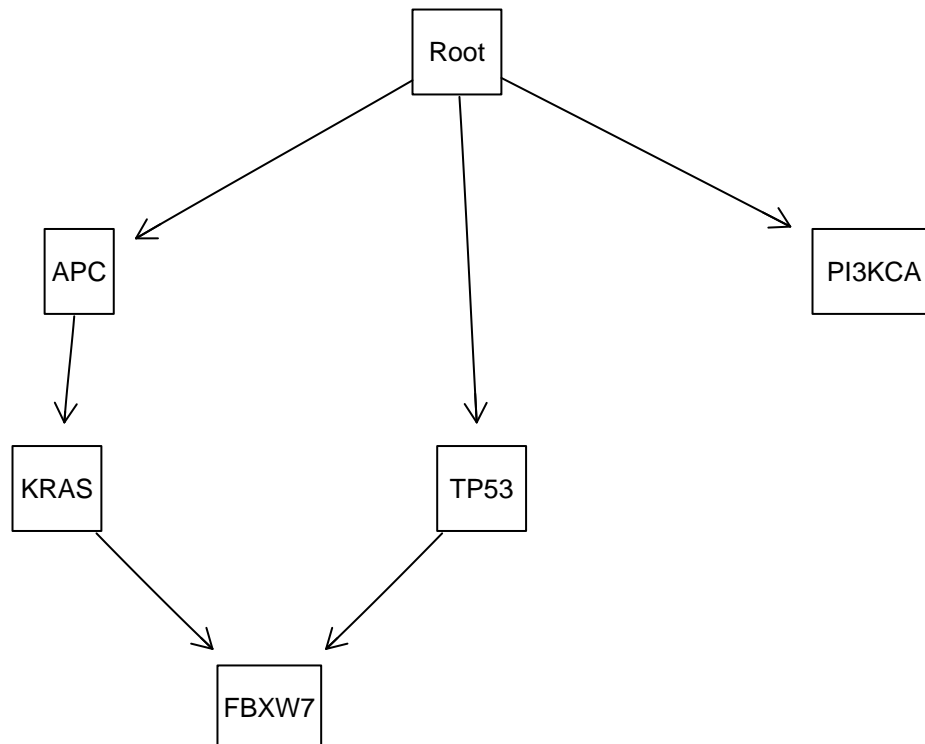


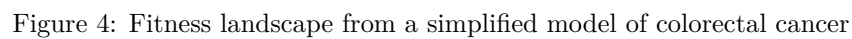
Figure 3: DAG from a simplified model of colorectal cancer


```
## Obtain all genotypes from the fitnessEffect
(Scolcancer_ge <- evalAllGenotypes(Scolcancer))
```

```
##                               Genotype Fitness
## 1                               APC 1.50000
## 2                               FBXW7 0.70000
## 3                               KRAS 0.70000
## 4                               PI3KCA 1.05000
## 5                               TP53 1.20000
## 6                   APC, FBXW7 1.05000
## 7                   APC, KRAS 1.65000
## 8                   APC, PI3KCA 1.57500
## 9                   APC, TP53 1.80000
## 10                  FBXW7, KRAS 0.49000
## 11                  FBXW7, PI3KCA 0.73500
## 12                  FBXW7, TP53 0.84000
## 13                  KRAS, PI3KCA 0.73500
## 14                  KRAS, TP53 0.84000
## 15                  PI3KCA, TP53 1.26000
## 16          APC, FBXW7, KRAS 1.15500
## 17          APC, FBXW7, PI3KCA 1.10250
## 18          APC, FBXW7, TP53 1.26000
## 19          APC, KRAS, PI3KCA 1.73250
## 20          APC, KRAS, TP53 1.98000
## 21          APC, PI3KCA, TP53 1.89000
## 22          FBXW7, KRAS, PI3KCA 0.51450
## 23          FBXW7, KRAS, TP53 0.88200
## 24          FBXW7, PI3KCA, TP53 0.88200
## 25          KRAS, PI3KCA, TP53 0.88200
## 26          APC, FBXW7, KRAS, PI3KCA 1.21275
## 27          APC, FBXW7, KRAS, TP53 2.07900
## 28          APC, FBXW7, PI3KCA, TP53 1.32300
## 29          APC, KRAS, PI3KCA, TP53 2.07900
## 30          FBXW7, KRAS, PI3KCA, TP53 0.92610
## 31 APC, FBXW7, KRAS, PI3KCA, TP53 2.18295
```

```
## Plot the fitness landscape.
plotFitnessLandscape(Scolcancer_ge,
  use_ggrepel = TRUE)
```

Figure 3 and Figure 4 show the DAG graph and fitness landscape of this simplified model, respectively. DAG showed in Figure 3 is the same as the DAG depicted in Figure 1, but without expanding modules (a gene from each module was selected for this simplification). Only one local maximum is depicted in the fitness landscape (see Figure 4). It corresponds to the genotype carrying mutations in the five genes. Note that fitness value associated to local maximum does not differ much from the fitness value associated to the genotype carrying mutations in APC-PI3KCA-TP53. It reflex the idea that mutations related to cancer onset contribute higher to fitness than mutations related to cancer progression. Hence, fitness contributions of mutations in genes FBXW7 or PI3KCA are minimal. On the other hand, only one local minimum is also depicted. It corresponds to the genotype more deviated from monotonicity and it refers to the genotype carrying mutations in both KRAS and FBXW7 genes. Both mutations are expected to occur after mutations in genes APC and TP53, respectively.



3.2 Simulating data from a simplified model

DAG were used as a guide line to built the fitness landscape (see [Figure 4](#)). This fitness landscape shows each possible genotype as well as its fitness. This landscape can be used to simulate fitness evolution in cancer progression. The function `oncoSimulIndiv` is used to simulate colorectal tumor progression. Poset with the order restrictions defined for the simplified model (see [subsection 3.1](#)) are included. McFarland model is used for simulation of cancer progression. Initial population size is set at 500. Only one mutation rate is used, $5e-5$. The parameter `keepPhylog` is set true to plot the parent-child relationships occurring in the simulation as well as its frequency (`plotClonePhylog` function). The `onlyCancer` parameter is set True to stop the simulation when cancer is reached.

```
set.seed(35) ## Fix the seed for reproducibility

Simul <- oncoSimulIndiv(Scolcancer, ## A fitnessEffects object
  model = "McFL", ## Model used
  mu = 5e-5, ## Mutation rate
  sampleEvery = 0.03, ## How often the whole population is sampled
  keepEvery = 1,
  initSize = 500, ## Initial population size
  keepPhylog = TRUE, ## Allow to see parent-child relationships
  onlyCancer = TRUE,
  detectionSize = NA
)

## Plot of simulation
plot(Simul, ## OncoSimulIndiv model
  show = "genotypes",
  type = "stacked"
)

## Plot of simulation
plot(Simul, ## OncoSimulIndiv model
  show = "genotypes",
  type = "line"
)

## Parent-child relationship derived from simulation
plotClonePhylog(Simul, fixOverlap = TRUE,
  N = 0, ## Specify clones that exist
  keepEvents = TRUE ## Arrows showing how many times each clones appeared
)
```

A stacked and line plot of the simulation is depicted in [Figure 5](#) and [Figure 6](#), respectively. Both plots show the evolution of cell population's genotype with time. Note that cancer is reached, and thus simulation is stopped, when not the local maximum genotype is fixed (genotype carrying all mutations, see [Figure 4](#)). Interestingly, it does not appear in simulation when cancer is reached. Although different genotypes coexist when cancer is reached, APC-KRAS-TP53 is the predominant genotype fixed in the simulation. This genotype carries mutations in the three genes yielding cancer onset. It was previously discussed that those

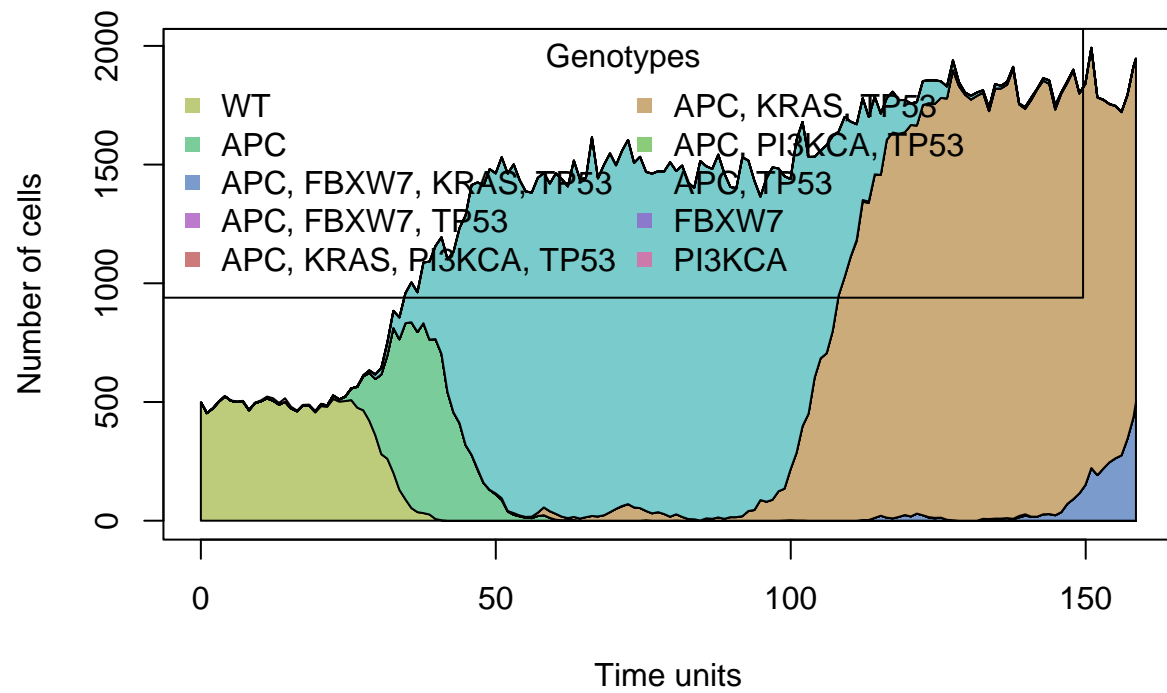


Figure 5: Simulation of cancer progression using the fitness landscape of the simplified model (stacked plot)

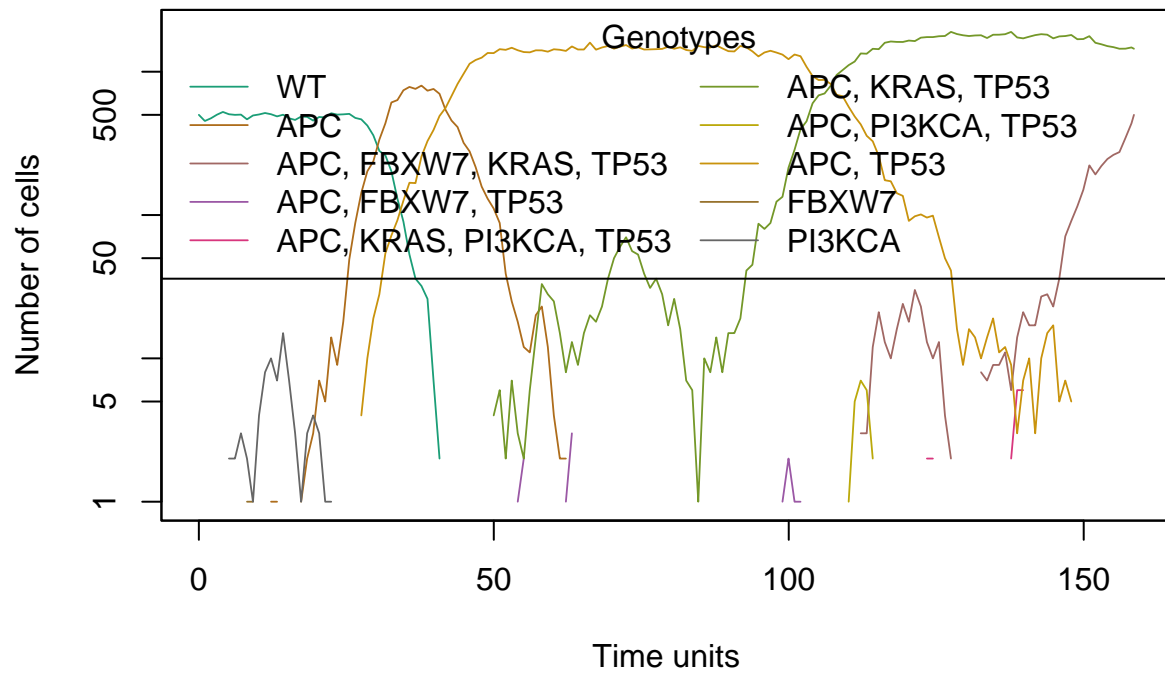


Figure 6: Simulation of cancer progression using the fitness landscape of the simplified model (line plot)

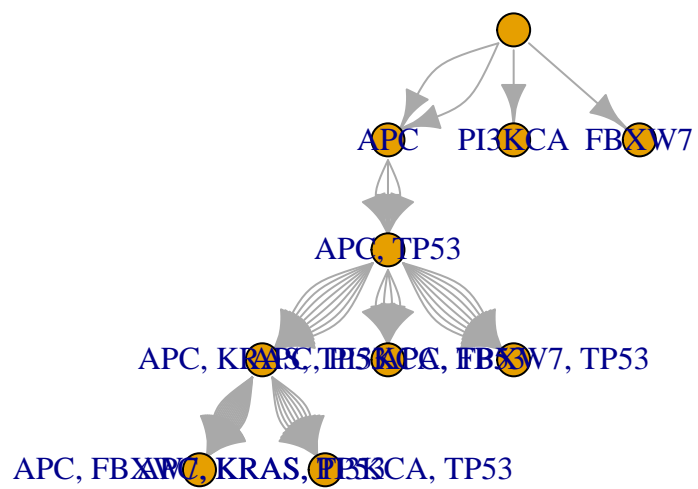


Figure 7: Parent-child relationship derived from simulation

three genes were responsible for cancer onset, and thus they have associated a higher fitness contribution than mutations responsible for cancer progression. Moreover, remember that its fitness was close to the local maximum peak in the fitness landscape (see [Figure 4](#)). In this context, it is not surprising that cancer is reached when genotype carrying those three mutations is fixed in simulation. Genotype carrying an additional mutation in the FBXW7 gene starts to arise, but this genotype is responsible for cancer progression, next step in cancer.

[Figure 6](#) shows a better perspective of genotypes in the simulation. Wild type genotype coexists with clones carrying mutations in APC or PI3KCA. However, APC genotype's increase triggers wild type extinction. PI3KCA genotype is not able to compete with other genotypes and ends by disappearing (deviation from monotonicity). Interestingly, when APC-TP53-KRAS genotype first shows up, it can not compete with the predominant genotype at that time. In fact, it is extincted, but when it shows up for the second time it is able to fix in simulation and leads to APC-TP53 genotype's extinction.

[Figure 7](#) shows the genealogical evolution of genotypes in the simulation. Arrows' width represent frequency of clone apparition. Wider arrows indicate a higher frequency of change from the parent genotype to the child genotype. Although Wild Type genotype mutates in APC, PI3KCA or FBXW7, only genotypes carrying APC mutation remain. The highest frequency between parent and child genotypes is located between genotypes APC-KRAS-TP53 and APC-FBXW7-KRAS-TP53. Thus, cancer progression is led by APC-FBXW7-KRAS-TP53 genotype that comes from APC-KRAS-TP53 genotype.

3.3 Order effects

To explore order effects in cancer progression, another evolutionary model derived from the generative model inferred by (10) is created. This simplified model just contains 4 genes: APC, TP53, FBWX7 and KRAS. The relationships between those genes was previously depicted in [subsection 3.1](#). Both, APC and TP53 genes have as "parent" the non-altered genotype ("Root"). APC gene has as "parent" KRAS gene. On the other hand, mutation in FBWX7 gene requires KRAS and TP53 genes already mutated. [Figure 8](#) shows the DAG of the simplified model just described. s and sh parameters are the same than in [section 3](#). Dependency between genes is set as monotonic ("MN").

Based on the waiting time rate parameter λ , the fitness effect of each possible order is given (see [Table 1](#)). APC is associated with the highest λ value, which means that it seems to mutate early in the cancer progression. λ for FBWX7 is the lowest between the four, meaning that it mutates the last. TP53 mutation occurs between APC and KRAS. The order effect favored is: $APC > TP53 > KRAS > FBWX7$. This order in mutations' acquisition is consistent with the time rate parameter λ , and is given the highest fitness. Other possible combinations of mutation's acquisition are not consistent with restrictions inferred in (10), and therefore a lower fitness value is given.

Order effects between genes is introduced in the argument `orderEffects` of the `allFitnessEffects` function, and is defined with ">" symbols; for instance, $A > B$ indicates that order effects is satisfied only when gene A is mutated before gene B. Fitness-genotype association is visualize using the `evalAllGenotypes` function. A table containing all possible mutations acquisition's orders and the fitness associated to each of these orders (genotypes) is obtained. In this approach, each possible genotype is associated to multiple fitness values (for instance, APC-FBWX7 genotype is mapped to a fitness value of 1 or 1.2 depending on gene order mutations), except for the genotypes only carrying one mutation. Different fitness values associated to each possible genotype depend on the mutation's path followed by the genotype. Mutation's path consistent with restrictions defined in the DAG are associated to higher fitness values. However, paths violating those restrictions (deviations from monotonicity) are penalize and fitness decreases. In total, there are 64 possible mutations' order. As expected, the highest fitness value corresponds to the genotype that suffered the mutations in the order that does not deviate from monotonicity.

In [section 3](#) and [subsection 3.1](#) order effect is not considered, and final fitness value is the same for genotypes carrying the same mutations. Nevertheless, if order effect is assumed, multiple fitness values are associated to each genotype yielding to a complex fitness landscape where a single genotype is multidepicted. An error raised when we try to plot the DAG and the fitness landscape. Order effects implemented in `OncoSimulR`

allow to evaluate all possible gene mutation's paths (genotype-fitness table), but does not allow to generate neither a DAG nor a fitness landscape from restrictions specified as order effects. This is one limitation of the OncoSimulR package, it does not yet allow to visualize those evolutionary scenarios.

```
cc <- data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
  child = c("A", "C", "B", "D", "D"),
  s = c(0.5, 0.2, 0.1, rep(0.05,2)),
  sh = -0.3,
  typeDep = "MN")

cc_visuali <- allFitnessEffects(cc,
  geneToModule =
    c("Root" = "Root",
      "A" = "APC",
      "B" = "KRAS",
      "C" = "TP53",
      "D" = "FBWX7") )

plot(cc_visuali, expandModules = TRUE, autofit = TRUE)
```

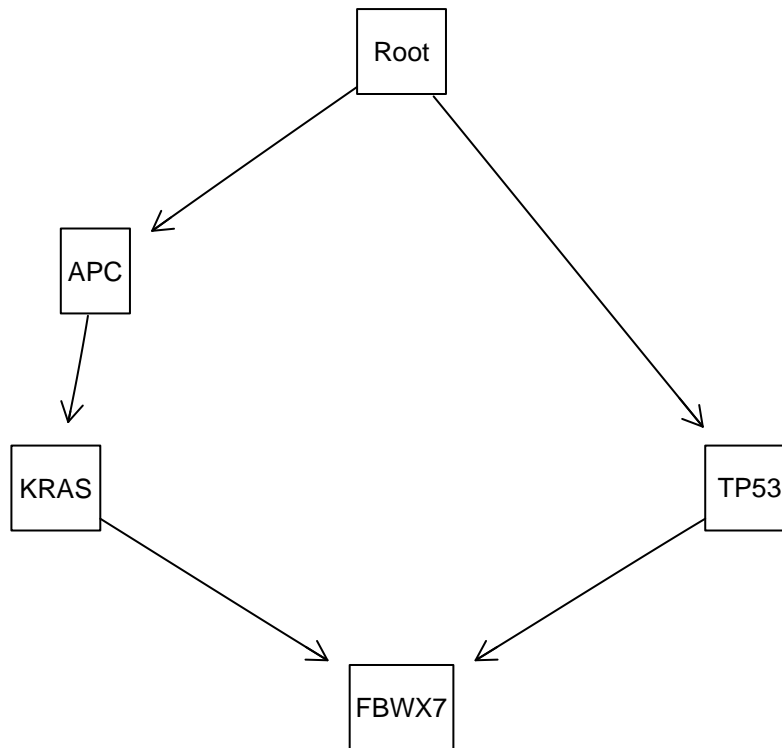


Figure 8: DAG from a simplified model of colorectal cancer

```
cc_order <- allFitnessEffects(
  orderEffects = c("A > B > C > D" = 0.5,
    "B > A > C > D" = 0.2,
```



```

        "B > C > A > D" = 0.1,
        "B > C > D > A" = 0.05,
        "A > C" = 0.2,
        "C > A" = 0.05,
        "D > A" = 0.05,
        "A > D" = 0.2,
        "B > D" = 0.2,
        "C > D" = 0.2,
        "B > C" = 0.2,
        "C > B" = 0.1,
        "B > A" = 0.1,
        "A > B" = 0.3),

geneToModule =
  c("A" = "APC",
    "B" = "KRAS",
    "C" = "TP53",
    "D" = "FBWX7") )

(cc_order_genotype <- evalAllGenotypes(cc_order, order = TRUE))

```

```

##           Genotype  Fitness
## 1              APC 1.000000
## 2             FBWX7 1.000000
## 3              KRAS 1.000000
## 4             TP53 1.000000
## 5      APC > FBWX7 1.200000
## 6      APC > KRAS 1.300000
## 7      APC > TP53 1.200000
## 8      FBWX7 > APC 1.050000
## 9      FBWX7 > KRAS 1.000000
## 10     FBWX7 > TP53 1.000000
## 11      KRAS > APC 1.100000
## 12      KRAS > FBWX7 1.200000
## 13      KRAS > TP53 1.200000
## 14      TP53 > APC 1.050000
## 15      TP53 > FBWX7 1.200000
## 16      TP53 > KRAS 1.100000
## 17     APC > FBWX7 > KRAS 1.560000
## 18     APC > FBWX7 > TP53 1.440000
## 19     APC > KRAS > FBWX7 1.872000
## 20     APC > KRAS > TP53 1.872000
## 21     APC > TP53 > FBWX7 1.728000
## 22     APC > TP53 > KRAS 1.716000
## 23     FBWX7 > APC > KRAS 1.365000
## 24     FBWX7 > APC > TP53 1.260000
## 25     FBWX7 > KRAS > APC 1.155000
## 26     FBWX7 > KRAS > TP53 1.200000
## 27     FBWX7 > TP53 > APC 1.102500
## 28     FBWX7 > TP53 > KRAS 1.100000
## 29     KRAS > APC > FBWX7 1.584000
## 30     KRAS > APC > TP53 1.584000
## 31     KRAS > FBWX7 > APC 1.386000

```

```
## 32      KRAS > FBWX7 > TP53 1.440000
## 33      KRAS > TP53 > APC 1.386000
## 34      KRAS > TP53 > FBWX7 1.728000
## 35      TP53 > APC > FBWX7 1.512000
## 36      TP53 > APC > KRAS 1.501500
## 37      TP53 > FBWX7 > APC 1.323000
## 38      TP53 > FBWX7 > KRAS 1.320000
## 39      TP53 > KRAS > APC 1.270500
## 40      TP53 > KRAS > FBWX7 1.584000
## 41 APC > FBWX7 > KRAS > TP53 2.246400
## 42 APC > FBWX7 > TP53 > KRAS 2.059200
## 43 APC > KRAS > FBWX7 > TP53 2.695680
## 44 APC > KRAS > TP53 > FBWX7 4.852224
## 45 APC > TP53 > FBWX7 > KRAS 2.471040
## 46 APC > TP53 > KRAS > FBWX7 2.965248
## 47 FBWX7 > APC > KRAS > TP53 1.965600
## 48 FBWX7 > APC > TP53 > KRAS 1.801800
## 49 FBWX7 > KRAS > APC > TP53 1.663200
## 50 FBWX7 > KRAS > TP53 > APC 1.455300
## 51 FBWX7 > TP53 > APC > KRAS 1.576575
## 52 FBWX7 > TP53 > KRAS > APC 1.334025
## 53 KRAS > APC > FBWX7 > TP53 2.280960
## 54 KRAS > APC > TP53 > FBWX7 3.284582
## 55 KRAS > FBWX7 > APC > TP53 1.995840
## 56 KRAS > FBWX7 > TP53 > APC 1.746360
## 57 KRAS > TP53 > APC > FBWX7 2.634509
## 58 KRAS > TP53 > FBWX7 > APC 2.200414
## 59 TP53 > APC > FBWX7 > KRAS 2.162160
## 60 TP53 > APC > KRAS > FBWX7 2.594592
## 61 TP53 > FBWX7 > APC > KRAS 1.891890
## 62 TP53 > FBWX7 > KRAS > APC 1.600830
## 63 TP53 > KRAS > APC > FBWX7 2.195424
## 64 TP53 > KRAS > FBWX7 > APC 1.920996
```

```
#DAG
plot(cc_order)
```

```
## Error in `tmp*`[[i]]: subíndice fuera de los límites
```

```
# Fitness landscape
plotFitnessLandscape(cc_order_genotype)
```

```
## Error in to_Fitness_Matrix(x, max_num_genotypes = max_num_genotypes): We cannot deal with order effects
```

4 References

1. Raphael BJ, Vandin F. Simultaneous Inference of Cancer Pathways and Tumor Progression from Cross-Sectional Mutation Data. *Journal of Computational Biology*. 2015;22(6):510–27. doi: [10.1089/cmb.2014.0161](https://doi.org/10.1089/cmb.2014.0161)
2. Schill R, Solbrig S, Wettig T, Spang R. Modelling cancer progression using Mutual Hazard Networks. *Bioinformatics*. 2020;36(1):241–9. doi: [10.1093/bioinformatics/btz513](https://doi.org/10.1093/bioinformatics/btz513)

3. Sprouffske K, Pepper JW, Maley CC. Accurate reconstruction of the temporal order of mutations in neoplastic progression. *Cancer Prevention Research*. 2011;4(7):1135–44. doi: [10.1158/1940-6207.CAPR-10-0374](https://doi.org/10.1158/1940-6207.CAPR-10-0374)
4. Pon JR, Marra MA. Driver and passenger mutations in cancer. *Annual Review of Pathology: Mechanisms of Disease*. 2015; doi: [10.1146/annurev-pathol-012414-040312](https://doi.org/10.1146/annurev-pathol-012414-040312)
5. Diaz-Uriarte R. Cancer progression models and fitness landscapes: A many-to-many relationship. *Bioinformatics*. 2018;34(5):836–44. doi: [10.1093/bioinformatics/btx663](https://doi.org/10.1093/bioinformatics/btx663)
6. Tomasetti C, Vogelstein B, Parmigiani G. Half or more of the somatic mutations in cancers of self-renewing tissues originate prior to tumor initiation. *Proceedings of the National Academy of Sciences of the United States of America*. 2013; doi: [10.1073/pnas.1221068110](https://doi.org/10.1073/pnas.1221068110)
7. Nowell PC. The clonal evolution of tumor cell populations. *Science*. 1976; doi: [10.1126/science.959840](https://doi.org/10.1126/science.959840)
8. Lee EYHP, Muller WJ. Oncogenes and tumor suppressor genes. 2010. doi: [10.1101/cshperspect.a003236](https://doi.org/10.1101/cshperspect.a003236)
9. Diaz-Uriarte R. Identifying restrictions in the order of accumulation of mutations during tumor progression: Effects of passengers, evolutionary models, and sampling. *BMC Bioinformatics*. 2015;16(1):1–26. doi: [10.1186/s12859-015-0466-7](https://doi.org/10.1186/s12859-015-0466-7)
10. Cristea S, Kuipers J, Beerenwinkel N. PathTiMEx: Joint Inference of Mutually Exclusive Cancer Pathways and Their Progression Dynamics. *Journal of Computational Biology*. 2017;24(6):603–15. doi: [10.1089/cmb.2016.0171](https://doi.org/10.1089/cmb.2016.0171)
11. Neyshabouri MM, Jun SH, Lagergren J. Inferring tumor progression in large datasets. *PLoS Computational Biology*. 2020;16(10):1–16. Available from: <http://dx.doi.org/10.1371/journal.pcbi.1008183>
12. Ortmann CA, Kent DG, Nangalia J, Silber Y, Wedge DC, Grinfeld J, et al. Effect of Mutation Order on Myeloproliferative Neoplasms. *New England Journal of Medicine*. 2015 Feb;372(7):601–12. [accessed 20 Jan 2021] Available from: <https://doi.org/10.1056/NEJMoa1412098>
13. Diaz-Uriarte R. OncoSimulR: Genetic simulation with arbitrary epistasis and mutator genes in asexual populations. *Bioinformatics*. 2017;33(12):1898–9. doi: [10.1093/bioinformatics/btx077](https://doi.org/10.1093/bioinformatics/btx077)
14. Wang X, Fu AQ, McNERNEY ME, White KP. Widespread genetic epistasis among cancer genes. *Nature Communications*. 2014 Nov;5(1):4828. [accessed 20 Jan 2021] Available from: <https://www.nature.com/articles/ncomms5828>
15. Haar J van de, Canisius S, Yu MK, Voest EE, Wessels LFA, Ideker T. Identifying Epistasis in Cancer Genomes: A Delicate Affair. *Cell*. 2019 May;177(6):1375–83. [accessed 20 Jan 2021] Available from: <http://www.sciencedirect.com/science/article/pii/S0092867419305033>
16. Reia SM, Campos PRA. Analysis of statistical correlations between properties of adaptive walks in fitness landscapes. *Royal Society Open Science*. 7(1):192118. [accessed 20 Jan 2021] Available from: <https://royalsocietypublishing.org/doi/10.1098/rsos.192118>
17. Gu Y, Wang R, Han Y, Zhou W, Zhao Z, Chen T, et al. A landscape of synthetic viable interactions in cancer. *Briefings in Bioinformatics*. 2018 Jul;19(4):644–55. [accessed 20 Jan 2021] Available from: <https://doi.org/10.1093/bib/bbw142>
18. Gerstung M, Eriksson N, Lin J, Vogelstein B, Beerenwinkel N. The temporal order of genetic and pathway alterations in tumorigenesis. *PLoS ONE*. 2011;6(10). doi: [10.1371/journal.pone.0027136](https://doi.org/10.1371/journal.pone.0027136)
19. Ciriello G, Cerami E, Sander C, Schultz N. Mutual exclusivity analysis identifies oncogenic network modules. *Genome Research*. 2012; doi: [10.1101/gr.125567.111](https://doi.org/10.1101/gr.125567.111)
20. Wood LD, Parsons DW, Jones S, Lin J, Sjöblom T, Leary RJ, et al. The genomic landscapes of human breast and colorectal cancers. *Science*. 2007; doi: [10.1126/science.1145720](https://doi.org/10.1126/science.1145720)
21. Beerenwinkel N, Antal T, Dingli D, Traulsen A, Kinzler KW, Velculescu VE, et al. Genetic progression and the waiting time to cancer. *PLoS Computational Biology*. 2007; doi: [10.1371/journal.pcbi.0030225](https://doi.org/10.1371/journal.pcbi.0030225)