Livestock Science 191 (2016) 91–96

Contents lists available at ScienceDirect

Livestock Science

journal homepage: www.elsevier.com/locate/livsci

Short communication

Accessing marker effects and heritability estimates from genome prediction by Bayesian regularized neural networks

Leonardo Siqueira Glória [a,c], Cosme Damião Cruz [b], Ricardo Augusto Mendonça Vieira [c], Marcos Deon Vilela de Resende [d], Paulo Sávio Lopes [a], Otávio H.G.B. Dias de Siqueira [a], Fabyano Fonseca e Silva [a,*]

[a] Department of Animal Science, Universidade Federal de Viçosa, Av. P.H. Holfs, 36570-000 Viçosa, Brazil
[b] Department of General Biology, Universidade Federal de Viçosa, Av. P.H. Holfs, 36570-000 Viçosa, Brazil
[c] Laboratory of Animal Science, Universidade Estadual do Norte Fluminense, 28013-602 Campos dos Goytacazes, Brazil
[d] Embrapa Florestas/UFV, Estrada da Ribeira, km 111, 83411-000 Colombo, Brazil

ARTICLE INFO

Article history:
Received 29 March 2016
Received in revised form
14 July 2016
Accepted 23 July 2016

Keywords:
Genomic selection
Artificial neural networks
QTL
Genetic parameters

ABSTRACT

Recently, there is an increasing interest on semi- and non-parametric methods for genome-enabled prediction, among which the Bayesian regularized artificial neural networks (BRANN) stand. We aimed to evaluate the predictive performance of BRANN and to exploit SNP effects and heritability estimates using two different approaches (relative importance-RI, and relative contribution-RC). Additionally, we aimed also to compare BRANN with the traditional RR-BLUP and BLASSO by using simulated datasets. The simplest BRANN (net1), RR-BLUP and BLASSO methods outperformed other more parameterized BRANN (net2, net3, … net6) in terms of predictive ability. For both simulated traits (Y1 and Y2) the net1 provided the best h^2 estimates (0.33 for both, being the true $h^2=0.35$), whereas RR-BLUP (0.18 and 0.22 for Y1 and Y2, respectively) and BLASSO (0.20 and 0.26 for Y1 and Y2, respectively) underestimated h^2. The marker effects estimated from net1 (using RI and RC approaches) and RR-BLUP were similar, but the shrinkage strength was remarkable for BLASSO on both traits. For Y1, the correlation between the true fifty QTL effects and the effects estimated for the SNPs located in the same QTL positions were 0.61, 0.60, 0.60 and 0.55, for RI, RC, RR-BLUP and BLASSO; and for Y2, these correlations were 0.81, 0.81, 0.81 and 0.71, respectively. In summary, we believe that estimates of SNP effects are promising quantitative tools to bring discussions on chromosome regions contributing most effectively to the phenotype expression when using ANN for genomic predictions.

1. Introduction

Over the last years, several parametric methods such as RR-BLUP, Bayes A, B, Cπ and BLASSO have been proposed to genome-enabled prediction. However, recently there is an increasing application of semi- and non-parametric methods, among which the artificial neural networks (ANN) has aroused some research interest. In summary, one of the theoretical advantages of ANN is the flexibility, since it is not necessary to make a priori assumptions about the relationships between inputs, e.g., SNP genotypes, and output as phenotypic observations. Furthermore, ANN has great flexibility to handle different types of complex non-additive effects such as epistasis (Felipe et al., 2014; Howard et al., 2014), which currently is a hot topic in the field of Statistical Genomics (Beam et al., 2014).

Despite this advantage, it is expected that ANN complexity and the number of markers are directly associated. It may lead to overfitting and consequently to poor predictive performance. For an empirical control of model complexity, the weights of ANN can be regularized. Under a Bayesian approach, the regularization is obtained by treating the weights as random variables following specific prior distributions (Okut et al., 2011). This special class of ANN is called Bayesian regularized artificial neural networks (BRANN).

Although BRANN have been shown to be effective for genome-enabled prediction in animal breeding (Gianola et al., 2011, Perez-Rodriguez et al., 2013), biological interpretation from the marker effects (related to QTL detection) and genetic parameter (heritability) estimates have been underexploited. Thus, we aimed to apply BRANN to genome-enabled prediction and to exploit SNP effects and heritability estimates using two different approaches. Additionally, we aimed also to compare BRANN with the

[*] Corresponding author.
E-mail address: fabyanofonseca@ufv.br (F. Fonseca e Silva).

http://dx.doi.org/10.1016/j.livsci.2016.07.015
1871-1413/© 2016 Elsevier B.V. All rights reserved.

traditional RR-BLUP and BLASSO by using simulated datasets.

2. Materials and methods

2.1. Simulated data set

The data was simulated according to Usai et al. (2014). The base generation (GEN_0) was composed by 1020 (20 males and 1000 females) unrelated individuals. Each one of the next four non-overlaping generations (GEN1, GEN2, GEN3 and GEN4) consisted of 20 males and 1000 females from GEN_0 by randomly mating each male with 50 females. It was considered that each female originated one female, except for 20 of them, which produced two individuals (one male and one female). The pedigree was composed by 4100 individuals (males only from GEN_0 and all the individuals from GEN_1 to GEN_4).

The simulated genome comprised five chromosomes, each one with a size of 99.95 Mb carrying 2000 equally distributed SNPs (1 SNP per 0.05 Mb). The GEN_0 haplotypes were generated in order to obtain fixed LD decay ($r^2 \approx 0.07$ on 1000 kb) and MAF (≈ 0.28) distribution. A total of 50 SNPs were randomly sampled and treated as QTLs, whose effects were generated from a gamma distribution with scale and shape parameters equal to 5.4 and 0.42, respectively. The sum of the QTLs' additive effects over each individual was used to define the true breeding values (TBV). Two traits (Y1 and Y2), whose the difference between them was the QTL positions on the genome, were generated assuming heritability equal to 0.35.

The phenotype of each individual was given by the true breeding value plus the random residual effect sampled from a normal distribution with mean zero and variance that ensures the heritability equal to 0.35. Data of 3000 females from GEN_1 to GEN_3 were used in the training population, so that the rest of the population (1020 individuals from GEN_4) were considered for validation.

The phenotype, TBV, and genotype files (including the map with known SNP location and tue QTLs on each chromosome) are available in the following electronic address: http://qtl-mas-2012.kassiopeagroup.com/en/dataset.php.

2.2. Feed-forward ANN

In summary, training algorithms of ANN were used to describe the interrelationships among the input data based on different levels of learning organized in the so called layers. Each layer is divided into units called neurons, which regulate specific weights for the predictor variables by means of linear and non-linear activation functions. In the field of genome enabled prediction, the general idea behind ANN use is to describe complex relationships between phenotypes (response variable) and genotypes (predictor variables). Once understood this relationship pattern, the ANN is trained and apt to make predictions of phenotypes based on new genotypic values.

The feed-forward ANN (FFANN) assumes that the output of any layer does not affect that same layer, only the next one (there is no feedback). The simplest FFANN consists of three layers. The input layer (IL) is given by genotypes of P markers for N individuals. The IL is connected to a hidden layer (HL) with T neurons), which in turn is connected to an output layer (OL) with only one neuron. These connections are directed by means of estimated weights that measure the influence of the predictor variables on the response variable. Additionally to the weights, a bias, also called intercept, is also estimated.

The mentioned FFANN can be described using the input genotype matrix **X** with N rows (i = 1, 2,…, N) and P columns (j = 1, 2, …, P). Each element of **X** is given by x_{ij}, with values −1 (aa), 0 (aA) or 1 (AA). The HL is composed by T neurons (t = 1, 2,…, T), each one generating a vector of weights, $w_{1j}^{[t]}$, plus a vector of bias, b_t. This resultant linear combination (1) is then transformed using an activation function $f(\cdot)$, thus generating the output $a_i^{[t]}$ of the neuron t for the individual i:

$$a_i^{[t]}=f\left(\sum_{j=1}^{P} w_{1j}^{[t]}x_{ji} + b_t\right), \tag{1}$$

The activation function can be either linear or non-linear and has to be monotonically increasing. A non-linear activation function in the HL gives the neural network a greater flexibility than standard linear regression models (Bishop, 2006). In the OL, the T derived output scores for individual i from HL are now considered as input in a new linear combination (2), that is transformed using an activation function $g(\cdot)$. Thus, the final output (y_i) for individual i depends on new estimated weight vectors w_{2t} and one scalar bias (b):

$$y_i=g\left(\sum_{t=1}^{T} w_{2t}a_i^{[t]}+b\right) = g\left(\sum_{t=1}^{T} w_{2t}f\left(\sum_{j=1}^{P} w_{1j}^{[t]}x_{ji} + b_t\right)+b\right) \tag{2}$$

In sequence, the results are back-propagated in order to update the weights and biases, which can be seen as the unknown parameters to be estimated under a statistical viewpoint.

2.3. Bayesian regularized artificial neural network (BRANN)

Based on our experience, the ANN complexity increases as the number of markers increases, and overfitting and poor predictive performance are corollaries. For an empirical control of this complexity, the weights of the FFANN can be regularized. Under a Bayesian approach, the regularization is obtained by treating the weights as random variables following specific prior distributions. Okut et al. (2011) and Gianola et al. (2011) proposed the Bayesian regularized artificial neural network (BRANN) for predicting complex phenotypes from genomic data. In summary, the posterior distribution for the weights from a BRANN can be accessed based on Bayes theorem terms:

$$P(w|Y,\alpha,\gamma,M)=\frac{P(Y|w,\gamma,M)P(w|\alpha,M)}{P(Y|\alpha,\gamma,M)}, \tag{3}$$

in which $P(w|Y,\alpha,\gamma,M)$ is the posterior distribution; $P(Y|w,\gamma,M)$ is the likelihood function; $P(w|\alpha,M)$ is the prior distribution for the weights vector; and $P(Y|\alpha,\gamma,M)$ is the normalization factor, also called marginal likelihood. In this context, Y represents the observed data (genotype matrix and observed phenotype values); **w** is the unknown weights vector; M represents the architecture of the neural network (such as a given model in the traditional regression framework); and α and γ are the regularization parameters that control the trade-off between goodness of fit and smoothing. The number of weights to be estimated in a BRANN is considerably smaller than the number of weights in a standard FFANN, because the regularization shrinks unnecessary weights towards zero, effectively eliminating them.

Although BRNN has been shown to be effective for genomic enabled prediction, unlike the traditional parametric models (such Bayesian regressions), BRNN do not provide direct estimates of marker effects and variance components. In this context, we applied two different methods for the identification of input variable relevance in ANN to access SNP effect estimates under the BRANN viewpoint.

The *trainbr* function of Neural Network Toolbox™ of MATLAB® (Beale, et al., 2010) was used for the BRANN implementation. Six