A Database System for High-Throughput Transposon Display Analyses of Rice

Etsuko Inoue[1] Takuya Yoshihiro[2] Hideya Kawaji[3] Akira Horibata[4] Masaru Nakagawa[2]
[1]*Graduate School of Systems Engineering, Wakayama University, Sakaedani, Wakayama 640-8510, Japan*
[2]*Faculty of Systems Engineering, Wakayama University, Sakaedani, Wakayama 640-8510, Japan*
[3]*NTT Software Corporation, Naka-ku, Yokohama, Kanagawa, 231-8554, Japan*
[4]*Faculty of Biology-Oriented Science and Technology, Kinki University, Uchida, Wakayama 649-6433, Japan*
Email : s031007@sys.wakayama-u.ac.jp, tac@sys.wakayama-u.ac.jp, kawaji@po.ntts.co.jp, horibata@waka.kindai.ac.jp, nakagawa@sys.wakayama-u.ac.jp

ABSTRACT: We developed a database system to enable efficient and high-throughput transposon analyses in rice. We grow large-scale mutant series of rice by taking advantage of an active MITE transposon *mPing*, and apply the transposon display method to them to study correlation between genotypes and phenotypes. But the analytical phase, in which we find mutation spots from waveform data called fragment profiles, involves several problems from a viewpoint of labor amount, data management, and reliability of the result. As a solution, our database system manages all the analytical data throughout the experiments, and provides several functions and well designed web interfaces to perform overall analyses reliably and efficiently.

4 INTRODUCTION

Our study group analyzes genotype-phenotype correlations in rice species. We are trying to apply the transposon display analysis using experimental rice lines which hold transposon *mPing*. *mPing* is an active MITE transposon found in a rice line by Nakazaki, et al [1]. When we grow the next generation lines by inbreeding, several *mPing*s move around in genome and sometimes inactivate genes. Our objective is to determine the mutated gene and take correlation with observed traits. Here, it is worth noting that *mPing* has several convenient features for our analysis: 1) *mPing* is a native DNA sequence found in an experimental rice line, so that we can easily grow the line in a large scale, 2) *mPing* has moderate mobility under a natural environment, so that we can breed mutants efficiently, 3) the number of *mPing* copies in each individual is not too much (about 100 copies), so that the number of mutant positions are moderate to analyze. Taking advantage of those features of *mPing*, we can easily obtain large-scale mutant series.

Our experiment starts with growing mutant series. We grow experimental lines of rice in a massive scale and obtain a series of next-generation individuals. At this point we can observe several mutations in phenotypes, so we collect the data of several target traits such as glume-shape, height of ears, and so on. Those data are used afterwards to take correlation with genotypes. Then for each individual, we examine its genotype by applying the transposon display method [2]. This method is a variation of AFLP method; the main difference is to use transposon-specific primers in PCR amplification. Anyway, we obtain a waveform data called "fragment profiles" as a result. This method includes the following steps: we first extract DNA from a sample individual. Then digest them with a restriction enzyme. Isolate the relative fragments with some *mPing*-specific primers. Amplify them by PCR method. Then electrophorese them using DNA sequencers to obtain a fragment profile. The fragment profile is in fact a data list of fluorescence with a constant time interval. Since the time implies the fragment length in the electrophoretic data, the fragment profile represents the amount of fragments of each length. Note that the fragment profiles always differ if some *mPing* moves around. Thus all we have to do is to find the differences among fragment profiles, which we call mutation spots hereafter.

In finding mutation spots, however, we found two problems: I) paper-based comparison of fragment profiles requires considerable costs of manual laborious procedure, II) the analytical results vary with researchers because the expressions of the profiles are so subtly. The former one is about the cost of comparing fragment profiles. Paper-based comparison is one of the most primitive ways indeed, but this is still the realistic way if no well-applicable software is available. In this case, finding mutant spots requires considerable labor because the fragment profile is itself subtle and moreover, its waveform is affected by experimental environments such as temperature or human variations. On the other hand, the latter problem, the variation among researchers, is about reliability of the analytical result. In fact, the result may vary even if the same person did it. This affects much when the scale of the experiment becomes larger. From the reasons above, this analytical phase becomes a bottleneck in large scale experiments.

To solve those problems, we designed and developed a database system. Our system provides web interfaces to perform whole process of the analytical phase efficiently. Particularly, to solve the problem of laborious cost, we provide a mechanism to overlap two fragment profiles in a screen so that we can find mutation spots quickly. Also, we provide a function of computing candidates of mutation spots to support finding mutation spots. From the other aspect, our system manages all the data in the analytical phase of our system. This enables us to confirm data afterwards to improve reliability of the analytical results. This also enables us to share all the analytical data within a group. Those advantages will really helpful especially when we plan to do large-scale transposon display analysis.

In this paper, we introduce a database system developed

to support transposon display analyses. This paper is organized as follows: In Section 2, we introduce the whole functions and interfaces of the database system. Section 3 describes the algorithms of the key functions, i.e., overlapping fragment profiles and computing candidates of mutation spots. Section 4 gives discussions about the database system. Finally, Section 5 gives conclusions and future work.

5 FUNCTIONS AND INTERFACES OF THE DATABASE SYSTEM

Our database system supports the whole process of our analysis. All data in our work (e.g., fragment profiles and mutation spots found in the transposon display analysis, lineage information and traits of individuals) are managed in our database, and the system provides interfaces to handle those data. Since our database system is implemented on Linux providing web service, we can do the whole analysis with a web browser.

Our database system consists of two parts. One provides the functions to support the process of the transposon display analysis, especially to facilitate the comparison of fragment profiles. The other provides the functions to register and browse lineage information and traits of individuals.

The former part, that is to facilitate comparison of fragment profiles, consists of four steps. i) Registering fragment profiles: we register fragment profiles, which are obtained as a data file through the transposon display analysis, into the database. ii) Displaying an image of overlapped two fragment profiles: it helps us to detect the mutation spots between two target individuals. iii) Registering mutation spots: to facilitate comparing two fragment profiles, our system provides a list of candidates of mutation spots calculated automatically. We can also edit the list manually, e.g., add or delete its elements, and register it as a list of mutation spots. iv) Comparing mutation spots in a group: we compare the mutation spots obtained in the step iii) to find the common mutation spots within a group. The mutation spots are shown as a tabulated list. We can integrate adjacent spots if we regard them as the same mutation spots.

The latter part, that is to manage data of lineage information and traits of individuals, can be handled as follows. We treat linage information and traits of individuals separately because each of them has a different timing to be registered into the database. Lineage information and traits of individuals are recorded in different CSV-format files, and we register the data into the system by uploading them. To update the data, we first export target data from the system as a CSV-format file, edit it, and then upload the file again.

In this section, we explain about those two parts in detail.

2.1 Interfaces for Transposon Display Analyses

Here, we explain the former part to facilitate comparison of fragment profiles in the transposon display analysis. As described above, this part consists of four steps. We give a detail explanation for each of them.

i) Registering fragment profiles. The first step for the transposon display analysis on this system is to register fragment profiles into the database. There are two ways to register fragment profiles. One is uploading files one by one via data registration screen, and the other is uploading a mass of files simultaneously via FTP. Every uploaded fragment profile is associated with an individual ID, and then stored in the database. The registered data can be seen by list form on the screen.

ii) Displaying an image of overlapped two fragment profiles. In this step, we select two individuals (one is a reference data as a standard of comparison and the other is a target data) from a list of individuals on the system, and then display a graph image of the overlapped two fragment profiles (Fig. 1). In this figure the graph is shown, and is created using the algorithm overlapping two fragment profiles; the mechanism of the algorithm is described in Section 3. The graph with fluorescence intensity as the vertical axis and EST size (base pair) as the horizontal axis has two waveform patterns with different colors: using red for reference data and blue for target data. Note that the graph has two vertical axes on either side, since the two fragment profiles have different scales of fluorescence intensity.

iii) Registering mutation spots. We support detecting mutation spots by providing candidates of mutation spots. Candidates of mutation spots are computed when a graph

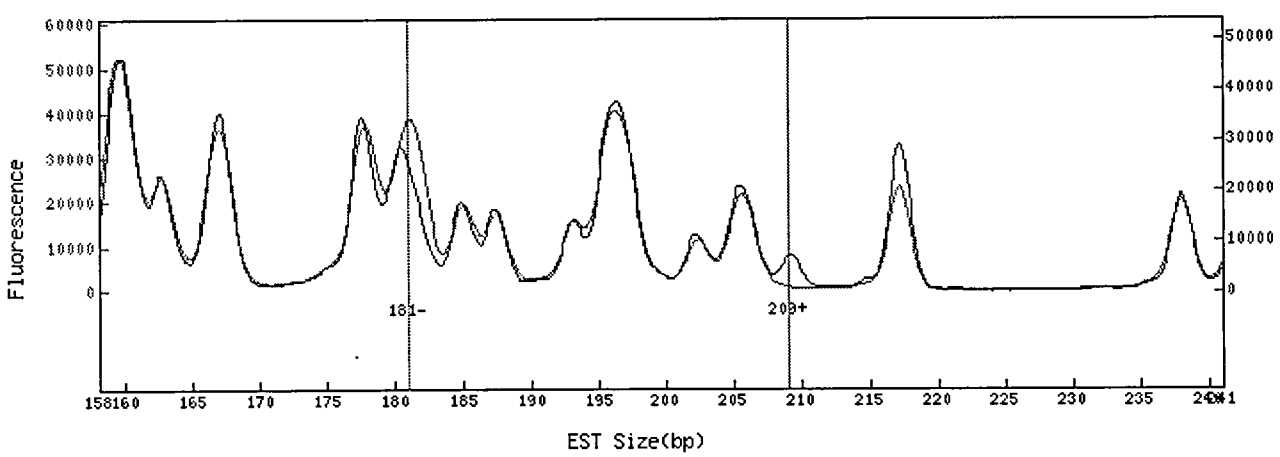


Figure 1: The overlapped fragment profiles with the candidates of mutation spots indicated.

image is created. A candidate also can be added manually. We can select mutation spots from a list of candidates. However, not all of candidates are actual mutation spots. Thus it is necessary to confirm manually whether each mutation spot is certainly regarded as a mutation spot or not. After the confirmation, mutation spots are registered into the system via mutation comparison screen. Computed candidates and registered mutation spots are shown as a vertical line on the graph with an integral value, which indicates the EST size of the mutation spot (Fig. 1). Suffix symbols on the values indicates a type of mutation, namely, "+" means the fragment (peak) appears only in the target data, and "-" means that it appears only in the reference data.

iv) Comparing mutation spots in a group. Registered mutation spots in a group can be displayed as a tabulated list simultaneously. Note that comparison of fragment profiles on the screen, which is described in step ii), deals with only two data, and so this limitation sometimes causes mistaking registration of mutation spots as different EST size. Suppose, for instance, that there is one mutation spot around 300bp. Then, it is registered as 299bp on some individuals' fragment profile and is also registered as 301bp on other individuals' fragment profiles. Such differences of mutation spots make troubles in accurate analyses because those mutation spots registered as different EST size are treated separately, i.e., 299bp is one mutation spot and 301bp is another one. To solve this problem, the interface enables us to integrate adjacent mutation spots that a user regards as the identical spot. Using this interface, we can integrate several adjacent spots into one mutation spot, or restore an integrated mutation spot into the original separated spots. Here, note that due to the tabulated view of mutation spots in a group, we can make the integrating work efficiently. The "group" here means an analytical group to compare mutation spots simultaneously, and the group can be created by users. The members of a group are usually expected to be compared simultaneously, and so it can be, for example, in the same experimental line. Figure 2 shows an example of a tabulated view with the group of the line "2004-MLR009". In this figure, character strings in the left-end column are individual IDs of group members, the numbers (106, 107,..., 622) on the first row indicate EST sizes, the symbols ("+" and "-") indicate presence of registered mutation spots, and EST size 300bp whose background is colored grey indicates that several spots are integrated into the spot of 300bp.

2.2 Managing Linage Information and Traits

Although lineage information and traits of individuals were conventionally managed with spreadsheet application software, such a file-based management is not useful enough if the handling data become complicated and large-scaled. In our case, there are also the same kind of problems that lineage information and traits of individuals have many items and we have to register them into the system in different timings. The best solution of this is to manage all data with database management system (DBMS). Also, in order to make it possible for researchers to handle data with the accustomed spreadsheet software, we adopt the way to register and derive data with CSV-format data files. Besides, we prepare several CSV formats to improve efficiency in several patterns of use.

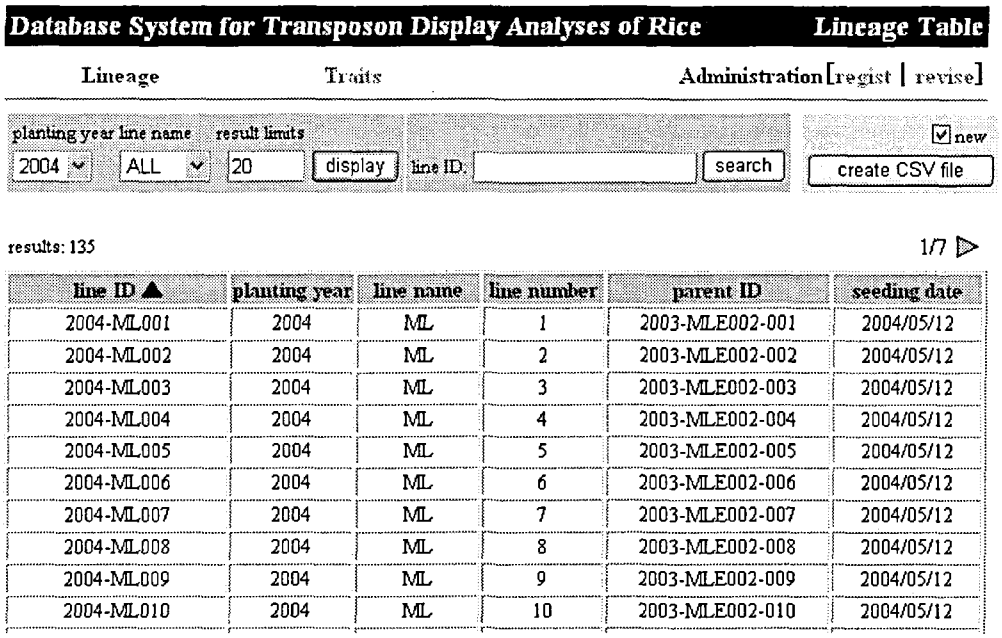
Lineage information consists of five items: planting year, name and number of lines, individual ID of a parent and seeding date. The lineage information is obtained when they are planted in a field. Trait information consists of an individual ID and more than 20 items of trait values which are classified into two groups; one is a group of the traits obtained during the time of harvest (e.g., heading date, ear length, grain shape, and so on.) and the other is a group of the traits which require additional tests or experiments (e.g., cool weather resistance, drought resistance, and so on). In addition to the two CSV formats for lineage information and traits of individuals, our system prepares the format for registration of a single trait only.

The registered data on the database can be displayed on the screen. Our system has a tabulated view interfaces for the lineage information and the traits separately, where simple searching and sorting functions are available. The displayed data can be exported into CSV format file via the viewing interfaces. When those data are registered into the database by uploading CSV format files, key strings to keep consistency are attached to them, which are used to update data by uploading exported files.

Figure 3 is an example of a tabulated view of lineage information. In this view, we can display lineage information by setting search condition for planting year, line name, and line ID. In the tabulated list, there are six items: line ID, planting year, line name, line number, parent ID (individual ID of the parent), and seeding date. The button "create CSV file" on right above of the list is for downloading a new data sheet for registering data and for exporting the data displayed in the list.

individual ID	106	107	117	228	232	285	290	300	356	362	391	392	413	419	458	478	486	492	502	514	535	558	620	622
2004-MLR009-002						-		+	+		-			-				+			-			
2004-MLR009-003								+			-			-								+	+	
2004-MLR009-004								+				-		-										+
2004-MLR009-005					+			+			-			-	+					-	-			
2004-MLR009-006		+	+	+			+	+						-						-	-			
2004-MLR009-007	-							+		-	-			-		-								
2004-MLR009-008	-							+		-	-			-						-				
2004-MLR009-009	-							+		-	-		+	-	+				+					
2004-MLR009-010	-							+			-			-		-	-				-			

Figure 2: The tabulated list of registered mutation spots.



line ID ▲	planting year	line name	line number	parent ID	seeding date
2004-ML001	2004	ML	1	2003-MLE002-001	2004/05/12
2004-ML002	2004	ML	2	2003-MLE002-002	2004/05/12
2004-ML003	2004	ML	3	2003-MLE002-003	2004/05/12
2004-ML004	2004	ML	4	2003-MLE002-004	2004/05/12
2004-ML005	2004	ML	5	2003-MLE002-005	2004/05/12
2004-ML006	2004	ML	6	2003-MLE002-006	2004/05/12
2004-ML007	2004	ML	7	2003-MLE002-007	2004/05/12
2004-ML008	2004	ML	8	2003-MLE002-008	2004/05/12
2004-ML009	2004	ML	9	2003-MLE002-009	2004/05/12
2004-ML010	2004	ML	10	2003-MLE002-010	2004/05/12

Figure 3: The tabulated list view of lineage information.

6 ALGORITHMS FOR COMPARING FRAGMENT PROFILES

Our database system improves the problem of laborious cost by providing functions to facilitate comparison of fragment profiles. Those are achieved, in the basis, by two algorithms: one is to overlap fragment profiles and the other is to compute candidates of mutation spots. The former algorithm enables us to compare fragment profiles efficiently and quickly in the screen so that the high-precision analysis is performed. The latter algorithm helps us detecting mutation spots to reduce analyzing time. Those two algorithms are both playing important roles in our interface to support analysis.

In this section, we describe the basic mechanism of those two algorithms. Note that, here, we introduce just basic ideas and mechanisms. In fact, we have to make several detail improvements to reach a sufficient level to be in practical use.

3.1 Overlapping Fragment Profiles

A fragment profile, which is an electrophoretogram obtained from DNA sequencers, is not in fact an image file but a data list of fluorescence intensities observed with a constant time interval. In order to detect mutation spots, it is necessary to overlap exactly the two fragment profiles to be compared. Fragment profiles, however, include various kinds of noises by nature. The profile is easily affected by experimental environments such as temperature or human variations. Also, even electrophoretic speed varies at every moment; especially it has a tendency to reduce speed as time passes. Therefore, overlapping fragment profiles cannot be done easily.

Our idea for overlapping profiles is to convert the profile data from a time-scale fluorescence list into basepair-scale one. This conversion is done by using DNA size-marker which is mixed in our experimental process (and dyed with different fluorochromes) into the sample to be electrophoresed. The size-marker includes only the particular sizes of fragments so that it makes the corresponding peaks in its electrophoretogram. Using the peak as a reference point, we convert the scale of the profiles from time index to fragment length. (We call the length "EST size" hereafter).

The basic algorithm of detecting marker peaks is as follows: in a fragment profile with a size-marker waveform, we detect the points where the fluorescence intensity of the size-marker is larger than a threshold. (Note that the threshold should be computed from the intensity values of the size-marker waveform. The scale of the waveform also varies with profiles.) The point that we detect is in fact an interval where the fluorescence intensities are larger than the threshold. So we found the exact point of largest intensity in the interval, and make it a reference point. In our experimental process, we use the size-marker which consists of the fragments of 80bp, 90bp, 100bp, 110bp, 120bp, 140bp, and so on. The reference points represent the place of those EST sizes in time-scaled fragment profiles.

Making use of reference points, we convert the scale of the fragment profiles. Now let p (p=1,2,3,..,n) be the sampling points of an electrophoretic data. Let l_i (i=1,2,3,..,m) be the EST size of the i-th peak of the size-marker and also let m_i be the sampling point of the i-th peak. Then, the EST size X_p at the sampling point p is calculated as shown in the following.

$$X_p = l_i + \frac{l_{i+1} - l_i}{m_{i+1} - m_i} * (p - m_i) \quad \text{if} \quad (m_i \le p < m_{i+1})$$

If we convert the scale into EST size, then we can naturally overlap exactly the fragment profiles. In this algorithm, sometimes the marker-detecting procedure fails, and in that case, exact overlapping also fails. This failure occurs even when the human recognition of peaks is

possible. Thus we need several improvements for practical use. Now we omit the explanation of additional improvement, but we achieved a sufficient level for practice use.

Finally, we give an issue about the fluorescence intensity. Certainly, the scale of the intensity varies with fragment profiles. However, this can be solved only by applying an appropriate measure scale for each profile in the screen. (See fig. 1 again. You will see the two measures of the intensity value.)

3.2 Listing candidates of mutation spots

The detection of mutation spots requires enough attention even if we overlap fragment profiles. To help the process, our system computes candidates of mutation spots automatically. Of course, the researchers must confirm all the mutation spots finally, but nevertheless, this function becomes of some help for researchers.

Computing candidates of mutation spots includes two steps: first we create two peak lists by detecting peaks from the fluorescence intensities of each fragment profile. Next we take a matching of the two peak lists, namely, we make pairs of peaks between two peak lists. Then we collect unmatched peaks as the candidates.

Creating a peak list for a fragment profile is done as follows: as a pre-processing procedure, we take moving averages of the fragment profiles with the range of 1 bp to reduce affection of subtle noises. Then we detect peaks by finding the point that the inclination of the waveform changes from positive value to negative.

After two peak lists are created, we take a matching of the two. First, for each peak in one peak list, we make a nomination to a peak of another peak list, by selecting the closest peak in EST size. We do the same on the other peak list. Then, if two peaks in the different peak list nominate each other, we regard the pair as matched. Since the matched peaks can be regarded as non-mutation spots (Of course only if the two EST sizes are so close), the rest is the candidates of the mutation spots.

Those are the basic ideas of the algorithm. However, if we implement the algorithm naively, the detecting rate is not so good. Thus we perform several improvements in both the peak detection phase and the matching phase. For instance, the following technique is applied. In taking a matching, we often meet the case that the intensities of the matched peaks are far different. In this case it is supposable that the peak contains several different fragments, namely, this may also be regarded as a mutation spot. Thus we do not match peaks if the intensities are far different. Similarly, several techniques are applied in our system to be in practical use.

7 DISCUSSION

From the experience of running our experiment for one period, we confirmed the usefulness of our system. Particularly, the interface to overlap fragment profiles is very effective and we can reduce laborious cost vastly, say, at least less than 1/10 in researchers' impression. Also, by making use of this system, we have achieved more reliable and accurate analysis. We consider that it is important for reliability to preserve the evidence to lead analytical results in a database. Through the experiment using the system, Horibata et al., has found a mutation spot which has correlation with grain-shape in rice [3].

As for algorithms, we evaluate the quality of the two by applying several test data. The algorithm to overlap two fragment profiles converts the scale of fragment profile from time scale to fragment length. We tried to convert hundreds of test data, and found several failures as a result. However, they are all found to be the case of electrophoresis failure. For instance, sometimes a part of the peaks of size-marker does not appear or too weak. In such a case, our algorithm does not run correctly indeed, but in the most case, the data itself is also useless. If we do not count the case of electrophoresis failure, our algorithm runs correctly for almost 100% cases.

The second algorithm is computing candidates of mutation spots. For the algorithm, we also apply hundreds of data, and compare the set of candidates computed by the algorithm and the set of mutation spots which is selected manually by researchers. As a result, about 75% of the manually selected mutation spots are included in the candidates computed. This result is clearly not so good. However, it will be not easy to improve this result because the errors are seen only when the differences between the two fragment profiles are so subtle. As the other result, only about 40% of candidates are the mutation spots. But this low rate is not so bad since the user do not feel troublesome to select several spots out of many candidates in the system interface. Users rather hate to register the new spots because it requires them to be more careful.

As another topic, to connect the two components of the system will be important as the next work. One is the part to support the transposon display analysis, and the other is the part to manage lines, individuals, and their traits. Now those two do not run in corporation with each other so that we have to find correlation between mutation spots and traits manually. This work, however, also takes considerable costs and if the scale of the experiment becomes larger, the costs also become much greater. Thus we are now planning to implement some statistical method to test correlations and some other useful methods, to reduce analytical costs.

8 CONCLUSION

In this paper, we introduced a database system to support large-scale transposon display analysis. To make a sequence of analyses efficiently, we provide a web interface to find mutation spots on the overlapped fragment profiles, and also provide a function to compute the candidates. Through using this system for one period of our experiment, we found that those interfaces really work well and the laborious cost reduces dramatically. Also, the reliability of the analytical result is improved by managing all the analytical data throughout our experiment.

As a future work, we have two directions of work. One is to improve algorithms to detect mutation spots. The improvement may not be easy, but the possibility still remains. The other is to implement the function of correlation analysis. On the basic correlation analysis, it is strongly desired to reduce cost of manual labor. It seems so interesting if some advanced analyses can discover some beneficial knowledge from our database.

ACKNOWLEDGEMENT

This work was partly supported by the Wakayama Prefecture Collaboration of Regional Entities for the Advancement of Technological Excellence, JST.

REFERENCES

[1] T. Nakazaki, Y. Okumoto, A. Horibata, S. Yamahira, M. Teraishi, H. Nishida, H. Inoue, and T. Tanisaka, "Mobilization of Transposon in the rice genome," Nature, 421, pp.170—172, 2003.

[2] S. Ayyadevara, J. J. Thaden and R. J. Shmookler Reis, "Anchor Polymerase Chain Reaction Display: A High-Throughput Method to Resolve, Score, and Isolate Dimorphic Genetic Markers Based on Interspersed Repetitive DNA Elements," Analytical Biochemistry, 284, pp.19—28, 2000.

[3] A. Horibata, K. Matsui, E. Inoue, T. Yoshihiro, H. Kawaji, M. Nakagawa, Y. Okumoto, N. Nakazaki, T. Tanisaka, "Spontaneous and Frequent Transposition of a Miniature Inverted-Repeat Transposable Element, *mPing*, in an Experimental Line of Rice (*Oryza sativa* L.)," The Society for the Advancement of Breeding Researches in Asia and Oceania (SABRAO), 2005.