Written Review — Deep Learning for Population Genetic Inference Sara Sheehan, Yun S. Song - PLOS Computational Biology

Adithya Ganesh, Armin Pourshafeie, Behrooz Ghorbani, Karen Yang, Philip Hwang, Wendy Liu

October 31, 2016

1 Overview

Statistical inference for population genetics can be challenging, since determining the likelihood of complex population genetic models is often computationally intractable. To address this, Sheehan and Song present a likelihood-free inference algorithm based on deep learning. In particular, their analysis focuses on 197 *Drosophila melanogaster* genomes from Zambia, with the goal of inferring their demographic history, as well as regions of the genome under selection.

The researchers transform each genomic region into a set of 345 summary statistics, with the goal of estimating demographic history data and selected regions of the genome. The researchers use the `msms` program to simulate different demographic and selection parameters, and use this data to evaluate the predictive accuracy of the deep learning model.

1.1 Related Works

Classical machine learning methods like SVMs and boosting have been applied to classifying the genome into neutral versus selected regions. There are also a broad range of methods that have been developed to infer ancestral population size changes, including PSCM, diCal, and MSMC. However, few research works have addressed the problem of jointly inferring population size changes and selection which Sheehan and Song tackle.

Approximate Bayesian Computation (ABC) is a likelihood-free inference method that has been used to study various models in population genetics. ABC has achieved popularity in the research community as it is easy to use and outputs a posterior distribution. However, ABC does not perform as well as the number of summary statistics grows, and is difficult to use for joint inference of continuous parameters and categorical distributions.

Deep learning has several advantages, being able to effectively model a joint inference problem and providing insight into which summary statistics had the greatest influence on the output parameters. However, unlike the ABC method, it does not provide a posterior distribution.

1.2 ABC Method Overview

Approximate Bayesian Computation (ABC) is a likelihood-free inference method based on simulating datasets and comparing their summary statistics. We have provided an overview of the algorithm, adapted from [2].

Rejection-based approximate Bayesian inference. Suppose we are trying to estimate a parameter of interest, ϕ. Then we perform the following rejection-sampling algorithm:

1. Choose a summary statistic S and calculate its value s for the observed data set
2. Choose a tolerance δ
3. Simulate ϕ' from the prior distribution for ϕ
4. Simulate a genealogical tree under the chosen model, such as a coalescent model

5. Simulate ancestral allelic types at the root of the tree, and then mutation events along the tree to generate a data set at the leaves
6. Compute s', the value of S for the simulated data set
7. If $||s' - s|| \leq \delta$, then accept ϕ', otherwise reject
8. Repeat steps 3 to 7 until k acceptances have been obtained.

The researchers use the implementation in ABCtoolbox [3] to benchmark the performance of their deep learning model.

2 Methods

2.1 Data

The authors train their deep learning model on simulated data. This data is generated from the following population dynamics model. First, it is assumed that the population effective size is a piece-wise constant function of time. In addition, this function can only have two change points. The change points are also fixed to be $t_1 = 0.5$ and $t_2 = 5$ in coalescence units. In the ancient period, the effective population size is called N_3, in the bottle-neck period the effective population size is called N_2, and in the recent period, it is called N_1. The authors generate $N_i = 10^5 \times \lambda_i$ where $\lambda_1 \sim Unif(3, 14)$, $\lambda_2 \sim Unif(0.5, 6)$, and $\lambda_3 \sim Unif(2, 10)$.

Each genome is divided into 160 regions of size $100kb$. Each base has a $\mu := 8.4 \times 10^{-9}$ probability of mutation per each generation. If a mutation occurs, the paper assumes that it occurs randomly in the middle $20kb$ of the region. Each region has also a recombination rate of r, where $r = \mu$. The acquired mutations can be categorized into {No Selection, Balancing, Hard Sweep, Soft Sweep} categories. For their analysis, the authors generate 2500 different demographic histories, with 160 regions for each one, for a total of $400,000$ data sets. 75% of these data sets are used for training and the rest are used as a test set.

2.2 Algorithm

In this section, we provide an overview of the deep learning algorithm the researchers used. Because it is hard to directly use raw genomic data, they transformed the input data into summary statistics first, similar to the procedure followed when applying the ABC method. They included a large number of potentially informative summary statistics in order to prevent negative influence from correlated or uninformative statistics. Each 100kb region is divided into three smaller regions: 1) close to the selected site (40–60 kb), 2) mid-range from the selected site (20–40 kb and 60–80 kb), and 3) far from the selected site (0–20 kb and 80–100 kb). This choice is based off of a previous simulation made by Peter *et al* [4]. Within each of these three regions, the following statistics are calculated:

1. Number of segregating sites within each smaller region
2. D statistic, computed as $D = \frac{\pi - S/a_1}{sqrt\hat{V}ar(\pi - S/a_1)}$, where π is the average number of pairwise differences between two samples, and $a_1 = \sum_{i=1}^{n-1} 1/i$
3. Folded site frequency spectrum
4. Length distribution between segregating sites
5. Identity-by-state tract length distributions
6. Linkage disequilibrium
7. H_1, H_{12} and H_2, which are statistics that help distinguish hard and soft sweeps

To achieve better initialization, the model is pre-trained using autoencoders. A KL divergence term is added in the cost function to ensure sparsity. Regularization is also included during the unsupervised pre-training, yielding the following autoencoder cost function:

$$A_\lambda(W,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}||\hat{x}^{(i)} - x^{(i)}||^2 + \beta\sum_{j=1}^{u_2} KL(\rho|\hat{\rho_j}) + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{j=1}^{u_l}\sum_{k=1}^{u_{l+1}}[w_{jk}^{(l)}]^2$$

The neural network itself regularizes the weights on the last layer during fine-tuning. The overall cost function is:

$$J_\lambda(W,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}||h_{W,b}(x^{(i)}) - y^{(i)}||^2 + \frac{\lambda}{2}\sum_{j=1}^{u_L-1}\sum_{k=1}^{u_L}[w_{jk}^{(L-1)}]^2$$

Softmax regression is used to make multi-class estimation. The classification cost function is:

$$J_\lambda^{\text{softmax}}(W,b) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{K}1\{y^{(i)}=j\}\log p(y^{(i)}=j|x^{(i)};W,b) + \frac{\lambda}{2}\sum_{s=1}^{u_L-1}\sum_{t=1}^{u_L}[w_{st}^{(L-1)}]^2$$

In order to determine the most informative subset of the statistics, they used both a permutation testing approach and a perturbation method, which will be introduced in the following section.

3 Results and Conclusions

3.1 Parameter estimation for *Drosophila melanogaster*

The NN trained with the simulated data is then used to estimate the demographic parameters and selection sites on a dataset with 197 *Drosophila melanogaster* genomes. Chromosome arms 2L, 2R, 3L, and 3R in the study are partitioned into 20kb windows and the method was ran on a sliding window of size 100kb. For each run, the middle 20kb was classified. The authors discovered that 14% of the runs (often near telomeres and centromeres) yielded values outside of the simulated region. These regions were excluded from the downstream analysis.

The estimated demographic history closely matched those found by PSMC. However, the authors note that the times for the demographic events were discovered by PSMC, and they acknowledge that further work may be necessary to evaluate the performance of their method. For both real data and simulated data, the estimates for the most recent population size has the highest uncertainty. This is not surprising given that there are fewer coalescence events that can be used for this period. Their demographic findings, along with a comparison to PSMC and the method presented by Duchen et al., can be seen in Figure 3 of original work.

The NN identified 42 hard sweeps, 34 soft sweeps and 17 regions under balancing selection with high confidence. The authors observe that soft sweeps and balancing selection seem to occur more frequently near the centromere of each chromosome.

This work explored two methods for discovering informative statistics in their data set. First, they apply a permutation method, where the value of one statistics is randomly permuted and the change in accuracy signifies the importance of the statistic. They also employ a perturbation method, where the change in performance after a small increase or decrease in a statistic is measured. Figure 5 in the paper presents a Venn diagram for the 25 most important statistics as identified by the permutation method. In this figure we can see that

1. Informative statistics for N_1 and N_2 have a higher overlap with the relevant statistics for selection than they do with the informative statistics for N_3. The authors note that this is reasonable since selection is a comparatively recent event.

2. IBS statistics play an important role as the population becomes more ancient.

3. Tajima's D statistic appears as an important variable for selection (as it was designed to do).

4. LD statistics are more important for selection than for demographic estimation.

3.2 Comparison to ABCtoolbox

The authors benchmarked the performance of their deep learning model against the ABC implementation in ABCtoolbox. For this section, the authors focused on demography estimation. Two experiments were performed 1) using all the 345 statistics, and 2) using 100 carefully chosen statistics. Table 8 compares the error rates. Deep learning outperforms ABC on N_1 estimation but the results for N_2, N_3 are inconclusive. This may be due to the observation, noted earlier, that selection (which is not modeled by ABC) is a comparatively recent phenomenon.

4 Future Work

While this work is notable as an early application of deep learning in the domain of population genetics, it has many limitations. Firstly, the network they use still relies on PSMC to obtain time change points. Important future work could focus on developing a more robust model which is independent of PSMC predictions. While the three event demographic description is a common model, as a future direction, it may be interesting to explore less restrictive models.

The authors argue that their deep learning model is appealing for its computational efficiency. This is in part because training can be parallelized across simulated datasets (since each dataset adds to the cost function independently). Training can be performed in a few days using the researchers' simulated dataset. By contrast, when applying the ABC method, each of the "training" datasets must be analyzed for *each* test dataset, which can take weeks. However, even while applying deep learning, the time required to simulate the data is computationally expensive (requiring a total of 370 hrs when 10 - 15 cores are used).

In Table 6, the authors present a confusion matrix comparing random initialization to autoencoder-based initialization. The results are surprising, as random initialization resulted in a large majority of the regions being classified as neutral, indicating that the network has not effectively learned from the data. It is possible that applying an initialization strategy such as the well-known approach described by Glorot and Bengio [5] would improve convergence.

The authors mention that classical methods like SVMs have been applied to classifying the genome into neutral versus selected regions. However, it is difficult to comparatively assess the performance of the proposed deep learning model, as no quantitative evaluation of classical methods is presented.

5 References

[1] Sheehan, Sara, and Yun S. Song. "Deep learning for population genetic inference." PLoS Comput Biol 12.3 (2016): e1004845.

[2] Beaumont, Mark A., Wenyang Zhang, and David J. Balding. "Approximate Bayesian computation in population genetics." Genetics 162.4 (2002): 2025-2035.

[3] Wegmann D, Leuenberger C, Neuenschwander S, Excoffier L. ABCtoolbox: a versatile toolkit for approximate Bayesian computations. BMC Bioinformatics. 2010; 11:116. doi: 10.1186/1471-2105-11- 116 PMID: 20202215

[4] Peter BM, Huerta-Sanchez E, Nielsen R. Distinguishing between selective sweeps from standing variation and from a de novo mutation. PLoS Genetics. 2012; 8(10):e1003011. doi: 10.1371/journal.pgen. 1003011 PMID: 23071458

[5] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Aistats. Vol. 9. 2010.