Wrestling with a WOMBAT: New features for linear mixed model analyses in the genomic age

Karin Meyer

Animal Genetics and Breeding Unit, University of New England, Armidale, Australia
What is (a) WOMBAT?

Enigmatic marsupial?

WOMBAT is a software package for linear mixed model analysis in quantitative genetics. Focus: REML estimation of (co)variance components. Aimed at animal breeding applications. Successor of DFREML. Presented at 8WCGALP in 2006.
What is (a) WOMBAT?

Enigmatic marsupial?

Or acronym?

**W**aste **O**f **M**oney, **B**rains **A**nd **T**ime

*Computer hacker’s dictionary*
What is (a) WOMBAT?

Enigmatic marsupial?

Or acronym?

**W**aste **O**f **M**oney, **B**rains **A**nd **T**ime

*Computer hacker’s dictionary*

**WOMBAT**

is a software package for linear mixed model analysis in quantitative genetics

- Focus: REML estimation of (co)variance components
- Aimed at animal breeding applications
- Successor of DFREML
- Presented at 8WCGALP in 2006
Selected topics

Many changes & expansion of capabilities over the last decade:

1. Parallel processing

2. Multivariate estimation: more than a few traits
   - Penalties to reduce sampling errors
   - Pooling estimates from analyses by parts

3. Solving (genomic) mixed model equations
   - Single-step BLUP modules
   - Genomic relationship matrices & friends
Parallel computing

Changes in Computing Paradigm

- From
  - Sparse mixed model equations
  - Single processor, limited memory

- To
  - Multiple processors & cores
    extensive RAM, vast disk space
  - Genomic information
    → dense matrix operations
Changes in Computing Paradigm

- **From**
  - Sparse mixed model equations
  - Single processor, limited memory

- **To**
  - Multiple processors & cores
  - Extensive RAM, vast disk space
  - Genomic information
    ➔ dense matrix operations

- **Adapt software and style of programming**
  - Parallel processing to minimize elapsed (‘wall’) time
Parallel execution for WOMBAT

- REML: iterative solution scheme
  - mostly sequential
- Factor / invert coefficient matrix for each iterate
- ‘Supernodal’ approach
  - identify and extract dense sub-blocks of sparse matrix
  - carry out computations using dense matrix manipulations
  - use efficient BLAS3 and LAPACK library routines
- Sparse vs. dense storage

- WOMBAT: Linux executable
  - compiled using ifort
  - load highly optimised routines from Intel® multi-threaded Math Kernel Library
  - use OMP directives to parallelize selected loops
  - set OMP_NUM_THREADS to limit no. of threads used
Penalized REML for ‘better’ MV estimates

👍 Need good estimates of $\Sigma$

- MV analyses for $q$ traits
  - technically feasible for larger $q$

- Estimates
  - $q(q + 1)/2$ parameters per $\hat{\Sigma}$
  - SAMPLING VARIATION!

- ‘Loss’ $\rightarrow$ difference: $\hat{\Sigma}$ and $\Sigma$
  
  $$L_1(\Sigma, \hat{\Sigma}) = \text{tr}(\Sigma^{-1}\hat{\Sigma}) - \log |\Sigma^{-1}\hat{\Sigma}| - q$$
  - bias$^2$ + sampling variance

○ Improve $\leftrightarrow$ reduce loss
  - Penalty on log $L$ designed to reduce SV
  - Estimates that are on average closer to true values

Choice of penalty?
- Need additional information: assume prior distribution of function of parameters to be estimated
- Penalty $\propto$ log of probability density
Penalized REML for ‘better’ MV estimates

- Need good estimates of $\Sigma$
  - MV analyses for $q$ traits
    - technically feasible for larger $q$
  - Estimates
    - $q(q+1)/2$ parameters per $\hat{\Sigma}$
    - SAMPLING VARIATION!
  - ‘Loss’ $\rightarrow$ difference: $\hat{\Sigma}$ and $\Sigma$
    \[ L_1(\Sigma, \hat{\Sigma}) = \text{tr}(\Sigma^{-1}\hat{\Sigma}) - \log|\Sigma^{-1}\hat{\Sigma}| - q \]
    - bias$^2$ + sampling variance

- Improve $\leftrightarrow$ reduce loss
  - Penalty on $\log L$ designed to reduce SV
  - Estimates that are on average closer to true values

- Choice of penalty?
  - Need additional information: assume prior distribution of function of parameters to be estimated
  - Penalty $\propto \log$ of probability density
Proposal: ‘Simple’ penalties

- Mild, default penalties on scale-free functions of $\hat{\Sigma}$
  - achieve high proportion of reductions in loss feasible
  - avoid laborious estimation of tuning factor

- Functions and assumed prior distributions
  - Canonical eigenvalues
    - Beta distribution on $[0, 1]$ 
    - shrink towards mean
  - Partial correlations
    - correlations for traits $i < j$ given $i + 1$ to $j - 1$
    - Beta distribution on $[-1, 1]$
    - shrink towards zero or phenotypic values

Penalized REML in WOMBAT

- **New and ‘simple’**
  - Invoke by SPECIAL option(s) in parameter file (single line)
    - Select ‘function’ to penalize
    - Choose \( ESS = \alpha + \beta \) of Beta(\( \alpha, \beta \))
    - Set shrinkage target

  ```
  # penalty on genetic partial correlations
  # shrink towards phenotypic; ESS = 8
  SPECIAL
  PENALTY PACORR PHENV animal 8.0
  END
  ```

- **Older, more complicated**
  - Invoke by run option `--bend` and SPECIAL options
    - still functional!
    - requires tuning factor(s)
    - multiple runs & side-by side comparisons

---

Example 19: Use and ☞ of details
Pooling results from analyses by parts

- MANY traits: analyse overlapping subsets
  - e.g. \( q(q - 1)/2 \) pairs of traits

- Pool into overall covariance matrix(es)
  - must be ‘safely’ positive definite
  - have elements ‘similar’ to part results
  - do not change variance ratios markedly
  - do not distort phenotypic variances

- Often done too naively
  - Shrink eigenvalues of one covariance matrix at a time

👉 ‘Bending’ (Hayes & Hill 1981)

Eigenvalues of \( \Sigma_p^{-1} \Sigma_G \)
Pooling results from analyses by parts

- MANY traits: analyse overlapping subsets
  - e.g. \( q(q - 1)/2 \) pairs of traits
- Pool into overall covariance matrix(es)
  - must be ‘safely’ positive definite
  - have elements ‘similar’ to part results
  - do not change variance ratios markedly
  - do not distort phenotypic variances
- Often done too naively
  - Shrink eigenvalues of one covariance matrix at a time
  - Better: Pool matrices for all RE jointly
    - allow for repartitioning due to sampling
    - keep \( \Sigma_P \) approx. same

‌

‘Bending’ (Hayes & Hill 1981)

\[ \text{Eigenvalues of } \Sigma^{-1}_P \Sigma_G \]
Analyses by parts

Likelihood based pooling

- ‘Iterative summation of expanded part matrices’ (Mäntysaari 1999)
- Convert $\hat{\Sigma}_i$ to pseudo-observations (Thompson et al. 2005)
  - use any REML software to pool

Recommend

- Pool covariance matrices for all sources of variation simultaneously
- Construct data matrix in $\log L$ from $\hat{\Sigma}_i$
- Impose ‘pseudo-pedigree’ structure
  ├ mimic sampling covariances between causal components
  │ e.g. balanced paternal-half sib families for simple animal model
- Place very mild penalty on $\log L$

$\Rightarrow$ Simulation: resulting estimates of pooled covariance matrices are on average closer to population values

Pooling using WOMBAT

- WOMBAT is set up to make analyses of subsets of traits easy
  - generates parameter files for part analyses; option **--subset**
  - picks out relevant info from overall data & pedigree files
  - writes out files with partial results; ready for pooling
- Invoke with run option **--pool**
- Additional choices in parameter file
  - pseudo pedigree
  - smallest eigenvalue allowed
  - penalty

```
POOL
# smallest eigenvalue in pooled matrix
  SMALL 0.001d0
# pseudo pedigree structure: pat. half sib
  PSEUPED PHS 50 10
# pool with penalty on canonical eigenvalues
  PENALTY KANEIG 4
END
```

Example 15: use and with details
Adapted for “single-step” analyses → research tool
- Iterative solution via PCG algorithm
- Multivariate incl. principal components
- ‘Explicit’ genetic groups

Breeding value model
- Run option --s1step
  ▶ MME in core; input $H^{-1}$
  ▶ Block-, diagonal or SSOR precond.
- Run option --s2step
  ▶ Iteration on data; input $G^{-1} - A_{22}^{-1}$
  ▶ $A^{-1}$ from pedigree
  ▶ Diagonal preconditioner only

Examples 18 and 21: use and with details
Modules for Iterative Solution of MME

Adapted for “single-step” analyses → research tool
- Iterative solution via PCG algorithm
- Multivariate incl. principal components
- ‘Explicit’ genetic groups

Breeding value model
- Run option --s1step
  - MME in core; input $H^{-1}$
  - Block-, diagonal or SSOR precond.
- Run option --s2step
  - Iteration on data; input $G^{-1} - A_{22}^{-1}$
  - $A^{-1}$ from pedigree
  - Diagonal preconditioner only

Hybrid model Fernando et al.
- Run option --s3step (new)
  - Input: marker allele counts
  - Includes imputation step
  - Diagonal precond.

Examples 18 and 21: use and with details
Genomic relationship matrices in WOMBAT

Many programs available to calculate relationship matrices for SS-BLUP

👍 New WOMBAT module
- Pre-analysis step
- Compatible Input/Output file formats for other WOMBAT tasks
- Choice of methods from literature
- Invoke with run option --hinv

🌟 Some options
- \( \mathbf{G} \) or \( \mathbf{A}_{22} \)
- Weighted average of \( \mathbf{G} \) and \( \mathbf{A}_{22} \)
- Scale \( \mathbf{G} \) to align with \( \mathbf{A}_{22} \)
- \( \mathbf{A}^{-\gamma} \) . . . with Meta-Founders
- \( \mathbf{G}^{-1} \) or \( \mathbf{A}^{-1}_{22} \)
- \( \log |\mathbf{H}| \)
- \( \mathbf{H}^{-1} \) inverse joint rel. matrix
- \( \mathbf{H}^{-\gamma} \) . . . with Meta-Founders

🔧 More options soon (e.g. APY-like)

Example 20: use and with documentation
Summary

- WOMBAT: features for the 21st century
  - Multi-threaded processing
  - Higher dimensional multivariate REML
  - Single step genomic BLUP

- Availability
  - Free download: Executable, manual & suite of examples

http://didgeridoo.une.edu.au/km/wombat.php