Simulating pathways, mutual exclusivity, et al.

Contents

1 Mutal exclusivity and modules

1.1 pathTiMEx, a generative probabilistic graphical model of cancer progression

Although cancer progression is a dynamic process, genotype data is usually obtained as **cross-sectional** samples. It is a combination of snapshots taken from different tumor progression at different stages (1). Assuming that these observations reflect the same stochastic process, they can be used to infer restrictions between tumor events. This order of effects can be report as a direct graph. Knowing this constraints between genes, it allows to define a therapeutic target to avoid cancer progression (2).

In (3), the authors 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.

This generative model allows to generate a cancer progression model including mutual exclusivity between groups and progression among pathways. In this approach, authors think in both, genes and modules effects (set of genes). Mutual exclusivity is a common phenomena in cancer progression. It is defined as the situation where two events (mutations) occur less frequently than expected by chance (1). Two genes are mutually exclude if the presence of one of them avoids the presence of the other. In nature, there are two mechanisms that can lead to this phenomena:

- Synthetic lethality, where carrying both mutations is detrimental for the viability of the cell.
- Null effect, where whichever mutation occurs first involves most of the selective advantage and decrease the selective pressure to occur for the others.

The colorectal cancer model depicted in Figure 3.A (3) is used as an example of model to map. The colorectal cancer dataset used to built that model is obtained from (4). The poset restrictions proposed can be coded using the **OncoSimulR package** (5), concretely, the `allFitnessEffects` function. It creates mutations effects given specification of restrictions, epistasis or order effects. In this case, restrictions are used to construct the graph.

Some parameters are mandatory when `allFitnessEffects` function is used. It is the case of the *restriction table*. It specifies the dependencies between genes or modules (genes/modules parents and genes/module children). Parameter `s` and `sh` refers to a numeric vector with the fitness effect that applies if the relationship is or is not satisfied, respectively. Authors don't specify its value since they are not interested in fitness. To justify the values given, we will use the waiting time rate parameter λ defined in the model. Early events in cancer progression will have greater λ values while late events will have a lower one (values for all genes or modules are showed in Table 1). Thus, genes/modules with higher λ will receive a higher fitness value (`s`). On the other hand, `sh` is given a constant value for all possible situations.

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

Although authors don't specify the sort of dependency, in this poset, a semimonotonic dependency is defined between modules B and C with E (`SM`), while a monotonic dependency is defined for the others (`MN`). Model will be represented as an **Diaciclic Direct Graph (DAG)** where arrows conecting genes or modules indicate direct dependencies or constraints between them (2).

```
## 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(rep(0.5, 3), rep(0.05, 3)),

                sh = -0.5,

                typeDep = c(rep("MN", 4), rep("SM", 2)) ## Type of dependency
                 )

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

                  geneToModule = c( ## Specification of the modules
```

```
                                "Root" = "Root",
                                "A" = "APC",
                                "B" = "TP53, EVC2",
                                "C" = "KRAS",
                                "D" = "PI3KCA, EPHA",
                                "E" = "FBXW7, TCF7L2"),

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


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

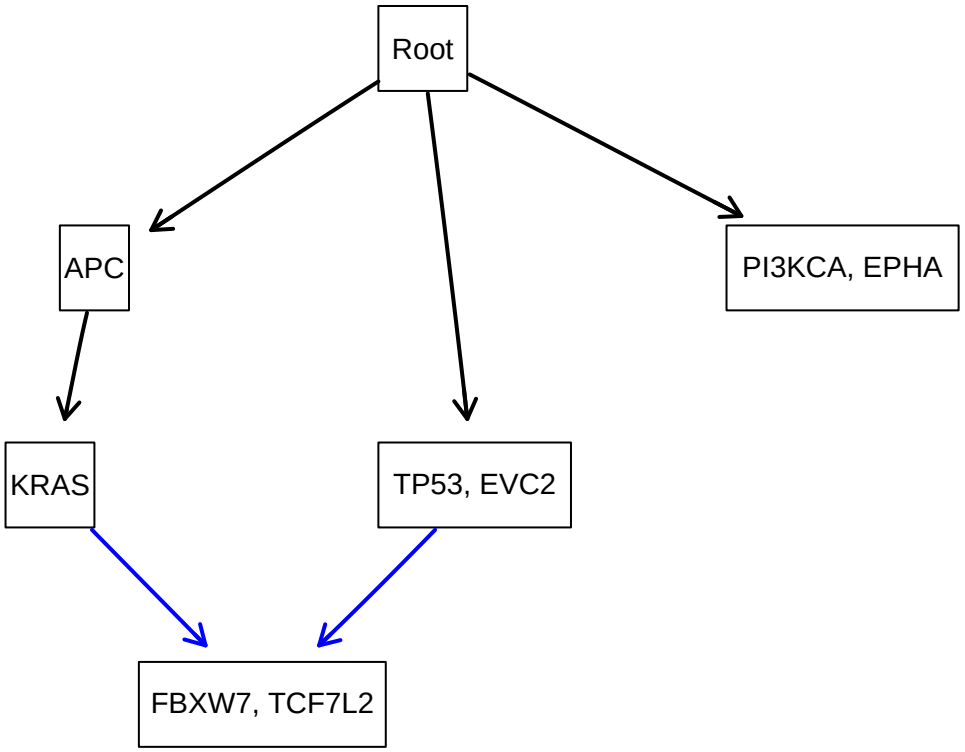

Figure 1: DAG from colorecta cancer

Model proposed by (3) indicates that, from a wild type genotype (depicted as "Root" in Figure 1) it is possible to follow three different paths. The wild type genotype can suffer a mutation in the `APC` gene or in the `TP53`, `EVC2` or `PI3KCA`, EPHA modules. These mutations occur independently, without the need for previous mutations to occur.

In the case of the `PI3KCA`, `EPHA` module, it is not essential for other mutated genes or modules to appear. On the other hand, the mutation in the APC gene is necessary for a mutation to occur in the `KRAS` gene. The

APC gene is the parent node of the KRAS gene (**monotonicity dependency**). Also, the KRAS gene would be necessary for the mutation to appear in the FBXW7, TCFL2 module. However, the relationship of the KRAS gene to the FBXW7, TCF7L2 module has been defined as a **semitonicity dependency**. This implies that there is no need for a mutation in the TP53, EVC2 module in order for it to mutate. Similarly, a mutation in the TP53, EVC2 module would be sufficient for a mutation in the FBXW7, TCF7L2 module to occur (this semitonicity dependency is dipected in blue color in Figure 1).

In a DAG, only the genotypes that fulfill the restrictions defined by the arrows connecting the genes/modules can exist. Moreover, restrictions set in DAG do not contain any information about fitness of each individual genotypes, it is just a pathway to follow to cancer progression. On the other hand, fitness landscapes (or genotype-fitness maps) show the fitness associated to each genotype allowing to know the impact of a specific mutation. Furthermore, the restrictions reflected in the DAG just show sign epistasis between genes/modules. Epistasis is an effect where the phenotypic consequences of a certain mutation depend on the genetic background in wich it take place.

A specific type of epistasis is called *sign epistasis* and it refers to the case where a mutation yields to a increase (beneficial) or decrease (deleterious) of the fitness depending on the genotypic background of the cell. It has a different sign depending on the other genes mutated in the clone cell. Although this kind of epistasis can be depicted using a DAG, it can not show reciprocal sign epistasis, a specific situation where two individual mutations increase the fitness, although combine reduce it (synthetic lethality).

Hence, to visualize the relationships between genotypes and effects in fitness, the fitness landscape using the restrictions specified in Figure 1 is generated. For that aim, the `evalAllGenotypes` function is used. It returns a table with all the genotypes from the fitnessEffects description indicated as well as the genotype associated to it. The table obtained can be used as an object to plot a fitness landscape of the Figure 1.

```
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)
```

Fitness landscape obtained is displayed in Figure 2. It shows a quite busy fitnees landscape due to the huge amount of possible genotypes combinations, each one with a different fitness value. However, it reflects genotype acquisition in terms of survival and adaptation for the cell, allowing to set what is the fitness associated to a clone that gets a certain mutations or a certain number of them.

2 Simplified cancer progression model

In order to properly visualize a fitness landscape, a simplified version of the model coded in subsection 1.1 is constructed. This model doesn't use modules, just individual genes. This approach will lead to clear fitness landscape and to properly identify processes that may occur.

Authors (3) claim that there is a phenomena of mutual exclusivity between certain genes of specific pathways. Mutual exclusivity means that the presence of one gene in a specific pathway will be enough to fitness contribution, since mutation in the other genes of the same pathway are negative selected and therefore, the presence of the other gene in the same module can be discarded, since they will not mutate.

Therefore, the same model as in the previous case will be coded, but without specifying modules. Each module will be consider as an specific gene.

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

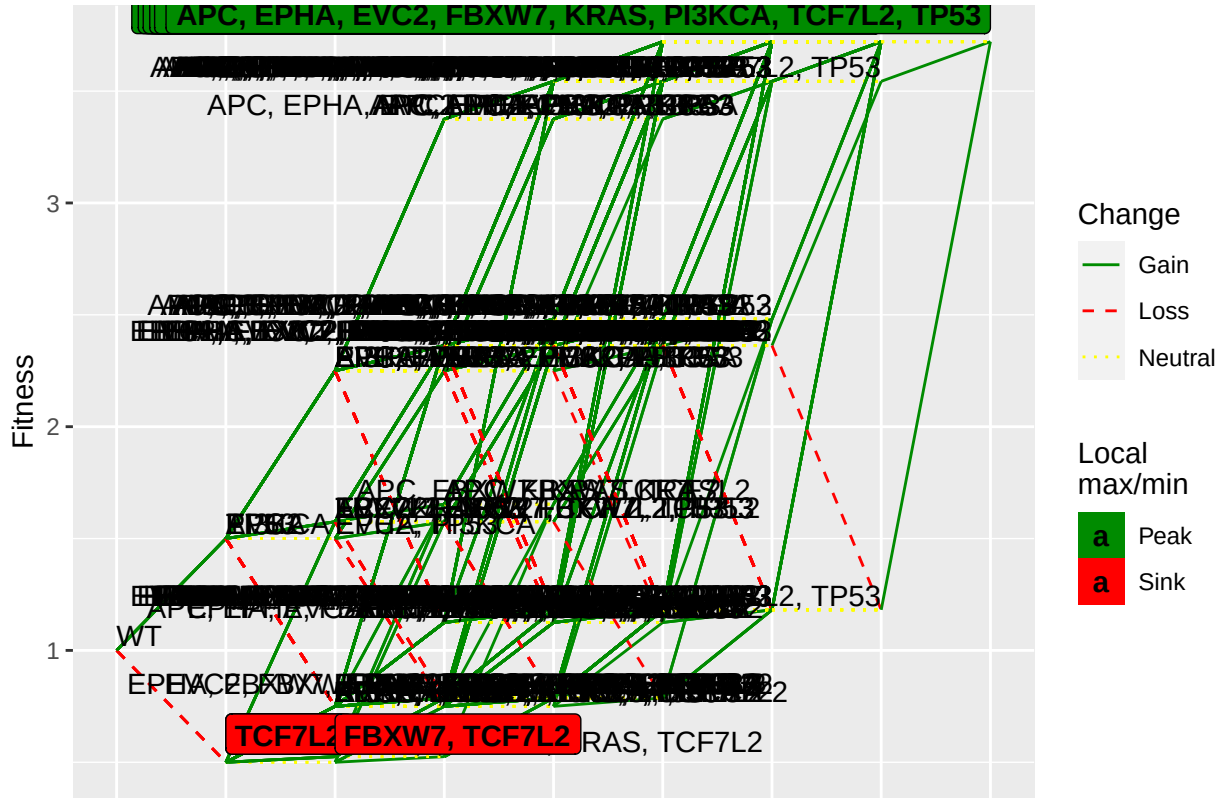

Figure 2: Fitness landscape from colorectal cancer

```
plot(Scolcancer)
```

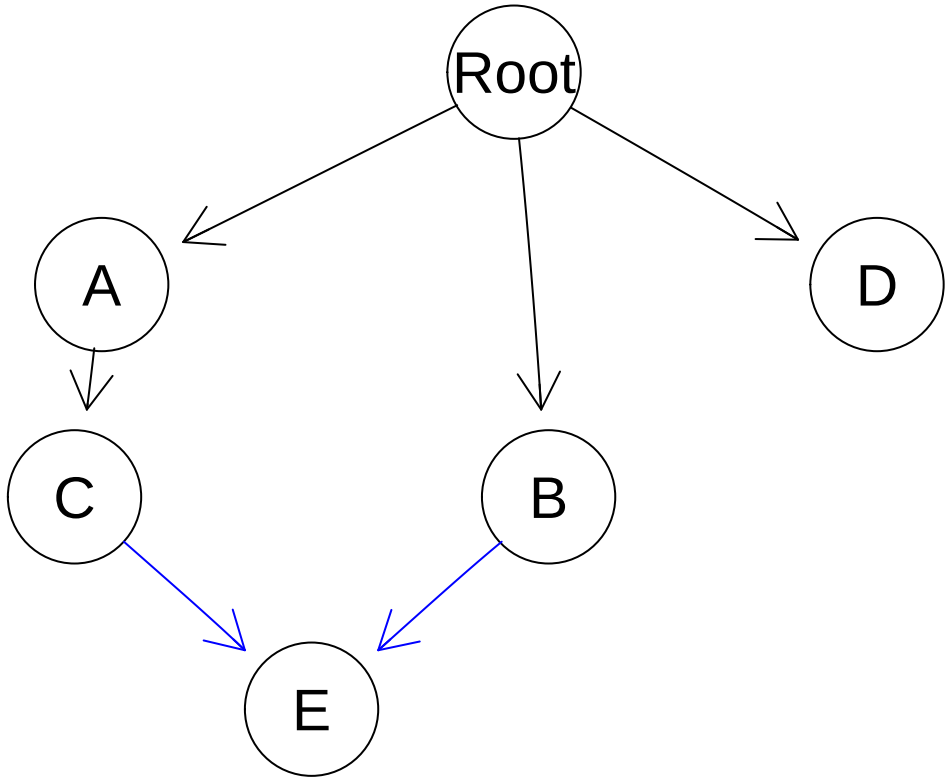

Figure 3: DAG from a simplified model of colorecta cancer

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

```
##         Genotype  Fitness
## 1              A 1.500000
## 2              B 1.500000
## 3              C 0.500000
## 4              D 1.500000
## 5              E 0.500000
## 6           A, B 2.250000
## 7           A, C 1.575000
## 8           A, D 2.250000
## 9           A, E 0.750000
## 10          B, C 0.750000
## 11          B, D 2.250000
## 12          B, E 1.575000
## 13          C, D 0.750000
## 14          C, E 0.525000
## 15          D, E 0.750000
## 16       A, B, C 2.362500
```

```
## 17       A, B, D 3.375000
## 18       A, B, E 2.362500
## 19       A, C, D 2.362500
## 20       A, C, E 1.653750
## 21       A, D, E 1.125000
## 22       B, C, D 1.125000
## 23       B, C, E 0.787500
## 24       B, D, E 2.362500
## 25       C, D, E 0.787500
## 26    A, B, C, D 3.543750
## 27    A, B, C, E 2.480625
## 28    A, B, D, E 3.543750
## 29    A, C, D, E 2.480625
## 30    B, C, D, E 1.181250
## 31 A, B, C, D, E 3.720938
```

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

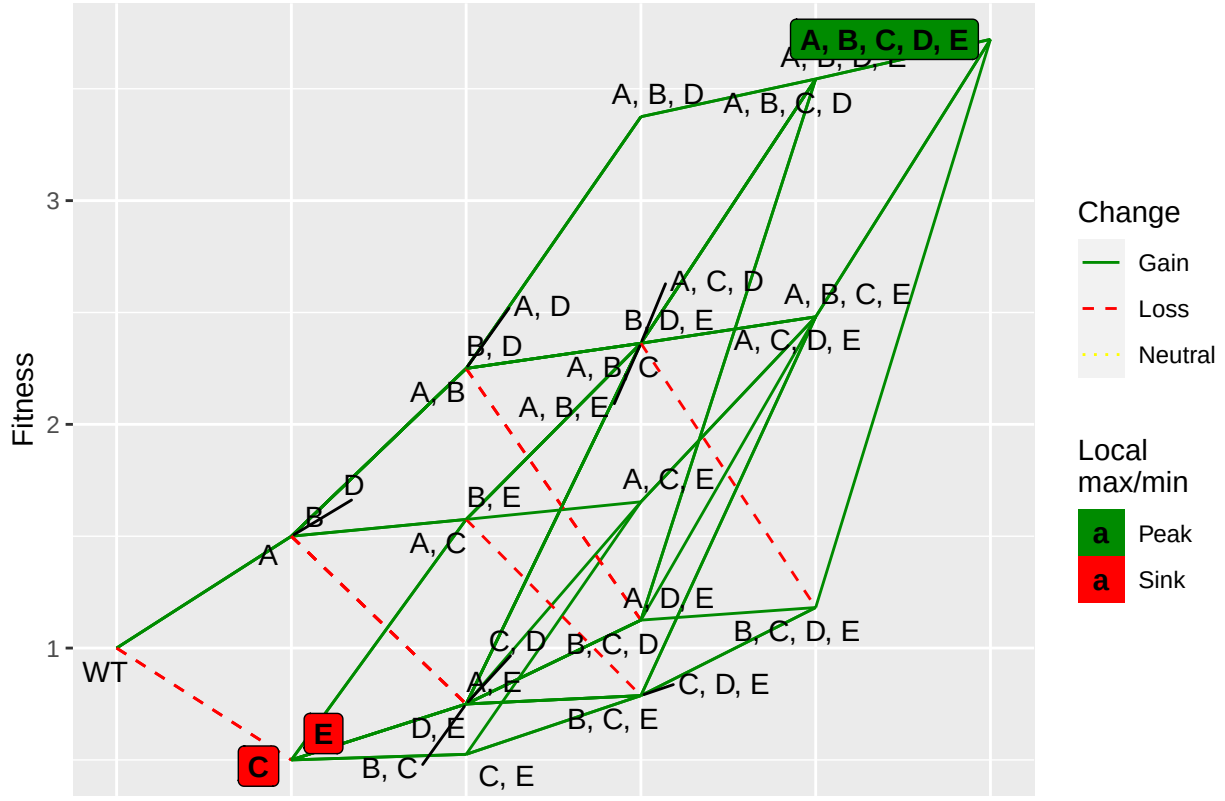

Figure 4: Fitness landscape from a simplified model of colorectal cancer

DAG graph and fitness landscape of this simplified model is depicted in Figure 3 and Figure 4, respectively. DAG showed in Figure 3 is the same as the DAG depicted in Figure 1 but without expanding modules. In this case, there is not an improvement in legibility or clarity. However, if we compare the fitness landscape

obtained with the simplification (see Figure 4) with the previous fitness landscape (see Figure 2), there are a clear difference in clarity. In this new fitness landscape is possible to visualize the fitness given to each genotype and therefore, give an evolutionary sight to the model.

3 Simulating data from a simplified model

Restrictions set in 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. `OncoSimulIndiv` function is used to simulate colorectal tumor progression. This function simulates a single evolutionary path. It is necessary to include the poset with the order restrictions defined for the simplified model (see section 2). 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 (6). Initial population size is set at 400. Only one mutation rate is used: 1e-4. Final time is set to 220 to visualize clones' evolution. Furthermore, keepPhylog parameter is set true to plot the parent-child relationships occurring in the simulation as well as its frequency (`plotClonePhylog` function).

```
set.seed(257) ## Fix the seed for reproducibility

Simul <- oncoSimulIndiv(Scolcancer, ## 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 = 400, ## Initial population size
                        finalTime = 220,
                        keepPhylog = TRUE, ## Allow to see parent-child relationships
                        onlyCancer = FALSE
)


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

)
```

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

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


)
```

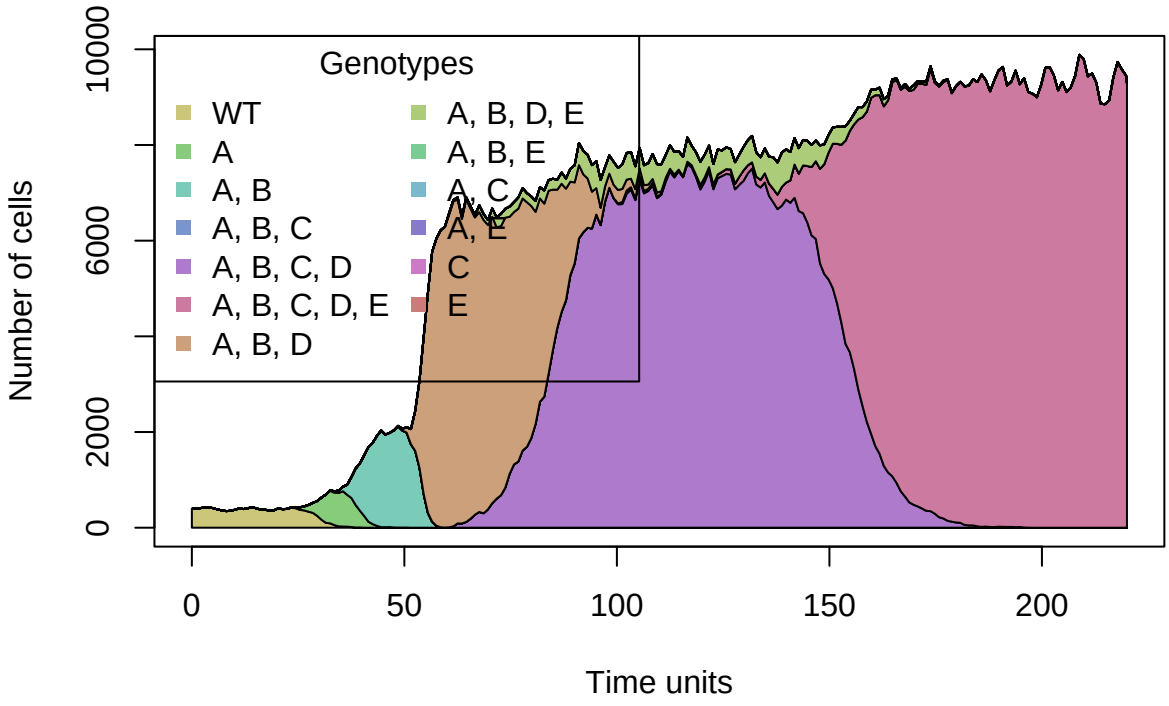

Figure 5: Simulation of cancer progression using the fitness landscape of the simplied model (stacked plot)

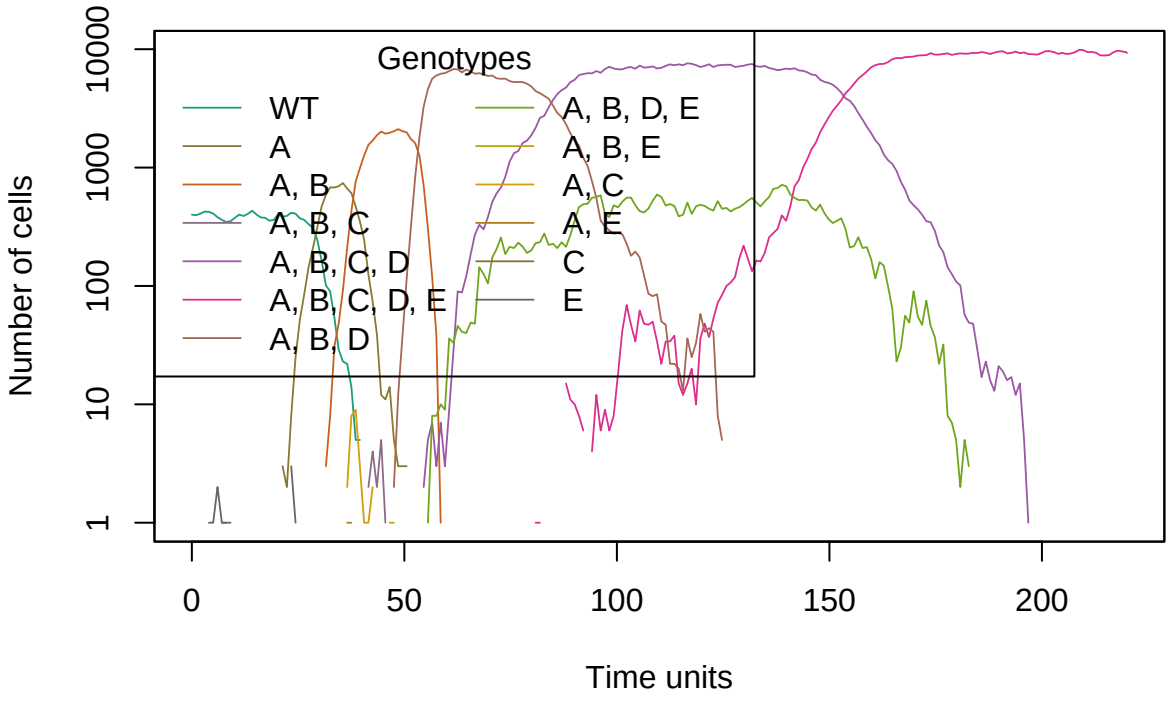

Figure 6: Simulation of cancer progression using the fitness landscape of the simplied model (line plot)

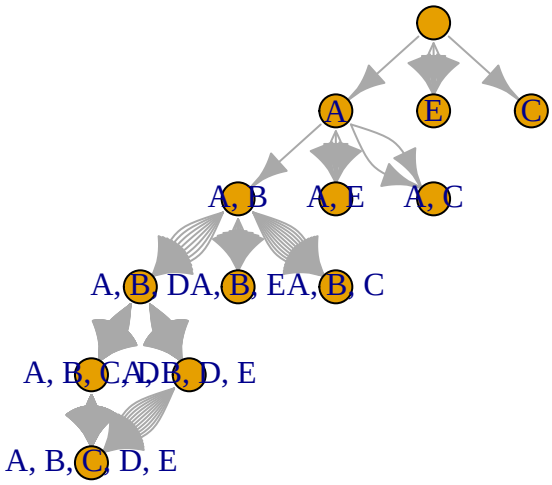

Figure 7: Parent-child relationship derived from simulation

A stacked and line plot of the simulation is depicted in Figure 5 and Figure 6, respectively. Both plots show the genotype acquisition by time and the number of clones carrying that genotype in the cell culture. Different cell population converge in different time moments, each carrying a diferent genotype and therefore, a different fitness.

In Figure 5 wild type genotype ("WT") progressively despairs while clones carrying a mutation in gene "A" arrive to culture. However, they are substituted by a new clone that also carries a mutation in "B". Then, this clone suffers different mutations resulting in the coexistence of different genotypes in the cell culture, each one with a different fitness. Finally, genotype carrying all genotypes is stabilized in the culture. As it fitness is the greatest among all the fitness of all genotypes. It is obvious that it is more selective favored and it will overcome the concurrence with the other genotypes.

On the other hand, Figure 6 shows the same information but it is possible to observe all genotypes generated in the simulation, even those that survive for little time. In addition to the genotypes seen in Figure 5, some other genotypes appear in the cell culture, but due to selective pressure, they are not able to survive.

Hence, genotypes observed in the simulation not only follow the restrictions set in DAG of subsection 1.1. It is just a generative model where dependencies between genes are defined, but they may not occur in real life in that order. Moreover, this model is constructed using cross-sectional data, tumor snapshots in a specific time of cancer progression, it is not a temporal dataset from the onset to the end of the cancer progression.

Figure 7 shows the genotype evolution in the simulation. Arrows' width represent frequency of clone creation. Widther arrows indicate a higher frequency of change from the parent genotype to the child genotype.

4 Order effects

To explore order effects in cancer progression, a simple model derived from the restriction model inferred by (3) is created.

This simplified model just contains 3 genes: `APC`, `TP53` and `KRAS`, genes considered as `superdrivers` (7), meaning that are the main responsible for cancer progression since they provide a higher fitness gain than the other genes in the model. This conclusion is obtained from its article where they used the same colorectal cancer dataset as (3). Thus, it can be extrapolated to our case.

The relationships between those genes was previously depicted in subsection 1.1. In this case, we will set `APC` as the parent of `KRAS`. Both, `APC` and `TP53` have as parent `Root`. Based on the waiting time rate parameter λ, the fitness values of each possible order is given (see Table 1).

λ is higher for `APC`, which means that it seems to appear before in the cancer progression. λ for `KRAS` is the lower between the three, meaning that it mutates the last. `TP53` mutation occurs between `APC` and `KRAS`. Order effects are defined following this criteria: clones suffering mutations in the previous order are favored with a higher fitness. Other possible paths of cancer progression are slightly less naturally selected (assumption based on (3)). Order effect is visualize using `evalAllGenotypes` function.

```
cc <- data.frame(parent = c(rep("Root", 2), "A"),
                 child = c("A", "C", "B"),
                 typeDep = "MN")

cc_order <- allFitnessEffects(
  orderEffects = c("A > B > C" = 0.5, "B > A > C" = 0.2,
                   "B > C > A" = 0.1,
                   "B > C" = 0.2,
                   "C > B" = 0.1,
                   "B > A" = 0.1,
                   "A > B" = 0.3),
```

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

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

```
##              Genotype Fitness
## 1                 APC   1.000
## 2                KRAS   1.000
## 3                TP53   1.000
## 4          APC > KRAS   1.300
## 5          APC > TP53   1.000
## 6          KRAS > APC   1.100
## 7         KRAS > TP53   1.200
## 8          TP53 > APC   1.000
## 9         TP53 > KRAS   1.100
## 10 APC > KRAS > TP53   2.340
## 11 APC > TP53 > KRAS   1.430
## 12 KRAS > APC > TP53   1.584
## 13 KRAS > TP53 > APC   1.452
## 14 TP53 > APC > KRAS   1.430
## 15 TP53 > KRAS > APC   1.210
```

We obtain a table with the different possible genotypes as well as the order of appearance. However, this approach doesn't allow to generate neither a DAG neither a fitness landscape. Thus, is not possible to visualize the evolution of the genotypes with time.

```
#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
```

Assuming a model where there is not an order effect, a mutation in gene "B" followed by a mutation in gene "A" will reach the same fitness as if the mutation in gene "A" occurs first. However, in the model just generated, the order of the mutation impacts the final fitness reached by the tumoral clone. Since, the previous alteration of some genes before can lead to an evolutionary advantage.

In a non order effect model, the final fitness value is the same for all the clones, while this is unlikely to happen in real life. Clones carrying a certain mutation from the beginning would survive easily than those reaching the same genotype in a different order.

This is one limitation of `OncoSimulR package`, it doesn't allow to visualize those scenarios (yet).

5 Epistasis to simulate order effects

Epistasis assume that there is a dependence between genotypes. The effect of a mutation depends on the genetic background in which it happens (8). Now, we will cope with dependencies between genes using epistasis.

For that, we will use the same model described in section 4. As explained before, it is supposed to be a certain cancer progression restriction and therefore, the fitness values given to each different genotype is based in that criteria.

```
## Fitness object defined using epistasias
cc_epi <- allFitnessEffects(epistasis =
                              c("A: -B: -C" = 0.4,
                                "-A: B: -C" = -0.4,
                                "-A: -B: C" = 0.3,
                                "A: B: -C" = 0.8,
                                "A : B: C" = 1.4,
                                "-A: B: C" = 0.1,
                                "A : -B: C" = 0.5
                                ),
                            geneToModule =
                              c("A" = "APC",
                                "B" = "KRAS",
                                "C" = "TP53")
                              )

## DAG (epistasis)
plot(cc_epi, expandModules = TRUE, autofit = TRUE)
```

```
## Genotypes derived from fitness defined with epistasia relationships
(cc_epi_geno <- evalAllGenotypes(cc_epi ))
```

```
##          Genotype Fitness
## 1             APC     1.4
## 2            KRAS     0.6
## 3            TP53     1.3
## 4       APC, KRAS     1.8
## 5       APC, TP53     1.5
## 6      KRAS, TP53     1.1
## 7 APC, KRAS, TP53     2.4
```

```
## Fitness landscape from this relationships
plotFitnessLandscape(cc_epi_geno, use_ggrepel = TRUE)
```

Using this approach, it is possible to visualize the DAG (see Figure 8). In this case, there are discontinues yellow lines connecting each gene. This lines indicate a dependence between them. Fitness landscape is also plotted (see Figure 9).

With this model, we promote the clones of tumoral cells beginning with a first mutation in `APC`. Conversely, other clones not starting with that mutation (`KRAS` or `TP53`) have a lower fitness value. On the other hand, all genotypes end with the same fitness, but it is selective favored clones following the order defined in section 4.

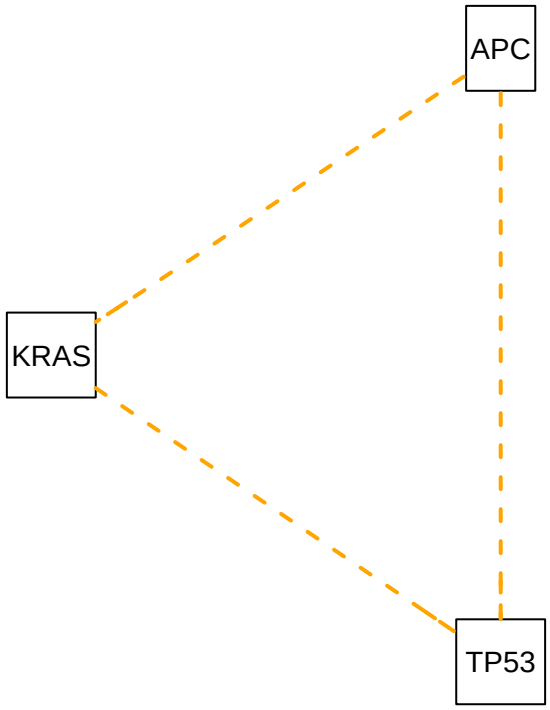

Figure 8: DAG showing epistasis between genes

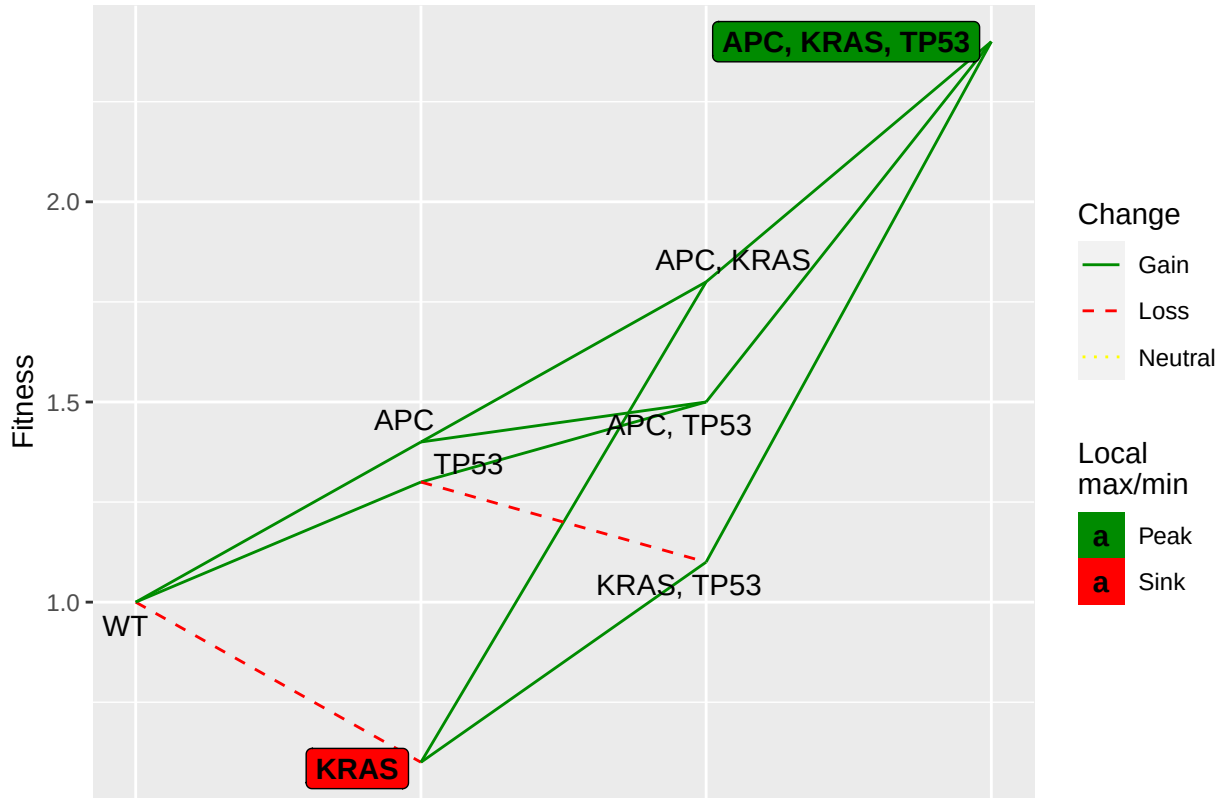

Figure 9: Fitness landscape of model defined by epistasis

6 References

1. 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

2. 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

3. 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

4. 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

5. 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

6. 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

7. 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

8. Poelwijk FJ, Kiviet DJ, Weinreich DM, Tans SJ. Empirical fitness landscapes reveal accessible evolutionary paths. Nature. 2007;445(7126):383–6. doi: 10.1038/nature05451