SIMULATION OF GENETIC SYSTEMS BY AUTOMATIC DIGITAL COMPUTERS

I. INTRODUCTION

By A. S. FRASER*

[*Manuscript received June* 26, 1957]

Summary

Methods of setting automatic digital computers to simulate the algebraic aspects of reproduction, segregation, and selection are discussed. The application of these methods to the problem of the importance of linkage in multifactorial inheritance is illustrated by results from the SILLIAC.

I. INTRODUCTION

In recent years a field of mathematics has become prominent which is based on the simulation of stochastic processes. This field has been termed the Monte Carlo method and its prominence can be directly attributed to the introduction of automatic electronic digital computers. The Monte Carlo method involves in most of its applications several hundreds of thousands of arithmetic steps and therefore would be impractical without the speed of automatic computers.

The majority of genetic problems depend for their resolution on the algebra of repetitive sequences, and it is relatively easy to apply the Monte Carlo method to these sequences. The general problem of applying the Monte Carlo method to genetic problems using an automatic digital computer (the SILLIAC) is discussed in this paper.

II. BINARY REPRESENTATION OF GENETIC FORMULA

If we consider two alleles, $+^a$ and a, at a locus, the genotypes are represented as $+^a+^a$, $+^a a$, and aa respectively, and the gametes as $+^a$ and a. In this system $+$ and *not-plus* specify the two alleles, a specifies the locus. The symbols 1 and 0 can

TABLE 1

NORMAL AND BINARY REPRESENTATION OF TWO HAPLOID GENOTYPES

Normal	$+^a$	b	c	d	$+^e$	f	g	$+^h$	$+^i$	$+^j$	$+^k$	l	m
Binary	1	0	0	0	1	0	0	1	1	1	1	0	0
Normal	$+^a$	$+^b$	$+^c$	$+^d$	$+^e$	f	g	h	i	j	k	$+^l$	$+^m$
Binary	1	1	1	1	1	0	0	0	0	0	0	1	1

be substituted for $+$ and *not-plus*, and the position of the symbols in a register can specify the locus. This is illustrated in Table 1 for two haploid genotypes.

Since it is possible to manipulate the individual digits of a register, this binary arithmetical representation of genetic formulae is the first step in simulating genetic systems.

*Animal Genetics Section, C.S.I.R.O., University of Sydney.

III. "Logical" Algebra

A special aspect of the circuits of the ILLIAC family of computers is the ability to perform the operations of "logical" algebra. These are illustrated below:

0 1 1 & 0 0 1 = 0 0 1	Logical product
0 1 1 $\equiv$ 0 0 1 = 0 1 0	Logical equivalent
0 1 1 $\wedge$ 0 0 1 = 1 0 0	Logical not-sum

These operations can be used to allow identification of the genetic nature of an individual at each locus, i.e. whether the individual is $+^i+^i$, $+^i i$, or ii at a locus i. Given two haploid genotypes A (paternal) and B (maternal), then

A & B identifies the $++$ loci,
$A \equiv B$ identifies the $+i$ loci,
$A \wedge B$ identifies the ii loci.

This is illustrated for a diploid genotype of 10 loci:

A (paternal genotype)	1 0 1 1 0 0 0 1 1 0
B (maternal genotype)	1 1 0 1 1 0 0 1 0 1
A & B	= 1 0 0 1 0 0 0 1 0 0
$A \equiv B$	= 0 1 1 0 1 0 0 0 1 1
$A \wedge B$	= 0 0 0 0 0 1 1 0 0 0

The importance of these transformations is most evident in the determination of the phenotypic value corresponding to a specific genotype.

IV. Determination of Phenotypic Value

If the contribution of the homozygous recessive loci to the phenotype is taken as zero it is only necessary to specify the phenotypic component of the homozygous dominant loci and the dominance term. If a is the phenotypic component and h is the dominance term, then the phenotypic contribution of a locus is 0, ah, or a, depending on whether it is 0/0, 0/1, or 1/1.

In the general case each locus may have a different phenotypic component and dominance term, and the vectors $\{a_i\}$ and $\{h_i\}$ need to be specified. If these be considered as diagonal matrices, and the logical product and logical sum of A and B be also considered as diagonal matrices (where A is the maternal and B the paternal genotype), then the phenotype of the individual is given by

$$\text{diag } A \text{ \& } B \,.\, \text{diag } \{a_i\} + \text{diag } A \equiv B \,.\, \text{diag } \{a_i\} \,.\, \text{diag } \{h_i\} = \text{diag } [P_i],$$

and

$$P_{AB} = \Sigma p.$$

This is illustrated for a genotype of four loci, in which $\{a_i\}$ is {1, 2, 3, 4} and $\{h_i\}$ is {1·0, 0·5, 0·25, 0·0}. Using binary notation, A = 1 0 1 0, and B = 1 1 0 0. Then diag A & B . diag $\{a_i\}$ =

$$\begin{bmatrix} 1 & . & . & . \\ . & 0 & . & . \\ . & . & 0 & . \\ . & . & . & 0 \end{bmatrix} \begin{bmatrix} 1 & . & . & . \\ . & 2 & . & . \\ . & . & 3 & . \\ . & . & . & 4 \end{bmatrix} = \begin{bmatrix} 1 & . & . & . \\ . & 0 & . & . \\ . & . & 0 & . \\ . & . & . & 0 \end{bmatrix} = C,$$

and diag $A \equiv B$. diag $\{a_i\}$. diag $\{h_i\} =$

$$\begin{bmatrix} 0 & . & . & . \\ . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 0 \end{bmatrix} \begin{bmatrix} 1 & . & . & . \\ . & 2 & . & . \\ . & . & 3 & . \\ . & . & . & 4 \end{bmatrix} \begin{bmatrix} 1{\cdot}0 & . & . & . \\ . & 0{\cdot}5 & . & . \\ . & . & 0{\cdot}25 & \cdot \\ . & . & . & 0{\cdot}0 \end{bmatrix} = \begin{bmatrix} 0 & . & . & . \\ . & 1 & . & . \\ . & . & 0{\cdot}75 & \cdot \\ . & . & . & 0 \end{bmatrix} = D,$$

and

$$C + D = \begin{bmatrix} 1 & . & . & . \\ . & 1 & . & . \\ . & . & 0{\cdot}75 & \cdot \\ . & . & . & 0 \end{bmatrix} = [P_i],$$

from which

$$P_{AB} = \Sigma p = 2{\cdot}75.$$

The steps of this sequence can easily be programmed and the required space in the memory is small: two sets of n registers where n is the number of loci.

V. Inter-locus Interactions

It is necessary for the completely general case to include inter-locus interactions. This can be done by specifying three matrices of order n, each matrix specifying the occurrence and order of interactions for each genetic state, i.e. separate matrices are required for the 0/0, 0/1, and 1/1 states. The space required to store these matrices would severely restrict the usefulness of a programme except for very few loci. This restriction can be avoided by restricting the number of degrees of interaction to a reasonable number, say 3. Here the required storage space would be 27 registers for the three matrices, and $2n^2$ digits for storage of the matrix specifying whether interaction occurs, and if so, of what type.

In the following matrix the rows specify the locus whose phenotype is modified by interaction, and the columns specify the locus performing the interaction. The elements of the matrix are modulo$_4$, i.e. 0, 1, 2, or 3, and specify, if 0, that no interaction occurs, if 1, 2, 3, that interaction does occur and is of type 1, 2, or 3. If n is 20, then only 20 registers are required to store this matrix which is a minor demand on the memory space:

$$\begin{bmatrix} b_{00} & . & . & . & . & b_{0n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ b_{n0} & . & . & . & . & b_{nn} \end{bmatrix}_{\text{modulo}_4}$$

These sequences produce a single-valued phenotype. It is possible, by paralleling to produce a multi-valued phenotype, thus allowing simulation of genetic systems involving several characters.

VI. Environmental Effects

The above method of determining the phenotype corresponding to a specific genotype does not include any effect of non-genetic factors. Since the majority of the unsolved problems of mathematical genetics occur in systems with environmental modification of the phenotype it is necessary to simulate the occurrence of an environmental component of the phenotype which is independent of the genotype.

This can be accomplished by specifying a function $r = \mathrm{f}(x)$ such that if r is a random number in the range 0 to 1, then x is a random normal deviate also in the range -1 to $+1$. Hastings (1955) has devised several functions which, using linear combinations of r, produce values corresponding closely to random normal deviates.

Given that P_i is the potential phenotype of the ith genotype, then by generating a random number, r_i, and finding x_i, the transformation of the potential phenotype into the actual phenotype is given by

$$[P_i]_{\text{actual}} = P_i \pm x_i \,.\, P_i.$$

It is clearly possible, by specifying different forms of $r = \mathrm{f}(x)$, to simulate any degree of environmental modification of the potential phenotype. It is also possible to specify relations between the genotype and $r = \mathrm{f}(x)$, i.e. to simulate genetic control of environmental stability. This is likely to be an important feature of programmes designed to examine systems showing "homeostasis" (Lerner 1954).

VII. Segregation

Gametes produced by an individual heterozygous for a single locus are of two types, both occurring with equal probability: e.g. the gametes produced by the heterozygote 0/1 are of the types 0 and 1 with equal probability. This process can be simulated by generating a random number, r_i, which lies in the range

$$0 \leqslant r_i \leqslant 1.$$

If this number is tested against 0·5, then, in a set of q random numbers, the occurrence of tests which exceed 0·5 will have the same probability as those which are less than 0·5, allowing for slight bias introduced by the accuracy of the random number. If the random numbers have an accuracy of 1×2^{-38}, as is usual in the SILLIAC, then any bias of this system is negligible.

This method can be used to simulate a system of n genetically independent loci. Given that A and B represent the paternal and maternal genotypes of an individual as before, then the following sequence will simulate the production of a gamete by such an individual.

Form A & B and $A \equiv B$. Operate on $A \equiv B$ in digital sequence, generating a random number, and testing it against 0·5 (as above) for each digital position which contains 1. The result is that $A \equiv B$ will be transformed into a term

$<A \equiv B>$ in which the digital positions containing 0 are unaffected, whereas those containing 1 are left alone or changed to 0 with equal probabilities which are independent between positions. If we represent the operation of the random transform by $\bar{R}$, then this is expressed by

$$\bar{R} : A \equiv B \rightarrow <A \equiv B>.$$

Then $A \,\&\, B + <A \equiv B>$ gives a number whose digital conformation is such that (i) it has 0 wherever the AB configuration was 0/0, (ii) it has 1 wherever the AB configuration was 1/1, and (iii) it has 1 or 0 at equal probability wherever the AB configuration was 0/1 or 1/0.

VIII. Recombination

The simulation of recombination can be accomplished given the vector of frequencies of recombinant and non-recombinant classes. The vector is illustrated below for the gametes produced by an individual heterozygous at three loci, in coupling, where r_1 is the recombination between the first and second loci, and r_2 is the recombination between the second and third loci.

	Vector of Types of Gametes	Frequencies
$\frac{1\ 1\ 1}{0\ 0\ 0}$ produces (individual's genotype)	0 0 0	$\frac{1}{2}(1-r_1)(1-r_2) = f_{000}$
	0 0 1	$\frac{1}{2}(1-r_1)\,r_2 = f_{001}$
	0 1 0	$\frac{1}{2}\,r_1\,r_2 = f_{010}$
	0 1 1	$\frac{1}{2}\,r_1(1-r_2) = f_{011}$
	1 0 0	$\frac{1}{2}\,r_1(1-r_2) = f_{100}$
	1 0 1	$\frac{1}{2}\,r_1\,r_2 = f_{101}$
	1 1 0	$\frac{1}{2}(1-r_1)\,r_2 = f_{110}$
	1 1 1	$\frac{1}{2}(1-r_1)(1-r_2) = f_{111}$

This illustration has been given for a triple heterozygote, but $\{f_i\}$, the vector of frequencies of recombinants and non-recombinants, is not restricted to a particular genotype. The same distribution of frequencies of recombinants and non-recombinants occurs for all genotypes. The effect of considering a different genotype is to change the vector of types of gametes. This is illustrated for three genotypes:

Vector of Types of Gametes					
$\frac{1\ 0\ 1}{0\ 1\ 0}$ produces	1 0 1	$\frac{1\ 1\ 0}{1\ 0\ 0}$ produces	1 1 0	$\frac{0\ 0\ 0}{0\ 0\ 0}$ produces	0 0 0
	1 0 0		1 1 1		0 0 0
	1 1 1		1 0 0		0 0 0
	1 1 0		1 0 1		0 0 0
	0 0 1		1 1 0		0 0 0
	0 0 0		1 1 1		0 0 0
	0 1 1		1 0 0		0 0 0
	0 1 0		1 0 1		0 0 0

The sequence of operations necessary to simulate the production of gametes by an individual of any genotype is shown below, given that the genotype of the individual is *abc/def* and that $\{f_i\}$ is the vector of frequencies of recombinants and non-recombinants. The first step is to form the vector of types of gametes as shown:

$$\frac{abc}{def} \text{ produces } \begin{matrix} \text{Vector of Types of Gametes} \\ a\ b\ c \\ a\ b\ f \\ a\ e\ c \\ a\ e\ f \\ d\ b\ c \\ d\ b\ f \\ d\ e\ c \\ d\ e\ f \end{matrix}$$

The second step is to transform $\{f_i\}$ by sequential summation to give

$$\{f_{000};\ f_{000}+f_{001};\ f_{000}+f_{001}+f_{010};\ \ldots\ldots\ldots\ldots\ldots\Sigma f\}.$$

It is convenient to set $\{f_i\}$ such that $\Sigma f = 1$. The vector produced by sequential summation will be termed the $\{F_i\}$ vector.

A random number, r_i, is then generated such that

$$0 \leqslant r_i \leqslant 1.$$

This number is then tested across $\{F_i\}$ until

$$F_i < r_i < F_{i+1}.$$

Then the ith term in the vector of types of gametes is taken as the gamete produced. Repetition of this sequence will produce a number of gametes in which the various types occur with probabilities corresponding to the frequencies of recombinants and non-recombinants.

The simulation of the formation of a gamete, g, by an individual of genetic constitution A/B is represented as

$$T : A/B \rightarrow g,$$

where T represents the operation of forming the vector of types of gametes, testing a random number against $\{F_i\}$, and then selecting the ith type of gamete from the vector of types of gametes.

This method of simulating recombination can be set to simulate the independent assortment of linkage groups by specifying one or more of the values of r to be 0·5. If the genetic system to be simulated is of six loci then the following values of r will simulate two linkage groups:

Location	0	1	2	3	4	5
r		0·1	0·1	0·5	0·1	0·1

A major limitation of this method is that 2^n registers, where n is the number of loci, are required to store $\{F_i\}$. This limitation can be reduced by simulating the formation of each linkage group separately. If linkage groups are numbered from 0 to N, then the operation can be represented as

$$T : A_i/B_i \rightarrow g_i,$$

where $i = 0$ to N. Then, since each of the transforms of this transformation are independent, the vector of "chromosomes", i.e. $\{g_i\}$, simulates the independent assortment of linkage groups and the occurrence of linkage within each linkage group. A major advantage of this system is that the number of loci is Nn', where n' is the number of loci per linkage group. The required space in the memory is then $N2^{n'}$, which is considerably less than $2^{Nn'}$; further, if the restriction that the linkage relations are the same for each linkage group be accepted, then the required storage is $2^{n'}$, and any restrictions of number of loci are imposed by the time necessary for calculation rather than by the size of the memory.

The sequences discussed above allow the simulation of (i) the formation of a set of genotypes, $\{A/B\}_{\text{progeny}}$, from a set of parental genotypes, $\{A/B\}_{\text{parents}}$, and (ii) the formation of a set of phenotypes, $\{p\}_{\text{progeny}}$, from the genotypes of the progeny, $\{A/B\}_{\text{progeny}}$.

IX. Selection

There are several methods of simulating selection, the majority being variations of the following sequence. The first step is to re-order the phenotypes of the progeny, i.e. $\{p\}_{\text{progeny}}$, in ascending or descending sequence, re-ordering the genotypes of the progeny correspondingly. Then, if there are N' progeny and the programme specifies that N'' of these be retained as parents, it is possible to take (i) the N'' genotypes of $\{A/B\}_{\text{progeny}}$ with the greatest phenotypes, (ii) the N'' genotypes of $\{A/B\}_{\text{progeny}}$ with the least phenotypes, or (iii) the N'' genotypes of $\{A/B\}_{\text{progeny}}$ with phenotypes closest to the mean. These three alternatives simulate selection for one or other extreme or against extremes.

An important aspect of re-ordering the set of genotypes of progeny is that, where phenotypes are identical, position in that section of the store be kept at random, since, where the programme specifies the formation of many progeny per mating, it is possible, without such a precaution, for selection to favour a specific mating.

A deficiency of this method of simulating the operation of selection is that no account is taken of the variation of the number of progeny produced per individual. This can be included by a variation of the method of random normal deviates discussed above. If r_i is a random number and $r_i = (x)$, where x is a random normal deviate in the range 0 to 1, then, given q, the potential number of progeny, it is possible to modify this by

$$q \cdot x_i = Q_i,$$

where Q_i is the actual number of progeny for the ith individual.

Clearly many variations are possible. An example is to relate q to the potential or actual phenotype. This is accomplished by setting $q = \text{f}\,(P_i)_{\text{potential}}$, or $q = \text{f}(P_i)_{\text{actual}}$. A limitation of all these more sophisticated models is that the time necessary for calculation is increased, sometimes markedly, over the simpler models.

X. Conclusions

The sequences discussed above, when formulated in a programme suitable for an automatic computer, allow simulation of genetic systems. There are various problems in which this approach may be useful. These are (i) the effects of linkage on the efficiency of selection, (ii) the construction of tables relating the competitive efficiencies of alleles to the parameters of population size, intensity of selection, etc., and (iii) the comparison of the efficiencies of different breeding plans for varying degrees of inter-locus interactions. In subsequent papers of this series results gained from running various programmes covering these problems will be discussed.

XI. Acknowledgments

I am deeply indebted to Dr. P. J. Claringbold for introducing me to the Monte Carlo method, and for his patient tuition in various aspects of programming. The staff of the Adolph Basser Computing Laboratory were extremely helpful, particularly Dr. J. C. Bennett, Dr. B. Chartres, and Mr. J. Butcher. I am grateful to Miss J. Ogilvie for her preparation and meticulous checking of the various programmes.

XII. References

Hastings, C. (Jr.) (1955).—"Approximations for Digital Computers." (Princeton University Press.)

Lerner, M. (1954).—"Genetical Homeostasis." (Oliver & Boyd: Edinburgh.)