Internship report

Comparative Analysis of CNA Timing Algorithms in a Simulated Tumor Evolution Model

Madeleine Hueber

April - July 2024

Supervisor : Khanh N. Dinh

École Polytechnique
IP Paris

Columbia
Herbert and Florence Irving
Institute for Cancer Dynamics

Declaration of academic integrity

I, *Madeleine Hueber*, hereby confirm :

— That the results presented in this report are exclusively the outcome of my work.
— That I am the author of this report.
— That no sources or materials were used in this report without being clearly acknowledged according to the recommended bibliographic rules.

I declare that this document cannot be suspected for plagiarism

July 23, 2024

Abstract

The accurate timing of Copy Number Alterations (CNAs) in tumor evolution is crucial for understanding cancer progression and optimizing treatment strategies. This study integrates four distinct algorithms into CINner, a comprehensive cancer simulation model developed by the Irving Institute of Cancer Dynamics (IICD), to compare their performances across various levels of analysis. We specifically focus on evaluating and comparing the performance of three timing algorithms: CancerTiming, MutationTimeR, and GRITIC. The algorithms were assessed based on their ability to predict the occurrence of gains and accurately determine their timing.

Our results demonstrate that GRITIC outperforms the other algorithms in multiple aspects. Above all it enables a more comprehensive detection of gains, particularly in complex genomic regions and in the presence of whole-genome duplication (WGD). CancerTiming and MutationTimeR showed limitations, particularly in handling WGD events and complex gain histories.

Finally, the study explores extending the GRITIC algorithm to single-cell data, which presents additional challenges due to lower coverage and higher rates of false positives, false negatives, and unknown reads. By incorporating single-cell sequencing data into our simulations, we propose an initial step towards addressing these challenges, including a preliminary solution to correct the primary issue of false negatives in single-cell data.

Acknowledgment

I would like to express my gratitude to my supervisor, Khanh Dinh, for his guidance and support throughout this project. His constant mentorship and availability to answer my questions have been extremely helpful.

My sincere thanks also go to Simon Tavaré, the Director of the Irving Institute of Cancer Dynamics (IICD), for always providing valuable insights and continuous encouragement.

I am deeply grateful to Zijin Xiang, a master's student at IICD, for her insightful help and collaboration, as well as her warm welcome and positivity throughout these months.

I would also like to express my thanks to Lorenza Favrot for always making sure everything went well and inviting us to numerous events and conferences.

Finally, I extend my thanks to all the members of IICD for welcoming me and creating a stimulating intellectual atmosphere.

Contents

Introduction

Cancer develops through a complex interplay of genetic alterations that disrupt normal cellular functions [1]. Among these alterations, copy number alterations (CNAs)[2] —involving the gain or loss of genome sections through mechanisms such as missegregation or whole genome duplication—play a major role in cancer development. CNAs facilitate the adaptation of cancer cells to various environmental challenges, contributing significantly to tumor progression and heterogeneity.

Accurately timing CNAs within cancer cells is crucial for understanding tumor evolution and progression. Determining the precise timing of these alterations can provide insights into the sequence of genetic events that drive cancer development. This knowledge is essential for unraveling the mechanisms of tumorigenesis and for developing targeted therapeutic strategies.

Advances in sequencing technologies have made it possible to detect single nucleotide variants (SNVs) across the entire genome using different techniques, including bulk and single-cell sequencing. These mutations [3] provide a rich source of information that can be leveraged to infer the timing of CNAs. By analyzing the co-occurrence and relative abundance of these mutations in relation to CNAs, multiple algorithms have been developed to infer the timing of CNAs based on bulk data.

At the Irving Institute of Cancer Dynamics, my supervisor recently presented CINner: a framework for modeling chromosomal instability during cancer evolution. This framework allows us to test existing algorithms on numerous simulated datasets, enabling a more detailed analysis and evaluation of these algorithms.

Using the CINner framework, our study aims to evaluate and compare various computational algorithms designed to time CNAs in cancer cells. Our goal is to identify the strengths and limitations of current approaches and to extend these methods to single-cell data, which present additional challenges due to data limitations.

1 CINner and origin of the project

The motivation for this project comes from the release of CINner[4] by the Irving Institute of Cancer Dynamics (IICD). CINner is an advanced model that simulates tumor development, encompassing various aspects of cancer progression, including the emergence of driver genes and copy number alterations (CNAs). Given that CNAs and single nucleotide variants (SNVs) are the most common and widespread alterations in cancer genomes, we decided it would be beneficial to add SNV simulation to CINner. This would allow us to study both CNAs and SNVs together using the data generated by CINner.

In particular, this project presents a valuable opportunity to evaluate and compare different approaches proposed in recent years for timing CNAs based on SNV data. By implementing these algorithms within the CINner framework, we can systematically compare their performance on the same simulated datasets. This comparative analysis aims to identify the most effective methods for inferring CNA timing, ultimately contributing to a deeper understanding of cancer genomics.

To adapt CINner for this purpose, our first step in this project was to add the simulation of passenger mutations into the model. Unlike driver mutations that promote cancer growth, passenger mutations do not directly contribute to tumor progression.

The first version of CINner operates through **three key phases**:

1. **Clone Evolution** : In the first phase, CINner simulates the evolution of clones in forward time. The clones are defined as groups of cells that have identical copy number (CN) and mutational characteristics (for driver genes). New clones are generated when CNAs and driver mutations occur, and the clone sizes evolve through time according to the branching process governing cell division and death.

2. **Cell Sampling** : In the second phase, CINner samples cells from predefined time points based on the characteristics of existing clones.

3. **Phylogeny Construction** : In the third step, CINner constructs the phylogeny of the sampled cells : a tree representing the evolutionary relationships among clones over time.

To incorporate passenger mutations into CINner, we introduced an additional phase. In this phase, we start at the origin of the phylogeny tree with no passenger mutations. The algorithm then goes through the phylogeny tree, adjusting inherited mutations for each branch and elapsed genotype based on encountered CNAs (doubling mutations for gains or deleting them for losses) and introducing new mutations according to the mutation rate per cell division.

Algorithm 1 Simulate Passenger Mutations

```
Initialize passenger_mutations as an empty list
for each branch in phylogeny do
    for each elapsed_gen in elapsed_genotypes do
        Retrieve mutations from previous genotype
        Update passenger_mutations according to CNAs:
        if CNA is a gain then
            Double the affected mutations
        else if CNA is a loss then
            Delete the affected mutations
        end if
        Add new passenger mutations based on mutation rate
    end for
end for
```

Finally, the algorithms we used were originally designed for **bulk data sets**. Bulk sequencing involves analyzing a mixture of DNA from many cells, providing an average signal of the genomic alterations present within a tumor. This approach contrasts with single-cell sequencing, which examines the DNA of individual cells, allowing for the detection of heterogeneity within the tumor. To adapt our simulations, we added an algorithm to simulate the bulk sequencing process. This additional layer mimics the pooling of genetic material from multiple cells, mirroring real-world bulk sequencing conditions. By incorporating this simulation, we can more accurately assess the performance of our algorithms in a bulk data context, ensuring our results are relevant and applicable to practical scenarios.

2 Link between CNA timing and SNV

In the context of timing copy number alterations (CNAs), regions with a history of gains are particularly suited for analysis because they preserve valuable single nucleotide variant (SNV) information. Unlike regions with losses, where SNVs are lost along with the genomic material, regions that have only experienced gains retain the original SNV landscape. This retention allows for a more accurate inference of the timing of CNAs, which is why we restrict ourselves to the case of regions with a **gain-only history**.

The timing of a gain can be inferred using SNV data within the gained region. Indeed, mutations that occurred before the gain become duplicated along the chromosomal region, effectively doubling their **multiplicity** — representing the number of allelic copies an SNV is present on. In contrast, mutations that occurred after the gain remain in a single copy with a multiplicity of 1. By analyzing the proportions of mutations with multiplicity one and two (or more), we can estimate the timing of the gain. This estimated time will be measured in **mutation time**, which reflects the accumulation of mutations within a cell.

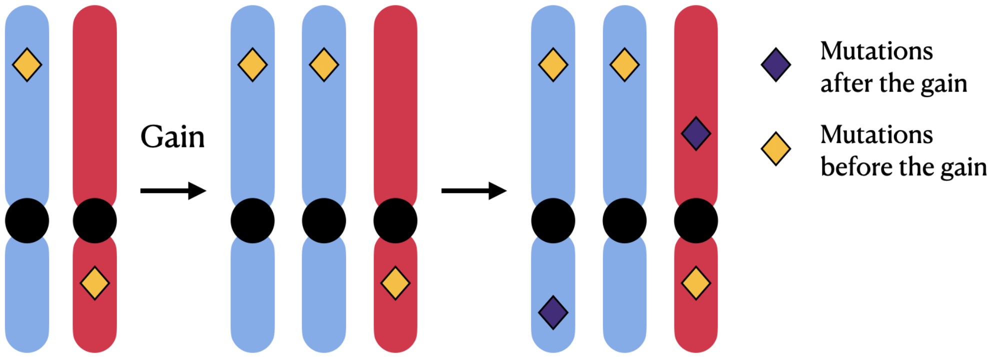


Figure 1: Schema of the accumulation of SNV during tumor development

Mutation time preserves chronological order, making it valuable for constructing timelines. However, it does not directly correlate with real-time due to variations in the mutation rate, which can fluctuate with genome length, copy number alterations (CNAs) and mutational characteristics of the cell. Despite these fluctuations, these factors have minimal impact on our analysis, allowing us to approximate mutation time as linear with respect to real time, which simplifies the comparison between inferred and actual timing of events.

2.1 Simple cases

To begin with, we can consider four different simple gain histories :

Single gain

Single gains occur when a segment of the genome is duplicated once, resulting in an increase in copy number from two to three. It typically happens after a **missegregation** event, when parts of chromosomes are incorrectly distributed between progeny cells during cell division, leading to an abnormal copy number.

We consider a region of the genome that has undergone a single gain, meaning there have been two stages in the tumor's life: before the gain and after the gain. Our goal is to infer $\pi = (\pi_0, \pi_1)$, representing the time (in mutation time) spent in each stage, with particular interest in π_0, which indicates the timing of the gain.

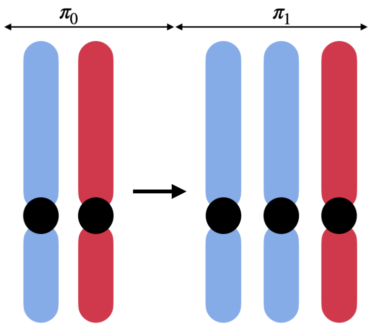


Figure 2: Schema of a single gain leading to copy number state 2+1

All mutations that occurred in the gained copy before the gain will have a multiplicity of 2 whereas others will have a multiplicity of 1. This results in the following equations :

$$N_1 = R(\pi_0 + 3\pi_1)$$

$$N_2 = R\pi_0$$

Where N_1, N_2 represent the number of mutations having multiplicity 1 and 2 respectively, and R is a constant relative to mutation time.
As $\pi_0 + \pi_1 = 1$ we can derive

$$\pi_0 = \frac{3q_2}{q_1 + 2q_2}$$

with $q_i = \frac{N_i}{N_1+N_2}$ the proportion of mutations having multiplicity i.

For other types of gains, we can have similar equations :

CNLOH

Copy-neutral loss of heterozygosity (CNLOH) occurs when one parental allelic copy is lost and the remaining copy is duplicated, resulting in two identical copies without a change in overall copy number.

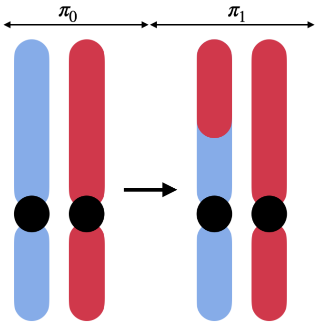


Figure 3: Schema of CNLOH leading to copy number state 2+0

$$N_1 = 2R\pi_1$$

$$N_2 = R\pi_0$$

so $\pi_0 = \frac{2q_2}{q_1+2q_2}$

Double allelic gain

Double allelic gain occur when both alleles are gained simultaneously, which typically happen during **whole-genome duplication**.

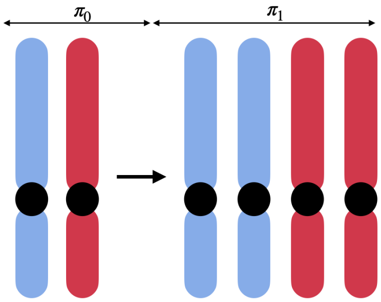


Figure 4: Schema of a double allelic gain leading to copy number state 2+2

$$N_1 = 4R\pi_1$$
$$N_2 = 2R\pi_0$$

so $\pi_0 = \frac{2q_2}{q_1+2q_2}$

Unbalanced two gains

Unbalanced two gains occur when one allele is gained twice and nothing happens to the other.

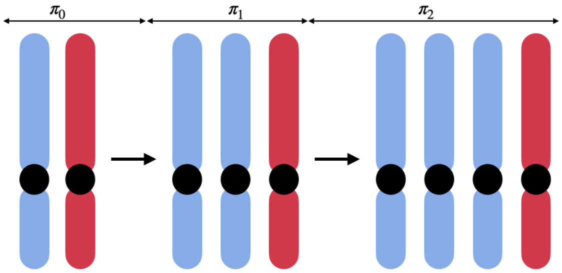


Figure 5: Schema of unbalanced two gains leading to copy number state 3+1

$$N_1 = R(\pi_0 + \pi_1 + 4\pi_2)$$
$$N_2 = R\pi_1$$
$$N_3 = R\pi_0$$

so $\pi_0 = \frac{4q_3}{q_1+2q_2+3q_3}$ and $\pi_1 = \frac{4q_2}{q_1+2q_2+3q_3}$

These equations can be written as the general form :

$$\pi = \frac{1}{c_q} \times Aq \ (*)$$

where $c_q = \sum_{i=1}^{M} iq_i$, A is a matrix reflecting the gain history and q is the vector representing the multiplicity proportions.

For genomic regions with higher allelic copy numbers, accurately timing the gains becomes more challenging. Different gain histories can lead to the same copy number state (see example below), but each gain history corresponds to a distinct set of equations. Due to the lack of information needed to determine the specific gain history that occurred, we first limit our analysis to the previously mentioned cases.

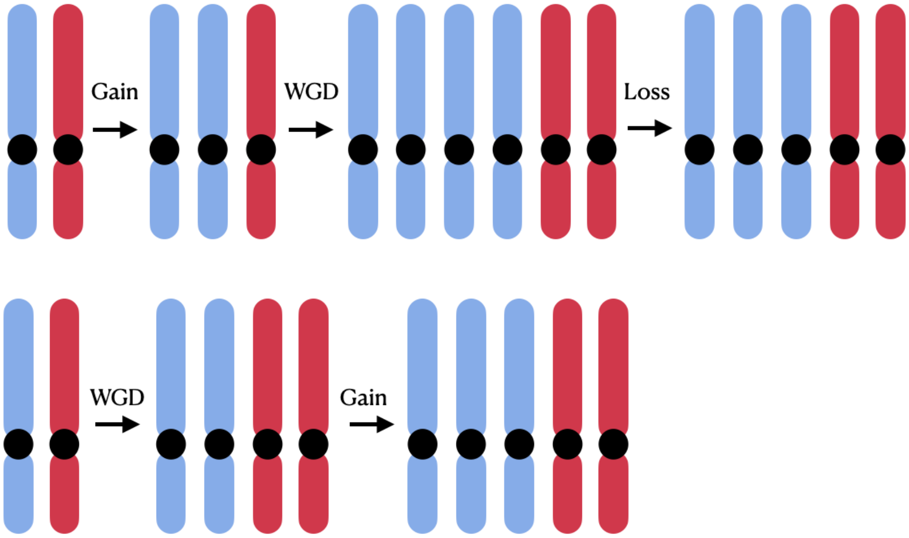


Figure 6: Example of two gain histories leading to the same copy number 3+2 state

2.2 Framework to infer the timings

To accurately infer the timing of CNAs, we must make certain assumptions and restrict our analysis to specific cases. First, we assumed that **mutation time is linear with respect to real time**, allowing us to compare the inferred timing π_0 to $\frac{T_{true\_time}}{T_f}$. Additionally, for the previous equations to hold, we assume an **infinite genome model**, where each position in the genome can only be mutated once.

As previously mentioned, we restrict our analysis to regions with a gain-only history. For the initial three algorithms, we further limit our focus to regions that went through a gain history listed above. The detection of these gains is based on the copy number profile provided for a sample, which specifies the major (M) and minor (m) copy number for each position.
For a segment of the genome :

- If M=2 and m=1 we consider that this segment has undergone a single gain
- If M=2 and m=0 we consider that this segment has undergone a CNLOH
- If M=2 and m=2 we consider that this segment has undergone a double allelic gain.
- If M=3 and m=1 we consider that this segment has undergone an unbalanced two gains.

If the copy number profile does not fit these specific cases, the algorithms do not time the corresponding regions.

Additionally, CNAs and SNVs can be classified based on their state within a tumor. Typically, a tumor sample has one clonal state and multiple subclonal states. A genomic alteration in a clonal state is present in all cancer cells within the tumor (having occurred before the most recent common ancestor (MRCA)). In contrast, a subclonal alteration arises after the MRCA and can only be found in a subset of cells.
Clonal CNAs, being a part of all tumor cells history, typically accumulate more associated mutations over time. This accumulation provides a richer dataset for timing analysis, which improves the accuracy and reliability of our conclusions. Therefore, we decided to focus exclusively on timing clonal CNAs to ensure strong and accurate sequencing of genetic events in tumor evolution.

3 Existing Algorithms

Four algorithms exist for inferring CNA timing based on SNV data: Cancertiming, MutationTimeR, Phylogic NDT and GRITIC. The goal of this project was to implement these algorithms and adapt them for use with CINner, a comprehensive model that simulates the development of cancer cells. Integrating these algorithms into CINner will enable us to compare their performance and determine the most effective one for application to real data.

For all following algorithms we consider a region of the genome that went through S events resulting in T = (M+m) copies of the region. According to our restrictions explained earlier, we restrict ourselves first with $S \in \{1, 2\}$ and $T \in \{2, 3, 4\}$.
We have :
- $\pi = (\pi_0, ..., \pi_S)$ the time between each event (in mutation time)
- $q = (q_1, ..., q_T)$ the multiplicity proportion over the region of the genome.
- N = the number of mutations on this region

And for the mutation at location i we have :
- X_i = the variant read count
- m_i = the total read count
- C_i = the multiplicity of the mutation
- P_i = the Variant Allele Frequency, which represents the proportion of the mutation present in the cell population.

The goal of these algorithms is to estimate the vector q representing the multiplicity proportion in order to calculate the vector π which gives the time information based on previous equations $(*)$. The purpose of this section is to present the principles of each algorithm.

3.1 CancerTiming

CancerTiming [5] is an algorithm able to time CNLOH, single gain and double allelic gains by inferring the multiplicity proportion vector q.

The set of possible variant allele frequency (VAF) of a mutation i is given by $P_i \in \{\frac{1}{T}, ..., \frac{T}{T}\}$ for a pure tumor sample, and in general cases given by $P_i \in \{a_1, ..., a_T\}$. We note for $j \in \{1, ..., T\}$ $q_j = \mathbb{P}(P_i = a_j | P_i > 0)$.

In this algorithm we model $X_i \sim Binomial(m_i, P_i)$

The log likelihood of $X = (X_1, ..., X_N)$ is then given by :

$$L(X,q) = \sum_i^N \log(\mathbb{P}(X_i|X_i > 0, q)) = \sum_i^N \log\left(\frac{\sum_{j=1}^S \mathbb{P}(X_i|P_i = a_j)q_j}{\mathbb{P}(X_i > 0|q_j)}\right) = \sum_i^N \log\left(\frac{\sum_{j=1}^S \mathbb{P}(X_i|P_i = a_j)q_j}{1 - \sum_j (1-a_j)^{m_i} q_j}\right)$$

To maximize this likelihood, CancerTiming uses an **Expectation - Maximization (EM) algorithm** which consists of repeating a certain amount of time an E-step (Expectation step) and an M-step (Maximization step).

Here the **E-step** gives us :

$$Q(q|q^{(t)}) = \mathbb{E}_{X,q^{(t)}}[\log(p(X,P|q))] = \sum_j Y_j \log(q_j) - \sum_i \log(1 - \sum_j (1-a_j)^{m_i} q_j) + constant$$

with $Y_j = \sum_i (\mathbb{P}(P_i = a_j | X_i, q^{(t)})$

We then need to maximize it on the set of feasible q. The constraints on q are the following : because it is a probability vector we need to have $1^T q = 1$ and $q \geq 0$ and because of the equation $(*)$ we need to have $Aq \geq 0$. So the space of feasible q is : $\Omega = \{q : 1^T q = 1, q \geq 0, Aq \geq 0\}$.

And the **M-step** gives us :

$$q^{(t+1)} = \text{argmax}_{q \in \Omega}\ Q(q|q^{(t)})$$

This EM - algorithm give us an estimation $\hat{q}$ that is then used to infer $\hat{\pi}$ according to $(*)$.

3.2 MutationTimeR

MutationTimeR [6]is an algorithm designed to time CNLOH, simple gains and double allelic gains.

In this algorithm, we model $X_i \sim BetaBin(m_i, P_i, \rho)$ with ρ a dispersion parameter ($\rho = 0.01$). and the VAF is calculated through $P_i = \frac{f_0 C_i}{f_0 T + 2(1-f_0)}$ where f_0 is the tumor purity

We note $\underline{C}_i$ and $\underline{S}_i$ the random variables that represent the unknown multiplicity and state of the mutation i (clonal or subclonal states).
And we consider our observations Y_i as $Y_i | X_i = X_i$ if $X_i > 3$ else absent.

The goal of the algorithm is to infer the multiplicity and state of each mutation in order to compute the multiplicity probabilities. To do so, MutationTimeR computes the probabilities $\mathbb{P}(\underline{C}|\underline{S}, Y)$.

Using **Bayes formula**, we have :

$$\mathbb{P}(\underline{C}, \underline{S}, Y) = \mathbb{P}(Y|\underline{C}, \underline{S}) \times \mathbb{P}(\underline{C}|\underline{S}) \times \mathbb{P}(\underline{S}) = \frac{\mathbb{P}(X = Y|\underline{C}, \underline{S}) \times \mathbb{P}(\underline{C}|\underline{S}) \times \mathbb{P}(\underline{S})}{\mathbb{P}(X < 3|\underline{C}, \underline{S})}$$

Which can be written as :

$$\mathbb{P}(\underline{C}, \underline{S}, Y) = \frac{\mathbb{P}(X = Y|\underline{C}, \underline{S}) \times \mathbb{P}(\underline{C}|\underline{S}) \times \mathbb{P}(\underline{S})}{Pow(\underline{C}|\underline{S}) \times Pow(\underline{S})}$$

where $Pow(\underline{C}|\underline{S}) = \mathbb{P}(X < 3, \underline{C}|\underline{S})$ and $Pow(\underline{S}) = \mathbb{P}(X < 3|\underline{S})$ are the probability of detecting mutations for a particular multiplicity $\underline{C}$ for a given state $\underline{S}$ and for a given state $\underline{S}$. MutationtimeR then computes $\mathbb{P}(\underline{C}|\underline{S}, Y)$ through iterations by computing the following probabilities at each iteration :

- $\mathbb{P}(\underline{C}|\underline{S})$ is computed as the average of $\mathbb{P}(\underline{C}|\underline{S}, Y)$ over all observations Y
- $\mathbb{P}(\underline{C}, \underline{S}, Y_i) = \frac{\mathbb{P}(X_i = Y_i|\underline{C}, \underline{S}) \times \mathbb{P}(\underline{C}|\underline{S}) \times \mathbb{P}(\underline{S})}{Pow(\underline{C}|\underline{S}) \times Pow(\underline{S})}$
- $\mathbb{P}(\underline{C}, \underline{S}|Y_i) = \frac{\mathbb{P}(Y_i, \underline{C}, \underline{S})}{\sum_{S', C'} \mathbb{P}(Y_i, C', S')}$
- $\mathbb{P}(\underline{C}, \underline{S}|Y) = \mathbb{E}[\mathbb{P}(\underline{C}, \underline{S}|Y_i)]$
- $\mathbb{P}(\underline{C}|\underline{S}, Y) = \mathbb{P}(\underline{C}, \underline{S}|Y) \times \mathbb{P}(\underline{S})$

We then use a **MAP estimate** to assign to each mutation a state and a multiplicity :
$c, s = argmax\mathbb{P}(\underline{C}, \underline{S}|Y)$
Then we can easily compute $q_C = \mathbb{P}(C|S = clonal)$ and go back to π through the previous equations $(*)$.

3.3 PhylogicNDT

In this algorithm [7] we model the **likelihood of a mutation i having multiplicity C_i** as :
$L(C_i) = B(X_i; m_i, P_i(C_i))$, where $B(k; n, p)$ denotes the binomial distribution's probability mass function and $P_i = \frac{f_0 C_i}{f_0 T + 2(1 - f_0)}$.

We assume a **uniform prior distribution** for π in $[0, 1]$ which gives us the prior distribution of q with the equations described earlier $(*)$.
For the timing of a single gain for example we then build a posterior probability of the mutation having multiplicity 2, for each mutation i:

$$f_i(p) = F_i(c_1) \times p + (1 - p) \times F_i(c_2)$$

where $F_i(c_k) = \frac{L(c_k)}{L(c_1)+L(c_2)}$

Then the **posterior distribution** on the overall quantity q_2 is :

$$q_2(p) = prior\_q_2 \times \prod_{i=1}^{N} f_i(p)$$

Then we can go back to the posterior distribution of π with the previous equations $(*)$ and take the inferred timing as the mean.
The same procedure is adapted to time regions with different gain history.

PhylogicNDT introduces a significant improvement over previous algorithms in the treatment of Whole Genome Duplication (WGD). Indeed, CancerTiming and MutationTimeR, cannot identify samples as WGD or non-WGD and can only compute double allelic gains for separate genomic regions. With PhylogicNDT the algorithm first identifies whether the sample has undergone WGD by examining the most common copy number. If the most common copy number is one, the sample is identified as non-WGD. If it is greater than two, the sample is identified as WGD. If the major copy number is two, the algorithm further examines segments with a major copy number of two. The algorithm times each segment individually and computes the point of maximum overlap, which is the point that overlaps with the maximum of their confidence intervals. If this point overlaps with more than 60% of the gains, the sample is identified as WGD.
If the sample is identified as WGD, the algorithm determines the timing of the WGD by jointly timing a single gain across all segments with a major copy number of two.

3.4 GRITIC

The algorithm GRITIC [8] is based on the same model as PhylogicNDT, but differs in one main aspect : the timing of gains with complex histories.

As we have seen, regions with higher copy numbers are more challenging to time due to the uncertainty in their gain history. To address this issue, GRITIC presents a method based on **binary trees**. For a given copy number information (major and minor) there are multiple routes leading to this same copy number. These routes can be represented as binary trees, where each node represents a CNA event for one allele. Leaf nodes represent the observed alleles, while ancestral nodes represent a copy number gain. These nodes are classified based on the type of CNA (gain or WGD). The edges between two nodes represent the inheritance pattern between time periods.

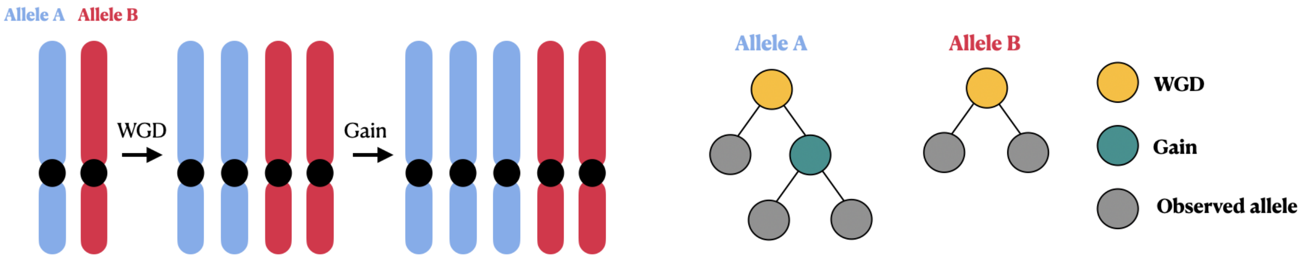


Figure 7: Example of the tree representation associated to one gain history

With a recursive approach, we can generate all possible binary trees leading to a particular copy number state. We restrict ourselves to the cases with only one WGD event (if the sample is identified as WGD, 0 if not). We can then use this representation to time complex gains by associating one equation system to each binary tree representing one route.

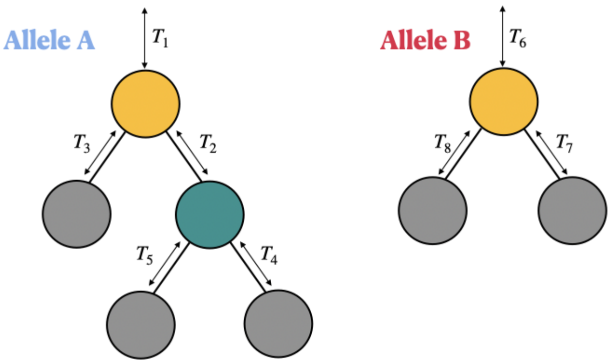


Figure 8: Example of binary trees

The relation between the time and the number of mutation with a certain multiplicity is denoted by the equation :

$$RA_MT = N \ (1)$$

where N and T are the vectors encoding SNV multiplicities and time periods, and A_M is a binary matrix.
In our example, we have :

$$R\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \end{pmatrix} = \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix}$$

There are also two constraints that apply to the vector T : First, all paths from root to leaf must sum to one. Additionally, the WGD is simultaneous, and its timing is predefined equal to T_{WGD}. These constraints are represented in the following equation :

$$A_C T = C \ (2)$$

where A_C is a binary matrix and C the vector encoding timing constraints.
In our example, we have :

$$R \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ T_{WGD} \\ T_{WGD} \end{pmatrix}$$

We can then use computational methods to find a set $\mathbb{T} = \{T, A_C T = C\}$ and then use the equation (1) to compute the associated multiplicity proportions. Each possible route is now associated to a set of possible multiplicity proportions. Moreover, it can be shown that the space of possible multiplicity proportions is often distinct for different routes, which allow us to distinguish routes using SNV information.

The algorithm then runs as follows :

- **Sampling timing :** For each possible route, the algorithm samples a set of possible timing from $\mathbb{T}$ and calculates the associated multiplicities proportions.
- **Evaluating likelihood :** The likelihood of each sample gain is then evaluated by evaluating the likelihood of the multiplicity proportion vector q :

  $$P(q) \propto \prod_j \sum_{i \in Q} q_i P(i|m+M, M, X_j, m_j)$$

  where $Q = \{1, ..., M\}$ and $P(i|T, M, X_j, m_j) \propto B(X_j; m_j, P_j(i))$ and $B(X_j; m_j, P_j(i))$ is the binomial distribution's mass function and as before $P_j(i)$ is the VAF for mutation j with multiplicity i : $P_j(i) = \frac{f_0 i}{f_0 T + 2(1-f_0)}$
  This provides the likelihood for the route's sample gain.
- **Calculating Route Probability :** The probability of each route is calculated by taking the average likelihood of its gain samples and dividing it by the sum of the average likelihoods for all possible routes.
- **Storing Probabilities and Likelihoods :** This probability is then stored along with the likelihood of gain timing for this route and the SNV multiplicities.

Finally, for each segment, the route with the highest probability is chosen, and the timing is determined as the median of the likelihood distribution.

4 Results on bulk data

After implementing these four algorithms into CINner, we aimed to compare their performances. The algorithm PhylogicNDT was not optimized for large-scale analyses and took too long to compute compared to the others. Since it was based on the same principles as GRITIC, we decided to focus on three algorithms for further analysis: CancerTiming, MutationTimeR, and GRITIC. We evaluated their performances through two main aspects: the accuracy in predicting the occurrence of gains and the accuracy in predicting the timing of these gains.

4.1 Predicting the occurrence of gains

To measure the performance of the algorithms, we used two metrics : **recall** and **precision**. Recall (p_1) represents the fraction of actual gains that were correctly predicted. And precision (p_2) represents the fraction of predicted gains that were actual gains .

$$p_1 = \frac{\#\text{ gains predicted and real}}{\#\text{gains real}} \text{ and } p_2 = \frac{\#\text{ gains predicted and real}}{\#\text{gains predicted}}$$

Each gain can be characterized by its location (chromosome), its type (missegregation gain, WGD, or CNLOH), and, except for WGD, its group. Gains are divided into three groups: the first group includes gains that are the only gain in a genomic region, the second group includes the first gain out of multiple gains in a genomic region, and the third group includes subsequent gains (second, third, etc.) in a genomic region.
We ran 200 simulations where the probability of a WGD event is zero and 200 simulations with a non-zero probability of having WGD. On these 200 simulations, a little bit more than 125 were actual WGD samples. These two batch of simulations are analyzed separately, as the presence of WGD has a significant impact and is treated differently by the various algorithms.

We then plotted histograms for p_1 and p_2 over the 200 simulations for each algorithm and filter the results according to chromosome, type, or group. Some of these comparative histograms can be found in the supplementary data (Figures 1-7)
These histograms reveal distinct differences in performance among the algorithms.

Non-WGD samples:

For simulations with a zero probability of encountering WGD, both MutationTimeR and GRITIC show robust results for p_2. This high p_2 suggests a strong confidence in the inferred gains, although CancerTiming performs slightly worse in this respect. The lower performance of CancerTiming may be attributed to its inability to identify double allelic gains.

In terms of p_1 GRITIC outperforms the other algorithms, which seems logical given its broader range of gain detection. On average, GRITIC detects more than half of the true gains, while MutationTimeR detects around 40%, and CancerTiming detects even less, approximately 30%. The disparity is particularly pronounced for group 3 gains (second CNAs), which is expected as CancerTiming and MutationTimeR are not designed to detect these gains effectively, as they are associated to high copy number.

WGD samples:

For simulations with a non-zero probability of encountering WGD, GRITIC maintains a high p_2 around 1, while both CancerTiming and MutationTimeR show a significant drop in p_2 compared to non-WGD simulations. This result is anticipated since these algorithms do not time segments with too high copy number, which is often the case on WGD samples.

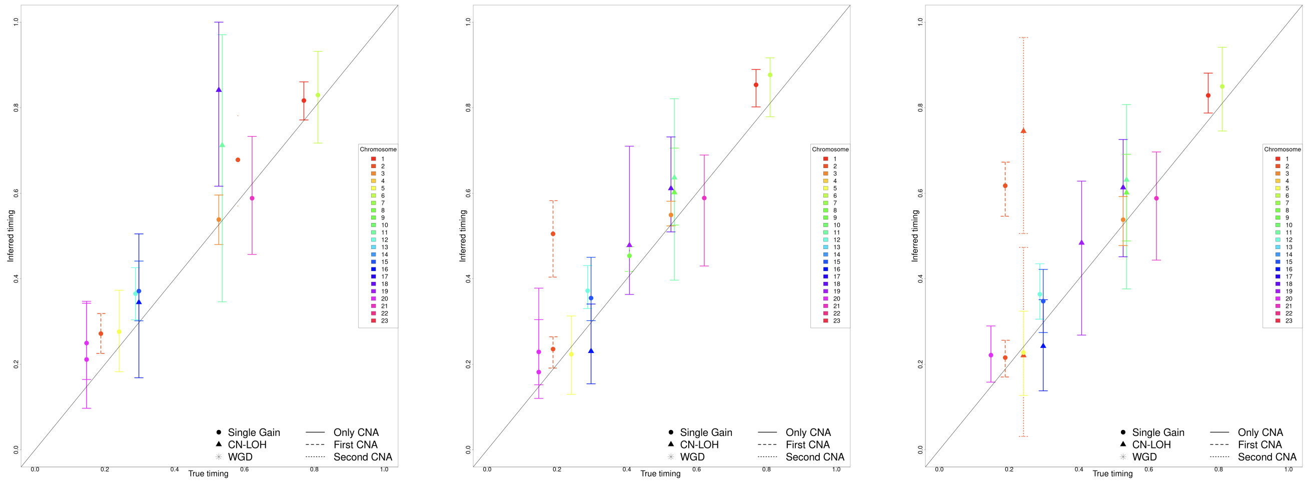
Interestingly, p_1 for MutationTimeR seems to improve for WGD samples because the algorithm considers double allelic gains, which are predominantly present in WGD samples. On the other hand, CancerTiming exhibits a substantial drop in performance, again due to its failure to account for double allelic gains, which are critical in WGD samples.

Overall, GRITIC achieves superior results compared to the other algorithms, particularly in terms of p_2. This is especially evident for WGD events, which GRITIC infers accurately as a global event, unlike the other algorithms.

These results underscore the effectiveness of GRITIC in detecting CNAs across different types of gain histories, highlighting its advantage in comprehensive and accurate gain detection.

4.2 Predicting the timing of the gains

For the actual gains that are predicted by an algorithm, we can compare and the inferred timing to real timing. We can first analyze results for one simulation :



(a) (b) (c)

Figure 9: Comparative analysis of the CNA timing for one simulation with CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples

These plots give us multiple information about the performance of the algorithm: it allows us to compare the deviation between true timing and inferred timing as well as the confidence interval of the inferred timing and all that depending on the chromosome, the type or the group. These plots, made for one simulation, already show us that the three algorithms tend to have similar results concerning the effectiveness of their timing.

To have a more precise analysis, we have, as before, computed histograms for the standard deviation and the confidence interval length over 200 simulations with a zero probability of WGD and over 200 simulations with a non-zero probability of WGD. These histograms can also be filtered by chromosome, group and type. Some of these histograms can be found in the supplementary data (Figures 8-13).

Our observations seem pretty consistent across non-WGD and WGD samples:

- **CI Length Histograms:** The CI length histograms exhibit similar profiles across the different algorithms. The mean CI length is approximately 0.13 for CancerTiming, MutationTimeR, and GRITIC, indicating comparable precision in their timing estimates.
- **Standard Deviation:** The deviation results show that CancerTiming and MutationTimeR tend to have deviations closer to zero. However, it is important to note that GRITIC is also timing more complex gains, which may contribute to its slightly higher deviation. When focusing on simpler gains from group 1, GRITIC's histogram demonstrates better performance.

These histograms are not only indicative of the algorithms' performance but also provide insights into the types of gains that are most represented and the chromosomes that encounter the most gains. This information can be leveraged for further detailed analyses.

Overall, while all three algorithms show similar precision in timing gains, GRITIC's ability to handle more complex gain scenarios gives it a consistent advantage, particularly in situations involving more intricate genomic alterations.

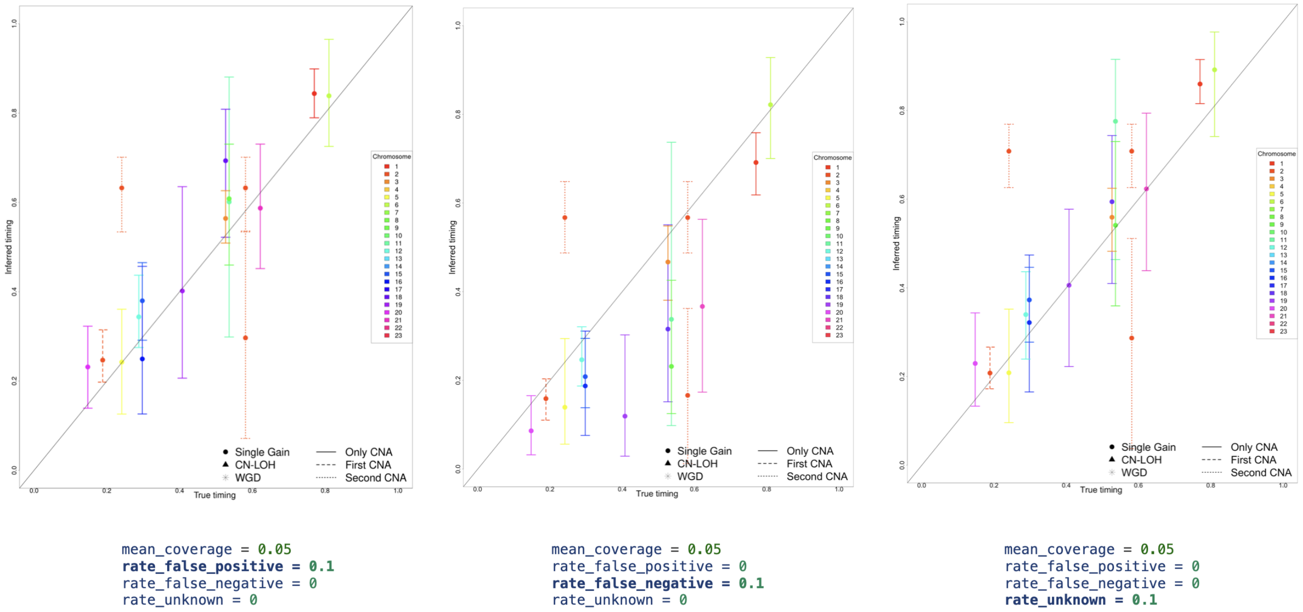
5 Results on single cell data

The previous algorithms were designed for bulk data sets, but oncologists are increasingly utilizing single-cell data in their analyses. Single-cell data provides a more granular view of the tumor's heterogeneity by analyzing individual cells, as opposed to averaging signals from numerous cells in bulk data. Given the superior performance of GRITIC in timing CNAs using SNV data, as demonstrated in our previous analyses, we sought to assess its effectiveness on single-cell data. However, this presented several challenges due to the inherent limitations of single-cell sequencing: the coverage is very low (around 0.05x, i.e. on average 0.05 reads covering a genomic location), and there are issues with false positives, false negatives, and unknown reads.

To test GRITIC on single-cell data, we needed to simulate single-cell sequencing data sets. When simulating single-cell data sets, we considered four key parameters:

- Mean coverage: Typically set to 0.05
- False positive rate
- False negative rate
- Unknown rate

To understand the impact of each parameter on the results, we conducted one simulation on CINner and simulated the single cell sequencing with a mean coverage of 0.05. Then, we varied one of the error rates (false positive, false negative, or unknown rate) to 0.1 while keeping the others at zero. This allowed us to isolate the effect of each parameter.



```
mean_coverage = 0.05
rate_false_positive = 0.1
rate_false_negative = 0
rate_unknown = 0
```

```
mean_coverage = 0.05
rate_false_positive = 0
rate_false_negative = 0.1
rate_unknown = 0
```

```
mean_coverage = 0.05
rate_false_positive = 0
rate_false_negative = 0
rate_unknown = 0.1
```

Figure 10: Results of GRITIC for simulated single cell data with different parameters values.

The figure above revealed that the most significant challenge arises from the false negative rate. This issue results in missed detections of true mutations, which can substantially affect the accuracy of the inferred CNA timings. We then decided to focus on the parameter with the greatest impact : the false negative rate (μ) and find a way to limit its effect, in order to have satisfying results with GRITIC on single cell data.

To address the impact of false negatives, we decided to introduce a **correction to the multiplicity likelihood**. With bulk data, the likelihood of a mutation i having multiplicity C is :

$$P(C|T, M, X_i, m_i) \propto B(X_i; m_i, P_i(C))$$

To account for false negatives, we change the value of this likelihood to :

$$P(C|T, M, X_i, m_i) = \sum_{k=0}^{C} B(X_i; m_i, P_i(C, k)) \times B(k; C, \mu)$$

The variable k represents the number of false negatives reads of mutation i. And $P_i(C, K)$ is the corrected VAF : $P_i(C, K) = \frac{f_0(C-k)}{f_0 T + 2(1-f_0)}$.

We then compared the performance of GRITIC on 200 simulations with and without the correction:

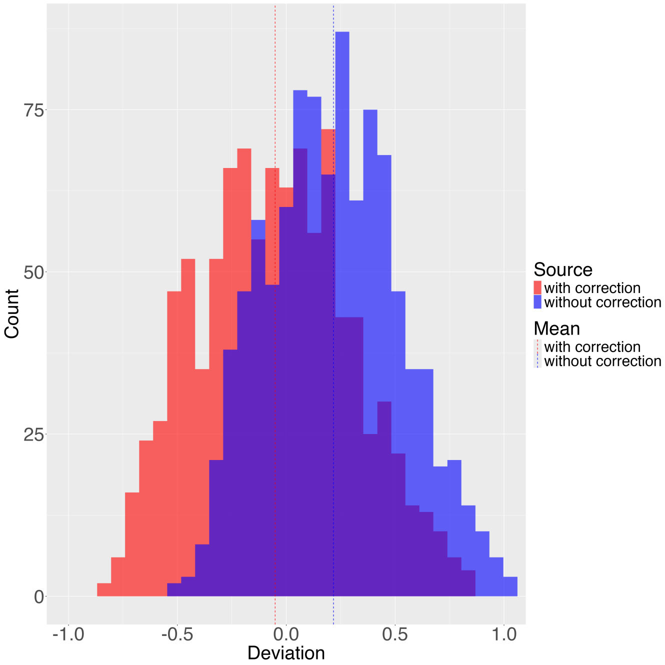


Figure 11: Compared Deviation histogram with and without the likelihood correction

The results show that the correction has successfully re-centered the deviation around 0, leading to improved accuracy. However, the deviation still exhibits significant variability after the correction, indicating an issue that needs to be addressed in future work. Ultimately, by also accounting for false positives and unknown reads, this correction can be further refined to incorporate these additional factors.

Conclusion

This study successfully demonstrates the enhanced capabilities of the CINner simulation model in analyzing the timing of Copy Number Alterations (CNAs) by integrating the simulation of single nucleotide variants (SNVs). Through a comparative analysis of three timing algorithms—CancerTiming, MutationTimeR and GRITIC—we have shown that GRITIC consistently outperforms the others, primarily due to its ability to enable a more comprehensive detection of gains, particularly in complex genomic regions and in the presence of whole-genome duplication (WGD).

The strong performance of GRITIC in detecting and timing CNAs encourages its extension to single-cell data. However, this extension faces notable challenges, primarily due to the inherent limitations of single-cell sequencing data, such as low coverage, false positives, false negatives, and unknown reads. Our study proposes a promising approach to overcome these challenges, focusing on addressing the issue of false negatives. Nonetheless, further work is needed to improve our approach and develop comprehensive strategies that also account for false positives and unknown reads.

In summary, this study has demonstrated the broad applicability of the GRITIC timing algorithm. However, future steps should aim to extend this range even further. Beyond the challenge of single-cell data, the initial restrictions of our study also present obstacles. One important challenge to consider could be the timing of subclonal gains. By addressing these issues, we could enhance our understanding of genomic alterations and cancer evolution.

References

1. Hanahan D., & Weinberg R. A. (2011). Hallmarks of Cancer: The Next Generation. *Cell*, *144*(5), 646–674. https://doi.org/10.1016/j.cell.2011.02.013

2. Martincorena I., & Campbell P. J. (2015). Somatic mutation in cancer and normal cells. *Science*, *349*(6255), 1483–1489. https://doi.org/10.1126/science.aab4082

3. Zack T. I., Schumacher S. E., Carter S. L., Cherniack A. D., Saksena G., Tabak B., Lawrence M. S., Zhang C. Z., Wala J., Mermel C. H., Sougnez C., Gabriel S. B., Hernandez B., Shen H., Laird P. W., Getz G., Meyerson M., & Beroukhim R. (2013). Pan-cancer patterns of somatic copy number alteration. *Nature Genetics*, *45*(10), 1134–1140. https://doi.org/10.1038/ng.2760

4. Dinh K. N., Vázquez-García I., Chan A., Malhotra R., Weiner A., McPherson A. W., & Tavaré S. (2024). CINner: modeling and simulation of chromosomal instability in cancer at single-cell resolution. *bioRxiv (Cold Spring Harbor Laboratory).* https://doi.org/10.1101/2024.04.03.587939

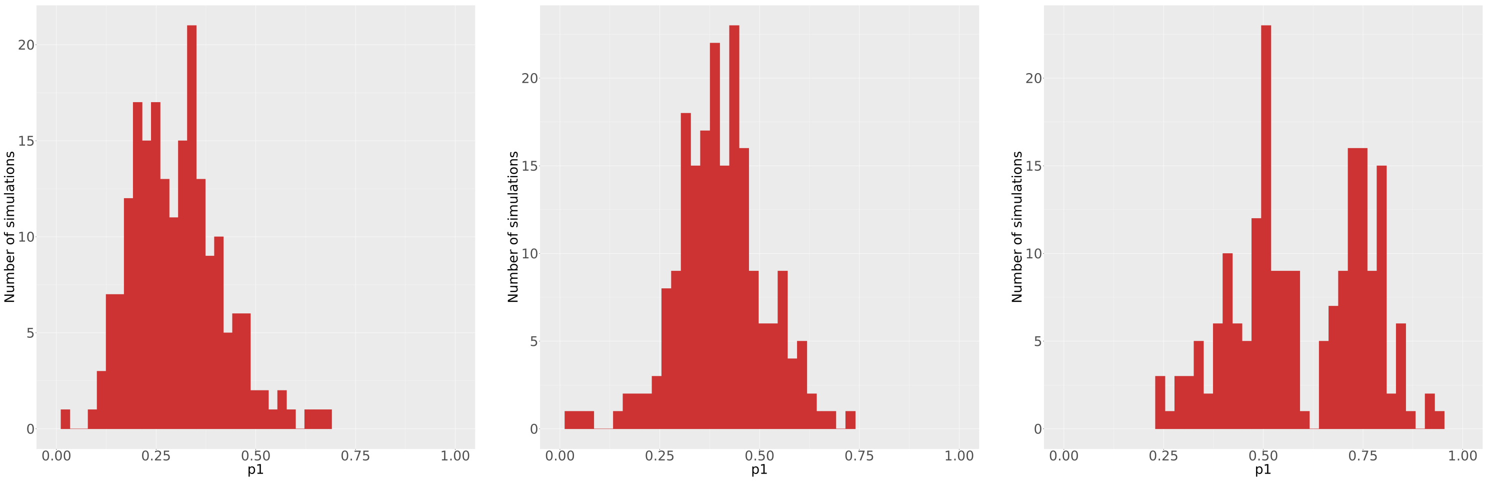
5. Purdom E., Ho C., Grasso C. S., Quist M. J., Cho R. J., & Spellman P. (2013). Methods and challenges in timing chromosomal abnormalities within cancer samples. *Bioinformatics*, *29*(24), 3113–3120. https://doi.org/10.1093/bioinformatics/btt546

6. Gerstung M., Jolly C., Leshchiner I., Dentro S. C., Gonzalez S., Rosebrock D., Mitchell T. J., Rubanova Y., Anur P., Yu K., Tarabichi M., Deshwar A., Wintersinger J., Kleinheinz K., Vázquez-García I., Haase K., Jerman L., Sengupta S., Macintyre G., Von Mering C. (2020). The evolutionary history of 2,658 cancers. *Nature*, *578*(7793), 122–128. https://doi.org/10.1038/s41586-019-1907-7

7. Leshchiner I. et al. Comprehensive analysis of tumour initiation, spatial and temporal progression under multiple lines of treatment. (2019) *Preprint at bioRxiv* https://doi.org/10.1101/508127 .

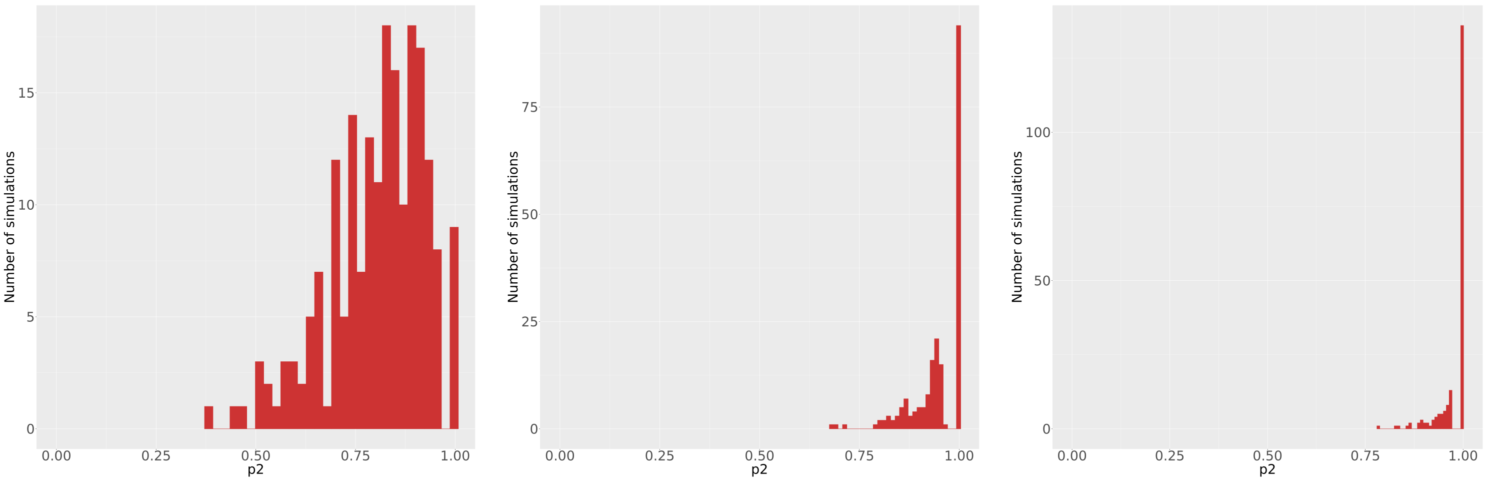
8. Baker T. M., Lai S., Lynch A. R., Lesluyes T., Yan H., Ogilvie H. A., Verfaillie A., Dentro S., Bowes A. L., Pillay N., Flanagan A. M., Swanton,C., Spellman P. T., Tarabichi M. & Van Loo P. (2024). The history of chromosomal instability in genome doubled tumors. *Cancer Discovery.* https://doi.org/10.1158/2159-8290.cd-23-1249

Supplementary data



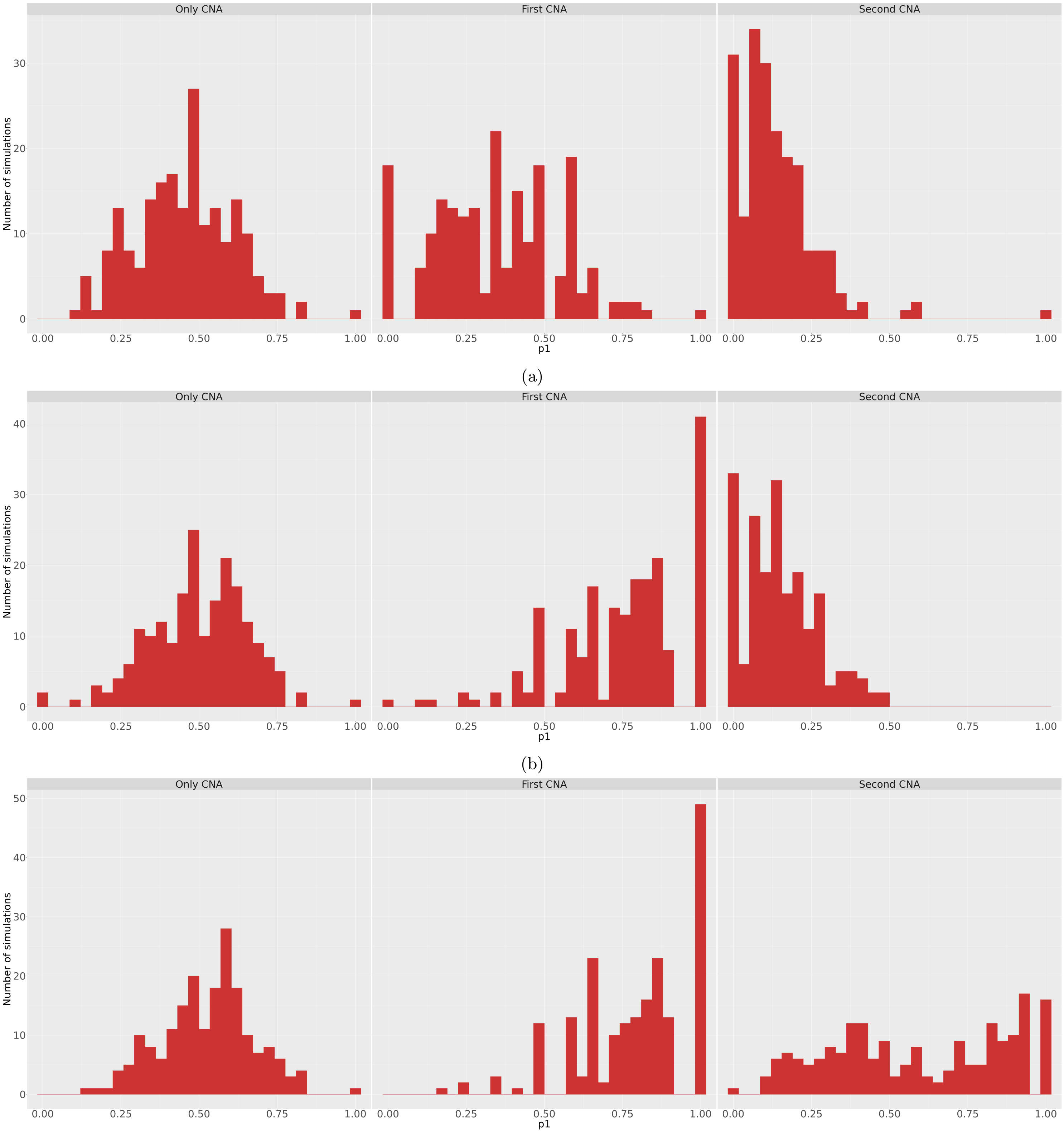
(a) (b) (c)

Figure 1: p_1 histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples



(a) (b) (c)

Figure 2: p_2 histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples

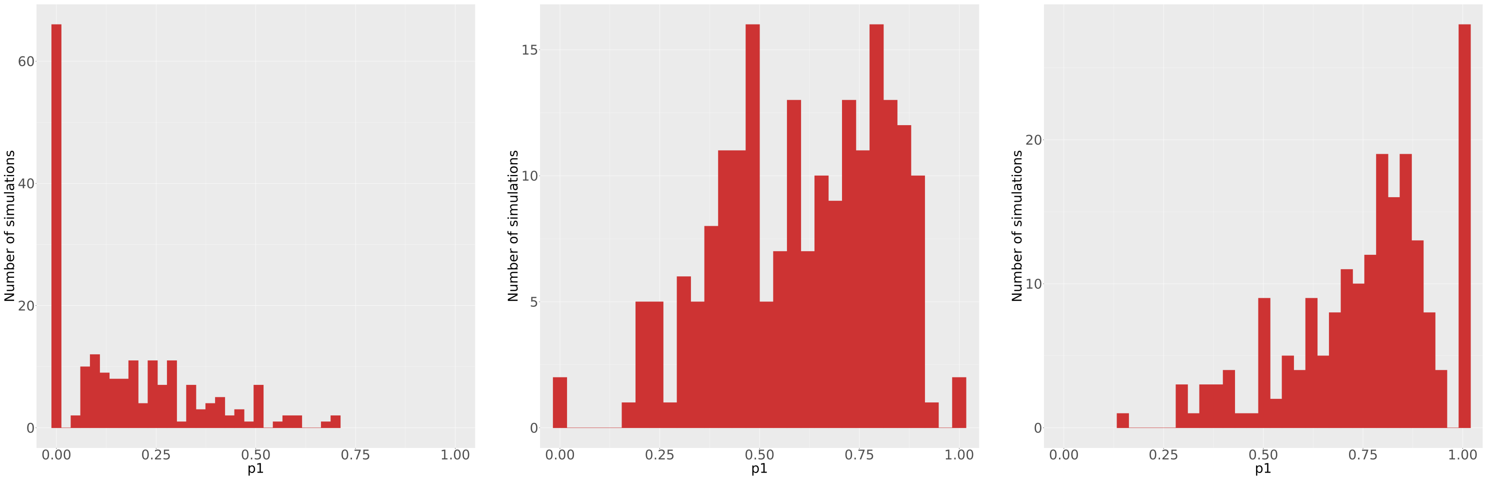


(a)

(b)

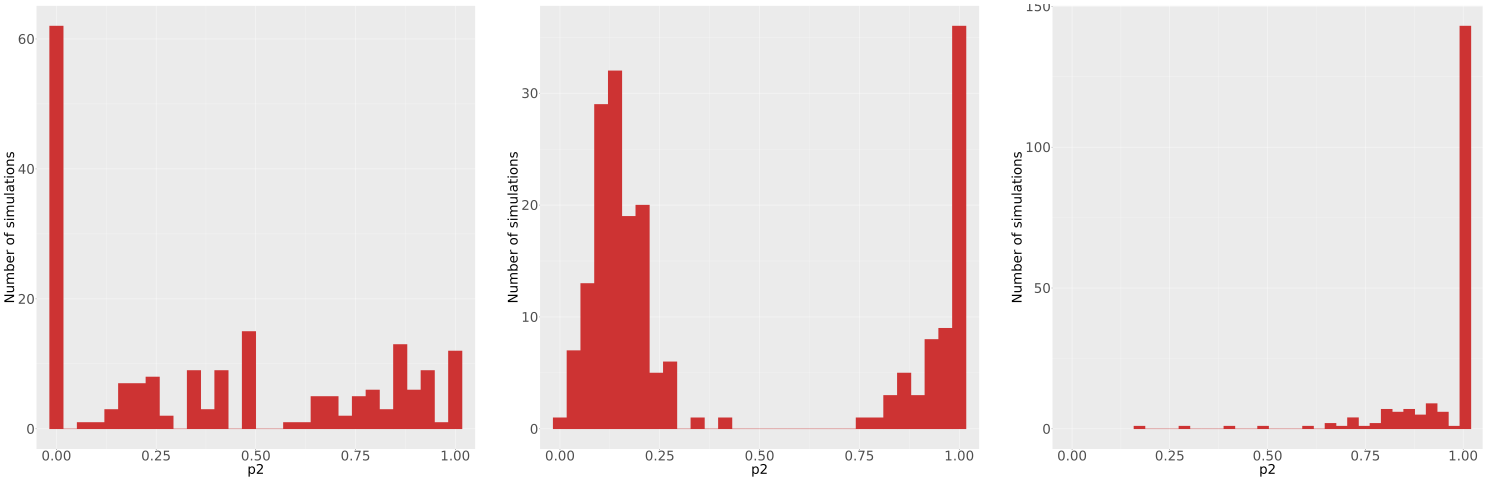
(c)

Figure 3: p_1 histograms filtered by group for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples



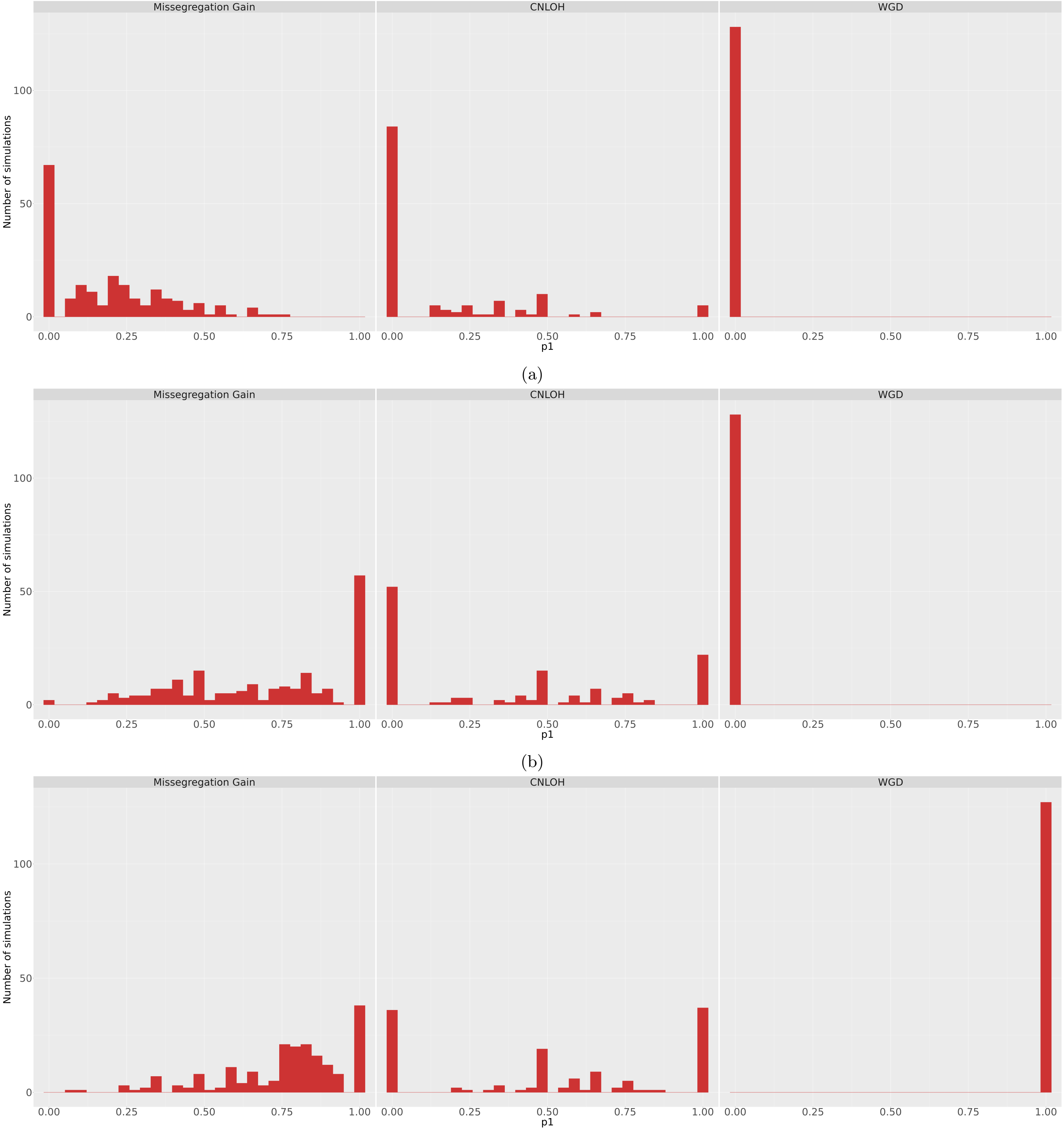
(a) (b) (c)

Figure 4: p_1 histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples



(a) (b) (c)

Figure 5: p_2 histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples

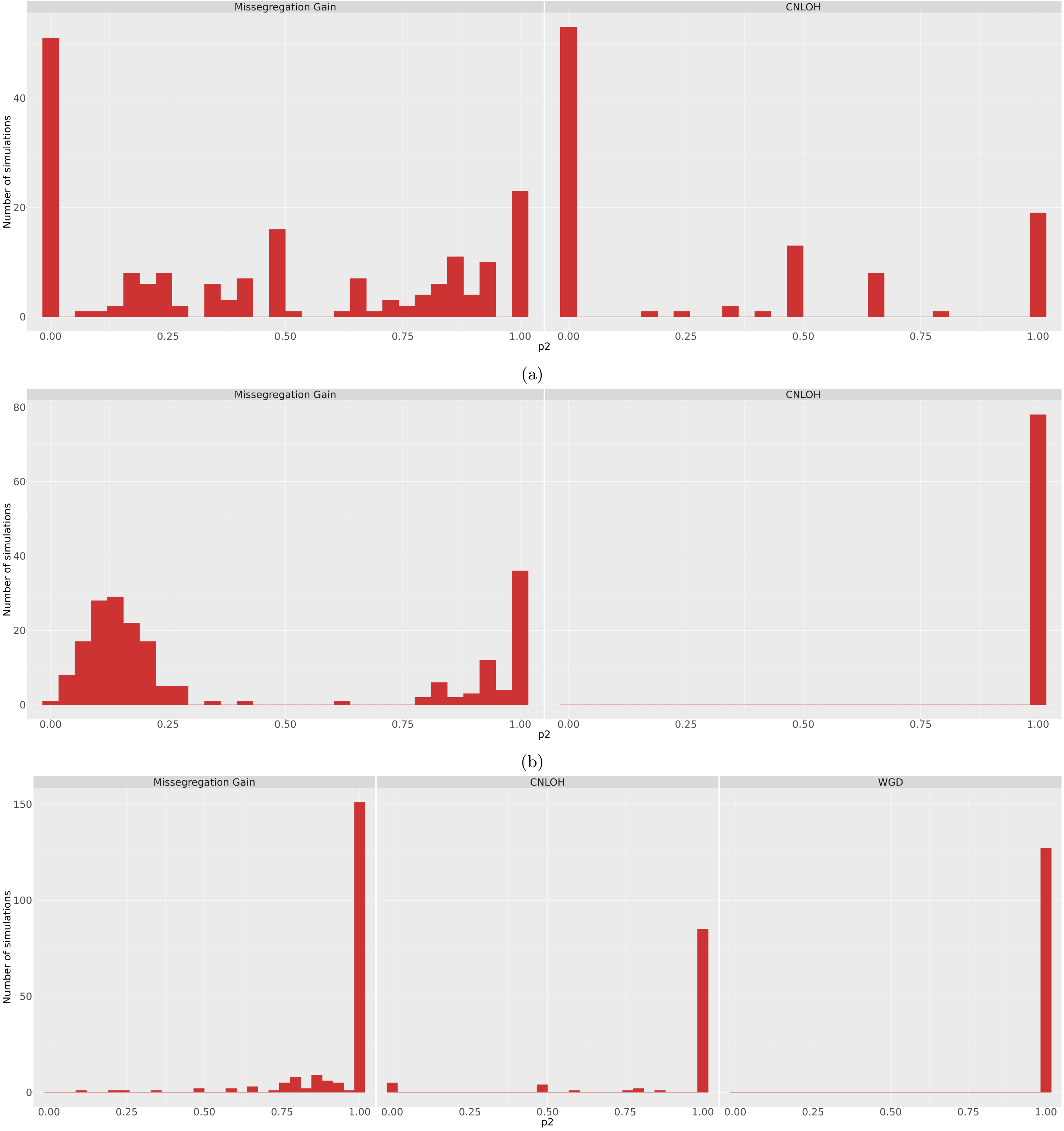


(a)

(b)

(c)

Figure 6: p_1 histograms filtered by type for CancerTiming (a), MutationTimeR (b) and GRITIC(c)on WGD samples

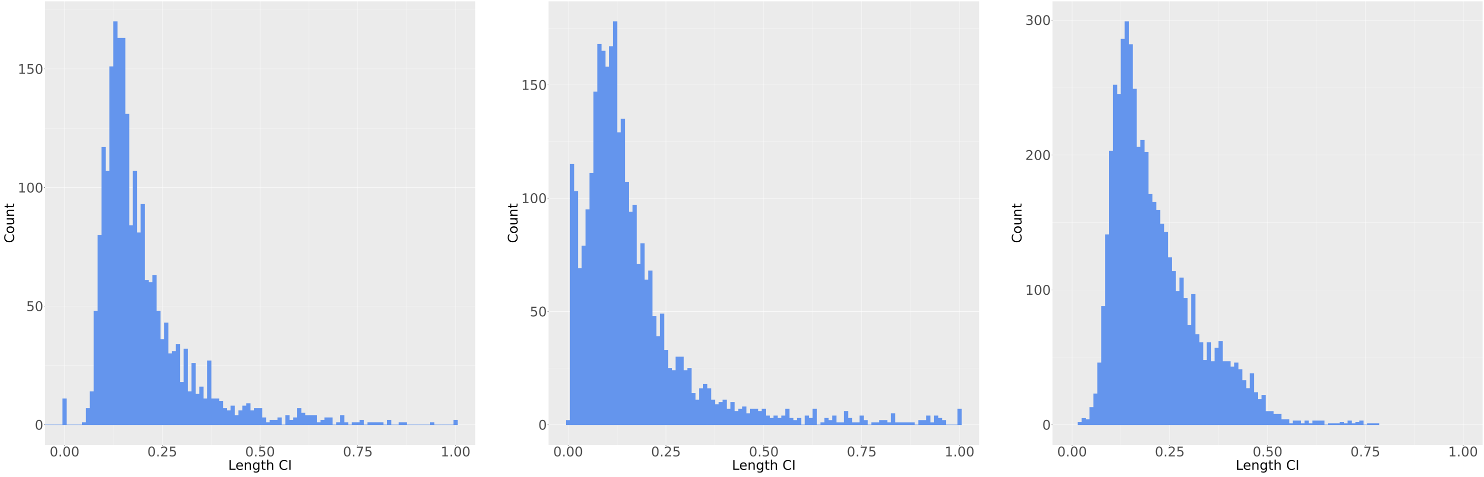


(a)

(b)

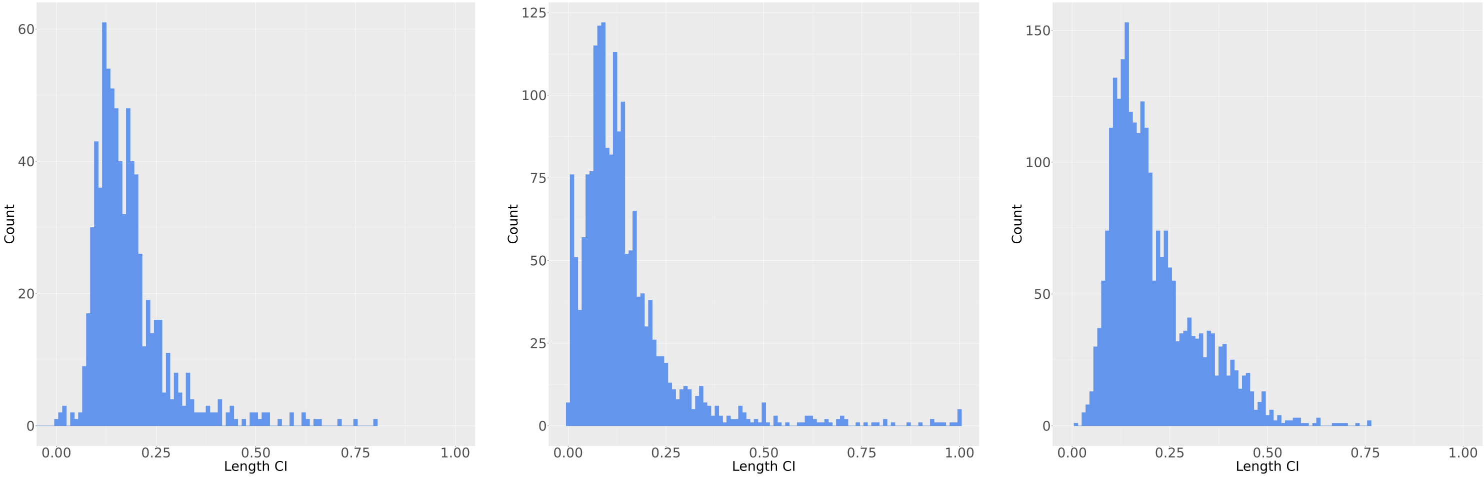
(c)

Figure 7: p_2 histograms filtered by type for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples



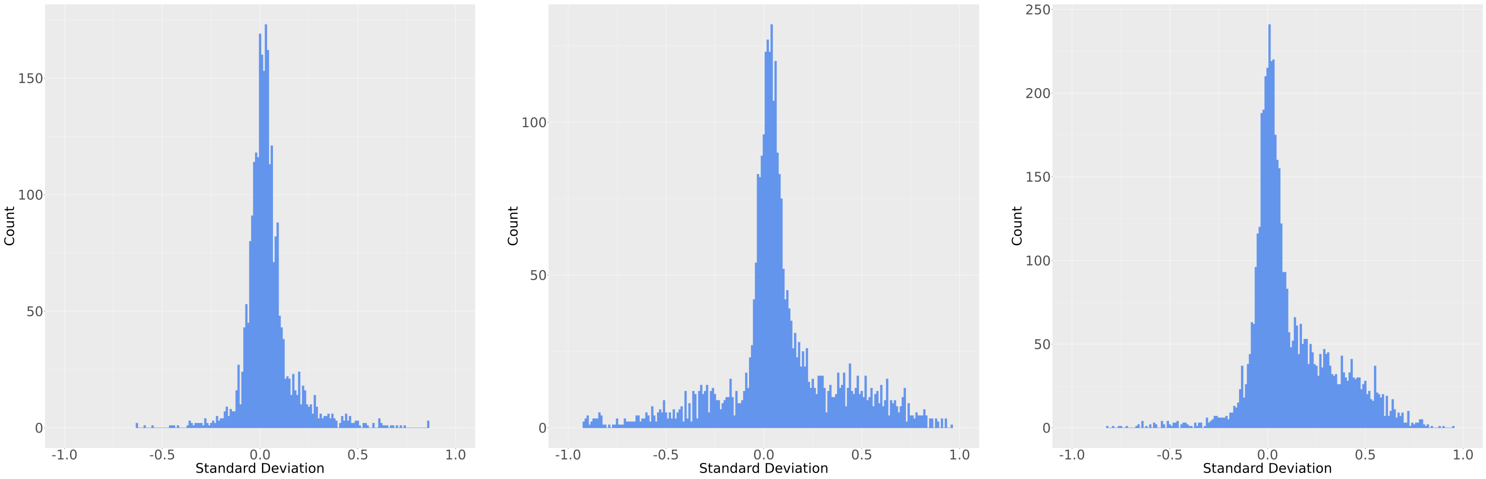
(a) (b) (c)

Figure 8: CI length histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples



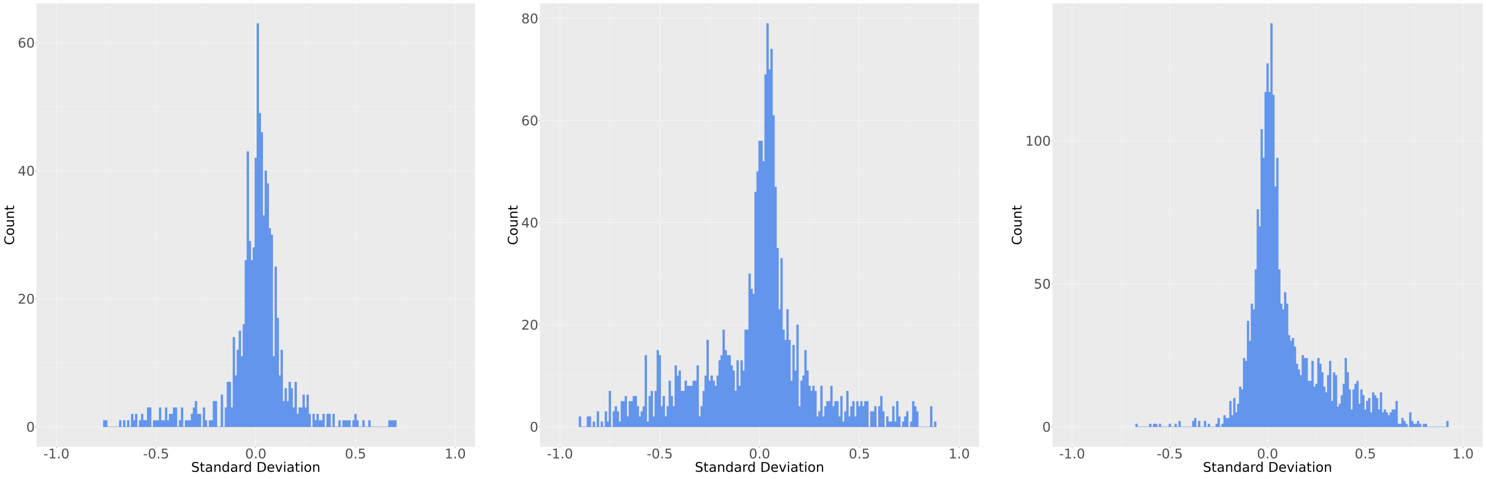
(a) (b) (c)

Figure 9: CI length histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples



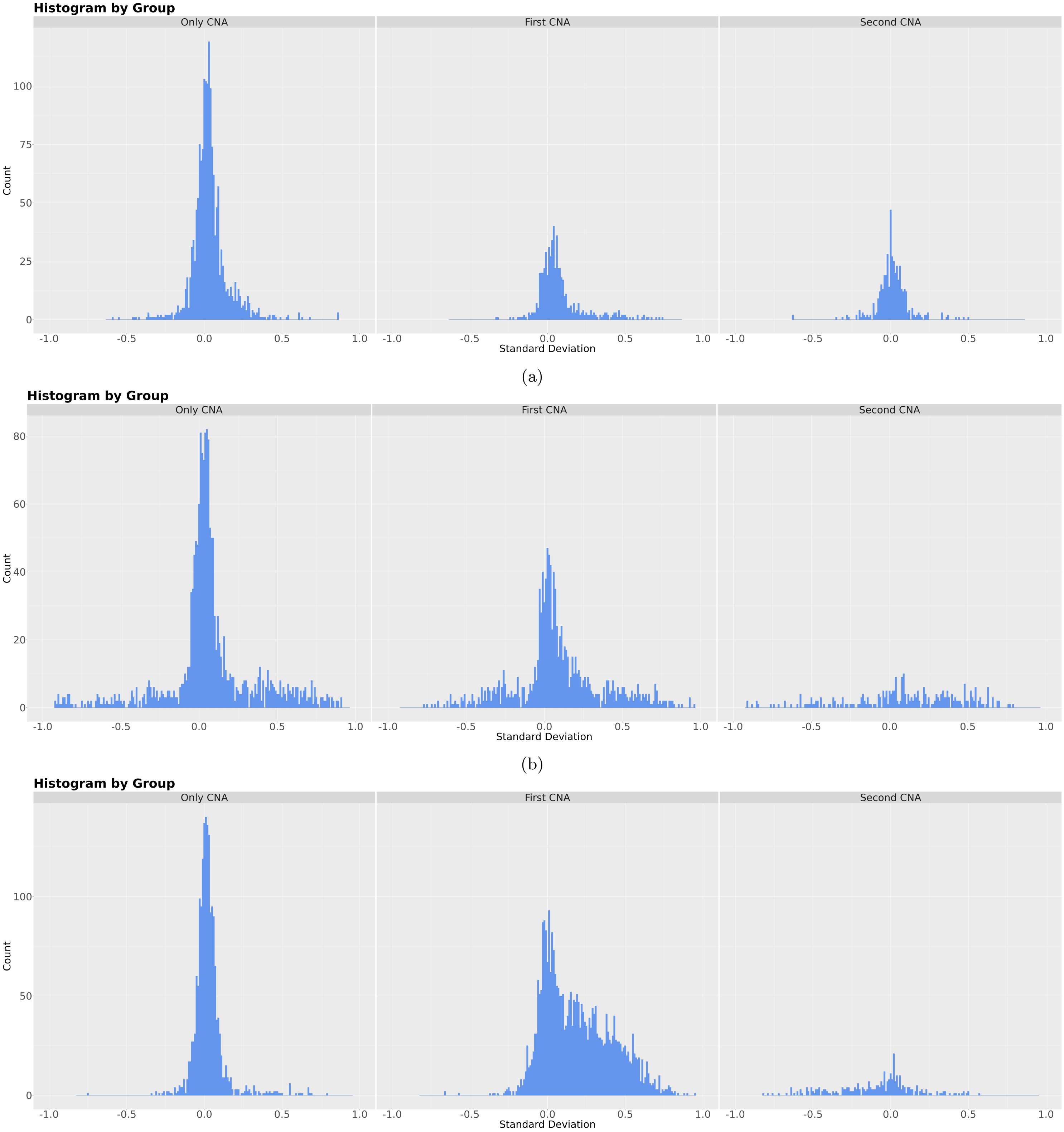
(a) (b) (c)

Figure 10: Deviation histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples



(a) (b) (c)

Figure 11: Deviation histograms for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples

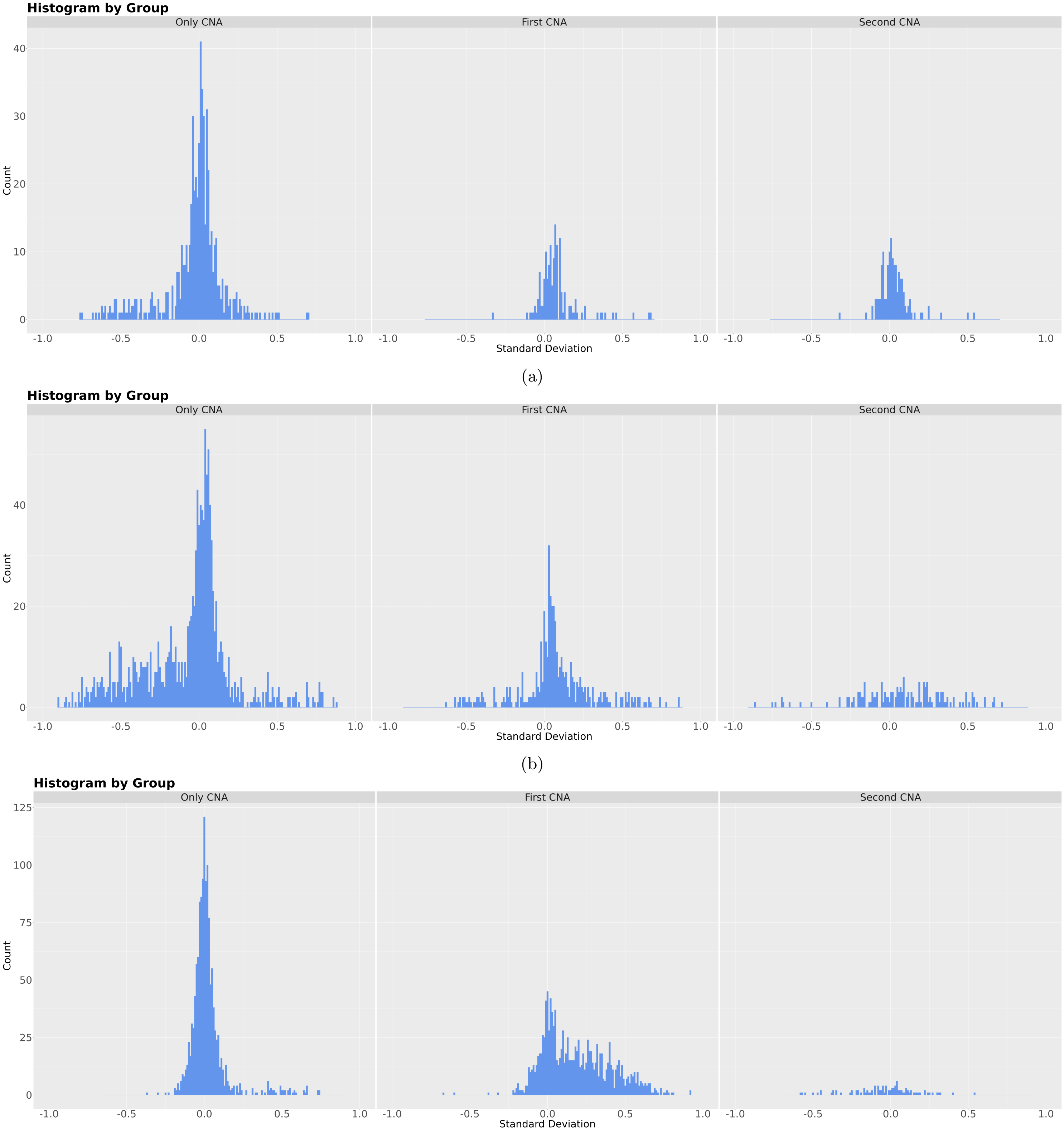


(a)

(b)

(c)

Figure 12: Deviation histograms filtered by group for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on non WGD samples



(a)

(b)

(c)

Figure 13: Deviation histograms filtered by group for CancerTiming (a), MutationTimeR (b) and GRITIC(c) on WGD samples