

## Improvement of DelMapper: software for deletion mapping of non-recombining region

K. Ishii,\*<sup>1</sup> Y. Kazama,\*<sup>1</sup> T. Ikeda,\*<sup>1</sup> and T. Abe\*<sup>1</sup>

Genetic mapping is usually constructed by calculating recombination rate as a measure of distance between genetic markers. However, this method cannot be applied to the non-recombining region such as sex chromosomes. To map the DNA markers or genes on the Y chromosome in *Silene latifolia*, we have developed new deletion mapping software, Del Mapper.<sup>1)</sup> Deletion mapping is an approach for mapping DNA markers or genes in the non-recombining region by collecting mutants having deletion on the mapping region and ordering the markers to have the least total number of chromosomal breakpoints in all mutants. DelMapper conducts deletion mapping with a new approach that boils down seeking the best marker order to solving traveling salesman problem.

DelMapper calculates the numbers of breakpoints in all possible permutations of the markers by applying the branch and bound algorithm. To promote greater efficiency of calculation, markers are clustered and each cluster is treated as a single virtual marker. When calculating the numbers of breakpoints in each permutation, DelMapper adds two virtual markers to the both ends of the mapping region. One virtual marker represents the essential chromosomal region that cannot be deleted and is presumed to be present in all mutants. There are two alternatives for the other virtual marker. Under the “Del” option, the mapping region is assumed to be located on the terminal of the chromosome and the virtual marker is regarded to be deleted. Under the “Any” option, the region is assumed to be located on the intermediate of the chromosome and the presence/absence of the marker is dependent on that of the neighboring marker. When run with the “Del” option, a previous version of DelMapper tends to output an increased number of marker orders with the least number of breakpoints. This problem is potentially caused by

concentrated deletions on the specific marker (e.g., marker B3 in Fig. 1). Such situation can be produced by the collection of mutants showing the same phenotype. In such cases, the concentrated deletions are located next to the terminal virtual marker. At the same time, “solitary deletions” that do not extend beyond the border of clusters appears due to disjunction of deletions that are to be continuous (Fig. 1; deletions in clusters B and C in Mutants 3, 5, and 6).

To cope with this problem, we added a routine to calculate the number of “solitary deletions.” In addition to the number of breakpoints, DelMapper counts the number of “solitary deletions” in all given permutations of markers. Users can adopt the permutation(s) of the markers with the least number of “solitary deletions” in those with the least number of breakpoints as the mapping result. As a verification of the effects of the routine, we made 30 virtual maps that consisted of 71 markers, 41 mutants, and 500 instances of deleted markers and had a tendency of biased deletion on the specific markers by a Perl script DelMapMaker.<sup>1)</sup> We tested DelMapper whether it could reconstruct these maps. DelMapper run with the “Any” option output an average of 1.3 marker orders with the least number of “solitary deletions” in those with the least number of breakpoints regardless of the implementation of the routine. On the other hand, DelMapper run with the “Del” option output an average of 2.5 marker orders before the implementation, while after the implementation, it did an average of 1.3 marker orders, which was the same value as that with the “Any” option.

In this study, we successfully decreased the number of the false-positive best marker orders from the output of DelMapper with the “Del” option, which enables users to conduct more clear deletion mapping.

A														B															
Cluster	A			B			C			L	R	P	S	Cluster	C			A			B			L	R	P	S		
Marker	L	A1	A2	A3	B1	B2	B3	C1	C2	C3	R	P	S	Marker	L	C2	C3	C1	A1	A2	A3	B1	B2	B3	R	P	S		
Mutant1	1	1	0	0	0	0	0	1	1	1	0	3	0	Mutant1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	
Mutant2	1	0	0	0	0	0	0	0	1	1	0	3	0	Mutant2	1	1	1	0	0	0	0	0	0	0	0	0	1	0	
Mutant3	1	1	1	1	1	0	0	0	0	0	0	1	0	Mutant3	1	0	0	0	1	1	1	1	0	0	0	3	2		
Mutant4	1	1	1	0	0	0	0	1	1	1	0	3	0	Mutant4	1	1	1	1	1	1	0	0	0	0	0	1	0		
Mutant5	1	1	1	1	1	1	0	0	0	0	0	1	0	Mutant5	1	0	0	0	1	1	1	1	1	0	0	3	2		
Mutant6	1	1	1	1	0	0	0	0	0	0	0	1	0	Mutant6	1	0	0	0	1	1	1	1	0	0	0	3	2		
Total													12	0	Total													12	6

Fig.1 Two examples of the result of DelMapper with the “Del” option. L and R indicate virtual markers representing the ends of the mapping region. “0” and “1” indicate that the markers are absent and present in the corresponding mutant, respectively. P and S indicate the numbers of breakpoints and “solitary deletion,” respectively. The routine calculating “solitary deletions” helps users to determine that the marker order of (A) is more plausible than (B) because of the smaller number of “solitary deletions” although the numbers of breakpoints are same.

### Reference

1) Y. Kazama and K. Ishii et al.: Sci. Rep. 6, 18917 (2016).

\*<sup>1</sup> RIKEN Nishina Center