Genet. Res. Camb. (1997), **70**, *pp.* 205–213. *With 2 figures.* Printed in the United Kingdom © 1997 Cambridge University Press

Information in molecular profile components evaluated by a Genetic Classifier System: a case study in *Picea abies* Karst.

FEDERICO MATTIA STEFANINI AND ALESSANDRO CAMUSSI*
Genetics Unit, Institute of Silviculture, University of Florence, Florence, Italy

(*Received 3 June 1997 and in revised form 28 July 1997*)

Summary

Individual records from the coding of molecular polymorphism (molecular profiles) are particularly useful for the identification of clones or cultivars, in pedigree analysis, in the estimation of genetic distances and relatedness, and as a tool in genome mapping and population genetics. A parametric statistical analysis of molecular profile components can be infeasible because of the huge number of observed markers, the presence of missing values and the high number of parameters required to evaluate the importance of interactions among markers. Moreover, new powerful molecular techniques make possible the analysis of numerous markers at one time; therefore parametric statistical methods could result in troublesome models with more parameters than data. The field of computer-based techniques offers new strategies to cope with the complexity of molecular profiles. We suggest the use of a Genetic Classifier System to evaluate the importance of profile components. The procedure is based on a Genetic Algorithm approach, a numerical technique that simulates some features of the natural selection process to solve problems. A set of isozyme data from a Norway spruce population is analysed in order to assess their ability to predict the individual plant response to the presence of abiotic stresses. The results, obtained by three different computer simulations, show that this computer-based approach is particularly effective for ranking profile components according to their relevance. Genetic Classifier Systems could also be used as a preliminary step to reduce the complexity of molecular data sets.

1. Introduction

Genetic fingerprinting (Jeffreys *et al.*, 1985), which can distinguish even closely related genotypes, is particularly useful for the identification of clones or cultivars, in pedigree analysis, in the estimation of genetic distances and relatedness in the frame of artificial selection procedures, and as a tool in genome mapping and population genetics. Fingerprinting can be based on the use of polymorphic markers (at protein or DNA level) and in most applications it does not involve the detection of association with QTLs or genes of economic importance.

If individual marker-based genetic profiles are used to assign individuals to a specific population or heterotic group (as, for instance, in the frame of a selection procedure), fingerprinting can be seen as a multivariate discriminant procedure, in which available marker information is used to predict the value of a classificatory variable.

* Corresponding author Istituto di Selvicoltura, via S. Bonaventura, 13 50145 Florence, Italy. tel: +39-55 30231 250. Fax: +39-55-307263. e-mail: gene_agr@cesit1.unifi.it.

The coding of biochemical or molecular polymorphism involves, however, inherent difficulties. One reason lies in the nature of the available genetic profile: for instance the heterozygote at a dominant marker has one of its alleles hidden; therefore a loss of genetic information is expected before a coding scheme for the information profile is chosen. Moreover different marker classes provide genetic information not only of different amounts but also of different quality. In fact most of the available molecular methods do not identify the presence of genes but simply cut the genome into fragments to be compared as a tool of classification, and the relative importance of markers in the assessment of genotyping cost is determined mainly by labour and requirements for supplies (Ragot & Hoisington, 1993). Nowadays, the number of available marker types is growing quickly in most laboratories, making the increase in markers studied less expensive than the increase in sample dimension.

As regards, more particularly, the statistical features of molecular profiles, parametric methods based on linear models can have serious limitations: for instance

https://doi.org/10.1017/S001667239700298X

the presence of multiple phenotypes (or bands in each electrophoretic lane) and the presence of missing data increase the complexity of the model; the increase in the number of available markers dramatically reduces the relative weight (partial determination coefficient) of a single marker; and, finally, the inclusion of many interaction parameters can result in overfitted or saturated models.

For these reasons, it seems of primary importance to develop methods that can help the researcher to evaluate the information contained within each molecular profile component. After ranking the available markers according to the amount of information they provide, the most relevant ones could be included in the individual genetic fingerprint and further analysed to estimate the genetic base of the observed variability of profiles or be used for classification purposes. Inferences based on the chosen subset of markers would be conditioned by the observed data and by the rule specified to select the subset of relevant markers. Therefore, a suitable method to assess the information of molecular profile components is required.

We suggested (Stefanini & Camussi, 1994) the use of Genetic Classifier Systems (GCSs; Holland, 1975, 1986*a*, *b*, 1987) to tackle these issues. GCSs are a wide class of adaptive machine learning algorithms that are theoretically and empirically studied and that have found many interesting applications in different fields such as systems for medical diagnosis, morphogenesis simulations, predictions of company profitability, description of consumer preferences (for more details see Goldberg, 1989*a*, chapter 6) and, more recently, the optimizing of pipe networks (Simpson *et al*., 1994), the simulation of magnetic resonance spectra (Clouser & Jurs, 1995) and optimization of the medium in microbiology (Weusterbotz & Wandrey, 1995). A general description of GCSs has been given by Holland (1986*a*) and more recently by Michalewicz (1994).

The core of a GCS, called Genetic Algorithm, can be considered an algorithmic description of some features typical of the evolutionary process within natural populations. Individual belonging to a population (of hypothetical answers) pass to the next generation (reproduction) depending on their fitness values (dissimilarity between hypothetical answers and the solution). Changes in the population of individuals (coded as strings of characters) are introduced using random point mutation and crossing-over between mated individuals. Therefore, a string-based representation of the problem domain is required together with the specification of a criterion to assess the fitness for each hypothetical answer solution, that is an individual of the simulated population of strings. This simulated evolutionary process moves towards an optimal population in which individuals have the highest possible fitness. Therefore the identified population represents the best answer to the formulated problem (according to the defined fitness function). Moreover, no explicit instruction on how to reach the goal is contained within a GCS because it scores the performance of individuals using a real and known data set built upon experiments in the domain of the problem (i.e. a data set of molecular profiles).

In this paper a set of isozyme data from a Norway spruce population is analysed using a GCS in order to evaluate the information content of molecular profile components from the standpoint of their ability to predict the individual plant response to the presence of abiotic stresses.

The proposed GCS performs a joint evaluation of profile components; therefore it takes into account the presence of the whole set of markers in the molecular profile without the need for parametric modelling. The information content estimated by the GCS can be used to express a preference relation (ranking) on profile components that could be used, together with pragmatic considerations, in the identification of the subset of the whole profile to be subsequently modelled using parametric statistical techniques.

Some specific improvements are also suggested to fulfil the requirements of genetic studies.

2. Materials and methods

The proposed algorithm is derived from Goldberg's GMBL (1989*a*) without the 'bucket brigade' procedure and it is implemented using the Borland TC^{++} compiler for Windows 3.1. In the following paragraph only a short introduction to GCS is provided, while the cited reference explains the subject in depth. A formal description of the reward scheme proposed by the authors is given in the Appendix.

(i) *The case study and the information profile*

Forests show variability in the degree of tolerance to 'new type' damage probably caused by low but chronic levels of pollutants, such as ozone, acid deposition and organic compounds (Ulrich, 1989). One hundred and ninety-seven pairs of Norway spruce (*Picea abies* Karst.) trees were sampled in different locations of the Northern Italian Alps. Plants within each sampling unit were growing within a short distance of each other and were characterized by their reaction to 'new type' damage: one tree was classified as tolerant and the other as susceptible on the basis of the degree of defoliation. This structured sampling is expected to produce a random association between genotype and environment. Individual trees were characterized at 18 isozyme loci, and a subset of 14 polymorphic loci was considered in the study. Details on biochemical techniques and the sampling design are given by Raddi *et al*. (1994).

The coded data set was obtained by assigning a progressive natural number to each genotype at each

https://doi.org/10.1017/S001667239700298X

Table 1. *A key to the adopted coding scheme*

Selected markers	Names	No. of genotypes	String length	Sense strings	Nonsense strings
1	LapA	4	2	00;01;10;11	
2	LapB	10	4	0000;0001;0010;0011;0100;0101; 0110;0111;1000;1001	1010;1011;1100; 1101;1110;1111
3	GotA	3	2	00;01;10	11
4	GotB	5	3	000;001;010;011;100	101;110;111
5	Fest	5	3	000;001;010;011;100	101;110;111
6	PgmA	2	1	0;1	
7	PgmB	5	3	000;001;010;011;100	101;110;111
8	PgiB	3	2	00;01;10	11
9	SkdA	2	1	0;1	
10	SkdB	7	3	000;001;010;011;100;101;110	111
11	IdhA	2	1	0;1	
12	IdhB	2	1	0;1	
13	MnrB	4	2	00;01;10;11	
14	Mnrc	3	2	00;01;10	11
			Total = 30		

For each isozyme locus different genotypes are represented by a binary string of a length depending on the number of possible genotypes. The coding scheme allows a greater number of strings for each locus than are needed to code the genotypes really present in the actual application (*sense* strings). The reward scheme, however, decreases progressively the fitness of *nonsense* strings until they are eliminated from the population.

isozyme locus (starting value is 0), expressed in binary base using the alphabet $\{0, 1\}$ (Table 1, fifth column from the left). Elementary substrings for each locus were joined to obtain a whole string for each individual tree. At the last string value on the right, the coded phenotypic value 0 (tolerant) or 1 (susceptible) was added.

The data set included 394 molecular profiles from the 197 pairs of plants with string length of 31. The studied profile components are isozyme markers that are coded into elementary substrings for each genotype, as described in Table 1.

(ii) *The Genetic Classifier System*

In the present context, any GCS can be seen as a quintuple $\Delta \triangleq \langle X, C, R, O, F\rangle$, where the letters are defined as follows:

X is the matrix of molecular profiles from the known data set, partitioned as $X = [X_i, X_{ij}]$ with X_i the submatrix of marker information and X_{ij} the phenotypic value or the classification tag (in this case 0 to tolerance or 1 for susceptible).

C is the coding system, so that each molecular profile is uniquely mapped in a string equivalent; the map is $X \rightarrow \{0, 1\}^{\ell}$, with ℓ the number of characters (31 in the case study).

R is the matrix of classification rules with the same number of columns as X. Rules can be seen as individuals of a population evolving towards optimality, which is the solution of the task. Each one belongs to the set of possible rules $\{r : r \in \{0, 1, \#\}^{\ell}\}$, therefore they are strings composed of three possible symbols; a rule includes some *don't care symbol* #, which can be considered as a short notation for the sentence '0 or 1', thus they indicate that corresponding characters do not make any contribution to the predictive or classificatory ability of the rule itself; the expanded set of a rule r is obtained by substituting each symbol # with 0 or 1 in the rule r in all the possible combinations.

O is the set of operators that can be used during the learning step of a GCS to cause changes of the R matrix, i.e. it includes an initialization schedule and a procedure that establishes how the reproduction of individuals (rules) is performed.

F is the scoring system constituted by a reward schedule that increases the fitness of a rule performing the right classification task, i.e. it correctly predicts X_{ij} given X_i.

The main device of the GCS procedure is represented by the rule matching. As it is divided into a condition (here the marker information) and an action (here the phenotypic tag), each rule expresses a relation of the type ⟨IF 'condition' THEN 'action'⟩. By the use of a variable number of # symbols in the condition part, a rule can express a relation between a subset of all the possible molecular profiles and the action part (the phenotypic value tag). The relative *specificity* of a rule is the ratio between the length of the specified (by 0 or 1) part and the total length of a rule. Stefanini & Camussi (1994) report an example of some rules and their attributes.

The search for better candidates outside the array of rules of the first generation needs the third component of a GCS: the Genetic Algorithm (GA). The GA is used to generate new candidate rules, while maintaining the old fittest ones. New rules are introduced by means of crossing-over (χ) and point mutation (μ), operators that simulate some features

https://doi.org/10.1017/S001667239700298X

of the corresponding biological phenomena in haploid populations. Details on the adopted GA are described by Stefanini & Camussi (1994), while methodological aspects in applied genetics are under development.

A computer run of a simulation starts with a randomly generated collection of individuals (array of rules) with constant fitness and repeats the following cycle for thousands of generations:

1. A molecular profile is given as input to the GCS and a check is made for each rule to verify whether the condition part matches the message (that is, whether the molecular profile belongs to the expanded set of that rule).
2. Each matched rule makes a 'bid' proportional to its fitness value and the highest bidding rule becomes the winner; its action part defines the predicted phenotype tolerant or susceptible).
3. The algorithm checks whether the prediction of the winner rule agrees with the experimental value recorded in the molecular profile; if the winner rule has made the right prediction (its action part is equal to the phenotypic value contained in the profile given as input), it is rewarded by an increase in fitness value.

After a fixed number of generations, the GA is invoked to explore the space of rules, searching for better individuals. Each rule in the array has a constant fitness value at the start of the computer run, but at the end a peaked fitness landscape is obtained, that is, rules more often rewarded (because they are effective) have the highest fitness value.

A simulation consists of several independent computer runs to take into account the stochastic nature of a GCS (Holland, 1986*a*). Even if the same initialization is given, different results can be obtained; but the amount of variability is partly regulated by the choice of reward scheme and the simulation parameters (see next paragraph).

(iii) *The reward scheme in the fitness function*

We propose a particular reward scheme to minimize the variability among computer runs. The fitness of a rule in the next generation is obtained from the fitness in the current generation by adding the reward obtained if it is the winner rule and by subtracting several terms. The first one is called 'life tax', and it is introduced to minimize the presence of bad, unrewarded rules over generations. The second one is the bid amount that is paid only by the winner. The last term is a tax paid by all the matched rules. Details are given in the Appendix.

The bid made by a rule is obtained as its current fitness value multiplied by the weighted linear summation of its specificity and its *generalized profile of univariate expectation* (Appendix). We included the generalized profile of univariate expectation in the fitness function to tune up the algorithmic performances in the field of applied genetics, by quantifying the amount of molecular profile that contains information regarding the phenotype. It is a weighted linear summation of Pearson's mean squared contingencies obtained for each component of the information profile.

The rationale underlying the definition of the fitness function rests on two opposite tendencies. The first one puts more and more symbols # in a rule, so that it recognizes as many profiles as possible, and the rule has more opportunities to be rewarded. The second one removes symbols # from rules in order to predict the right phenotype for a small number of profiles, eventually only one.

A rule with low specificity (many #) matches many profiles, but if the predicted phenotype is not correct then it is not rewarded. A rule with high specificity matches few profiles; while it is likely that for this small class of profiles it could give the right phenotypic prediction, it is infrequently rewarded due to the small number of profiles that it recognizes.

(iv) *The Genetic Algorithm*

The GA component uses single and double point crossing-over, with probabilities p and p^2 respectively, and it substitutes a small proportion of the whole population of rules at each generation (about 4%). Probabilistic parameters are defined using *the no preferred value* principle. When no reason is given to prefer an outcome, a uniform distribution on the possible alternatives is adopted. An exception is represented by the frequency of crossing-over points that are contained in the condition part of the rules. It is set to 0·85, a value that makes the recombination of the message substring more frequent. The adopted GA uses only substitutions in the mutation operator. A new individual generated by GA has a fitness value equal to the minimum of its two parents.

It must be emphasized that values above described as parameters of the simulation are chosen according to suggestions of some authors (Goldberg, 1989*a*; Michalewicz, 1994) and to heuristic considerations. Further work in this area is required.

(v) *Output analysis*

The collection R_Δ of optimal rules at the end of a computer run is used to build the subset R_X according to the following procedure:

1. The fittest rule is chosen at first, because it has the best performance in the predictive task; profiles that match this rule are removed from the data set.
2. Subsequent rules are chosen according to the decrease in fitness value, and for each rule added to R_X, the profiles it matches are removed from the data set.

https://doi.org/10.1017/S001667239700298X

3. The procedure stops if no more profiles of the data set are recognized or the whole set of rules R_Δ is used.

The collection of rules R_X recognizes the maximum number of molecular profiles using the minimum number of rules; therefore it jointly makes the maximum number of right phenotypic predictions using an association structure summarized by R_X.

Some specific indexes are proposed to resume the observed outcomes after the end of a GCS run:

The *local compression efficiency* obtained by the GCS is defined as:

$$\Psi_\ell = 1 - \frac{M_c}{M - M_n},$$

where M_c is the number of rules contained in R_X, M_n is the number of unrecognized molecular profiles out of the total M, and, consequently, $M - M_n$ is the number of those recognized. Its value is contained in the interval 0–1, with large values preferred to small ones.

The index of *global compression* $\Psi = 1 - (M_c + M_n)/M$ includes the unrecognized M_n profiles in the numerator to establish the original information.

The statistic ϕ_m, the *Information Contribution of a Marker* as determined by R_X, is defined as

$$\phi_m = \frac{\text{card}(\ell_t) \sum_{\ell_m} p_i}{\text{card}(\ell_m) \sum_{\ell_t} p_j}.$$

It is the ratio between the summation of the frequencies of assigned characters 0 and 1 for the substring corresponding to the marker m and the summation of assignment frequencies in the whole string. The weighting terms are respectively the number of characters needed to code marker m, indicated as card (ℓ_m), and the total number of string characters, card (ℓ_t); ℓ_m and ℓ_t indicate the number of characters used to code for marker m and to code the whole set of markers. The rationale underlying the definition of ϕ_m is based on the chosen coding scheme. If a profile component is useless for the predictive task (phenotype), it has many symbols # within R_X; otherwise it has few or no # symbols in it. By using the index ϕ_m, it is possible to rank the profile components on the basis of their information content.

(vi) *Computer simulations and values of parameters*

Three simulations were executed in the spruce case study, each one composed of several computer runs, in order to evaluate the stability of the estimates received from the GCS. This characterization was obtained under the same set of parametric values: 100 generations between GA calls, 30000 generations each computer run, a matrix R of 100 rules, 1·0 initial fitness value, point mutation probability equal to 0·007, crowding factor and crowding subpopulation equal to 3.

The initial rule setting was done by a random drawing of profiles from the known data set and then by putting # symbols in each position with probability equal to 0·8. The derived initial population contains rather non-informative rules (high frequency of #) and it is likely that the few 0 or 1 symbols it contains are in relevant positions.

In the first simulation (S1) the whole set of profiles was used in the learning phase and the computer run repeated 350 times. The second (S2) and the third (S3) simulations were planned to verify the classification ability of the system. The known data set of molecular profiles was split into two subsets: one was used in the learning phase, and the second was used to verify the ability of R_X to predict the right phenotype for those profiles that were not used to obtain R_X. Each simulation consisted of 175 computer runs. Data set splitting was performed once in the S2 simulations, while it was randomly defined at the beginning of each computer run in the S3 simulation.

Also in this case, values of parameters for each simulation were chosen according to suggestions of some authors (Goldberg, 1989*a*; Michalewicz, 1994) and to heuristic considerations. At present, no general rule to define values for parameters is known.

3. Results

Several computer runs within the three simulations allowed us to evaluate the ability of the proposed Genetic Classifier System to assign the right phenotype to each tree from its marker profile. The overall ability within simulations and the possibility of cross-validating the results starting from different learning data sets are summarized by appropriate statistics.

The main results of the three simulations are reported, as box plots, in Fig. 1. All three simulations showed a quite similar distribution of recognized profiles, as shown by the values received by the $M - M_n$ statistic, even if the use of the whole data set for the S1 learning phase and the random splitting in S3 could have produced more similar distributive shapes in comparison with S2 results. The simulation S2, based on a fixed splitting, had perhaps penalized the outcomes. Similar considerations can be made as regards the cardinality of R_X.

Results in Fig. 1 show that the distribution of values is quite concentrated around the median – an important feature required by applied domains. More work is required to check whether this feature is also present in the analysis of other data sets, or whether it can be improved by further parameter tuning. For the analysed case study, the algorithm seems fairly insensitive to the initial conditions. Furthermore, the S2 and S3 simulations support the hypothesis of robustness for the proposed algorithm that produces similar distributions even if half the profiles are not used to find the optimized set of rules. An alternative

https://doi.org/10.1017/S001667239700298X

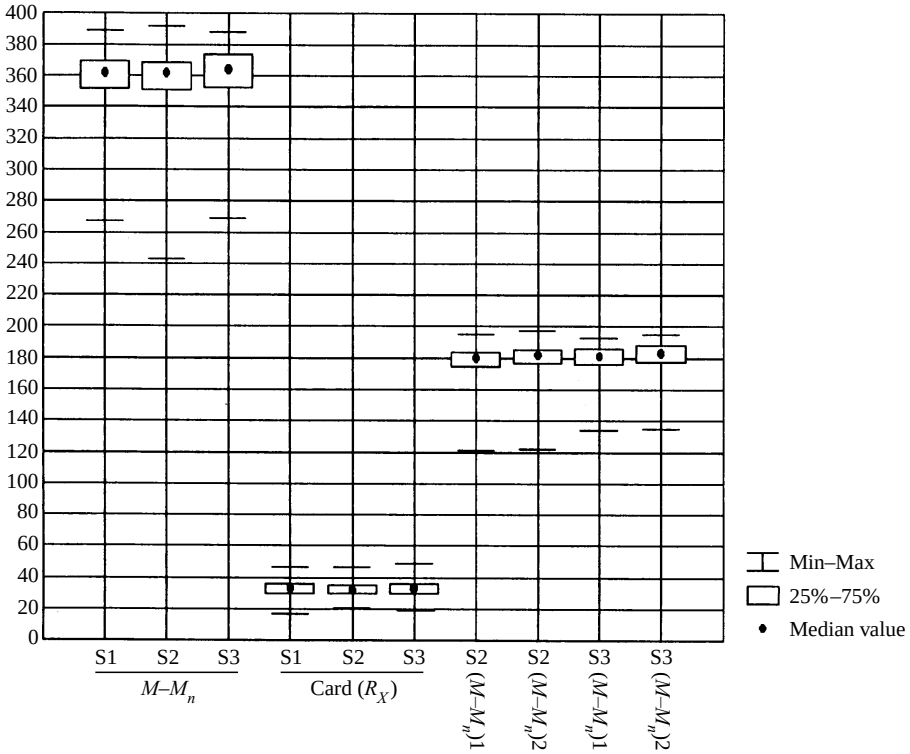

Fig. 1. Box plots of main results of the S1, S2 and S3 simulations. $M-M_n$ is the number of recognized profiles, Card (R_X) is the cardinality of R_X, $(M-M_n)1$ and $(M-M_n)2$ are the number of recognized profiles in the used and unused halves of the data set respectively. The first three variables on the left have an upper bound of 394, the last three an upper bound of 196, and the remainder of 100.

explanation of this result could be that the algorithm does not use the evidence in the data set. But it is unlikely that similar results are due only to the effect of randomness, as it results from the predictive performances at an early stage in a computer run.

The descriptive statistics from the simulations are reported in Table 2: the cardinality of R_X, the number of recognized profiles and the global and local compression efficiencies, are as described in Section 2. The high similarity of results, particularly as regards the number of recognized profiles and the cardinality of R_X, suggests that our GCS is able to exploit all the information included in the data set. Moreover if the halved data set can be considered homogeneous in its components from the information point of view, then the algorithmic performances are not strictly dependent on the availability of the whole data set.

In the simulation S2, while 50% of the computer runs had a number of recognized profiles near 185 in both the halved data sets, the 50% of runs globally produced more than 350 recognized profiles. The cardinality of R_X is 30–40 rules in about 50% of runs. The total number of recognized profiles showed a range of 362–392 profiles out of 394, with a mean value of 358 profiles. The 95% interval comprises 325–380 profiles. These are good values, confirmed by a high level of compression, which is always over 80%. The odds ratio (OR) was calculated from two-way contingency tables (recognized, unrecognized by used half data set, unused half data set). They received values near 1, demonstrating that the number of recognized profiles is independent of their belonging to the half data set used during the learning step. As the single run can be considered a Monte Carlo simulation, the null hypothesis H°: $OR = 1$ was accepted in both S2 and S3 simulations, on the basis of the 5% to 95% percentile interval as reported in Table 2. S3 and S2 simulations showed similar numerical results, particularly as regards the number of recognized profiles within subdivided data sets.

These results suggest an effective independence of the performance of the GCS from the choice of data set subdivision during the initialization procedure. A null hypothesis of equality of the two distributions in respect of the whole frequency of recognized profiles was not rejected by a Kolmogorov–Smirnov test (data not reported). Accordingly, S2 and S3 simulations were considered jointly and compared with S1 as regards the frequency of recognized profiles. Also this Kolmogorov–Smirnov test resulted in no rejection of the null hypothesis (data not reported).

The whole collection of 700 computer runs was screened to identify the best run, in which the criterion $\frac{3}{4}\text{card}\,(R_X)+\frac{1}{4}M_n$ is minimized. The weights in the formula above are chosen so that a decrease of one unit in M_n is three times less important than a decrease of one unit in card (R_X).

The best computer run belongs to the S1 simulation (file 280: card $(R_X) = 17$, $M_n = 28$, $\Psi = 0{\cdot}88$, $\Psi_l = 0{\cdot}95$). The frequencies of digit assignment within the

https://doi.org/10.1017/S001667239700298X

Table 2. *Main descriptive statistics from the simulations*

Simulation/ Variable	Mean	50th perc.	Min.	Max.	5th perc.	95th perc.	SD	Variance
S1								
$M-M_n$	357·92	362	267	389	327	378	17·37	301·83
Card (R_X)	32·70	33	17	47	25	41	5·01	25·12
Ψ	0·8254	0·8325	0·5990	0·9117	0·7462	0·8782	0·0438	0·0019
Ψ_l	0·9085	0·9083	0·8708	0·9535	0·8864	0·9303	0·0140	0·0002
S2								
$(M-M_n)1$	178·13	180	121	195	162	189	8·96	80·35
$(M-M_n)2$	180·04	182	122	197	163	191	9·03	81·52
$(M-M_n)T$	358·17	362	243	392	325	380	18·00	323·67
Card (R_X)	32·07	32	21	47	24	41	5·13	26·281
Ψ	0·8276	0·8350	0·5584	0·9162	0·7462	0·8832	0·0434	0·0019
Ψ_l	0·9104	0·9106	0·8750	0·9375	0·8851	0·9320	0·0136	0·0002
OR	0·9944	0·9890	0·9871	1·1191	0·9879	1·0441	0·0202	0·0004
S3								
$(M-M_n)1$	178·91	181	134	193	161	190	9·95	99·07
$(M-M_n)2$	180·82	183	135	195	162	192	10·05	100·92
$(M-M_n)T$	359·72	364	269	388	323	382	20·00	399·87
Card (R_X)	32·68	33	20	49	25	41	5·07	25·74
Ψ	0·8301	0·8426	0·5939	0·9036	0·7437	0·8883	0·0502	0·0025
Ψ_l	0·9090	0·9099	0·8665	0·9432	0·8822	0·9329	0·0144	0·0002
OR	0·9963	0·9892	0·9860	1·20	0·9880	1·05	0·0260	0·0007

Card (R_X) is the cardinality of R_X; $M-M_n$ is the number of recognized profiles; $(M-M_n)1$, $(M-M_n)2$ and $(M-M_n)T$ are the number of recognized profiles in the used and unused halves of the data set and their summation, respectively. Ψ and Ψ_l are the global and the local compression efficiency. Within the S1 simulation, 350 runs are considered. Within simulation S2 and S3, statistics are evaluated from a sample of 175 computer runs.
perc., percentile. OR, Odds Ratio.

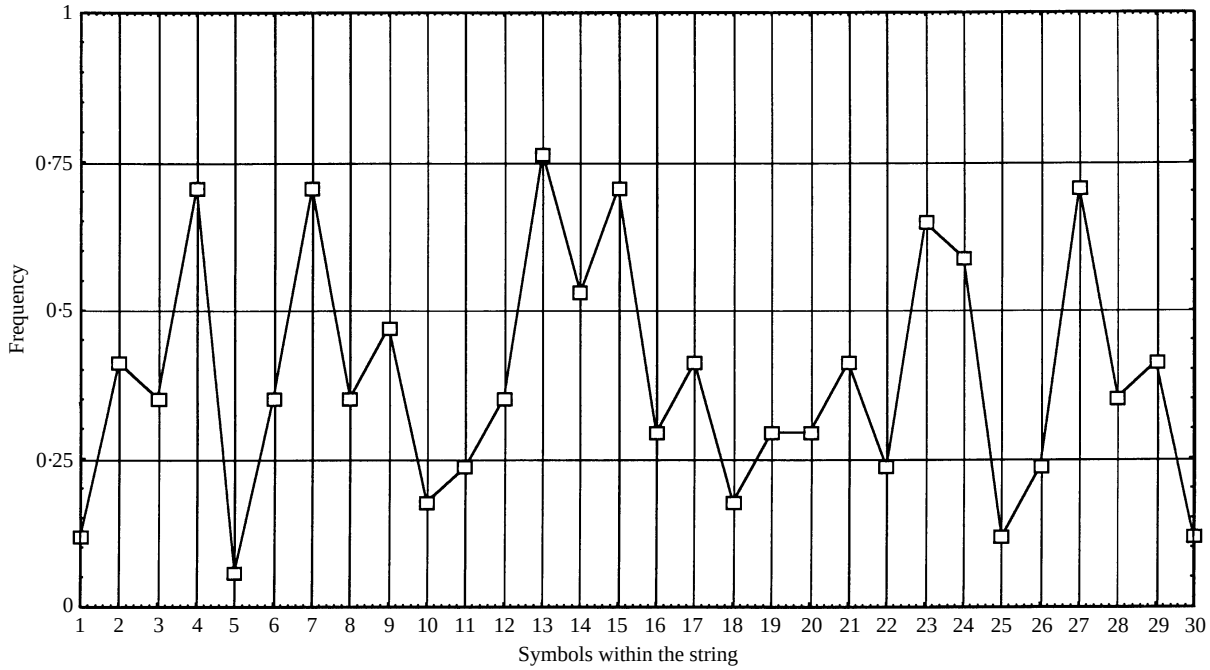

Fig. 2. Frequencies of symbol (digit) assignment with R_X for a selected computer run. No symbol has null frequency, suggesting that it is a necessary condition to extract the maximal amount of information from the data set.

best run are reported as a frequency profile in Fig. 2. No digit had null frequency, suggesting that this is a necessary condition to extract the maximum amount of information from the data set. The information of molecular profile components is shown in Table 3, where they are sorted according to the received ϕ_m values. It is evident that in practical applications the useful markers to classify individual phenotypes are to be chosen within the highest scored as regards the ϕ_m values.

https://doi.org/10.1017/S001667239700298X

Table 3. *The relevance of single markers is shown by means of an ordinal ranking according to ϕ_m, the information in molecular profile component m evaluated using the best computer run (see Section 3)*

Marker	ϕ_m	Order	Marker	ϕ_m	Order
M1	0·685	9	M6	1·827	1
M2	0·952	7	M5	1·421	2
M3	1·371	3	M3	1·371	3
M4	0·761	8	M13	1·370	4
M5	1·421	2	M10	1·269	5
M6	1·827	1	M9	1·066	6
M7	0·761	8	M2	0·952	7
M8	0·761	8	M4	0·761	8
M9	1·066	6	M7	0·761	8
M10	1·269	5	M8	0·761	8
M11	0·305	11	M1	0·685	9
M12	0·609	10	M14	0·685	9
M13	1·370	4	M12	0·609	10
M14	0·685	9	M11	0·305	11

In the second block of columns, the first column indicates the marker number, the second column shows ϕ_m values, and the third indicates ordinal ranking. In the first block of columns the same information is reported but the original marker order is conserved.

4. Discussion

In this paper, the proposed GCS was able to assess the information in molecular profile components without the need for complex parametric models and computations involving algebraic manipulation (i.e. differentiation and integration). The underlying rationale rests on the chosen coding scheme and on the definition of ϕ_m: a profile component has no information if it is useless for the predictive task (phenotype), that is if it has only symbols # with R_X. Moreover, it has the maximum amount of information if no symbol # is present in the correspondent substring within R_X.

The results obtained in this case study are preliminary, because more work is required to find a general procedure for choosing simulation parameters, but they are also interesting for the new trends present in applied genetics: the necessity to include a huge amount of information in the same analysis makes permanent the need for a general method that uses simple automated computations in preliminary steps of the analysis.

According to the *fundamental theorem of Genetic Algorithms* (Goldberg, 1989*a*), a GA is effective if regularities exist in the real data set of molecular profiles and if they are expressed in substrings of short length, so that its main optimizing operator (the crossing-over) can usefully work.

Moreover, a different set of GCS parameters causes different performances, even if the computational routines remain unchanged, so the main problem is not completely solved. While some general advice on the choice of parameters can be formulated (Goldberg, 1989*a*), the final decision rests more on intuitive-artistic feeling than on numerical criteria, because of the high non-linearity of the GCS dynamic. The same comments can be made about the structure of the algorithm. However, the choice of the computational scheme can be inspired by a *simplicity principle* that introduces only well-motivated procedures by heuristic reasoning based on computer simulation, while the study of optimal parameter identification could adopt standard statistical techniques in future.

Before this approach can be applied widely, sampling variation must be taken into account and a general rule to choose the length of a computer run should be formulated. In this paper, some stability was observed even if a reduced amount of information was furnished during the learning phase, but a refined study of these features should be performed. Nevertheless, our preliminary results support the idea of the GCS method for discovering and exploiting information, perhaps even if huge data sets with sparse missing values are considered. Moreover, the selection of a subset of profile components to make further parametric analysis should include both the information found using the proposed GCS and pragmatic considerations coming from the field of applications.

We proposed the use of a term called the generalized profile of univariate expectation to restrict the algorithmic dynamic so that no unsound conclusions from the standpoint of applied genetics can result. This choice is rather conservative of the opinion built on univariate statistics, but a deeper study is required to prove that no interesting solution is lost by including this term during the fitness evaluation.

Finally, according to Goldberg (1989*b*), the use of our reward scheme could be criticized because it forces the dynamic of the GCS by restricting the space of searches. However, even in the worst case, when the solution of a similar GCS is totally different from that of our GCS, it would be hard to believe that it has a sound biological meaning from only heuristic considerations. A further look at these aspects is in progress.

Appendix

Some computational details are shown here regarding the reward scheme of the adopted GCS, which is formulated to minimize the variability among simulations. The difference equation of fitness changes is:

$$S(i,t+1) = S(i,t) - P(i,t) - T(i,t) - Q + R(i,t),$$

$$P(i,t) = B(i,t) I_{\{\text{`winner'}\}}(i),$$

$$T(i,t) = C_t\, S(i,t) I_{\{\text{`matched'}\}}(i),$$

$$R(i,t) = C_r\, I_{\{\text{`winner'} \cap \text{`right'}\}}(i),$$

where $S(i,t+1)$ is the fitness of rule i at the next generation $t+1$, Q is the *life tax*, $P(i,t)$ is the bid amount paid only by the winner, $T(i,t)$ is a *tax* paid by all the matched rules, $R(i,t)$ is the reward term

https://doi.org/10.1017/S001667239700298X

collected only by the winner, $I(i)$ is the characteristic function, and the remaining quantities are set as follows:

The bid $B(i,t)$ made by a rule i at cycle t is

$$B(i,t) = C_b\, S(i,t)\,[C_1 + C_2\, C(i,t) + C_3\, U(i,t)],$$

where $S(i,t)$ indicates the fitness of rule i at generation t, $C(i,t)$ is the specificity and the other terms are constants. We choose constant values as follows: $C_b = 5\times10^{-6}$, $C_t = 1\times10^{-6}$, $C_r = 1\times10^{-4}$, $Q = 1\times10^{-6}$, $C_1 = 0{\cdot}01$, $C_2 = 1{\cdot}0$, $C_3 = 2{\cdot}0$.

$U(i,t)$ is the *generalized profile of univariate expectation*, a weighted linear summation of Pearson's mean squared contingencies obtained for each binary symbol within the profile component. At generation t, a generic rule i has a generalized profile of univariate expectation equal to

$$U(i,t) = \frac{\sum u_j\, I_{\{1,0\}}(x)}{m},$$

with j the position of the considered symbol in the condition part of rule i, m the total number of those symbols, and I the characteristic function taking into account only symbol values that differ from the *don't care* #. $U(i,t)$ is bounded in $[0,1]$. The values u_j are obtained on the basis of the original information coding. A k by 2 contingency table is built counting for k genotypes at locus g, and for the correspondent phenotype (two classes of new type damage in the present application). The Pearson's mean squared contingency coefficient is derived for each locus as

$$u_i(t) = w_g \frac{\left(\sum_{k,r} \frac{p_{k,r}}{p_{k,\bullet}\, p_{\bullet,r}}\right) - 1}{n_k - 1},$$

where p_{kr} is the relative frequency of profiles with genotype k at locus g and phenotype r, while n_k is the number of genotypes at locus g and w_g is the inverse of the number of symbols necessary to code the marker information. The point notation (e.g. $p_{k,\bullet}$ and $p_{\bullet,r}$) refers to marginal frequency values.

The work was supported by a grant from the Italian Ministry for Agriculture (MIRAAF), in the framework of the National Project 'Biotecnologie vegetali. area 3'.

References

Clouser, D. L. & Jurs, P. C. (1995). Simulation of the C-13 nuclear magnetic resonance spectra of trisaccharides using multiple linear regression and neural network. *Carbohydrate Research* **271**, 67–77.

De Jong, K. A. (1975). An analysis of the behaviour of a class of genetic adaptive systems. (Doctoral dissertation, University of Michigan.) *Dissertation Abstracts International*, Michigan, 5140B, 36(10).

Goldberg, D. E. (1989*a*). *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison Wesley.

Goldberg, D. E. (1989*b*). Zen and the art of genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms* (ed. J. D. Schaffer). San Mateo, CA: Morgan Kaufmann.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Holland, J. H. (1986*a*). A mathematical framework for studying learning in classifier systems. *Physica* **22D**, 307–317.

Holland, J. H. (1986*b*). Escaping brittleness: the possibility of general purpose learning algorithms applied to parallel rule based systems. In *Machine Learning II* (ed. R. S. Michalski, J. G. Carbonell & T. M. Mitchell), pp. 595–623. Los Altos, CA: Morgan Kaufmann.

Holland, J. H. (1987). Genetic algorithms and classifier systems: foundations and future directions. In *Genetic Algorithms and Their Applications. Proceedings of the Second International Conference on Genetic Algorithms* (ed. J. J. Grefenstette), pp. 82–89. Hove: Lawrence Erlbaum.

Jeffreys, A. J., Wilson, V. & Thein, S. L. (1985) Individual-specific "fingerprints" of human DNA. *Nature* **316**, 76–79.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer-Verlag.

Raddi, S., Giannini, R., Stefanini, F. M. & Camussi, A. (1994). Forest decline index and genetic variability in *Picea abies* (L. Karst.). *Forest Genetics* **1** (1), 33–40.

Ragot, M. & Hoisington, D. A. (1993). Molecular markers for plant breeding: comparison of RFLP and RAPD genotyping costs. *Theoretical and Applied Genetics* **86**, 975–984.

Siegel, S. & Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill.

Simpson, A. R., Dandy, G. C. & Murphy, L. J. (1994). Genetic Algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management. ASCE* **120**, 423–443.

Stefanini, F. M. & Camussi, A. (1993). APLOGEN: an object oriented Genetic Algorithm performing Monte Carlo optimization. *Computer Applications in Biological Sciences* **9**, 695–700.

Stefanini, F. M. & Camussi, A. (1994). Estimates of relationships between quantitative traits and molecular markers by means of Genetic Classifier Systems. In *Biometrics in Plant Breeding: Application of Molecular Markers* (ed. J. W. van Oojen & J. Jansen), pp. 178–185. Wageningen, The Netherlands: DLO Centre for Plant Breeding and Reproduction Research.

Ulrich, B. (1989). Effect of acidic precipitation on forest ecosystems in Europe. In *Acidic Precipitation*, vol. 2 (ed. D. C. Adriano & A. H. Jonson), pp. 189–272. Berlin: Springer-Verlag.

Weusterbotz, D. & Wandrey, C. (1995). Medium optimization by genetic algorithm for continuous production of formate dehydrogenase. *Process Biochemistry* **30**, 563–571.