Related Inference: A Supervised Learning Approach to Detect Signal Variation in Genome Data

Mario Banuelos
Department of Mathematics
California State University, Fresno
Fresno, USA
mbanuelos22@csufresno.edu

Omar DeGuchy
Department of Applied Mathematics
University of California, Merced
Merced, USA
odeguchy@ucmerced.edu

Suzanne Sindi
Department of Applied Mathematics
University of California, Merced
Merced, USA
ssindi@ucmerced.edu

Roummel F. Marcia
Department of Applied Mathematics
University of California, Merced
Merced, USA
rmarcia@ucmerced.edu

***Abstract*—The human genome, composed of nucleotides, is represented by a long sequence of the letters A,C,G,T. Typically, organisms in the same species have similar genomes that differ by only a few sequences of varying lengths at varying positions. These differences can be observed in the form of regions where letters are inserted, deleted or inverted. These anomalies are known as structural variants (SVs) and are difficult to detect. The standard approach for identifying SVs involves comparing fragments of DNA from the genome of interest and comparing them to a reference genome. This process is usually complicated by errors produced in both the sequencing and mapping process which may result in an increase in false positive detections. In this work we propose two different approaches for reducing the number of false positives. We focus our attention on refining deletions detected by the popular SV tool delly. In particular, we consider the ability of simultaneously considering sequencing data from a parent and a child using a neural network and gradient boosting as a post-processing step. We compare the performance of each method on simulated and real parent-child data and show that including related individuals in training data greatly improves the ability to detect true SVs.**

***Index Terms*—Computational genomics, structural variants, machine learning, deep learning**

I. Introduction

The genome, the complete DNA sequence, of an organism is a long sequence of nucleotides represented by the letters {A,C,G,T}. For mammals, the length of the genome is approximately 3 billion letters whereas for the single celled yeast (*S. cerevisiae*), the length is around 12 million letters. Structurally, DNA consists of two complementary strands. (As such, we use the term "base pair" (bp), "nucleotide" and "letter" interchangeably.) Most individuals within the same species have highly-similar genomes, and differences between the genome of two individuals in the same species are characterized by their lengths. Single-nucleotide variants (SNVs) correspond to a single letter difference. In addition, there are also short regions where multiple letters are inserted or deleted termed In/Dels ($\leq$ 50 letters). Structural variants (SVs) are genomic regions ($>$ 50bp) that vary between members of the same species. SVs may be insertions, deletions, inversions or more general and complex exchanges of DNA segments between regions of the genome [1], [2].

While DNA sequencing costs continue to decline, it is still cost prohibitive to determine the complete DNA sequence for humans. However, because we have a high-quality reference genome for humans and a variety of other species, genomic variants can detected by comparing samples of DNA sequence to the reference. It is far easier to assess the presence of single-letter differences (SNVs) than SVs. Indeed, such technology is readily available to the general public from companies like Ancestry [3] or 23andMe [4]. The dominant method of SV detection is to take samples (fragments) of DNA from an unknown genome and compare them to a high-quality reference. The resulting configuration of mapped fragments is analyzed, and structural differences between the unknown and reference genome should conform to arrangements of mappings that are discordant (with respect to order, length, orientation, etc) or regions of the reference with higher or lower than expected numbers of fragments [1]. For example, a fragment that has portions matching two distant regions of the reference genome, a split-alignment, indicates a potential SV. The problem of SV detection is complicated by errors in the sequencing and mapping process which can create observations that look like true SVs.

One approach to improve SV detection is to simply take more samples from the test genome to separate the true from false predictions, but this approach will result in an increase in cost. An alternative approach is to make better predictors which explicitly incorporate known biases and multiple signals from related individuals [5], [6]. We follow in this spirit; but rather than predefining the signals or features of interest, we use a deep-learning approach by building a feed-forward neural network. Machine-learning approaches are becoming more common in genomics and have been previously used for

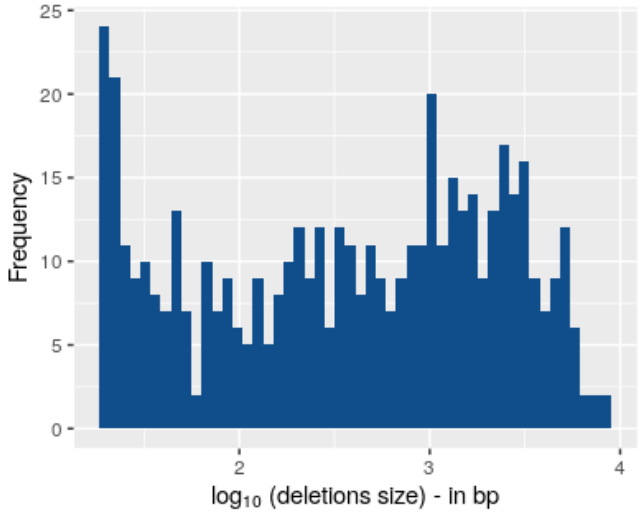

Fig. 1. Size distribution of 500 simulated deletions, which is based on deletion sizes from the Database of Genomic Variants. We note that the majority of deletions will be less than 1000bp in length.

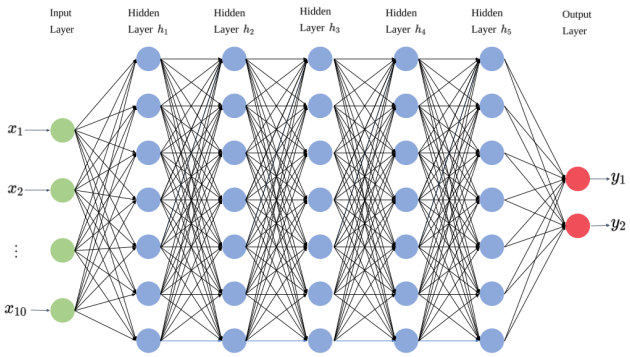

Fig. 2. Neural network architecture used for identifying structural variation in genomic data. The inputs $x_1 \dots x_{10}$ represent the features corresponding to the delly deletion calls. Each entry of the output $y \in \mathbb{R}^2$ is the probability corresponding to the absence or presence of a signal variation.

variant detection [7]–[10]. However, our work is distinguished from these methods by the fact that we consider the ability of simultaneously considering sequencing data from a parent and a child. Because the *de novo* formation rate of SVs is low, all but a very small minority of variants present in a child's genome will have been inherited from the parent. We have previously used parent-child trios to improve SV detection but not in a deep-learning framework (see e.g., [11]).

In this work, we propose two different approaches for reducing the number of false positives SV predictions from a popular SV tool delly [5]. For simplicity, we focus on deletions (see Fig. 1 for the estimated size distribution of these variants in humans). We compare the performance of each method on both simulated and real parent-child sequencing data. Our results on both simulated and real data demonstrate that deep-learning and gradient boosting are powerful tools for SV detection but that including related individuals in the data set greatly boost the ability to recover true SVs.

II. Method

In this section we describe the two machine learning approaches implemented for SV detection. The first method uses a feed forward neural network of fully connected layers. While well established as the state of the art in the world of computer vision, neural networks are emerging as powerful tools for processing tabular data [12]–[14]. Unlike images, time series, or text datasets, tabular data consists of a columnar format where each column contains variable information for a given number of data points. The second method, XGBoost, has already been established as a workhorse for classification applications in the tabular data domain. XGBoost is an ensemble method that uses decision-trees in concert with gradient descent in order to improve performance. In the following section we describe the parameters as well as the preprocessing to the data for each method.

A. Neural Network

We begin with the general formulation of a feed forward deep neural network whose defining characteristic is the number of hidden layers. In order to describe the architecture implemented in this paper, we adopt the generalized formulation presented in [15]. Since our network consists of fully connected layers, the process can be described by the equation

$$h_i = \phi(W_i^T h_{i-1} + b_i),$$

where $h_{i-1} \in \mathbb{R}^m$ is the output of the previous layer and $h_i \in \mathbb{R}^n$ is the output of the current layer with n being the number of neurons in the current layer and m being the number of neurons in the previous layer. The weight matrix $W_i \in \mathbb{R}^{m \times n}$ consists of trainable parameters and $b_i \in \mathbb{R}^n$ is the bias vector. Finally ϕ describes the activation function which provides a non-linearity to the process. Each data pair can be described as (X_j, y_j) for $j = 1 \dots q$, where q is the number of training points. The input $X_j \in \mathbb{R}^{10}$ consists of the features provided by delly and the target $y_j \in \mathbb{R}$ is the true binary label indicating the absence (0) or presence (1) of a structural variant. The architecture consists of one input layer $h_0 = \phi(W_0^T X_j + b_o)$, five hidden layers $h_1 \dots h_5$ and an output layer $o = \phi(W_o^T h_4 + b_o)$. All hidden layers contain 120 neurons and use the ReLU activation function with the exception of the output layer which consists of two neurons and uses a log softmax activation function [16]. Before each layer we apply batch normalization to improve the performance and stability of our network [17].

Given the two possible classes (the presence or absence of an SV), the output of the neural network is a distribution of probabilities $\hat{p_k}$ with $k = 1 \dots K$ with K equal to the number classes (in this case $K = 2$). We seek to minimize the Cross Entropy cost function

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log(\hat{p}_k^{(i)}) \quad (1)$$

TABLE I
PARAMETERS USED FOR TRAINING XGBOOST ALGORITHM.

Simulated Data Experiment	
Parameter	**Value**
Estimators	800
Min. Child Weight	0.5
Max. Depth	7
Gamma	1
Subsample Ratio	1.0
Column Subsampling	1.0
Platinum Genomes Data Experiment	
Parameter	**Value**
Estimators	400
Min. Child Weight	20
Max. Depth	3
Gamma	0.5
Subsample Ratio	0.8
Column Subsampling	0.6

where $y_k^{(i)}$ is the true probability of the k^{th} class, consisting of 0 or 1, and m is the number of training samples for a given batch [18]. We tuned hyperparameters with the Adam optimization algorithm to minimize (1) and selected batch size of 16 over 100 epochs. Before training and testing, the data is normalized so that all columns of the features have zero mean and unit variance.

B. XGBoost

Extreme gradient boosting or XGBoost is an optimized implementation of the Gradient Boosting Trees Algorithm. It has been shown to be highly effective in classification problems across a variety of disciplines [19]–[21]. XGBoost is an ensemble machine learning algorithm built using decision trees. Given our data set X_j, y_j, XGBoost creates an ensemble of K additive Classification and Regression Trees (CART) which we express as $T_1(X_i, y_i) \dots T_K(X_i, y_i)$ in order to predict the class label y_i. The prediction scores for each CART are summed up as a final score expressed by the equation

$$\hat{y_i} = \sum_{k=1}^{K}, f_k \in F \tag{2}$$

where f_k is a tree structure and F is the space of all trees [20], [22]. Given the final score, we seek to minimize the cost function

$$J(\Theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k}^{K} \Omega(f_k) \tag{3}$$

where the first term l in (3) is a data fidelity term between the prediction $\hat{y_i}$ and the target y_i. The regularization term Ω penalizes the model complexity to avoid overfitting. We refer the reader to [22] for the optimization routine for (3) which cannot be optimized using traditional optimization methods in Euclidean space.

Hyperparameter tuning is as essential to XGBoost as it is to the neural networks. We use grid search methods in order to perform parameter sweeps using k fold cross validation in order to find the optimal model parameters. The optimal hyperparameters for both models are shown in Table I. In either case the learning rate was set to 0.02 and a binary logistic objective function was used as a cost function. All parameters not listed in Table I were set to the default values. Both datasets were preprocessed using a min-max normalization.

TABLE II
FEATURES CORRESPONDING TO THE DELLY DELETION CALLS AND A DESCRIPTION OF EACH FEATURE.

Features	**Description**
Chr	Chromosome
Start	Predicted start position of deletion
End	Predicted end position of deletion
FILTER	PE/SR support < 3 or mapping quality < 20
IMPRECISE	SR support > 0
PE	Number of paired-end reads supporting deletion
MAPQ	Median mapping quality of paired-ends
CIPOS	Paired-end confidence interval around Start
CIEND	Paired-end confidence interval around End
SR	Number of split-reads supporting deletion

III. NUMERICAL EXPERIMENTS

A. Simulated Data

Using the first twelve chromosomes of the hg19 build of the human reference genome, we introduced 500 deletions using RSVSim [23]–[25]. We simulated corresponding reads with read lengths $L = 75$bp and $L = 150$bp using `dwgsim` and aligned reads with `speedseq` [26], [27]. We follow a similar approach to simulate 2 different individuals, one offspring derived from the mutated parent and one unrelated individual. Since the rate of *de novo* variations is less than one per generation, the offspring only had 3 novel deletions not present in the parent [28], [29].

To obtain candidate genomic variant locations, we call deletions with delly [5]. We incorporated variant call format (vcf) files into our Python workflow using `cyvcf2` to extract a total of 10 features corresponding to the delly deletion calls [30]. We summarize these in Table II. Since we know the truth signal for each individual, we apply our proposed methods to reduce the number of false positives predicted by delly. For both methods, we use the parent as the training data, and the offspring as the testing data.

B. Platinum Genomes Data

Following a similar framework as the simulated data, we apply our proposed methods to delly calls for individual NA12878 (child) and NA12891 (mother) from the CEU population [31]. Both individuals are from a 17-member pedigree, where the true genomic variants have been experimentally validated. We filter out deletions smaller than 50bp from the truth set. From the catalogued true variations, we create the truth signal $\vec{y}$ corresponding to the delly predictions. In this case, we use NA12891 as the training data and NA12878 as testing set.

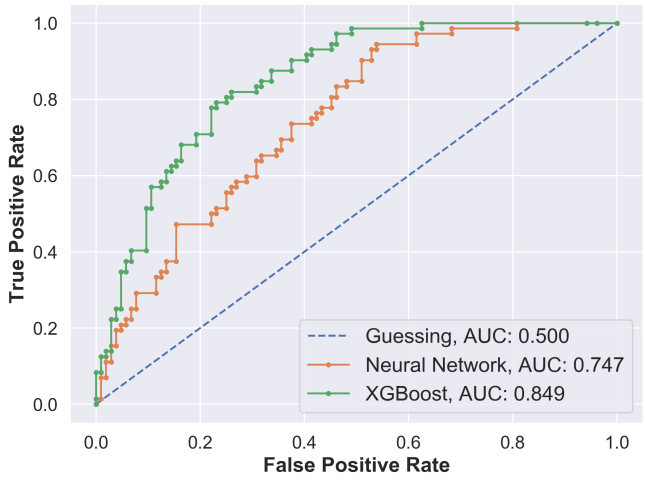

Fig. 3. Receiver operating characteristic curve (ROC) for the simulated data offspring signal. We also report the area under the curve (AUC) for each method.

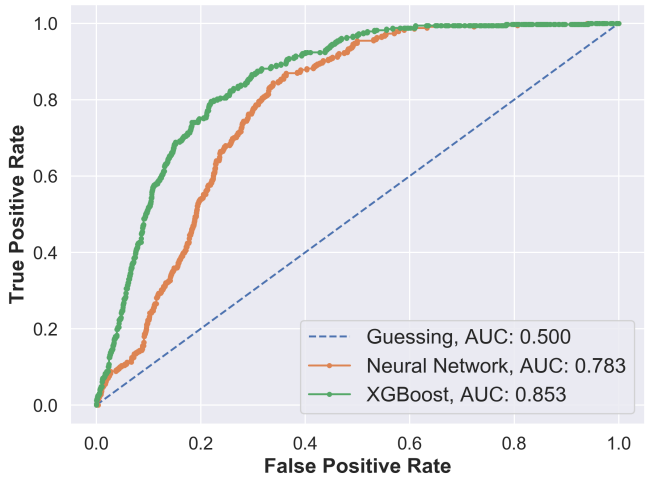

Fig. 4. Receiver operating characteristic curve (ROC) for the *Platinum Genomes* NA12878 offspring signal. We also report the area under the curve (AUC) for each method.

IV. Results

The proposed methods were able to significantly reduce the number of false positive classifications identified by the delly SV caller. In Table III we report a variety of metrics to evaluate performance of the methods on both datasets. It is clear from both Table III and the AUC curves in Figures 3 and 4 that XGBoost is clearly outperforming the Neural Network. Even though the ensemble method improves on the scores of the Deep Learning method, the authors feel that the results are promising and warrant further exploration. We are also encouraged by that fact that the AUC for both the simulated dataset and the real dataset behaved similarly under both methods.

For the simulated data, using an unrelated individual for the training data yields less predictive power for the offspring

TABLE III
Performance metrics for both experiments.

Simulated Data				
Method	Precision	Recall	F1	AUC
Neural Network	0.66	0.67	0.65	0.75
XGBoost	0.76	0.77	0.75	0.85
Platinum Genomes Data				
Method	Precision	Recall	F1	AUC
Neural Network	0.59	0.50	0.48	0.78
XGBoost	0.65	0.53	0.53	0.853

(results not shown). Although the simulated data reflects biologically-informed deletion sizes, the training and testing set resulted in balanced number of observations for each class. In contrast, for the *Platinum Genomes* data, we find less than one in five delly predictions to be true deletions. This imbalance may account for less improvement in precision and recall than in the simulated data tests. Including more related individuals across multiple generations may also improve the reduction of false positives in SV callers.

V. Conclusions

We present a supervised learning framework that incorporates relatedness information to reduce the number of false positives in SV-callers, like delly. Although we present our results in the context of deletions, our framework can be adapted for predicting other classes of structural variants. In the context of applying such methods, we also find that population-level supervised learning techniques may be more appropriate in refining variant predictions than an approach that does not consider differences in ancestry.

Acknowledgment

This work was supported by National Science Foundation Grant IIS-1741490.

References

[1] S. S. Ho, A. E. Urban, and R. E. Mills, "Structural variation in the sequencing era," *Nature Reviews Genetics*, pp. 1–19, 2019.
[2] J. Weischenfeldt, F. Symmons, O.and Spitz, and J.O. Korbel, "Phenotypic impact of genomic structural variation: insights from and for human disease," *Nature Reviews Genetics*, vol. 14, no. 2, pp. 125–138, 2013.
[3] "Ancestry," https://www.ancestry.com/, Accessed: 2020-02-29.
[4] "23andme," https://www.23andme.com/, Accessed: 2020-02-29.
[5] T. Rausch, T. Zichner, A. Schlattl, A. M. Stütz, V. Benes, and J. O. Korbel, "Delly: structural variant discovery by integrated paired-end and split-read analysis," *Bioinformatics*, vol. 28, no. 18, pp. i333–i339, 2012.
[6] S. S Sindi, S. Önal, L. C. Peng, H. Wu, and B. J. Raphael, "An integrative probabilistic model for identification of structural variation in sequencing data," *Genome biology*, vol. 13, no. 3, pp. R22, 2012.
[7] G. H. Lubke, C. Laurin, R. Walters, N. Eriksson, P. Hysi, T. D. Spector, G. W. Montgomery, N. G. Martin, S. E. Medland, and D. I. Boomsma, "Gradient boosting as a snp filter: An evaluation using simulated and hair morphology data," *Journal of data mining in genomics & proteomics*, vol. 4, 2013.
[8] H. Park, S. Chun, J. Shim, J. Oh, E. J. Cho, H. S. Hwang, J. Lee, D. Kim, S. J. Jang, S. J. Nam, et al., "Detection of chromosome structural variation by targeted next-generation sequencing and a deep learning application," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.
[9] D. Antaki, W. M. Brandler, and J. Sebat, "Sv2: accurate structural variation genotyping and de novo mutation detection from whole genomes," *Bioinformatics*, vol. 34, no. 10, pp. 1774–1777, 2018.

[10] E. Alzaid and A. E. Allali, "Postsv: A post–processing approach for filtering structural variations," *Bioinformatics and Biology Insights*, vol. 14, pp. 1177932219892957, 2020.

[11] M. Spence, M. Banuelos, R. F. Marcia, and S. Sindi, "Detecting novel structural variants in genomes by leveraging parent-child relatedness," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec 2018, pp. 943–950.

[12] B. Baesens, R. Setiono, C. Mues, and J. Vanthienen, "Using neural network rule extraction and decision tables for credit-risk evaluation," *Management science*, vol. 49, no. 3, pp. 312–329, 2003.

[13] H. S. Bhat and S. J. Goldman-Mellor, "Predicting adolescent suicide attempts with neural networks," *arXiv preprint arXiv:1711.10057*, 2017.

[14] A. Khemphila and V. Boonjing, "Heart disease classification using neural network and feature selection," in *2011 21st International Conference on Systems Engineering*. IEEE, 2011, pp. 406–409.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.

[16] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics New York, 2001.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019.

[19] X. Chen, L. Huang, D. Xie, and Q. Zhao, "Egbmmda: extreme gradient boosting machine for mirna-disease association prediction," *Cell death & disease*, vol. 9, no. 1, pp. 1–16, 2018.

[20] Ismail Babajide M. and F. Saeed, "Bioactive molecule prediction using extreme gradient boosting," *Molecules*, vol. 21, no. 8, pp. 983, 2016.

[21] J. Fan, X. Wang, L. Wu, H. Zhou, F. Zhang, X. Yu, X. Lu, and Y. Xiang, "Comparison of support vector machine and extreme gradient boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in china," *Energy Conversion and Management*, vol. 164, pp. 102–111, 2018.

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[23] International Human Genome Sequencing Consortium et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, 2001.

[24] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, "Detection of large-scale variation in the human genome," *Nature genetics*, vol. 36, no. 9, pp. 949–951, 2004.

[25] C. Bartenhagen and M. Dugas, "Rsvsim: an r/bioconductor package for the simulation of structural variations," *Bioinformatics*, vol. 29, no. 13, pp. 1679–1681, 2013.

[26] N. Homer, "Dwgsim: whole genome simulator for next-generation sequencing," *GitHub repository*, 2010.

[27] C. Chiang, R. M. Layer, G. G. Faust, M. R Lindberg, D. B. Rose, E. P. Garrison, G. T. Marth, A. R. Quinlan, and I. M. Hall, "Speedseq: ultra-fast personal genome analysis and interpretation," *Nature methods*, vol. 12, no. 10, pp. 966, 2015.

[28] W. P. Kloosterman, L. C. Francioli, F. Hormozdiari, T. Marschall, J. Y. Hehir-Kwa, A. Abdellaoui, E. Lameijer, M. H. Moed, V. Koval, I. Renkens, et al., "Characteristics of de novo structural changes in the human genome," *Genome research*, vol. 25, no. 6, pp. 792–801, 2015.

[29] D. C. Jeffares, C. Jolly, M. Hoti, D. Speed, L. Shaw, C. Rallis, F. Balloux, C. Dessimoz, J. Bähler, and F. J. Sedlazeck, "Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast," *Nature communications*, vol. 8, no. 1, pp. 1–11, 2017.

[30] B. S. Pedersen and A. R. Quinlan, "cyvcf2: fast, flexible variant analysis with python," *Bioinformatics*, 2017.

[31] M. A. Eberle, E. Fritzilas, P. Krusche, M. Källberg, B. L. Moore, M. A. Bekritsky, Z. Iqbal, H. Chuang, S. J. Humphray, A. L. Halpern, et al., "A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree," *Genome research*, vol. 27, no. 1, pp. 157–164, 2017.