Musette: Supplementary Material

Jennifer Wong, Thomas Picchetti, François Radvanyi and Etienne Birmele

1 Score distribution, solution size and threshold choice

The aim of that paragraph is to illustrate that a threshold choice for the step-score is needed in order to avoid an explosion of the solution size and to discuss that choice.

To do so, the Musette algorithm was run on the Bladder Cancer data by looking for the 20000 best solutions, the red/blue samples beeing the basal/luminal ones. This would correspond to a threshold choice of $2.5e^{-4}$ for the step-score, which seems a priori to be a quite stringent threshold.

1.1 Overfitting

Figure 1.1 demonstrates that larger solutions have higher scores. This phenomenon is due to the the fact that the number of considered alterations is large. It can be compared to overfitting in high-dimensional prediction methods.

Indeed, the score of almost any solution can be improved by taking into account a new alteration that neighbors a few new red samples, and a moderate number of blue ones. Figure 1.1 for instance shows that the number of new red samples covered by an extension decreases rapidly to reach less than 10 samples among the 221 red ones. Thus, relying only on the score implies that biologically meaningful solutions are augmented by a lot of alterations that may not be relevant and will mislead the interpretation.

This is the reason why the choice is made to augment a solution if the new added solution has a score significantly higher than the score obtained with a randomly chosen alteration.

1.2 A combinatorial explosion

Figure 1.2 shows the distribution of the size in the 20000 first solutions. A combinatorial explosion of the number of solutions can be seen, the final decrease beeing related to the finite number of solutions considered.

The question of the choice of the step-score threshold has thus to be answered. The value of $2.5e^{-4}$ may seem a priori stringent. However, as shown in Figure 1.2, the combinatorial explosion of the number of solutions happens

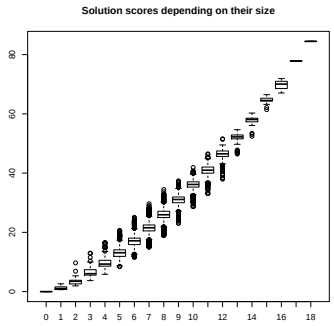

Figure 1: Distribution of the scores depending on the set size

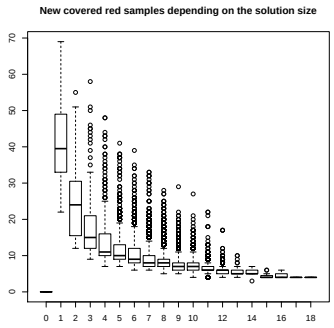

Figure 2: Distribution of the ranks of the number of red samples covered by the last added alteration, depending on the set size

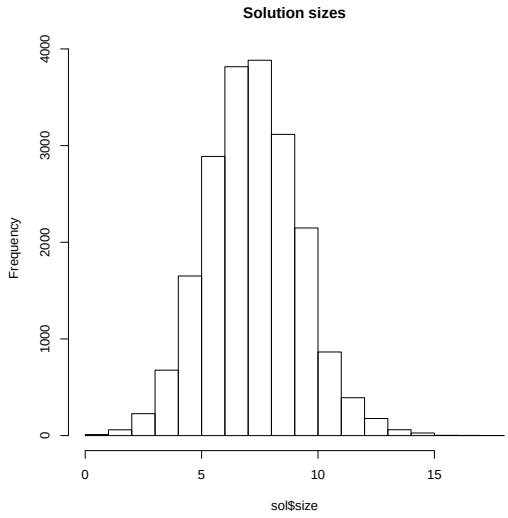

Figure 3: Solution size distribution among the 20000 first solutions

at an even lower value. The choice of the threshold can be done either on a statistical criterion by detecting the change in the slope in Figure 1.2, or by limiting the number of solutions because sets of moderate size may be easier to interpret biologically.

1.3 Sensitivity and specificity

A last plot in Figuer 1.3 shows the distribution of the sensitivity and specificity of the selected alteration sets depending on their size. It confirms that the two indices evolve slowlier when the size of the solution increases.

2 Computation of the step-score

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 neighbours of the alteration is drawn according to the empirical distribution of the alterations' degrees;

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

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

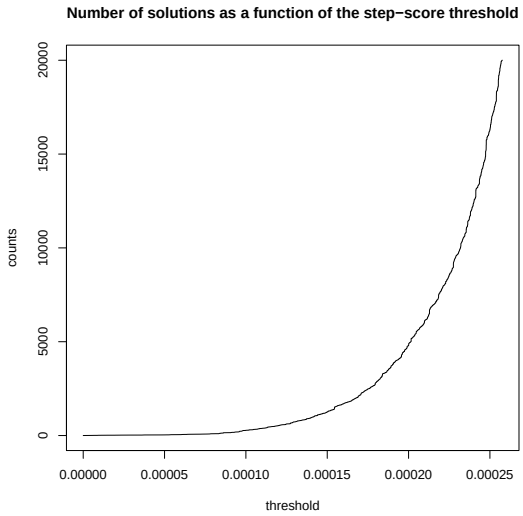

Figure 4: Number of solutions as a function of the chosen threshold

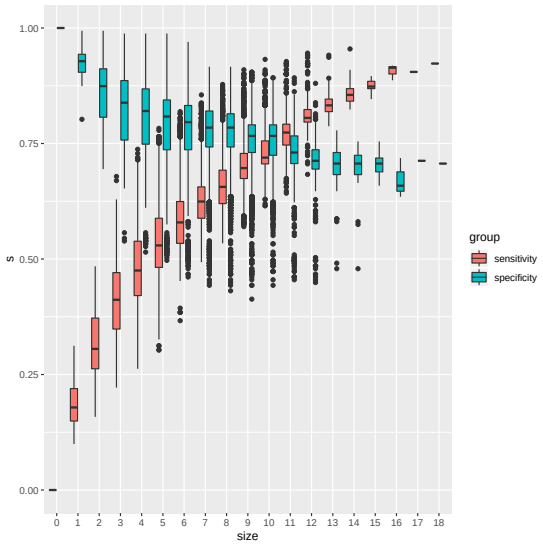

Figure 5: Evolution of the sensitivity and specificity depending in the set size.

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

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

In practice, the distribution of $c(A \cup \{b\})$ depends only on the data set and on A. More precisely, on $R, B, N_R(A), N_B(A)$, and the distribution of the alterations' degree. All of these parameters are fixed for a given data set, except $N_R(A)$ and $N_B(A)$, which can vary from 0 to R and from 0 to B. Given a pair $\big(N_R(A), N_B(A)\big)$, the following steps are performed in order to compute this distribution :

- $A \cup \{b\}$ will touch the samples already touched by A, as well as those touched by b. Some of the $N_T(b)$ samples touched by b are already touched by A and will have no effect at all. All that counts is the number of *new* samples touched by b, which will be n with probability $\sum_{k=0}^{T} P\big(N_T(b)) = k\big) \times h(k,n)$ where $h(k,n)$ is the probability, choosing k neighbours at random, of choosing n of them among those not already touched. This factor is computed thanks to the hypergeometric law.

- For each n, the probability distribution of m the number of red samples among those n new samples touched by b is computed using the hypergeometric law again.

- Each possible pair (n,m) happens with a certain probability and results in a certain score $c(A \cup \{b\})$: these scores are all looked up, and the pairs with their probability and score are sorted using this score as an ordering key.

- The result of this sorting is the probabilistic law of $c(A \cup \{b\})$. A summation gives the cumulative distribution, from which the p-value defining the step-score can easily be obtained.

In the algorithm below, this distribution, which depends on A, will be used many times in a row for the same A (to compute the step-score of adding every alteration a to it), and maybe reused later if another alteration set is considered, which touches as many red and blue samples as A. Thus, the distribution is computed on the first time, it is used to look up the step scores of all the extensions of A, then stored. If we later encounter another alteration set with the same numbers of red and blue neighbours, these two numbers will serve as a key to check if this distribution has already been computed. In theory this would require many Gigabytes of memory but in practice only few of all $R \times B$ distributions are actually needed.