Neural Networks for self-adjusting Mutation Rate Estimation when the Recombination Rate is unknown

Klara Elisabeth Burger[1], **Peter Pfaffelhuber**[2], **and Franz Baumdicker**[1,3,✉]

[1]Cluster of Excellence "Machine Learning: New Perspectives for Science", University of Tübingen, Germany
[2]Department of Mathematical Stochastics, University of Freiburg, Germany
[3]Cluster of Excellence "Controlling Microbes to Fight Infections", Mathematical and Computational Population Genetics, University of Tübingen, Germany

Estimating the mutation rate, or equivalently effective population size, is a common task in population genetics. If recombination is low or high, the optimal linear estimation methods, namely Fu's and Watterson's estimator, are known and well understood. For intermediate recombination rates, the calculation of optimal estimators is more involved. As an alternative to model-based estimation, neural networks and other machine learning tools could help to develop good estimators in these involved scenarios. However, if no benchmark is available it is difficult to assess how well suited these tools are for different applications in population genetics.

Here we investigate feedforward neural networks for the estimation of the mutation rate and compare their performance with the frequently used optimal estimators introduced by Fu and Watterson. We find that neural networks can reproduce the optimal estimators if provided with the appropriate features and training sets. Remarkably, only one hidden layer is necessary to obtain a single estimator that performs almost as well as the optimal estimators for both, low and high recombination rates and provides a superior estimation method for intermediate recombination rates at the same time.

mutation rate estimation | neural nets | Watterson's estimator | population size estimate | population genetics | machine learning

Correspondence: ***franz.baumdicker@uni-tuebingen.de***

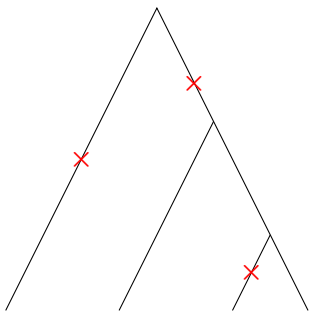
Introduction

Although the development of machine learning methods for population genetics faces multiple specific challenges, it is meanwhile clear that they are a promising technique to build more powerful inference tools, especially for problems that are hard to tackle down with classical methods. In contrast, in population genetics, many theoretical results have been obtained within the last decades and enabled us to identify the best inference technique for specific scenarios. For example, the variance of estimators of the mutation rate, or equivalently the effective population size, is well understood, at least if the rate is constant and recombination is low or high.

Estimating the effective population size or mutation rate. Estimating the scaled mutation rate, usually denoted by $\theta = 4N_e\mu$, is a fundamental task in population genetics. If the per generation mutation rate μ is known, estimating the scaled mutation rate corresponds to estimating the effective population size N_e, which is often of primary interest and correlates with the genetic diversity in a given population.

Many estimators of θ are developed within the coalescent framework without recombination, where mutations arise along a genealogy given by Kingman's coalescent (1). However, if recombination is included into the framework (2), estimators of θ often require an estimate of the recombination rate (3, 4). One of the most common estimators is Watterson's estimator (5). It is an easy to compute, unbiased and asymptotically consistent estimator, which has a low variance for high recombination rates, when compared to alternative model-based, unbiased estimators. For Watterson's estimator the only necessary input is the total number of segregating sites in the sample. However, for low recombination rates Watterson's estimator, while still unbiased and consistent, usually has a high variance, when compared to alternative model-based, unbiased estimators. Multiple such estimators have been developed based on the site frequency spectrum (SFS), which is the number of segregating sites that occur in k out of n individuals of the sample for $k = 1, \dots, n-1$. In particular, without recombination, the optimal unbiased linear estimator based on the SFS has been developed by Fu (6) and the linear estimator that minimizes the mean squared error has been identified by Futschik et. al. (7). Knowledge of the optimal estimators without and with high recombination rates enables us here to assess the overall performance of artificial neural net estimators which have been trained either for exactly these scenarios or with an unknown recombination rate.

Machine Learning in Population Genetics. Artificial neural networks are popular in various scientific fields and often used in cases of limited theoretical insights (8–10). Interestingly, current machine learning approaches in population genetics, including support vector machines, neural nets, and random forests, often use summary statistics of the genetic data borrowed from the theoretical literature as input data (11, 12), such as the site frequencies. An exception is a recent publication by Flagel et al. (13) who showed that effective population genetic inference can also be reached with deep learning structures like convolutional neural networks without precomputed summary statistics. Another approach is suggested in (14) introducing an exchangeable neural network for population genetic data that makes the ordering of the samples invisible to the neural network. Recently, Sanchez et al. (15) showed that combining deep learning structures and approximate Bayesian computation provides an effective method to reconstruct the



$S = (S_1, S_2, S_3) = (2, 0, 1)$

Fig. 1. Mutations along a coalescent tree for $n = 4$ and the corresponding site frequency spectrum S.

effective population size through time for unknown recombination rates. Thus, artificial neural nets are a promising tool in more involved scenarios with limited theoretical insights although the often complex architectures impair the comprehension of the underlying learning process.

Here, we take a different perspective and consider the estimation of the mutation rate from single nucleotide frequencies in the coalescent framework, a scenario where the optimal solution (which is a linear map of the SFS) is known. In particular, we consider a dense feedforward neural net with at most one hidden layer, which already suffices to achieve almost the performance of the optimal linear estimators and at the same time to provide a superior estimator for variable recombination rates.

Materials and Methods

Model-based Estimators of Mutation Rate and Population Size. In this section, we recall the properties of known estimators for θ, that are linear in the site frequency spectrum. More precisely, we consider mutations as modeled by a neutral Wright Fisher model with infinitely many sites model along Kingman's coalescent.

To obtain Kingman's coalescent for a sample of size n we trace back the series of ancestors through time. Therefore, we define an ancestral tree for this sample by repeatedly coalescing each pair of two lineages at a rate 1, such that, when k is the number of remaining lineages, at rate $\binom{k}{2}$ two randomly chosen lineages coalesce. The resulting random tree is called Kingman's coalescent. We denote the random duration the coalescent spends with exactly k lineages by $T_k \sim \text{Exp}\left(\binom{k}{2}\right)$. Neutral mutations are independently added upon this tree. More precisely, for a given genealogical tree, mutations can happen everywhere along the branches at rate $\frac{\theta}{2}$, see Fig 1. Given the length ℓ of a branch, the number of mutations on this branch is $\text{Poi}\left(\frac{\theta}{2}\ell\right)$ distributed. Consequently there are $M \sim \text{Poi}(\frac{\theta L}{2})$ mutations along the tree , if $L = \sum_{k=2}^{n} kT_k$ is the total length of the tree. We consider the infinitely many sites model, where each mutation hits a unique site. Thus the frequency of the derived allele in the sample population is given by the number of descendants of the branch where the mutation occurred. We define the site frequency spectrum as $S = (S_1, \ldots, S_{n-1})$, where S_i is the number of mutations that occur on a branch that is ancestral to exactly i out of n individuals in the sample.

In the following, we are looking for estimators of the form

$$\hat{\theta} = \sum_{i=1}^{n-1} a_i S_i.$$

which are uniquely defined by the vector $a = (a_1, \ldots, a_{n-1})$. We will see that optimal choices for a frequently depend on θ, which will lead to an iterative estimation procedure.

Unbiased linear Estimators of the Mutation Rate θ. Watterson's estimator is given by setting $a_1 = \cdots = a_{n-1} = \mathbb{E}[L]^{-1}$, i.e.

$$\widehat{\theta}_{\text{W}} := \frac{M}{h_n} = \sum_{i=1}^{n-1} \frac{S_i}{h_n} \quad \text{with} \quad h_n := \sum_{i=1}^{n-1} \frac{1}{i}$$

being the n-th harmonic number and $S = (S_1, ..., S_{n-1})$ the site frequency spectrum.

From a theoretical point of view, only the cases of no recombination or in the limit of high recombination (leading to independence between loci) can easily be treated. Watterson's estimator is unbiased for θ for all recombination rates and if there is no recombination the variance, as found by Watterson (5), is given by

$$\mathbb{V}_\theta\left[\widehat{\theta}_{\text{W}}\right] = \theta \frac{1}{h_n} + \theta^2 \frac{g_n}{h_n^2} \quad \text{where} \quad g_n := \sum_{i=1}^{n-1} \frac{1}{i^2}.$$

In the limit of high recombination we get that $S_k \sim \text{Poi}(\frac{\theta}{k})$ and $(S_1, \ldots, S_{n-1})$ are independent, such that the variance reduces to

$$\mathbb{V}_\theta\left[\sum_{i=1}^{n-1} \frac{S_i}{h_n}\right] = \frac{\theta}{h_n}.$$

In this case, using standard theory on exponential families, $\sum_{i=1}^{n-1} S_i$ is a complete and sufficient statistic for θ, and it follows from the Lehmann–Scheffé theorem that Watterson's estimator is a unique Uniformly Minimum Variance Unbiased Estimator (Uniformly MVUE) (16). However, this only holds for unlinked loci.

If no recombination is included the MVUE estimator was found by Fu (6). In this case, the best linear unbiased estimator of θ from the site frequency spectrum $S = (S_1, \ldots S_{n-1})$ is given in matrix notation by

$$f_\theta(S) = \frac{\alpha^\top (D_\alpha + \theta\Sigma)^{-1}}{\alpha^\top (D_\alpha + \theta\Sigma)^{-1} \alpha} S,$$

whereby

$$\alpha = (\alpha_1, \ldots, \alpha_{n-1}) = \left(1, \frac{1}{2}, \ldots, \frac{1}{n-1}\right),$$
$$D_\alpha = diag(\alpha_1, \ldots, \alpha_{n-1}), \quad \text{and}$$
$$\Sigma = \{\sigma_{ij}\}, \qquad i, j = 1, \ldots, n-1,$$

with

$$\sigma_{ii} := \begin{cases} \beta_n(i+1) & \text{if } i < \frac{n}{2}, \\ 2\frac{h_n - h_i}{n-i} - \frac{1}{i^2} & \text{if } i = \frac{n}{2}, \\ \beta_n(i) - \frac{1}{i^2} & \text{if } i > \frac{n}{2}, \end{cases}$$

and for $i > j$

$$\sigma_{ij} := \begin{cases} \frac{\beta_n(i+1) - \beta_n(i)}{2} & \text{if } i+j < n, \\ \frac{h_n - h_i}{n-i} + \frac{h_n - h_j}{n-j} - \frac{\beta_n(i) + \beta_n(j+1)}{2} - \frac{1}{ij} & \text{if } i+j = n, \\ \frac{\beta_n(j) - \beta_n(j+1)}{2} - \frac{1}{ij} & \text{if } i+j > n, \end{cases}$$

where

$$\beta_n(i) := \frac{2n(h_{n+1} - h_i)}{(n-i+1)(n-i)} - \frac{2}{n-i}.$$

We note that this MVUE is not uniform, since the optimal coefficients $a(\theta) = (a_1(\theta), ..., a_{n-1}(\theta))$ depend on the real θ.

General linear Estimators of the Mutation Rate θ. So far we only considered unbiased estimators and minimised their variance. Allowing for a potential bias of the estimator can decrease the mean squared error when compared to the unbiased estimators. The SFS based estimator with minimal mean square error in the absence of recombination was found by Futschik and Gach (7):
The linear estimator of θ from the site frequency spectrum $S = (S_1, \dots S_{n-1})$ with minimal mean square error is given by

$$\tilde{f}_\theta(S) = \alpha^\top \left(\frac{D_\alpha}{\theta} + \Sigma + \alpha\alpha^\top \right)^{-1} S,$$

where α, D_α, and Σ are as above.

Iterative estimation of θ. The estimators of Fu and Futschik depend on the true but unknown θ. In practice, we thus have to build an iterative estimator, which will yield a θ-independent estimator and approximate the variance minimizing or mean-square-error-minimizing estimators. Starting with some $\widehat{\theta}_0$, e.g. $\widehat{\theta}_0 = \widehat{\theta}_W$, Wattersons estimator, we set

$$\widehat{\theta}_{k+1}(S) := f_{\widehat{\theta}_k}(S)$$

which usually quickly converges. For example, for $n = 40$ and $\theta = 40$ it takes about 5 iterations until the estimate of θ is obtained up to 3 decimal places.
We call the resulting estimator $\widehat{\theta}_{ItV}$ for Fu's estimator f_θ, and $\widehat{\theta}_{ItMSE}$ for Futschik's estimator $\tilde{f}_\theta$. Note that due to the iteration the estimators are no longer explicitly linear or unbiased, but do not depend on θ.

Estimating the mutation rate with a dense feedforward network. We trained dense feedforward neural networks with zero or one hidden layer to perform the estimation task. Note that the architecture of the neural networks determine the type of functions the neural network can approximate. For example, if no hidden layer and no bias is included, the architecture ensures that the resulting estimator is a linear function in S.

Simulation of training data. In contrast to model-based estimators, the estimators first have to be trained in order to optimise parameters within the neural network. Therefore, we rely on simulations to train the neural network and then evaluate the resulting estimators. Five training data sets were generated with the software msprime by Kelleher et al. (17) for various parameters. For each training data set with $2 \cdot 10^5$ independent site frequency spectra, the sample size is given by $n = 40$ and the recombination rate was either set to 0 (no recombination), 20, 35 (moderate recombination), or 1000 (high recombination). Within all training data sets the mutation rate θ is chosen uniformly in $(0, 100)$. In addition, we combined the data sets with no recombination and high recombination with a data set with a uniformly chosen recombination rate in $(0, 50)$. This creates the fourth training data set of total size $6 \cdot 10^5$ with a variable recombination rate.

Feedforward neural networks. In this project we consider two artificial neural networks: a linear dense feedforward neural network (no hidden layer) and a dense feedforward neural network with one hidden layer and an adaptive loss function to ensure a robust performance. All neural networks take S as input and are trained for θ chosen uniformly in $(0, 100)$. ReLU was used as activation function and as optimzer the Adam algorithm (18) was chosen. The neural networks have been implemented in Python 3 via the library tensorflow and keras. The code is made available on Github: fbaumdicker/ML_in_pop_gen.

Linear neural network. As a simple check whether the network is able to learn the mutation rate at all we considered a neural net with no hidden layer and no bias node included, i.e. a linear neural network. This simple network structure ensures that the estimator is linear in S, but not necessarily unbiased.

Neural network with one hidden layer. To investigate how more complexity in the architecture of the neural net improves the performance, we also implemented a neural net with one hidden layer and one bias node. See Fig 2 for visualization. Naturally, the question arises on how many hidden nodes to include in this additional layer. We observed that the optimal number of nodes depends on the sample size n. The larger the sample size the more hidden nodes are beneficial. It is advisable to use at least $2n$ nodes. For $n = 40$, using 200 nodes in the hidden layer has proven to be a good choice.

Adaptive reweighing of the loss function based on optimal estimators. All neural nets have been trained on simulated data-sets where θ is uniformly chosen in $(0, 100)$. As for low recombination rates the coefficients of the optimal estimator depend on θ, we included an adaptive reweighing of the training data which ensures that the inferred estimator is not worse than the optimal estimator for all possible values of θ. A visualization of the training procedure is shown in Fig 3. The training of the "adaptive" neural net is done in several iterations. Each iteration consists of training a neural network as before and subsequently increasing the weight of those parts of the training data where the normalized mean square error (nMSE), i.e. the MSE divided by θ, is not close enough

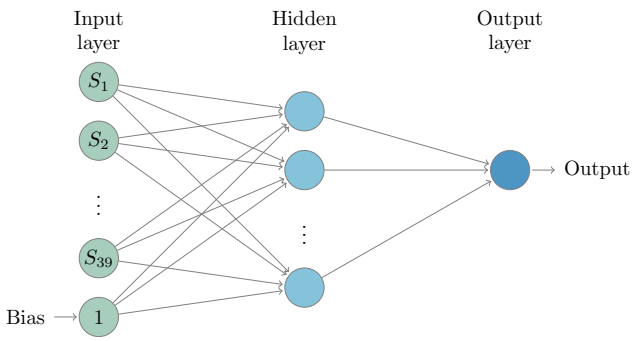


Fig. 2. The dense feedforward neural network architecture The network has one hidden layer, the site frequency spectrum as input and one bias node.

to the minimal nMSE of Fu's, Futschik's, Watterson's, and the linear NN estimator.

For this we divided the test and training data into six subsets with respect to θ defined by $0 = t_0 < t_1 < \cdots < t_6 = 100$, where in the k-th subset $t_{k-1} < \theta \leq t_k$. The t_k are chosen such that the change of coefficients $a_j(\theta)$ in Fu's estimator equals in all subsets, i.e. $\sum_{j=1}^{n-1} a_j(t_k) - a_j(t_{k-1})$ equals for all $1 \leq k \leq 6$. The total loss function is then given by

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{\theta}_i - \theta_i}{\theta_i}\right)^2 \cdot \omega(\theta_i)$$

with $\omega(\theta_i) = \sum_{k=1}^{6} \omega_k \cdot 1_{\{t_{k-1} < \theta_i \leq t_k\}}$. In particular, the loss function can penalize errors in some subsets more than in others.

The weights ω_k are initialised by 1 and ω_k is updated in every iteration of comparable poor performance by setting

$$\omega_k = \omega_k + R \cdot \frac{\max\left(\frac{D_k}{b_k}, 0\right)}{\max_k\left(\max\left(\frac{D_k}{b_k}\right), 0\right)}$$

with

$$b_k := \min\left(\text{nMSE}_k\left(\hat{\theta}_{\text{W}}\right), \text{nMSE}_k\left(\hat{\theta}_{\text{ItV}}\right), \text{nMSE}_k\left(\hat{\theta}_{\text{ItMSE}}\right), \text{nMSE}_k\left(\hat{\theta}_{\text{LinearNN}}\right)\right),$$

and $D_k := \text{nMSE}_k\left(\hat{\theta}_{\text{ANN}}\right) - b_k$, where $\text{nMSE}_k\left(\hat{\theta}\right)$ is the empirical normalized mean square error of $\hat{\theta}$ on the subset where $t_{k-1} < \theta \leq t_k$ and R is drawn uniformly between 0.25 and 0.5.

The training is finished as soon as an iteration does not result in a weight update, i.e. the adaptive neural net performs comparable or better than the model-based estimators and the linear NN on each of the six test subsets.

Results

To investigate the capabilities of neural nets, two dense feedforward neural networks, with zero or one hidden layer, as a

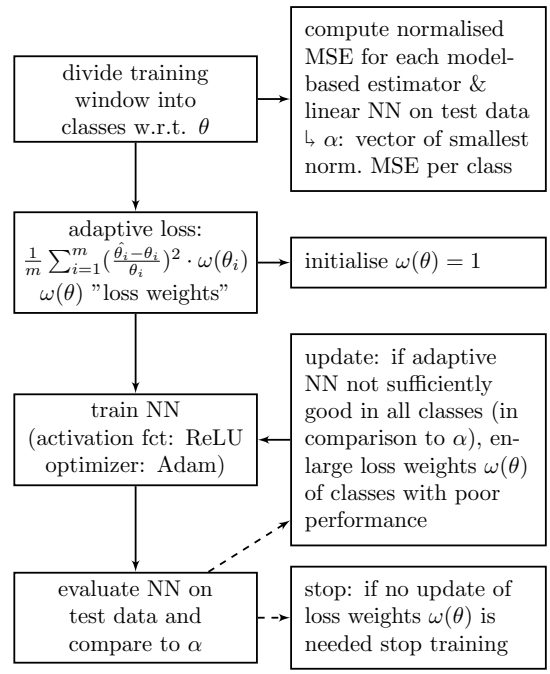


Fig. 3. Overview on adaptive training procedure This figure illustrates the basic principle of the adaptive training procedure used for the adaptive neural networks.

function of the site frequency spectrum were considered in this work. Those fairly simple network architectures were consciously chosen to facilitate a better understanding of the underlying learning process of the neural net. Additionally, neural nets as a function of the site frequency spectrum can be easily compared to the model-based estimators, which use the same information. We observed multiple properties in our simulations:

If no recombination is included in the training and test data, as in Fig 4 A, the neural nets perform comparably to Futschik and Gach's estimator, which has the lowest mean square error among linear estimators in this scenario. If the recombination rate is high in the training and test data, as in Fig 4 D the neural nets perform comparably to the Watterson's estimator, which is a uniformly MVUE for θ in this situation. For training and test datasets with moderate recombination (Fig 4 B-C), the neural nets outperform the other estimators.

So far the considered neural network estimators depend on the recombination rate as the neural networks are trained on datasets with the same recombination rate as the test data. But often the recombination rate is unknown or varies along the genome sequence such that a method that has a low variance regardless of the local recombination rate is desirable. To see if the neural networks can achieve this we trained them on a dataset with variable recombination rate. In this case, due to the restriction to linear estimators, the linear neural net is not able to find an estimator that has a uniformly low mean square error (Fig 5). In contrast, the non-linear neural net with one hidden layer produced an estimator that had the lowest mean square error for almost all possible values of the recombination rate (see Fig 5). This could be beneficial when a sliding window approach is used to estimate the local mutation rate along a genome sequence. In this case, the

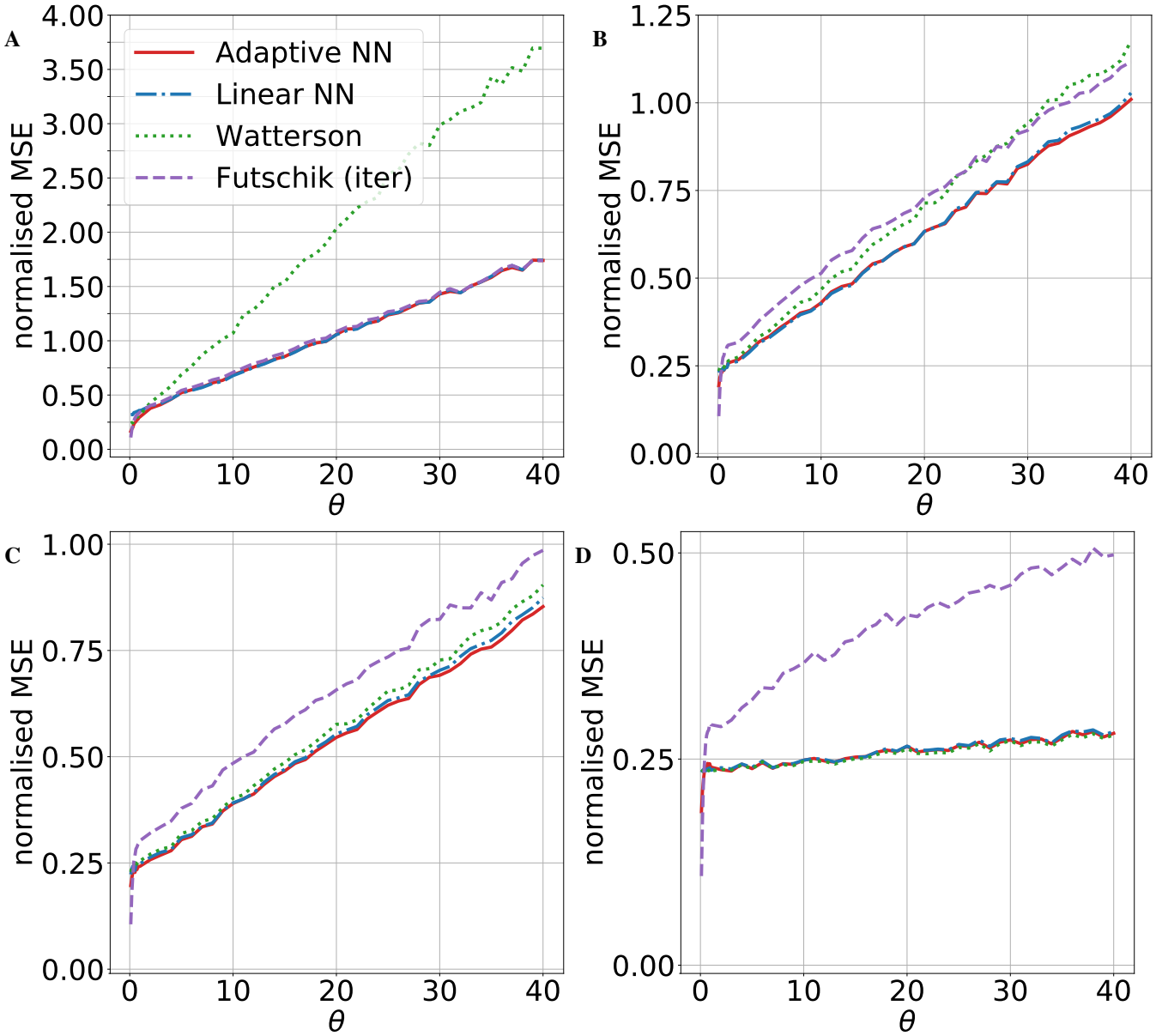


Fig. 4. Performance of mutation rate estimators The normalized mean square errors for four different estimators are shown: The iterated minimal mean square error estimator (Futschik (iter)), Watterson's estimator, and two neural network estimators. A linear network without a hidden layer and an adaptive network with one hidden layer and 200 hidden nodes. All shown neural networks have been trained on datasets with the corresponding recombination rate ρ. For each shown data point 10000 simulations with sample size $n = 40$, recombination rate ρ, and mutation rate θ have been created with msprime. A: recombination rate $\rho = 0$, B: $\rho = 20$, C: $\rho = 35$, D: $\rho = 1000$.

estimate will depend on the local recombination map, such that the neural net trained for variable recombination rates has the advantage to produce good estimates without an explicit estimate of the recombination map.

Discussion and Conclusion

Different optimal estimators for the mutation rate or population size are known in scenarios of low and high recombination. Fu's and Watterson's estimator are linear unbiased estimators with minimal variance in the absence of recombination and in the limit of large recombination rates, respectively. But simulations show that they do not adapt well to other frequencies of recombination. In the case of unknown recombination rates, it is for model-based estimators thus necessary to estimate the recombination rate in a separate step and numerically compute a suitable estimator(3).

We were able to circumvent this step and created a simple neural network that has a low mean square error regardless of the recombination rate. It is worth noting that the true recombination rate was not used as an additional input here. Instead, the neural network adapts to the unknown recombination rate solely based on the observed SFS.

In contrast to the recombination rate, variability in the mutation rate is in practice no issue, although some of the considered estimators depend on the true but unknown mutation rate θ. Iterative procedures denoted ItV and ItMSE here, are usually converging after 3-5 iterations and show almost the same performance as the optimal estimators (S1 Fig). For the neural nets, caution regarding the parameter range of the train-

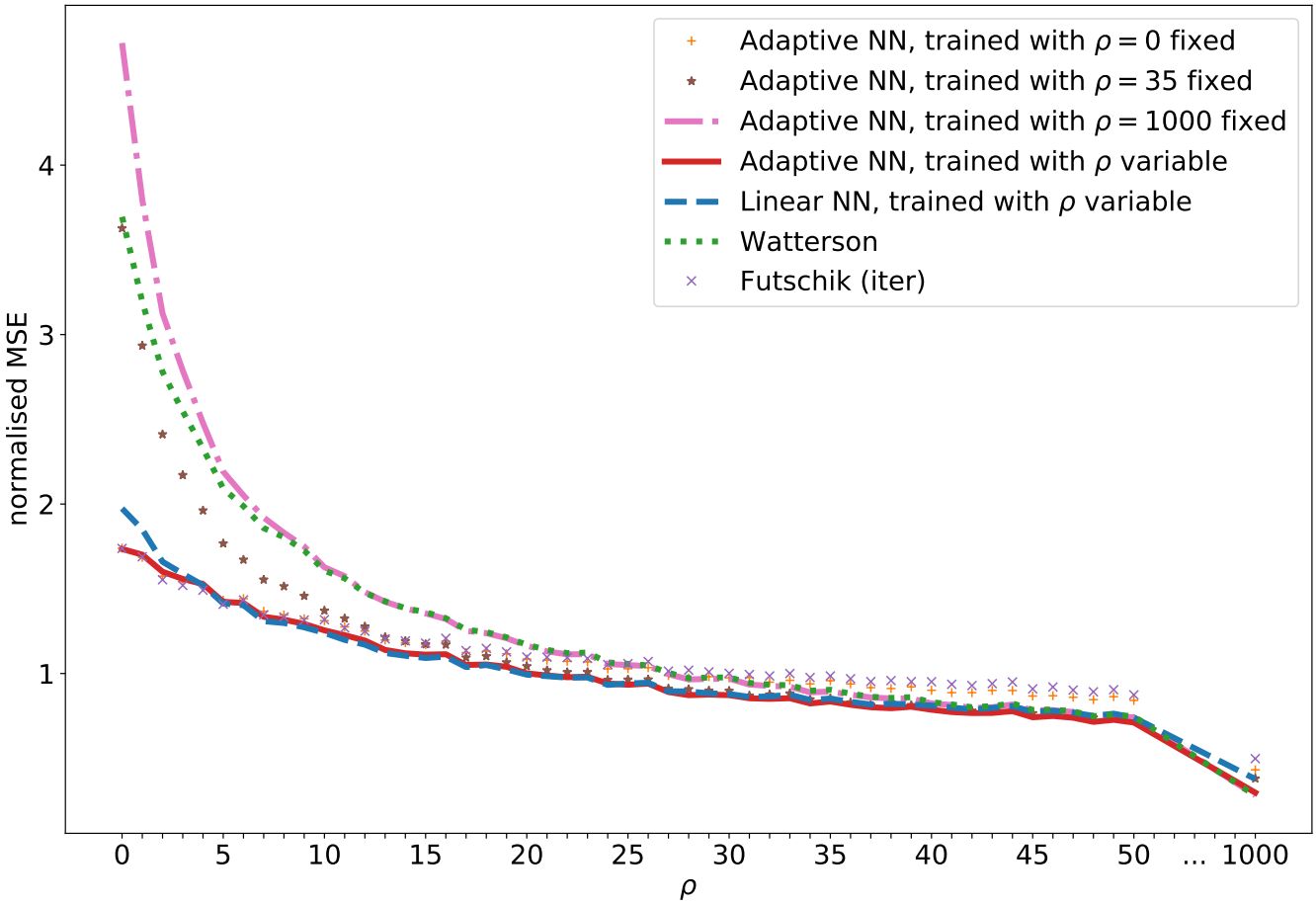


Fig. 5. Mean squared error for variable recombination rates. The normalized mean squared error of feedforward neural network estimators of the mutation rate compared to model-based estimators. In particular, the adaptive networks trained with a fixed recombination rate are compared to the networks trained with variable recombination rates. The recombination rate is given by $\rho = 0, \ldots, 50$ or $\rho = 1000$, while the sample size $n = 40$ and mutation rate $\theta = 40$ are fixed.

ing data is necessary, as they can learn the simulation range and will consequently dramatically over or underestimate the mutation rate outside of this range (S2 Fig).

We observe that the neural nets approximate model-based estimators in situations where the model-based estimators are optimal (no or high recombination) and outperform them in scenarios that are hard to treat theoretically (moderate recombination). Clearly, neural networks are computationally more demanding than model-based approaches. For example, it is necessary to train the network separately for each sample size. However, training is fast, even for the adaptive neural network. For $n = 40$ and training data with 200.000 simulated SFS, training is completed after approximately $20 - 50$ iterations, which takes about 10 minutes on a standard laptop. If these pitfalls are considered, even minimal artificial neural networks are attractive alternatives to model-based estimators of the scaled mutation rate since they can perform uniformly well for different recombination scenarios.

It is worth noting that the observed outperformance of the neural nets was obtained solely based on the site frequency spectrum, but genetic data usually contains more information about the distribution of mutations among samples and their location along the genome. Machine learning methods based on the raw genetic data will require more complex network architectures and could help to automatically extract this additional information. Recent advances in the development of machine learning methods for population genetics (15, 19–21), indicate that expanding the toolbox of neural network-based inference tools to more complex estimation tasks and larger data sets is a promising approach.

Data availability

All results in this manuscript are reproducible using code available at fbaumdicker/ML_in_pop_gen. This includes the simulation of training data, as well as the adaptive training procedure for the neural network estimators. Furthermore, the optimal coefficients can be computed with a python script available in the same repository.

Supporting information

S1 Fig. comparison of iterative and non-iterative estimators for $\theta \in [0, 40]$.

S2 Fig. performance outside the training window: θ in $[0, 500]$.

S3 Fig. normalized bias of estimators.

S4 Fig. coefficients of Fu's estimator with choice of subsets for adaptive training.

ACKNOWLEDGEMENTS
We thank Philipp Harms for helpful discussions and Libera Lo Presti for careful proof-reading of the manuscript. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2064/1 – Project number 390727645, and EXC 2124 – Project number 390838134. PP is supported in part by the Freiburg Center for Data Analysis and Modeling.

Bibliography

1. J F C Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19A: 27–43, 1982.
2. R R Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23:183–201, 1983.
3. Yun-Xin Fu. Estimating Effective Population Size or Mutation Rate Using the Frequencies of Mutations of Various Classes in a Sample of DNA Sequences. *Genetics*, 138:1375–1386, 1994.
4. Jody Hey and John Wakeley. A coalescent estimator of the population recombination rate. *Genetics*, 145(3):833–846, 1997.
5. G A Watterson. On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology*, 7(2):256–276, 1975.
6. Yun-Xin Fu. A Phylogenetic Estimator of Effective Population Size or Mutation Rate. *Genetics*, 136:685–692, 1994.
7. Andreas Futschik and Florian Gach. On the inadmissibility of Watterson's estimator. *Theoretical Population Biology*, 73(2):212–221, 2008.
8. Mukta Paliwal and Usha A. Kumar. Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, 36(1):2–17, 2009.
9. David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, 2016.
10. Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint*, 1910.03193, 2020.
11. Daniel R. Schrider and Andrew D. Kern. Supervised Machine Learning for Population Genetics: A New Paradigm. *Trends in Genetics*, 34(4):301–312, 2018.
12. Sara Sheehan, Yun S. Song, EO Buzbas, DA Petrov, AR Boyko, and A Auton. Deep Learning for Population Genetic Inference. *PLOS Computational Biology*, 12(3):e1004845, 2016.
13. Lex Flagel, Yaniv Brandvain, and Daniel R. Schrider. The unreasonable effectiveness of convolutional neural networks in population genetic inference. *Molecular Biology and Evolution*, 36(2):220–238, 2019.
14. Jeffrey Chan, Jeffrey P. Spence, Sara Mathieson, Valerio Perrone, Paul A. Jenkins, and Yun S. Song. A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in Neural Information Processing Systems*, pages 8594–8605, 2018.
15. Théophile Sanchez, Jean Cury, Guillaume Charpiat, and Flora Jay. Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation. *Molecular Ecology Resources*, pages 1–16, 2020.
16. Jun Shao. *Mathematical Statistics*. Springer-Verlag New York Inc, 2nd edition, 2003.
17. Jerome Kelleher, Alison M Etheridge, and Gilean McVean. Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. *PLoS Comput Biol*, 12(5):1–22, 2016.
18. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations*, 2015.
19. Jeffrey R. Adrion, Jared G. Galloway, and Andrew D. Kern. Predicting the landscape of recombination using deep learning. *Molecular Biology and Evolution*, 37(6):1790–1808, 2020.
20. Anton Suvorov, Joshua Hochuli, and Daniel R. Schrider. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. *Systematic Biology*, 69(2): 221–233, 2020.
21. Hussein A Hejase, Ziyi Mo, Leonardo Campagna, and Adam Siepel. SIA: Selection Inference Using the Ancestral Recombination Graph. *bioRxiv preprint*, 2021.06.22.449427, 2021.

S1 supplemental figure: comparison of iterative and non-iterative estimators for $\theta \in [0, 40]$

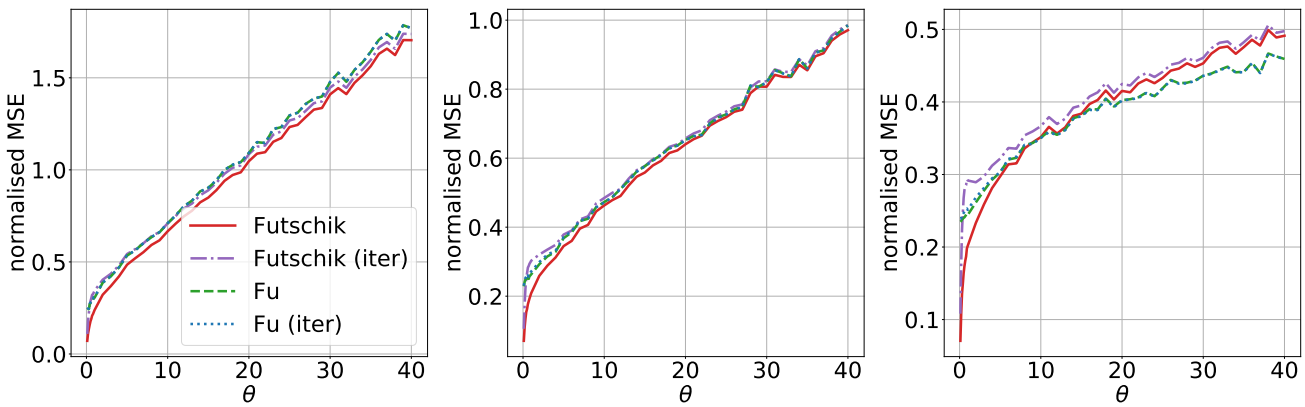


Fig. S1. Mean squared error iterative estimators vs. non-iterative estimators The normalized mean squared errors for four different estimators are shown: The minimal mean square error estimator (Futschik), the iterated minimal mean square error estimator (Futschik (iter)), the minimal variance estimator (Fu) and the iterated minimal variance estimator (Fu (iter)). For each datapoint 10000 simulations with sample size $n = 40$, recombination rate ρ, and mutation rate θ have been created with msprime. A: recombination rate $\rho = 0$ B: $\rho = 35$ C: $\rho = 1000$

S2 supplemental figure: performance outside the training window: θ in $[0, 500]$

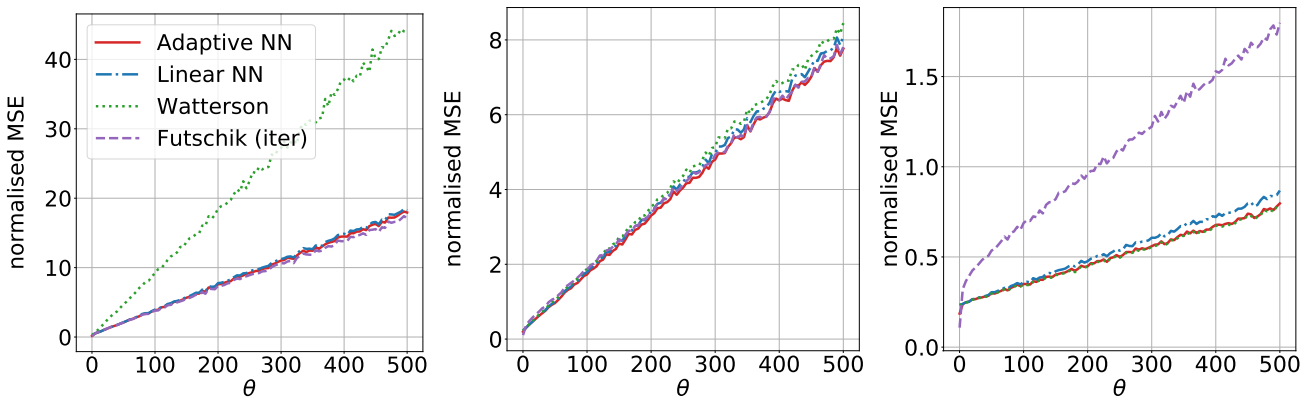


Fig. S2. Mean squared error for estimators up to $\theta = 500$. The normalized mean square errors for four different estimators are shown: The iterated minimal mean square error estimator (Futschik (iter)), Watterson's estimator, and two neural network estimators. A linear network without a hidden layer and an adaptive network with one hidden layer and 200 hidden nodes. All shown neural networks have been trained on datasets with the corresponding recombination rate ρ. For each datapoint 10000 simulations with sample size $n = 40$, recombination rate ρ, and mutation rate θ have been created with msprime. A: recombination rate $\rho = 0$ B: $\rho = 35$ C: $\rho = 1000$

S3 supplemental figure: normalised bias of estimators

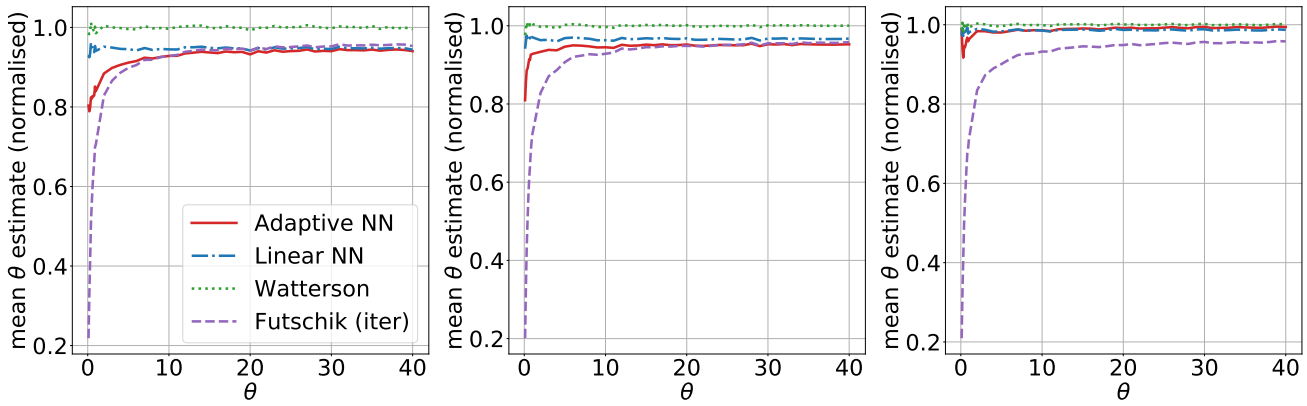


Fig. S3. Normalised bias of estimators The normalised bias for four different estimators are shown: The iterated minimal mean square error estimator (Futschik (iter)), Watterson's estimator, and two neural network estimators. A linear network without a hidden layer and an adaptive network with one hidden layer and 200 hidden nodes. All shown neural networks have been trained on datasets with the corresponding recombination rate ρ. For each datapoint 10000 simulations with sample size $n = 40$, recombination rate ρ, and mutation rate θ have been created with msprime. A: recombination rate $\rho = 0$ B: $\rho = 35$ C: $\rho = 1000$

S4 supplemental figure: coefficients of Fu's estimator with choice of subsets for adaptive training

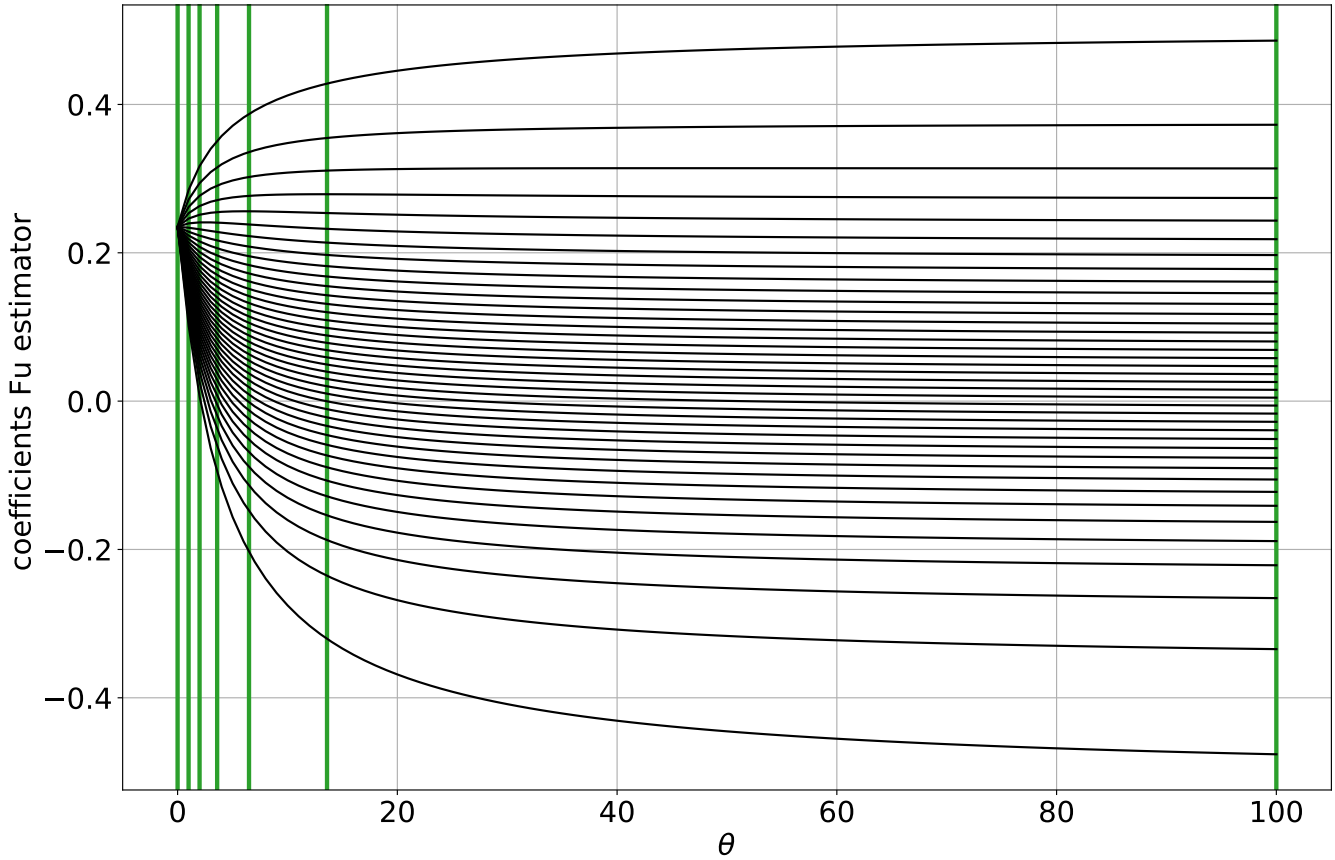


Fig. S4. Coefficients of Fu's estimator with subsets for adaptive training procedure This figure shows the coefficients of non-interative Fu estimator for $n = 40$ (black) and the vertical lines represent the boundaries of the classes used for training the adaptive neural networks for $n = 40$ (green).