YAMS
A sparse matrix package

Yutaka Masuda

University of Georgia

Summer Course May 25, 2018

YAMS

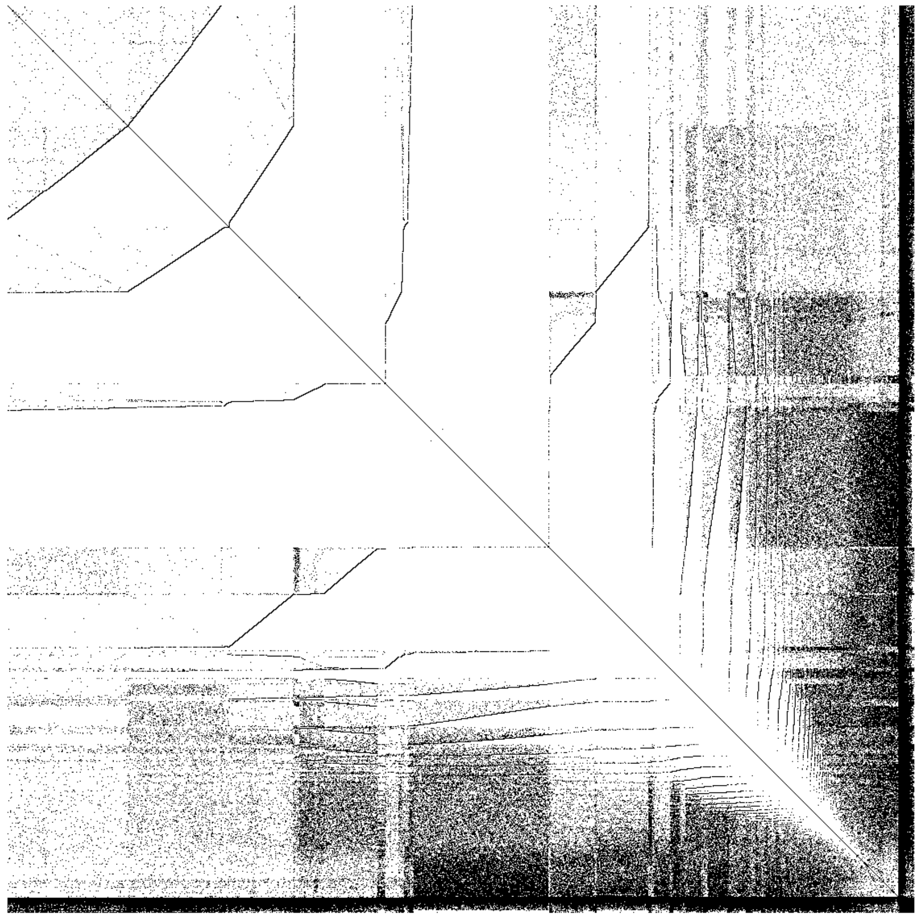
Bruce's
YAMS
CUT SWEET POTATOES IN SYRUP
NET WT. 40 OZ. (2 LB. 8 OZ.) 1.134 Kg

- Yet Another Mixed-model Solver
- Sparse matrix operations
  - For symmetric semi-positive definite matrix (LHS of MME)
  - Mainly Cholesky factorization and sparse inversion
- Faster version of FSPAK
  - Supernodal methods
- `OPTION use_yams`

Real yams

Chinese-yam, cinnamon vine, nagaimo, igname de Chine,
山药, 山藥, 마, 長芋

Sparse matrix operations



- LHS of mixed model equations with $\mathbf{A}^{-1}$ is sparse.
- Iterative methods are perfect just for solving the equations.
- “Direct methods” are still needed for some purposes.
- When to use it? Any special considerations?

Direct sparse methods

- Direct methods = Cholesky factorization & inversion
  - Equations $\mathbf{Cx} = \mathbf{b}$
  - Factorization $\mathbf{C} = \mathbf{LL}'$
  - Reformulation $\mathbf{L}(\mathbf{L}'\mathbf{x}) = \mathbf{b}$
  - Solution
    $$\mathbf{Ly} = \mathbf{b}$$
    $$\mathbf{L}'\mathbf{x} = \mathbf{y}$$
  - Inversion
    $\mathbf{C}^{-1} = (\mathbf{LL}')^{-1} = \mathbf{L}^{-T}\mathbf{L}^{-1}$
  - Costly but easy for triangular $\mathbf{L}$!

- When to use 1: exact solutions
  - REML computations
  - Tests for computations
- When to use 2: inverse of LHS
  - REML computations
  - Accuracy or reliability of EBVs with prediction error variance (PEV)
- When to use 3: ssGBLUP
  - $\mathbf{A}_{22}^{-1} = \mathbf{A}^{22} - \mathbf{A}^{21}\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}$

Not so easy…

- Tricky data storage
  - The computations are based on indirect access to the elements.

$$\mathbf{C} = \begin{bmatrix} 5 & 3 & 0 & 2 \\ 3 & 4 & 1 & 0 \\ 0 & 1 & 5 & 0 \\ 2 & 0 & 0 & 4 \end{bmatrix}$$

Sparse IJA format:

$$IA = [1 \quad 4 \quad 6 \quad 7 \quad 8]$$
$$JA = [1 \quad 2 \quad 4 \quad 2 \quad 3 \quad 3 \quad 4]$$
$$A = [5 \quad 3 \quad 2 \quad 4 \quad 1 \quad 5 \quad 4]$$

Not so easy…

- Tricky data storage
  - The computations are based on indirect access to the elements.
- More non-zero elements in the Cholesky factor (*fill-in*)
  - The density in the factor changes by the ordering of the equations!

Original matrix

$$\begin{bmatrix} 5 & 1 & 1 & 1 & 1 \\ 1 & 5 & & & \\ 1 & & 5 & & \\ 1 & & & 5 & \\ 1 & & & & 5 \end{bmatrix}$$

Reordered matrix

$$\begin{bmatrix} 5 & & & & 1 \\ & 5 & & & 1 \\ & & 5 & & 1 \\ & & & 5 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{bmatrix}$$

$$\Big\downarrow \quad \textbf{Factorization} \quad \Big\downarrow$$

$$\begin{bmatrix} * & & & & \\ * & * & & & \\ * & * & * & & \\ * & * & * & * & \\ * & * & * & * & * \end{bmatrix}$$

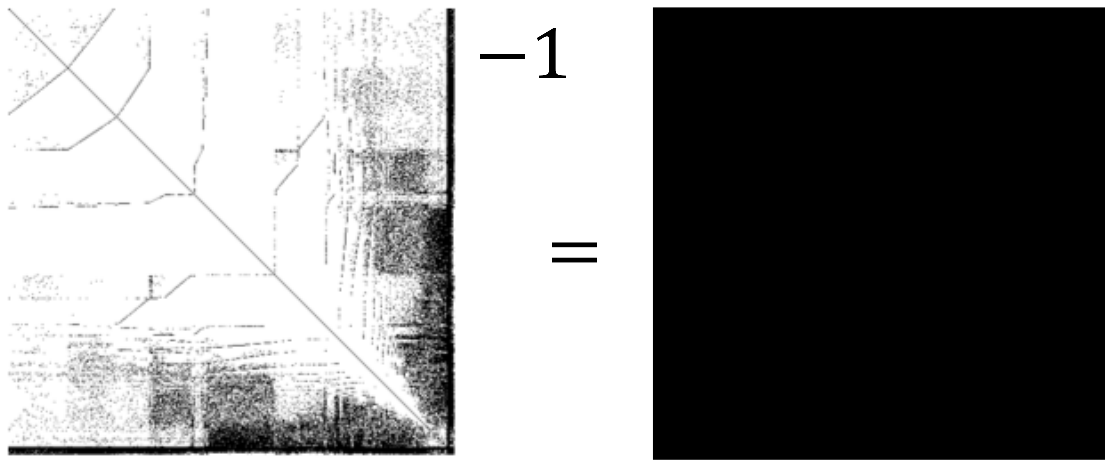
Many fill-ins

$$\begin{bmatrix} * & & & & \\ & * & & & \\ & & * & & \\ & & & * & \\ * & * & * & * & * \end{bmatrix}$$

No fill-ins

Not so easy…

- Tricky data storage
  - The computations are based on indirect access to the elements.
- More non-zero elements in the Cholesky factor
  - The density in the factor changes by the ordering of the equations!
- Dense inverse
  - The inverse of a sparse matrix is usually dense.



Sparse inversion

- A subset of inverse
  - Only the nonzero positions in the original sparse matrix
  - The original matrix updated with the inverse without extra storage
  - Referred as *selected inversion* in computer science
- "Takahashi" algorithm
  - Much less computing cost than the standard inverse

$$\mathbf{C} = \begin{bmatrix} 5 & 3 & 0 & 2 \\ 3 & 4 & 1 & 0 \\ 0 & 1 & 5 & 0 \\ 2 & 0 & 0 & 4 \end{bmatrix}$$

Full $\mathbf{C}^{-1} = \begin{bmatrix} 0.61 & -0.48 & 0.10 & 0.31 \\ -0.48 & 0.65 & -0.13 & 0.24 \\ 0.10 & -0.13 & 0.23 & -0.48 \\ 0.31 & 0.23 & -0.48 & 0.40 \end{bmatrix}$

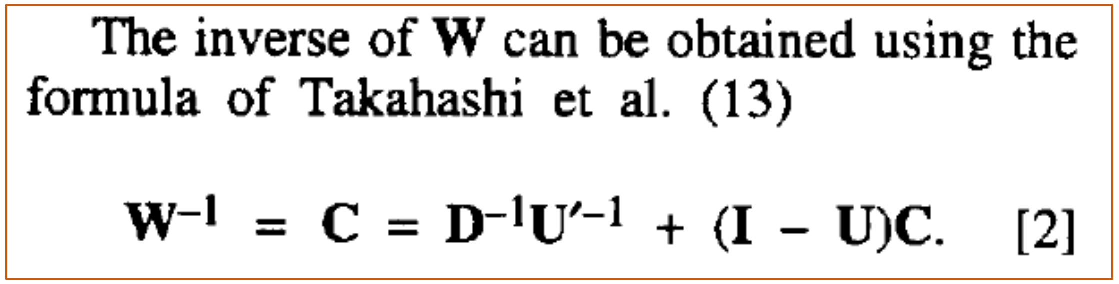
Sparse inverse

$$\begin{bmatrix} 0.61 & -0.48 & 0 & 0.31 \\ -0.48 & 0.65 & -0.13 & 0 \\ 0 & -0.13 & 0.23 & 0 \\ 0.31 & 0 & 0 & 0.40 \end{bmatrix}$$

Sparse matrix computations

1. Build LHS and RHS of MME
2. **Ordering**: Find the ordering to minimize fill-in
3. **Symbolic factorization**: Determine the positions of non-zero elements in the factor
4. **Numerical factorization**: Compute the Cholesky (or LDL’) factor
5. Solve the equations
6. (Optional) Compute the sparse inverse of LHS

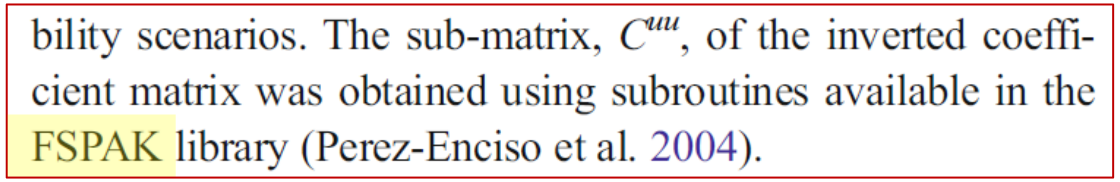
Sparse inversion in animal breeding



The inverse of **W** can be obtained using the formula of Takahashi et al. (13)

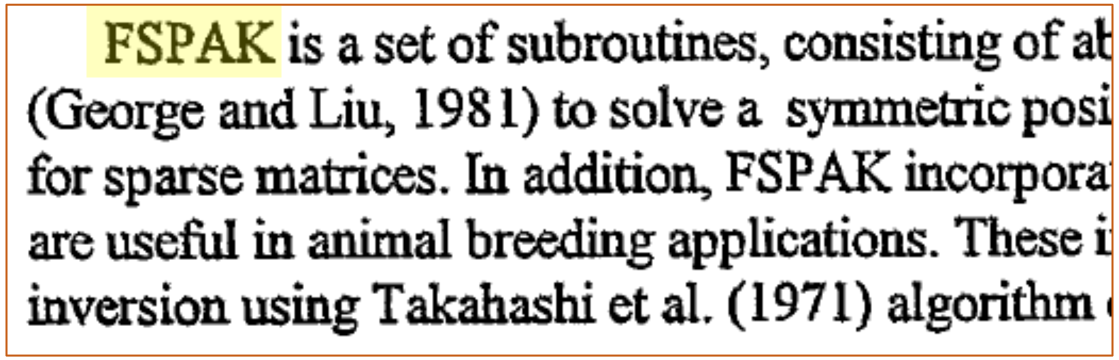
$$\mathbf{W}^{-1} = \mathbf{C} = \mathbf{D}^{-1}\mathbf{U}'^{-1} + (\mathbf{I} - \mathbf{U})\mathbf{C}. \qquad [2]$$

Misztal and Perez-Enciso (1993)



bility scenarios. The sub-matrix, C^{uu}, of the inverted coefficient matrix was obtained using subroutines available in the FSPAK library (Perez-Enciso et al. 2004).

Kerr et al. (2015)



FSPAK is a set of subroutines, consisting of ab
(George and Liu, 1981) to solve a symmetric posi
for sparse matrices. In addition, FSPAK incorpora
are useful in animal breeding applications. These i
inversion using Takahashi et al. (1971) algorithm

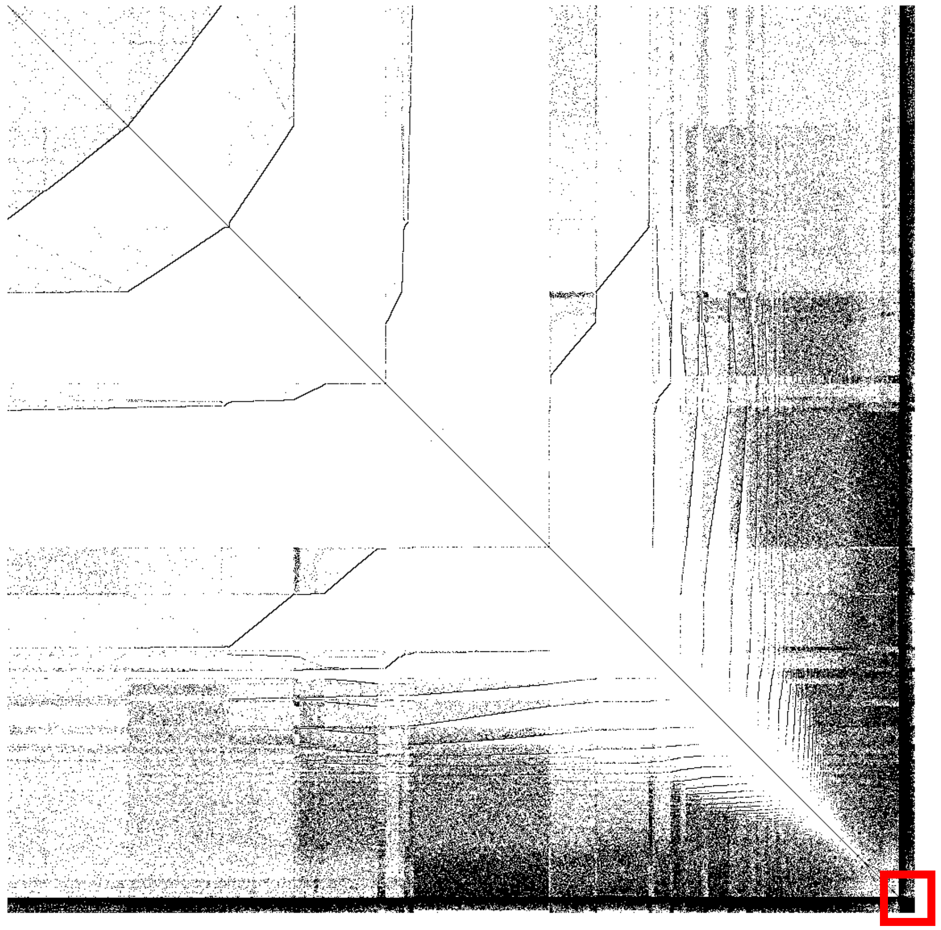
FSPAK by Perez-Enciso et al. (1994)

FSPAK [= George & Liu (1981) + Takahashi algorithm (1973)] is still alive!

Why FSPAK is slow in AIREML?

	Time (s)	
Operation	Animal Model	ssGBLUP Model
Creating MME	1	167
Ordering	30	159
Sym. Factorization	0	680
Num. Factorization	0	717
Sparse Inversion	1	2,077
Other	1	51
Total	<1 m	1 h 4 m

Non-zero structure of MME



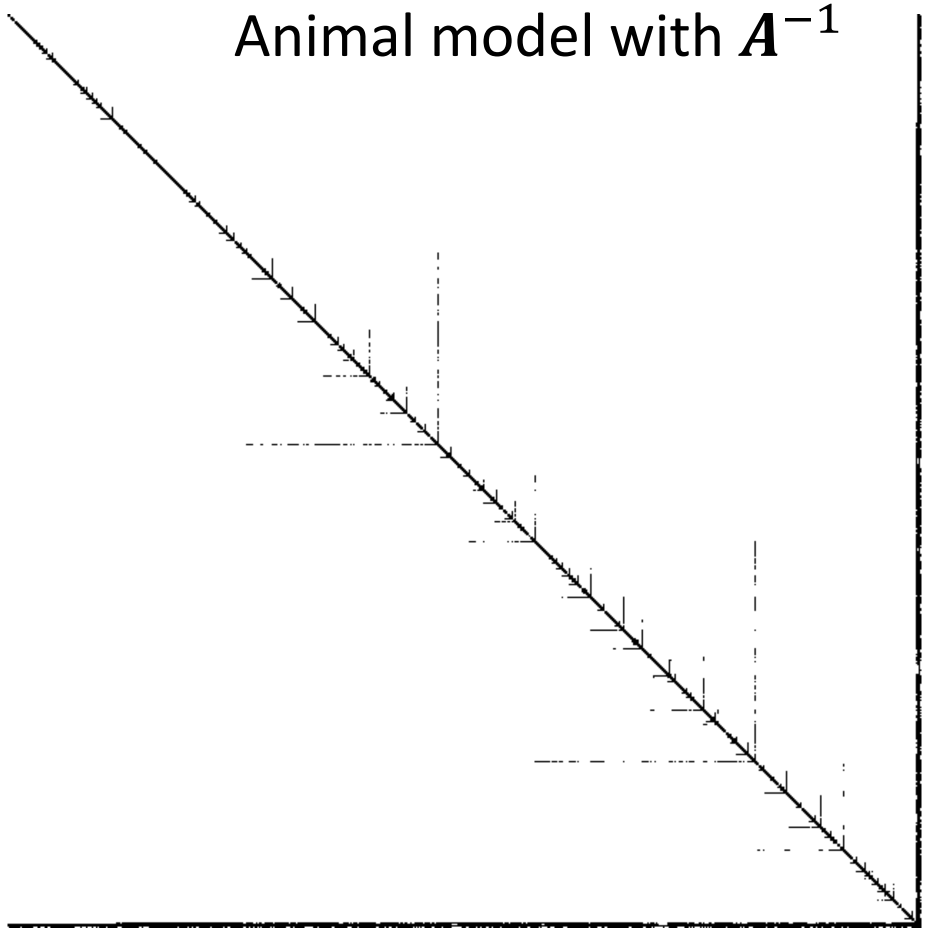
- A large, single dense block in reordered LHS
  - Computing cost $\sim t^3$ (traits)
  - With the same rank of LHS, multiple-trait model is always slower than single-trait model.
  - This happens even in non-genomic models!

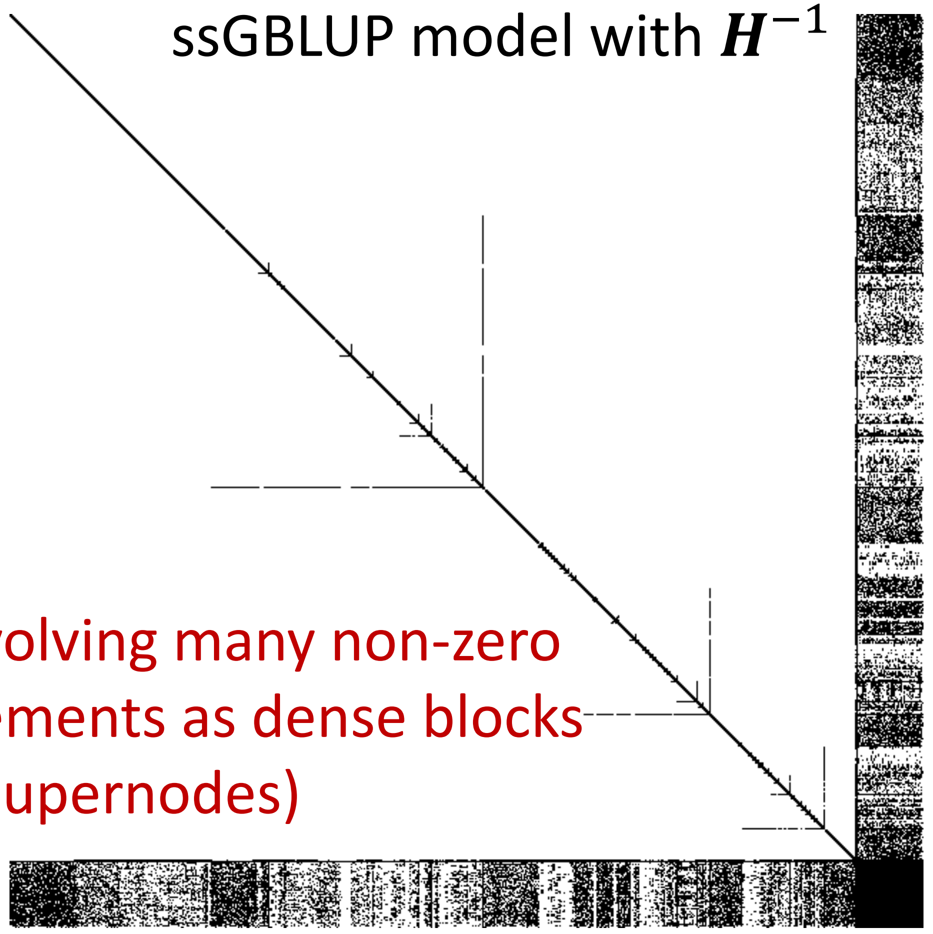
This is the bottleneck!

* Ordered by AMD; 3-trait model i.e. *t*=3

Masuda et al. (2014) J. Anim. Breed. Genet.

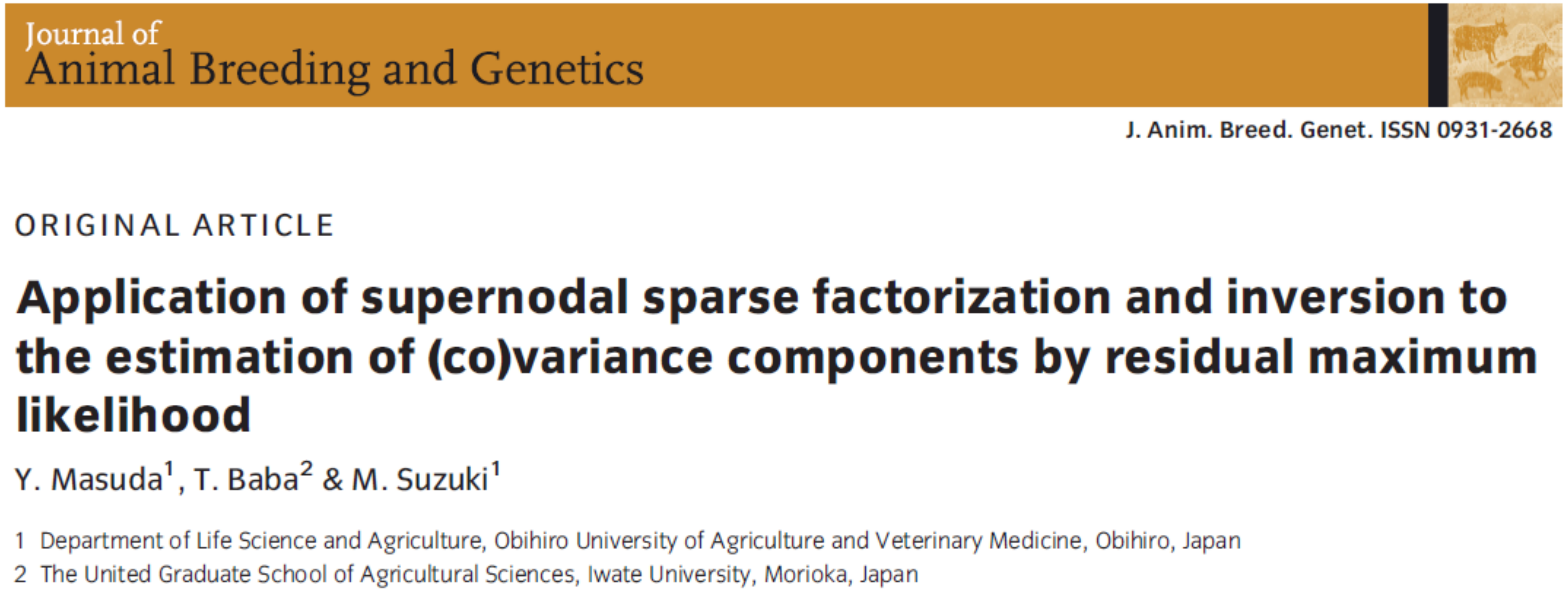
Single-step GBLUP





Involving many non-zero elements as dense blocks (=supernodes)

Yet Another MME Solver (YAMS)


Journal of
Animal Breeding and Genetics

J. Anim. Breed. Genet. ISSN 0931-2668

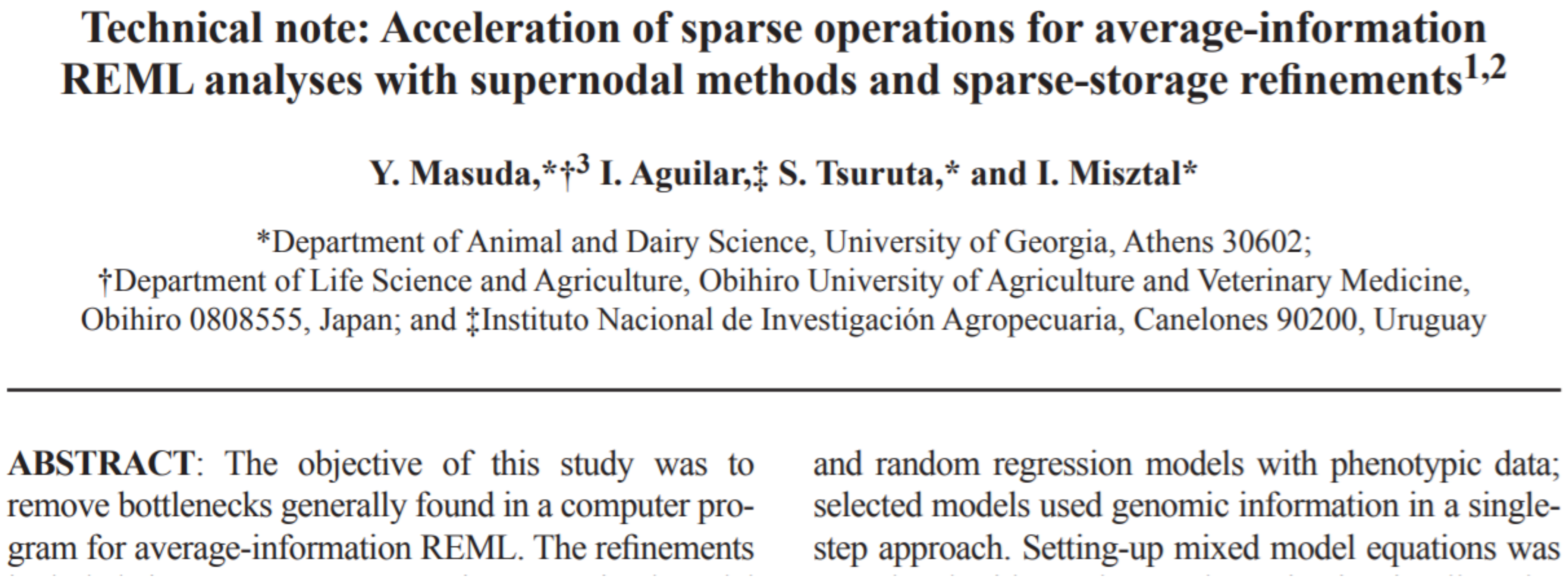
ORIGINAL ARTICLE

Application of supernodal sparse factorization and inversion to the estimation of (co)variance components by residual maximum likelihood

Y. Masuda[1], T. Baba[2] & M. Suzuki[1]

1 Department of Life Science and Agriculture, Obihiro University of Agriculture and Veterinary Medicine, Obihiro, Japan
2 The United Graduate School of Agricultural Sciences, Iwate University, Morioka, Japan

Tests for genomic data



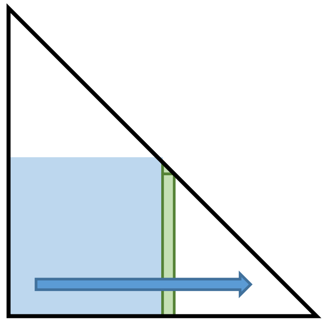
Technical note: Acceleration of sparse operations for average-information REML analyses with supernodal methods and sparse-storage refinements[1,2]

Y. Masuda,[*,†,3] I. Aguilar,[‡] S. Tsuruta,[*] and I. Misztal[*]

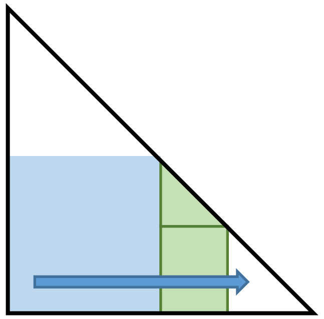
[*]Department of Animal and Dairy Science, University of Georgia, Athens 30602;
[†]Department of Life Science and Agriculture, Obihiro University of Agriculture and Veterinary Medicine, Obihiro 0808555, Japan; and [‡]Instituto Nacional de Investigación Agropecuaria, Canelones 90200, Uruguay

ABSTRACT: The objective of this study was to remove bottlenecks generally found in a computer program for average-information REML. The refinements and random regression models with phenotypic data; selected models used genomic information in a single-step approach. Setting-up mixed model equations was

Supernodal methods



Column-based Factorization (FSPAK)



Supernodal Factorization

Basic Idea:

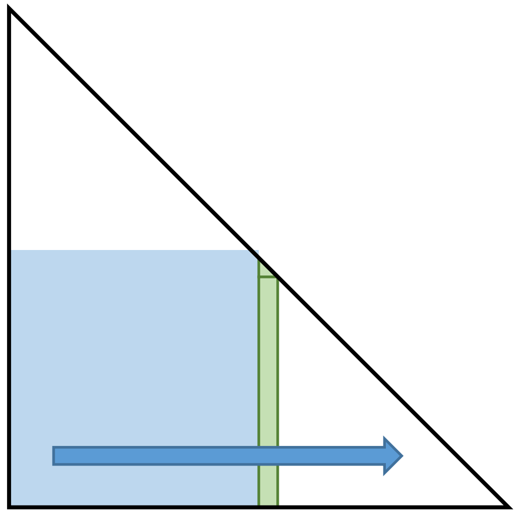
A process of the factorization (or inversion) consists of a set of operations between ~~elements.~~

dense blocks (supernodes)

Acceleration

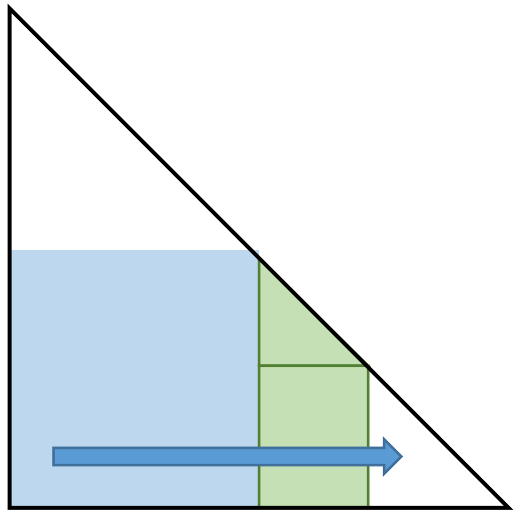
- Use of Parallelized libraries for dense operations
- Simultaneous updates of multiple columns
- More efficient for a sparse matrix involving larger blocks

Supernodal factorization



Left-looking in FSPAK (George & Liu 1981)

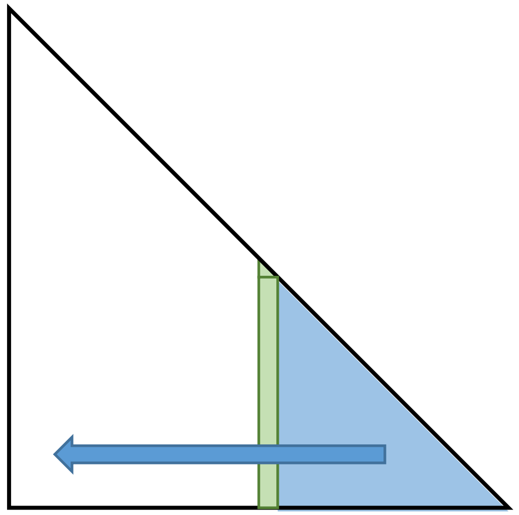
- Column by column
- Low memory requirement, but slow if there are many dense blocks



Supernodal Left-looking in YAMS (Ng & Payton 1993)

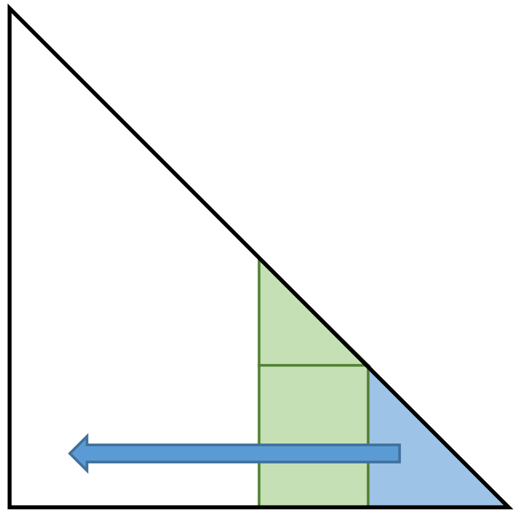
- block by block (supernode by supernode)
- Fast, but more memory required

Supernodal inversion



Takahashi algorithm in FSPAK
(Takahashi et al. 1971)

- Column by column
- Low memory requirement, but slow if there are many dense blocks



Inverse Multifrontal in YAMS
(Campbell 1995)

- A supernodal version of the Takahashi algorithm
- Fast, but more memory required

Benchmarks

Breed	Model	Traits	Pedigree Animals	Genotyped Animals
Broiler	Animal	1, 2, 3, 4	213,297	0
	ssGBLUP	1, 2, 3, 4	213,297	15,723
Pig	Maternal	1	109,113	0
	RRM	1	282,695	0
Beef Cattle	Animal	1, 2, 3	322,451	0
Dairy Cattle	RR-TDM	1, 2, 3	55,063	0
	Animal	1, 2, 4, 8	100,775	0
	ssGBLUP	1	100,775	34,506

Animal model (first 5 rounds in AI REML)

			Time		Speed
Breed	Model	Traits	FSPAK	YAMS	Up
Broiler	Animal	1	<1 m	<1 m	0.9
		4	18 m	7 m	2.6
Pig	Maternal	1	<1 m	<1 m	1.1
	RRM	1	4 m	<1 m	4.3
Beef Cattle	Animal	1	47 m	3 m	16.9
		2	24 h 41 m	1 h 10 m	21.3
Dairy Cattle	RR-TDM	1	44 m	3 m	15.5
		3	5 h 57 m	18 m	20.2
	Animal	1	4 m	<1 m	9.5
		2	34 m	2 m	14.4

Single-step GBLUP (first 5 rounds in AI REML)

			Time		Speed
Breed	Model	Traits	FSPAK	YAMS	Up
Broiler	ssGBLUP	1	4h 11 m	20 m	12.7
		2	N/A	58 m	
		3	N/A	2 h 38 m	
		4	N/A	5 h 10 m	
Dairy Cattle	ssGBLUP	1	N/A	2 h 7 m	

N/A: Crashed during the numerical factorization

Conclusion

```
OPTION use_yams
```