BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 16 2007, pages 2155–2162
doi:10.1093/bioinformatics/btm313

Data and text mining

PowerCore: a program applying the advanced M strategy with a heuristic search for establishing core sets

Kyu-Won Kim[1,2,†], Hun-Ki Chung[1,†], Gyu-Taek Cho[1], Kyung-Ho Ma[1], Dorothy Chandrabalan[3], Jae-Gyun Gwag[1], Tae-San Kim[1], Eun-Gi Cho[1] and Yong-Jin Park[1,3,*]

[1]National Institute of Agricultural Biotechnology, 247 Seodun-dong, Suwon, 441-707, [2]Qubesoft,R/No, Dongyoung Central B/D, 847-2 Geumjeong-dong, Gunpo 434-050, R.Korea and [3]Bioversity International, APO Office, Serdang 43400, Malaysia

Received on February 28, 2007; revised on May 24, 2007; accepted on June 5, 2007

Advance Access publication June 22, 2007

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Core sets are necessary to ensure that access to useful alleles or characteristics retained in genebanks is guaranteed. We have successfully developed a computational tool named 'PowerCore' that aims to support the development of core sets by reducing the redundancy of useful alleles and thus enhancing their richness.

Results: The program, using a new approach completely different from any other previous methodologies, selects entries of core sets by the advanced M (maximization) strategy implemented through a modified heuristic algorithm. The developed core set has been validated to retain all characteristics for qualitative traits and all classes for quantitative ones. PowerCore effectively selected the accessions with higher diversity representing the entire coverage of variables and gave a 100% reproducible list of entries whenever repeated.

Availability: PowerCore software uses the .NET Framework Version 1.1 environment which is freely available for the MS Windows platform. The files can be downloaded from http://genebank.rda.go.kr/powercore/. The distribution of the package includes executable programs, sample data and a user manual.

Contact: yjpark@rda.go.kr

1 INTRODUCTION

Useful alleles, especially those contributing to valuable agronomic traits are often conserved in genebanks worldwide. The potential use of these large collections could be greatly enhanced by constituting subsamples also known as core collections or core sets (Basigalup *et al.*, 1995; Brown, 1989; Franco *et al.*, 2006; Frankel and Brown, 1984; Upadhyaya *et al.*, 2006). Effective deployment of useful alleles from genebanks has been made possible especially with the recent technological revolution brought upon by genomic and bioinformatics tools. Allele mining exploits the deoxyribonucleic acid (DNA) sequence of one genotype to isolate useful alleles from related genotypes (Latha *et al.*, 2004). Discovering the full diversity of available genes and their agronomic significance will allow genebanks to achieve their full potential thus contributing to sustainable development by deployment of the right alleles in the right places at the right time (Hamilton and McNally, 2005).

Over the years, tremendous progress has been achieved using different methodologies including the stratified random sampling, and such methodologies have been successfully applied to develop core collections for various uses (Balfourier *et al.*, 1998; Chandra *et al.*, 2002; Hu *et al.*, 2000; Peeters and Martinelli, 1989; Spagnoletti and Qualset, 1993). Several other strategies have also been proposed for use including proportional allocation, log frequency allocation and the constant allocation (Brown, 1989; Spagnoletti and Qualset, 1993; van Hintum *et al.*, 2000). New trials such as the M (maximization) strategy or nested selection methods (Bataillon *et al.*, 1996; Marita *et al.*, 2002; Schoen and Brown, 1993) have been conducted to select specific combinations of accessions that include complete coverage and retention. Similarly, using iterative procedures of selecting the highest diversity among subsets by the criterion of richness and the highest sum of squares of active variables based on the M strategy, the MSTRAT program was developed and released (Gouesnard *et al.*, 2001). To date, the M strategy is clearly the most powerful function for selecting entries with the most diverse alleles and eliminating redundancy that comes from non-informative alleles, which arise from co-ancestry and certain assertive mating systems in establishing core sets (Franco *et al.*, 2006).

As a solution to the traveling salesman problem (TSP), the 'heuristic algorithm' was designed for selecting the optimal pathway to the last goal following the Karg–Thompson's algorithm (Karg and Thompson, 1965) and later improved to not only search the best increment for each node, but also the next-best increment (Raymond, 1969). Various applications of the heuristic algorithm include the FASTA program for sequence comparison (Altschul *et al.*, 1990), GeneMark for the *ab initio* gene search program (Besemer and Borodovsky, 2005),

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

GenAlignRefine for the multiple sequence alignment program (Wang and Lefkowitz, 2005) and Bounded Sparse Dynamic Programming (BSDP) (Slater and Birney, 2005). The heuristic algorithm was also applied in developing the core set for the Arabidopsis collection using single nucleotide polymorphism (SNP) data (McKhann *et al.*, 2004).

Here, we present a new software application named PowerCore, which can be applied for developing core sets using the advanced M strategy and possessing the power to represent all alleles or classes of their observations.

2 DESIGN CONCEPT

Scales for variables expressing traits of genetic accessions vary based on their characteristics and measurement methods. These are the nominal, ordinal, interval and ratio scales. The interval and ratio scales may categorize and divide variants into an appropriate interval. They can then be categorized under the ordinal scale. The ordinal scale may also be used as a nominal scale as shown in Figure 1.

When one converts several variables expressing traits of accessions into one nominal scale according to the method above, one may assume a set, S_v^A, with elements of all nominal values in the set of the whole accessions, A with respect to a certain variable, v (certain repetitive nominal values may occupy an element of S_v^A). S_V^A is a set with elements $S_{v_1}^A, S_{v_2}^A, \ldots, S_{v_m}^A$, with respect to variables $v_1, v_2, \ldots, v_m$. In other words, $S_V^A = \{S_v^A \mid v \in$ all the variables of the whole accessions$\}$. In addition, if $S_v^A = S_v^B$ for all the variables, then let S_V^A be equal to S_V^B ($S_V^A = S_V^B$) (Fig. 1).

At this point, one may consider subsets, A_{sub}, of the set of whole accessions, A, in which $S_V^A = S_V^{A_{\text{sub}}}$. Each A_{sub} exhibits all nominal values of each variable expressed by the set A, one of which with the minimum number of elements can be represented as a core collection. Thus, the problem in finding the representative accessions with the minimum number of accessions may be expressed as the problem of finding an A_{sub} with the minimum number of elements out of every A_{sub} sufficing $S_V^A = S_V^{A_{\text{sub}}}$.

To find an A_{sub} where $S_V^A = S_V^{A_{\text{sub}}}$ with the approach above, one may create an empty set, E, and add a certain appropriate accession to E recursively until S_V^E and S_V^A become equal. This process may also be described as the shortest-path problem. If the set, E, contains no element, then it is in the initial state. If S_V^E and S_V^A are equal to each other, then it becomes the final state, or in other words, the goal. Selecting an entry and adding it to E is an expansion of a node. Thus, reaching the goal with the minimum number of elements in E using this method involves minimizing the number of nodes from the initial node to the goal. However, this search process does not consider the order of accessions. For example, suppose there are accessions, a, b and c, then six different paths may exist when adding to E. These paths all have the same significance: $a \rightarrow b \rightarrow c$, $a \rightarrow c \rightarrow b$, $b \rightarrow a \rightarrow c$, $b \rightarrow c \rightarrow a$, $c \rightarrow a \rightarrow b$ and $c \rightarrow b \rightarrow a$. In other words, if one of them were to be expanded in a search process, it would not be necessary to expand the rest.

The problem in finding a core collection, therefore, may be expressed as searching for the shortest path with the minimum

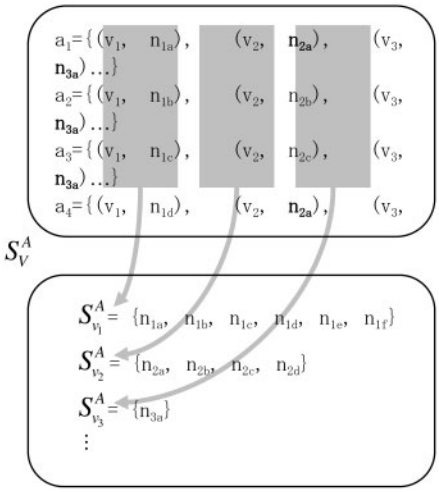

Fig. 1. A set of nominal values of variables expressing traits of genetic accessions (a: accession; v: variable; n: nominal value).

number of nodes in the search process above which may be discovered using the A*-algorithm.

If an optimal path exists from the initial node, s, to the final node via a node, n, one may define the cost of the optimal path from s to n as $g^*(n)$ and the cost of the optimal path from n to the final node as $h^*(n)$. Then, let us define the sum of $g^*(n)$ and $h^*(n)$ as $f^*(n)$ as follows:

$$f^*(n) = g^*(n) + h^*(n).$$

A graph search using an evaluation function is known as the A*-algorithm in which an evaluation function, f, is a measure of f^* expressed as follows:

$$f(n) = g(n) + h(n).$$

In this equation, g and h are measures of $g^*(n)$ and $h^*(n)$, respectively. An algorithm suffices $h(n) \leq h^*(n)$ for all nodes, n, at all times is called the A*-algorithm, it always finds the goal if it exists, and this path is the shortest path (Hart *et al.*, 1968).

When implementing a search for a core collection using the graph search with an evaluation function, f, one may define $g(n)$ as the number of accessions added to E, and $h(n)$ as the number of accessions added to E until the final state, the goal is reached. Then, one may evaluate $\hat{h}(n)$ sufficing $\hat{h}(n) \leq h^*(n)$ as follows.

One may denote a set, $S_V^{A-E} = S_V^A - S_V^E$, from all the sets, $S_{v_1}^{A-E}, S_{v_2}^{A-E}, \ldots, S_{v_m}^{A-E}$ with respect to all variables, $v_1, v_2, \ldots, v_m$ that may find a relative complement, $S_v^{A-E} = S_v^A - S_v^E$, for each variable. Then,

$\hat{h}(n)$ = the maximum number of elements in S_v^{A-E} among the elements, $S_{v_1}^{A-E}, S_{v_2}^{A-E}, \ldots, S_{v_m}^{A-E}$ in S_V^{A-E}.

An accession may not have more than one nominal value per variable so that the number of nodes from a node, n, to the goal, must be equal to or greater than $\hat{h}(n)$. Thus, $\hat{h}(n) \leq h^*(n)$ for all nodes if and only if $\hat{h}(n)$ is defined as above. The graph search using an evaluation function, $\hat{f}(n) = g(n) + \hat{h}(n)$, is an A*-algorithm. This search finds E sufficing $S_V^A = S_V^E$ with the minimum number of accessions if the set, E exists, as shown in Figure 1.

3 IMPLEMENTATION

A core collection obtained using the above search method $h\hat{}(n)$ guarantees the shortest path, but many nodes are expected to expand in this search. Furthermore, the number of accessions in the actual analysis is extremely high and implementation of the above search method cannot assure expected results in the limited time given. Thus, another method was seen as necessary to find an optimal path, close to the shortest path in plausible time, which may not guarantee the shortest path to the goal. In order to implement the new method, the search method was modified.

Considering the search method to find the entry for core collection in the previous section, an element in A was added to E as each node expands. Thus, one will always find the goal as the depth of nodes expands with the number of elements in A. In other words, all nodes lead to the goal. Also within a path, a deeper node is closer to the goal.

With this characteristic in mind, priority was given to $h\hat{}(n)$ of deeper nodes and the comparison of their values. Then, a node with the minimum value was selected and expanded.

One may consider S_v^A a set A of all the accessions as its elements with respect to a variable, v. If $S_v^A = \{d_1, d_2, \ldots, d_k\}$, and another set, $S_{v,t}^A$ with ordered pairs $(d_1, t_1), (d_2, t_2), \ldots, (d_k, t_k)$ as its elements where the first element of each pair is an element of S_v^A and the second element is an integer, t, denoted as,

$S_{v,t}^A = \{(d_1, t_1), (d_2, t_2), \ldots, (d_k, t_k)\}$. In this set, $d_1, d_2, \ldots, d_k$ are defined as items in $S_{v,t}^A$ and $t_1, t_2, \ldots, t_k$ as the 'filled values' of each item. Each ordered pair is a 'diversity cell'.

In particular, $S_{v,t}^A$ is defined as $S_{v,0}^A$when all the filled values, $t_1, t_2, \ldots, t_k$, are 0. That is, $S_{v,0}^A = \{(d_1, 0), (d_2, 0), \ldots\ldots, (d_k, 0)\}$.

Then, we denote a set with elements $S_{v_1,t}^A$, $S_{v_2,t}^A \ldots, S_{v_m,t}^A$ with respect to all the variables, $v_1, v_2, \ldots v_m$ as $S_{V,t}^A$ and a set with elements $S_{v_1,0}^A$, $S_{v_2,0}^A, \ldots, S_{v_m,0}^A$ as $S_{V,0}^A$.

For an accession, a (if $a \in A$), we define $S_{V,t}^A + a$ as follows and express it as 'filling an accession, a, into $S_{V,t}^A$'.

$S_{T,t}^A + a$:

for each v in all the variables of whole accessions

if $(v(a),t) \in S_{v,t}^A, t \leftarrow t + 1(v(a) =$ the value of a variable,v, for an accession, a)

Here, we express $(v(a), t) \in S_{v,t}^A$ as 'filling an item, $v(a)$, in $S_{v,t}^A$'.

The search process is as follows:

(1) Create an $S_{V,t}^A$ sufficing $S_{V,t}^A = S_{V,0}^A$for the set of the whole accessions, A.

(2) Create an empty set, E.

(3) Create a list, N.

for each e (if $e \in A - E$)
$N(e) \leftarrow S_{V,t}^A + e$ ($N(e)$ must be a value of the item, e, in N)

(4) ▷ Calculate $h\hat{}(n)$:

create a list, H.
for each e (if $e \in A - E$)
create a list, NUMBER.
for each v in all the variables of the whole accessions
find the number of ordered pairs sufficing $t = 0$ among every ordered pair, (d, t) and NUMBER$(v) \leftarrow (d, t) \in S_{v,t}^A \in N(e)$
(NUMBER(v) must be a value of the item, v, in NUMBER).

$H(e) \leftarrow$ NUMBER is the maximum value ($H(e)$ must be a value of the item, e, in H).

(5) Select an item, e, with $H(e)$ as its minimum value,

$E \leftarrow E \cup \{e\}$ (if several e's exist, then one e is selected randomly).

$$S_{V,t}^A \leftarrow S_{V,t}^A + e$$

(6) $T \leftarrow 0$

for each $S_{v,t}^A$ (if $S_{v,t}^A \in S_{V,t}^A$)
for each (d, t) (if $(d, t) \in S_{v,t}^A$)
$T \leftarrow T + t$

(7) If $T \neq 0$, then proceed to Step (3).

In this search, Step (3) is a process of expanding children nodes by adding an entry, e, from a parent node and the Step (4) is a process of evaluating the expanded node with an evaluation function, $h\hat{}(n)$.

However, evaluating nodes with $h\hat{}(n)$ above will create several nodes with the same depth minimizing $h\hat{}(n)$ so that a path will be randomly selected. We have modified and improved the method above to evaluate an optimal node with more information instead of by random selection as follows.

One may define the number of filled values sufficing $t = 0$ among every diversity cell, (d_v, t) in $S_{v,t}^A$ (if $S_{v,t}^A \in S_{V,t}^A$) of a node as empty ($S_{V,t}^A$). Selecting a node with an empty value ($S_{V,t}^A$) at its minimum does not guarantee the shortest path, but the empty ($S_{V,t}^A$) value only decreases in the above search process. We have modified the above search to select a node with the minimum empty ($S_{V,t}^A$) value with respect to the goal when several nodes exist with $h\hat{}(n)$ at their minimum.

If several nodes exist with the minimum empty ($S_{V,t}^A$) value, we will select a node to which an accession, e, with less abundant nominal value among accessions in E is added to E. We have defined an added accession to expand a node as e. The value of a variable item, v, in this newly added accession might be expressed as $v(e)$. Thus, $S_{v,t}^A(e)$ now expresses the value of t which suffices $(v(e),t) \in S_{v,t}^A$ $(e \in A)$. If e has variables, $v_1, v_2, \ldots, v_m$, then it may be defined as an overlap.

$$\text{Overlap}\,(S_{V,t}^A, e) = \frac{S_{v_1,t}^A(e) + S_{v_2,t}^A(e) + \ldots + S_{v_m,t}^A(e)}{m}$$

The values of $S_{v_1,t}^A(e)$, $S_{v_2,t}^A(e), \ldots, S_{v_m,t}^A(e)$ increase by one as an accession with the nominal values of $v_1(e)$, $v_2(e), \ldots, v_m(e)$ fill in $S_{V,t}^A$. This overlap ($S_{V,t}^A$, e) can be an indicator of how many repetitive nominal values e, in average, has for each variable in a set, E. In other words, e, on average, has nominal values for each variable unlike other accessions in a set, E, as the value overlap ($S_{V,t}^A$, e) gets smaller. Therefore, a node with the minimum overlap ($S_{V,t}^A$, e) will be selected to take an accession with less abundant values in a set, E.

If several nodes with the minimum overlap ($S^A_{V,t}$, e) value exist, then a node with an accession with higher rarity is selected using predefined values of rarity of accessions in the whole accessions, A. Before executing the above search process, $S^A_{V,t} \leftarrow S^A_{V,t} + a$ must be performed for every a sufficing $a \in A$. Then, lists P and D are created to find values for $P(a) \leftarrow \text{overlap}(S^A_{V,t}, a)$ and $D(a) \leftarrow 1/m \sum_v |S^A_{v,t}(a) - P(a)|$ for every a (if $a \in A$) in advance ($P(a)$ and $D(a)$ are values of an element, a, in lists P and D, respectively).

$P(a)$ can serve as an indicator for the rarity of an accession, a and $D(a)$ indicates the degree of deviation of rarity for each nominal value of a, with respect to the whole accession set, A. The node with the minimum $P(a)$ value will be selected to take an accession with high rarity.

When several nodes with the minimum value of $P(a)$ exist, the node with the highest $D(a)$ value will be chosen. That selects an accession with an exceptionally rare characteristic in a specific trait rather than an accession with evenly distributed rare characteristics in all traits among the accessions with the same $P(a)$ value: the higher the $D(a)$ value, the higher the deviation of rarity of a's nominal value with respect to each variable. Hence, nominal values with high rarity with respect to certain variables are concentrated in such accessions.

The new program's source code is written in Microsoft C# and compiled with Microsoft Visual Studio .NET 2003. The program has been tested in the Microsoft Windows XP environment, and the specifications of the testing computer include a 1.5 GHz Intel mobile processor and a 1 GB RAM.

4 VALIDATION

4.1 Analysis with statistical indicators

Ten sets of 100 virtual accessions were created, each with four nominal variables and three continuous variables as materials for the analysis. Within the PowerCore program, a component divided intervals of continuous variables to nominalize them; the continuous variables in this analysis were automatically classified into different categories based on Sturges' rule (Sturges, 1926).

$$k = 1 + \log_2 n.$$

(n : number of observed accessions)

The search using the PowerCore was heuristic. The core set was generated via this search by calculating the mean difference (MD,%), variance difference (VD,%), coincidence rate (CR,%) and variable rate (VR,%) for continuous variables and computing a frequency distribution for each variable (Hu *et al.*, 2000).

$$\text{MD}(\%) = \frac{1}{m}\sum_{j=1}^{m}\frac{|\text{Me} - \text{Mc}|}{\text{Mc}} \times 100$$

(Me: Mean of entire collection, Mc: Mean of core collection)

$$\text{VD}(\%) = \frac{1}{m}\sum_{j=1}^{m}\frac{|\text{Ve} - \text{Vc}|}{\text{Vc}} \times 100$$

(Ve: Variance of entire collection, Vc: Variance of core collection)

$$\text{CR}(\%) = \frac{1}{m}\sum_{j=1}^{m}\frac{\text{Rc}}{\text{Re}} \times 100$$

(Re: Range of entire collection, Rc: Range of core collection)

$$\text{VR}(\%) = \frac{1}{m}\sum_{j=1}^{m}\frac{\text{CVc}}{\text{CVe}} \times 100$$

(CVe: coefficient of variation of entire collection, CVc: coefficient of variation of core collection, m: number of traits)

4.2 Comparative analysis with a non-heuristic random method to retain whole diversity cells, provided from PowerCore

The basis for generating the core collection using PowerCore is the nominalization of continuous variables. Nominalizing these variables led to the decrease in number of accessions collected in a core collection which was considered necessary in performing the heuristic search through its evaluation function using the given data.

A comparative analysis was performed with the non-heuristic random search wherein no prior information was required for the generation of the core set. The procedure for the random search was as follows:

(1) $S^A_{V,t}$ sufficing $S^A_{V,t} = S^A_{V,0}$ for a set of the whole accessions, A is created.

(2) An empty set, E is created

(3) **for each** v in all the variables of the whole accessions

for each item d in $S^A_{v,t}$
if $S^A_{v,t}(d)$ equals to 0 ($S^A_{v,t}(d)$ must be a filled value of d)
then an element from $e \leftarrow A - E$ is selected to fill d randomly

$$E \leftarrow E \cup \{e\}$$

This random search was performed 10 times to compute the average values of the MD, VD, CR and VR, and frequency distribution.

One hundred virtual accessions were created, each with four nominal variables and three continuous variables for the analysis.

5 RESULTS AND DISCUSSION

5.1 Results of analysis with statistical indicators

The number of accessions, MD, VD, CR and VR values for the core collection are displayed in Table 1. PowerCore selected an average of 11 out of 100 virtual accessions thus reducing the number of accessions by 89% for the entire collection.

MD exhibits the difference in averages of accessions between the core set and the entire collection. MD values in Table 1 show that the mean of the core collections selected by 'PowerCore' is similar to the mean of the entire collection (Table 1).

VD displays the difference in distribution. VD values in Table 1 show that the variance of the core collections selected

Table 1. Average values for core collections using heuristic search

Set	Number of accessions	MD (%)	VD (%)	CR (%)	VR (%)	Coverage (%)
1	13	1.75	33.2	95.0	68.5	100
2	11	7.70	33.7	95.9	71.6	100
3	10	2.85	37.8	93.3	64.0	100
4	10	1.66	28.7	90.3	72.6	100
5	12	3.09	24.3	90.5	77.3	100
6	9	9.25	42.4	88.9	62.1	100
7	10	2.99	42.0	98.3	56.3	100
8	11	4.43	37.7	100	59.6	100
9	12	6.52	32.1	95.7	72.9	100
10	11	2.07	33.4	90.1	65.9	100
Average	11 ± 1.20	4.23 ± 2.68	34.5 ± 5.65	93.8 ± 3.79	67.1 ± 6.68	100 ± 0.00

Mean Difference (MD), Variance Difference (VD), Coincidence Rate (CR) and the Variable Rate (VR).

by 'PowerCore' is rather different from the variance for the entire collection. It was noted that the VD values fluctuated among the different sets.

VR allows a comparison between the coefficient of variation values existing in the core collections and the entire collection and determines how well it is being represented in the core sets. VR values in Table 1 show an average value of 67.1%.

CR indicates whether the distribution ranges of each variable in the core set are well represented when compared to the entire collection. Results obtained (Table 1) show that the average CR value is 93.8%. In order for core collections to represent the whole accessions, some researchers claim that the CR value should be $\geq$80% (Hu *et al.*, 2000).

MD, VD and VR are used to measure the statistical consistency between the core and entire collections. Core collections do not aim for statistical consistency such as average or variation but they seek to cover the genetic diversity of the entire collection. Thus, even well-collected core sets would not show high scores of these statistical indexes based on values attained for average and variation. Moreover, these methods can only be applied to continuous variables.

Particular attention needs to be given to the high CR of core collections as indicated in Table 1. Compared to the other statistical indicators used in this study, PowerCore specifically indicates an exceptionally high CR value for the core sets. Once classification of the continuous variables is performed by PowerCore, the software takes into account all classes, without omission of any of its variables. Thus, PowerCore possesses the capability to cover all the distribution ranges of each class. However, 100% CR value is not attained in Table 1. The reason is that in the case of continuous variables wherein classes are generated, PowerCore would only require the least number of accessions from each class.

In view of the above, we suggest a new indicator, '*Coverage*', which can be used to evaluate a core set for its coverage of variables.

$$\text{Coverage}(\%) = \frac{1}{m}\sum_{j=1}^{m}\frac{\text{Dc}}{\text{De}} \times 100$$

Table 2. Values of variables for core collections using the heuristic and random searches

Search type	Number of accessions	MD(%)	VD(%)	CR(%)	VR(%)
Heuristic	10	5.82	1.45	87.5	96.8
Random	17.1	5.17	4.19	91.7	99.0

Where Dc is number of classes occupied in core collection and De is number of classes occupied in entire accessions in each character and m is the number of variables. The core sets resulted by PowerCore show 100% coverage of variables without any deviations. This suggests PowerCore maintains all the diversity present in each class.

5.2 Results of the comparative analysis with a non-heuristic random method, implemented within PowerCore

The heuristic search selected 10 out of 100 virtual accessions compared to the random search which selected an average of 17.1 accessions. Table 2 shows the MD, VD, CR and VR values obtained using the heuristic search of PowerCore and the random method. The frequency distribution of core collections with respect to each variable is exhibited in Figure 2. The CR value obtained using the random method was slightly higher since more accessions were selected. Heuristic search always resulted in the same value as the number of accessions selected in every try is the same. However, the random search does not provide the same results whenever repeated.

The frequency distribution of core collections with respect to each variable is exhibited in Figure 2. The heuristic method used in PowerCore and the random method are both well illustrated in Figure 2 wherein the core subsets generated contain intervals of values for the whole collection with respect to each variable. Figure 2 also shows that the categorization

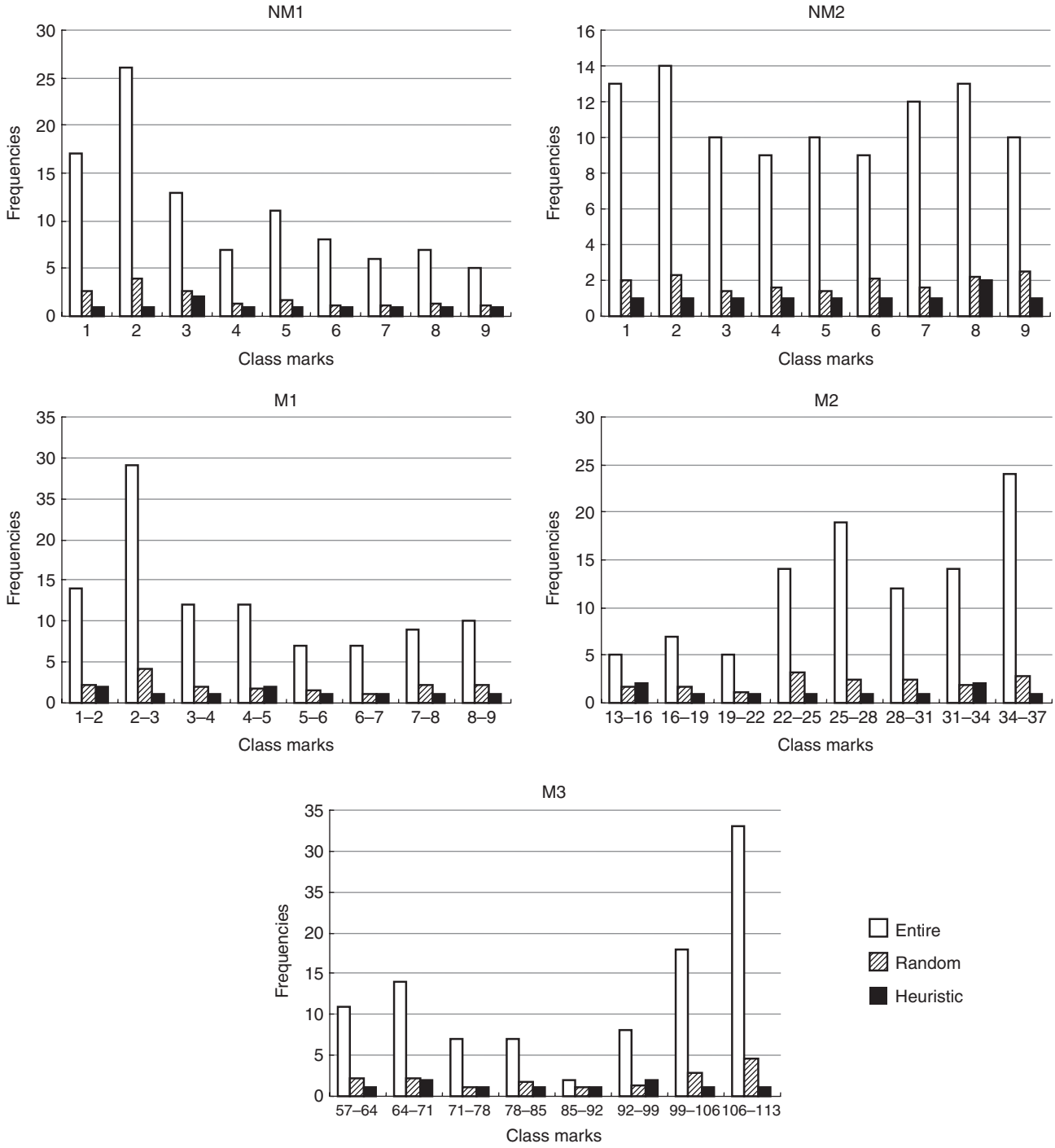

Fig. 2. Frequency distribution of core collections with respect to each variable. (Note: NM1 and NM2 are nominal variables, and M1, M2 and M3 are continuous variables.)

values for each variable of these core collections exhibited extremely low frequency as opposed to the entire collection. If one considers the frequency of each categorization value from the frequency distribution in Figure 2 as accessions with repetitive values, an extremely small or negligible frequency indicates these have been significantly discarded from the core collections.

The heuristic and random searches have greatly reduced the number of accessions, since nominalizing continuous variables in the preparation procedures for establishment of core collections efficiently discards unnecessary accessions. It was noted, however that the heuristic search reduces the size of core collections to $\leq$60% as compared to the random search. The results attained confirms that the modified A*-algorithm of

Table 3. Comparison of the heuristic method with other conventional methods using the two different rice real data sets of 1000 accessions, respectively for phenotype and SSRs

Methods		R-core	P-core	MSTRAT	PowerCore
Phenotype ($n = 1000$[a])	Number of entries	100	100	45	45
	Coverage (%)	75.9	75.4	94.8	100.0
SSR ($n = 1000$[b])	Number of entries	100	100	87	87
	Coverage(%)	46.8	55.0	88.9	100.0

[a]Phenotype data contains 28 qualitative and 11 quantitative traits.
[b]SSR data contains the allele information for 18 loci. R- and P-cores stand for conventional random method and conventional proportional method, respectively at 10% of the number of entries to the entire collection by using clustering method of SPSS 13.0 program (SPSS Inc. 2004) (see the user's manual for the detail procedure used). MSTRAT was run under the default conditions (3 for replicates; 30 for maximum iterations) the software provides using the same number of entries as PowerCore.

PowerCore is more effective than a random search that does not apply the evaluation function for determining the shortest search path.

5.3 Comparison of the heuristic method (PowerCore) with other conventional methods using real rice data sets

To compare the selecting efficiency of PowerCore to Random (R-), Proportional (P-) and MSTRAT methods, two different real rice data sets were used. The phenotype set comprise of 28 quantitative and 11 qualitative traits while the SSR (simple sequence repeat) set includes 18 loci. Both independent sets contain 1000 accessions, respectively. It has been proven that PowerCore has better efficiency than any other conventional methods when the same number of entries was selected in the comparison core sets (Table 3). The core sets developed by PowerCore, retained all different alleles or intervals which two different entire collections possess in both the phenotype and SSR sets of real rice data, ensuring 100% of coverage in developed core sets relative to entire collections. MSTRAT was revealed to be the best method in the coverage rate (94.8% for phenotype and 88.9% for SSRs), compared with the other conventional methods (Table 3).

Basically, PowerCore implements the heuristic algorithm for selecting candidate entries by calculating the costs to reach the goal. So, even if the users repeat the selecting of subsets using the same data, the same list of entries is generated. This is another benefit for users of PowerCore.

6 CONCLUSION

PowerCore is a completely new approach differing from any other previous methodologies, which effectively simplifies the generation process of a core set while significantly cutting down the number of core entries, maintaining 100% of the diversity as categorical variables. For continuous variables, 100% diversity is achieved based on precision of classification. PowerCore is applicable to various types of genomic data including SNPs.

ACKNOWLEDGEMENTS

We thank Drs V. Ramanatha Rao, Prem Mathur, Zongwen Zhang, Xavier Scheldeman and Andrew Jarvis from Bioversity International, and the group of Dr Felipe dela Cruz, University of the Philippines, Los Banos for validating this software using their national plant genetic resources collections (India, China, South America and Philippines), and their valuable comments for improving various options for different users in national genebanks. This study was supported by the National Institute of Agricultural Biotechnology (#NIAB 05-6-11-30-2), the Bio-Green 21 program (Grant code # 20050401034738) of the Rural Development Administration (RDA) and Agricultural Research and Development Promotion Center (ARPC), Republic of Korea.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
Balfourier,F. *et al.* (1998) Comparison of different spatial strategies for sampling a core collection of natural populations of fodder crops. *Genet. Sel. Evol.*, **30** (Suppl. 1), 215–235.
Basigalup,D.H. *et al.* (1995) Development of a core collection for perennial Medicago plant introductions. *Crop Sci.*, **35**, 1163–1168.
Bataillon,T.M. *et al.* (1996) Neutral genetic markers and conservation genetics: simulated germplasm collection. *Genetics*, **144**, 409–417.
Besemer,J. and Borodovsky,M. (2005) GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Res.*, **33**, W451–W454.
Brown,A.H.D. (1989) Core collections: a practical approach to genetic resources management. *Genome*, **31**, 818–824.
Chandra,S. *et al.* (2002) Optimal sampling strategy and core collection size of Andean tetraploid potato based on isozyme data—a simulation study. *Theor. Appl. Genet.*, **104**, 1325–1334.
Franco,J. *et al.* (2006) Sampling strategies for conserving maize diversity when forming core subsets using genetic markers. *Crop Sci.*, **46**, 854–864.
Frankel,O.H. and Brown,A.H.D. (1984) Plant genetic resources today: a critical appraisal. In Holden,J.HW. and Williams,J.T. (eds) *Crop Genetic Resources: Conservation and Evaluation*. Allen and Unwin, Winchester, Massachusetts, USA, pp. 249–257.
Gouesnard,B. *et al.* (2001) MSTRAT: an algorithm for building germplasm core collections by maximizing allelic or phenotypic richness. *J. Hered.*, **92**, 93–94.
Hamilton,R.S. and McNally,K. Unlocking the genetic vault. *Geneflow*, International Plant Genetic Resources Institute, Rome, Italy, p. 29.
Hart,P. *et al.* (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. *Syst. Sci. Cybernet.*, **4**, 100–107.
Hu,J. *et al.* (2000) Methods of constructing core collections by stepwise clustering with three sampling strategies based on the genotypic values of crops. *Theor. Appl. Genet.*, **101**, 264–268.
Karg,R.L. and Thompson,G.L. (1965) A heuristic approach to solving the traveling-Salesman Problem. *Manage. Sci.*, **10**, 225–248.
Latha,R. *et al.* (2004) Allele mining for stress tolerance genes in Oryza species and related germplasm. *Mol. Biotechnol.*, **27**, 101–108.
Marita,J.M. *et al.* (2002) Development of an algorithm identifying maximally diverse core collections. *Genet. Resour. Crop Evol.*, **47**, 515–526.
McKhann,H.I. *et al.* (2004) Nested core collections maximizing genetic diversity in *Arabidopsis thaliana*. *Plant J.*, **38**, 193–202.
Peeters,J.P. and Martinelli,J.A. (1989) Hierarchical cluster analysis as a tool to manage variation in germplasm collections. *Theor. Appl. Genet.*, **78**, 42–48.

Raymond,T.C. (1969) Heuristic algorithm for the traveling-salesman problem. *IBM J. Res. Dev.*, **13**, 400–407.

Schoen,D.J. and Brown,A.H.D. (1993) Conservation of allelic richness in wild crop relatives is aided by assessment of genetic markers. *Proc. Natl Acad. Sci. USA*, **90**, 10623–10627.

Slater,G. St C. and Birney,E. (2005) Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, **6**, 31.

Spagnoletti,Z.P.L. and Qualset,C.O. (1993) Evaluation of five strategies for obtaining a core subset from a large genetic resource collection of durum wheat. *Theor. Appl. Genet.*, **87**, 295–304.

Sturges,H. (1926) The choice of a class-interval. *J. Am. Stat. Assoc.*, **21**, 65–66.

Upadhyaya,H.D. *et al*. (2006) Development of a composite collection for mining germplasm possessing allelic variation for beneficial traits in chickpea. *Plant Genet. Resour.*, **4**, 13–19.

van Hintum,T. et al. (2000) Core collections of plant genetic resources. IPGRI Technical Bulletin No. 3. International Plant Genetic Resources Institute, Rome, Italy.

Wang,C. and Lefkowitz,E.J. (2005) Genomic multiple sequence alignments: refinement using a genetic algorithm. *BMC bioinformatics*, **6**, 200.