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

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

2021-01-22

Contents

1 Introduction 2

1.1 Cancer Progression Models . . . 2

1.2 Evolutionary Models . . . 2

1.3 Order of Effects . . . 2

1.4 Epistatic Interactions . . . 3

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 5

3.1 Simplified Cancer Progression Model . . . 7

3.2 Simulating Data from a Simplified Model . . . 11

3.3 Order Effects . . . 15

4 Pathway Linear Progression Model: Raphael & Vanding, 2015 18

4.1 Simplified Model . . . 22

4.2 Simulating Data from Simplified Model . . . 24

4.3 Synthetic Lethality . . . 32

4.4 Synthetic Viability . . . 37

5 A Probabilistic Model of Mutually Exclusive Linearly Ordered Driver Pathways 41

5.1 Simplified Cancer Progression Model . . . 45

5.2 Frequency-dependent Fitness . . . 53

[*] Universidad Autónoma de Madrid, Bionformatics and Computational Biology Master

6 Discussion and Conclusions 60

7 References 60

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 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 Conjuctive 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 understands 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 mechanism 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 inividual 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

One of cancer's key features is intratumor heterogeneity, which is the coexistence of multiple malignant clones in time. In an evolutionary context, different cells compete for niche resources, establishing relationships which impact each clone's fitness. Several authors state this behavior can be explained by **evolutionary game theory** (18), in which the "survival game" takes place between cells with different strategies to keep replicating. These strategies are the phenotypes acquired by certain accumulation of mutations, whose effect is also dependent on the temporary space in which they happen. Ancestor clones and their effect on modulating the tumor microenvironment (TME) would have condition the fitness of current phenotypes, which also impact each other while coexisting. Thus, it is reasonable to consider clones affect one another in frequency-dependent manner, meaning fitness is actually a function of the relative frequency of other clones (18).

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 codded 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 PlotFitnessLanscape 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 (19). 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 (20). 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 (21). 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 (22). 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
```

```r
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)
```

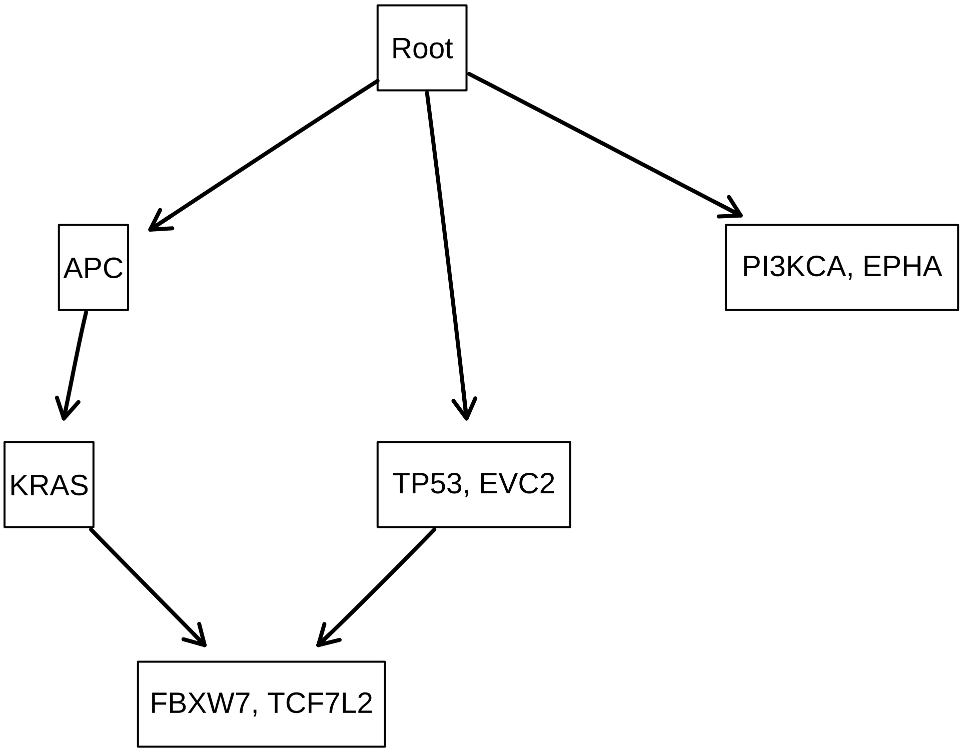

Figure 1: DAG from colorectal cancer dataset

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 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)
```

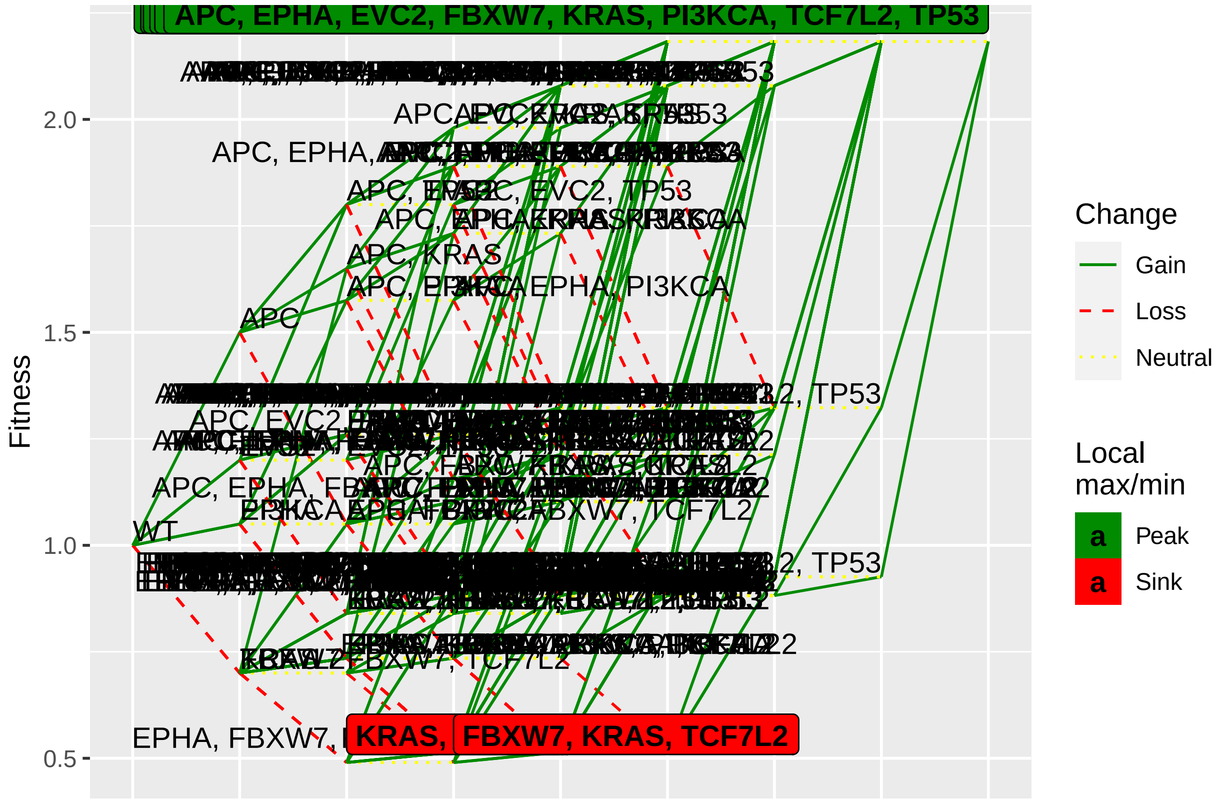

Figure 2: Fitness landscape from colorectal cancer

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.

```r
## 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")
                  )
```

```r
plot(Scolcancer, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

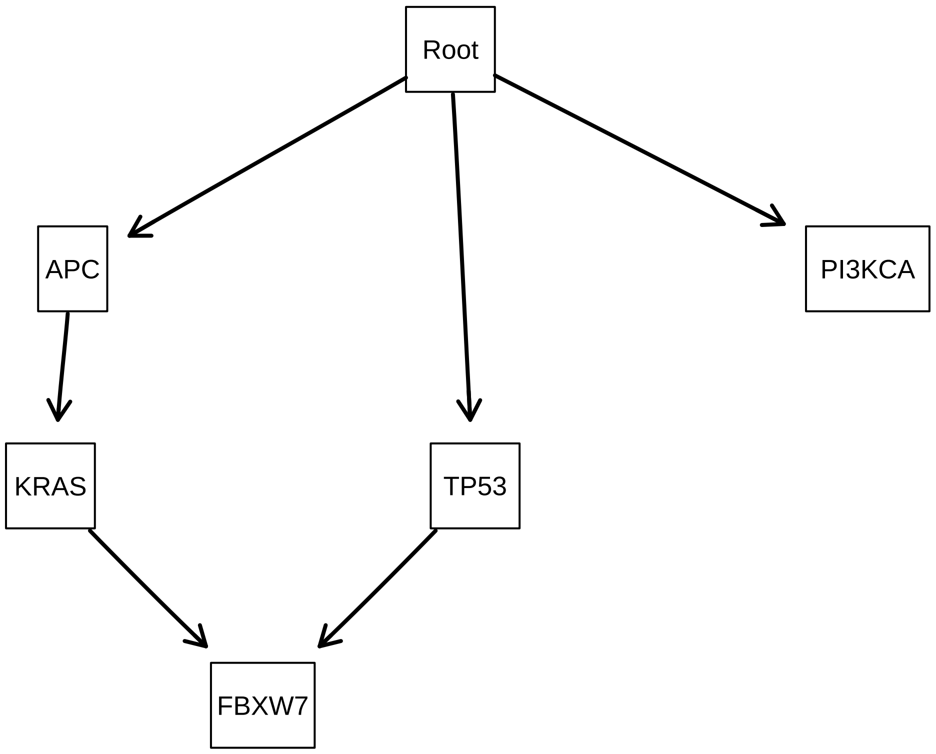

Figure 3: DAG from a simplified model of colorectal cancer

```r
## 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
```

```r
## 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.

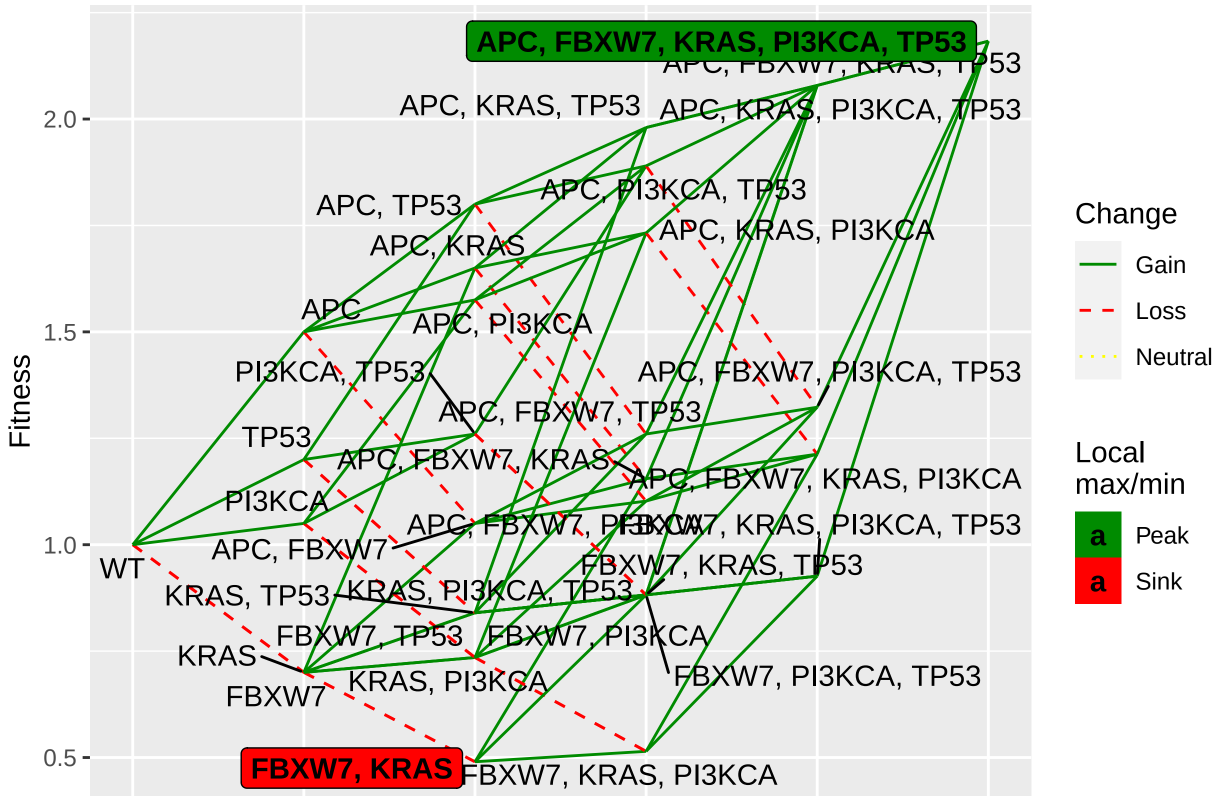

Figure 4: Fitness landscape from a simplified model of colorectal cancer

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.

```r
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, ## OncoSimulIndv model
     show = "genotypes",
     type = "stacked"

)
```

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

```r
## 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

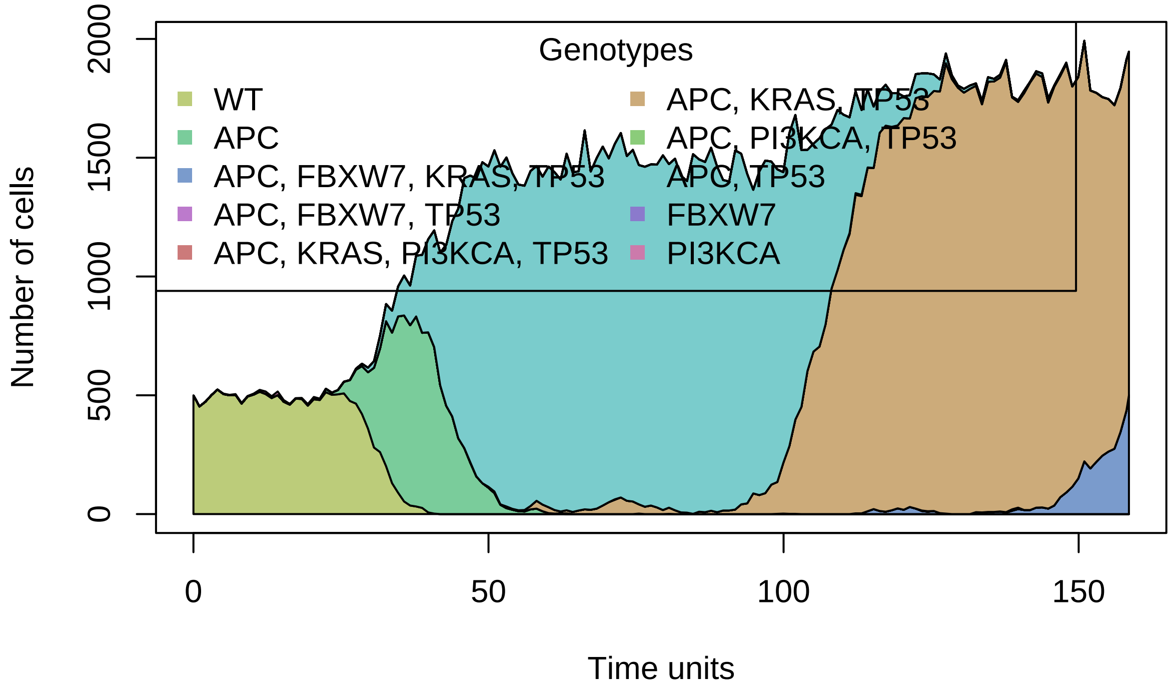

Figure 5: Simulation of cancer progression using the fitness landscape of the simplied model (stacked plot)

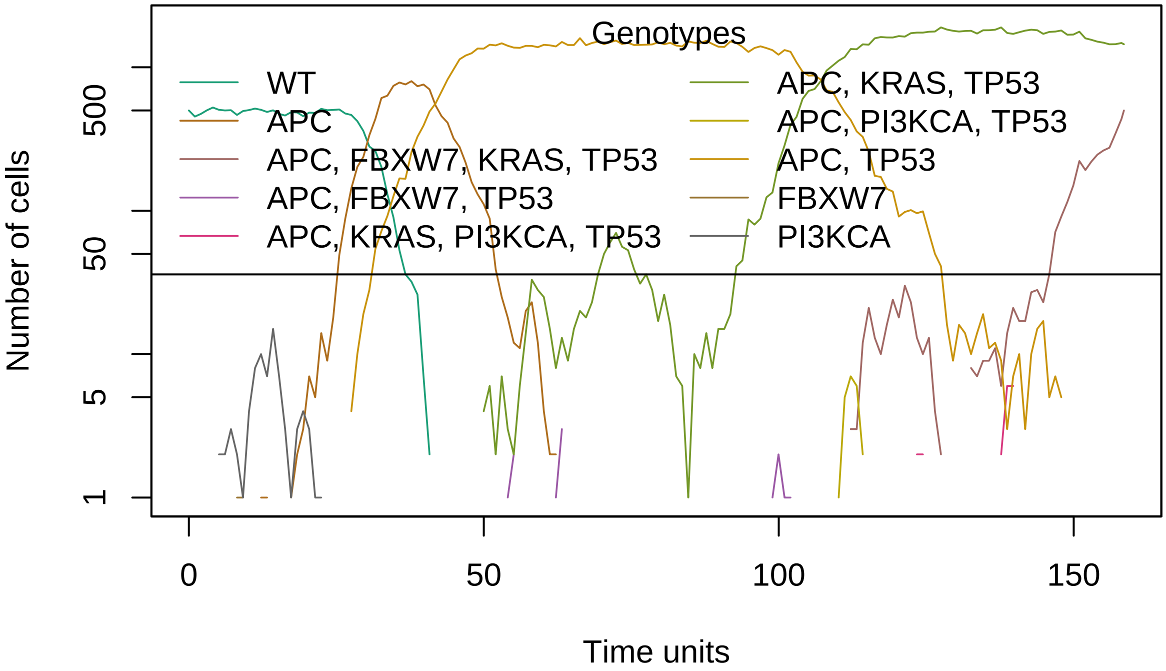

Figure 6: Simulation of cancer progression using the fitness landscape of the simplied model (line plot)

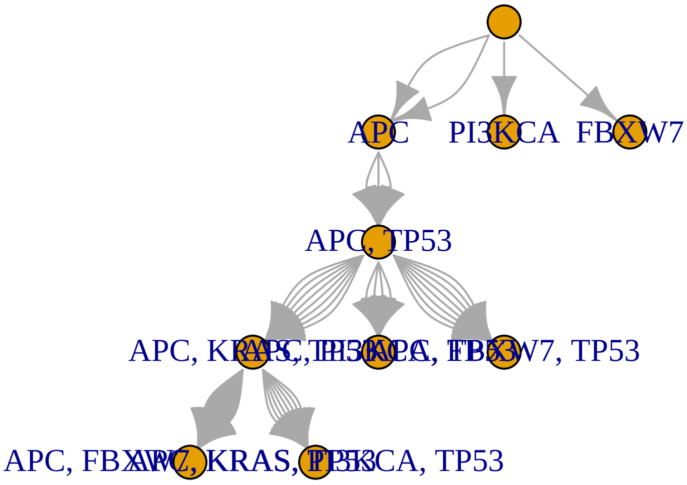

Figure 7: Parent-child relationship derived from simulation

three genes where 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 surprisingly 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. Widther 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), an 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 gen 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" = "FBXW7") )
## DAG
plot(cc_visuali, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

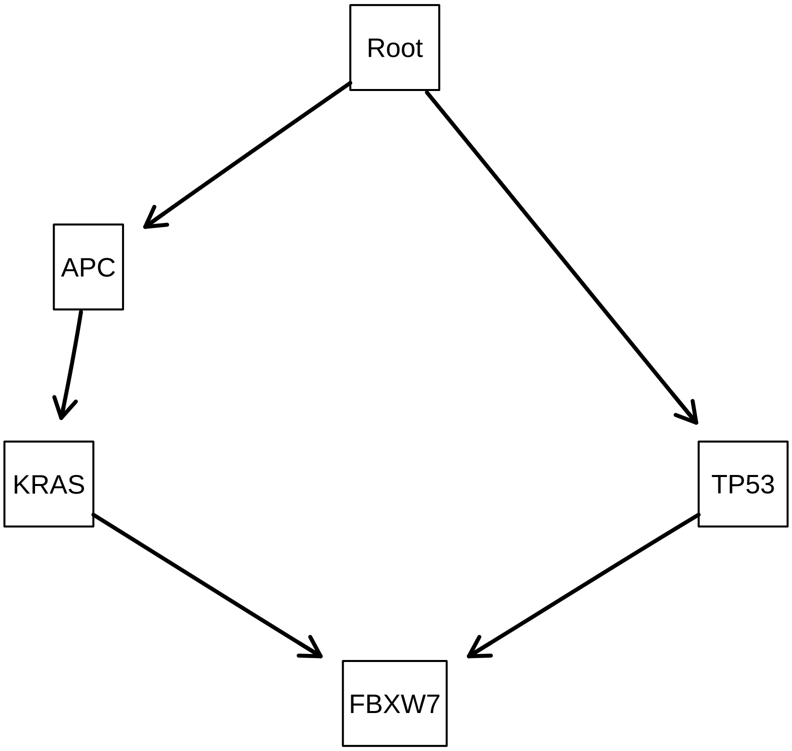

Figure 8: DAG from a simplified model of colorectal cancer

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

```r
                   "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" = "FBXW7") )

(cc_order_geno <- evalAllGenotypes(cc_order, order = TRUE))
```

```
##                      Genotype  Fitness
## 1                         APC 1.000000
## 2                       FBXW7 1.000000
## 3                        KRAS 1.000000
## 4                        TP53 1.000000
## 5                 APC > FBXW7 1.200000
## 6                  APC > KRAS 1.300000
## 7                  APC > TP53 1.200000
## 8                 FBXW7 > APC 1.050000
## 9                FBXW7 > KRAS 1.000000
## 10               FBXW7 > TP53 1.000000
## 11                 KRAS > APC 1.100000
## 12               KRAS > FBXW7 1.200000
## 13                KRAS > TP53 1.200000
## 14                 TP53 > APC 1.050000
## 15               TP53 > FBXW7 1.200000
## 16                TP53 > KRAS 1.100000
## 17         APC > FBXW7 > KRAS 1.560000
## 18         APC > FBXW7 > TP53 1.440000
## 19         APC > KRAS > FBXW7 1.872000
## 20          APC > KRAS > TP53 1.872000
## 21         APC > TP53 > FBXW7 1.728000
## 22          APC > TP53 > KRAS 1.716000
## 23         FBXW7 > APC > KRAS 1.365000
## 24         FBXW7 > APC > TP53 1.260000
## 25         FBXW7 > KRAS > APC 1.155000
## 26        FBXW7 > KRAS > TP53 1.200000
## 27         FBXW7 > TP53 > APC 1.102500
## 28        FBXW7 > TP53 > KRAS 1.100000
## 29         KRAS > APC > FBXW7 1.584000
## 30          KRAS > APC > TP53 1.584000
## 31         KRAS > FBXW7 > APC 1.386000
## 32       KRAS > FBXW7 > TP53 1.440000
## 33         KRAS > TP53 > APC 1.386000
## 34       KRAS > TP53 > FBXW7 1.728000
## 35        TP53 > APC > FBXW7 1.512000
## 36         TP53 > APC > KRAS 1.501500
## 37        TP53 > FBXW7 > APC 1.323000
## 38       TP53 > FBXW7 > KRAS 1.320000
## 39         TP53 > KRAS > APC 1.270500
## 40       TP53 > KRAS > FBXW7 1.584000
## 41 APC > FBXW7 > KRAS > TP53 2.246400
## 42 APC > FBXW7 > TP53 > KRAS 2.059200
## 43 APC > KRAS > FBXW7 > TP53 2.695680
## 44 APC > KRAS > TP53 > FBXW7 4.852224
## 45 APC > TP53 > FBXW7 > KRAS 2.471040
## 46 APC > TP53 > KRAS > FBXW7 2.965248
## 47 FBXW7 > APC > KRAS > TP53 1.965600
## 48 FBXW7 > APC > TP53 > KRAS 1.801800
## 49 FBXW7 > KRAS > APC > TP53 1.663200
## 50 FBXW7 > KRAS > TP53 > APC 1.455300
## 51 FBXW7 > TP53 > APC > KRAS 1.576575
## 52 FBXW7 > TP53 > KRAS > APC 1.334025
## 53 KRAS > APC > FBXW7 > TP53 2.280960
## 54 KRAS > APC > TP53 > FBXW7 3.284582
## 55 KRAS > FBXW7 > APC > TP53 1.995840
## 56 KRAS > FBXW7 > TP53 > APC 1.746360
## 57 KRAS > TP53 > APC > FBXW7 2.634509
## 58 KRAS > TP53 > FBXW7 > APC 2.200414
## 59 TP53 > APC > FBXW7 > KRAS 2.162160
## 60 TP53 > APC > KRAS > FBXW7 2.594592
## 61 TP53 > FBXW7 > APC > KRAS 1.891890
## 62 TP53 > FBXW7 > KRAS > APC 1.600830
## 63 TP53 > KRAS > APC > FBXW7 2.195424
## 64 TP53 > KRAS > FBXW7 > APC 1.920996
```

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

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

```
# Fitness landscape
plotFitnessLandscape(cc_order_geno)
```

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

4 Pathway Linear Progression Model: Raphael & Vanding, 2015

The Pathway Linear Progression Model (PLPM) described in (1) introduces the idea that driver mutations target pathways. This is an important concept since different individuals have driver mutations in different genes that affect the same pathway (1). Therefore, the order in which mutations arise are better described at the pathway level instead of a gene level (1).

Here, we mapped the progression model from colorectal cancer data inferred by (1) (originally described in (21)) into an evolutionary model, allowing deviations from the restriction imposed in the DAG. For this, we used a vector `s` to indicate the fitness effects when the restrictions are satisfied and a vector sh for deviations.

In (1), the authors analyzed eight genes: APC, EPHA3, EVC2, FBXW7, KRAS, PIK3CA, TCF7L2, and TP53. In this model, APC mutations is an early event, followed by mutations in TP53 and PIK3CA (mutually exclusive). KRAS mutations appear after TP53/PIK3CA mutations.

We used the allFitnessEffects function to define the nodes and their relationships. Moreover, we used modules to represent mutually exclusive genes that affect the same pathway. Assigned fitness effects (s) values were higher for earlier mutations and lower for late mutation, since an earlier mutation in more prevalent in the clonal population than a later mutation, as explained in (23). A single negative value was set for deviations from restrictions (sh) and a monotic relationship (MN) was used for relationships between nodes of the DAG since nodes have only one parent.

Figure 9 shows the DAG inferred by (1) mapped to an evolutionary model that allows deviation from restrictions. Note that genes within a module are mutually exclusive and the restrictions goes top-down (i.e. from the root to the later mutation).

```r
## Define poset restrictions, mapping of genes to modules, and driver genes
CRC_W <- allFitnessEffects(data.frame(parent = c("Root", "A", "B", "C"),
                                      child = c("A", "B", "C", "D"),
                                      s = c(0.6, 0.4, 0.1, 0.05),
                                      sh = -0.5,
                                      typeDep = "MN"),
                           geneToModule = c("Root" = "Root",
                                            "A" = "APC, EPHA3, TCF7L2",
                                            "B" = "EVC2, PIK3CA, TP53",
                                            "C" = "KRAS",
                                            "D" = "FBXW7"),
                           drvNames = c("APC", "EPHA3", "TCF7L2", "EVC2", "PIK3CA",
                                        "TP53", "KRAS", "FBXW7"))

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

The function evalAllGenotypes was used to map genotypes to fitness values. Figure 10 shows the fitness landscape inferred from the DAG of Figure 9. As mentioned before, this fitness landscape with eight genes is difficult to visualize. Nevertheless, we can give a general description of the topology of the landscape. Note that there are multiple peaks and valleys, suggesting a high degree of ruggedness. Moreover, note that KRAS constitutes a local minima. This results confirms the order of restrictions imposed by the DAG. It is important to mention that some genotypes in the local maxima are composed of genes that belong to the same module. Such genes participate in the same pathway and are mutually exclusive. Nevertheless, modues does not allow to capture this idea. A combinations of order of restrictions (XOR relationships) and epistatic interactions are able to better simulate mutual exclusivity (see below).

```r
## Map genotypes to fitness
CRC_F <- evalAllGenotypes(CRC_W, order = FALSE, addwt = TRUE)

## Plot of fitness landscape

plot(CRC_F)
```

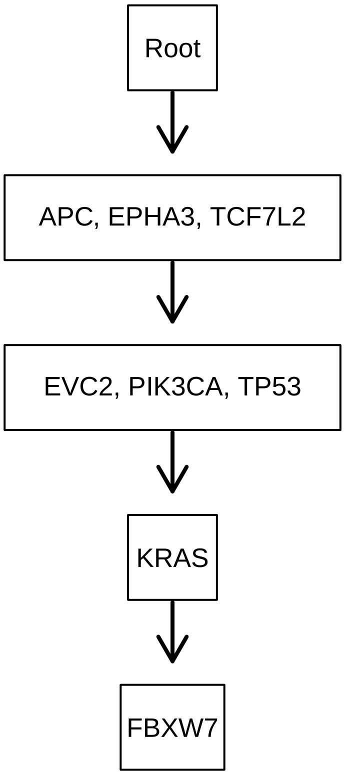

Figure 9: DAG from colorectal cancer dataset

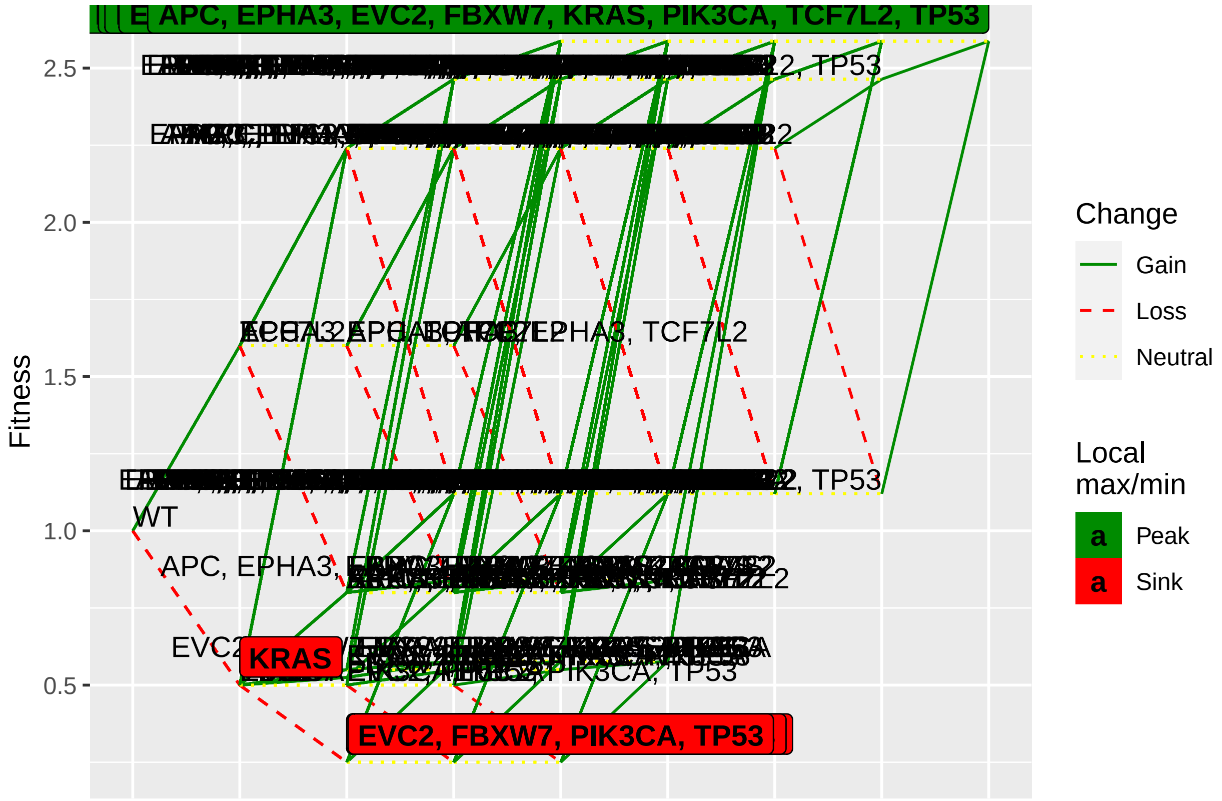

Figure 10: Fitness landscape inferred from colorectal cancer DAG

4.1 Simplified Model

Given that our initial DAG contains eight genes, then number of possible genotypes is $2^8 = 256$ which makes difficult to visualize the fitness landscape. For this, reason a smaller number of genes will be used to build a slightly different DAG to model other interesting scenarios (see Figure 11). The idea is to represent mutual exclusivity with a XOR relationship (red edges). Also, note that the fitness value for mutual exclusive genes (APC and TP53) is almost the same.

```
## Simplified model
## Define poset restrictions, mapping of genes to modules, and driver genes
CRC_W2 <- allFitnessEffects(data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
                                       child = c("A", "B", rep("C", 2), "D"),
                                       s = c(0.2, 0.1, rep(0.05, 2), 0.01),
                                       sh = -0.5,
                                       typeDep = c(rep("XMPN", 4), "MN")),
                            geneToModule = c("Root" = "Root",
                                             "A" = "APC",
                                             "B" = "TP53",
                                             "C" = "KRAS",
                                             "D" = "FBXW7"),
                            drvNames = c("APC", "TP53", "KRAS", "FBXW7"))

# DAG representation
plot(CRC_W2, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

Since only four genes are used in the DAG, then the possible number of genotypes is $2^4 = 16$, which are easier to interpret in a fitness landscape (see Figure 12). Now, the genotypes with the highest fitness are the ones that fulfill the order of restrictions imposed by the DAG (e.g. APC, FBXW7, KRAS - APC, FBXW7, KRAS, TP53). On the other hand, genotypes that deviates from the imposed restrictions have the lowest fitness (e.g. KRAS - FBXW7 - APC, KRAS, TP53). However, specifying mutual exclusivity with XOR relationships cannot capture null effect or synthetic lethality between APC and TP53. Also, if an AND relationship is defined from the Root to APC and TP53, then there is no change in fitness values.

```
## Simplified Model
## Map genotypes to fitness
(CRC_F2 <- evalAllGenotypes(CRC_W2, order = FALSE, addwt = TRUE))
```

```
##                   Genotype Fitness
## 1                       WT 1.00000
## 2                      APC 1.20000
## 3                    FBXW7 0.50000
## 4                     KRAS 0.50000
## 5                     TP53 1.10000
## 6               APC, FBXW7 0.60000
## 7                APC, KRAS 1.26000
## 8                APC, TP53 1.32000
## 9              FBXW7, KRAS 0.50500
## 10             FBXW7, TP53 0.55000
## 11              KRAS, TP53 1.15500
## 12       APC, FBXW7, KRAS 1.27260
## 13       APC, FBXW7, TP53 0.66000
## 14        APC, KRAS, TP53 0.66000
## 15      FBXW7, KRAS, TP53 1.16655
## 16 APC, FBXW7, KRAS, TP53 0.66660
```

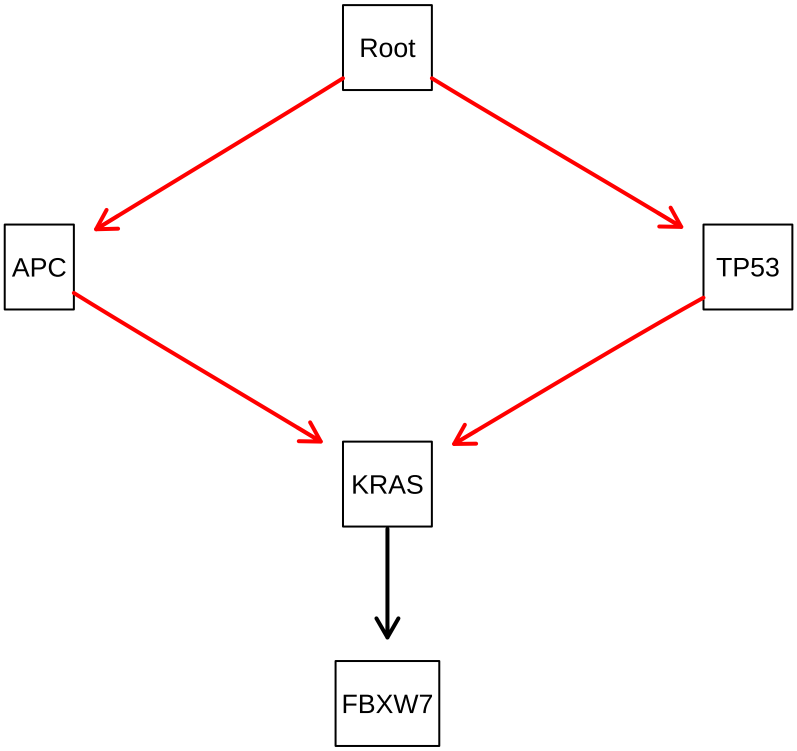

Figure 11: Simplified model from colorectal cancer DAG

```
## Plot of fitness landscap

plot(CRC_F2, use_ggrepel = TRUE)
```

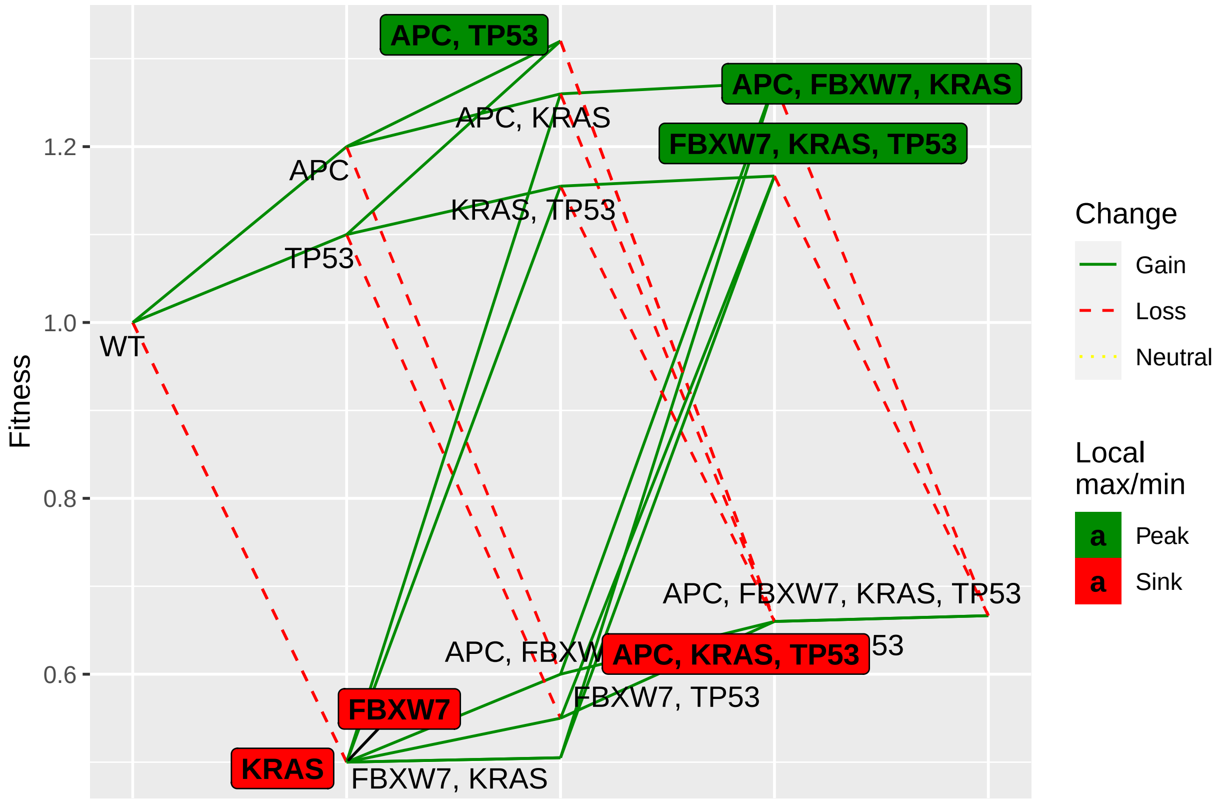

Figure 12: Fitness landscape from simplified model

4.2 Simulating Data from Simplified Model

Fitness effects and restrictions defined in the DAG from Figure 11 from previous section was used to simulate clonal evolution. The same parameters from subsection 3.2, except initSize and finalTime, were set in the OncoSimulIndiv function. Figure 13 shows the genotypes during that arises during clonal evolution. The genotype APC, TP53 fixates quickly in the clonal population. This result supports the fitness value for APC, FBXW7, KRAS depicted in Figure 12, since that genotype is one of the local maxima. A more detailed order of genotype appearances and extinctions is shown in Figure 14. Note that not all the 16 genotypes appear in the simulation because the best fitted genotype fixates rapidly in the population, leading to the extinction of some genotypes, whereas other cannot even appear. When simulation were executed with onlyCancer = TRUE, cancer is never reached, although the fixated genotype is the global maxima of the fitness landscape (see Figure 15 and Figure 16).

```
## Fix the seed for reproducibility
set.seed(87)
```

```
CRC_W2_S <- oncoSimulIndiv(CRC_W2, ## A fitnessEffects object
  model = "McFL", ## Model used
  mu = 1e-4, ## Mutation rate
  sampleEvery = 0.02, ## How often the whole population is sampled
  keepEvery = 1,
  initSize = 2000, ## Initial population size
  finalTime = 2000,
  keepPhylog = TRUE, ## Allow to see parent-child relationships
  onlyCancer = FALSE)

## Plot of simulation for genotypes
plot(CRC_W2_S,
show = "genotypes",
type = "stacked")
```

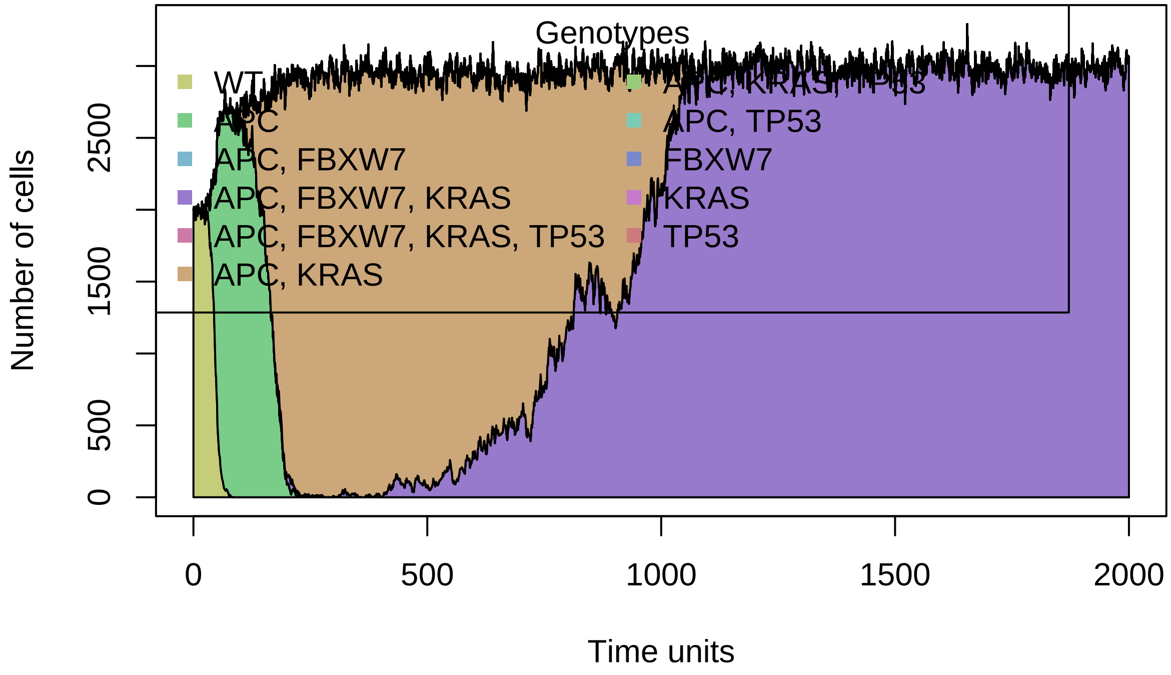

Figure 13: Simulation of cancer progression for the simplified model. Genotypes are shown stacked

```
## Plot of simulation for genotypes
plot(CRC_W2_S,
show = "genotypes",
legend.ncols = 2,
xlim = c(0, 1500),
type = "line")
```

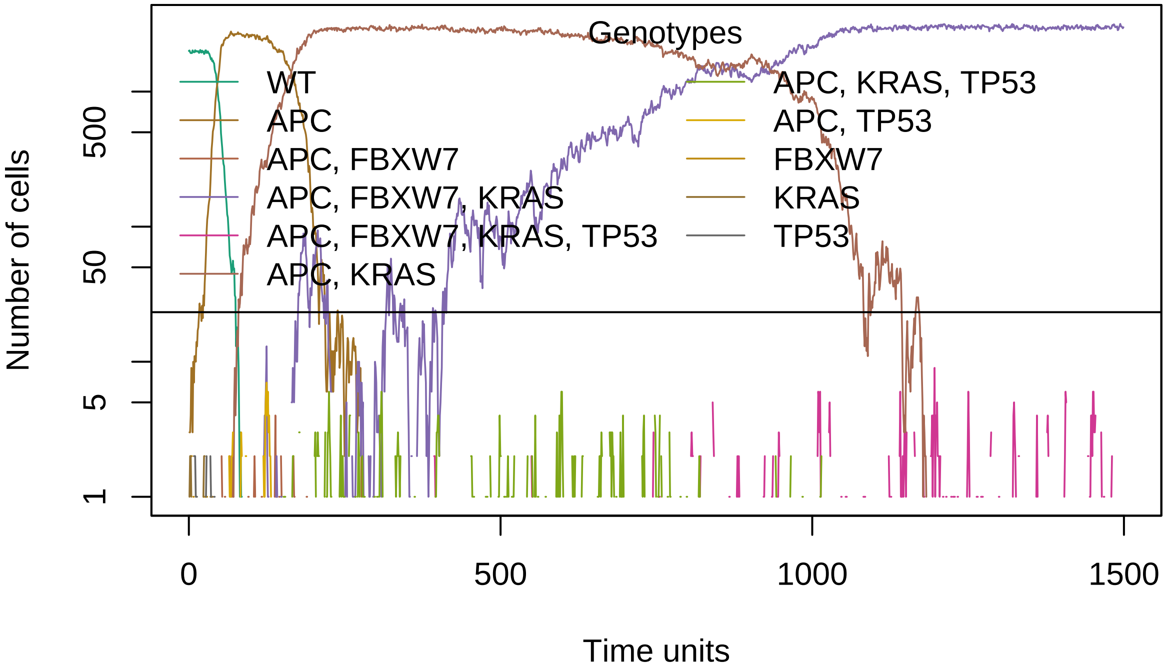

Figure 14: Simulation of cancer progression for the simplified model. Genotypes are shown as lines

```
## Fix the seed for reproducibility
set.seed(52)


CRC_W2_S1 <- oncoSimulIndiv(CRC_W2, ## A fitnessEffects object
  model = "McFL", ## Model used
  mu = 1e-4, ## Mutation rate
  sampleEvery = 0.02, ## How often the whole population is sampled
  keepEvery = 1,
  initSize = 2000, ## Initial population size
  finalTime = 800,
  keepPhylog = TRUE, ## Allow to see parent-child relationships
  onlyCancer = TRUE,
  errorHitWallTime = FALSE, ## See results even if stopping conditions are not met
  errorHitMaxTries = FALSE)
```

```
##
##  Hitted maxtries. Exiting.
```

```
## Time to reach cancer
(CRC_W2_S1$FinalTime)
```

```
## [1] 800
```

```
## Plot of simulation for genotypes
plot(CRC_W2_S1,
show = "genotypes",
legend.ncols = 2,
xlim = c(0, 500),
type = "stacked")
```

```
## Plot of simulation for genotypes
plot(CRC_W2_S1,
show = "genotypes",
legend.ncols = 1,
xlim = c(0, 300),
type = "line")
```

Figure 17 and Figure 18 shows the genealogical relationships of clones that appeared during the simulations. The number of the arrows represent the times that each clone appeared. When simulation are set to reach cancer the clones that have a genotype belonging to the two local optima appear(Figure 18). Whereas if simulation are executed without the cancer parameter, the most represented clone is the one that has the best fitted genotype (see Figure 17).

```
## Plot of genealogical relationships
plotClonePhylog(CRC_W2_S, N = 0, keepEvents = TRUE)
```

```
## Plot of genealogical relationships
plotClonePhylog(CRC_W2_S1, N = 0, keepEvents = TRUE)
```

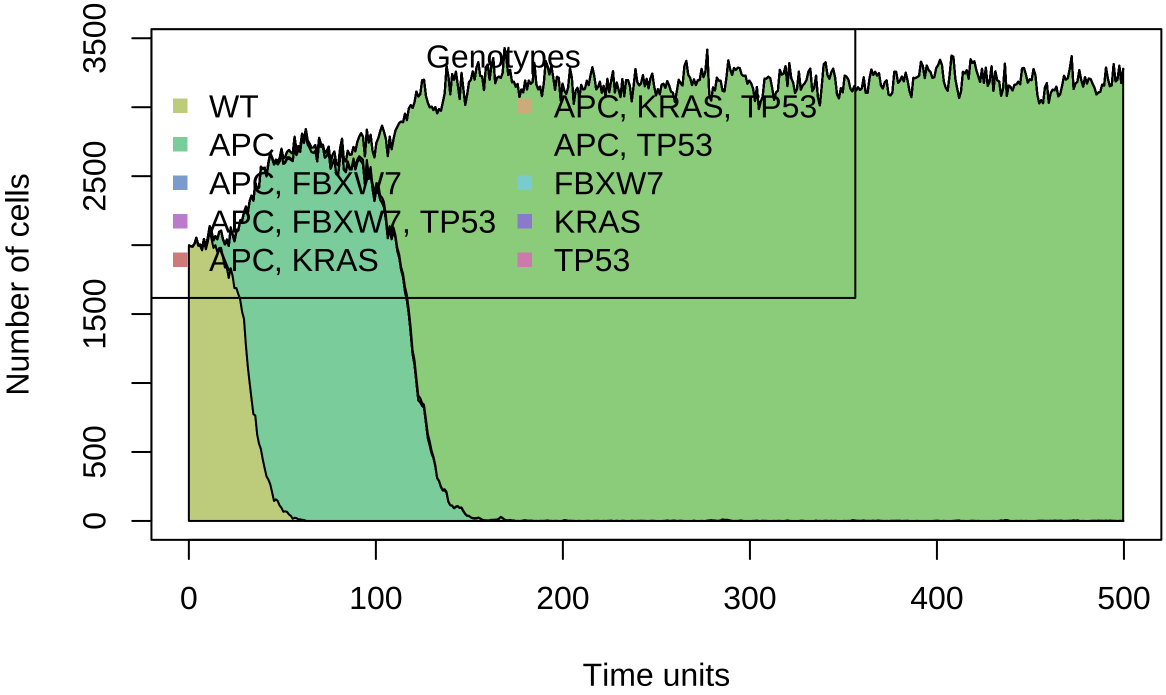

Figure 15: Simulation of cancer progression for the simplified model when onlyCancer = TRUE. Genotypes are shown stacked

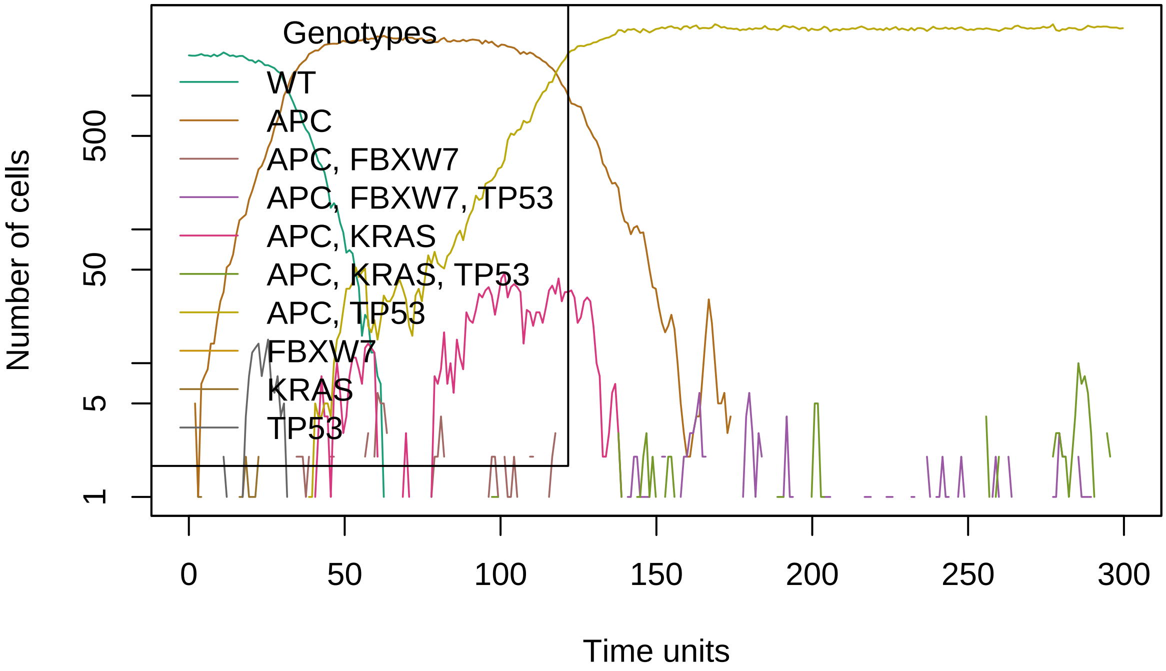

Figure 16: Simulation of cancer progression for the simplified model when onlyCancer = TRUE. Genotypes are shown as lines

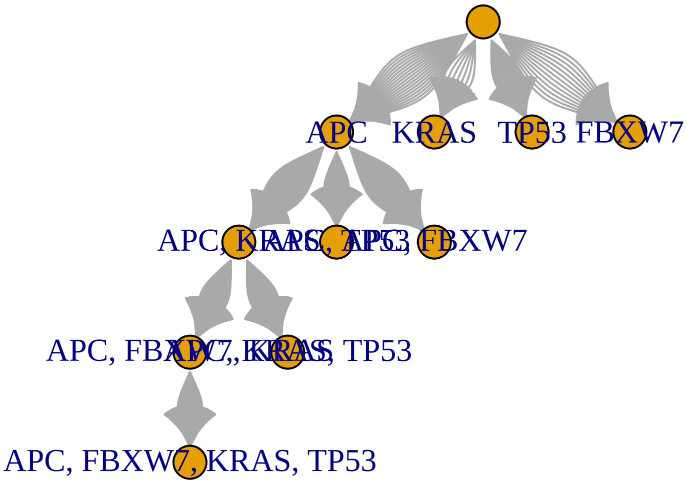

Figure 17: Genealogical relationships of clones

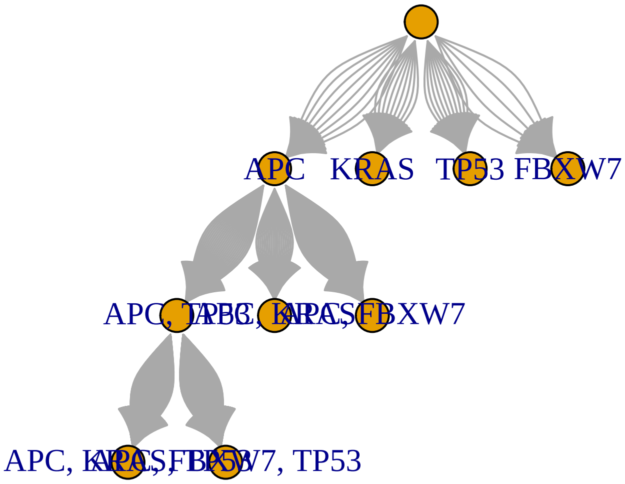

Figure 18: Genealogical relationships of clones when onlyCancer = TRUE

4.3 Synthetic Lethality

Synthetic lethality is a special type of epistasis. Therefore, we used the epistasis module inside allFitnessEffects to define an epistatic interaction between TP53 and AP (see Figure 19) and restriction imposed by the DAG (*i.e.* XOR relationships). The fitness values were assigned such that an scenario where synthetic lethality via pairwise interaction occurs.

The fitness landscape shows that the genotype for which the synthetic lethality was specified has a lower fitness value as expected, although it is not a local minima. Similarly to fitness landscape in Figure 12, the local maxima is composed by the genotyopes that satisfy both epistatic interactions and restrictions imposed. Whereas, local minima is composed by genotypes that contain genes with synthetic lethality and other genes that have top-down dependencies (see Figure 20).

```
## Simplified model
## Define poset restrictions, mapping of genes to modules, and driver genes
CRC_W3 <- allFitnessEffects(data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
                                       child = c("A", "B", rep("C", 2), "D"),
                                       s = c(0.2, 0.1, rep(0.05, 2), 0.01),
                                       sh = -0.5,
                                       typeDep = c(rep("XMPN", 4), "MN")),
                             epistasis = c("-A : B" = 0.1,
                                           "-B : A" = 0.2,
                                           "A:B" = -0.5),
                            geneToModule = c("Root" = "Root",
                                             "A" = "APC",
                                             "B" = "TP53",
                                             "C" = "KRAS",
                                             "D" = "FBXW7"),
                            drvNames = c("APC", "TP53", "KRAS", "FBXW7"))

# DAG representation
plot(CRC_W3, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

```
## Map genotypes to fitness
CRC_F1 <- evalAllGenotypes(CRC_W3, order = FALSE, addwt = TRUE)

(CRC_F1)
```

```
##                   Genotype  Fitness
## 1                       WT 1.000000
## 2                      APC 1.440000
## 3                    FBXW7 0.500000
## 4                     KRAS 0.500000
## 5                     TP53 1.210000
## 6               APC, FBXW7 0.720000
## 7                APC, KRAS 1.512000
## 8                APC, TP53 0.660000
## 9              FBXW7, KRAS 0.505000
## 10             FBXW7, TP53 0.605000
## 11              KRAS, TP53 1.270500
## 12        APC, FBXW7, KRAS 1.527120
## 13        APC, FBXW7, TP53 0.330000
## 14         APC, KRAS, TP53 0.330000
```

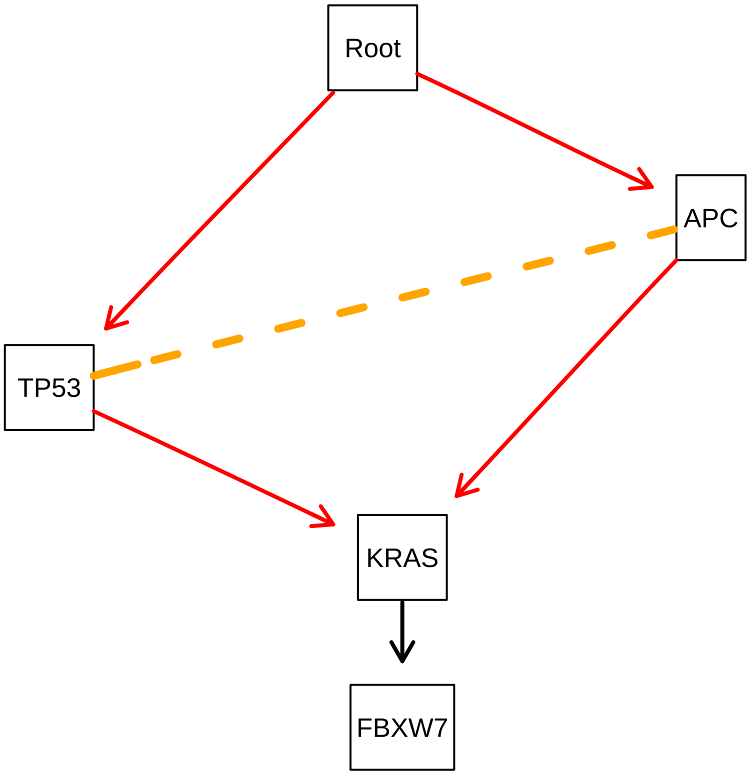

Figure 19: DAG with synthetic lethality

```
## 15       FBXW7, KRAS, TP53 1.283205
## 16 APC, FBXW7, KRAS, TP53 0.333300
```

```r
## Plot of fitness landscape

plot(CRC_F1, use_ggrepel = TRUE)
```

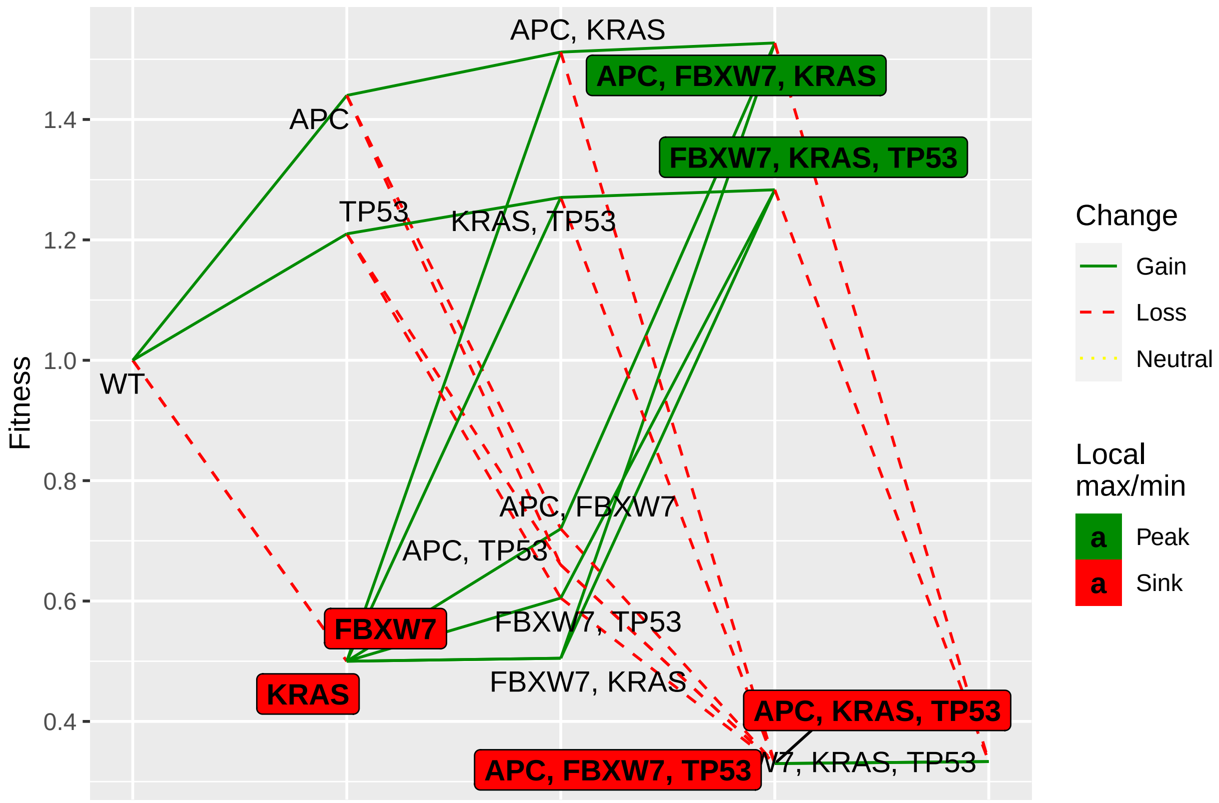

Figure 20: Fitness landscape inferred from simplified DAG with synthetic lethality

In order to simulate synthetic lethality via three-way interaction, we set fitness values that reflect slightly deleterious effect (if two genes appear) or a highly deleterious effect (if three genes appear). Figure 21 shows the DAG derived for the three-way interaction between APC, TP53, and KRAS. The inferred fitness landscape shows that the global minima is composed by the genotype that carries the synthetic lethality. Whereas, local maxima is composed by genotypes that follow the restrictions imposed in the DAG. Also, note that the global maxima is APC. This is not surprising given that APC is an earlier mutation and has the highest fitness values compared to other genes/genotypes (see Figure 22).

```r
## Simplified model
## Define poset restrictions, mapping of genes to modules, and driver genes
CRC_W4 <- allFitnessEffects(data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
                                        child = c("A", "B", rep("C", 2), "D"),
                                        s = c(0.2, 0.1, rep(0.05, 2), 0.01),
                                        sh = -0.5,
                                        typeDep = c(rep("XMPN", 4), "MN")),
                             epistasis = c("A : -B : -C" = 0.2,
                                             "-A : B : -C" = 0.1,
                                             "-A : -B : C" = 0.05,
                                             "A : B : -C" = 0.01,
                                             "-A : B : C" = 0.02,
                                             "-B : A : C" = 0.02,
                                             "A : B : C" = -0.5),
                              geneToModule = c("Root" = "Root",
                                               "A" = "APC",
                                               "B" = "TP53",
                                               "C" = "KRAS",
                                               "D" = "FBXW7"),
                              drvNames = c("APC", "TP53", "KRAS", "FBXW7"))

# DAG representation
plot(CRC_W4, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

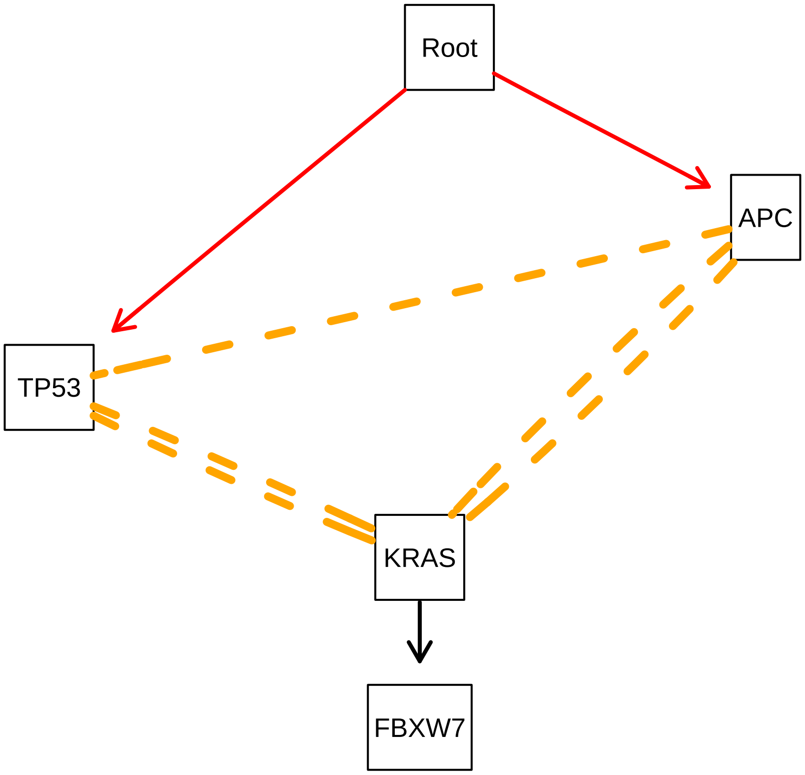

Figure 21: DAG with synthetic lethality (three-way interaction)

```
## Map genotypes to fitness
CRC_F2 <- evalAllGenotypes(CRC_W4, order = FALSE, addwt = TRUE)

(CRC_F2)
```

```
##                    Genotype  Fitness
## 1                       WT 1.000000
## 2                      APC 1.440000
## 3                    FBXW7 0.500000
## 4                     KRAS 0.525000
## 5                     TP53 1.210000
## 6               APC, FBXW7 0.720000
## 7                APC, KRAS 1.285200
## 8                APC, TP53 1.333200
## 9              FBXW7, KRAS 0.530250
## 10             FBXW7, TP53 0.605000
## 11              KRAS, TP53 1.178100
## 12       APC, FBXW7, KRAS 1.298052
## 13       APC, FBXW7, TP53 0.666600
## 14        APC, KRAS, TP53 0.330000
## 15      FBXW7, KRAS, TP53 1.189881
## 16 APC, FBXW7, KRAS, TP53 0.333300
```

```r
## Plot of fitness landscape

plot(CRC_F2, use_ggrepel = TRUE)
```

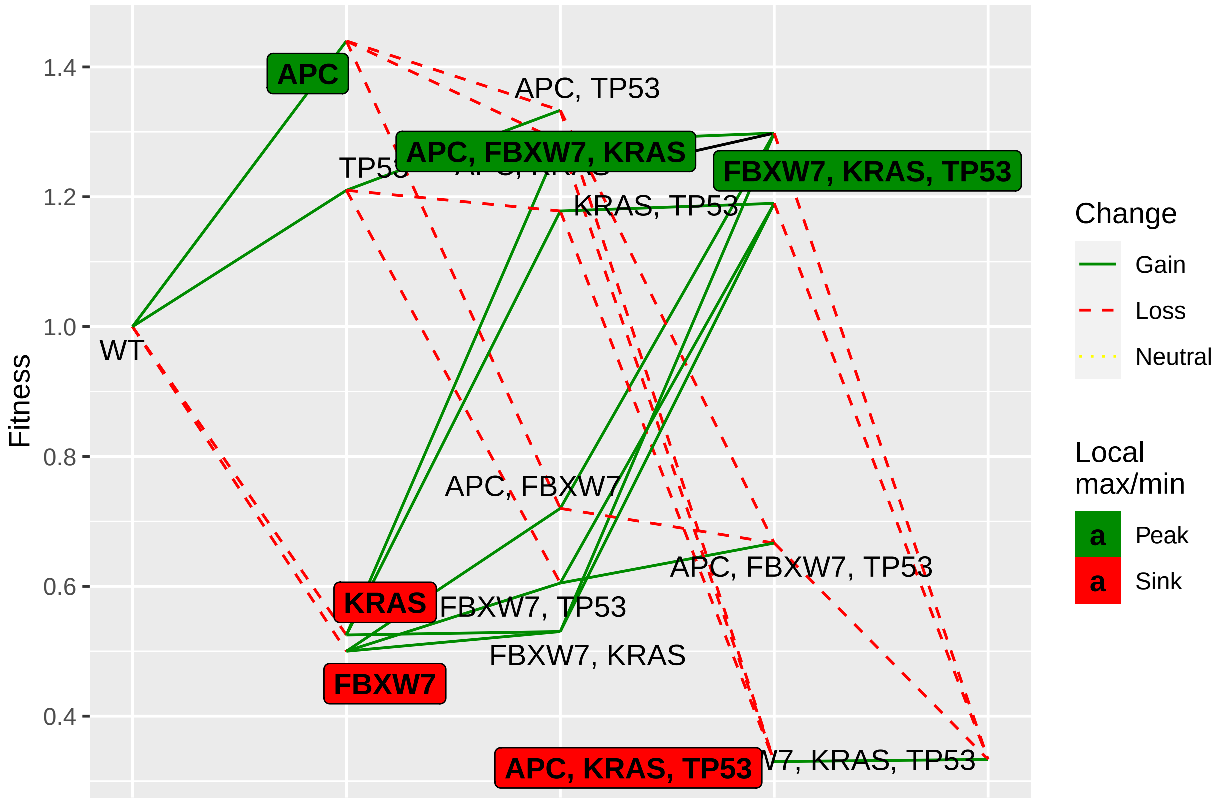

Figure 22: Fitness landscape inferred from simplified DAG with synthetic lethality (three-way interaction)

4.4 Synthetic Viability

Synthetic viability is specified for genotype APC, TP53 (see Figure 23). Here the genotypes composed only by APC or TP53 are deleterious. Figure 24 shows the fitness landscape for synthetic viability via pairwise interaction. Note that the global maxima is composed by the genotype that contains all genes. On the other hand, local minima is composed by genotypes that contains one gene that has a deleterious effect. Note that despite the lower fitness value of genotype FBXW7, KRAS, it conforms a local maxima, although the restrictions imposed in the DAG are not completely satisfied. Moreover, in this fitness landscape, the global maxima may not be reached because the mutational paths required lead to a region composed of multiplpe valleys. It important to mention that order of effects could provide a more realistic fitness landscape. For example, a possible path that leads to the global maxima requires a mutation in KRAS before a mutation in FBXW7.

```
## Simplified model
## SM because synthetic viability requires both parent nodes.
## Define poset restrictions, mapping of genes to modules, and driver genes
CRC_W5 <- allFitnessEffects(data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
                                       child = c("A", "B", rep("C", 2), "D"),
                                       s = c(0.2, 0.1, rep(0.05, 2), 0.01),
                                       sh = -0.5,
                                       typeDep = c(rep("MN", 5))),
                             epistasis = c("-A : B" = -0.2,
                                           "-B : A" = -0.3,
                                           "A:B" = 0.5),
                            geneToModule = c("Root" = "Root",
                                             "A" = "APC",
                                             "B" = "TP53",
                                             "C" = "KRAS",
                                             "D" = "FBXW7"),
                            drvNames = c("APC", "TP53", "KRAS", "FBXW7"))

# DAG representation
plot(CRC_W5, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

```
## Map genotypes to fitness
CRC_F3 <- evalAllGenotypes(CRC_W5, order = FALSE, addwt = TRUE)
(CRC_F3)
```

```
##                  Genotype Fitness
## 1                      WT 1.00000
## 2                     APC 0.84000
## 3                   FBXW7 0.50000
## 4                    KRAS 0.50000
## 5                    TP53 0.88000
## 6              APC, FBXW7 0.42000
## 7               APC, KRAS 0.42000
## 8               APC, TP53 1.98000
## 9             FBXW7, KRAS 0.50500
## 10            FBXW7, TP53 0.44000
## 11             KRAS, TP53 0.44000
## 12      APC, FBXW7, KRAS 0.42420
## 13      APC, FBXW7, TP53 0.99000
## 14       APC, KRAS, TP53 2.07900
```

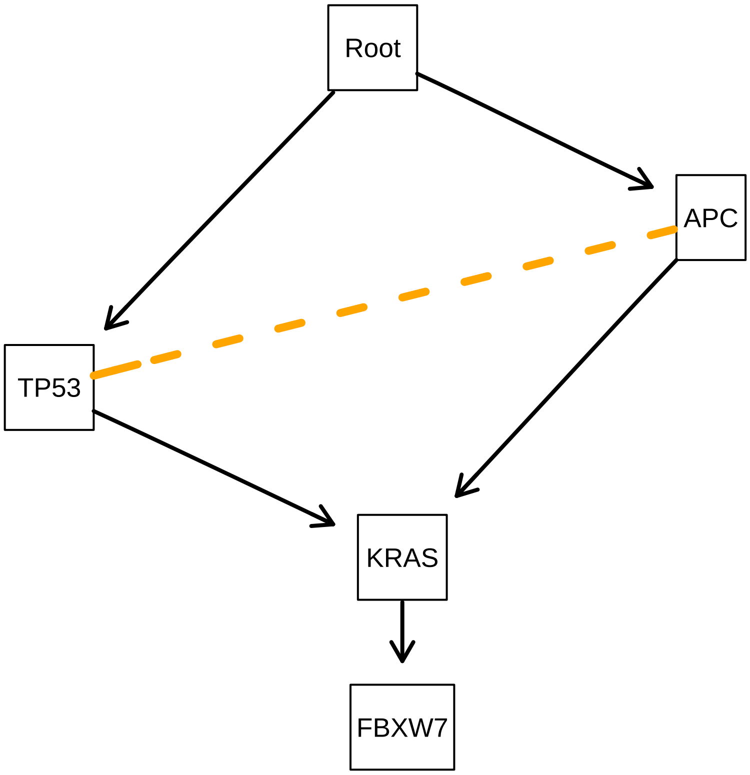

Figure 23: DAG with synthetic viability (pairwise interaction)

```
## 15      FBXW7, KRAS, TP53 0.44440
## 16 APC, FBXW7, KRAS, TP53 2.09979
```

```
## Plot of fitness landscape

plot(CRC_F3, use_ggrepel = TRUE)
```

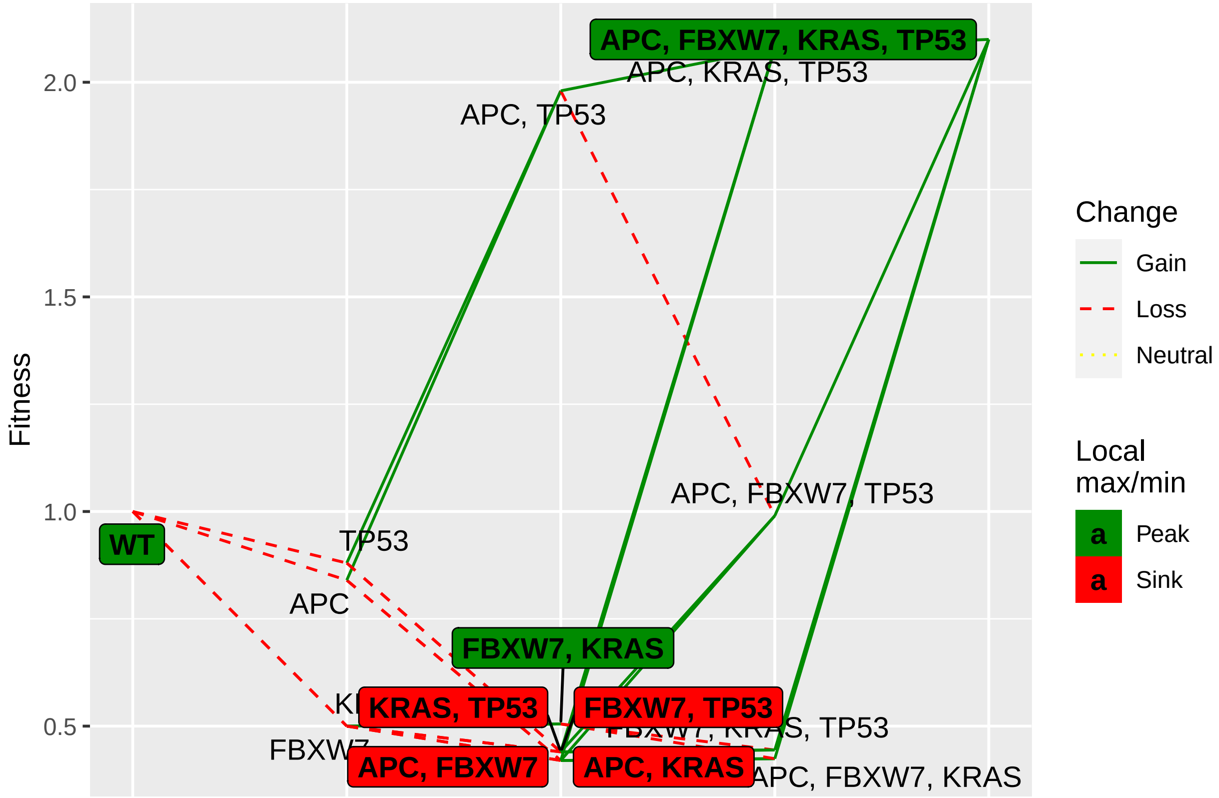

Figure 24: Fitness landscape inferred from simplified DAG with synthetic viability

Figure 25 shows synthetic viability with a three-way interaction between APC, TP53, and KRAS. For this, we specified highly deleterious effects if APC, TP53, or KRAS appear independently. Whereas, slightly deleterious effects were set if two of those genes appear in a genotype. The fitness landscape for this scenario (see Figure 26) shows the order of restrictions and epistatic interactions set lead to the global maxima composed by the genotype APC, TP53, KRAS, FBXW7. This result support the idea that DAGs are better suited to represent sign epistasis (5). Nevertheless, as mentioned above, including order of effects can give a more realistic fitness values associated to genotypes.

In this work, we have represented synthetic lethality via pairwise and three-way interactions. However, this can be achieved if the DAG is composed by individual genes instead of modules because modules does not allow to define epistatic relationships between genes of the same module. This is important because genes of the same module can participate in the same pathway, as discussed in (1,2).

```
## Simplified model
## SM because synthetic viability requires both parent nodes.
## Define poset restrictions, mapping of genes to modules, and driver genes
```

```
CRC_W6 <- allFitnessEffects(data.frame(parent = c(rep("Root", 2), "A", "B", "C"),
                                      child = c("A", "B", rep("C", 2), "D"),
                                      s = c(0.2, 0.1, rep(0.05, 2), 0.01),
                                      sh = -0.5,
                                      typeDep = c(rep("MN", 5))),
                            epistasis = c("A : -B : -C" = -0.2,
                                          "-A : B : -C" = -0.2,
                                          "-A : -B : C" = -0.3,
                                          "A : B : -C" = -0.05,
                                          "-A : B : C" = -0.01,
                                          "A : -B : C" = -0.01,
                                          "A : B : C" = 0.5),
                           geneToModule = c("Root" = "Root",
                                            "A" = "APC",
                                            "B" = "TP53",
                                            "C" = "KRAS",
                                            "D" = "FBXW7"),
                           drvNames = c("APC", "TP53", "KRAS", "FBXW7"))

# DAG representation
plot(CRC_W6, expandModules = TRUE, autofit = TRUE, lwdf = 2)
```

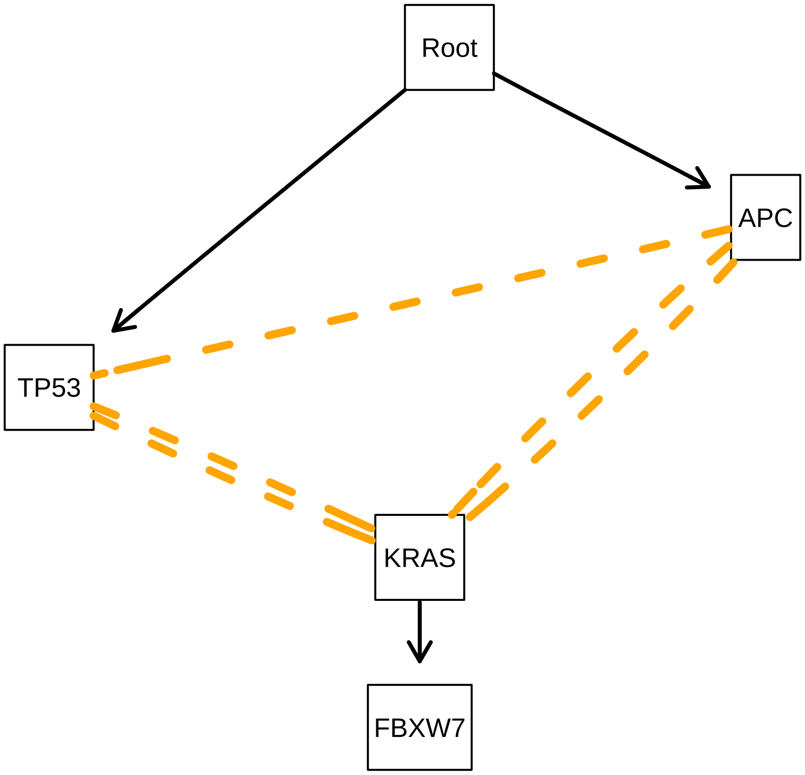

Figure 25: DAG with synthetic viability (three-way interaction)

```
## Map genotypes to fitness
CRC_F4 <- evalAllGenotypes(CRC_W6, order = FALSE, addwt = TRUE)
(CRC_F4)
```

```
##                   Genotype  Fitness
## 1                       WT 1.000000
## 2                      APC 0.960000
## 3                    FBXW7 0.500000
## 4                     KRAS 0.350000
## 5                     TP53 0.880000
## 6               APC, FBXW7 0.480000
## 7                APC, KRAS 0.594000
## 8                APC, TP53 1.254000
## 9              FBXW7, KRAS 0.353500
## 10             FBXW7, TP53 0.440000
## 11              KRAS, TP53 0.544500
## 12        APC, FBXW7, KRAS 0.599940
## 13        APC, FBXW7, TP53 0.627000
## 14         APC, KRAS, TP53 2.079000
## 15       FBXW7, KRAS, TP53 0.549945
## 16 APC, FBXW7, KRAS, TP53 2.099790
```

```
## Plot of fitness landscape

plot(CRC_F4, use_ggrepel = TRUE)
```

5 A Probabilistic Model of Mutually Exclusive Linearly Ordered Driver Pathways

Mohaghegh Neyshabouri et al. (11) propose a probabilistic model of mutually exclusive linearly ordered driver pathways and analyze one large dataset of colorectal adenocarcinoma (COADREAD) from IntOGen-mutations database. Their model assumes driver genes are over-represented among those mutated across a large tumor collection and, thus, they can be identified in terms of frequency. Also, those participating of the same pathway are mutated in a mutually exclusive manner because more than one mutation in a pathway does not give any selective advantage to the clone.

Like with previous generative models, we map the COADREAD generative model to an actual evolutionary model using different **OncoSimulR** functionalities. This time, we extent what authors model using the frequency-dependent fitness specification, to illustrate how differently fitness landscapes evolve even though they are built from exact CPMs when we consider this additional evolutionary event

Figure 7.C from (11) shows the CPM inferred from the COADREAD dataset, consisting of seven modules with between 1 to 4 genes each. The model clearly reconstructs the well-known initiator events in colorectal cancer, including mutations in APC, TP53 and KRAS (19). Using the DAG of restrictions as starting point[1], the evolutionary model is created specifying same genotype fitness for all modules as authors do not state any differences in fitness for when the restrictions in the DAG are satisfied (s). However, based on the confidence parameter used by the authors to assess the reliability of modeled restrictions, different fitness are set when the DAG of restrictions is not satisfied (sh) (Table 2). Since this method reconstructs linear models (*i.e.* oncogenic trees), there is no need to specify any particular type of dependency between modules

[1] Several genes were removed from the original set in order to get a clear fitness landscape

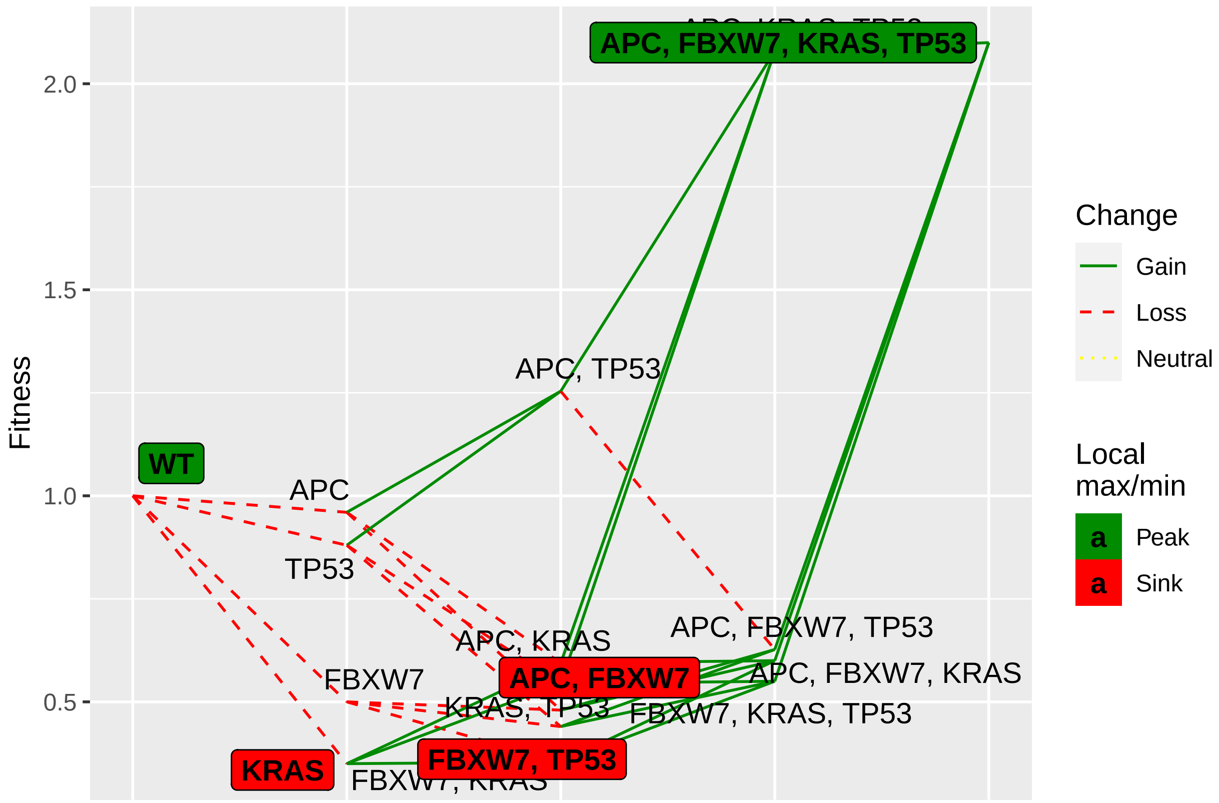

Figure 26: Fitness landscape inferred from simplified DAG with synthetic viability (three-way interaction)

(typeDep), so we set it to monotonic (MN) as it is a mandatory argument for allFitnessEffects function. Figure 27 shows the DAG of restrictions created with `allFitnessEffects`, recapitulating the poset inferred in (11).

Table 2: Confidence parameter for each module transition

Module	Confidence parameter (%)
APC	100
TP53	100
KRAS	100
PIK3CA, NRAS, LRP1B	100
FBXW7, TCF7L2, FAT4, ARID1A	87.7
ATM, SMAD2, ERBB3, MTOR, CTNNB1	86.9
SOX9, SMAD4	66.7

```r
## Restriction table, including DAG of restrictions
## specifications and associated fitness
COADREAD_rT <- data.frame(
              parent = c("Root", "A", "B", "C", "D", "E", "F"), # Parent nodes
              child = c("A", "B", "C", "D", "E", "F", "G"), # Child nodes
                             s = 0.5,
                             sh = c(rep(-1, 4), rep(-.5, 2), -.2),
                             typeDep = "MN")


## Create fitness specifications from DAG of restrictions considering modules
COADREAD_fitness <- allFitnessEffects(
              COADREAD_rT,
              geneToModule = c(
                 "Root" = "Root",
                 "A" = "APC",
                 "B" = "TP53",
                 "C" = "KRAS",
                 "D" = "PIK3CA, NRAS",
                 "E" = "FBXW7, ARID1A",
                 "F" = "ATM, SMAD2",
                 "G" = "SOX9, SMAD4")) # Modules


## DAG of restrictions representation
plot(COADREAD_fitness, expandModules = TRUE, autofit = TRUE)
```

```r
## Evaluation of all possible genotypes fitness
## under the previous fitness specifications
COADREAD_FL <- evalAllGenotypes(COADREAD_fitness, max = 131072)
```

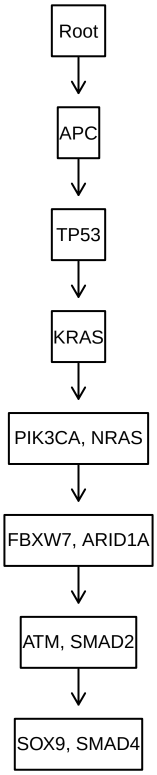

Figure 27: DAG of restrictions for the COADREAD dataset

```r
## Fitness landscape representation
plotFitnessLandscape(COADREAD_FL)
```

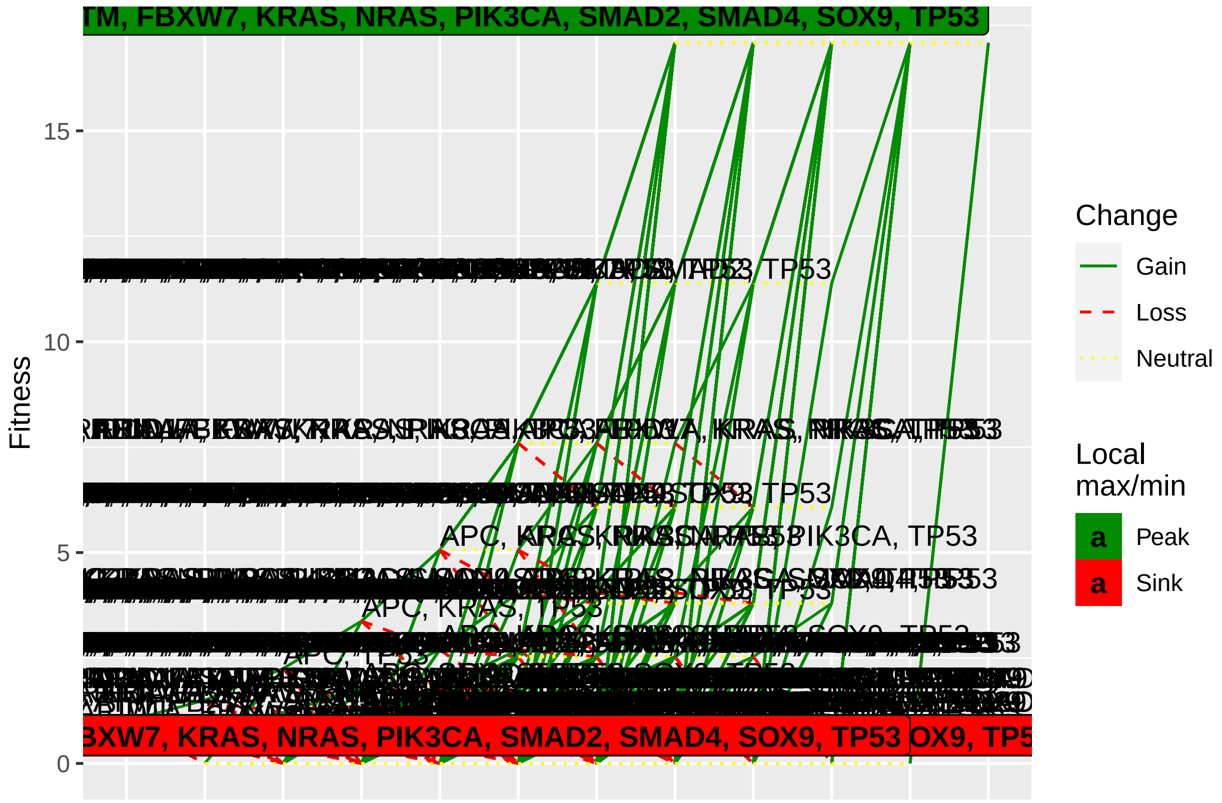

Figure 28: Fitness landscape corresponding with the DAG of restrictions for the COADREAD dataset

Figure 28 plots all possible genotypes in a very busy fitness landscape. Although it is not possible to visualize genotypes clearly, we can see an exponential trend towards local maxima, corresponding to the clones carrying seven driver-genes genotypes. Stands out how very different genotypes behave the same due to the mutual exclusivity effect, which is probably simplifying much more complex interactions among genes, specially the effect in fitness different genes from the same module could have if we didn't account for the mutual exclusivity phenomenon. In next section we discuss genotypes distribution with a simplified model.

5.1 Simplified Cancer Progression Model

As we did with previous models, for illustrating purposes we designed a simplified version of the CPM by (11) to assess its reliability when considering other evolutionary scenarios. This time, we take the first four modules of the complete DAG (maintaining same fitness parameters), being the fourth the only one carrying two genes and, thus, affected by mutual exclusivity circumstance (Figure 29).

The fitness landscape shown in Figure 30 includes some of the possible genotypes. Clearer than in Figure 28, we see fitness increases exponentially following the restrictions established in the DAG. Local minima (red squares) correspond to genotypes violating these constraints, and gain changes (green lines) are consistent with satisfied restrictions. However, one of the local maxima genotypes (green squares) corresponds to a genotype violating mutual exclusivity (PIK3CA and NRAS appear together). This is explained because the

Modules functionality in OncoSimulR sets fitness to zero when genes from the same module mutate at a time as there is a null effect because both genes participate from the same pathway. Because there is not a decrease in the fitness of the genotype (*i.ei* no deleterious effect), it can still be a local maxima. Yet, it is unlikely a genotype needed of an extra mutation to compete with the other maximum-fitness genotypes will survive long during cancer progression to fixate.

```
## Restriction table, including five-modules DAG of
## restrictions specifications and associated fitness
COADREAD_rT_5d <- data.frame(parent = c("Root", "A", "B", "C"), # Parent nodes
                             child = c("A", "B", "C", "D"), # Child nodes
                             s = 0.5,
                             sh = c(rep(-1, 4)),
                             typeDep = "MN")

## Create fitness specifications from simplified DAG of restrictions
COADREAD_fitness_5d <- allFitnessEffects(COADREAD_rT_5d,
                                         geneToModule = c( "Root" = "Root",
                                                           "A" = "APC",
                                                           "B" = "TP53",
                                                           "C" = "KRAS",
                                                           "D" = "PIK3CA, NRAS"),
                                         drvNames = c("APC", "TP53", "KRAS",
                                                      "PIK3CA", "NRAS"))

## Simplified DAG of restrictions representation
plot(COADREAD_fitness_5d, expandModules = TRUE, autofit = TRUE)
```

```
## Evaluation of all possible genotypes fitness under the previous fitness specifications
(COADREAD_FL_5d <- evalAllGenotypes(COADREAD_fitness_5d))
```

```
##                         Genotype Fitness
## 1                            APC  1.5000
## 2                           KRAS  0.0000
## 3                           NRAS  0.0000
## 4                         PIK3CA  0.0000
## 5                           TP53  0.0000
## 6                      APC, KRAS  0.0000
## 7                      APC, NRAS  0.0000
## 8                    APC, PIK3CA  0.0000
## 9                      APC, TP53  2.2500
## 10                    KRAS, NRAS  0.0000
## 11                  KRAS, PIK3CA  0.0000
## 12                    KRAS, TP53  0.0000
## 13                  NRAS, PIK3CA  0.0000
## 14                    NRAS, TP53  0.0000
## 15                  PIK3CA, TP53  0.0000
## 16               APC, KRAS, NRAS  0.0000
## 17             APC, KRAS, PIK3CA  0.0000
## 18               APC, KRAS, TP53  3.3750
## 19             APC, NRAS, PIK3CA  0.0000
## 20               APC, NRAS, TP53  0.0000
## 21             APC, PIK3CA, TP53  0.0000
## 22            KRAS, NRAS, PIK3CA  0.0000
```

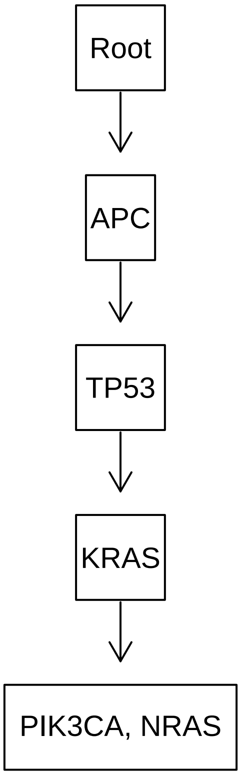

Figure 29: Simplified DAG of restrictions for the COADREAD dataset

```
## 23              KRAS, NRAS, TP53  0.0000
## 24            KRAS, PIK3CA, TP53  0.0000
## 25            NRAS, PIK3CA, TP53  0.0000
## 26       APC, KRAS, NRAS, PIK3CA  0.0000
## 27         APC, KRAS, NRAS, TP53  5.0625
## 28       APC, KRAS, PIK3CA, TP53  5.0625
## 29       APC, NRAS, PIK3CA, TP53  0.0000
## 30      KRAS, NRAS, PIK3CA, TP53  0.0000
## 31 APC, KRAS, NRAS, PIK3CA, TP53  5.0625
```

```
## Fitness landscape representation
plotFitnessLandscape(COADREAD_FL_5d, use_ggrepel = TRUE)
```

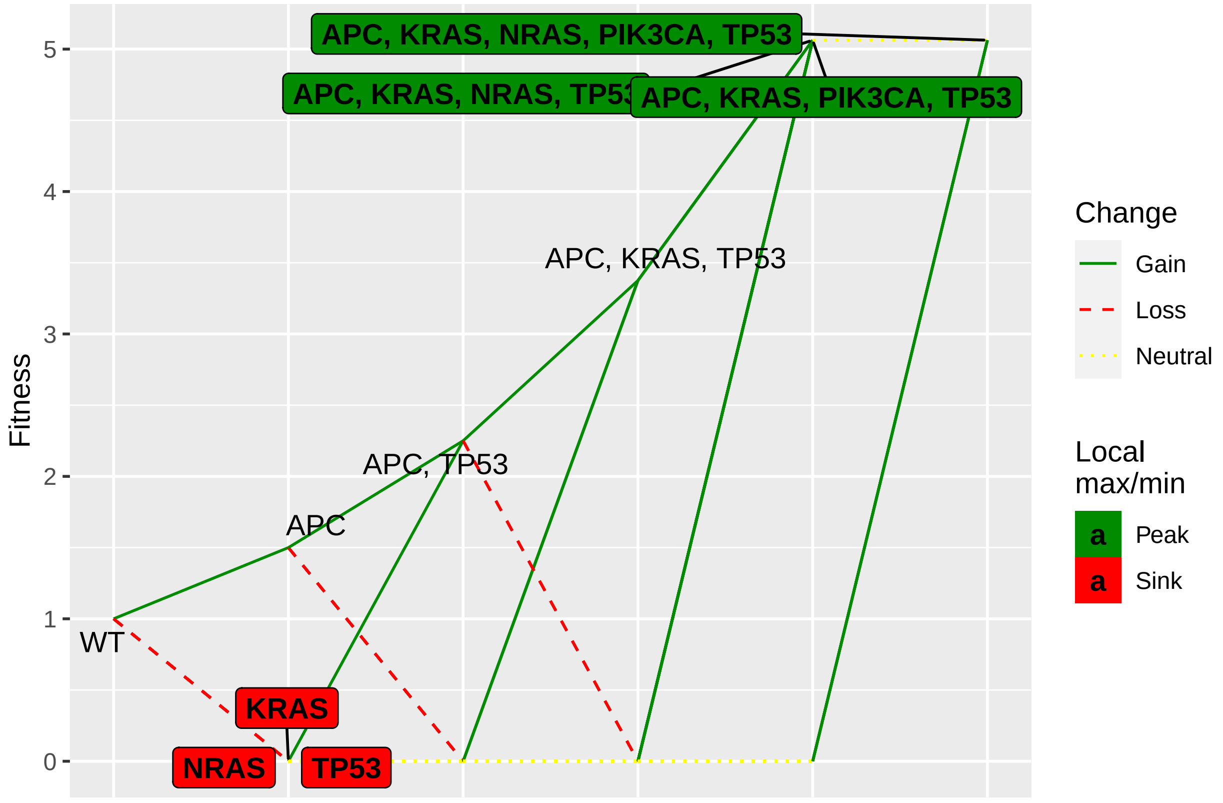

Figure 30: Fitness landscape corresponding with the simplified DAG of restrictions for the COADREAD dataset

The previous assumption can be better addressed simulating tumor progression under the proposed evolutionary model. Accordingly, we would expect clones to evolve following the optimal pathway of the fitness landscape, that is satisfying the DAG of restrictions. We use `oncoSimulPop` function to simulate tumor progression in ten different individuals and test whether the genotype APC, TP53, KRAS, PIK3CA, NRAS would ever fixate even at a very small frequency. Parameters are adjusted appropriately so that simulation stops when APC, TP53, KRAS, PIK3CA, NRAS genotype is reached at a frequency of 0.1 at least ("onlyCancer = TRUE, fixation = c(c("_, APC, TP53, KRAS, PIK3CA, NRAS"), fixation_tolerance = 0.9)").

Figure 31 shows a representative stacked plot of the genotype abundance over time. Surprisingly, APC, TP53, KRAS, PIK3CA, NRAS genotype appeared in the ten simulations significantly (Data not shown), able to coexist with clones of the same fitness. Often, it arises from APC, TP53, KRAS, PIK3CA or APC, TP53, KRAS, NRAS clones, which appear together even less frequently, probably because both descend from the same ancestor and have much higher fitness, leading to its extinction once one of the two emerges. Figure 32 supports this idea as it clearly plots the behavior of each clone separately, displaying abrupt extinctions the moment the following clone with higher fitness appears. Retrieving the phylogeny of the clones (representative graph of one simulation in Figure 33), we confirm there is a tendency of the clones to mutate until the complete genotype APC, TP53, KRAS, PIK3CA, NRAS, either following the linear mutual exclusivity path or from a more diverse landscape.

```
set.seed(125)

## Simulate cancer progression in 10 individuals until APC,
## TP53, KRAS, PIK3CA, NRAS genotype fixates
COADREAD_Simul_5d <- oncoSimulPop(
               10, COADREAD_fitness_5d,
               model = "McFL", ## Model used
               mu = 1e-4, ## Mutation rate
               sampleEvery = 0.02, ## How often the whole population is sampled
               keepEvery = 1,
               initSize = 200, ## Initial population size
               finalTime = 2000,
               keepPhylog = TRUE, ## Allow to see parent-child relationships
               onlyCancer = TRUE,
               detectionSize = NA,
               fixation = c(c("_, APC, TP53, KRAS, PIK3CA, NRAS"),
                            fixation_tolerance = 0.7),
               detectionDrivers = NA,
               detectionProb = NA,
               max.num.tries = 500,
               max.wall.time = 20,
               errorHitMaxTries = TRUE)
```

```
## You are running Windows. Setting mc.cores = 1
```

```
## Plot of simulation for genotypes
plot(COADREAD_Simul_5d[[3]],
show = "genotypes",
type = "stacked",
ylim = c(0, 35000))
```

```
plot(COADREAD_Simul_5d[[5]],
     show = "genotypes",
     type = "line",
     ylim = c(1, 100000000))
```

```
## Parent-child relationship derived from simulation
plotClonePhylog(COADREAD_Simul_5d[[3]],
                N = 0, ## Specify clones that exist
                keepEvents = TRUE ## Arrows showing how many times each clones appeared
)
```

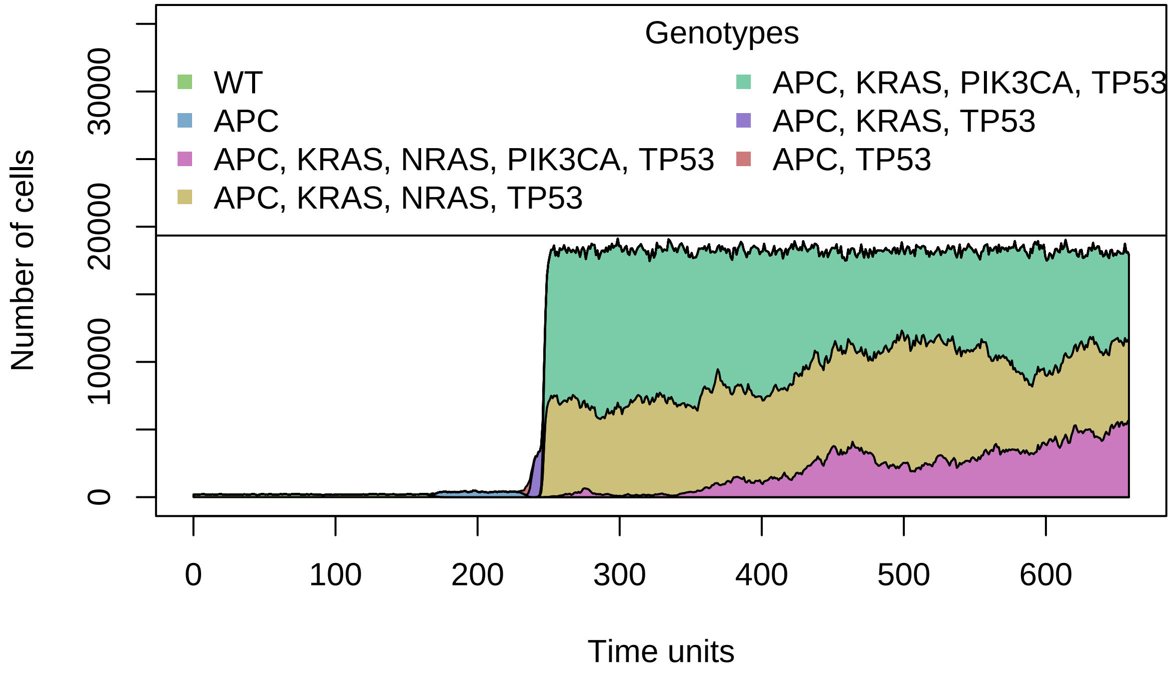

Figure 31: One of the 10 simulations of cancer progression using the four-modules fitness landscape until APC, TP53, KRAS, PIK3CA, NRAS genotype arises (stacked plot)

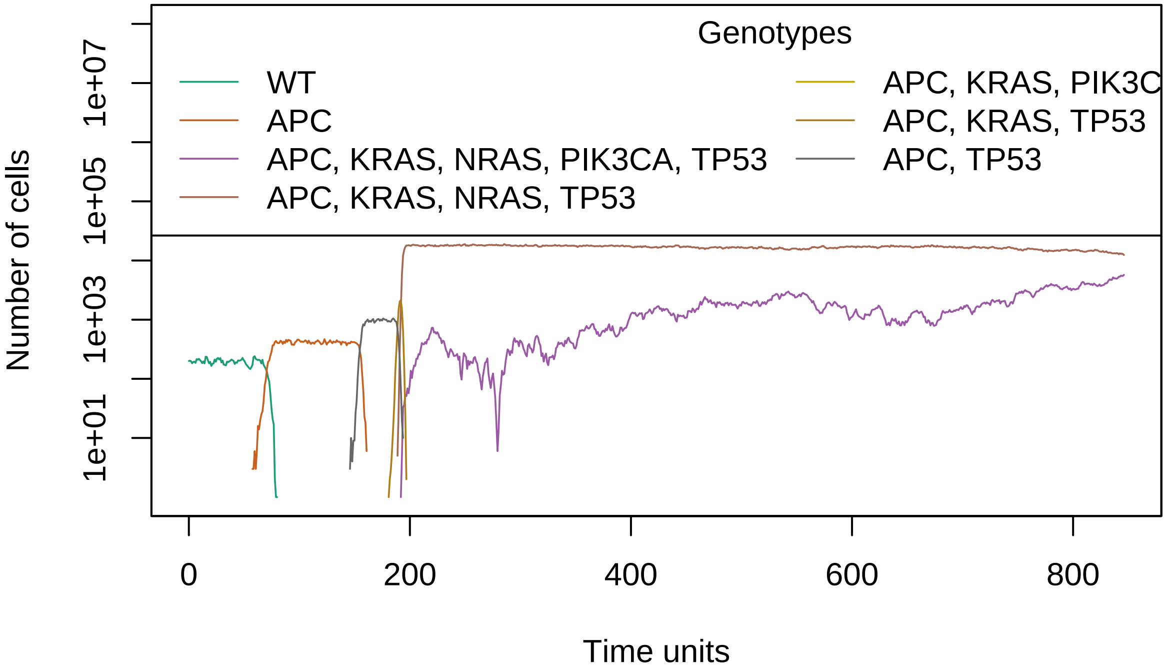

Figure 32: One of the 10 simulations of cancer progression using the four-modules fitness landscape until APC, TP53, KRAS, PIK3CA, NRAS genotype arises (line plot)

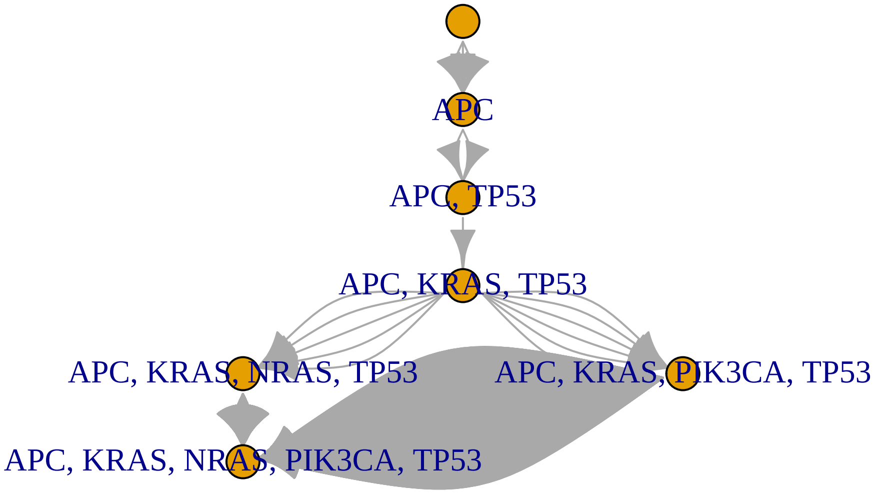

Figure 33: Parent-child relationship derived from one of the 10 simulations

5.2 Frequency-dependent Fitness

As previously introduced, clones that coexist in a tumor can influence the fitness of each other in a frequency-dependent manner when a mutation produces a phenotype able to modulate the tumor microenvironment. OncoSimulR incorporates the frecuencyDependentFitness specification to allow modelling interactions among clones during tumor progression. In the simulations ran with the simplified model, we significantly observed APC, TP53, KRAS, PIK3CA, NRAS genotype in the final stage of the simulation, an unexpected event considering PIK3CA and NRAS are mutually exclusive. This might denote there is an additional evolutionary event directing the very infrequent coexistence of these two genes in tumor samples (11). Here, we propose APC, TP53, KRAS, PIK3CA, NRAS genotype fitness could depend on the frequency of APC, TP53, KRAS, PIK3CA and APC, TP53, KRAS, NRAS genotypes in the context of a competitive relationship among clones for niche nutrients. APC, TP53, KRAS, PIK3CA, NRAS clones would be more energetically demanding and thus, coexistence with other clones would be detrimental (considering the three clones have same fitness).

To use the frecuencyDependentFitness functionality, it is necessary to set to TRUE de frequencyDependentFitness parameter in the allFitnessEffects function, as well as providing a “mapping of genotypes to fitness” data frame. Fitness values are taken from the fitness specifications previously used (subsection 5.1). To evaluate genotypes with the evalAllGenotypes function is mandatory the parameter spPopSizes to build a fitness landscape in accordance with the size of the different clones. Here, we define a scenario in which all the clones except for the four-genes and five-genes clones are almost extincted. In this context, we already see in the fitness landscape (Figure 34) an expected fitness decreased in APC, TP53, KRAS, PIK3CA, NRAS genotype, which is not a local maxima anymore.

```
## Mapping of genotypes to frequency-dependent fitness
# Not explicitly mapped genotypes are assigned a fitness of cero
COADREAD_gen_freqdep <- data.frame(
                Genotype = c("WT", "APC","APC, TP53",
                            "APC, TP53, KRAS",
                            "APC, TP53, KRAS, PIK3CA",
                            "APC, TP53, KRAS, NRAS",
                            "APC, TP53, KRAS, PIK3CA, NRAS"),
                            Fitness = c("1", "1.5",
                                        "2.25", "3.375", "5.0625", "5.0625",
              "5.0625 - ((f_APC_TP53_KRAS_PIK3CA + f_APC_TP53_KRAS_NRAS))/2"),
                            stringsAsFactors = FALSE)

## Fitness specifications
COADREAD_fitness_freqdep <- allFitnessEffects(genotFitness = COADREAD_gen_freqdep,
                                                 frequencyDependentFitness = TRUE,
                                                 frequencyType = "rel")

## Evaluate all genotypes considering population sizes of the clones
(COADREAD_FL_freqdep <- evalAllGenotypes(COADREAD_fitness_freqdep,
                                 spPopSizes = c("WT" = 5, "APC" = 5, "APC, TP53" = 5,
                                                "APC, TP53, KRAS" = 10,
                                                "APC, TP53, KRAS, PIK3CA" = 50,
                                                "APC, TP53, KRAS, NRAS" = 50,
                                                "APC, TP53, KRAS, PIK3CA, NRAS" = 50)))
```

```
##                           Genotype  Fitness
## 1                               WT 1.000000
## 2                              APC 1.500000
## 3                             KRAS 0.000000
## 4                             NRAS 0.000000
## 5                        PIK3CA 0.000000
## 6                          TP53 0.000000
## 7                     APC, KRAS 0.000000
## 8                     APC, NRAS 0.000000
## 9                   APC, PIK3CA 0.000000
## 10                    APC, TP53 2.250000
## 11                   KRAS, NRAS 0.000000
## 12                 KRAS, PIK3CA 0.000000
## 13                   KRAS, TP53 0.000000
## 14                 NRAS, PIK3CA 0.000000
## 15                   NRAS, TP53 0.000000
## 16                 PIK3CA, TP53 0.000000
## 17              APC, KRAS, NRAS 0.000000
## 18            APC, KRAS, PIK3CA 0.000000
## 19              APC, KRAS, TP53 3.375000
## 20            APC, NRAS, PIK3CA 0.000000
## 21              APC, NRAS, TP53 0.000000
## 22            APC, PIK3CA, TP53 0.000000
## 23           KRAS, NRAS, PIK3CA 0.000000
## 24             KRAS, NRAS, TP53 0.000000
## 25           KRAS, PIK3CA, TP53 0.000000
## 26           NRAS, PIK3CA, TP53 0.000000
## 27      APC, KRAS, NRAS, PIK3CA 0.000000
## 28        APC, KRAS, NRAS, TP53 5.062500
## 29      APC, KRAS, PIK3CA, TP53 5.062500
## 30      APC, NRAS, PIK3CA, TP53 0.000000
## 31     KRAS, NRAS, PIK3CA, TP53 0.000000
## 32 APC, KRAS, NRAS, PIK3CA, TP53 4.776786
```

```
## Fitness landscape representation
plotFitnessLandscape(COADREAD_FL_freqdep)
```

Next, we try to run the same simulation we did before with these new fitness specifications. After modifying the fixation_tolerance parameter so that we could detect the APC, TP53, KRAS, PIK3CA, NRAS genotype rapidly, it never arises. This can be explained by its slightly lower fitness compare to its ancestors, which by the moment PIK3CA or NRAS mutate to generate the fivefold-mutated genotype have a very high frequency, slowing down APC, TP53, KRAS, PIK3CA, NRAS clone growth up to extinction. Running an additional short simulation, in which we just set onlyCancer as FALSE (so as for the simulation to run until finalTime), in Figure 35 we don't see the presence of the fivefold-mutated clone, yet Figure 36 shows an oscillating pattern of growth for this clone. Also, the phylogeny recorded for each simulation shows the appearance of APC, TP53, KRAS, PIK3CA, NRAS genotype (Figure 37. Either way, its frequency is so low that it cannot trigger the onlyCancer condition to end the first simulation. Although we cannot assure whether the phenomenon of frequency-dependent fitness could be influencing on the lack of coexistence of PIK3CA and NRAS in real colorectal cancer samples (11), these findings support there are probably additional evolutionary events leading genotype frequency apart from mutual exclusivity.

```
## Simulate cancer progression in 10 individuals until APC, TP53, KRAS,
## PIK3CA, NRAS genotype fixates

set.seed(125)
COADREAD_Simul_freqdep <- oncoSimulIndiv(
               COADREAD_fitness_freqdep,
               model = "McFL", ## Model used
               mu = 1e-4, ## Mutation rate
```

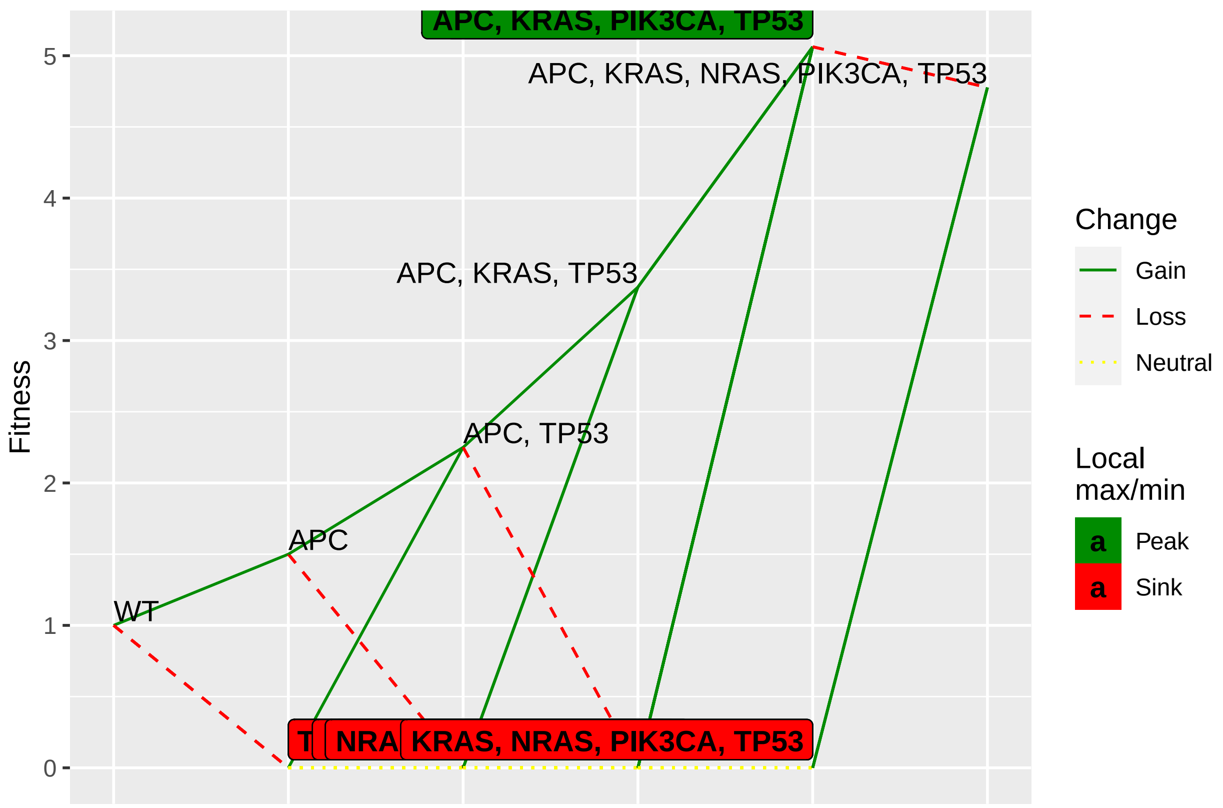

Figure 34: Fitness landscape corresponding with the simplified DAG for the COADREAD dataset accounting for frequency-dependent fitness

```
               sampleEvery = 0.02, ## How often the whole population is sampled
               keepEvery = 1,
               initSize = 200, ## Initial population size
               finalTime = 2000,
               keepPhylog = TRUE, ## Allow to see parent-child relationships
               onlyCancer = TRUE,
               detectionSize = NA,
               fixation = c(c("_, APC, TP53, KRAS, PIK3CA, NRAS"),
                            fixation_tolerance = 0.99),
               detectionDrivers = NA,
               detectionProb = NA)
```

```
##
##  Hitted wall time. Exiting.
##  Hitting wall time is regarded as an error.
```

```
## Simulate cancer progression in 10 individuals for a final time of 300 time units

set.seed(125)
COADREAD_Simul_freqdep <- oncoSimulPop(10,
               COADREAD_fitness_freqdep,
               model = "McFL", ## Model used
               mu = 1e-4, ## Mutation rate
               sampleEvery = 0.02, ## How often the whole population is sampled
               keepEvery = 1,
               initSize = 200, ## Initial population size
               finalTime = 300,
               keepPhylog = TRUE, ## Allow to see parent-child relationships
               onlyCancer = FALSE,
               detectionSize = NA,
               fixation = NA,
               detectionDrivers = NA,
               detectionProb = NA)
```

```
## You are running Windows. Setting mc.cores = 1
```

```
## Plot of simulation for genotypes
plot(COADREAD_Simul_freqdep[[9]],
show = "genotypes",
type = "stacked",
ylim = c(0, 35000))
```

```
plot(COADREAD_Simul_freqdep[[9]],
     show = "genotypes",
     type = "line",
     ylim = c(1, 100000000))
```

```
## Parent-child relationship derived from simulation
plotClonePhylog(COADREAD_Simul_freqdep[[9]],
                N = 0, ## Specify clones that exist
                keepEvents = TRUE ## Arrows showing how many times each clones appeared
)
```

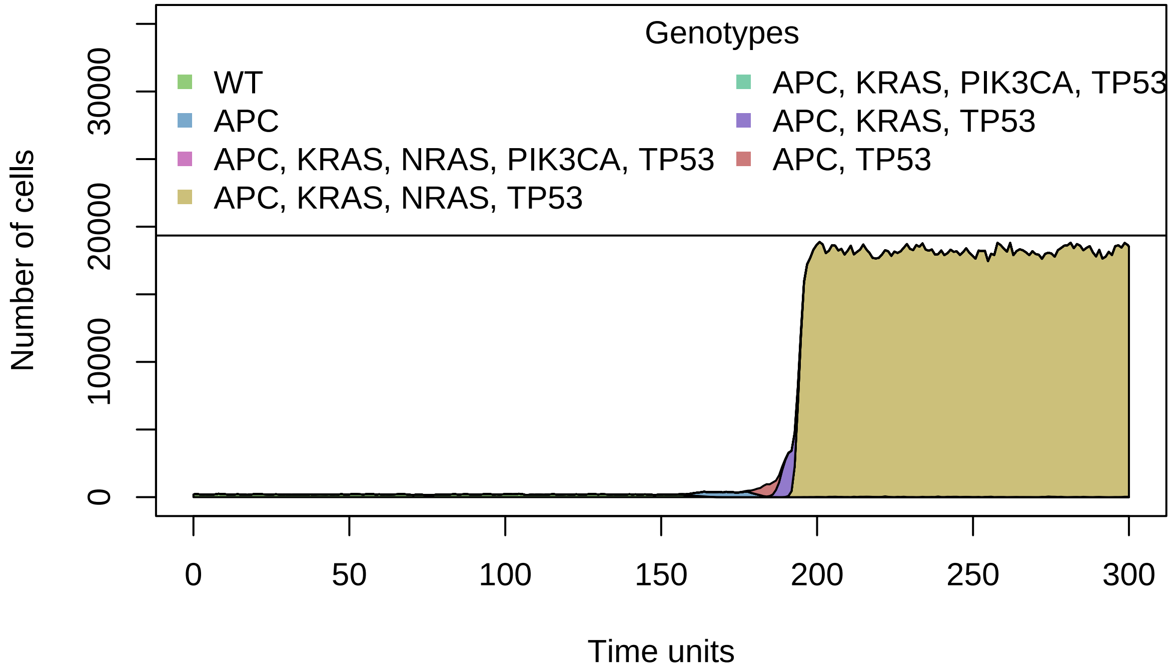

Figure 35: One of the 10 simulations of cancer progression using the frequency-dependent fitness model for the COADREAD dataset (stacked plot)

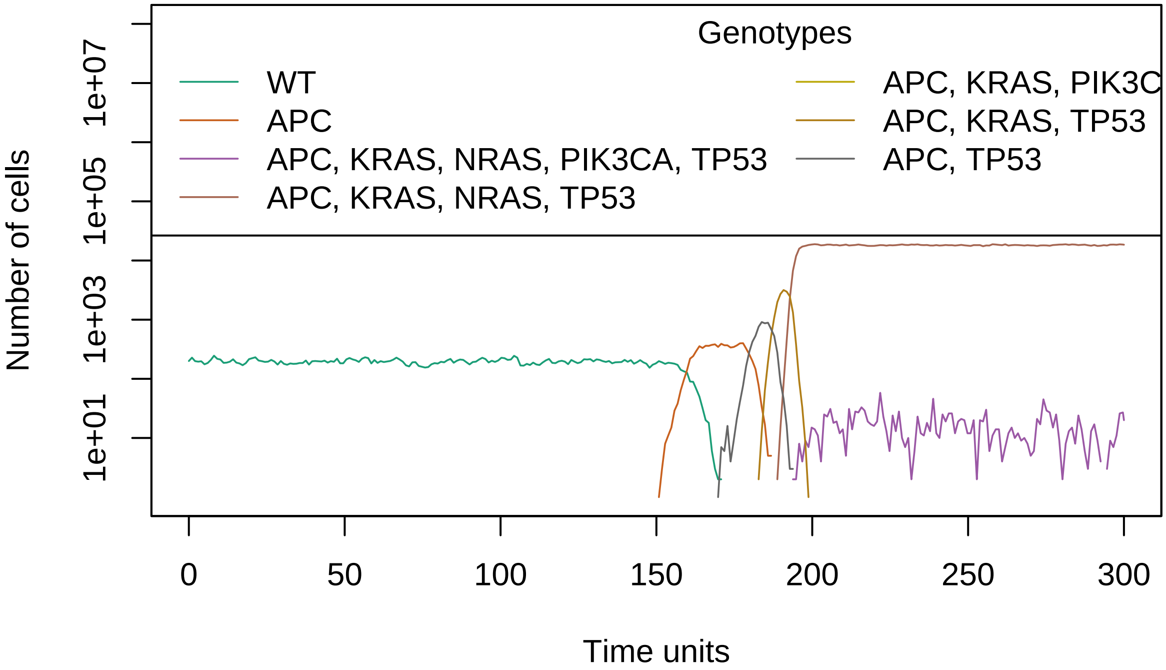

Figure 36: One of the 10 simulations of cancer progression using the frequency-dependent fitness model for the COADREAD dataset (line plot)

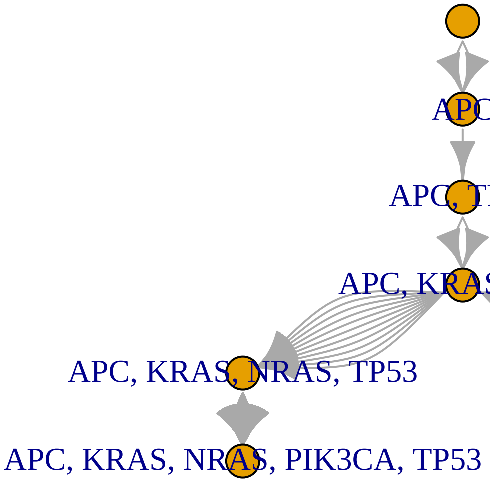

Figure 37: Parent-child relationship derived from one of the 10 simulations

6 Discussion and Conclusions

Cancer progression models (CPMs) are constructed upon genotype frequency data obtained from cross-sectional tumor samples of different individuals. They provide information on mutational restrictions in the form of Directed Acyclic Graphs (DAGs), specifying strict cancer development paths. Accordingly, CPMs do not allow deviations from DAGs of restrictions, meaning any genotype not included in the model cannot exist (13). However, cancer progression occurs in an evolutionary scenario that makes the process much more flexible, as well as conditioned by other phenomena. Thus, it is paramount to approach cancer progression from an evolutionary perspective, accounting for fitness specification of genotypes in each temporary space. Fitness landscapes include all possible paths of tumor progression and enable to study different outcomes depending on the selected evolutionary model (5,9).

In this work, we mapped various CPMs (1,10,11) to different evolutionary models and analyze the diverse fitness landscapes in which their inferred restrictions could fit. They apply a generative approach to derive constrains and mutual exclusivity between genes. Maintaining their mutual exclusivity assumptions, we constructed their corresponding fitness landscapes under different fitness specifications, including deviation from monotonicity, order effects, reciprocal sign epistasia and frequency-dependent fitness. Authors ambiguously define which is the relationship between the mutually exclusive genes, neither there is a null or lethal epistatic effect. We tested how this consideration strongly affects the resulting fitness landscape, with synthetic lethality being the relationship that best fits the mutual exclusivity restrictions state in the CPMs. Conversely, a null effect on fitness does not avoid genotypes violating the mutual exclusivity restriction to reach the highest fitness stage in the models and eventually arise in the simulations.

Order effects consider the path in which mutations are accumulated when mapping the fitness of genotypes. A same genotype can be associated to different fitness values depending if the mutations acquired violate or not restrictions defined in the model. Favoring genotypes consistent with gene's dependencies. In this context, the fitness landscape will show the same genotypes at different heights, depending on the order in which mutations are acquired. Frequency-dependent fitness also shapes the fitness landscape so that mutually exclusive genes are less likely to appear together due to a lower fitness in the presence of other clones. All this evidence supports the idea that cancer progression is better explained through an evolutionary perspective, as evolution significantly modulates the landscape in which clones develop. Current CPMs, based on probabilistic methodologies, cannot recapitulate these phenomena and thus, conclusions derived from them should be questioned and examine under an evolutionary approach.

OncoSimulR (13) is a very complete and convenient clone-based genetic simulator to evaluate cancer progression under evolutionary models. Apart from enabling fitness specification considering order and restriction of mutations, epistasia and clone's frequency for running cancer simulations, it displays all possible paths of tumor progression in fitness landscapes so it is possible to examine differences among evolutionary models and their subsequent cancer simulations. However, fitness landscapes are limited to a small set of genes to clearly identify all possible paths, decreasing the capacity of this software. In addition, it does not display DAG nor fitness landscape when order effects is used to define fitness effects. Also, DAGs of restrictions admit only eight nodes, hampering the mapping of large CPMs, which usually correspond to the most realistic.

7 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

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

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

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

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

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

7. Nowell PC. The clonal evolution of tumor cell populations. Science. 1976; doi: 10.1126/science.959840

8. Lee EYHP, Muller WJ. Oncogenes and tumor suppressor genes. 2010. doi: 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

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

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

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. Archetti M, Pienta KJ. Cooperation among cancer cells: Applying game theory to cancer. Nature Reviews Cancer. 2019;19(2):110–7.

19. 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

20. Ciriello G, Cerami E, Sander C, Schultz N. Mutual exclusivity analysis identifies oncogenic network modules. Genome Research. 2012; doi: 10.1101/gr.125567.111

21. 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

22. 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

23. Yeang C-H, McCormick F, Levine A. Combinatorial patterns of somatic gene mutations in cancer. The FASEB Journal. 2008; doi: 10.1096/fj.08-108985