Using OncoSimulR to get accessible genotypes and transition matrices

Ramon Diaz-Uriarte[1,2,†]

[1]Dept. of Biochemistry, School of Medicine, Universidad Autónoma de Madrid, Madrid, Spain
[2]Instituto de Investigaciones Biomédicas 'Alberto Sols' (UAM-CSIC), Madrid, Spain
[†]To whom correspondence should be addressed: r.diaz@uam.es
`https://ligarto.org/rdiaz`

2023-07-06
Version d6a84b0

Contents

1 Introduction

Here I explain how we can use OncoSimulR[1] to get accessible genotypes and transition matrices for CBN (and MCCBN), OT, HESBCN, and OncoBN. The code for using OncoSimulR is implemented in access_genots_from_oncosimul.R.

[1] A BioConductor package for forward population genetic simulation in asexual populations; it allows us to specify fitness, among other ways, using DAGs of restrictions. Repo at `https://github.com/rdiaz02/OncoSimul`. Citation: Diaz-Uriarte, R. (2017). OncoSimulR: Genetic simulation with arbitrary epistasis and mutator genes in asexual populations. Bioinformatics, 33(12), 1898–1899. `https://doi.org/10.1093/bioinformatics/btx077`.

(This document is written, on purpose, using an itemized list style, with plenty of repetition and detailed examples, to make it suitable for instance for class use.)

2 Using OncoSimulR to get accessible genotypes and transition matrices

OncoSimulR has had, for a long time, the AND, XOR, OR operations (see the help of "allFitnessEffects", under "typeDep"), if a gene depends on other genes with the same relationship for all parents. Since we can obtain the fitness of genotypes, obtaining accessible genotypes is simple:

- Use an appropriate setting for the "s"
- Use $-\infty$ for sh (so if restrictions are not satisfied, a genotype has fitness 0).
- Evaluate the fitness of genotypes.
- Call function "genots_2_fgraph_and_trans_mat".
  - This is a general function, not linked to any specific cancer progression model. In other words, given a fitness landscape (a mapping from genotypes to fitness) find the accessible genotypes and the transition matrices (not transition rate matrices) between genotypes.
  - For example, this procedure does not assume that mutations that do not kill a genotype always increase fitness or at least do not decrease it. A mutation might increase fitness in some contexts (with some other mutations) and decrease in other contexts, such as with sign and reciprocal sign epistasis.
  - This procedure **assumes SSWM (strong selection, weak mutation)**. Moreover, **we assume evolution can only move uphill in fitness**. For example, a genotype is considered not accessible if its fitness is less than, or equal to (note the “or equal to”) that of its immediate ancestor, and we cannot cross fitness valleys[2].
  - This function returns accessible genotypes, fitness graph, and transition matrices directly from the fitness of the genotypes.

2.1 Computing fitness of genotypes: for CBN (and MCCBN) and OT

- OncoSimulR, when using DAGs, uses a model of fitness (birth rate), for a genotype with restrictions satisfied as $\Pi(1 + s_i)$.
  - Again, to emphasize the above: s_i, when using OncoSimulR with a DAG, is the selection coefficient from gene i **with its restrictions satisfied.**
- Recall that for CBN the transition probabilities can be computed from competing exponentials. For example, suppose from genotype A we can go to enotypes AB and AC. The probability of going to AB should be $\lambda_B/(\lambda_B + \lambda_C)$.
- As in p. 7 of the supplementary material of Weinreich et al., 2006, (Weinreich, D. M., Delaney, N. F., DePristo, M. A., & Hartl, D. L. (2006). Darwinian Evolution Can Follow Only Very Few Mutational Paths to Fitter Proteins. Science, 312(5770),

[2] This excludes, for example, the scenarios studied in Weinreich, D. M., & Chao, L. (2005). Rapid evolutionary escape by large populations from local fitness peaks is likely in nature. Evolution; international journal of organic evolution, 59(6), 1175–1182. http://dx.doi.org/10.1111/j.0014-3820.2005.tb01769.x .

111–114.https://dx.doi.org/10.1126/science.1123539), let us define the selective coefficient of a mutation i as the relative fitness difference that it causes along the mutational pathway.

  - $W_{AB} = W_A \ (1 + s_B)$ or $s_B = \frac{W_{AB} - W_A}{W_A}$.
  - Using our previous example, $Pr(A \rightarrow AB) = \frac{W_{AB} - W_A}{(W_{AB} - W_A) + (W_{AC} - W_A)}$, where W_x is fitness of genotype x.
  - Thus, we get from the above $Pr(A \rightarrow AB) = \frac{s_B}{s_B + s_C}$.
  - (We wrote $W_{AB} = W_A \ (1 + s_B)$. This we can do as we explained what the meaning of the s_i are: selection coefficient from gene i with its restrictions satisfied. See below: Transition probabilities using an epistatic specification.)

- Note that this is the same procedure as in Weinreich et al., 2006, (Weinreich, D. M., Delaney, N. F., DePristo, M. A., & Hartl, D. L. (2006). Darwinian Evolution Can Follow Only Very Few Mutational Paths to Fitter Proteins. Science, 312(5770), 111–114.https://dx.doi.org/10.1126/science.1123539) supplementary material, p. 4): $s_{i \rightarrow j}$ "the selection coefficient for the mutation that carries allele i to allele j"[3].

- Specifically, see equation S5b in the supplementary material of Weinreich et al., 2006, which shows the relationship between the expected value of the conditioned probability of fixation in a mutation from i to j and the expected value of the ratio of the selection coefficient for the mutation that turns i to j over the sum of selection coefficients of beneficial mutations that turn i into all other alleles; see also their figure S1 in p. 7 of the supplementary material that shows the accuracy of their expression.

- Note that this is similar to what is done in Hosseini et al., 2019 (Hosseini, S., Diaz-Uriarte, Ramon, Markowetz, F., & Beerenwinkel, N. (2019). Estimating the predictability of cancer evolution. Bioinformatics, 35(14), 389–397. https://dx.doi.org/10.1093/bioinformatics/btz332), p. i392. The difference is that in Hosseini et al. the s_i is defined as the fitness difference, not the relative fitness difference (and in Hosseini et al there is a normalizing constant, as given by eq. 8).

- Additional note: In Gerstung et al., 2011 (Gerstung, M., Eriksson, N., Lin, J., Vogelstein, B., & Beerenwinkel, N. (2011). The Temporal Order of Genetic and Pathway Alterations in Tumorigenesis. PLoS ONE, 6(11), 27136. https://dx.doi.org/10.1371/journal.pone.0027136), PLoS ONE (p.8) the relationship between λ_i and s_i is also discussed, with additional references given.

- So when using OncoSimulR we do as follows:

  - Set $s_i = \lambda_i$ (for OT, we use edgeWeight instead of λ).
  - Obtain the fitness of all genotypes from OncoSimulR.
  - If so desired (e.g., to ensure the maximum fitness is a specific number), scale all fitnesses by the appropriate factor (that also ensures that WT is kept at one; see, for instance, function `scale_fitness_2` in file `access_genots_from_oncosimul.R`.

- Is the above correct for OT? Strictly not as OT are untimed oncogenetic trees. (And, yes, we are aware that under OT if you have, say, both A and B descend from root, the probability of genotype A is $p_a(1 - p_b)$).

[3] Selection coefficient has the usual textbook definition. For example, Gillespie, 2004 (Population genetics: a concise guide, 2nd. Baltimore, Md: The Johns Hopkins University Press.), p. 63. But here we write $W_{AB} = W_A \ (1 + s_B)$, and thus if $s_B > 0$ AB is fitter than A; see also p.7 of the supplementary material of Weinreich et al., 2006

- It is important to emphasize that we are not claiming λ_i should be taken as equal to s_i. We are using this procedure to obtain accessible genotypes and transition probabilities between genotypes, but not transition rate matrices. For example, as mentioned above, multiplying all s_i by the same constant leaves these transition probabilities unchanged[4]. But even if multiples of the λ_i result in the same transition probabilities, the transition rate matrices are different (the evolutionary process is faster or slower).

2.2 Crucial assumption above

- We compute fitness above assuming that only one of two things can happen: a mutation provides a fitness benefit or it leads to death. When the requirements are satisfied, a mutation conveys a fitness increase (λ_i); otherwise, the cell with the mutation has fitness 0.

- Strictly, mutations without dependencies satisfied might not be lethal, but they should not confer any fitness advantage, so that we will not observe them become fixated in the population (Gerstung et al., 2009, p. 2810: "(. . .) mutations that need to be present before mutation i can fixate.". Gerstung and Beerenwinkel, 2010, Waiting time models of cancer progression. Mathematical Population Studies, 17, 115–135; p. 126: "with steps including both mutation and clonal expansion occurring at effective rates λ_j". Beerenwinkel, N., & Sullivant, S. (2009). Markov models for accumulating mutations. Biometrika, 96(3), 645, p. 659: “In an evolutionary process, this waiting time includes the generation of the mutation plus the time it takes for the allele to reach fixation in the population” and p. 660 “The parameters λ correspond to the rate of evolution, i.e. the product of population size, mutation rate and fixation probability”).

- In OncoSimulR, in addition to the s_i, it is possible to set `sh = 0`, meaning there is no penalty for not respecting the restrictions. When `sh = 0` there is also no fitness gain, either, so fitness for those genotypes ends up being the fitness of the immediate parent (there is no contribution from the gen without restrictions satisfied to the fitness of the parent genotype). Regardless, when `sh = 0`, the transition matrix does not change compared to the transition matrix we obtain if we assume that mutations to genotypes with non-satisfied dependencies lead to a fitness of 0: we said above that a genotype is considered not accessible if its fitness is less than, or equal to (note the “or equal to”) that of its immediate ancestor.

- To elaborate on this point: The output from the code, with `sh = 0`, will result in more genotypes being shown as accessible. It is arguable, though, that those genotypes are not really accessible, since their fitness is never larger than the fitness of their ancestor. So the probability of transitioning to them will be 0 under the expressions above when in SSWM. We have changed the code so that now something is only shown as accessible if its fitness is strictly larger than the fitness of its ancestor.

- (Actually, in OncoSimulR, the s_h can vary by gene, so we could have different s_{hi}, but this does not affect these arguments).

2.3 Fitness specification with OncoSimulR: DAGs vs. epistatic fitness specifications

- We said above: "Again, to emphasize the above: s_i, when using OncoSimulR with a DAG, is the selection coefficient from gene i **with its restrictions satisfied.**"

[4] We can scale all fitness with a function like $W^* = 1 + (W - 1)\ \alpha$.
$W_A = 1 + s_A$. $W^*_A = 1 + s^*_A$. Thus $s^*_A = (W_A - 1)\ \alpha = s_A\ \alpha$.

- This also means, when using DAGs in OncoSimulR, that terms such as s_{ij} are not used in that specification: they are not needed as the DAG models do not include epistasis beyond that given by the DAG, and all these epistatic interactions we capture with the DAG and the s_i and s_h, which denote the fitness effects when restrictions are satisfied and not satisfied, respectively.

- But with OncoSimulR you can also specify fitness with the usual multiplicative expression where you specify explicitly the contribution of genes and gene interactions (e.g., s_{ij} for the effect of the interaction between genes i and j, so that fitness of the genotype with both i and j mutated would be $(1 + s_i)\ (1 + s_j)\ (1 + s_{ij})$).

- In other words, suppose j depends on i. The usual epistatic interaction fitness specification would write: $W_{ij} = (1 + s_i)\ (1 + s_j)\ (1 + s_{ij})$ and $W_j = (1 + s_j)$.

- Using the DAG, if the restriction is not satisfied, i.e., for genotype with only j: $W_j = (1 + s_h)$. If the restriction is satisfied, $W_{ij} = (1 + s_i)(1 + s_j)$. So the meaning of the s is different.

- To fully elaborate here, and to give a more complex example, suppose C depends on both A and B, according to the DAG.
  - When using the DAG, then, these are the expressions for some genotypes:
    - $W_{ABC} = (1 + s_A)(1 + s_B)(1 + s_C)$
    - $W_{AC} = (1 + s_A)(1 + s_h)$
    - (If we had gene-specific s_h, such as s_{hC}, that does not change anything fundamental, just adds a subscript)
  - If we were to use an epistatic specification:
    - $W_{ABC} = (1 + s_A)(1 + s_B)(1 + s_C)(1 + s_{AB})(1 + s_{AC})(1 + s_{BC})(1 + s_{ABC})$
    - $W_{AC} = (1 + s_A)(1 + s_C)(1 + s_{AC})$

- Therefore, the meaning of the s_i is not the same under both specifications. That is why we said "s_i, when using OncoSimulR with a DAG, is the selection coefficient from gene i **with its restrictions satisfied.**" and "terms such as s_{ij} are not used in that specification: they are not needed as the DAG models do not include epistasis beyond that given by the DAG, and all these epistatic interactions we capture $(\ldots)$".

- Yes, sure, we could always re-write the s_i and s_{hi} in the DAG specification as a function of the s_i, s_{ij}, s_{ijk} in the epistatic specification. (See section Transition probabilities using an epistatic specification).

- This was just for the sake of completeness. The use of s_h and the epistatic fitness specification is fully explained in the documentation of OncoSimulR and its vignette, and is not in the scope of this document.

2.4 Transition probabilities using an epistatic specification

- Suppose B and C both depend on A. If we were to use an specification with epistasis, instead of how we have used and interpreted the s_i using the DAGs, then we would have to write $W_{AB} = W_A\ (1 + s^*_B)\ (1 + s^*_{AB})$, where now I am using $s*$ to make the sets of s clearly distinct. We can express the s_B as a function of s^*_B and s^*_{AB}. If we set $s^*_B = 0$ (similar to setting `sh = 0`) then $s_B = s^*_{AB}$. Otherwise, the expression will be $s_B = ((1 + s^*_B)\ (1 + s^*_{AB})) - 1$; and, to respect the restrictions, it must be the case that $s^*_B < 0$.

- The expressions for probabilities of transition become messier, but you end up with a ratio of

  $$\frac{increase\_in\_fitness\_from\_acquiring\_B}{increase\_in\_fitness\_from\_acquiring\_B + increase\_in\_fitness\_from\_acquiring\_C}$$

  where $increase\_in\_fitness\_from\_acquiring\_B$ would include the effect of B, s^*_B, and the epistatic interaction, s^*_{AB}.

- s_B is still the relative fitness difference $\frac{W_{AB}-W_A}{W_A}$. Which is the same as saying that $((1+s^*_B)\ (1+s^*_{AB}))-1=\frac{W_{AB}-W_A}{W_A}$ is the relative fitness difference.

- This shows we can directly use the DAG fitness specification where we take the s_i as the selection coefficient from gene i with its restrictions satisfied.

- And why do we do what we do with CBN? Because it simplifies everything and fitness can be written as $\prod(1+s_i)$ for any genotype with its restrictions satisfied.
  - If neither A nor B depend on anything, then the expression for fitness is $(1+s_A)\ (1+s_B)$ because, under CBN, there is no epistasis here so $s_{AB}=0$ (look, for example, at the transition rate matrix in Montazeri et al., 2016, Figure 1, for the transition from genotype 1 to genotype 1,2 or from genotype 2 to genotype 1,2).
  - If B depends on A, when we consider the transition from A to B, we can use a single term, $(1+s_X)$ to multiply $(1+s_A)$, and that $s_X=\lambda_B$. That λ_B is the (relative) increase in fitness due to B, when B's restrictions are satisfied (for example, in Example 1 in Montazeri et al., 2016 (Large-scale inference of conjunctive Bayesian networks. Bioinformatics, 32(17), 727–735. https://dx.doi.org/10.1093/bioinformatics/btw459), see the transition rate matrix from genotype 2 to genotype 2,4[5]). You can think of this s_X as the joint combination of the effect of B on its own and the epistasis of A and B; but thinking of B on its own is a moot point, since B on its own (i.e., without A, without its restrictions satisfied) is not a genotype that can be observed.
  - Thus, for any genotype, do $\prod(1+s_i)$, where $s_i=\lambda_i$ when the restrictions are satisfied.

2.4.1 Another example about the relationship between s, λ, s_h

- Remember that having $\lambda_i<0$ makes no sense.

- Suppose a model where A and B depend on no one, D depends on A and C depends on both A and B.

- Simple case:
  - $W_{AD}=(1+\lambda_A)(1+s_D)(1+s_{AD})$
  - $W_{AD}=(1+\lambda_A)(1+\lambda_D)$
  - So: $1+s_{AD}=\frac{1+\lambda_D}{1+s_D}$
  - If $s_D=0$ we get the $s_{AD}=\lambda_D$ or "the epistatic term is equal to the lambda".
  - If $s_D<0$ then the epistatic term, $s_{AD}>\lambda_D$: it has to be large enough to compensate for the decrease in fitness from the single D.

[5] Notice that Figure 1 is correct, but the matrix in Example 1 has a typo, and is missing the entry for λ_4; or look at the transition from 1,2 to 1,2,3 and 1,2,4

  - This can matter if we try to generate $s_{xy\ldots}$ from some distribution and match them to the λ.

- Beware, though, of a simple interpretation of the s_D as s_h, specially when there are more genes. An example:

  - $W_{ADC} = (1 + \lambda_A)(1 + s_D)(1 + s_{AD})(1 + s_C)(1 + s_{DC})(1 + s_{AC})(1 + s_{ACD})$
  - But we can replace the second and third terms:
    - $W_{ADC} = (1 + \lambda_A)(1 + \lambda_D)(1 + s_C)(1 + s_{DC})(1 + s_{AC})(1 + s_{ACD})$
  - OncoSimulR is NOT replacing all the extra terms by s_h.
    - If it did you would get:
      - $W_{ADC} = (1 + \lambda_A)(1 + \lambda_D)(1 + s_h)^4$
    - But what OncoSimul actually gives you is:
      - $W_{ADC} = (1 + \lambda_A)(1 + \lambda_D)(1 + s_h)$
    - Why? Because only one gene, C, has not got its restrictions satisfied.
    - In other words, the number of $(1+s_h)$ is equal to the number of genes (not genes and gene combinations) with their restrictions not satisfied.
  - In particular, note that this is not correct:
    - $W_{ADC} = (1 + \lambda_A)(1 + s_h)(1 + s_{AD})(1 + s_h)(1 + s_h)(1 + s_h)(1 + s_h)$
    - Where the first s_h would correspond to s_D and the rest to C, AC, DC, ACD.
    - And thus, it is not correct to write: $1 + s_{AD} = \frac{1+\lambda_D}{1+s_h}$
  - Of course, if $s_h < 0$ then $W_{ADC} < W_{AD}$.

- And with this same DAG, we can write either:

  - $W_{ABC} = (1 + \lambda_A)(1 + \lambda_B)(1 + \lambda_C)$
  - $W_{ABC} = (1 + \lambda_A)(1 + \lambda_B)(1 + s_C)(1 + s_{AC})(1 + s_{BC})(1 + s_{ABC})$
  - As before we could do: $(1 + s_{ABC}) = \frac{1+\lambda_C}{(1+s_C)(1+s_{AC})(1+s_{BC})}$
  - And this shows again that the epistatic term for ABC (i.e., when restrictions are satisfied) might have to be very large to compensate for large negative fitness effects of mutations without restrictions satisfied (e.g., s_C).

3 What about H-ESBCN/PMCE, with AND, XOR, OR?

By H-ESBCN/PMCE I mean the method described in

- Angaroni, F., Chen, K., Damiani, C., Caravagna, G., Graudenzi, A., & Ramazzotti, D. (2021). PMCE: efficient inference of expressive models of cancer evolution with high prognostic power. Bioinformatics, 38(3), 754–762. http://dx.doi.org/10.1093/bioinformatics/btab717

We can repeat what we did above, with OR and XOR replaced by, well, OR and XOR in OncoSimulR (OR and XOR are also called SM and XMPN in OncoSimulR). OncoSimulR has dealt with OR, XOR, AND, and mixtures of them since many years ago. Remember also that in the H-ESBCN model if a gene depends on a set of genes, it has the same type of dependency on all the genes of that set.

4 OncoBN

What about OncoBN, the method described in Nicol, P. B., Coombes, K. R., Deaver, C., Chkrebtii, O., Paul, S., Toland, A. E., & Asiaee, A. (2021). Oncogenetic network estimation with disjunctive Bayesian networks. Computational and Systems Oncology, 1(2), 1027. http://dx.doi.org/10.1002/cso2.1027?

OncoBN can fit both conjunctive (AND) and disjunctive (OR, not XOR) models; for the first you specify `model = ‘‘CBN’’` and for the second `model = ‘‘DBN’’`. So it resembles CBN and HESBCN. However, the θs returned by OncoBN are not rates, as in CBN, HESBCN, or MHN, but rather probabilities of seeing specific alterations at the time of observation as in OT. So probably a better way to think of OncoBN is as an extension of OT, where nodes can have multiple parents, and the relationship of dependence can be AND or OR (but not both).

We deal with OncoBN as with any other method, but as we do with OT, we do not interpret the parameters as rates. This also means that our transition matrices (again, transition matrices, not transition rate matrices: no transition rate matrices are returned for OT or OncoBN), as for OT, are really an abuse of the untimed oncogenetic model.

When using OncoSimulR to represent OncoBN models, there is nothing new. If OncoBN was fitted specifying “CBN”, we use ANDs, if it used “DBN” we use ORs when computing fitness.

5 MHN

MHN has been described in Schill, R., Solbrig, S., Wettig, T., & Spang, R. (2020). Modelling cancer progression using Mutual Hazard Networks. Bioinformatics, 36(1), 241–249. http://dx.doi.org/10.1093/bioinformatics/btz513.

We cannot use OncoSimulR as for the rest of the modes, because the MHN model is rather peculiar if taken at face value as an evolutionary model (see Diaz-Colunga, J., & Diaz-Uriarte, R (2021). Conditional prediction of consecutive tumor evolution using cancer progression models: What genotype comes next? PLOS Computational Biology, 17(12), 1009055. http://dx.doi.org/10.1371/journal.pcbi.1009055 ; in particular, see section 1.7 of the Supporting Information: https://doi.org/10.1371/journal.pcbi.1009055.s001).

To express MHN in terms of fitness of genotypes, we would need to express it as a model where order of acquisition of mutations matters. This is possible with OncoSimulR[6], but it does not provide any additional intuition, and can lead to a huge number of fitnesses for a genotype (a genotype with k mutated loci could possibly have $k!$ different fitnesses, one for each of its $k!$ different ways of mutation its k loci).

6 Benefits of this exercise with OncoSimulR

- We make the fitness model explicit.
- We can double check the code in `evamtools` for obtaining fitness graphs and transition probabilities as some critical computations are being done with very different code.

7 License and copyright

This work is Copyright, ©, 2021, Ramon Diaz-Uriarte.

[6] We would need to use “order effects” for the fitness specification. See the vignette for OncoSimulR https://rdiaz02.github.io/OncoSimul/OncoSimulR.html#36_Order_effects, and the help for function `allFitnessEffects`.

The source of this document and the EvAM-Tools package is at https://github.com/rdiaz02/EvAM-Tools.

8 Funding

Supported by grant PID2019-111256RB-I00 funded by MCIN/AEI/10.13039/501100011033 and Comunidad de Madrid's PEJ-2019-AI/BMD-13961 to R. Diaz-Uriarte.

MINISTERIO
DE CIENCIA
E INNOVACIÓN

AGENCIA
ESTATAL DE
INVESTIGACIÓN