Genet. Res., Camb. (2012), **94**, *pp.* 133–150. doi:10.1017/S001667231200033X

An ensemble-based approach to imputation of moderate-density genotypes for genomic selection with application to Angus cattle

CHUANYU SUN[1]*, XIAO-LIN WU[1,2], KENT A. WEIGEL[1], GUILHERME J. M. ROSA[2,3], STEWART BAUCK[4], BRENT W. WOODWARD[4], ROBERT D. SCHNABEL[5], JEREMY F. TAYLOR[5] AND DANIEL GIANOLA[1,2,3]

[1] *Department of Dairy Science, University of Wisconsin, Madison, WI 53706, USA*
[2] *Department of Animal Sciences, University of Wisconsin, Madison, WI 53706, USA*
[3] *Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706, USA*
[4] *Merial Limited, Duluth, GA 30096, USA*
[5] *Division of Animal Sciences, University of Missouri, Columbia, MO 65211, USA*

(*Received 15 March 2012; revised 15 May 2012; accepted 17 May 2012; first published online 18 July 2012*)

Summary

Imputation of moderate-density genotypes from low-density panels is of increasing interest in genomic selection, because it can dramatically reduce genotyping costs. Several imputation software packages have been developed, but they vary in imputation accuracy, and imputed genotypes may be inconsistent among methods. An AdaBoost-like approach is proposed to combine imputation results from several independent software packages, i.e. Beagle(v3.3), IMPUTE(v2.0), fastPHASE(v1.4), AlphaImpute, findhap(v2) and Fimpute(v2), with each package serving as a basic classifier in an ensemble-based system. The ensemble-based method computes weights sequentially for all classifiers, and combines results from component methods via weighted majority 'voting' to determine unknown genotypes. The data included 3078 registered Angus cattle, each genotyped with the Illumina BovineSNP50 BeadChip. SNP genotypes on three chromosomes (BTA1, BTA16 and BTA28) were used to compare imputation accuracy among methods, and the application involved the imputation of 50K genotypes covering 29 chromosomes based on a set of 5K genotypes. Beagle and Fimpute had the greatest accuracy among the six imputation packages, which ranged from 0·8677 to 0·9858. The proposed ensemble method was better than any of these packages, but the sequence of independent classifiers in the voting scheme affected imputation accuracy. The ensemble systems yielding the best imputation accuracies were those that had Beagle as first classifier, followed by one or two methods that utilized pedigree information. A salient feature of the proposed ensemble method is that it can solve imputation inconsistencies among different imputation methods, hence leading to a more reliable system for imputing genotypes relative to independent methods.

1. Introduction

Single nucleotide polymorphism (SNP) genotyping chips have enabled an era of genomic selection, in which dense SNP genotypes covering the genome are used to predict the genetic merit of candidate individuals or lines for breeding purposes (Meuwissen *et al*., 2001; Heslot *et al*., 2012). In cattle, for example, genomic estimated breeding values (GEBVs) can be predicted with a considerably good accuracy (Saatchi *et al*., 2011). However, commercial moderate density SNP arrays, such as the Ilumina BovineSNP50 Beadchip (Matukumalli *et al*., 2009), are costly, which have limited their applications to males and elite females. Although high predictive accuracy has been documented, e.g. by Weigel *et al*. (2009), using low-density assays with 300–2000 selected SNPs, validity of each of these low-density genotyping panels is usually intrinsic to a specific trait and a given breed. As a cost-effective alternative solution to generating moderate density genotypes, various imputation strategies have been sought. The idea is to genotype candidate animals with a low-density platform comprising equally spaced SNPs, and then to impute moderate-density genotypes via appropriate

* Corresponding author: 1675 Observatory Dr., Department of Dairy Science, University of Wisconsin, Madison, WI 53706, USA. Tel: +6082637824. E-mail: csun28@wisc.edu

statistical models (e.g. Habier *et al*., 2009; Weigel *et al*., 2009; Zhang & Druet, 2010).

Several software packages have been developed for genotype imputation in humans or livestock. Based on the sources of information used to infer missing genotypes, imputation methods can be divided into family-based or population-based, or those that make use of both sources. The family-based approach makes use of linkage and Mendelian segregation rules, and is most accurate for animals having genotyped relatives. The population-based approach utilizes linkage disequilibrium (LD) information between missing SNPs and the observed flanking SNPs, and is well suited for a set of unrelated individuals or for animals whose close ancestors have not been genotyped. In practice, however, choosing an appropriate method is not always an easy decision. One may wish to choose a method that yields the greatest imputation accuracy, but such information is not available before the data at hand are actually analysed. In addition, none of the current methods provide perfect imputation, and imputed genotypes may be inconsistent among programs. Solving such inconsistencies poses another challenge in imputation. From the viewpoint of machine learning, genotype imputation can be considered as a classification problem, and each imputation method can be viewed as an independent classifier. Ensemble learning algorithms (e.g. Polikar, 2006) can be helpful for combining predictions from alternative models, and can yield final classification results that are more robust than those from individual classifiers.

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. Unlike ordinary machine learning approaches, which learn about a sole hypothesis from training data, an ensemble method constructs a set of hypotheses and combines them in the final decision. Dasarathy & Sheela (1978) discussed the problem of partitioning the feature space using two or more classifiers, and this was one of the earliest studies on ensemble systems. A wave of research on ensemble learning started in the 1990s. Hansen & Salamon (1990) showed that the generalization performance of a neural network can be improved by using an ensemble of similarly configured neural networks; Schapire (1990) proved that a strong classifier (i.e. approximately correct) can be generated by combining weak classifiers through boosting, which was the predecessor of AdaBoost algorithms. Thereafter, research in ensemble systems has expanded rapidly, leading to many methods (Polikar, 2006). Within these, AdaBoost (Freund & Schapire, 1996) became one of the most widely used ensemble methods, since it can improve generalization performance relative to individual methods or classifiers (Sewell, 2011). The basic principle of AdaBoost is to combine multiple base classifiers to produce a committee, whose performance is better than that of any of the base classifiers. The latter are trained in sequence using a weighted form of the dataset in which the weights associated with each data point depend on the performance of the previous classifiers. Points that are misclassified by one of the base classifiers are given greater weight when used to train the next classifier in the sequence. Once all classifiers have been trained, their predictions are combined through a weighted majority voting scheme (Bishop, 2006). At present, no reports are available on the application of Adaboost to genotype imputation in animal genetics and breeding.

The objective of this study was to investigate the performance of an ensemble approach to imputing moderate-density SNP genotypes. This approach was used to impute 50K genotypes from 5K genotypes in a registered Angus cattle population.

2. Materials and methods

(i) *Data*

Data were from Merial Limited and consisted of 3078 Angus animals, each genotyped using the Illumina BovineSNP50 BeadChip. Quality control and data editing were carried out by Merial Limited, and we further deleted the individuals and markers that had more than 10 % missing genotypes. The final dataset has 3059 animals and 51 911 SNPs across the whole genome. All animals are sires with known parents, and a pedigree file including 10 163 individuals was created by tracing ancestors. Since all genotyped animals were male, we traced their sire, paternal-grandsire and maternal-grandsire among all genotyped animals, and found that there were 2919 individuals with more than one genotyped relative, and 140 individuals did not have any genotyped relatives.

To assess imputation accuracy, cross-validation was used, with the dataset randomly divided into three approximately equal portions. Two of the portions were used for training the imputation models, and the remaining portion was used for testing imputation accuracy. To compare imputation accuracy among methods, we focused on three representative chromosomes: 1 (longest), 16 (moderate size) and 28 (one of the shortest). After data editing and quality control, there were 3348 SNPs on chromosome 1, 1628 SNPs on chromosome 16 and 944 SNPs on chromosome 28 in the training sets. In the testing sets, there were 357, 192 and 103 SNPs with known genotypes on these three chromosomes, respectively, which corresponded to subsets of 5K (now known as the Illumina BovineLD 7K assay; Boichard *et al*., 2012) genotypes. All of the remaining genotypes for animals in the testing set were treated as 'missing' and were subsequently imputed (Table 1).

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table 1. *Number of animals and number of SNP markers with known genotypes in the training and testing sets*

	Chromosome					
	1		16		28	
Partition	Training	Testing	Training	Testing	Training	Testing
No. of markers	3348	357	1628	192	944	103
No. of animals	2281	777	2281	777	2281	777

(ii) *Imputation programs*

The six imputation software packages used to impute 'missing' genotypes in the testing set were Beagle3.3 (Browning & Browning, 2009), IMPUTE2.0 (Howie *et al*., 2009), fastPHASE1.4 (Scheet & Stephens, 2006), findhap version 2 (VanRaden *et al*., 2011), AlphaImpute (Hickey *et al*., 2011) and Fimpute version 2 (Sargolzaei *et al*., 2011).

The first three packages use population-based approaches. fastPHASE (version1.4) package (fPH) is based on the idea that haplotypes in a population tend to cluster into groups of similar haplotypes over short regions. This package allows membership in haplotype clusters to change as the analysis moves along the chromosome, using a hidden Markov model (HMM) to describe each observed haplotype as a mosaic of a small number of common haplotypes. Beagle3.3 (Bgl) is based on a graphical model that constructs a tree of haplotypes that are present in the reference population, and then summarizes it in a direct acyclic graph (DAG) by joining nodes of the tree based on haplotype similarity. When imputing biallelic markers with alleles A and B in unrelated individuals, for example, posterior genotype probabilities are calculated by summing the probabilities of the HMM states that correspond to each of the AA, AB and BB genotypes. The probability of a missing genotype is computed by averaging posterior genotype probabilities over multiple iterations (Browning & Browning, 2009). This method is attractive because it can adapt to the local haplotype diversity that occurs in the data, similar to fastPHASE, but with a variable number of clusters across a region (Marchini & Howie, 2010). Most HMM-based imputation methods simultaneously estimate missing genotypes and analytically integrate over the unknown phase of SNPs typed in both the study sample and the reference panel. However, IMPUTE2.0 (Imp) proposes to alternately estimate haplotypes at SNPs which are typed in both the study and the reference samples and imputes alleles at SNPs which are not typed in the study sample, but are typed in the reference panel. Separating the phasing and imputation steps allows Imp to place more computational effort on phasing, and the extra computation used in this step is largely balanced by the quick haploid imputation in the step that follows.

In the second group of imputation packages, AlphaImpute (Alp) calculates allele probabilities using segregation analysis based on long-range phasing (Kong *et al*., 2008) and haplotype library imputation. Alp uses information from multiple surrogate parents and a more robust definition of surrogacy using the concepts of cores and tails (Hickey *et al*., 2011). Missing alleles are imputed by matching the allelic probabilities from the segregation analysis to the haplotypes from the phasing step. Hence, Alp is viewed primarily as a family-based imputation package, though it can use information from unrelated animals as well. The findhap version 2 package (fhap) accounts for both population-based and family-based haplotypes in two steps. In the first (population-based) step, it divides chromosomes into blocks of 'x' SNPs and generates a library of haplotype blocks, which are sorted by frequency. Haplotypes from low density panels are searched in the library until a match is found. Unknown alleles are replaced by alleles from the matched haplotype. The animal's second haplotype is obtained by removing its first haplotype from its genotype and matching the second haplotype with haplotypes in the library. In the following step, both pedigree and population methods are used to locate matching haplotypes. Fimpute version 2 (Fimp) reconstructs haplotypes using family information, and uses information from parents, ancestors, progeny and sibs to perform family imputation. Next, it performs population imputation using a haplotype search based on a sliding window approach (i.e. walking along each chromosome using different window sizes).

(iii) *AdaBoost-like ensemble algorithm*

An AdaBoost-like algorithm was designed to combine imputed results from the aforementioned software packages. A wrapper program was used to coordinate individual packages and to implement computations for the proposed ensemble method. Let X be a set of imputed genotypes, and $\boldsymbol{y}$ be a vector of observed ('true') genotypes at a given SNP locus. Define $T=6$ to be the number of independent

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

classifiers (i.e. the imputation software). Given a training set of N individuals, we have $\mathbf{S}=[(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N)]$, where $x_i \in \mathbf{X}=(x_{i1}, x_{i2}, x_{i3} | i=1, 2, \ldots, N)$, $y_i \in \mathbf{y}=(g_1, g_2, g_3)$, and g_1, g_2 and g_3 are the three genotypes at the SNP, in question, for individual i in the training sample.

Initialize: each individual was assigned with an equal weight, $W_1(i)=1/N$, for $i=1, \ldots, N$.

Training: For $t=1, 2, \ldots, T$ classifiers

1. Call classifier t, which in turn generates hypothesis h_t (i.e. inferred haplotypes and genotypes in the training set).
2. Calculate the error of h_t:

$$\varepsilon_t = \frac{\sum_{i=1}^{N} W_t(i) I(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} W_t(i)},$$

where $I(h_t(x_i) \neq y_i)$ is an indicator function that is equal to 1 when $h_t(x_i) \neq y_i$ and 0 otherwise. Looping is aborted if $\in_t > 1/2$.

3. Set $\beta_t = \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
4. Update the weight distribution $W_t(i)$ for next classifier as

$$W_{t+1}(i) = W_t(i) \exp(\beta_t I(h_t(x_i) \neq y_i)).$$

Testing: In the testing set, each 'unknown' genotype is classified via the so-called 'weighted majority voting'. Briefly, the wrapper program:

1. Computes the total vote received by each genotype (class)

$$v_i = \sum_{t=1}^{T} \{\beta_t I'(h_t(x_i) = g_j)\}, \quad \text{for} \quad j=1, 2, 3,$$

where $I'(h_t(x_i)=g_j)$ is an indicator function that is equal to 1 when $h_t(x_i)=g_j$ and 0 otherwise.
2. Assigns the genotype (class) that received the largest total vote as the final ('putative') genotype.

Above, the algorithm maintains a weighted distribution $W_t(i)$ of training samples x_i, for $i=1, \ldots, N$, from which a sequence of training data subsets S_t is chosen for each consecutive classifier (package) t. Initially, the distribution of weights is uniform, meaning that all samples contribute equally to the error rate. Next, the logit β_t of the rate of correctly classified samples is calculated for classifier t. A higher β_t is an indicator of better performance; for instance, when $\varepsilon_t=0.5$, β_t takes the value 0, and increases as $\varepsilon_t \to 0$. Then, the distribution update rule is as follows: the weights of samples that are classified correctly by the current method are unchanged, whereas the weights of the misclassified instances are increased by a factor of e^{β_t}. Hence, by iterating classifiers, the algorithm tends to focus on increasingly difficult samples. At the end, a weighted majority voting is used, and the class (genotype) that receives the largest total vote from all classifiers is the ensemble decision. This voting scheme is like that used in AdaBoosting.

(iv) *Bootstrap sampling and parallel computing*

Bootstrapping was used to generate empirical confidence intervals of imputation accuracy for the six packages and for the ensemble systems as well. For each method, 50 replicates were created by drawing random samples with replacement from the original testing set, each conducted on the genotype data for one of the three chromosomes, and the size of each bootstrap sample equalled the size of the original testing set. Finally, summary statistics were computed from the 50 bootstrap samples. Note that the distribution is conditional on the training set.

Given six independent packages, there were $6!=720$ combinations, each defining a unique ensemble system. The computing task was formidable. For example, given this design, there were a total of $(720+6) \times 50 \times 3 = 1\,08\,900$ independent jobs. Hence, distributed high-throughput computing solutions were utilized, and these jobs were submitted to run on the University of Wisconsin Condor Systems and Open Science Grid (Wu *et al.*, 2012).

(v) *Evaluation of imputation accuracy*

Genotype error was scored as 0 when the imputed and observed marker types were identical or 1 for otherwise. In other words, only when the two imputed alleles were the same as the observed two alleles was regarded as correct. Here, the genotypes 'A/B' and 'B/A' were considered to be the same. Error counting only considered markers/animals where observed marker types were not missing in the original non-imputed dataset. The error rate was calculated as the total number of errors divided by the number of imputed loci. This gives the number of falsely predicted genotypes. The imputation accuracy is one minus the error rate. Zhang & Druet (2010) and Dassonneville *et al.* (2011) reported other ways of measuring error rate.

3. Results and discussion

(i) *Comparing imputation accuracy among software packages*

The six packages varied in imputation accuracy when evaluated on Angus chromosomes 1, 16 and 28 (Table 2). Bgl had the greatest imputation accuracy on all three chromosomes, followed by Fimp and fhap. On chromosome 1, for example, mean imputation accuracy obtained with Bgl was 0·9858 and that obtained with each of the remaining five packages ranged from 0·9084 (Alp) to 0·9788 (Fimp). Similar

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table 2. *Summary statistics of the bootstrap distribution of imputation accuracy obtained using each of the six software packages on chromosomes 1, 16 and 28*†‡*

	Method	Min	Median	Max	Mean	SD
Chrom_1	Bgl	0·9844	0·9858	0·9869	0·9858	0·0006
	Imp	0·9346	0·9375	0·9403	0·9375	0·0013
	fPh	0·9253	0·9286	0·9314	0·9286	0·0014
	Fhap	0·9619	0·9649	0·9681	0·9648	0·0014
	Alp	0·8972	0·9083	0·9164	0·9084	0·0039
	Fimp	0·9771	0·9789	0·9803	0·9788	0·0007
Chrom_16	Bgl	0·9826	0·9836	0·9851	0·9837	0·0006
	Imp	0·9291	0·9323	0·9358	0·9325	0·0012
	fPh	0·9173	0·9207	0·9237	0·9208	0·0015
	Fhap	0·9498	0·9537	0·9580	0·9536	0·0018
	Alp	0·8968	0·9098	0·9159	0·9092	0·0041
	Fimp	0·9701	0·9728	0·9751	0·9728	0·0010
Chrom_28	Bgl	0·9691	0·9712	0·9740	0·9712	0·0011
	Imp	0·8824	0·8890	0·8938	0·8887	0·0024
	fPh	0·8618	0·8679	0·8718	0·8677	0·0021
	Fhap	0·9293	0·9355	0·9401	0·9354	0·0025
	Alp	0·8852	0·8937	0·9022	0·8937	0·0039
	Fimp	0·9542	0·9592	0·9613	0·9589	0·0015

*Bgl, Beagle3.3; Imp, IMPUTE2.0; fPh, fastPHASE1.4; fhap, findhap version 2; Alp, AlphaImpute; Fimp, Fimpute version 2.
†Min, minimum value; Max, maximum value; SD, standard deviation.
‡Accuracy obtained from 50 boostrap replicates for each imputation package.

patterns were observed on the other two chromosomes: imputation accuracy varied from 0·9092 (Alp) to 0·9837 (Bgl) on chromosome 16, and from 0·8677 (fPh) to 0·9712 (Bgl) on chromosome 28.

Bgl, Imp and fPh, which impute missing SNP genotypes using LD information, were primarily developed for humans (Li *et al*., 2009; Marchini & Howie, 2010), but have been applied to animals as well (Weigel *et al*., 2010; Calus *et al*., 2011; Hayes *et al*., 2011; Johnston & Kistemaker, 2011; Saatchi *et al*., 2011). Within the three population-based packages, we found that Bgl consistently yielded the best imputation accuracy on all three chromosomes. In the present study, Bgl was used with the assumption that all animals were unrelated, although this package could handle parent–offspring trios and parent–offspring pairs. The latter option, however, was not used because making use of such information would consume a high amount of memory with our data size. Nevertheless, imputation accuracy was high even when relationship information was ignored. This is in agreement with observations by Browning & Browning (2009).

Alp, Fimp and fhap can utilize both pedigree and LD information in imputation. Within these, imputation accuracy was the greatest with Fimp and the poorest with Alp. Overall, these three packages did not show any obvious improvement in imputation accuracy over the population-based packages. Although Fimp and fhap yielded better imputation accuracy than two of the population-based packages, Imp and fPH, none outperformed Bgl, which had the greatest imputation accuracy overall. Our results imply that simple, sequential utilization of population and family information, as we did with Fimp and fhap, does not necessarily produce better imputation accuracy than population-based methods. An explanation may be the existence of a considerable amount of LD in this Angus population; Bgl can capture familial information using long identical haplotypes (from close relatives) and is efficient with both close and distant relatives. Hence, well-designed population-based algorithms could predict the missing genotypes fairly well.

We observed that imputation accuracy was positively associated with chromosome size. Chromosome 1 is the longest, chromosome 16 has moderate size and chromosome 28 is the shortest of the chromosomes studied, and imputation accuracy was greatest for chromosome 1, intermediate for chromosome 16 and poorest for chromosome 28. This association with size may be due to the fact that longer chromosomes harbour more markers, and hence provide more information for inferring unknown haplotypes and imputing missing genotypes. Another reason, possibly related, is that imputation accuracy suffers at the beginning and end of the chromosomes. In longer chromosomes, these two problem regions are

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

relatively less important than in shorter chromosomes, leading to higher accuracy. Although the 7K has been specifically designed to overcome this issue by having more low density markers at the ends, the issue seems to remain. Weigel *et al*. (2010) reported mean imputation accuracy from 80 to 95% when animals were genotyped with a medium-density panel comprising 2000–4000 SNPs; less than 80% when animals were genotyped for 1000 SNPs or less, and greater than 95% when animals were genotyped for more than 8000 SNPs. However, their study was performed with a single chromosome.

Computational requirements are also one important issue in imputation. Bgl, Imp, fPH and Alp run on a per chromosome base, whereas fhap and Fimp are both parallel programs that can impute all chromosomes simultaneously. In this study, all jobs were submitted to run in parallel in the University of Wisconsin Condor Systems and the Open Science Grid (OSG). Each chromosome could be imputed by a different CPU, so we did not record the specific computing time for each of the jobs. In general, Fimp and fhap consume the least computing time due to their parallel nature. Alp is faster than Bgl and fPH, and fPH is the slowest software. Take chromosome 1 as the example, the imputation took around 1 week with fPh, one day or so with Bgle, and not more than 1 h with Fimp and fhap. Memory consumption also varied dramatically with these software packages. Imp needs to break chromosomes into pieces of more manageable size, hence it does not pose computing time and memory problems for parallelized running. In contrast, Bgl had a memory problem – it required more than 6G memory for chromosome 1 in this study.

(ii) *Comparing imputation accuracy between ensemble methods and individual packages*

As none of the software packages provides perfect imputation, combining results from two or more packages may bring an improvement in accuracy. As previously noted, while many animals have moderate-density genotypes for both parents, the proportion of correctly imputed genotypes could increase with more relatedness between genotyped ancestors and target animals (Weigel *et al*., 2010; Johnston & Kistemaker, 2011). For this reason, a two-step approach has been proposed, in which animals with genotyped parents (or other close ancestors) are processed first using a family-based method, and animals lacking such information are processed subsequently using a population-based algorithm (Druet & Georges, 2010). However, as indicated by our results, simply utilizing family- and population-based information sequentially will not necessarily give the best predictions. Further, Weigel *et al*. (2010) suggested that such an approach could be considered as a form of 'boosting', in which two or more complementary models, each of which treats a significant percentage of the data optimally, are implemented jointly to solve the imputation problem.

AdaBoost, which is a more general version of the original boosting algorithm (Freund & Schapire, 1996), is flexible and can ensemble results from various weak classifiers, resulting in an improved accuracy. The ensemble method proposed here resembled AdaBoost, yet it had some slight differences. Like AdaBoost, our ensemble method computed weights sequentially for all the individual imputation packages, and combined results through weighted majority voting of the genotype classes predicted by individual packages. The results indicated that the proposed ensemble method was better than each of the individual imputation methods (Fig. 1). The extent of improvement, however, was small, possibly because all six packages imputed 'missing' genotypes with high accuracy, so there was not much space for further improvement.

Within the 720 unique ensemble systems, imputation accuracies of the top five ensemble systems, evaluated on each of the three chromosomes, were compared with those of each of the individual packages (Fig. 1). These ensemble systems performed similarly to each other, and all were at least as good as each of the six individual imputation packages. Their performance over the 50 bootstrap samples is shown in Fig. 2. We observed that imputation accuracy varied with the order of the software packages in the ensemble system. For each of the three chromosomes, all top 120 ensemble systems with the highest accuracy of imputation had Bgl as the first classifier (Appendix Tables A1–A3). The ranks of the top 120 ensemble systems, however, differed by chromosome. For example, ensemble system 67 (Bgl-Alp-Fimp-Imp-fPH-fhap, Appendix Table A4) ranked second with data from chromosomes 1 and 28, but it ranked first with data on chromosome 16. This consistency was also true with ensemble system 61 (Bgl-Alp-fhap-Imp-fPH-Fimp): it ranked fifth on chromosome 1, fourth on chromosome 16 and first on chromosome 28. Nevertheless, there were also ensemble systems that showed relatively large changes in rank between chromosomes. For example, ensemble system 109 (Bgl-Fimp-Alp-Imp-fPH-fhap) ranked fourth on chromosome 1 and third on chromosome 16, but it was 38th on chromosome 28. Ensemble systems with Fimp and Bgl as the first two classifiers also had high imputation accuracy (data not presented). The lowest imputation accuracy was observed with fPH and Imp appearing as the first two classifiers (data not presented). Similarly, Johnston & Kistemaker (2011) reported varied imputation accuracy arising from different sequences of imputation packages in their

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

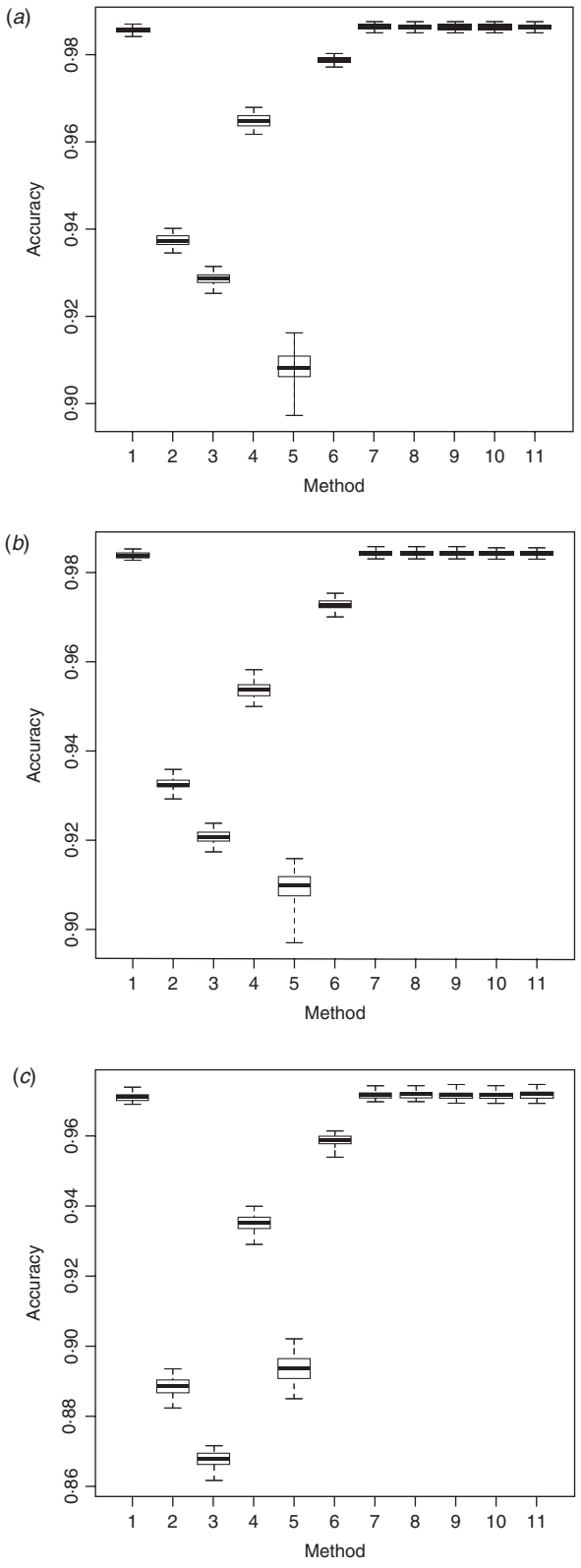

Fig. 1. Box plots of imputation accuracy on (a) chromosome 1, (b) chromosome 16 and (c) chromosome 28, obtained using six imputation software packages and five ensemble methods. Results are obtained from 50 bootstrap replicates. For x-axis labels, 1 = 'Beagle3.3'; 2 = 'IMPUTE2.0'; 3 = 'fastPHASE1.4'; 4 = 'findhap version 2'; 5 = 'AlphaImpute'; 6 = 'Fimpute version 2'; 7 - 11 = five ensemble systems.

two-step approach. They first conducted imputation using Fimp, and then exported results to either fhap or Bgl for a second-step imputation. They found that Fimp plus Bgl could provide more accurate imputation than the team represented by Fimp and fhap.

Our results indicate that an ensemble method starting with the best individual classifier (i.e. Bgl) could have the best overall performance. Also, alternating population-based and family-based approaches could enhance imputation as well. Therefore, optimal ensemble systems, as supported by the present data, turned out to be those starting with Bgl, followed by one or two of the packages that can use pedigree information for imputation (e.g. fhap, Fimp and Alp). For examples, ensemble systems 67 and 61 satisfied the above-mentioned feature, and ranked within the top five when evaluated with genotype data on the three chromosomes.

We proposed an ensemble-based imputation method that involves using several imputation software packages jointly. Alternatives would include either developing a single method that combines all sources of information in some optimal manner, or perhaps using a pair of existing methods that exploit complementarity between sources. We did not explore the latter approach because our study focused on testing whether ensemble methods behave as expected from theory, in comparison with some commonly used imputation software packages. Boosting algorithms have been developed for enhancing weak learners, so when the extant methods are strong classifiers, there is little room for improvement. This was confirmed in our study, the first of its kind in the context of genotype imputation. We conjecture that the same situation would hold if the comparison involved a method that can utilize all sources of information. Our study showed that the proposed ensemble method can perform as well as, if not better, than any of the individual imputation methods. At the same time, the ensemble method provides a solution to solving inconsistencies among different imputation methods. In order to decrease computational cost, it is not necessary to include all the six imputation software packages in practice. The paper recommended some general conclusions on how to combine different independent imputation packages in ensemble-based methods.

(iii) *An application: imputation of moderate-density genotypes in Angus cattle*

Based on a 5K-genotype panel, moderate-density (50K) genotypes on 29 chromosomes were imputed for 3078 animals using the aforementioned six imputation packages and five ensemble systems. All five selected ensemble systems had Bgl and Alp as the first

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

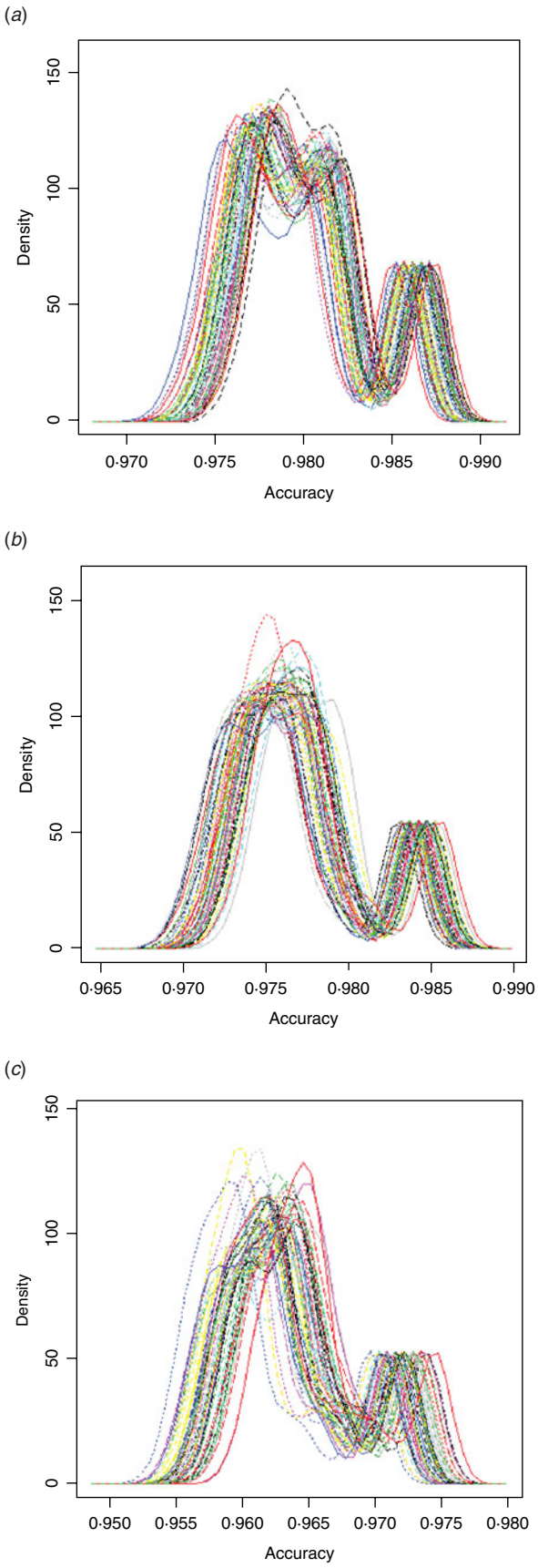

Fig. 2. Kernel density plots of imputation accuracy for 720 ensemble methods obtained on (a) chromosome 1, (b) chromosome 16 and (c) chromosome 28.

two classifiers, and were as follows: (1) Bgl-Alp-fhap-Imp-fPH-Fimp, (2) Bgl-Alp-Fimp-Imp-fPH-fhap, (3) Bgl-Alp-Fimp-fPH-Imp-fhap, (4) Bgl-Alp-Imp-fhap-fPH-Fimp and (5) Bgl-Alp-Imp-Fimp-fPH-fhap. Imputing accuracies using the six packages and the five ensemble systems are illustrated in Fig. 3. The five ensemble systems gave similar imputation results, and were consistently better than each of the six imputation packages. Among the 29 autosomes, imputation accuracy ranged from 0·9715 (chromosome 28) to 0·9855 (chromosome 1) with the five ensemble systems, and it varied from 0·8869 (Alp, chromosome 10) to 0·9853 (Bgl, chromosome 1) with the six independent packages.

We did not observe a 100 % imputation success rate in any of the 29 chromosomes. This could be due to, for example, small training sample size, density of markers, or degree of genetic similarity between the training and testing samples. Genotyping error rate may also be a crucial factor that affects imputation accuracy. As there are genotype errors in the reference populations, using these genotypes in training would cause errors in imputation results. Genotyping errors might also be present in the testing set.

It would seem that attaining 100 % imputation accuracy may not be possible in practice with current genotyping technologies and imputation methods. Zhang & Druet (2010) reported error rates of 3–4 % using DAGPHASE (Druet & Georges, 2010). Daetwyler *et al.* (2011) reported slightly higher error rates with their implementation of the long-range phasing algorithm, possibly because they used a smaller reference population. Other factors that may be crucial for enhancing imputation accuracy are as follows. First, it is important that there is a strong genetic similarity between the training and testing populations, since imputation accuracy is likely to depend on the genetic distance of target individuals from the reference population (Zhang & Druet, 2010). If an individual does not have parents or relatives in the training sample, and if there is no intervening recombination, the chance of observing a haplotype of this individual in the training set would be small. Next, given considerable genetic similarity between training and testing populations, the training set should be large enough to capture all of the haplotypes in the testing set. If a target haplotype is encountered which has not been previously observed in the training sample, the imputation of missing genotypes is unlikely to be accurate. Using a large training sample is also important to ensure that rare alleles are captured and accurately imputed into target individuals. Finally, a sufficient number of markers is essential for accurate imputation using population-based methods, to assure that there is substantial linkage disequilibrium between markers. Otherwise, population-based algorithms such as

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

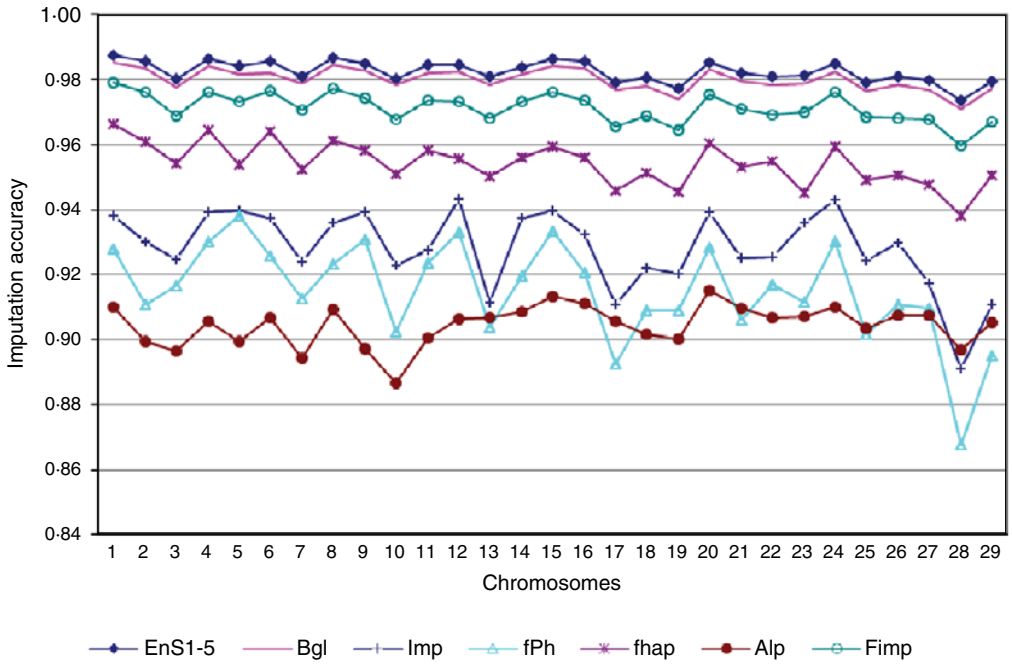

Fig. 3. Comparison of imputation accuracy evaluated on 29 autosomes in registered Angus cattle using 6 independent imputation packages and 5 ensemble systems. For EnS1-5, the figure gives the average accuracy of the 5 ensembles.

those implemented in Bgl, Imp and fPH will perform poorly.

Again, we observed a slight decrease in imputation accuracy for the shorter chromosomes, but this trend was not as evident as in the previous applications involving chromosomes 1, 16 and 28 only. These results were obtained as the averages of 50 replicates for each chromosome, but the results in this section were obtained from a single analysis. This could partially explain the difference. As no replication was performed, chance could contribute non-trivially to imputation results, leading to random fluctuation in accuracy. It is possible that the trend could become more evident if the results were obtained as averages across a large number of bootstrap replicates.

4. Conclusions

Genotype imputation can be viewed as a classification problem. Several imputation methods (i.e. software packages) are available, but results may be inconsistent among them. Ensemble methods can be used to solve such inconsistencies, and thus further improve imputation accuracy. This was corroborated in our study. The proposed ensemble method resembles AdaBoost, in that weights for each of the classifiers are computed sequentially and imputed genotypes are decided by weighted majority voting. The idea is intuitive: classifiers that have a good performance during training are rewarded with higher voting weights than the others. Our ensemble systems combined results from six imputation packages: Bgl, Imp, fPH, Alp, fhap and Fimp. In this set, Bgl and Fimp had the highest imputation accuracy. The proposed ensemble systems improved imputation accuracy in our data, but the degree of improvement depended on the order of these classifiers in the ensemble systems. The best ensemble systems were those with Bgl as the first classifier, followed by one or two software packages that used pedigree information during imputation. Rotating different types of imputation packages in the ensemble systems is desirable, because training by consecutive classifiers may be better geared towards increasingly hard-to-classify instances (Polikar, 2006).

Further improvements through adjustment of the proposed ensemble method may be possible. First, one may form a committee of classifiers with higher diversity, each focusing on a different scenario guiding imputation. This is an essential idea of AdaBoost, which works well provided that each classifier can produce an imputation that is slightly better than a random guess. We have included two types of imputation packages, i.e. family-based and population-based. Some more options include imputation based on population frequencies only (a weak classifier) and imputation based on posterior probabilities of unknown genotypes given observed phenotypes and prior information about the genotypes (also a weak classifier). The latter two options, however, were not investigated, because the six packages we used provided relatively accurate imputation, and including these two weak classifiers would have made little difference in imputation accuracy. Also, individual

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

packages can be modified so that a set of classifiers can be trained more efficiently and adaptively, but this may not always be possible due to the lack of availability of source code. Nevertheless, there are some ensemble methods that do not require modification of each independent imputation package, such as stacked generalization (Wolpert, 1992; Polikar, 2006) or mixture of experts (Jacobs *et al*., 1991; Jordan & Jacobs, 1994). These two ensemble methods can use the outputs of a set of individual classifiers as inputs to a second level meta-classifier, which then learns the mapping between the ensemble outputs and the correct classes. These methods may be worth investigating in future studies.

Finally, while AdaBoost is well known for its capacity to boost the classification performance of a set of weak classifiers, we proposed an AdaBoost-like ensemble algorithm for combining results from variable imputation methods (packages), each of which may not necessarily be weak classifiers. Given the fact that some of the current methods, if not all, produce highly accurate imputation results, the scope for improving imputation accuracy may be limited for ensemble-based systems. On the other hand, because no independent method can make a perfect imputation, the proposed ensemble provides a more-robust system for solving inconsistencies among existing imputation methods.

This research was supported by the Wisconsin Agriculture Experiment Station, and a Genomic Selection Grant by Merial Ltd. JFT was supported by National Research Initiative grants number 2008-35205-04687, 2008-35205-18864 and 2009-35205-05099 from the USDA Cooperative State Research, Education and Extension Service, and grant number 2009-65205-05635 from the USDA Agriculture and Food Research Initiative.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Boichard, D., Chung, H., Dassonneville, R., David, X., Eggen, A., Fritz, S., Gietzen, K. J., Hayes, B. J., Lawley, C. T., Sonstegard, T. S., Van Tassell, C. P., Vanraden, P. M., Viaud-Martinez, K. A., Wiggans, G. R. & for the Bovine LD Consortium. (2012). Design of a bovine low-density SNP array optimized for imputation. *PLoS One* **7**, e34130.

Browning, B. L. & Browning, S. R. (2009). A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American Journal of Human Genetics* **84**, 210–223.

Calus, M. P. L., Veerkamp, R. F. & Mulder, H. A. (2011). Imputation of missing single nucleotide polymorphism genotypes using a multivariate mixed model framework. *Journal of Animal Science* **89**, 2042–2049.

Daetwyler, H. D., Wiggans, G. R., Hayes, B. J., Woolliams, J. A. & Goddard, M. E. (2011). Imputation of missing genotypes from sparse to high density using long-range phasing. *Genetics* **189**, 317–327.

Dasarathy, B. V. & Sheela, B. V. (1978). Composite classifier system design: concepts and methodology. *Proceedings of the IEEE* **67**, 708–713.

Dassonneville, R., Brøndum, R. F., Druet, T., Fritz, S., Guillaume, F., Guldbrandtsen, B., Lund, M. S., Ducrocq, V. & Su, G. (2011). Effect of imputing markers from a low-density chip on the reliability of genomic breeding values in Holstein populations. *Journal of Dairy Science* **94**, 3679–3686.

Druet, T. & Georges, M. (2010). A hidden Markov model combining linkage and linkage disequilibrium information for haplotype reconstruction and QTL fine mapping. *Genetics* **184**, 789–798.

Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148–156.

Habier, D., Fernando, R. L. & Dekkers, J. C. M. (2009). Genomic selection using low-density marker panels. *Genetics* **182**, 343–353.

Hansen, L. K. & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, 993–1001.

Hayes, B. J., Bowman, P. J., Daetwyler, H. D., Kijas, J. W. & van der Werf, J. H. J. (2011). Accuracy of genotype imputation in sheep breeds. *Animal Genetics* **43**, 72–80.

Heslot, N., Yang, H. P., Sorrells, M. E. & Jannink, J. L. (2012). Genomic selection in plant breeding: a comparison of models. *Crop Science* **52**, 146–160.

Hickey, J. M., Kinghorn, B. P., Tier, B., Wilson, J. F., Dunstan, N. & van der Werf, J. H. J. (2011). A combined long-range phasing and long haplotype imputation method to impute phase for SNP genotypes. *Genetics Selection Evolution* **43**, 12.

Howie, B. N., Donnelly, P. & Marchini, J. (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genetics* **5**, e1000529.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation* **3**, 79–87.

Johnston, J. & Kistemaker, G. (2011). Success rate of imputation using different imputation approaches. Available at http://www.cdn.ca/Articles/GEBAPR2011/Success%20rate%20of%20imputation%20-%20Jarmila%20Johnston.pdf (accessed 7 March 2012).

Jordan, M. J. & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* **6**, 181–214.

Kong, A., Masson, G., Frigge, M. L., Gylfason, A., Zusmanovich, P., Thorleifsson, G., Olason, P. I., Ingason, A., Steinberg, S., Rafnar, T., Sulem, P., Mouy, M., Jonsson, F., Thorsteinsdottir, U., Gudbjartsson, D. F., Stefansson, H. & Stefansson, K. (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature Genetics* **40**, 1068–1075.

Li, Y., Willer, C., Sanna, S. & Abecasis, G. (2009). Genotype imputation. *Annual Review of Genomics and Human Genetics* **10**, 387–406.

Marchini, J. & Howie, B. (2010). Genotype imputation for genome-wide association studies. *Nature Review Genetics* **11**, 499–511.

Matukumalli, L. K., Lawley, C. T., Schnabel, R. D., Taylor, J. F., Allan, M. F., Heaton, M. P., O'Connell, J., Moore, S. S., Smith, T. P., Sonstegard, T. S. & Van Tassell, C. P. (2009). Development and characterization of a high density SNP genotyping assay for cattle. *PLoS One* **4**, e5350.

Meuwissen, T. H. E., Hayes, B. J. & Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* **157**, 1819–1829.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* **6**, 21–45.

Saatchi, M., McClure, M. C., McKay, S. D., Rolf, M. M., Kim, J. W., Decker, J. E., Taxis, T. M., Chapple, R. H., Ramey, H. R., Northcutt, S. L., Bauck, S., Woodward, B., Dekkers, J. C. M., Fernando, R. L., Schnabel, R. D., Garrick, D. J. & Taylor, J. F. (2011). Accuracies of genomic breeding values in American Angus beef cattle using K-means clustering for cross-validation. *Genetics Selection Evolution* **43**, 40.

Sargolzaei, M., Chesnais, J. P. & Schenkel, F. S. (2011). FImpute – An efficient imputation algorithm for dairy cattle populations. *Journal of Animal Science*. **89** (E-Suppl. 1)/*Journal of Dairy Science* **94**(E-Suppl. 1): 421 (abstr. 333).

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning* **5**, 197–227.

Scheet, P. & Stephens, M. (2006) A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *American Journal of Human Genetics* **78**, 629–644.

Sewell, M. (2011). Ensemble learning. Available at http://www-typo3.cs.ucl.ac.uk/fileadmin/UCL-CS/research/Research_Notes/RN_11_02.pdf (accessed 7 March 2012).

VanRaden, P. M., O'Connell, J. R., Wiggans, G. R. & Weigel, K. A. (2011). Genomic evaluations with many more genotypes. *Genetics Selection Evolution* **43**, 10.

Weigel, K. A., de los Campos, G., Gonzalez-Recio, O., Naya, H., Wu, X. L., Long, N., Rosa, G. J. M. & Gianola, D. (2009). Predictive ability of direct genomic values for lifetime net merit of Holstein sires using selected subsets of single nucleotide polymorphism markers. *Journal of Dairy Science* **92**, 5248–5257.

Weigel, K. A., Van Tassell, C. P., O'Connell, J. R., VanRaden, P. M. & Wiggans, G. R. (2010). Prediction of unobserved single nucleotide polymorphism genotypes of Jersey cattle using reference panels and population-based imputation algorithms. *Journal of Dairy Science* **93**, 2229–2238.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks* **5**, 241–259.

Wu, X-L., Hayrettin, O., Duan, H., Bessinger, T., Bauck, S., Woodward, B., Rosa, G. J. M., Weigel, K. A., de Leon, N., Taylor, J. F. & Gianola, D. (2012). Parallel-BayesCpC on OSG: grid-enabled high-throughput computing for genomic selection in practice. *International Plant & Animal Genome 2012* (http://www. intlpag.org/web/index.php/abstracts/poster-abstracts).

Zhang, Z. & Druet, T. (2010). Marker imputation with low-density marker panels in Dutch Holstein cattle. *Journal of Dairy Science* **93**, 5487–5494.

Appendix A. Summary statistics of imputation accuracy for 720 ensemble systems

Table A1. *Evaluated on bovine chromosome 1 (BTA1)*

BTA1	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	99	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	67	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	92	0·9852	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	109	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	61	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	53	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	68	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	75	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	85	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	116	0·9851	0·9860	0·9865	0·9869	0·9874	0·9864	0·0006
	102	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	110	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	51	0·9851	0·9860	0·9865	0·9869	0·9875	0·9864	0·0006
	101	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	100	0·9851	0·9860	0·9865	0·9869	0·9874	0·9864	0·0006
	69	0·9851	0·9861	0·9864	0·9869	0·9874	0·9864	0·0006
	91	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	62	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	86	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	115	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	11	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	97	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	78	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	95	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	111	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	77	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	98	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	21	0·9851	0·9860	0·9865	0·9869	0·9874	0·9864	0·0006
	119	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	9	0·9851	0·9860	0·9865	0·9869	0·9874	0·9864	0·0006

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A1. (*Cont.*)

BTA1	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	70	0·9851	0·9860	0·9864	0·9868	0·9875	0·9864	0·0006
	71	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	76	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	94	0·9851	0·9860	0·9865	0·9868	0·9875	0·9864	0·0006
	93	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	113	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	112	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	65	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	63	0·9850	0·9860	0·9864	0·9868	0·9875	0·9864	0·0006
	117	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	118	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	54	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	15	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	89	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	96	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	120	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	87	0·9851	0·9860	0·9864	0·9868	0·9875	0·9864	0·0006
	19	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	107	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	105	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	73	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	72	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	20	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	114	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	74	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	64	0·9850	0·9859	0·9864	0·9868	0·9875	0·9864	0·0006
	66	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	52	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	103	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	104	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	88	0·9851	0·9859	0·9864	0·9868	0·9875	0·9864	0·0006
	90	0·9851	0·9860	0·9864	0·9869	0·9875	0·9864	0·0006
	50	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	49	0·9850	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	23	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	108	0·9851	0·9859	0·9864	0·9869	0·9874	0·9864	0·0006
	106	0·9850	0·9859	0·9864	0·9869	0·9875	0·9864	0·0006
	14	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	7	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	13	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	17	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	8	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	18	0·9851	0·9859	0·9864	0·9869	0·9874	0·9864	0·0006
	22	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	24	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	12	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	81	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	83	0·9851	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	16	0·9851	0·9860	0·9864	0·9869	0·9874	0·9864	0·0006
	10	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	80	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	79	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	84	0·9851	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	57	0·9850	0·9859	0·9864	0·9868	0·9875	0·9864	0·0006
	59	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	82	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	56	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	6	0·9851	0·9860	0·9864	0·9868	0·9874	0·9864	0·0006
	3	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	5	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	4	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	55	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A1. (*Cont.*)

BTA1	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	2	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	1	0·9850	0·9859	0·9864	0·9868	0·9874	0·9864	0·0006
	58	0·9850	0·9859	0·9864	0·9868	0·9875	0·9863	0·0006
	60	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	37	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	44	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	43	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	38	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	39	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	33	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	45	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	35	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	41	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	47	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	32	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	31	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	29	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	26	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	25	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	30	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	28	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	27	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	42	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	48	0·9850	0·9859	0·9864	0·9868	0·9874	0·9863	0·0006
	40	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	46	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	36	0·9850	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006
	34	0·9849	0·9859	0·9863	0·9868	0·9874	0·9863	0·0006

Table A2. *Evaluated on bovine chromosome 16 (BTA16)*

BTA16	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	67	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	99	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	109	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	61	0·9831	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	69	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	68	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	75	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	85	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	63	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	70	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	53	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	62	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	91	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	97	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	116	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	93	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	72	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	71	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	92	0·9831	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	111	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	101	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	117	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	87	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	98	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	51	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	115	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A2. (*Cont.*)

BTA16	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	110	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	100	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	120	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	77	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	65	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	114	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	113	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	86	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	94	0·9831	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	118	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	96	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	102	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	66	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	119	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	95	0·9831	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	112	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	103	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	105	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	90	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	64	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	21	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	89	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	9	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	11	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	15	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	107	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	74	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	88	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	49	0·9830	0·9838	0·9841	0·9846	0·9855	0·9842	0·0006
	78	0·9831	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	73	0·9830	0·9838	0·9841	0·9846	0·9855	0·9842	0·0006
	76	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	19	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	20	0·9830	0·9838	0·9841	0·9846	0·9856	0·9842	0·0006
	50	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	104	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	59	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	7	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	54	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	14	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	81	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	8	0·9830	0·9838	0·9841	0·9847	0·9855	0·9842	0·0006
	13	0·9830	0·9838	0·9841	0·9846	0·9855	0·9842	0·0006
	108	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	57	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	83	0·9831	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	79	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	52	0·9830	0·9838	0·9841	0·9847	0·9856	0·9842	0·0006
	106	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	23	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	80	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	17	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	56	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	22	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	55	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	24	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	12	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	5	0·9829	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	16	0·9830	0·9838	0·9840	0·9846	0·9854	0·9842	0·0006
	84	0·9830	0·9837	0·9840	0·9846	0·9855	0·9842	0·0006
	6	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	10	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	1	0·9829	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A2. (*Cont.*)

BTA16	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	58	0·9830	0·9837	0·9840	0·9846	0·9855	0·9842	0·0006
	2	0·9829	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	60	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	18	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	82	0·9830	0·9838	0·9840	0·9847	0·9855	0·9842	0·0006
	3	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	4	0·9830	0·9838	0·9840	0·9846	0·9855	0·9842	0·0006
	43	0·9829	0·9838	0·9840	0·9846	0·9855	0·9841	0·0006
	33	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	35	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	32	0·9830	0·9838	0·9840	0·9846	0·9855	0·9841	0·0006
	31	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	44	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	37	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	29	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	39	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	45	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	25	0·9830	0·9838	0·9840	0·9846	0·9855	0·9841	0·0006
	38	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	26	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	30	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	28	0·9830	0·9838	0·9840	0·9846	0·9855	0·9841	0·0006
	27	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	47	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	41	0·9830	0·9837	0·9840	0·9846	0·9855	0·9841	0·0006
	34	0·9830	0·9837	0·9839	0·9846	0·9855	0·9841	0·0006
	48	0·9829	0·9837	0·9840	0·9846	0·9854	0·9841	0·0006
	36	0·9829	0·9837	0·9840	0·9846	0·9854	0·9841	0·0006
	42	0·9830	0·9837	0·9840	0·9846	0·9854	0·9841	0·0006
	46	0·9829	0·9837	0·9839	0·9846	0·9854	0·9841	0·0006
	40	0·9829	0·9837	0·9840	0·9846	0·9854	0·9841	0·0006

Table A3. *Evaluated on chromosome 28 (BTA28)*

BTA28	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	61	0·9698	0·9711	0·9721	0·9726	0·9747	0·9720	0·0011
	67	0·9698	0·9711	0·9721	0·9725	0·9747	0·9720	0·0011
	51	0·9697	0·9711	0·9721	0·9726	0·9747	0·9720	0·0011
	63	0·9697	0·9710	0·9721	0·9725	0·9746	0·9720	0·0011
	53	0·9698	0·9711	0·9721	0·9725	0·9747	0·9720	0·0011
	9	0·9697	0·9711	0·9721	0·9726	0·9746	0·9719	0·0011
	68	0·9698	0·9710	0·9721	0·9725	0·9747	0·9719	0·0011
	62	0·9697	0·9711	0·9720	0·9725	0·9747	0·9719	0·0011
	75	0·9698	0·9711	0·9721	0·9725	0·9746	0·9719	0·0011
	85	0·9697	0·9711	0·9721	0·9725	0·9746	0·9719	0·0011
	69	0·9698	0·9711	0·9721	0·9725	0·9746	0·9719	0·0011
	15	0·9697	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	87	0·9698	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	11	0·9697	0·9711	0·9721	0·9725	0·9747	0·9719	0·0011
	64	0·9697	0·9711	0·9721	0·9724	0·9746	0·9719	0·0011
	77	0·9698	0·9711	0·9721	0·9725	0·9745	0·9719	0·0011
	72	0·9697	0·9711	0·9721	0·9725	0·9746	0·9719	0·0011
	93	0·9698	0·9711	0·9721	0·9724	0·9746	0·9719	0·0011
	70	0·9697	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	71	0·9698	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	91	0·9697	0·9711	0·9721	0·9725	0·9746	0·9719	0·0011
	49	0·9697	0·9710	0·9721	0·9725	0·9747	0·9719	0·0011
	7	0·9697	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A3. (*Cont.*)

BTA28	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	86	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	13	0·9697	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	50	0·9697	0·9710	0·9720	0·9725	0·9747	0·9719	0·0011
	65	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	8	0·9697	0·9711	0·9721	0·9725	0·9747	0·9719	0·0011
	66	0·9697	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	99	0·9697	0·9711	0·9721	0·9724	0·9746	0·9719	0·0011
	101	0·9698	0·9711	0·9721	0·9725	0·9745	0·9719	0·0011
	54	0·9698	0·9711	0·9720	0·9725	0·9747	0·9719	0·0011
	94	0·9698	0·9711	0·9721	0·9725	0·9745	0·9719	0·0011
	21	0·9698	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	20	0·9698	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	73	0·9697	0·9711	0·9720	0·9725	0·9745	0·9719	0·0011
	52	0·9697	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	109	0·9698	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	111	0·9698	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	115	0·9697	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	19	0·9697	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	14	0·9697	0·9711	0·9720	0·9725	0·9745	0·9719	0·0011
	92	0·9698	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	88	0·9698	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	23	0·9698	0·9711	0·9721	0·9725	0·9745	0·9719	0·0011
	90	0·9697	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	17	0·9697	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	76	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	97	0·9697	0·9711	0·9721	0·9724	0·9746	0·9719	0·0011
	116	0·9697	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	57	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	110	0·9698	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	117	0·9697	0·9711	0·9720	0·9725	0·9746	0·9719	0·0011
	74	0·9698	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	96	0·9697	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	98	0·9697	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	81	0·9698	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	100	0·9698	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	55	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	78	0·9697	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	89	0·9697	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	118	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	59	0·9698	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	56	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	102	0·9697	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	12	0·9697	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	1	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	79	0·9697	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	114	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	10	0·9696	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	95	0·9698	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	105	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	103	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	112	0·9697	0·9711	0·9720	0·9724	0·9746	0·9719	0·0011
	113	0·9698	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	2	0·9697	0·9710	0·9720	0·9724	0·9747	0·9719	0·0011
	120	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	16	0·9696	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	3	0·9697	0·9710	0·9720	0·9725	0·9746	0·9719	0·0011
	80	0·9698	0·9710	0·9719	0·9724	0·9745	0·9719	0·0011
	119	0·9697	0·9710	0·9720	0·9725	0·9745	0·9719	0·0011
	18	0·9696	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	104	0·9697	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	83	0·9698	0·9710	0·9719	0·9724	0·9745	0·9719	0·0011
	107	0·9697	0·9711	0·9720	0·9724	0·9745	0·9719	0·0011
	4	0·9697	0·9710	0·9719	0·9724	0·9745	0·9719	0·0011

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A3. (*Cont.*)

BTA28	Ensemble system ID	Min	1st QU	Median	3rd QU	Max	Mean	SD
	22	0·9698	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	24	0·9698	0·9710	0·9720	0·9724	0·9745	0·9719	0·0011
	6	0·9698	0·9710	0·9719	0·9724	0·9745	0·9719	0·0011
	5	0·9697	0·9710	0·9720	0·9724	0·9746	0·9719	0·0011
	60	0·9697	0·9710	0·9719	0·9724	0·9746	0·9718	0·0011
	58	0·9697	0·9710	0·9720	0·9724	0·9745	0·9718	0·0011
	31	0·9697	0·9709	0·9720	0·9724	0·9745	0·9718	0·0011
	82	0·9697	0·9710	0·9719	0·9723	0·9745	0·9718	0·0011
	33	0·9697	0·9709	0·9720	0·9724	0·9745	0·9718	0·0011
	25	0·9697	0·9709	0·9720	0·9724	0·9745	0·9718	0·0011
	108	0·9698	0·9710	0·9719	0·9724	0·9745	0·9718	0·0011
	32	0·9697	0·9709	0·9720	0·9724	0·9746	0·9718	0·0011
	84	0·9698	0·9710	0·9720	0·9724	0·9745	0·9718	0·0011
	106	0·9697	0·9710	0·9720	0·9723	0·9745	0·9718	0·0011
	26	0·9697	0·9709	0·9720	0·9724	0·9746	0·9718	0·0011
	39	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	37	0·9697	0·9709	0·9719	0·9724	0·9745	0·9718	0·0011
	27	0·9697	0·9709	0·9719	0·9724	0·9745	0·9718	0·0011
	35	0·9697	0·9709	0·9719	0·9724	0·9745	0·9718	0·0011
	38	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	28	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	43	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	44	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	30	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	29	0·9697	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	41	0·9697	0·9709	0·9718	0·9723	0·9745	0·9718	0·0011
	45	0·9697	0·9709	0·9719	0·9723	0·9744	0·9718	0·0011
	47	0·9696	0·9709	0·9719	0·9723	0·9744	0·9718	0·0011
	36	0·9696	0·9709	0·9719	0·9724	0·9745	0·9718	0·0011
	34	0·9696	0·9709	0·9719	0·9723	0·9745	0·9718	0·0011
	40	0·9696	0·9708	0·9718	0·9723	0·9745	0·9717	0·0011
	46	0·9696	0·9709	0·9718	0·9723	0·9744	0·9717	0·0011
	42	0·9696	0·9709	0·9718	0·9722	0·9744	0·9717	0·0011
	48	0·9696	0·9709	0·9718	0·9723	0·9744	0·9717	0·0011

Table A4. *List of the top 120 ensemble systems and combinations*

ID	Combination	ID	Combination
1	Bgl-Imp-fPH-Alp-fhap-Fimp	61	Bgl-Alp-fhap-Imp-fPH-Fimp
2	Bgl-Imp-fPH-Alp-Fimp-fhap	62	Bgl-Alp-fhap-Imp-Fimp-fPH
3	Bgl-Imp-fPH-fhap-Alp-Fimp	63	Bgl-Alp-fhap-fPH-Imp-Fimp
4	Bgl-Imp-fPH-fhap-Fimp-Alp	64	Bgl-Alp-fhap-fPH-Fimp-Imp
5	Bgl-Imp-fPH-Fimp-Alp-fhap	65	Bgl-Alp-fhap-Fimp-Imp-fPH
6	Bgl-Imp-fPH-Fimp-fhap-Alp	66	Bgl-Alp-fhap-Fimp-fPH-Imp
7	Bgl-Imp-Alp-fPH-fhap-Fimp	67	Bgl-Alp-Fimp-Imp-fPH-fhap
8	Bgl-Imp-Alp-fPH-Fimp-fhap	68	Bgl-Alp-Fimp-Imp-fhap-fPH
9	Bgl-Imp-Alp-fhap-fPH-Fimp	69	Bgl-Alp-Fimp-fPH-Imp-fhap
10	Bgl-Imp-Alp-fhap-Fimp-fPH	70	Bgl-Alp-Fimp-fPH-fhap-Imp
11	Bgl-Imp-Alp-Fimp-fPH-fhap	71	Bgl-Alp-Fimp-fhap-Imp-fPH
12	Bgl-Imp-Alp-Fimp-fhap-fPH	72	Bgl-Alp-Fimp-fhap-fPH-Imp
13	Bgl-Imp-fhap-fPH-Alp-Fimp	73	Bgl-fhap-Imp-fPH-Alp-Fimp
14	Bgl-Imp-fhap-fPH-Fimp-Alp	74	Bgl-fhap-Imp-fPH-Fimp-Alp
15	Bgl-Imp-fhap-Alp-fPH-Fimp	75	Bgl-fhap-Imp-Alp-fPH-Fimp
16	Bgl-Imp-fhap-Alp-Fimp-fPH	76	Bgl-fhap-Imp-Alp-Fimp-fPH
17	Bgl-Imp-fhap-Fimp-fPH-Alp	77	Bgl-fhap-Imp-Fimp-fPH-Alp
18	Bgl-Imp-fhap-Fimp-Alp-fPH	78	Bgl-fhap-Imp-Fimp-Alp-fPH
19	Bgl-Imp-Fimp-fPH-Alp-fhap	79	Bgl-fhap-fPH-Imp-Alp-Fimp
20	Bgl-Imp-Fimp-fPH-fhap-Alp	80	Bgl-fhap-fPH-Imp-Fimp-Alp
21	Bgl-Imp-Fimp-Alp-fPH-fhap	81	Bgl-fhap-fPH-Alp-Imp-Fimp

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press

Table A4. (*Cont.*)

ID	Combination	ID	Combination
22	Bgl-Imp-Fimp-Alp-fhap-fPH	82	Bgl-fhap-fPH-Alp-Fimp-Imp
23	Bgl-Imp-Fimp-fhap-fPH-Alp	83	Bgl-fhap-fPH-Fimp-Imp-Alp
24	Bgl-Imp-Fimp-fhap-Alp-fPH	84	Bgl-fhap-fPH-Fimp-Alp-Imp
25	Bgl-fPH-Imp-Alp-fhap-Fimp	85	Bgl-fhap-Alp-Imp-fPH-Fimp
26	Bgl-fPH-Imp-Alp-Fimp-fhap	86	Bgl-fhap-Alp-Imp-Fimp-fPH
27	Bgl-fPH-Imp-fhap-Alp-Fimp	87	Bgl-fhap-Alp-fPH-Imp-Fimp
28	Bgl-fPH-Imp-fhap-Fimp-Alp	88	Bgl-fhap-Alp-fPH-Fimp-Imp
29	Bgl-fPH-Imp-Fimp-Alp-fhap	89	Bgl-fhap-Alp-Fimp-Imp-fPH
30	Bgl-fPH-Imp-Fimp-fhap-Alp	90	Bgl-fhap-Alp-Fimp-fPH-Imp
31	Bgl-fPH-Alp-Imp-fhap-Fimp	91	Bgl-fhap-Fimp-Imp-fPH-Alp
32	Bgl-fPH-Alp-Imp-Fimp-fhap	92	Bgl-fhap-Fimp-Imp-Alp-fPH
33	Bgl-fPH-Alp-fhap-Imp-Fimp	93	Bgl-fhap-Fimp-fPH-Imp-Alp
34	Bgl-fPH-Alp-fhap-Fimp-Imp	94	Bgl-fhap-Fimp-fPH-Alp-Imp
35	Bgl-fPH-Alp-Fimp-Imp-fhap	95	Bgl-fhap-Fimp-Alp-Imp-fPH
36	Bgl-fPH-Alp-Fimp-fhap-Imp	96	Bgl-fhap-Fimp-Alp-fPH-Imp
37	Bgl-fPH-fhap-Imp-Alp-Fimp	97	Bgl-Fimp-Imp-fPH-Alp-fhap
38	Bgl-fPH-fhap-Imp-Fimp-Alp	98	Bgl-Fimp-Imp-fPH-fhap-Alp
39	Bgl-fPH-fhap-Alp-Imp-Fimp	99	Bgl-Fimp-Imp-Alp-fPH-fhap
40	Bgl-fPH-fhap-Alp-Fimp-Imp	100	Bgl-Fimp-Imp-Alp-fhap-fPH
41	Bgl-fPH-fhap-Fimp-Imp-Alp	101	Bgl-Fimp-Imp-fhap-fPH-Alp
42	Bgl-fPH-fhap-Fimp-Alp-Imp	102	Bgl-Fimp-Imp-fhap-Alp-fPH
43	Bgl-fPH-Fimp-Imp-Alp-fhap	103	Bgl-Fimp-fPH-Imp-Alp-fhap
44	Bgl-fPH-Fimp-Imp-fhap-Alp	104	Bgl-Fimp-fPH-Imp-fhap-Alp
45	Bgl-fPH-Fimp-Alp-Imp-fhap	105	Bgl-Fimp-fPH-Alp-Imp-fhap
46	Bgl-fPH-Fimp-Alp-fhap-Imp	106	Bgl-Fimp-fPH-Alp-fhap-Imp
47	Bgl-fPH-Fimp-fhap-Imp-Alp	107	Bgl-Fimp-fPH-fhap-Imp-Alp
48	Bgl-fPH-Fimp-fhap-Alp-Imp	108	Bgl-Fimp-fPH-fhap-Alp-Imp
49	Bgl-Alp-Imp-fPH-fhap-Fimp	109	Bgl-Fimp-Alp-Imp-fPH-fhap
50	Bgl-Alp-Imp-fPH-Fimp-fhap	110	Bgl-Fimp-Alp-Imp-fhap-fPH
51	Bgl-Alp-Imp-fhap-fPH-Fimp	111	Bgl-Fimp-Alp-fPH-Imp-fhap
52	Bgl-Alp-Imp-fhap-Fimp-fPH	112	Bgl-Fimp-Alp-fPH-fhap-Imp
53	Bgl-Alp-Imp-Fimp-fPH-fhap	113	Bgl-Fimp-Alp-fhap-Imp-fPH
54	Bgl-Alp-Imp-Fimp-fhap-fPH	114	Bgl-Fimp-Alp-fhap-fPH-Imp
55	Bgl-Alp-fPH-Imp-fhap-Fimp	115	Bgl-Fimp-fhap-Imp-fPH-Alp
56	Bgl-Alp-fPH-Imp-Fimp-fhap	116	Bgl-Fimp-fhap-Imp-Alp-fPH
57	Bgl-Alp-fPH-fhap-Imp-Fimp	117	Bgl-Fimp-fhap-fPH-Imp-Alp
58	Bgl-Alp-fPH-fhap-Fimp-Imp	118	Bgl-Fimp-fhap-fPH-Alp-Imp
59	Bgl-Alp-fPH-Fimp-Imp-fhap	119	Bgl-Fimp-fhap-Alp-Imp-fPH
60	Bgl-Alp-fPH-Fimp-fhap-Imp	120	Bgl-Fimp-fhap-Alp-fPH-Imp

https://doi.org/10.1017/S001667231200033X Published online by Cambridge University Press