A comparison of random forests, boosting and support vector machines for genomic selection using SNP markers

By Joseph O. Ogutu, Hans-Peter Piepho & Torben Schulz-Streeck

University of Hohenheim, Bioinformatics

Outline

0.0 **Objective**

Methods

1.0 Classification and Regression Trees (CART)

2.0 Random Forests-Ensemble extension of CART

3.0. Boosting—with CART as base learners

4.0. Support Vector Machines

5.0 **Results**

6.0 **Conclusions**

0. 0 Objective

Evaluate and compare predictive accuracy of random forests, boosting and support vector machines for predicting genetic breeding values using SNP markers.

Methods

1.0 Classification and Regression Trees (CART)

- CARTs are the building blocks (base procedure or base learners) for random forests.
- We used CARTs as base procedure for boosting as well.
- CARTs recursively partition observations into two groups with minimal within-group variance at a time.
- CART performs flexible nonparametric classification and regression.

- **Three key steps in building Classification Trees**

Let response variable be $\mathbf{y} = (y_1, y_2, y_3, .., y_n)$ and

Let a set of *p* predictors be $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_p)$.

with values $x_i = (x_{1i}, x_{2i}, x_{3i}, \ldots, x_{pi})$

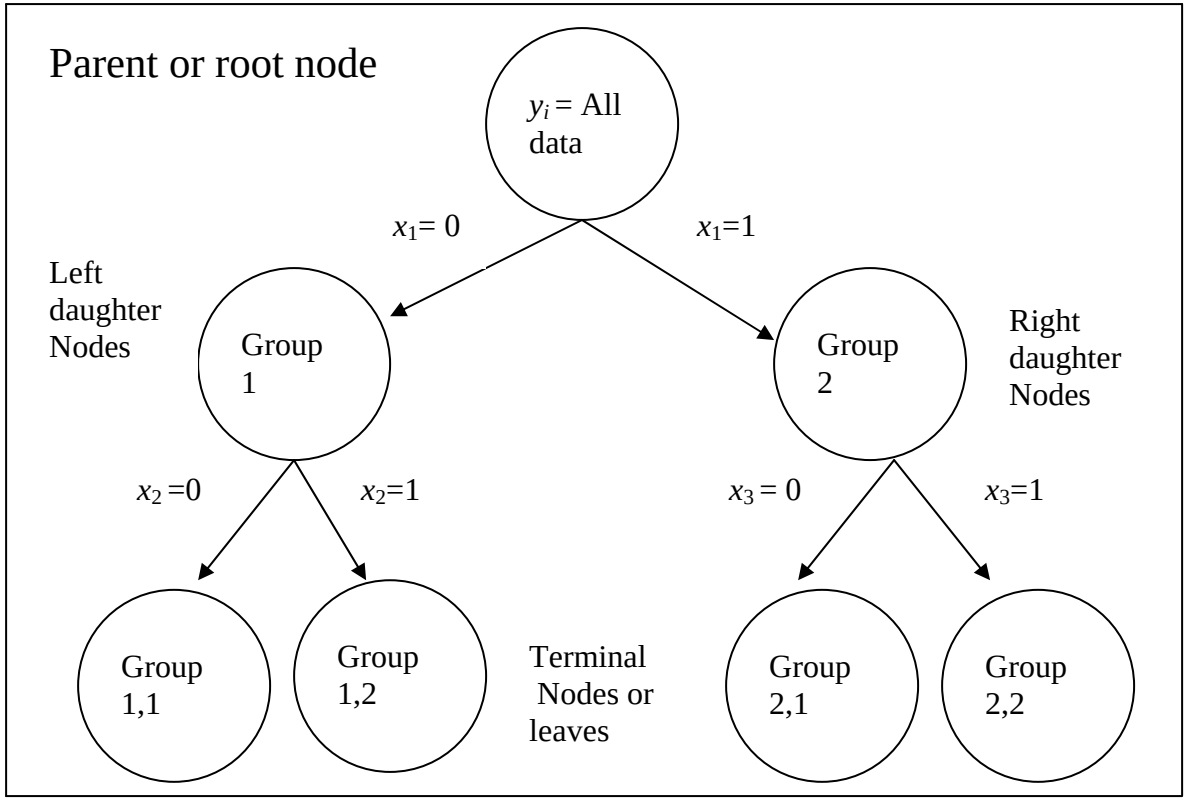
Classification Trees are build to relate **y** to **x as** follows:

Step 1. At each step select only one of the $X_1, X_2, \ldots, X_p$ predictors (say, X_j) that is most predictive of **y**. Choose a split point $X_j = C$ that optimally splits the observations into two subgroups. One group falls in $X_j \leq \mathbf{C}$ and the other in $X_j > C$.

Step 2. Fit a constant model in each cell of the resulting partition, e.g. compute mean or median for continuous responses.

Step 3. Repeat the partitioning process until some stopping rule is satisfied, e.g. until a terminal cell has at most 5 observations.

Schematic Tree Structure for a binary (0,1) response

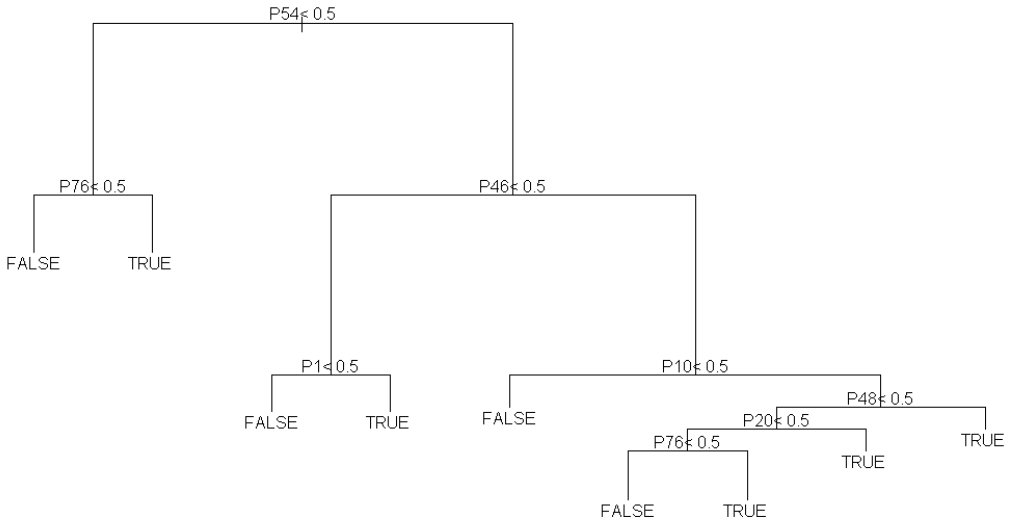


Notes:

1. "Stumps" are trees for which only one break is allowed in the predictor.

2. The trees typically stand upside down in print.

An example classification tree for genomic selection of a discrete trait using 99 predictors called P1, P2, …, P99.



The statistical model fitted by CART:

CART fits additive linear combinations of basis functions:

$$f(x) = \sum_{j=1}^{p} \sum_{m=1}^{M_j} \beta_{jm} h_{jm}(x) \qquad (1)$$

where $j = 1, 2, 3, \ldots p$ predictors each having
$m = 1, 2, 3, \ldots M_j$ transformations, basis functions or breakpoints = $(h_{jm}(.))$.

Each basis function $(h_{jm}(.))$ has weight = β_{jm}.

The specific basis functions used by CART are indicator variables:

Example: $f(x, z) = \beta_0 + \beta_1[\mathrm{I}(x \leq c_1)] + \beta_2[\mathrm{I}(x > c_1 \ \& \ z \leq c_2)]$
$+ \beta_3[\mathrm{I}(x > c_1 \ \& \ z > c_2)]$

where x and z are predictors,
c_1 and c_2 are cutpoints
I(.) are indicator functions or indicator basis functions.

NB: This examples shows that splits beyond the initial split of the root node represent interaction effects.

2.0 Random Forests

Random forest predicts a new observation x using

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x, \Psi_b) \qquad (2)$$

Where $\hat{f}_{rf}^{B}(x)$ = prediction for new observation x

$\{T(x, \Psi_b)\}_1^B$ = ensemble of B trees.

Ψ_b = bth Random forest tree.

- The Random forest settings we used:
- The number of regression trees, *ntree* =1000.

- Number of SNPs sampled randomly without replacement and used to split each tree node, *mtry* = 3000≈ 10031÷3.

- Minimum number of observations allowed in terminal nodes of the trees, *nodesize* = 1.

Importance ranking of SNPs

- Random forest was used to rank the SNPs in order of importance based on the relation:

Importance of the j^{th} SNP, $I_j = \sum_{k=1}^{K}\left[\frac{1}{K}(\upsilon_j - \upsilon)\right]$, (3)

where j = 1,2,…,10031 SNPs

K = total number of trees (1000),

υ_j = prediction error with the j^{th} SNP permuted)

υ = prediction error with no SNP shuffled.

- We mapped the 30 most important SNPs from the 10031 to their locations on chromosomes.

3.0 Boosting

Boosting fits an additive model:

$$f(x) = \sum_{m=1}^{M} \beta_m b(x;\gamma_m) \qquad (4)$$

where β_m, m =1,2,…, M are basis expansion coefficients,

$b(x;\gamma)$=basis functions of x, having parameters γ.

- We used stochastic gradient boosting to fit generalized boosted regression models assuming a normal distribution for the quantitative trait Q and squared error loss.

- Regression trees were used as basis functions, base learners, or base procedures similar to random forests.

- The minimum number of observations allowed in the terminal tree nodes was 1.

- The change in the out-of-bag squared error loss was plotted against the number of iterations and boosting stopped when the plot bottomed out.

- The fraction of the training set observations randomly selected to propose the next tree in the expansion (out-of-bag) was set to 0.5.

- The shrinkage parameter applied to each tree in the additive expansion of the basis functions was set to 0.001.

4.0 Support Vector Machines (SVM)

- We performed epsilon- SVM regression using linear kernel basis functions to fit the model:

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2}\|\beta\|^2, \qquad (5)$$

$$\text{where } V_\varepsilon(r) = \begin{cases} 0 \quad if\, |r| < \varepsilon, \\ |r| - \varepsilon, otherwise. \end{cases} \qquad (6)$$

Optimization of problem (5& 6) using quadratic programming produces solution functions:

$$\hat{\beta} = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i, \qquad (7)$$

$$f(x) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) K(x, x_i) + \beta_0,$$

Where $\hat{\alpha}_i, \hat{\alpha}_i^*$ are positive coefficients and the kernel function $K(x_i, x_j)$ is a $N \times N$ symmetric and positive definite matrix.

- We set the value of the cost parameter at $C = 0.001$ and allowable error at $\varepsilon = 10$ (tolerable error) using trial and error since automatic tuning of the cost function proved unfeasible on our quad core PC.

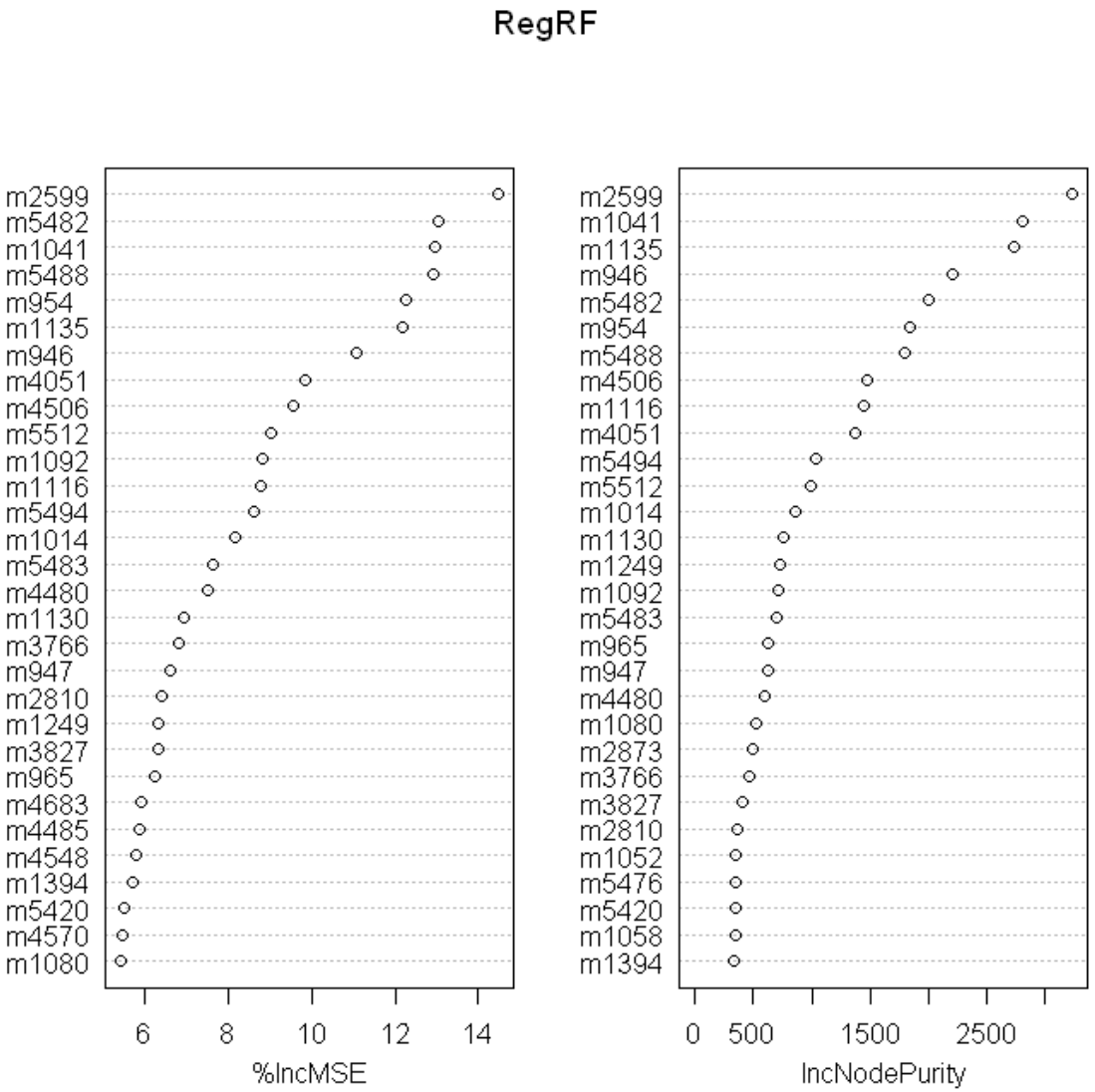
- With 10031 SNPs as predictors, mapping the data to higher dimensional feature spaces using nonlinear kernels would not improve the fit much.

5.0 Results

Random forests

Table 1: Performance of random forest regression of Q against 10031 SNPs based on the Pearson correlation between GEBV and TBV from a 5-fold cross-validation.

Replicate	N	5-fold-CV		
		Correlation	95% LCL	95% UCL
1	439	0.5878	0.5225	0.6455
2	416	0.4866	0.4092	0.5563
3	447	0.4183	0.3382	0.4916
4	490	0.3930	0.3151	0.4651
5	514	0.4448	0.3723	0.5113
Average		**0.4661**	**0.39146**	**0.53396**

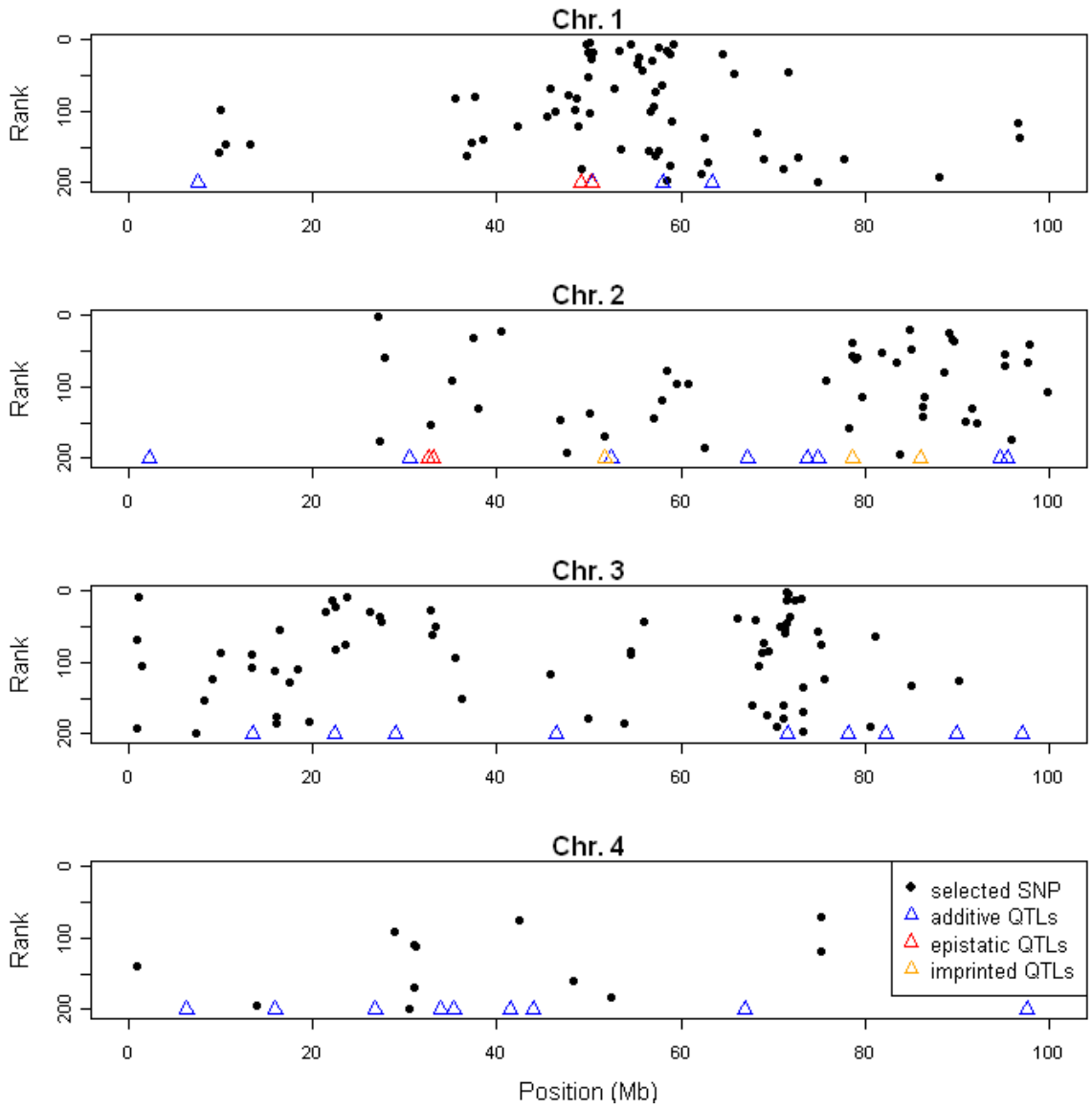


Importance plot for regression of Q on the 10031 SNPs (labelled m1 to m10031).

y-axis =Rank order of importance of each SNP.

x-axis = % decrease in accuracy (%IncMSE)

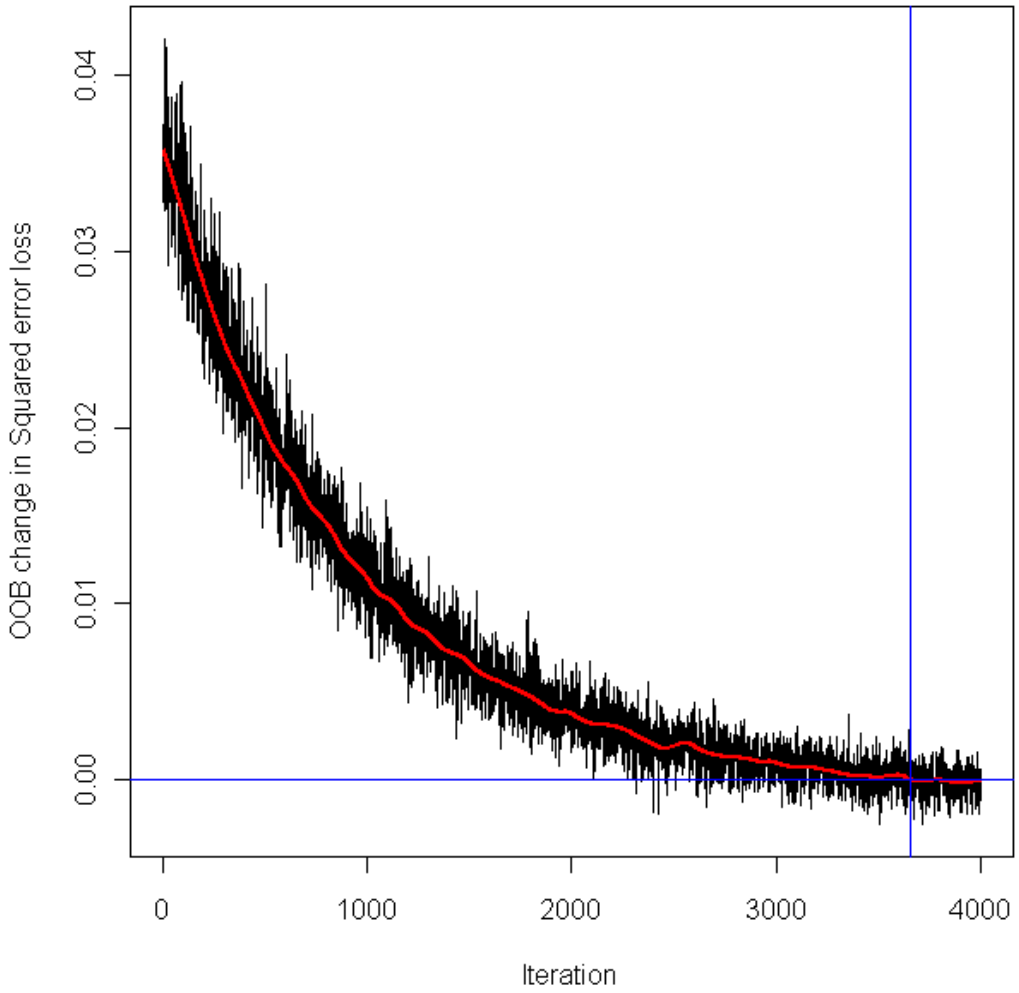
or mean decrease in node impurity (%decIMP) in predicting Q when values of a given SNP are randomly permuted.



Chromosomal positions of the 200 most important SNPs from the pool of 10031 for the quantitative trait Q.

And positions of QTL.

Boosting



The change in squared error loss against the number of iterations for generalized boosted regression of Q on 10031 SNPs. Stop at 3656 iterations.

Black line = training data (100% of all data),
Red line = OOB data (50% of all data).

Table 2: Performance of boosted generalized regression model, using Gaussian errors, squared error loss and regression trees as base learners based on Pearson correlation between GEBV and observed values in the validation datasets used in the 5-fold-CV.

Replicate	N	No. iterations	Correlation	95% LCL	95% UCL
1	439	3242	0.6003	0.5364	0.6567
2	416	3386	0.5183	0.4438	0.5849
3	447	3327	0.4447	0.3666	0.5158
4	490	3353	0.4443	0.3699	0.5124
5	514	3296	0.5057	0.4380	0.5671
Average			**0.4957**	**0.4234**	**0.5610**

Support vector machines

Table 3. Performance of epsilon-support vector machine regression model with the linear kernel, penalty = 0.001 and epsilon=10 based on Pearson correlation between GEBV and observed values in the validation datasets used in the 5-fold-CV.

Replicate	N	Correlation	95% LCL	95% UCL
1	439	0.6100	0.5472	0.6652
2	416	0.5670	0.4975	0.6284
3	447	0.4298	0.3506	0.5022
4	490	0.4117	0.3349	0.4823
5	514	0.4962	0.4276	0.5583
Average		**0.5029**	**0.4316**	**0.5673**

6.0 Conclusions

The correlation between GEBV and TBV was:

- Boosting : 0.547 (95% CL= 0.499-0.591)
- Support vector machines: 0.497 (CL= 0.446-0.545)
- Random forests : 0.483 (CL= 0.431-0.531)

- So, all three methods achieved similar levels of predictive accuracy. Boosting was somewhat more accurate than SVM and random forests.

- All the three methods had reasonable predictive accuracies and can thus be used to predictive quantitative traits using dense SNP markers.

Thank you for listening!