Redundant siRNA Activity (RSA) Analysis by Example

For simplicity we assume a hypothetical small library of 14 genes represented by 40 independently designed siRNAs, one to four siRNAs per gene. The varying number of siRNAs per gene can be a result of many factors over time, such as merge of multiple siRNA libraries, change in gene structures, elimination of non-specific siRNAs, availability of reagent, etc. Cutoff is the most popular method, where 40 siRNAs are ranked by their activities and the top *X* wells are hit picked for validation. This is how the siRNAs are sorted in Figure 1, the eight most active siRNAs are highlighted as Cutoff Hits.

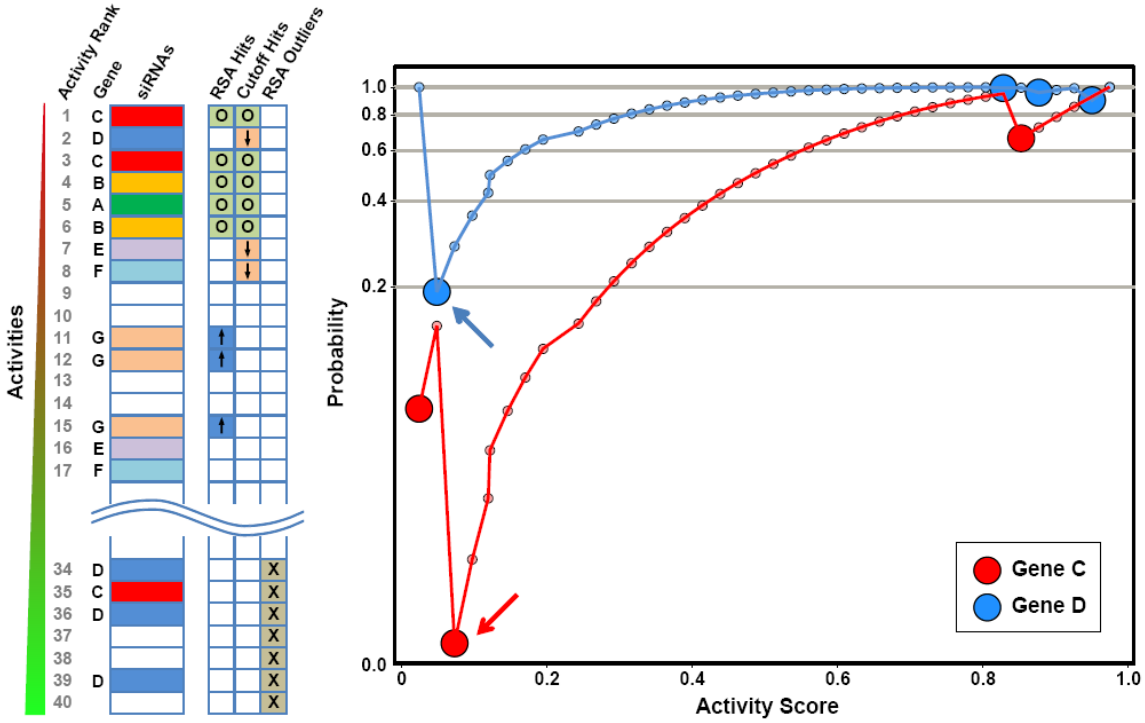

Figure 1. Illustration of RSA algorithm in siRNA hit selection. (a) Forty siRNAs are ranked according to their activities (potent on top) and colored according to their target gene identities. The top eight hits by both RSA and Cutoff algorithms are highlighted, with five common hits marked as "O", RSA-only hits as "↑" and Cutoff-only hits as "↓". siRNAs identified as outliers by RSA are marked as "X". (b) Interative RSA p-value calculation process as illustration by Gene C (3 siRNAs) and Gene D (4 siRNAs). For a given gene, accumulative hypergeometric p-values are calculated for each siRNA, the curve dips at each siRNA targeting the gene itself (big filled circle). The global minimum is then identified (indicated by arrow) and separate siRNAs into two groups: hits and outliers. One and three least potent siRNAs are identified as outliers for Gene C and D, respectively. Gene C achieves a global minimum of 0.01, much lower than the 0.2 for Gene D, therefore, the activity distribution of Gene C is much less likely to occur by chance, therefore the gene is more likely to be confirmed.

A reader is likely to agree that a gene having 4 relatively active siRNAs is more likely to be confirmed compared to a gene with 1 very active siRNA and 2 inactive siRNAs based on his own intuition. In the above scenario, the latter gene will be chosen first by the Cutoff method, because the single very active siRNA is ranked highest. The RSA algorithm, however, is aware of

the underlying siRNA library design and considers the behavior of all siRNAs of the same gene in the scoring function, which agrees well with our intuition. In an activity-sorted list, multiple siRNAs for a true positive gene would tend to be positioned towards the top. Such an upward bias in signal distribution would not occur if the gene is a true negative, the RSA scoring function essentially statistically characterized such a bias in signal distribution.

For gene C in red, its three siRNAs correspond to hypergeometric *p*-values of 0.08, 0.01 and 0.66, respectively. The minimum p-value 0.01 is obtained when its first two siRNAs are considered as hits and the last as outlier (red circles in Figure 1.b), all three siRNAs are assigned an RSA score of 0.01 (step 3-6 in Appendix). Similarly for gene D in blue, its four siRNAs correspond to *p*-values of 0. 2, 1.0, 1.0 and 0.9, respectively (blue circles in Figure 1.b). Only the first siRNA is considered as a hit and the remaining three are removed as outliers. The analysis is repeated for all the 14 genes and positive siRNAs are first ranked by their gene *p*-value (ascending) then by individual activities (potent to weak), the best eight siRNAs are highlighted as RSA hits and mostly inactive siRNAs are automatically identified as RSA outliers (marked as "X" in Figure 1.a).

Five out of the top eight hits between Cutoff and RSA algorithm are in common (marked as "O" in Figure 1.a), although their hit ranks may differ. It will be most enlightening to examine the other six hits unique to the two methods. Gene D (discussed previously) only has one out of four siRNAs being active, therefore the active well is likely to be a false positive and thus deprioritized by RSA (↓). Gene E and Gene F both have two siRNAs each, all are relatively active and therefore hit picked by the Cutoff method. However, Gene G has three siRNAs and all are relatively active as well. As three out of three siRNAs being active is certainly a stronger piece of evidence, RSA promote (↑) Gene G ahead of Gene E and F. Looking at individual siRNA signal alone, Cutoff method recruits Gene D, which is likely to be a false positive and missed Gene G, which is likely to be a false negative. The RSA algorithm on the other hand takes the signal distribution of all siRNAs of a gene into account and score the gene probabilistically. As the number of genes being studied increases and statistical fluctuation decreases, the edge of statistical advantage of RSA will be magnified and improvement of its hit list is expected to be more pronounced.

There is sometimes a misconception that RSA algorithm may bias towards genes with more siRNAs. This can be clarified by Gene D (four siRNAs, discussed previously) being scored poorly, Gene B (two siRNAs) being ranked as the second best gene for having both siRNAs being highly potent. Even Gene A with a single siRNA can still be ranked favorably under RSA for its unusually high potency. Nevertheless, the assumption that evidence from genes with single or fewer siRNAs is less compelling compared to genes with multiple active siRNAs agrees well with statistical sampling theory. Given a limited validation capacity, recruitment of genes of single active siRNA (e.g., Gene D) is often made at the expense of missing genes of multiple active siRNAs (e.g., Gene G). This is why Cutoff method results in much lower confirmation rate at the end.

Appendix

For simplicity we above discussed the RSA version without user-specified boundaries (as outlined below), please refer to the Supplementary Material & Method for details of a more complete RSA version (Renate et al. Nature Method. 2007. doi: 10.1038/nmeth1089).

1. Rank all siRNAs based on their activities in descending order (most potent on top)
2. For each gene *i*
3. For each of its siRNA$_{ij}$ (j = 1, ..., n_i)
4. Calculate enrichment factor by $f_{ij} = p(N_T, n_i, R_{ij}, j)$[§];
5. j^* = arg min$_j$ f_{ij} and $f_i^* = f_{ij*}$; Assign f_i^* to siRNA$_{ij}$ (j = 1, ..., n_i)
6. siRNA$_{ij}$ with $j \le j^*$ are marked as positives, with $j > j^*$ are removed as negatives
7. Rank all remaining siRNAs based on f_i^* in ascending order, then by R_{ij} is in ascending order

[§] N_T: total number of siRNAs in the library; R_{ij}: the rank number of siRNA$_{ij}$ in the sorted list; p: accumulated hypergeometric distribution function.