Note: The following is an unpublished memo from 1989. Some small clerical or editorial changes are made in 2018, but otherwise this is the original document. All new comments from 2018 are provided by footnotes.

RUNNING HEAD: INVERSE GENOMIC TABLE

INVERTING THE EXTENDED GENOMIC TABLE: A CASE STUDY

S.P.Smith and A. Mäki-Tanila
Department of Animal Breeding
Agricultural Research Centre, MTTK
31600 Jokioinen, Finland

SUMMARY

A series of programs are described which calculate the inverse of the extended genomic table. These programs are demonstrated using a pedigree taken from a selection experiment involving egg-laying hens. The calculations are feasible and the inverse matrix was found to be sparse.

I. INTRODUCTION

SMITH & MÄKI-TANILA (1990) proposed a method to model dominance and inbreeding via the mixed model - which need not be linear. Using combinatorial arguments these authors presented the formulae for inverting the extended genomic table **E**. Matrix **E** is almost a (co)variance matrix for additive and dominance terms to the effect that a submatrix of $\mathbf{E}^{-1}$ can be used directly in HENDERSON'S (1973) mixed model equations. More importantly, when effects are additive over loci the usage of **E** is consistent with genetic theory described by early researchers (GILLOIS 1964; HARRIS 1964).

SMITH's & MÄKI-TANILA's left many unanswered questions. In particular, the pertinent submatrix of $\mathbf{E}^{-1}$ tends to be larger than desired and the proposed method of calculation has not been tested. The purpose of this paper is to demonstrate: that the submatrix can indeed be calculated for a real pedigree; that the calculations are feasible; and that while the submatrix is large it tends to be sparse. First we present detailed descriptions of the software used.

II. PROGRAMS

There were six FORTRAN programs designed for the evaluation of $\mathbf{E}^{-1}$. These were named ZYGNUM, PRUNE, GAMETE, SUBSORT, BLOCKS, INVERT, SORT. Each program creates

data sets which feed into subsequent programs as indicated by Figure 1.

A. ZYGNUM

This program has four parts:

1. Renumbering of Zygotes. This is very simple as it entails: extracting animal identifiers from all sources; sorting the identifiers; and assigning consecutive integers to animals based on sort order of identifiers. The integers are then used as addresses for computer memory as required by subsequent operations. The identifiers are extracted from two sources: the pedigree ZYGPED; and a listing of animals with phenotypic measurements ZYGLST.

2. Depth-first Search Extraction of All Ancestors. This is an important step that involves extracting all known ancestors of animals which have records. Pedigrees generally have animals which are not ancestors of animals with records. Including the non-ancestors in the ensuing analysis is inefficient.

The extraction is achieved by depth-first search. Depth-first searches have many applications. A theoretical treatment of the topic can be found on page 181 of REINGOLD & HANSEN (1983).

The depth-first search is seeded with the list of renumbered animals with records. The search for ancestors is only active at the end of the list. When animal n at the end of the list is selected for evaluation, the list is shortened automatically by one to n-1. If the sire and dam of animal n have not already been flagged as ancestors they are added to the end of the list and flagged accordingly. This process continues until the list is empty.

It is also possible to extract ancestors via breadth-first search: search from the beginning of the list and add to the end of the list. However, the size requirements of the work array are generally smaller for depth-first search. This is because the depth-first search is active only for one set, an animal and its ancestors, at a time. Therefore, depth-first searches should be preferred.

3. Topological Numbering of Zygotes. SMITH & MÄKI-TANILA required gametes to be ordered such that j>i if gamete i is an ancestor of gamete j. This is called a topological numbering and it can also be found via depth-first search (REINGOLD & HANSEN 1983). For the present case, however, ZYGNUM proceeds through the pedigree and switches integer identifiers between parent and offspring when they are out of correct order. The process is continued until no more switches can be made. A work vector is used to store the present integer identifiers. Thus the pedigree itself is not modified because switches are only applied to the work vector. The pedigree is easily interpreted using the work vector. This process is done iteratively and the required number of sweeps through the pedigree is determined by the maximum number of generations present. The output file NAME1 is use to store the topological numbering.

4. Gametic Numbering. The gametic pedigree, GAMPED, and the gametic pairs associated with

records, GAMLST, is formed using the useful convention that animal i represents two gametes 2*i and 2*i-1. This numbering procedure maps the topological numbering system for animals into a topological system for gametes.

B. PRUNE

Not only is it desirable to remove non-ancestral zygotes from the pedigree, it also advisable to prune off those ancestors which contribute nothing to the ensuing analysis. A well pruned pedigree reduces the computational burden with no effect on statistical accuracy. In order to identify which ancestors can be pruned it is simpler to consider pruning of gametic pedigrees. A gamete is said to met the "non-redundancy" criterion if it can be connected by at least two non-intersecting genetic pathways leading to two different gametes with recorded phenotypes. One useful pruning rule states, "any base gamete can be pruned if: it fails to met the non-redundancy criterion; and has no phenotypic record itself." As the rule refers to base gametes it must be applied recursively; one round of pruning produces a new collection of base gametes that may also be eligible for pruning. Program PRUNE use gamete list GAMLST as a seed and prunes the pedigree GAMPED to produce PEDPRU. There are two major parts:

1. Extraction of Common Useful Ancestors. We have borrowed the phrase "common useful ancestor" from NADOT & VAYSSEIX (1973) with minor modification. For the present application a common useful ancestor is understood to be any base or non-base gamete meeting the non-redundancy criterion.

The algorithm processes gametes of GAMLST one at a time. Each gamete seeds a work vector and a series of recursive operations are performed in depth-first fashion. These are listed below.

★ determines if parent gametes x and y of gamete n, at the end of the list, are ancestors of earlier seed gametes excluding all pathways which intersect gamete n and its immediate descendants leading to the seed gamete. If at least one pathway can be found, x and y are added to the list of common useful ancestors. The term earlier is used to represent those seed gametes which have already been processed by the algorithm.

★ records the fact that x and y are ancestors of gamete n and its immediate descendants leading to the seed gamete.

★ and adds x and y to the end of the list after removing gamete n. That is, x and y are added to the present list if they have not already been added.

The above steps are implemented efficiently by PRUNE using only a few work vectors. This is possible because the program evaluates seed gametes sequentially. Extracting the entire set of common useful ancestors is, however, computer intensive because PRUNE takes no shortcuts when tracking back through the pedigree from each seed gamete. Nevertheless, pruning pedigrees is highly recommended.

2. Pruning. Once the set of common useful ancestors is determined it is combined with the set of gametes with records. Call this set the inclusion set. The inclusion set is sorted so as to allow bisection. Program PRUNE sweeps through the pedigree and prunes any base gamete that is not in the inclusion set. This is done repeatedly until nothing further can be pruned.

There are some negative aspects of pruning. It is possible to prune only one gamete of the two which define a zygote. This is not proper because the recursions in SMITH & MÄKI-TANILA (1989) assume that either both maternal and paternal gametes are present or that they are both unknown. Fortunately, because maternal and paternal gametes of zygote i are indexed by 2*i and 2*i-1 it is possible to restore those pruned indices needed for the recursions; at least one of 2*i or 2*i-1 is present to calculate the missing index. This is one of the last operations in PRUNE before creating the output pedigree PEDPRU.

Because PEDPRU represents whole zygotes it is feasible to prune zygotic pedigrees with the above approach. Note, however, that the non-redundancy criterion does not apply to zygotes directly. In particular, seemingly redundant base animals which contribute to the inbreeding of other non-redundant animals should not be pruned. Alternatively, when working at the gametic level there is no such issue as gametes are never inbred.

C. GAMETE

After removal of non-ancestors and pruning of ancestors there are many gaps left in the indices used to number gametes. Morever, one requirement for the numbering of gametes still must be met: j>i implies that i is a base gamete if j is (SMITH & MÄKI-TANILA 1989). Program GAMETE renumbers gametes of PEDPRU and GAMLST using consecutive numbers and does it in such a way as to fulfil the last requirement. Moreover, the topological numbering is maintained even though new indices are assigned partially on sorted order of old indexes. The renumbered pedigree PEDREN and gamete pair list GAMREN are created. The integer assigments are saved for future reference in file NAME2.

D. SUBSORT

Program SUBSORT is a sophisticated version of the depth-first search given in Appendix A of SMITH & MÄKI-TANILA (1989). This program reads PEDREN and GAMREN and identifies the row and column labels of **E**. The labels are sorted automatically as they are found using an efficient linked-list (TIER & SMITH 1990). The labels are placed in file SUBSEQ.

E. BLOCKS

This program[1] is the heart of the procedure: calculation of the absorbed diagonal blocks (SMITH & MÄKI-TANILA 1989) of **E**. Program BLOCKS reads PEDREN and SUBSEQ and calculates the blocks and stores them in file ABSBLO. The evaluation is done using a nested recursion which is a new development. From SMITH & MÄKI-TANILA the blocks correspond to particular gametes. For base gametes there are no such blocks but an initialization block was described. For a non-base gamete i the corresponding block is E{**S**'**S**} where

$$\mathbf{S} = \tfrac{1}{2}\,\mathbf{H}_x - \tfrac{1}{2}\,\mathbf{H}_y ;$$

$$\mathbf{H}_z = \{\ \mathbf{a}_z , \mathbf{d}_{j1\,z} , \mathbf{d}_{j2\,z} , \ldots , \mathbf{d}_{jm\,z}\ \} , \ z=x \text{ or } y;$$

x and y are parent gametes of i; j1 , j2 , . . . jm is some sequence of gamete indices which characterize the block;[2] finally $\mathbf{a}_z$ and $\mathbf{d}_{jz}$ are vectors of additive and dominance effects typical to gametes z and j. The vectors of genetic effects are of length n and contain the contributions of n loci to the genotype.

To evaluate E{**S**'**S**}, BLOCKS first computes a sorted list of ancestors of gamete i. This involves a simultaneous sort and depth-first search. The ancestor list is very useful as it determines the kind of recursions that are allowable which are described next.

Typical elements of E{**S**'**S**} are calculated as indicated below. Here we are assuming that k>j, the largest base index is β, the set of ancestors of gamete i is Ω, parent gametes of k>β are p and q , and parent gametes of j>β are f and h. The additive genetic variance and dominance variance are denoted as σ^2_A and σ^2_D, respectively.

★ $E\{(\tfrac{1}{2}\,\mathbf{a}_x - \tfrac{1}{2}\,\mathbf{a}_y)'(\tfrac{1}{2}\,\mathbf{a}_x - \tfrac{1}{2}\,\mathbf{a}_y)\} = \tfrac{1}{2}\,\sigma^2_A\,(1 - F_{xy})$;
F_{xy} is the inbreeding coefficient in the zygote representing x and y. It is calculated via gametic recursion.

---

[1] The 1989 version of BLOCKS had three small errors in the computer code, and these have been corrected in the 2017 version that is now available. Moreover, when singularities are not removed by leaving the homozygotic dominance effects included, some additional recursions are needed and must be amended to BLOCKS for calculating U(k) and V(j,k).

[2] Because singularities have been removed following SMITH & MÄKI-TANILA, jm≠i. If singularities are not to be removed then it is possible for jm=i, and this substitutes $\mathbf{d}_{jm\,z}$ with $\mathbf{d}_{zz}$ in the definition of $\mathbf{H}_z$.

★ $E\{(½\mathbf{a}_x - ½\mathbf{a}_y)'(½\mathbf{d}_{xk} - ½\mathbf{d}_{yk})\} = U(k)$
where:

$$U(k) = \begin{cases} 0, \text{ if } k<\beta \text{ and } k \notin \Omega. \\ ½U(p)+½U(q), \text{ if } k>\beta \text{ and } k \notin \Omega \\ \text{Enter secondary recursion if } k \in \Omega. \end{cases}$$

★ $E\{(½\,\mathbf{d}_{xj} - ½\,\mathbf{d}_{yj})'(½\,\mathbf{d}_{xk} - ½\,\mathbf{d}_{yk})\} = V(j,k)$
where:

$$V(j,k) = \begin{cases} 0, \text{ if } j \,\&\, k<\beta \text{ and at least one of } j \text{ or } k \notin \Omega. \\ 0, \text{ if } j<\beta,\ k>\beta,\ k \in \Omega, \text{ but } j \notin \Omega. \\ ½\,\sigma^2_D\,(1-F_{xy}), \text{ if } j=k<\beta \text{ and } k \notin \Omega. \\ ½V(j,p) + ½V(j,q), \text{ if } k>\beta, \text{ and } k \notin \Omega. \\ ½V(f,k) + ½V(h,k), \text{ if } j>\beta,\ j \notin \Omega \text{ but } k \in \Omega. \\ ½V(p,p) + ½V(q,q), \text{ if } j=k>\beta \text{ and } k \in \Omega. \\ \text{Enter secondary recursion if both } j \text{ and } k \in \Omega. \end{cases}$$

The recursions listed above are called primary recursions and use only two genetic parameters. If the population is not inbred the secondary recursions are never called upon. The secondary recursions are found by applying the formulae of SMITH & MÄKI-TANILA to U(k) and V(j,k). These formulae use three additional genetic parametes.

Program BLOCKS is a FORTRAN program and it is interesting that recursive operations were coded with much ease; FORTRAN is usually criticized for its lack of recursive compatibility. Recursions are calculated in depth-first fashion using a work list with pointers. Once the search is completed the list is evaluated sequentially in reverse order while the pointers are used to compute the recursive additions.

Because BLOCKS was developed on a micro computer no effort was made to avoid redundant calculation. Doing so would have increased the memory requirements.

F. INVERT

Program INVERT reads the absorbed blocks one at a time from file ABSBLO and calculates contributions to $\mathbf{E}^{-1}$ using sparse matrix absorption (TIER & SMITH 1990).

Let $\mathbf{B}_i$ signify the absorbed block associated with gamete i. Define $\mathbf{L}_i$ as given by SMITH & MÄKI-TANILA. Matrix $\mathbf{L}_i$ represents the recursive operations used to calculate those elements of **E** above the diagonal block for the i-th gamete.

For each non-base gamete i, INVERT sets up a linked-list represented by:

$$\begin{array}{c|cc} \mathbf{B}_i & \mathbf{I} & -\mathbf{L}_i \\ \hline \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{L}_i' & \mathbf{0} & \mathbf{0} \end{array}$$

Matrix $\mathbf{B}_i$ is then absorbed into the null matrix to create the negative contributions to $\mathbf{E}^{-1}$:

$$\begin{array}{cc} -\mathbf{B}_i^{-1} & \mathbf{B}_i^{-1}\mathbf{L}_i \\ \mathbf{L}_i'\mathbf{B}_i^{-1} & -\mathbf{L}_i'\mathbf{B}_i^{-1}\mathbf{L}_i \end{array}$$

Matrix $\mathbf{L}_i$ can take on various forms the most common being a column permutation of (½**I** , ½**I**). In general, $\mathbf{L}_i$ is determined by identifying the recursions it represents. Because of the singularity given by equation (6) in SMITH & MÄKI-TANILA special alterations are occasionally needed. It is fortunate that the SMITH & MÄKI-TANILA theorems imply that the singularities given by equation (6) are the only singularities. This allows a systematic determination of $\mathbf{L}_i$ which would otherwise be difficult.

To evaluate $\mathbf{E}^{-1}$, INVERT also calculates the inverse of the so- called initialization block. As pointed out by SMITH & MÄKI- TANILA this is a trival task.

An efficient strategy is to add the contributions of $\mathbf{E}^{-1}$ directly to a second linked-list. This automatically sorts elements of $\mathbf{E}^{-1}$ based on row and column indices and combines like terms. However, this step has been omitted given time constraints. Program INVERT writes the contributions of $\mathbf{E}^{-1}$ to an exterior file EINVER as they are calculated.

G. SORT

Program SORT is the only program of the six not designed on a micro computer. This program uses extended addressing to sort and combine like terms in file EINVER. First SORT reads file SUBSEQ and forms a skeleton of a linked-list. Next the elements of EINVER are added to the list. Finally, the non-zero elements of $\mathbf{E}^{-1}$ are written to file INVSRT.

III. EXAMPLE

To demonstrate the procedure, pedigree data was borrowed from a selection experiment involving egg-laying hens at the Department of Animal Breeding (Agricultural Research Centre, Jokioinen, Finland). There were five non-overlapping generations representing 2386 females and 867 males present in file ZYGPED. Generation one represents the base population which is followed by four generations depicting two divergent lines. The last three generations of phenotypic measurements - eqq number - were used to seed the processes described in section II. File ZYGLST contained 1889 records representing female phenotypes.

IV. BEHAVIOR OF ALGORITHM

The computing times used by each program are listed in Table 1. These times apply to a VAX 8200 computer. From start to finish the process required 21 minutes and 23 seconds with the largest share of computing coming from sorting and summing the large file processed by SORT. To understand the significance of the figures in Table 1, we first consider the magnitudes and the ramifications of the chores encountered by each program.

Program ZYGNUM renumbered 3468 zygotes and extracted 5052 pedigree records (gametic records) via depth-first search for the two studies.

Program PRUNE identified 894 common useful ancestors. These were used to prune off 94 gametes (or 47 zygotes). In effect pruning reduced the number of generations to something less than 5.

As determined by program GAMNUM, our study had 4958 gametes in the analysis. There were 341 base gametes in the pruned pedigrees.

The order of the **E** submatrix was calculated by SUBSORT and it was 72327. The initialization block was of order 20179, and was larger than the block described in the feasibility study of SMITH & MÄKI-TANILA. This is due to the fact that the pedigree used by SMITH & MÄkI-TANILA had only 55 base gametes. The distribution of size for the diagonal blocks are

given in Table 2. As with SMITH & MÄKI-TANILA most of the blocks were of order 2 or 3. There were only a few of order 176 to 275 and none higher than 275. Not only were the blocks tending to be small but they were also very sparse. There were only 10775 off-diagonal non-zero elements among the initialization block and absorbed blocks in matrix of size 72327. This means that the absorbed blocks calculated by BLOCKS were almost diagonal.

Program INVERT worked well in calculating $\mathbf{E}^{-1}$. The ratio of the maximum pivot over the minimum pivot was about 7.8, suggesting that rounding errors were well in check. However, this comparison is affected by the genetic parameters used. Because no singularities were encountered, the results verify the SMITH & MÄKI-TANILA theorems. As noted above the output file is an unsorted collection of terms which need to be sorted and combined; file EINVER contained 452037 records. The end result was that the pertinent submatrix of $\mathbf{E}^{-1}$ contained 317040 non-zero elements.

V. DISCUSSION

This study confirms that $\mathbf{E}^{-1}$ can be evaluated for a real pedigree. As with all case studies, however, it is difficult and dangerous to generalize to other scenarios. It would seem that many pedigrees from selection experiments can be handled. Perhaps pedigrees from field records are outside our grasp because the feasibility of the method seems to be more a function of the number of generations rather than the number of zygotes.

No claims are made about the efficiency of the programs used. As the approach is so involved it is certain that the programs can be improved. For example, the last large step - sort and combining of like terms - can best be done while elements of $\mathbf{E}^{-1}$ are being created in INVERT using a linked-list. Furthermore, there may be some advantage in avoiding the redundant calculations in BLOCKS. This can be accomplished by saving some of the calculations and increasing the storage requirements. The programs are being upgraded for release to the scientific community.

New research is needed in evaluating our method and other competing approaches. Now that we have evaluated $\mathbf{E}^{-1}$, how is a matrix with 317040 elements to be used? The inverse matrix depends on 5 genetic parameters. How are the parameters to be estimated? These important questions need answers.

REFERENCES

GILLOIS, M. (1964) La relation d'identite en genetique. These Fac. Sci. Paris. pp. 205.

HARRIS, D.L. (1964) Genotypic covariances between inbred relatives. Genetics 50, 1319-1348.

HENDERSON, C.R. (1973) Sire evaluation and genetic trends. Proc. of the Anim. Breeding and

Genet. Symp. in Honor of Dr. Jay L. Lush, ASAS-ADSA, Champaign, pp. 10-41.

NADOT, R. & VAYSSEIX, G. (1973) Apparentement et identite. Algorithme du calcul des coefficients d'identite. Biometrics 29, 347-359.

REINGOLD, E.M. & HANSEN (1983) Data Structures. 450 pp. Little, Brown & Company; Boston.

SMITH, S.P. & MÄKI-TANILA, A. (1990) Genotypic covariance matrices for models allowing dominance and inbreeding. Gènèt. Sèl. Evol., 22, 65-91.

TIER, B. & SMITH, S.P. (1990) Use of sparse matrix absorption in animal breeding. Submitted Gènèt. Sèl. Evol., 21, 457-466.

Table 1. Computing times required by the six FORTRAN programs on a VAX 8200 computer to calculate the sample $\mathbf{E}^{-1}$.

Programs	Minutes	Seconds
ZYGNUM	0	12
PRUNE	0	15
GAMETE	0	10
SUBSORT	0	18
BLOCKS	6	24
INVERT	4	57
SORT	9	7
Total	21	23

Table 2. Distribution of sizes of absorbed blocks of **E**.

Order	Number
2- 3	3065
4- 7	522
8- 15	350
16- 27	213
28- 50	225
51- 75	112
76- 175	160
176- 275	6

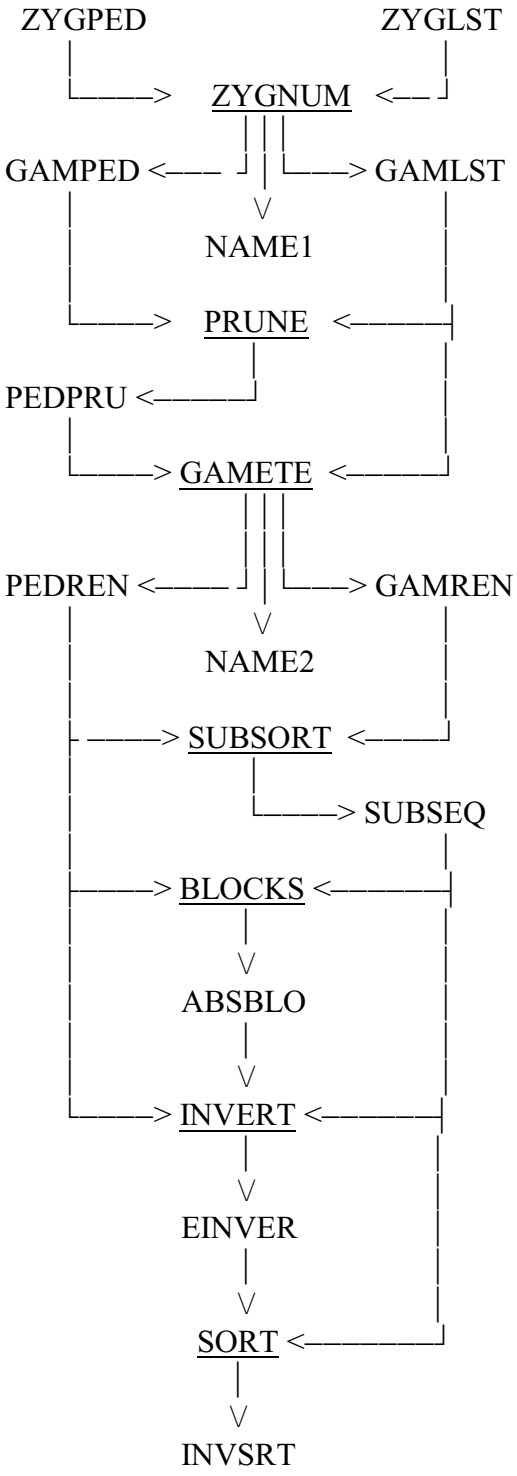

Figure 1. Flow chart depicting programs (underlined) and files in concert.