ENUMERATING SETS OF GENOMIC ALTERATIONS CHARACTERIZING A USER-DEFINED SUBGROUP

Jennifer Wong, Thomas Pichetti, François Radvanyi, Etienne E. Birmelé

ENUMERATING SETS OF GENOMIC ALTERATIONS CHARACTERIZING A USER-DEFINED SUBGROUP

Jennifer Wong
Institut Curie, PSL Research University, CNRS
UMR144, Equipe Labellisée Ligue Contre le Cancer
jennifer.wong@curie.fr

Thomas Pichetti
Laboratoire MAP5
Université Paris Descartes and CNRS, Sorbonne Paris Cité
thomas.picchetti@parisdescartes.fr

François Radvanyi
Institut Curie,PSL Research University, CNRS
UMR144, Equipe Labellisée Ligue Contre le Cancer
Francois.Radvanyi@curie.fr

Etienne Birmelé
Laboratoire MAP5
Université Paris Descartes and CNRS, Sorbonne Paris Cité
etienne.birmele@curie.fr

March 19, 2021

ABSTRACT

Genetic alterations driving cancer are known to be spread over a large number of genes. Deciphering driver alterations from passenger alterations that may however be selected by single gene analysis is a major challenge. Alterations that characterize a given subtype of cancer are of particular interest. However, characterizing alteration sets rather then handling single alterations is a difficult task because of the combinatorial explosion of the number of sets.

We consider a set of gene amplifications, deletions or mutations in tumor samples for which the subtypes of a given cancer are known. We consider that an alteration set characterizes a given subtype with respect to the others if they are frequent in that given subtype and rare for the others. We propose an efficient algorithm that outputs a ranked list of such alteration sets or pathways. The relevance of the output is illustrated using alteration data on bladder cancer.

1 Background

The evolution of cancer is driven by complex and heterogeneous patterns of genetic alterations. The large amount of cancer data provided by international research networks like the International Cancer Genome Consortium (ICGC) or The Cancer Genome Atlas (TCGA) allows to build algorithmic methods to stratify the tumors into subtypes, based on expression or mutational data. Unsupervised classification techniques on those data may however lead to classes that are clinically hard to interpret. The supervised framework, that is finding out which combination of gene features characterizes a given subtype of cancer, is therefore an active field of bioinformatic tools development. In the case of alteration data, that is genetic mutations, losses and amplifications, it moreover corresponds to the biological question

of differentiating driver alterations, which give insight into the tumorigenic mechanisms, from passenger alterations which correspond to co-occurences with driver alterations.

The International Cancer Genome Consortium [1] classifies the methods to identify driver mutations into three main families, and gives an exhaustive review of the methods in each family. The first approach consists on mapping and annotating alterations to compare them to known variants. The second relies on the assessment of the functional impact of the alterations. The third family, to which the present paper belongs to, consists in a statistical selection across a cohort. The underlying idea is that an alteration that appears more often than in a well-chosen null model corresponds to a positive selection during the tumor's development.

The most common way to develop the statistical approach is to develop it genewise by selecting the alterations which frequence deviate significantly from the basal mutation rate. Such approaches became possible with the development of databases as The Cancer Genome Atlas (TCGA) [2], which are large enough to run statistical analysis. The question of the right choice for a null model is however crucial and non trivial [1]. The null model has for example to be adjusted in a patient-specific way to avoid a high number of false positives [3]. The deviation of the mutational rate may not be the only gene-related relevant feature to discriminate between driver and passenger alterations, and can be associated to other criteria by machine learning methods as the CHASM algorithm [4].

Cancer development is however linked to combinations of alterations rather than to single genetic events. It is moreover probably more relevant to consider their impact on the scale of pathways rather than on the gene level [5]. Indeed, a subtype of cancer may be characterized by the key pathways that are altered, a small number of pathways giving raise to a large number of possibly affected gene combinations. A way to overcome this combinatorial issue is to look for alterations at the scale of interaction subnetworks [6, 7]. Another one is to concentrate on gene combinations that are mutually exclusive, the underlying assumption being that a pathway doesn't need to be affected by two driver alterations. Several such methods have been developed [8, 9, 10, 11, 12], some of them also taking into account co-occurrence or functional data [13, 14] or tumor progression [15, 16]. To the best of our knowledge, the most competitive algorithms to address the problem of cancer-subtype specific mutations on quite large datasets based on mutual exclusivity are CoMEt [17] and its weighted extension WExT [18]. The combination of the mutual exclusivity and the pathway scale has been taken into account in [19, 20, 21]. Recent algorithms adapt the mutual exclusivity approach to the cases where subtypes are replaced by continuous outcomes [22, 23].

We introduce a new algorithm called Musette, available as an R-package at https://git.mi.parisdescartes.fr/ebirmele/Musette. It allows to select combinations of alterations that characterize a given cancer subtype from other subtypes. Its main contribution are the following:

1. it does not require that the selected sets are mutually exclusive, though it recovers also mutual exclusive sets. The alteration sets are selected using a score which is flexible and could be modified by the user.
2. the space of all possible sets is explored starting from the empty set and by extending current sets only if this extension brings a significant improvement in terms of score. This procedure avoids both combinatorial explosion and overfitting, and the choice of the significativity level allows the user to choose more or less stringent solutions. The pruning moreover allows to explore genome-wide data without defining an upper-bound on the size of the alteration sets.
3. a preliminary domination step allows to filter alterations that are redundant from a combinatorial or biological point of view, and to restrict the number of returned sets.
4. the interpretation of the results can also be made on the scale of single alterations or of pathways, by scanning the scores of the solutions containing a given alteration or intersecting a given pathway.

We demonstrate the efficiency of Musette on both simulated and real data.

2 Results and discussion

2.1 The Musette algorithm

A data set of genomic alterations, that is mutations, amplifications and deletions, is considered for individuals who are split into two groups, referred to as the *case group* and the *reference group*. The goal of the method is to identify the alterations characterizing the case group. A typical application will be to consider all tumors of a given clinical subtype of tumors as the case group and all other tumors of the same organ as the reference group.

The problem can formally be represented by a bipartite colored graph, which two layers represent respectively the gene alterations and the individuals, and such that edges connect each individual to the set of his altered genes. The individuals corresponding to the case group (resp. the reference group) are moreover colored in red (resp. blue). Sets of

alterations that characterize the case group then ideally correspond to those which cover the red individuals without having blue neighbors (Figure 1).

Enumeration of the sets that are maximal in terms of covering the red samples and minimal in terms of covering the blue ones is NP-hard [24]. We therefore associate, to every set A of alterations, a score $c(A)$ which reflects that high score sets should (a) contain mostly red samples, and (b) contain a large portion of the red group. However, the choice of the score is flexible and the method can also be run with any user-defined score, including the classical $c(A) = -log(p_A)$ where p_A denotes the hypergeometric p-value.

A quality required for an alteration set to be biologically related to the case group is to consist in only a few alterations. Indeed, the case group is biologically more likely related to just a few distinct alterations. Moreover, adding an alteration which concerns a single case individual to some alteration set may increase its score, without truly gaining biological relevance.

Therefore, rather than enumerating alteration sets that optimize the score, we develop a method that constructs relevant sets by adding alterations one by one. A new alteration a is added to the set A only if the gain in score is significant with respect to the distribution of gains obtained when adding random alterations following the observed degree distribution. The p-value associated to the latter test is called the *step-score* between the sets A and $A \cup \{a\}$.

This method allows to mimic Dijkstra's algorithm in order to construct a tree linking all sets which can be constructed with step-scores lower than a given threshold α. The returned solutions are then the nodes, and especially the leaves of that tree.

We moreover highlight that this Dijkstra-like approach is from a theoretical point of view a systematic way to visit the whole sets of alterations for a given significativity threshold. It avoids the use of any random exploration of the alteration set space, as for instance MCMC algorithms [17, 18].

It is worth noting that some pre- and post-treatments are applied to the algorithm for biological reasons. Indeed, an amplification or deletion may concern several neighboring genes, and treating each of them separately may produce a high number of solutions which correspond to the same biological interpretation. A notion of domination between alterations is therefore introduced to group neighboring genes which are altered in similar sets of individuals.

Moreover, as pathways are a more relevant biological entity to interpret alterations, genes belonging to a same pathway can be grouped in each solution during a post-treatment phase.

The solution dataframe contains, for each selected alteration set, its score and the maximal step-score used to build it. It however also contains the specificity/sensitivity of the samples covered by the set and a variable *leaf* telling if it is possible to improve significantly the set by adding a new alteration. However, as the number of proposed solutions may be quite large, some visual tools are also provided for the interpretation of the results.

The first one is the drawing of an alteration-set based oncoplot allowing a vizualisation of the quality of each set in terms of its covering of the red vertices and non-covering of the blue ones.

The second one assigns to each alteration an *influence index* which is proportional to the sum of the scores of the selected solutions containing that alteration. A similar index can be computed for pairs of alterations by considering the solutions containing both alterations. An *influence graph* can then be drawn with node and edge thickness proportional to the influence of respectively single and pairs of alterations. This graph does not contain all the information returned by the method but allows to see the major alterations in a glance.

2.2 Comparison with mutual exclusion driven approaches on simulated data

We first run 500 simulations according to the simulation scheme described in Section 4.8. 50 red and 50 blue samples are considered, as well as 100 alterations among which two groups of three alterations, T and U, are those really characterizing red samples. We set $q_T = q = 0.023$ and $q_U = 0.3$, so that T is a highly mutually exclusive set whereas U is a moderately mutually exclusive set.

All experiments were run on a one-core laptop. The CoMEt and WeXT methods correspond to the code available on the GitHub repository run with the respective RE and WRE arguments, each with 10 Monte-Carlo Markov Chains.

Figure 2 shows the ranks of T and U among the 10 sets of solutions of highest score. T and U are found almost every time by *musette*, and most of the time in the two first places. On the contrary, there are almost never ranked in the top solutions by CoMEt and WeXT.

To highlight the fact that the right solutions are pointed by our method, we computed the influence indices for a single simulation as well as the mean influences indices for a set of ten simulations. Figure 3 confirms that the six relevant

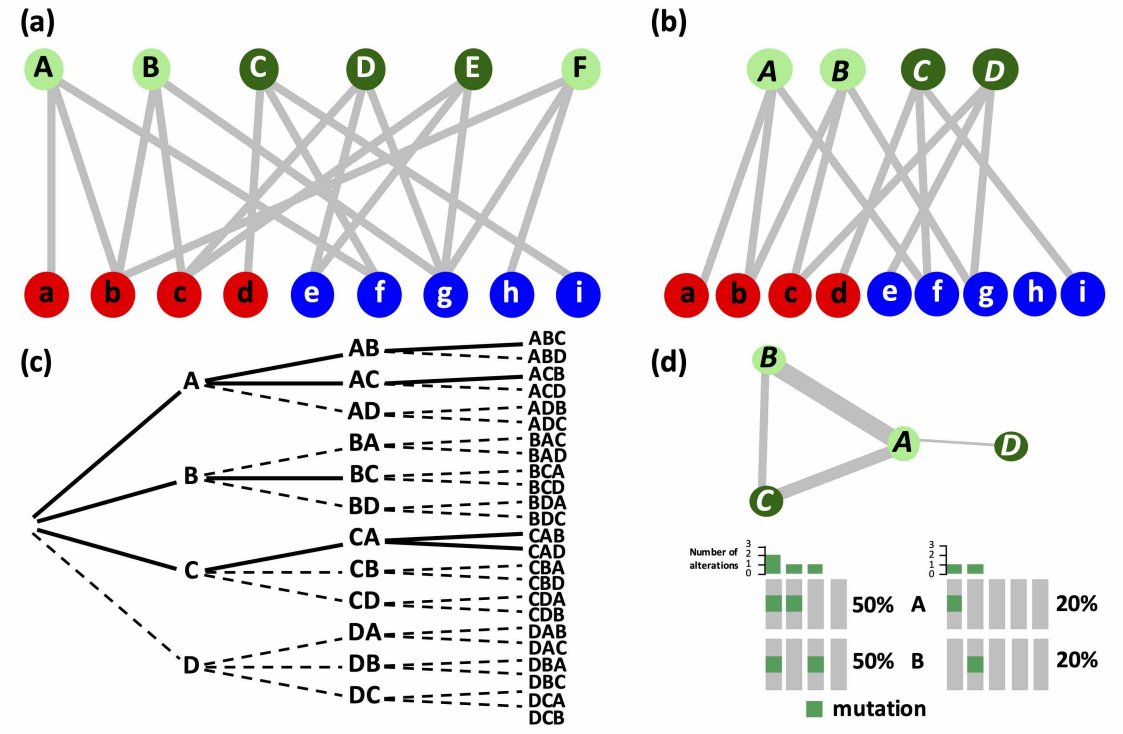

Figure 1: Snapshot of the method. (a) The original data is a bipartite graph linking alterations (the upper layer) to samples they appear in (the bottom layer). The aim of the method is to enumerate sets of alterations that characterize a given sub-sample colored in red. (b) Some alterations are filtered because they are dominated by another alteration. Here for instance, choosing A instead of E or B instead of F will always give more relevant sets. (c) The space of all possible sets is considered as a rooted tree from which only a subtree is explored. At each step of the algorithm, only the most significant edges in terms of score gain are added to the current tree. (d) The result of the algorithm is a list of alteration sets. However, to give a more graphical view of the result, a graph can be drawn in which the size of the node (resp. an edge) is proportional to the total score of the solutions it belongs to.

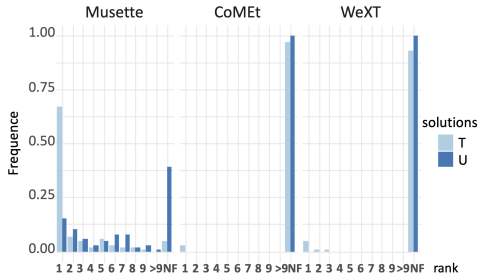

Figure 2: Distribution of the ranks of the T and U sets among all solutions for *musette*, $WeXT$ and $CoMEt$. 100 simulations were run on 100 alterations, with 50 blue and 50 red sets.

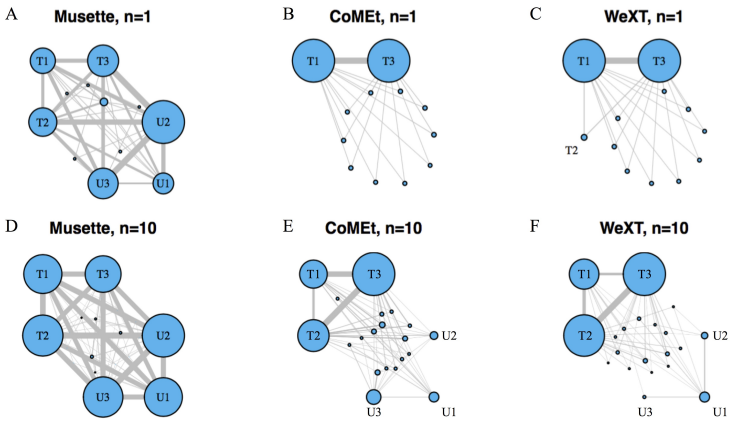

Figure 3: Influence graphs for Musette and Comet on simulated data with 100 alterations. The size of the nodes are proportional to the sum of the scores of the alterations in the 20 top solutions, the thickness of the edges is proportional to the sum of the scores of the solutions containing both alterations. In the second row, means of those quantities over 10 simulations are considered.

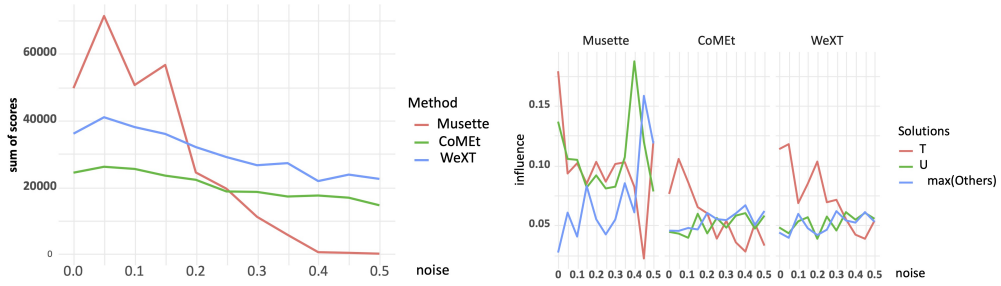

Figure 4: Left: sums of the scores of the twenty best ranked solutions for Musette, WeXT and Comet and increasing noise in the simulated data. Right: normalized influence of alterations under Musette, WeXT and Comet on simulated data. The figure shows the mean influences of the elements of T and U , as well as the maximum influence among the other alterations.

alterations are those that are mainly present in the top solutions. Moreover, it shows that this preminence is higher for Musette compared to the two other methods, especially for the less mutually exclusive solution U.

The influence index was then computed, for both Musette and Comet, for increasing noise, that is a fraction from 0 to 50% misclassified samples. Figure 4 and 5 shows this evolution, the results being averaged on 10 runs for each noise parameter. The sum of influence indices before their normalization step decreases with noise, which is not surprising as the scores of a relevant solution decreases when the noise increases. Once normalized, the alterations of the T and U sets are the only ones having a high proportion of influence and their proportion are equivalent and stable up to a noise of 30%. It becomes unstable for a higher noise, which is not surprising as the sum of influences is close to 0 for higher rates of noise.

From a method comparison point of view, this graph confirms the results obtained in Figure 3 for a noise-free dataset, that is that both methods point mainly to the relevant alterations, but that Musette highlights them stronger and even if they are not mutually exclusive.

Finally, we run the experiments for varying sample sizes, averaging the running time on three runs for each experiment. Figure 5 shows that Musette significantly outperforms CoMEt and WeXT, which need time-consuming MCMC runs.

2.3 Application to bladder cancer data

In this section, we applied *Musette* on bladder cancer dataset. We tested *Musette* on three bladder cancer subgroups: a well-known subgroup *i.e.* luminal subgroup and two others subgroups *i.e.* bladder cancer tumors from carcinoma *in situ* pathway and bladder cancer tumors non-mutated for RAS family genes. The results of theses subgroups were shown

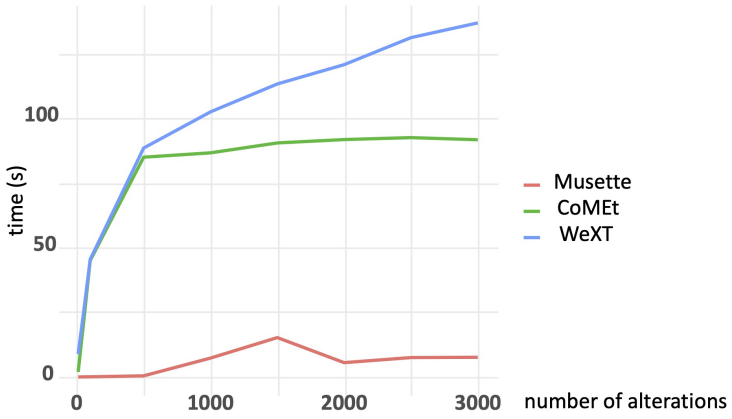

Figure 5: Running time comparison of *musette*, *WeXT* and *Comet*

respectively in Figure 6 to Figure 8. For each subgroup, we drawn the oncoplot of the best solution found by *Musette* and the influence graph of the 50 first best solutions.

For the luminal subgroup, the best solution of *Musette* detected tumors with frequent mutations of KDM6A and BRAF but also frequent mutations and amplifications of FGFR3 (Figure 6a). These genetic alterations are specific of this subgroup. The influence graph allowed the visualization of more altered genes specific of our subgroup, *e.g.* mutation of ELF3. Among all solutions found by musette, the genes ELF3, KDM6A, FGFR3 were frequently detected within the same solutions (Figure 6b). For the subgroup of bladder cancer tumors from carcinoma *in situ*, RB1 was frequently mutated (Figure 7a). This mutation was frequently associated with amplification of CDKAL1 and E2F3 (Figure 7b). Those genes are important in the regulation of cell cycle. Finally, the subgroup of bladder cancer tumors non-mutated for RAS family genes had frequent mutations of FGFR3 and TSC1 (Figure 8a). The mutations of FGFR3 and RAS family genes seemed to be mutually exclusive, almost no tumor was mutated for FGFR3 in the reference group (Figure 8a). FGFR3 mutation was associated to amplification of CCND1. This gene participates as well as RB1 in the regulation of the cell cycle. (Figure 8b).

3 Conclusion

We introduce the *Musette* algorithm for identifying sets of cancer-type specific alterations. Its originality is the method used to cover the space of candidate sets. The growth of a tree-structure allows to explore only sets of solutions that are relevant with respect to their subsets. The combinatorial explosion of the number of explored sets is then avoided without adding an additional constraint like mutual exclusion to the initial problem.

The input data being a boolean matrix, the method could be applied to any types of alterations in pan-cancer studies or to any problem for set of variables characterizing given subtypes of individuals have to be selected.

We applied *Musette* to the TCGA bladder cancer data and we found results consistent with the literature.

sectionMaterial and methods

3.1 Score of an alteration set

Let R and N denote respectively the number of red individuals (the case group) and the total number of individuals.

For any set A of alterations, we denote by $N_R(A)$, $N_B(A)$ and $N_T(A)$ the set of red neighbours, blue neighbours and total neighbors of A and by $d_R(A)$, $d_B(A)$ and $d_T(A)$ their respective sizes.

The proportion of red samples in $N_T(A)$ is accounted for by an hypergeometric p-value $p_h(A)$: it corresponds to the probability to increase or have the same number of red neighbors, if we were to redistribute the red and blue labels randomly, preserving their total number (or alternatively, the probability of getting at least $d_R(A)$ red neighbours when drawing $d_T(A)$ neighbours at random)

$$p_h(A) = \sum_{k \geq d_R(A)} \frac{\binom{R}{k}\binom{N-R}{d_T(A)-k}}{\binom{N}{d_T(A)}}$$

This p-value is a standard tool to detect over-represented classes, but does not take into account the fact that a solution should contain a large proportion of the red group. To do so, we multiply the opposite of its logarithm by the proportion of red neighbors that are covered to obtain our score of interest.

$$c(A) = -\frac{d_R(A)}{N_R} \log(p_h(A))$$

3.2 The step-score

As we are looking for alterations that characterize a cancer subtype, a small number of alterations is desirable. This quality is detrimental to the previously defined score, judging by the distribution of the scores of the 100 best alterations sets of size k, for different values of k : not a single solution of size k can compete with the 100 best solutions of size $k+1$ (see supplementary material). An explanation for this observation is that no matter how good a solution of size k is, there very likely exists an alteration which, if added to it, will increase its score, however slightly.

An alteration set is therefore considered as a sequence of single alterations to be added one by one, starting from the empty set. This approach allows to get rid of alterations which effect on the score is too small to be valued over the resulting loss of compactness.

Indeed, to assess the significance of the score increase obtained by adding an alteration a to an existing set A, we compare it to the distribution obtained under the following random model.

(a) The number X_T of neighbors of the alteration is drawn according to the empirical distribution of the alterations degrees;

(b) Given X_T, the neighbors of the alteration are uniformly chosen among the samples.

Using computations on the hypergeometric distribution, detailed in the supplementary material, we can compute the p-value associated to the gain in score, called step-score.

$$s(A, a) = \mathbf{P}\big(c(A \cup \{b\}) \geq c(A \cup \{a\})\big)$$

where b is a random alteration drawn according to the latter model.

3.3 A Dijkstra-like algorithm

By setting a threshold α on the step-score, a sequence of alterations in a given order passes the threshold if, starting from the empty set and adding these alterations one after the other, every step has a step-score smaller than α.

The full set of alteration sequences naturally forms a tree where the root is the empty sequence, and a sequence's children are obtained by appending a gene to it. It is easy to see that if a sequence passes a given threshold, all of its ancestors do as well. Thus, sequences that pass a given threshold form a subtree of this exhaustive tree.

This subtree is constructed with an algorithm resembling Dijkstra's shortest paths algorithm, where the node-to-node distance is replaced by the step-score from a sequence to its child, and the distance of a node to the origin is replaced by the maximum of the step-scores along the path.

The algorithm makes use of a priority queue where nodes (i.e. alteration sequences) wait to be added to the subtree, sorted by their step-score. Initially the subtree is empty, the threshold is set to zero and the waiting queue only contains the empty sequence. The new threshold is then set to be the step-score of the first element in the queue. While the step-score of the first element in the queue does not exceed this new threshold, this element is popped, added to the subtree, and its children are added into the queue, after computing their step-score. Note that some of those children may have a step-score strictly smaller than the new threshold, in which case they will be popped before this whole growth step ends. This operation is repeated until the tree contains a pre-defined size or until the threshold reaches to a pre-defined value.

3.4 Alternative step-score definitions

The solutions built by the former algorithm depend on the order of the alterations in the sequence : the same set of alterations is represented by a number of different sequences, some of which may pass the threshold and some may not. Since our original problem deals with sets of alterations and not ordered sequences, two alternative ways are possible in the method.

1. *The strict approach:* an alteration set is to be avoided if its score is not significantly better than it would be without one of its elements. This is achieved by modifying the function that computes the step-score for a solution to be inserted into the waiting queue : instead of focusing on adding the last element of the sequence to the rest of it, we do the same with all the other elements. This yields as many step-scores as there are elements in the sequence, and the worst one is chosen as the final result. We also check if the same set is already present in the waiting queue, and in this case return the worst possible step-score, to avoid inserting a duplicate.
2. *The best-first approach:* the additional tasks in the strict approach are computationally expensive, hence we also consider an heuristic approach to replace the above process with a less accurate, but faster one. It involves ranking all the alterations based on their hypergeometric scores. Then, we only check sequences of alterations in which alterations respect this order. This handily removes the need to check for duplicates, and the assumption is that the last alteration added, being the worst one in some sense, would be the one to yield the worst step-score in the above procedure.

The "original" approach, the "strict" one, and the "best first" heuristic represent three different trade-offs between computational cost and accuracy, so we implemented them all for the user to choose which one best suits their needs.

3.5 Score p-value

Once top solutions with highest scores have been determined, a natural question is to test their significance. A way to do so is to compute an empirical p-value by running the algorithm on a large number of replicas of the data for which the red and blue labels have been shuffled randomly. Indeed, in those replicas, the red/blue assignation has lost its biological signification and the obtained scores correspond to a sample under a null model with fixed graph structure.

The proportion of those scores higher than the score of interest gives a p-valuer for its significance.

3.6 Domination between alterations

Due to the large number of alterations with respect to the number of samples, several alterations may affect the same individuals. Moreover, amplifications and deletions often affect not only one gene, but an entire chromosome section. This means that if the amplification of some gene is biologically linked to our sample groups, amplification of the genes that lie close to it on its chromosome will also appear as significant and flood the results with this misleading information.

This situation induces in practice a high number of redundant solutions. Three notions of equivalence or domination are therefore introduced and applied as a pre-treatment on the data in order to reduce the number of alterations taken into account.

1. *Equivalence:* Alterations a and b are equivalent if they share the same neighborhood, and are therefore identical from the algorithm's point of view. They are merged into a single node as each solution with one of the alterations would induce an identical solution with the other one.
2. *Domination:* Alteration a dominates alteration b if they are not equivalent but $N_R(b) \subset N_R(a)$ and $N_B(a) \subset N_B(b)$.

   This situation implies that a is always a better choice than b and that adding b to a set already containing a will lead to no gain in score. b is then stored in the set of alterations dominated by a and removed from the alteration set.
3. *Local domination:* If the two alterations a and b affect genes that are close enough on the genome, that is at distance lower than a given threshold δ, there is more evidence for a domination relation, so that the criteria are relaxed.

   a is then considered as dominating b if a proportion β of the red neighbors of b are neighbors of a and a proportion γ of the blue neighbors of a are neighbors of b.

Setting $\beta = \gamma = 1$ is equivalent to domination. Setting them to a smaller value allows to replace two alterations by a single one if they are similar enough and close on the genome.

Note that different values of β and γ can be chosen for different types of alterations. A choice $\beta = \gamma = 1$ may for example be chosen for mutations which are position-specific, whereas lower values may be more relevant for deletions and amplifications which occur on genomic segments.

3.7 Pathways

Finally, once the results are obtained, biological pathway information can be exploited by grouping together genes that belong to the same pathway, thus decreasing the size of some solutions and increasing their interpretability.

3.8 Choice of the parameters

To run our algorithm, several parameters have to be defined, namely the step-score threshold α and the proportions β and γ in the local domination definition. Our advice is to try several values to obtain a good trade-off concerning the number of solutions and the running time.

Concerning α, one may for example try first a higher value of α and plot the number of solutions $n(\alpha')$ such that the maximum step-score on the path is lower than α', $0 \leq \alpha' \leq \alpha$. The resulting curve often shows an elbow corresponding to the point were the number of selected solutions explodes. The number of solutions at this point may however already be huge and the user might prefer choose a smaller threshold to examine less solutions that are more significant.

The choice of β, γ and δ is left to the user, who can thus define what the similarity in terms of mutated samples and distance along the genome should be to consider two alterations as equivalent. Relaxing the criteria will lower the number of solutions.

3.9 Simulations

To illustrate the performances of our method and compare it to the Comet algorithm, we run both algorithm on simulated datasets.

Those datasets are built with 100 alterations and N samples. The alteration set contains in particular two subsets of size three, denoted $T = \{t_1, t_2, t_3\}$ and $U = \{u_1, u_2, u_3\}$.

Denote by q the ground rate of alterations, which numerical value is set to 0.023 based on the TCGA Bladder Cancer data used for the evaluation on real data. For each sample, the alterations not belonging to T or U are activated independently with probability q. The alteration in U and T are activated in a way such that, for $i \neq j$, $\mathbf{P}(t_i \text{ is altered } | t_j \text{ is altered }) = q_T$ and $\mathbf{P}(u_i \text{ is altered } | u_j \text{ is altered }) = q_U$. Varying q_T and q_U between 0 and 1 allows to generate sets that are more or less mutually exclusive.

Samples are then drawn such that half of them, called the red samples, have at least one alteration in T and U, and half of them have no alteration for at least one set. The aim is to mimic a case where a subtype of cancer is characterized by a mutation in the set T AND a mutation in the set U. The method will therefore perform well if T and U appear in the top-list of selected solutions.

The data is finally made noisy by miscoloring each sample with a probability p_{noise}.

The algorithm is evaluated and compared to Comet on several criterias.

The first one is to determine if the exact sets T ans U are found as top-scored leafs, or as any top-scored node of the explored tree. We call top-scored the 20 first elements of the list. A leaf is a set that has no child in the tree, that is which cannot be improved significantly by adding another alteration.

The second evaluation is the *influence index* of the alterations in T and U, which is proportional to the sum of the scores of the top-scored solutions containing the alteration of interest.

We draw the *influence graphs* which node sizes are proportional to the influence index and edge thickness to the sum of the scores of the solutions containing both endvertices. Ideally, this graph should consist of two triangles formed by the T and the U set. We also study the evolution of the influence indices of the elements of T and U in terms of the noise parameter p_{noise}.

Finally, we compare the running time for both methods in function of N.

3.10 Bladder cancer dataset

The dataset consists of n = 400 bladder tumors. Mutation data were obtained by exome sequencing and the CNV data have been obtained with Affymetrix Genome wide SNP 6.0 arrays. All exome-sequencing and CNV data were downloaded from the TCGA open-access HTTP directory (https://portal.gdc.cancer.gov/projects/ TCGA-BLCA) and are level 3 data. In our analysis, for CNV data, we only considered genes with homozygous deletion and gene with an amplification of at least two copies.

References

[1] Gonzalez-Perez, A., Pathways, I.C.G.C.M., of the Bioinformatics Analyses Working Group, C.S., *et al.*: Computational approaches to identify functional genetic variants in cancer genomes. Nature methods **10**(8), 723–729 (2013)

[2] McLendon, R., Friedman, A., Bigner, D., Van Meir, E.G., Brat, D.J., Mastrogianakis, G.M., Olson, J.J., Mikkelsen, T., Lehman, N., Aldape, K., *et al.*: Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature **455**(7216), 1061–1068 (2008)

[3] Lawrence, M.S., Stojanov, P., Polak, P., Kryukov, G.V., Cibulskis, K., Sivachenko, A., Carter, S.L., Stewart, C., Mermel, C.H., Roberts, S.A., *et al.*: Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature **499**(7457), 214–218 (2013)

[4] Carter, H., Chen, S., Isik, L., Tyekucheva, S., Velculescu, V.E., Kinzler, K.W., Vogelstein, B., Karchin, R.: Cancer-specific high-throughput annotation of somatic mutations: computational prediction of driver missense mutations. Cancer research **69**(16), 6660–6667 (2009)

[5] Vogelstein, B., Kinzler, K.W.: Cancer genes and the pathways they control. Nature medicine **10**(8), 789–799 (2004)

[6] Vandin, F., Upfal, E., Raphael, B.J.: Algorithms for detecting significantly mutated pathways in cancer. Journal of Computational Biology **18**(3), 507–522 (2011)

[7] Hofree, M., Shen, J.P., Carter, H., Gross, A., Ideker, T.: Network-based stratification of tumor mutations. Nature methods **10**(11), 1108–1115 (2013)

[8] Leiserson, M.D., Blokh, D., Sharan, R., Raphael, B.J.: Simultaneous identification of multiple driver pathways in cancer. PLoS Comput Biol **9**(5), 1003054 (2013)

[9] Szczurek, E., Beerenwinkel, N.: Modeling mutual exclusivity of cancer mutations. PLoS Comput Biol **10**(3), 1003503 (2014)

[10] Constantinescu, S., Szczurek, E., Mohammadi, P., Rahnenführer, J., Beerenwinkel, N.: Timex: a waiting time model for mutually exclusive cancer alterations. Bioinformatics **32**(7), 968–975 (2016)

[11] Kim, Y.-A., Madan, S., Przytycka, T.M.: Wesme: uncovering mutual exclusivity of cancer drivers and beyond. Bioinformatics **33**(6), 814–821 (2017)

[12] Deng, Y., Luo, S., Deng, C., Luo, T., Yin, W., Zhang, H., Zhang, Y., Zhang, X., Lan, Y., Ping, Y., *et al.*: Identifying mutual exclusivity across cancer genomes: computational approaches to discover genetic interaction and reveal tumor vulnerability. Briefings in Bioinformatics **20**(1), 254–266 (2019)

[13] Dao, P., Kim, Y.-A., Wojtowicz, D., Madan, S., Sharan, R., Przytycka, T.M.: Bewith: A between-within method to discover relationships between cancer modules via integrated analysis of mutual exclusivity, co-occurrence and functional interactions. PLoS computational biology **13**(10), 1005695 (2017)

[14] Mina, M., Raynaud, F., Tavernari, D., Battistello, E., Sungalee, S., Saghafinia, S., Laessle, T., Sanchez-Vega, F., Schultz, N., Oricchio, E., *et al.*: Conditional selection of genomic alterations dictates cancer evolution and oncogenic dependencies. Cancer cell **32**(2), 155–168 (2017)

[15] Cristea, S., Kuipers, J., Beerenwinkel, N.: pathtimex: joint inference of mutually exclusive cancer pathways and their progression dynamics. Journal of Computational Biology **24**(6), 603–615 (2017)

[16] Raphael, B.J., Vandin, F.: Simultaneous inference of cancer pathways and tumor progression from cross-sectional mutation data. Journal of Computational Biology **22**(6), 510–527 (2015)

[17] Leiserson, M.D., Wu, H.-T., Vandin, F., Raphael, B.J.: Comet: a statistical approach to identify combinations of mutually exclusive alterations in cancer. Genome Biology **16**(1), 1–20 (2015). doi:10.1186/s13059-015-0700-7

[18] Leiserson, M.D., Reyna, M.A., Raphael, B.J.: A weighted exact test for mutually exclusive mutations in cancer. Bioinformatics **32**(17), 736–745 (2016). doi:10.1093/bioinformatics/btw462. http://oup.prod.sis.lan/bioinformatics/article-pdf/32/17/i736/24151390/btw462.pdf

[19] Babur, Ö., Gönen, M., Aksoy, B.A., Schultz, N., Ciriello, G., Sander, C., Demir, E.: Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations. Genome Biology **16**(1), 1–10 (2015). doi:10.1186/s13059-015-0612-6

[20] Lu, S., Lu, K.N., Cheng, S.-Y., Hu, B., Ma, X., Nystrom, N., Lu, X.: Identifying driver genomic alterations in cancers by searching minimum-weight, mutually exclusive sets. PLoS computational biology **11**(8), 1004257 (2015)

[21] Kim, Y.-A., Cho, D.-Y., Dao, P., Przytycka, T.M.: MEMCover: integrated analysis of mutual exclusivity and functional network reveals dysregulated pathways across multiple cancer types. Bioinformatics **31**(12), 284–292 (2015). doi:10.1093/bioinformatics/btv247. http://oup.prod.sis.lan/bioinformatics/article-pdf/31/12/i284/17101817/btv247.pdf

[22] Kim, J.W., Botvinnik, O.B., Abudayyeh, O., Birger, C., Rosenbluh, J., Shrestha, Y., Abazeed, M.E., Hammerman, P.S., DiCara, D., Konieczkowski, D.J., *et al.*: Characterizing genomic alterations in cancer by complementary functional associations. Nature biotechnology **34**(5), 539–546 (2016)

[23] Basso, R.S., Hochbaum, D.S., Vandin, F.: Efficient algorithms to discover alterations with complementary functional association in cancer. PLoS computational biology **15**(5), 1006802 (2019)

[24] Picchetti, T., Chiquet, J., Elati, M., Neuvial, P., Nicolle, R., Birmelé, E.: A model for gene deregulation detection using expression data. BMC Systems Biology **9**(6), 1–8 (2015). doi:10.1186/1752-0509-9-S6-S6