Machine Learning Random Forest for predicting onco-somatic variants NGS analysis

Eric Pellegrino (eric.pellegrino@univ-amu.fr)
APHM, CHU Nord
Coralie Jacques
APHM, CHU Nord
Nathalie Beaufils
APHM, CHU Nord
Isabelle Nanni
APHM, CHU Nord
Antoine Carlioz
APHM, CHU Nord
Philippe Metellus
Centre Hospitalier Clairval
L'Houcine Ouafik
APHM, CHU Nord

Research Article

Keywords: Machine Learning Random Forest, cancer diagnosis and treatment

Posted Date: April 27th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-426134/v1

Machine Learning Random Forest for predicting onco-somatic variants NGS analysis

Eric Pellegrino[1,*], Coralie Jacques[1], Nathalie Beaufils[1], Isabelle Nanni[1], Antoine Carlioz[1], Philippe Metellus[1,2], and L'Houcine Ouafik[1,3]

[1]APHM, CHU Nord, Service d'Onco-Biologie, Marseille, France
[2]Centre Hospitalier Clairval, Departement de Neurochirurgie, Marseille, France
[3]Aix Marseille Univ, CNRS, INP, Inst Neurophysiopathol, Marseille, France
[*]eric.pellegrino@univ-amu.fr

ABSTRACT

Motivation: Since 2017, we are using IonTorrent NGS platform in our hospital in order to diagnose cancer and treatment. Analysis variants at each run take us a longtime and we are still struggling with some variants which look correct on the first look at their metrics but found to be negative when we look further into them. Can any Machine Learning algorithm help us to classify NGS variant calling ? This has determined us to investigate which ML could fit to our NGS data and to develop a tool which can be implemented in Routine in order to help Biologists. **Introduction:** Nowadays, one of medicine challenges is processing a significant amount of data. It's particularly true in molecular biology with the advantage of Next Generation Sequencing (NGS) for molecular tumor profile determination and treatment selection. In addition to bioinformatics pipelines, Artificial Intelligence (AI) can offer a very valuable help in analyzing. Generating sequencing data from patient DNA samples has become easy to perform in clinical trials. But analyzing the huge amount of genomic or transcriptomic data and extracting the key biomarkers associated with a clinical response to a specific therapy requires a formidable combination of scientific expertise, biomolecular skill and a panel of bioinformatics and biostatistics tools, in which artificial intelligence is now a success factor in developing future routine diagnostics. However, cancer genome complexity and technical artifacts make identification of real variants a challenge. We present a Machine Learning method to classify pathogenic Single Nucleotide Variants (SNVs), SNP (Single Nucleotide Polymorphism), MNVs (Multiple Nucleotide Variants), Insertion, Deletion detected by NGS from tumors specimens for Colorectal, Melanoma, Lung and Glioma cancer. **Methods:** We compared our NGS data to different machine learning algorithms using the 10-fold cross validation method and to neural networks (Deep Learning) in order to measure the performance of the different ML algorithms and determine which one is a valid model for confirming NGS variant calls in cancer diagnostic. We trained our Machine Learning with 70 % of our data samples, extracted from our local database (our data structure had 7 parameters: chromosome, position, exon, variant allele frequency, minor allele frequency, coverage and protein description) and validated it with 30 % remaining. The model offering the best accuracy was chosen and implemented in NGS analysis routine. The artificial intelligence was developed with R script language version 3.6.0. **Results:** We trained our model on 102011 variants. Our best error rate (0.22%) was found with Random Forest Machine Learning (ntree=500 and mtry=4) with an AUC of 0.99. Neural Networks achieved some good scores. The final trained model with Neural Network was able to achieve an accuracy of 98% and a ROC-AUC of 0.99 with validation data. We tested our RF model to interpret more than 2000 variants from our NGS database: 20 variants were misclassified (error rate <1%). The errors were nomenclature problems and false positive. After adding false positive in our training database and implementing our RF model in routine, our error rate was always < 0.5%. **Conclusion:** Our RF model shows excellent results for onco-somatic NGS interpretation and it could easily be implemented in other molecular biology laboratories. AI is taking an increasingly important place in molecular biomedical analysis and could be very helpful on processing of amount medical data. Neural Networks showed a good capacity in the classification of variants and in the future may be useful in the prediction of more complex variants.

Introduction

Tumor molecular profile has become in recent years a major element in diagnosis, classification, and therapeutic management. As a result, the number of molecular analysis has increased considerably. These last years, the Next Generation of Sequencing (NGS), with onco-somatic genes panel, has become one of the reference techniques used routinely in many laboratories for molecular typing. Data analysis from sequencing requires the use of bioinformatic pipelines, on one hand, allowing aligning the results from sequencing on the reference sequence of the genome; and on the other hand, to filter mapped reads against the reference genome to perform variant analysis, including variant calling and predicting the effects found variants producing on known genes. After all these steps, Biologists can begin the interpretation: for each patient, they have to check a list of quality parameters and to determine mutation's pathogenicity, trying to establish which influence these mutations do on the protein.

For example, if it is a synonymous variant, it will probably have low influence on the gene as such a change codes for the same amino acid (Supplementary Figure b). However, if it is a large deletion, you can assume that it will have a large effect on the gene function. Artificial Intelligence (AI) can be very helpful for all interpretation process. AI is defined as the set of theories and techniques used to produce machines capability of simulating human intelligence. It's taking an increasingly important place in our daily life and especially in the medical world with development of biotechnologies. Many studies are published and show extraordinary results in healthcare's domain like radiomic, digital pathology[1]. Machine learning, Neural Networks and Deep learning are part of artificial intelligence. Both approaches result in computers being able to make intelligent decisions. Neural networks make up the backbone of deep learning algorithms. In fact, it is the number of node layers, or depth, of neural networks that distinguishes a single neural network from a deep learning algorithm, which must have more than three. However, Deep learning is a subcategory of Machine learning because it relies on unsupervised learning. Both technologies absolutely need to have large amounts of data, which they use as a basis for learning. The similarities end there. Machine learning is the oldest and simplest technology. It is based on an algorithm which adapts the system itself based on feedback from humans. The implementation of this technology implies the existence of organized data. The system is then fed with structured and categorized data allowing it to understand how to classify new similar data. Based on this classification, the system then performs the programmed actions. After a first phase of use, the algorithm is optimized based on the developer's feedback, which informs the system of incorrect classifications and indicates the correct categories. In AI, Machine Learning (ML) is defined by algorithms with the ability to learn without being explicitly programmed. Random forest (RF) is a ML algorithm which is defined by an ensemble of learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training phase and outputting the classes (classification problem) or mean prediction (regression problem) of the individual trees. RF corrects decision trees to prevent any overfitting during the training phase[2]. The algorithm builds each of the decision trees that compose it, training them all with a subset of the problem data. It randomly chooses data to which part of the decision trees will not have access while another will have access to it in order to make them totally blind to others and to ensure that all the decision trees have a different experience of the problem. Once all the decision trees have been trained, RF makes its decisions regarding the classification or regression problem to be solved, by voting on all the decision trees that compose it. The RF's randomness is found in the sampling on which the trees are built and in the selection of the variables on which the segmentation is carried out. That's why RF remains a black box in many cases. Indeed, if you build a forest of 500, 1000, even 10,000 units, you will not be able to have a look at the mechanism of each tree. Deep learning does not need structured data. The system operates on several layers of neural networks, which combine different algorithms based on the human brain. Thus, the system is able to work from unstructured data. This approach is particularly suitable for complex tasks, when all aspects of the objects to be treated cannot be categorized upstream. The Deep learning system itself identifies the discriminating characteristics. In each layer, it searches for a new specific criterion of the object, which is used as a basis to decide on the classification retained for the object at the end of the process.

With Deep learning, the system itself identifies the discriminating characteristics of the data, without the need for prior categorization. The system does not need to be trained by a developer. It assesses by itself the need to modify the classification or create new categories based on the new data. While machine learning works from a controllable database, deep learning needs a much larger amount of data. The system must have much more entries to give reliable results. In our study, we are interested in the use of Machine Learning and Neural Networks in order to interpret NGS onco-somatic results. We focused on Random Forest ML Algorithm as it was the one which offers the best accuracy. In parallel, we tried to find if any Neural Network architecture can achieve a good somatic classifier. We validated our RF and ANN (Artificial Neural Network) model by comparing our results to Biologist's results.

Materials and Methods

Methodology and model selection NGS Analysis

Our dataset is composed of a cohort of 102011 SNVs, SNPs and indels variants coming from of our 27 genes panel sequenced for colorectal, melanoma, lung and 18 genes for glioma cancer. Each SNV, SNP and indels were manually inspected and labeled as either pathogenic (51008 variants) or benign (51003 variants) as part of clinical laboratory workflow. Our panel Oncomine Solid Tumor and Oncomine Solid Tumor + (OST/OST+) contains hotspots of target region of 27 interest's genes in oncogenesis: KRAS, EGFR, BRAF, PIK3CA, AKT1, PTEN, NRAS, HRAS, STK11, MAP2K1, ALK, DDR2, CTNNB1, MET, TP53, SMAD4, FBXW7, FGFR1, FGFR2, FGFR3, NOTCH1, ERBB4, KIT, PDGFRa, POLE, RET, ROS. Our in house Glioma panel was made in order to define glioma grade and we target the following genes by sequencing full sequences and hotspots of: AKT1, ATRX, BRAF, CDKN2A, CIC, EGFR, FGFR1, H3F3A, HIST1H3B, IDH1, IDH2, NOTCH1, PIK3CA, PIK3R1, PTEN, PTPN11, TERT and TP53.

For each clinical samples, after tumoral genomic DNA extraction by Maxwell RSC DNA FFPE kit or Maxwell RSC Cell DNA (Promega, France), we performed Sequencing with Ion Torrent S5XL (ThermoFisher, France) with a sensitivity of 5% and a minimum coverage of 500X. Then, data from sequencing areanalyzed throughout 2 pipelines. First pipeline was developed

by ThermoFisher on IonTorrent Suite + Ion Reporter. IonTorrent Suite generates FASTQ data and ensures BAM (Binary Alignment Mapping) alignment with the hg19 reference genome by using TMAP (Torrent Mapping Alignment Program). Ion Reporter makes variant caller and variant annotations. The second pipeline was developed in our laboratory and uses open source softwares such as bwa mem for alignment, Samtools for mpileup, VarScan2 as variant caller and VEP Ensembl for annotations. All data are stored in our local MySQL database.

In the manual variant analysis, biologists review variant as the following process if these following statements are true:

- Variant Allele Frequency $\geq 5\%$
- Minor Allele Frequency is absent
- Variant reads ≥ 500
- Amino Acid change is different from silent ($\neq p.(=)$ or $\neq p.?$) in the case of variant's location is in exonic region. This statement is not applied for example for MET mutations in cancer lung which are in splice site region and coded as $p.?$. This is also the case for Glioma panel where TERT mutation hotspot is in the promoter region.
- Variants are present in several onco somatic database such as COSMIC, VarSome, TP53 IARC and known as pathogenic or uncertain significance.
- Alignment Sequences are clear and show no strand bias in the region where the variant is. This is performed thanks to Integrative Genomics Viewer (IGV). It allows us to eliminate false positive.
- In the case of hotspot mutation variant, if he has *reads* ≥ 500 and VAF above or upper than 5% the variant is validated by the Biologist.
- We look the presence of these pathogenic variants in our second in-house pipeline in order to validate it

With these requirements, Biologist decides whether a variant can be validated or not according to patient's pathology.

Construction of the training dataset and Feature selection

Before starting the construction of dataset, we need to calculate entropy and information gain. Calculating information and entropy is a useful tool in machine learning and is used as the basis for techniques such as feature selection, building decision trees, and, more generally, fitting classification models. Entropy is defined by the equation:

$$H(p_1 \dots p_n) = \sum_{i=1}^{n} p_i log_2 p_i$$

where p_i is the probability of value i and n is the number of possible values.

Entropy is the measurement of the impurity or randomness in the data points, and it is calculated between 0 and 1[16]. Entropy is the lowest (no disorder = 0) at extremes (both end = 1) and maximum (high disorder $= \frac{1}{2}$) in the middle. The concept of entropy plays an important role in calculating Information Gain (IG)[17]. It is applied to quantify which feature provides maximal information about the classification based on the notion of entropy[17], so it answers this question by measuring how much "information" a feature gives us about the class[18]. The information gain is the product of class probabilities with logarithm base 2 of that class probability.

$$IG = H_p - \sum_{i=1}^{n} p_{ci} H_{ci}$$

Where H_p is the entropy of the parent node, n is the number of values of our target variable, p_{ci} is the probability of an observation is in child i, and H_{ci} is the entropy of child segment i.

Construction of the training dataset

Our data were extracted from our local MySQL database and they needed to get cleaned and unified in order to make them reliable for any analytic or machine learning. Some data being recorded did not follow standard schemas or break integrity constraints. Much of the work involved cleaning dataset, removing errors and anomalies or replacing observed values with true values from data. As, at each NGS run, Biologists review variants and decide whether a variant can be validated or not. So each variant is flagged YES or NO (in our dataset: $validationVariant = y = \{YES = 1; NO = 0\}$). Yes means variant is considered as pathogenic for the pathology and NO as benign.

We decided to define the following data structure for machine learning with 7 parameters:

1.Chromosome (Chr) => Chromosome number (integer)

2.Position (POS) => Chromosome position (Big Integer)

3.Exon => Integer

4.Variant Allele Frequency (VAF) => Float

5.Minor Allele Frequency (MAF) => defined as Boolean 1 or 0

6.Coverage => Read depth (Big integer)

7.Protein Description (protdesc) => Amino Acid Change (Boolean) {0 : silent, 1 : protein change}

A total of 51008 compiled of onco somatic variants are labelled as pathogenic from colorectal, lung, melanoma and glioma cancer and a total of 51003 are considered as SNP or not having enough requirements to get retained by the Biologist.

We formalized the somatic mutation prediction problem as a classification problem. Each candidate mutation site is represented by a feature vector x={Chromosome,Position,Exon,VAF, MAF, Coverage, Protein Description} with 7 inputs described previously:

$$x_i = \sum_{i=1}^{n=7} x_i = \{x_1, \ldots x_7\}$$

The goal is to predict the variable y ={Validation variant}$= \left\{ \frac{0=FALSE(NO)}{1=TRUE(YES)} \right\}$

Entropy of our dataset $H = 1$, as expected is different from zero. Entropy for class YES, $H_{YES} = 0$ and entropy for class NO, $H_{No} = 0$, as expected, we see that our entropy is indeed 0 for $\{H_{YES}, H_{NO}\}$, indicating that we have complete knowledge of the contents of this set. If we were asked to draw one observation at random and predict the variant, we know we'd draw a pathogenic or a benign variant.

Information Gain IG about features contained in our dataset is:

Feature	IG
Chr	0.06
POS	0.06
Exon	0.01
VAF	0.25
MAF	0.36
Coverage	0.01
Protdesc	0.45

We captured some information out of each feature (variable), it seems Amino Acid Change (Protdesc) provides the greatest Information Gain.

Model selection

We compared different predictive models according to our NGS data in order to choose the best one. First, we created a set of common training controller object with the same train/test folds and model evaluation metrics (accuracy) that we will re-use. This was important to guarantee fair comparison between the different models. We needed to know which model we had to apply to our data and for that, we would use statistical methods to estimate the accuracy of the models that we created on unseen data. We also wanted more concrete estimate of accuracy of the best model on unseen data by evaluating it on actual unseen data. For that, we hold back some data that the algorithms would not get to see and we would use this data to get a second and independent idea of how accurate the best model might be. The number of instances (row) that belong to each class would give us the class distribution of our data. In order to create some models of the data and estimate their accuracy on unseen data, we began by set-up the test harness to use 10-fold cross validation[3] , then we built 5 different models to predict validation variant (y) from variant features to finally select the best model. This would split our dataset into 10 parts, train in 9 and test on 1 and release for all combinations of train-test splits[19].

k	fold 1	fold 2	fold 3	...	fold 10
1	validation	training	training	...	validation
2	training	validation	training	...	training
3	training	training	validation	...	training
⋮	⋮	⋮	⋮	...	⋮
10	training	training	training	...	validation

$$\left\{\begin{array}{cc} k=1\rightarrow & Performance_1 \\ k=2\rightarrow & Performance_2 \\ \vdots & \vdots \\ k=10\rightarrow & Performance_{10} \end{array}\right\} \Rightarrow Performance = \frac{1}{10}\sum_{i=1}^{n=10} Performance_i$$

We would also repeat the process 3 times for each algorithm with different splits of the data into 10 groups, in an effort to get a more accurate estimate[19]. We used the metric of Accuracy to evaluate models. This is a ratio of the number of correctly predicted instances in divided by the total number of instances in the dataset multiplied by 100 to have a percentage[19]. We confronted with 5 ML algorithms mixing linear, non linear and complex nonlinear methods (Linear Discriminant Analysis (LDA) ⇒ Linear, Classification and Regression Trees (CART) ⇒ Nonlinear, k-Nearest Neigbords (kNN) ⇒ Nonlinear, Support Vector Machines (SVM) ⇒ Complex nonlinear, Random Forest (RF) ⇒ Complex nonlinear). At each Algorithm run, we reset the random number seed in order to ensure that the evaluation of each algorithm was performed with exactly the same data splits.

Results

Model selection with 10-fold cross validation was done with 70% of dataset (Train set) and 30% remaining is for the validation. Data was randomly split into 2 subsets that were mutually exclusive: training and validation sets. The result of 10-fold cross validation of the 5 models is the following (Table 1 & Figure 1). According to the results presented in Figure 1 & Table 1, we can see the accuracy of each classifier and also metrics like Kappa (see Table 3 for Kappa details). The plot of the model evaluation results allow us to compare the spread and the mean accuracy of each model. There is a population of accuracy measure for each algorithm because each model was evaluated 10 times (10 fold cross validation). Based on the obtained data, it is clear that the Random Forest is the most accurate model with our NGS dataset.

Random Forest can be summarized like that after 10-fold cross validation: 81610 samples 7 predictor 2 classes: 'NO', 'YES'

No pre-processing Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 73450, 73449, 73448, 73449, 73448, 73449, ...

Resampling results across tuning parameters:

mtry	Accuracy	Kappa
2	0.9973656	0.9947311
4	0.9978557	0.995113
7	0.9975861	0.9951722

Kappa coefficient obtained with mtry=4 is 0.9957113 (Table 2) which is above 0.90, according to Kappa Cohen table (Table 3) it is Almost Perfect agreement. Kappa coefficient is frequently used in statistic to test interrater reliability[20]. The importance of rater reliability lies in the fact that it represents the extent to which the data collected in the study are correct representations of the variables measured[20]. Measurement of the extent to which data collectors (raters) assign the same score to the same variable is called interrater reliability[20].

Now we know which model to use with our data, it time to tune the model and test on validation dataset in order to make data prediction on future unseen NGS dataset.

Random Forest Algorithm Training,Tuning,Testing and Predicting

Random Forest produces a lot of small classification trees on a random fraction of the data and then votes them[4]. From this vote are deduced the order and the importance of the explanatory variables are deducted. We use all the decision trees produced to make the prediction, with a majority vote (for classification, predicted variable of factor type), or an average (for regression, predicted variable of numeric type). The majority decision then prevails. Each tree of the forest is built on a fraction ("in bag") of the data (it is the fraction which is used for training the algorithm). So for each of the individuals of the remaining fraction ("out of bag") the tree can predict a class. So the goal is to get the smallest possible out of bag (OOB) estimate error. The OOB error is a measure of the Random Forest prediction error and evaluate its performance. We analyzed it with default parameters and then we would tune the model by varying the input parameters such as ntree (number of trees), mtry (number of variables to take in consideration for each tree node) to fit model accuracy. Tuning hyperparameters (ntree, mtry) achieve two goals: increase the predictive power of the model and make it faster. Mtry refers to number of variables available for splitting at each tree node. For classification model, the default is square root of predictors variables (rounded down), in our case mtry=3 as default. Thanks to TuneRF package found for us the optimal mtry in order to get the smallest OOB. When satisfied with results classification of training dataset, we would apply the algorithm on new unseen data to see how RF prediction goes. Our results

were presented on a confusion matrix (also known as an error matrix): it's a table presenting the observed data predicted by RF algorithm. It demonstrates the performance of the RF algorithm to predict the variable y.

Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. In abstract terms, the confusion matrix is as follows:

	Positive Prediction	Negative Prediction
Actual Positive class	TP	FP
Actual Negative class	FN	TN

Where TP = True Positive; FP = False Positive; FN = False Negative; TN = True Negative. A true positive (TP) is a result where the model correctly predicts the positive class. Similarly, a true negative (TN) is a result where the model correctly predicts the negative class.

A false positive (FP) is a result where the model incorrectly predicts the positive class. And a false negative (FN) is a result where the model incorrectly predicts the negative class.

To perform our analysis, we worked on a Linux Centos 7 workstation with 16 GB Memory and with a CPU Intel Xeon E5-2603 v3 1.60GHz (12 cores).

We trained our ML model on 102011 variants which sums 3 years of NGS analysis in our platform (2017-2019). Our confusion matrix reported 71249 well classified variants and 158 badly classified.

Confusion Matrix on our dataset:

	NO	YES
NO	35550	158
YES	0	35699

Our best OOB was 0.22% error rate and was stabilized with ntree = 500 and mtry = 4 (Supplementary Figures 1a, 2a) and (Figure 2). Our strategy to tune our model was to choose the number of trees keeping the default value of mtry and to test several values by evaluating them. Plotting the model helped us choose the best ntree: before 500 trees, all OOB values oscillated too much and after 500 it was stabilized. Using more than 500 trees generates more machine time consuming, that's why we fixed the value of ntree = 500. (Supplementary Figure 1a). The data demonstrates that the best OOB was clearly improved with mtry=4 (Supplementary Figure 2a).

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

in our case Accuracy comes out to 99.77%. That means our onco somatic classifier is doing a good performance of identifying malignancies of mutation variants. Accuracy alone is not enough when we're working with a class-imbalanced data set, like this one, where there is a significant disparity between the number of positive and negative labels. We need to evaluate others metrics like precision and recall. Precision is defined as follows:

$$Precision = \frac{TP}{TP+FP}$$

Precision attempts to answer what is the proportion of positive identifications was actually correct. In our dataset, $\frac{35550}{35550+158} = 0.99$. Our model has a precision of 0.99, in other words when it predicts a variant is pathogenic, it is correct 99% of the time.

Recall (also known as Sensivity[14]) attempts to answer what is the proportion of actual positives was identified correctly. Mathematically, recall is defined as follows:

$$Recall = \frac{TP}{TP+FN}$$

As FN=0, $Recall = \frac{TP}{TP} = 1$, our model has a recall of 1, which means: it correctly identifies 100% of all pathogenic tumors.

Specificity measures the proportion of true negatives. It is the proportion of those who truly do not have the condition (unaffected) who are correctly identified as not having the condition, and it is defined as follows:

$$Specificity = \frac{TN}{TP+FP}$$

Specificity=0.99, means when a variant is not pathogenic (benign) it is correct 99% of the time. In other word, Specificity is the opposite of Precision.

Classification: ROC Curve and AUC

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP+FN}$$

False Positive Rate (FPR) is defined as follows:

$$FPR = \frac{FP}{FP+TN}$$

An ROC curve plots TPR versus FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives[21]. AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). AUC provides an aggregate measure of performance across all possible classification thresholds[21]. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.

If we look at the ROC for our model (Figure 3), we can see this is the best possible ROC curve, as it ranks all positives above all negatives. It has an AUC of 0.99 for classes YES and NO label.

We need now to evaluate the optimal model classifier on the independent Validation data set, which represents the 30 % remaining. It is composed of 30603 SNV variants (15215 are SNPs and 15389 are pathogenic onco somatic variants). Applying the model on it, give us:

	NO	YES
NO	15215	0
YES	80	15309

100% of 15215 TP SNP were predicted real. 15309 out of 15389 (99.5%) were labeled as pathogenic. More importantly, none of the TP SNP were misclassified as pathogenic while only 0.52% were labeled as benign and require further manual investigation to correct it. These 80 on 15309 pathogenic variants badly would not have an impact on clinical outcome as the Biologist inspects manually all variants and the proportions of 0.5% error is very few in comparison of the efficiency of the classifier in predicting the correct class.

Feature Importance

In order to determine the importance of the features and their contribution in our optimal model, we will use the Mean Decrease Gini index. The higher this indicator, the more important the variable is in the model (it measures the decrease in the Gini index if we no longer include this feature in the model).

There is indeed a correlation between these 3 parameters in onco-somatic, because a variant comprising a MAF and a silent amino acid change with an allelic frequency close to 90-100% has a high chance of being a SNP (Supplementary Figure c). A correlation does exist between Amino Acid Change (protdesc), MAF and Variant Allele Frequency (Freq) (Supplementary Figure d).

Amino Acid Change (protdesc) was recognized as the most important feature, followed by Variant Allele Frequency (Freq) and Minor Allele Frequency (MAF) (Supplementary Figure 3a). The distribution of minimal depth (Supplementary Figure e) means what is the first time this variable (feature) is used to split the tree. The more important variables have lower min depth values. The splits that cause the larger increases in purity happen early and so the important variables are split on early.

Random Forest routine implementation and impact on our workflow

Since March 2020, we used RF in NGS analysis routine at each variant import into our MySQL database. Architecture of our Information System was developed in Hypertext PreProcessor (PHP) so it has to deal with R script in order to call Machine Learning at each run (Figure 4).

Our NGS database of the 2 first months of the year represented more than 2000 variants. Our RF model misclassified 20 variants (which represents an error rate < 1%). The errors were nomenclature problems or false positives. Since March 2020, we found 2 errors about the same MET mutation in splice site.

Testing Neural Network on our dataset

We tried several Machine Learning algorithms, but it can be interesting to see if Neural Networks can achieve a good recognition pattern on our dataset. We will use NeuralNet R package version 1.44.2.

Our aim is to connect the 7 input features (chr, POS, Exon, Freq, MAF, Coverage and Protdesc) to the correct output class (YES, NO) by using an artificial neural network fully connected. We will use 10-cross-validation (10-cv) method on our ANN.

Neural networks, commonly known as artificial neural networks (ANN) are simple imitations of the functions of a neuron in the human brain to solve machine learning problems. As we did with Machine Learning, in order for our " system " to be able to compare these variants, we must give to our neuron network a multitude of pathogenic variants and benign variants by indicating to it what each variant represents. We are in the learning phase. ANN will analyze each of the data and draw its own result. The goal at the end, once the learning phase is finished, if we give a variant, the system will able to predict if pathogenic or not with a certain probability. Neural Network fully connected, which means all nodes (neurons) are connected between each other. it is a set of neurons organized in layers and interconnected by what we called " weights ". Each weight is a value which will used to spread information and they are the base of our networks. Thanks to them, all the learning process can be done. The first layer is called Input layer and the last layer is called Output layer. In our study case, we have 7 inputs and 2 outputs. All layers between the input layer and the output layer are called hidden layer. The amount of hidden layers and nodes are very inconstant. Potentially from zero to infinity. There are only empiric rules to define how many hidden layers and node do we need. As every problem is different from each other and each network approximates a system differently there are no precise rules or even a mathematical formula that allows us to determine which numbers are appropriate for that specific problem[13] . Despite this fact, we can use an optimum choice of the number of neurons:

- Small number of neurons will lead to high error for your system, as the predictive factors might be too complex for a small number of neurons to capture[13]
- Large number of neurons will overfit to your training data and not generalize well[13]
- The number of neurons in each hidden layer should be somewhere between the size of the input and the output layer, potentially the mean[13]
- The number of neurons in each hidden layer shouldn't exceed twice the number of input neurons, as you are probably grossly overfit at this point[13]

There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following: The number of hidden neurons should be between the size of the input layer and the size of the output layer. The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer. The number of hidden neurons should be less than twice the size of the input layer[12].

The first step is to give understandable information to our input layer: numbers. The information will propagate layer by layer, each neurons will react with a numerical value based on its incoming connections, according to their weights. This will define their activation rates.

This neurons receives several entrance from input neurons $[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$ values and produces one output value. Each of these entrances are balanced by one weight w_i. Inside this neuron there is an activation function σ.

It is mathematical function which takes input a numbers and output a result. All we need to know about the activation function is that it will help us to make our input values fit in a small interval. Most of the time between 0 and 1 or between -1 and 1.

The most frequent used functions are the hyperbolic tangent (tanh), sigmoid function, the linear adjustment (relu), softmax. For our ANN, we will use sigmoid function, which is defined as:

$$\Psi(x) = \frac{1}{1+e^{-x}}$$

Please see section Supplementary data - Working principle of ANN for further information.

As input and output are already defined, the question of how many hidden layers and nodes do we need?

To answer this question, we will use tune grid method with 10 cross validation. We will define 3 hidden layers $L_i = (L_1, L_2, L_3)$ each Layers contains a grid of values of nodes: $L_1 = [1:5]$, $L_2 = [0:5]$, $L_3 = [0:5]$. It took 22 hours to perform the tune grid search against 30 min with Random Forest to have a result. After performing the 10 cross validation tuning grid layer optimization, system found the best accuracy is giving by $L_1 = 5$,$L_2 = 0$ and $L_3 = 5$ with an accuracy of 97.94% ($\simeq$98%) and a Kappa value of 0.9588 ($\simeq$0.96) (Figure 5). Results below:

Accuracy : 0.9794

95% CI : (0.9783, 0.9804)

No Information Rate : 0.5001
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.9588
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.9843
Specificity : 0.9745
Pos Pred Value : 0.9747
Neg Pred Value : 0.9842
Prevalence : 0.5001
Detection Rate : 0.4922
Detection Prevalence : 0.5050
Balanced Accuracy : 0.9794 '
Positive' Class : NO

Confusion Matrix on Test set (Supplementary Figure 5a). Looking at the validation set results, the ANN performed good performance on unseen data with an accuracy of 97.91% and a Kappa of 96% (Supplementary Figure 6a). Results performance on validation test:

Accuracy : 0.9791
95% CI : (0.9774, 0.9807)
No Information Rate : 0.5002
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.9582 Mcnemar's
Test P-Value : 0.0007797
Sensitivity : 0.9819
Specificity : 0.9763
Pos Pred Value : 0.9764
Neg Pred Value : 0.9818
Prevalence : 0.4998
Detection Rate : 0.4907
Detection Prevalence : 0.5026
Balanced Accuracy : 0.9791
'Positive' Class : NO

AUC for YES class label is 0.994532 and for NO label is 0.9945321. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. As AUC is close to 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. ROC curve shows a good classifier as we can on the (Supplementary Figure 7a).

Our Neural Network variant classifier look likes (Figure 6), 7 inputs + 2 hidden layers of 5 nodes each and two outputs.

Discussion

Our aim was to use AI to interpret NGS onco-somatic analysis relying on 7 parameters: chromosome, position, exon, variant allele frequency, minor allele frequency, coverage, protein description. We developed an accurate and efficient method based on Random Forest Machine Learning which classifies a set of variants with high precision from NGS data. Our implemented model gives decisions and probability for each variant call and lets the hand to the Biologist of accepting or rejecting variants (Supplementary Table a). The model was developed in R language (R version 3.6.0, RandomForest version 4.6-14, NeuralNet version 1.44.2) which are Machine Learning open source modules and hence more users can tune this model on their own data by using our training database.

ML and more particularly RF model has already been used in several publications as a tool for risk evaluation, quality control or predictions based on sequencing data[2,4,9]. Furthermore, the applications we found using RF rely on the analysis of VCF (Variant Caller) files such as Smurf which predicts a consensus set of somatic mutation calls or Octopus which uses BAM files as training dataset to classify variant calls[9,10].

Our tool classifies onco-somatic variants when already annotated (See supplementary data Figure a). It is very easy to implement in Information System and user friendly for "non computer" science person. Moreover, our tool is not intended to replace biologists but is rather a decision help facing the multitude of somatic variants present at each NGS run analysis. To our knowledge, it has never been published in order to interpret NGS analysis in routine. Our RF model shows excellent results with an error rate < 0.5% since we use it in routine. Errors are: nomenclature errors and false positive absent from our training database. Nomenclature errors are difficult to correct because Ion reporter annotates variants automatically, so our error rate

could never be zero and we have to stay vigilant to spot these nomenclature errors. But for some of them, we can manually change it if there is a nucleotide's error, for example, after that our RF model gets it right. Nevertheless, we can't cancel read errors from the sequencing. We easily improve about false positive errors: at every new false positive mutation, we increment our training database. So, our RF model can improve as the NGS runs go. ML is the oldest and simplest technology. It is based on an algorithm which adapts the system itself based on feedback from humans. The implementation of this technology implies the existence of organized data. The system is then fed with structured and categorized data allowing it to understand how to classify new similar data. Based on this classification, the system then performs the programmed actions. After a first phase of use, the algorithm is optimized based on the developer's feedback, which informs the system of incorrect classifications and indicates the correct categories. ML works from a controllable database, which means at each moment we modifytraining database to correct prediction errors. The system must have much more entries to give reliable results. We also have an error rate < 0.5% for our Glioma panel (AmpliSeq Gliome custom panel : AKT1, ATRX, BRAF, CDKN2A, CIC, EGFR, FGFR1, H3F3A, HIST3B, IDH1, IDH2, NOTCH1, PIK3CA, PIK3R1, PTEN, PTPN11, TERT, TP53) NGS analysis. We choose RF as first model for several reasons. First, it's easier to set up, since RF combines regression and classification, which saves time on data preparation. Second, there is less chance of " overfitting " (these cases where the model is perfectly adapted to data, but does not generalize very well). Third, RF does more precise predictions but does not need so much pruning. Indeed, we have a multitude of random trees using random features and thus the individual trees are strong but not so correlated with each other. Pruning is a method used to prevent overfitting of decision trees, by trimming certain levels of the tree. Implementing RF in routine is very easy and fast. Processing time for one chip of 32 samples takes less than one minute to predict each variant. RF has already been tested and used by other authors for NGS analysis. It seems to be a good reproducible model for this type of interpretation[2,5,6,11]. Moreover, Random forest algorithm is a machine learning algorithm that runs efficiently on large datasets with high accuracy[15]. Neural Networks shows his ability to classify mutation variant but tuning a neural network costs time when another approach might be better and save you a lot of time. Consider using a different ML algorithm to ANNs - decision tree based on approaches such as Random Forest can be faster and more effective than deep learning on many problems. However, they also have capacity to learn subtle differences from large amounts of data where simpler algorithms may reach a limit. Neural networks have provided good accuracy in the prediction of variant mutations. We compared performance of Neural Network and Random Forest on unseen data, and RF was mostly of time a little bit over in accurary in comparison with Neural Network. However, ANN shows his ability to adapt itself to new data where RF failed. For example, the two MET mutations in splice site missed by RF were well predicted by Neural Networks as pathogenic (Supplementary Table a). After modifying our database, our RF model does a correct classification. It is always a good idea to make running in parallel our ANN and compare results with RF. A study in the optimization of the number of hidden layers and neurons which composed them can be a good approach in a future study, in particular for the predictions of the pathogenicity of the BRCA1 / BRCA2 genes. They present more parameters in the interpretations of pathogenicity. Despite their good prediction of ANN, we rather decided to implement the RF in routine because the classifier showed us very good values of predictions. The final trained model was able to achieve an accuracy of 99,77% and a ROC-AUC of 0.99, we are close to a perfect classifier.

Conclusion

We presented a Machine Learning method to classify Onco Somatic variants detected by NGS on Lung, Colorectal, Melanoma and Glioma tumors. RF is a great AI algorithm for classification problems to produce a predictive model. Easy and fast to implement in Information System and it does not need any powerful machine to perform. Our Model was optimized to use the lowest amount computer resource, making it faster even on average computers. The only disadvantage is high number of trees might take the computation process much slower. Since we used it in routine, our RF model shows good results with ANN in parallel. It could easily be implemented in other molecular biology laboratories and become a true help in NGS onco-somatic interpretation in routine. AI is taking an increasingly important place in molecular biomedical analysis and could be very helpful on processing a large amount medical data.

References

1. Miller DD, Brown EW. Artificial Intelligence in Medical Practice: The Question to the Answer? Am J Med. 2018;131(2):129-33
2. Li J, Jew B, Zhan L, Hwang S, Coppola G, Freimer NB, et al. ForestQC: Quality control on genetic variants from next-generation sequencing data using random forest. PLoS Comput Biol
3. Oneto L. Model Selection and Error Estimation in a Nutshell [Internet]. Cham: Springer International Publishing; 2020. (Modeling and Optimization in Science and Technologies; vol. 15)
4. Ram M, Najafi A, Shakeri MT. Classification and Biomarker Genes Selection for Cancer Gene Expression Data Using Random Forest. 2017;9

5. Wood DE, White JR, Georgiadis A, Van Emburgh B, Parpart-Li S, Mitchell J, et al. A machine learning approach for somatic mutation discovery. Sci Transl Med. 5 sept 2018;10(457)
6. Janßen R, Zabel J, von Lukas U, Labrenz M. An artificial neural network and Random Forest identify glyphosate-impacted brackish communities based on 16S rRNA amplicon MiSeq read counts. Marine Pollution Bulletin. 1 déc 2019;149:110530
7. López-Reig R, Fernández-Serra A, Romero I, Zorrero C, Illueca C, García-Casado Z, et al. Prognostic classification of endometrial cancer using a molecular approach based on a twelve-gene NGS panel. Sci Rep. déc 2019;9(1):18093
8. Marceddu et al. Analysis of machine learning algorithms as integrative tools for validation of next generation sequencing data. European Review. 2019
9. Njage PMK, Henri C, Leekitcharoenphon P, Mistou M-Y, Hendriksen RS, Hald T. Machine Learning Methods as a Tool for Predicting Risk of Illness Applying Next-Generation Sequencing Data. Risk Analysis. 2019
10. Park H, Chun S-M, Shim J, Oh J-H, Cho EJ, Hwang HS, et al. Detection of chromosome structural variation by targeted next-generation sequencing and a deep learning application. Sci Rep. déc 2019
11. Wang H-Y, Chang S-C, Lin W-Y, Chen C-H, Chiang S-H, Huang K-Y, et al. Machine Learning-Based Method for Obesity Risk Evaluation Using Single-Nucleotide Polymorphisms Derived from Next-Generation Sequencing. J Comput Biol. 2018
12. Introduction to Neural Networks for Java (second edition) by Jeff Heaton – 1st October 2008
13. Neural Networks with R - Smart models using CNN, RNN, deep learning, and artificial intelligence principles, Giuseppe Ciaburro Balaji Venkateswaran - 2017 Packt Publishing ISBN 978-1-78839-787-2
14. Youden, W. J. Index for rating diagnostic tests. Cancer 3, 32–35 (1950)
15. Breiman, L. Random forests. Machine learning 45, 5–32 (2001)
16. Maheshkar, Saurav. "Brief Introduction to Decision Trees." DEV Community, DEV Community, 9 Nov. 2020, dev.to/sauravmaheshkar/brief-introduction-to-decision-trees-15dg.
17. Malviya, Nikita. "DECISION TREE." Medium, Analytics Vidhya, 8 Sept. 2020, medium.com/analytics-vidhya/decision-tree-2855f7e198f0.
18. Sujan, Nasir Islam. "What Is Entropy and Why Information Gain Matter in Decision Trees?" Medium, Coinmonks, 5 Feb. 2021, medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01
19. Brownlee, Jason. "Your First Machine Learning Project in R Step-By-Step." Machine Learning Mastery, 7 Oct. 2019, machinelearningmastery.com/machine-learning-in-r-step-by-step/.
20. McHugh, M. L. (2012). Interrater reliability: the kappa statistic. Biochemia medica, 22(3), 276-282.
21. Classification: ROC Curve and AUC | Machine Learning Crash Course. developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

Accuracy

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	NA's
lda	0.8431757	0.8468371	0.8521627	0.8508151	0.8545082	0.8562500	0
cart	0.9169219	0.9226504	0.9408207	0.9406823	0.9603283	0.9638525	0
knn	0.9578483	0.9581253	0.9607279	0.9606788	0.9630855	0.9637344	0
svm	0.9693665	0.97237720	0.9731021	0.9727975	0.9734996	0.9745129	0
rf	0.9974268	0.9974577	0.9977943	0.9978557	0.9980394	0.9986525	0

Table 1. Accuracy of ML algorithms after 10-fold cross validation

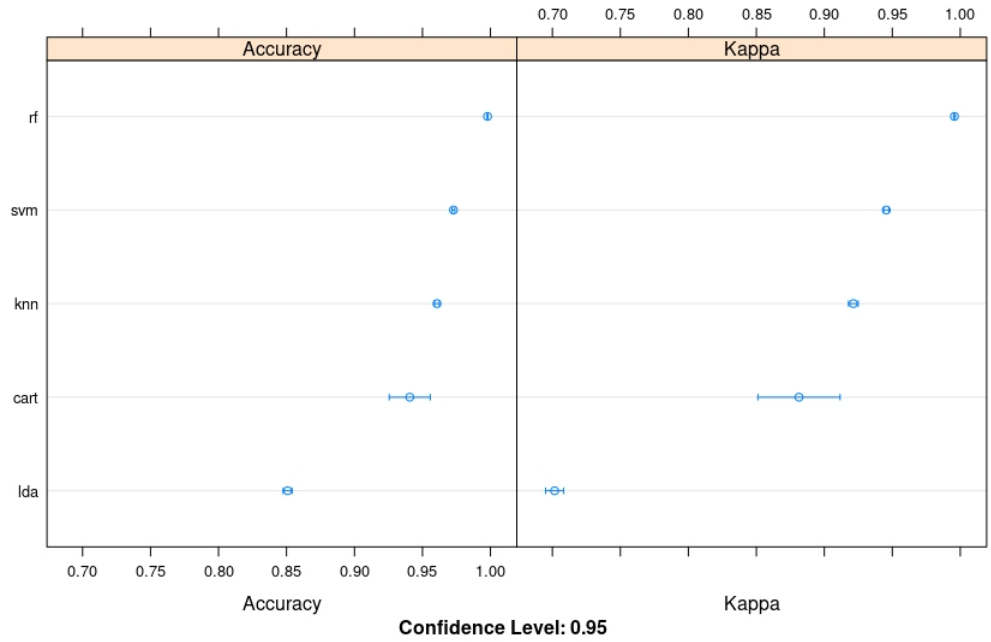

Figure 1. ML algorithms comparison : Linear Discriminant Analysis (LDA) ⇒ Linear, Classification and Regression Trees (CART) ⇒ Nonlinear, k-Nearest Neigbords (kNN) ⇒ Nonlinear, Support Vector Machines (SVM) ⇒ Complex nonlinear, Random Forest (RF) ⇒ Complex nonlinear. Number of resamples:10

Kappa results:

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	NA's
lda	0.6863514	0.6936705	0.7043258	0.7016283	0.7090152	0.7125000	0
cart	0.8338433	0.8453000	0.8816417	0.8813644	0.9206570	0.9277045	0
knn	0.9156957	0.9162500	0.9214557	0.9213574	0.9261705	0.9274688	0
svm	0.9387329	0.9447439	0.9462042	0.9455949	0.9469992	0.9490259	0
rf	0.9948536	0.9949153	0.9955885	0.9957113	0.9960788	0.9973043	0

Table 2. Kappa metric results after 10-fold cross validation

Value of Kappa	Level of Agreement	% of Data that are Reliable
0-.20	None	0-4%
.21-.39	Minimal	4-15%
0.40-.59	Weak	15-35%
.80-.90	Strong	64-81%
Above .90	Almost Perfect	82-100%

Table 3. Interpretation of Cohen's Kappa Values

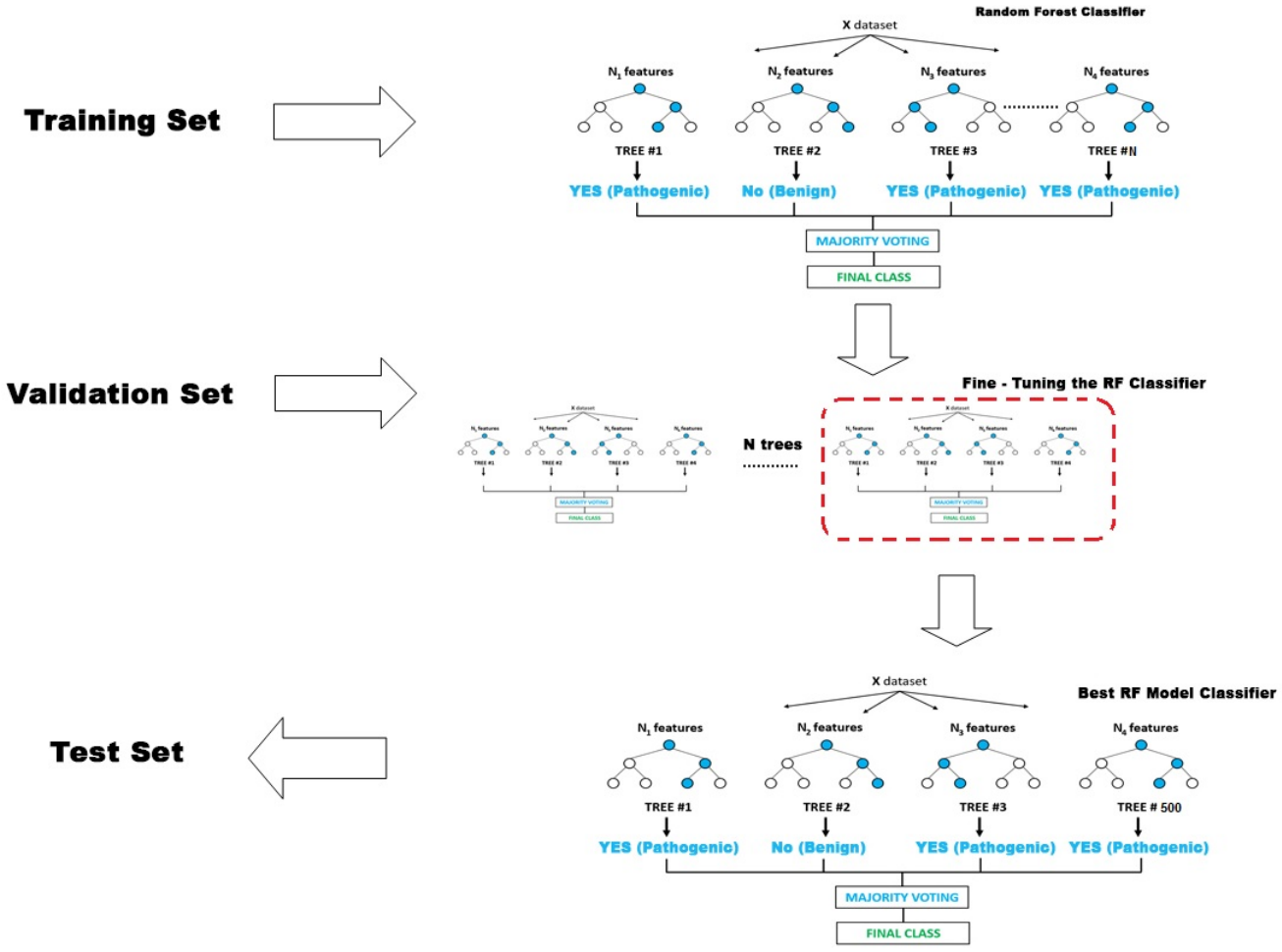

Figure 2. Computational classifier development. Model was trained using the training set. Different sets of parameters were evaluated to identify the best performing model using a validation set, before using the test set in NGS Routine

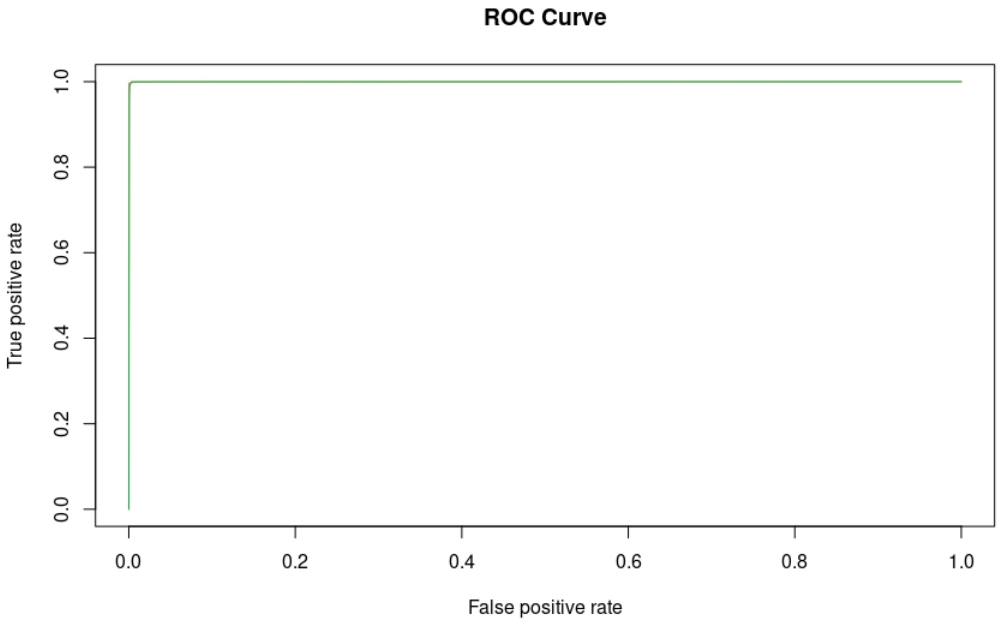

Figure 3. ROC Curve for Validation set

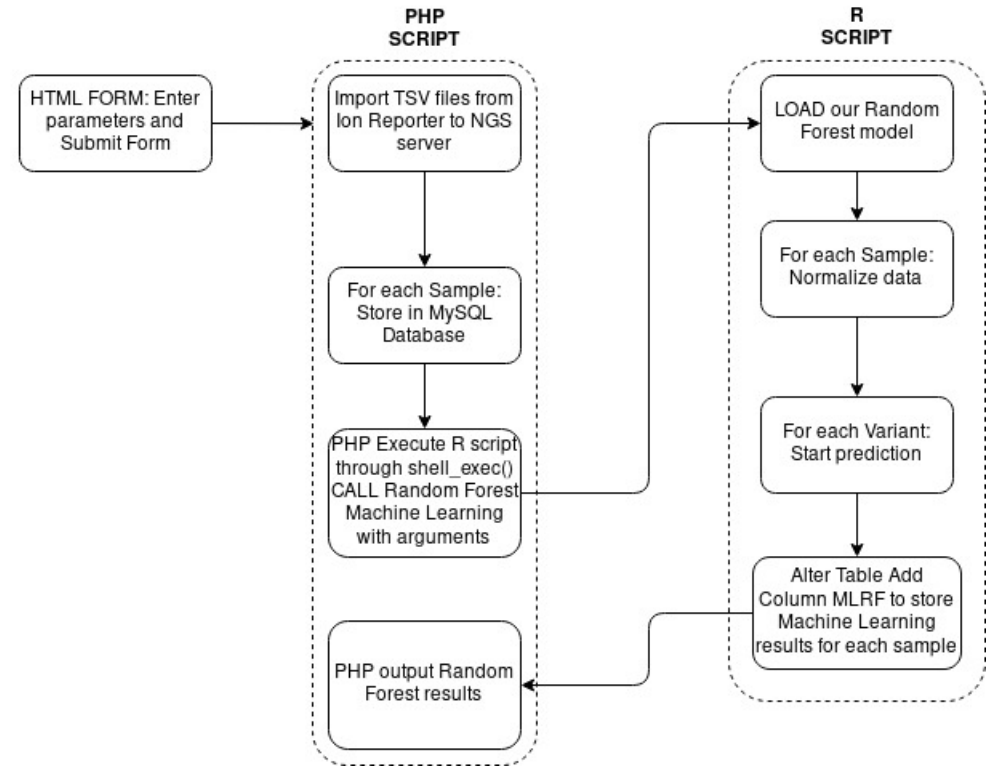

Figure 4. Machine Learning data processing. It shows how data are imported and analyzed by Random Forest Machine Learning. This architecture was implemented as well in our laboratory for each NGS run.

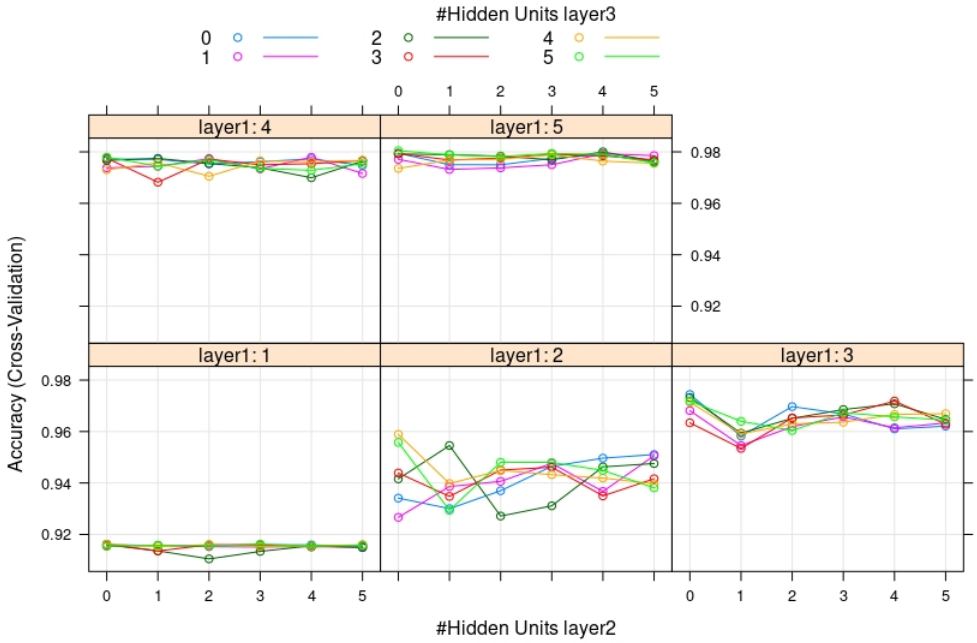

Figure 5. Results of Accuracy (Cross-Validation) in function of Hidden Units Layers parameters.

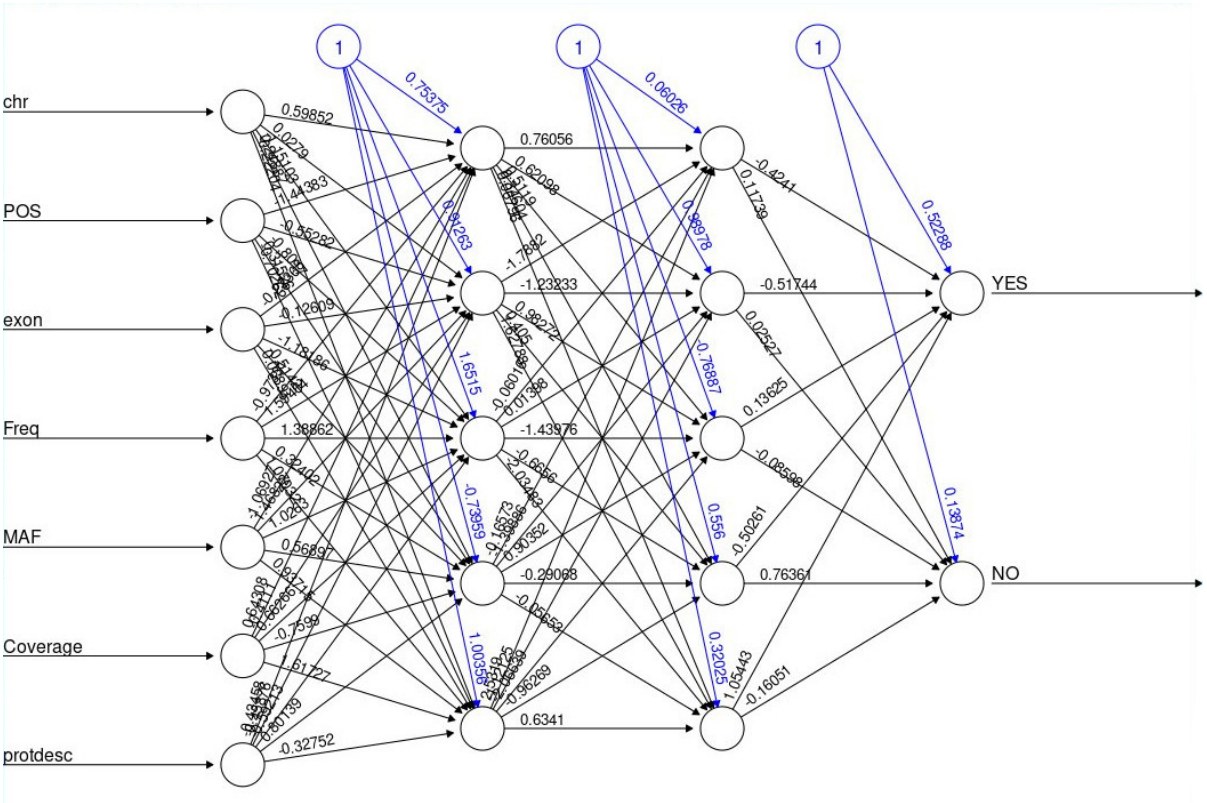

Figure 6. Our Artificial Neural Network for Onco-Somatic Variant.

Figures

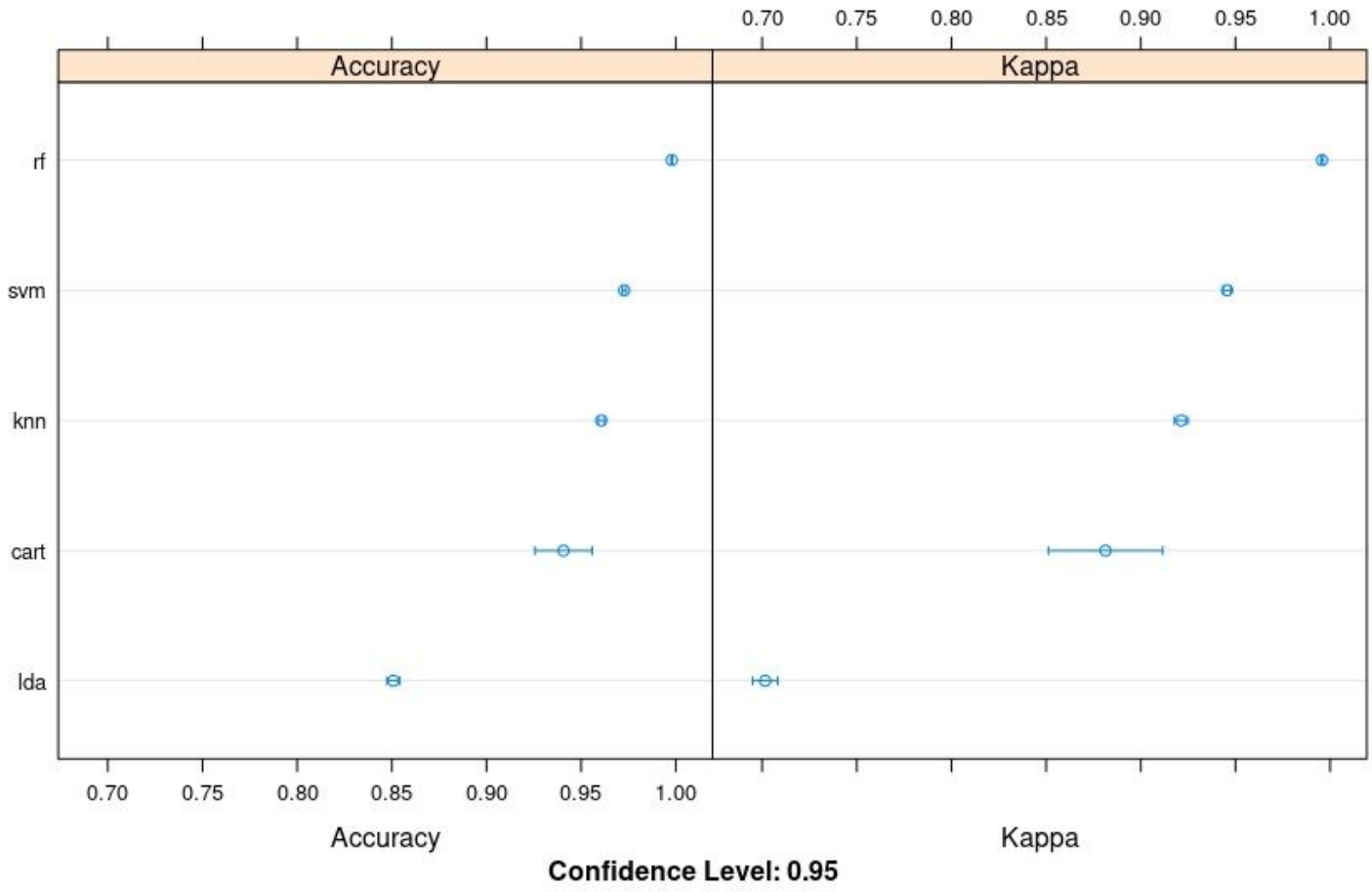

Figure 1

ML algorithms comparison : Linear Discriminant Analysis (LDA)) Linear, Classification and Regression Trees (CART)) Nonlinear, k-Nearest Neigbords (kNN)) Nonlinear, Support Vector Machines (SVM))Complex nonlinear, Random Forest (RF))Complex nonlinear. Number of resamples:10

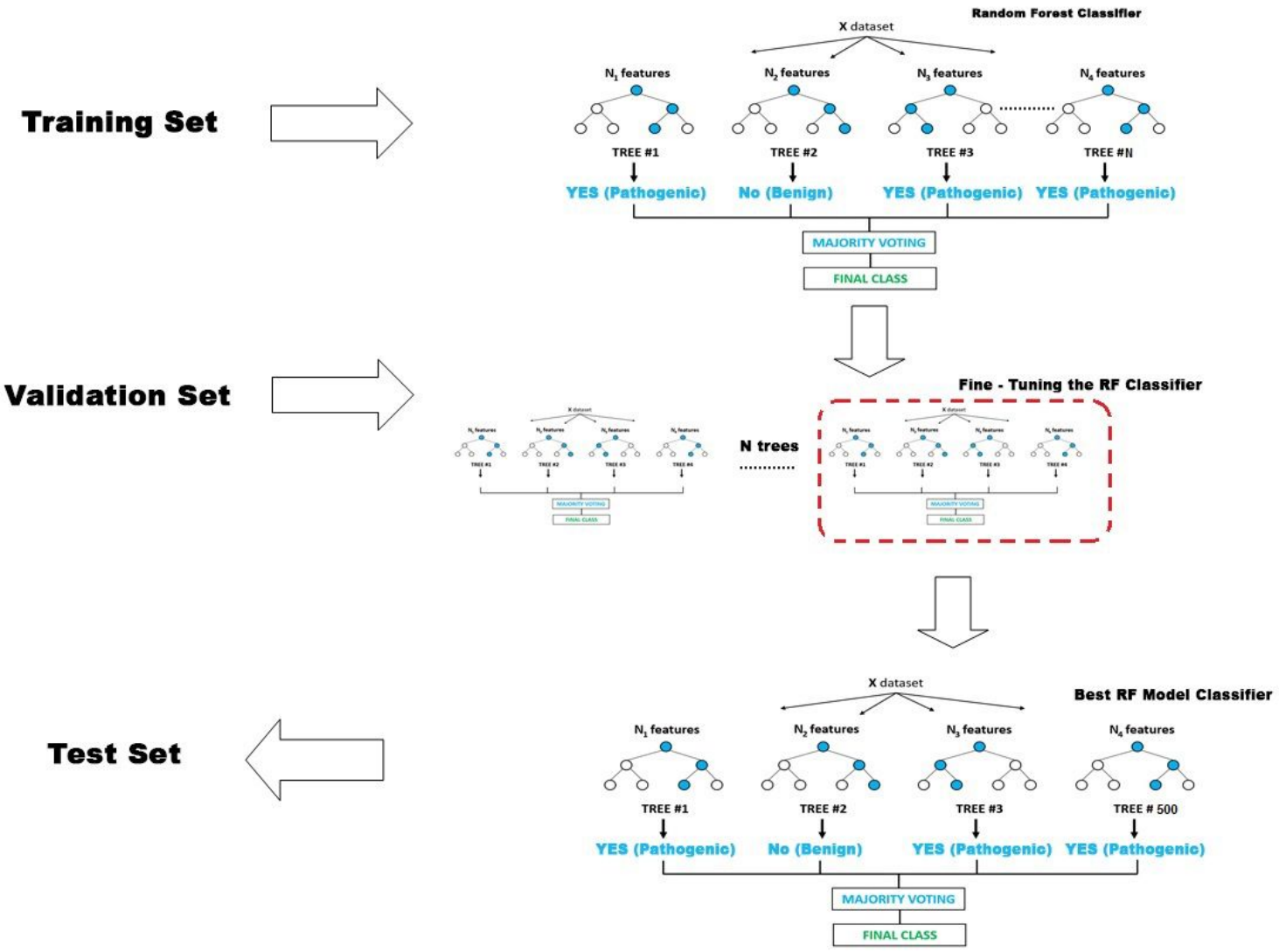

Figure 2

Computational classifier development. Model was trained using the training set. Different sets of parameters were evaluated to identify the best performing model using a validation set, before using the test set in NGS Routine

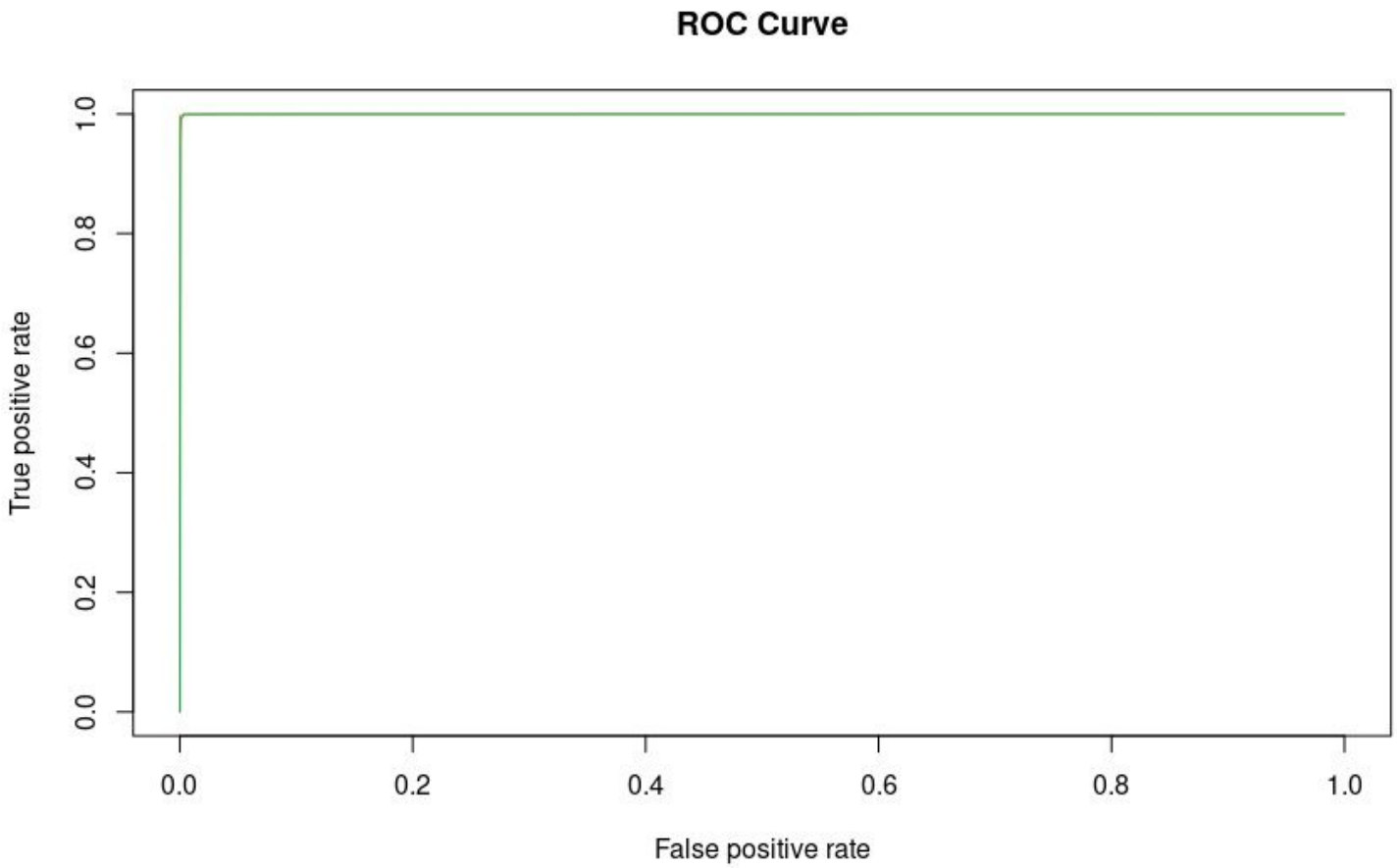

Figure 3

ROC Curve for Validation set

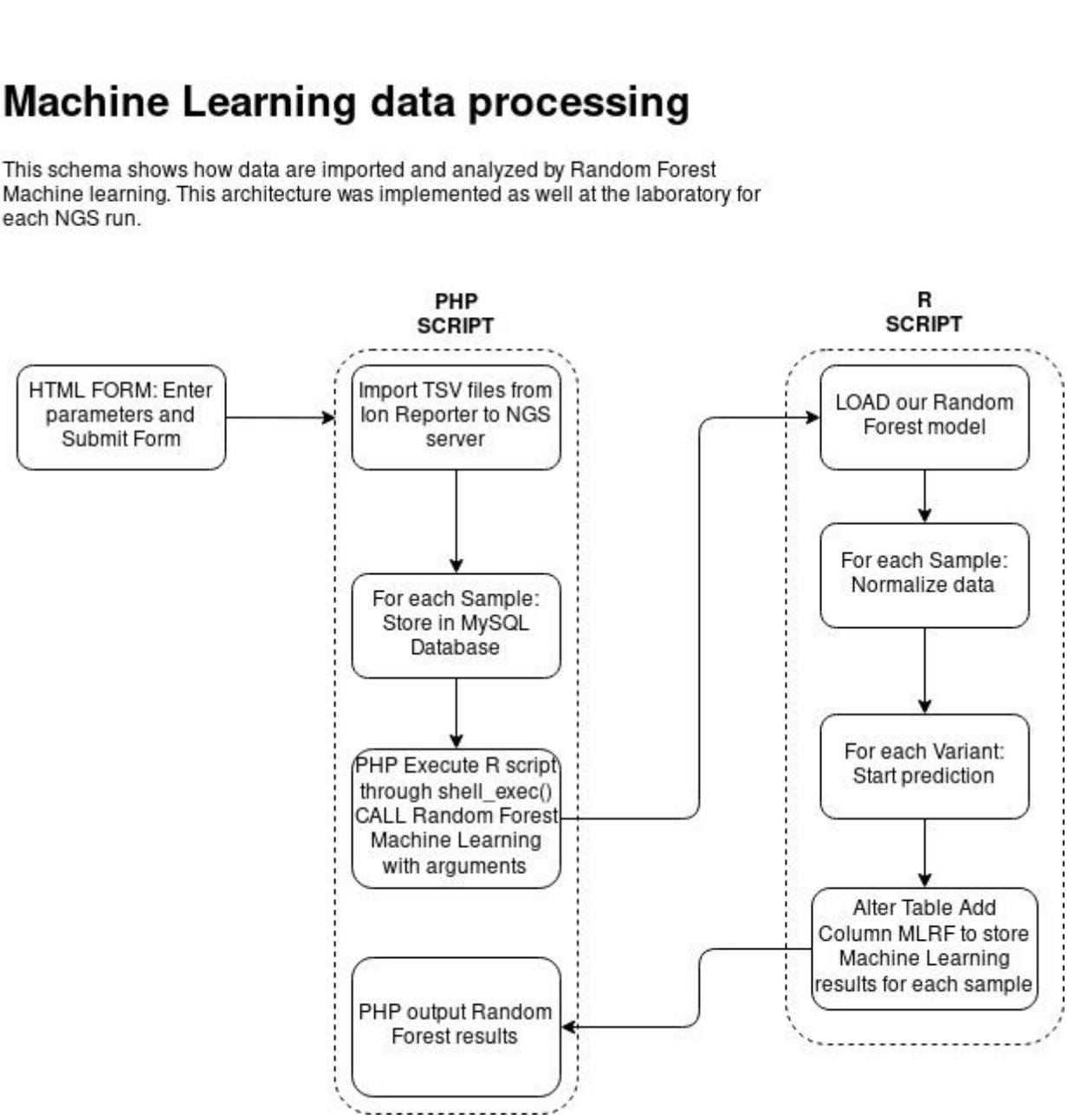

Figure 4

Machine Learning data processing. It shows how data are imported and analyzed by Random Forest Machine Learning. This architecture was implemented as well in our laboratory for each NGS run.

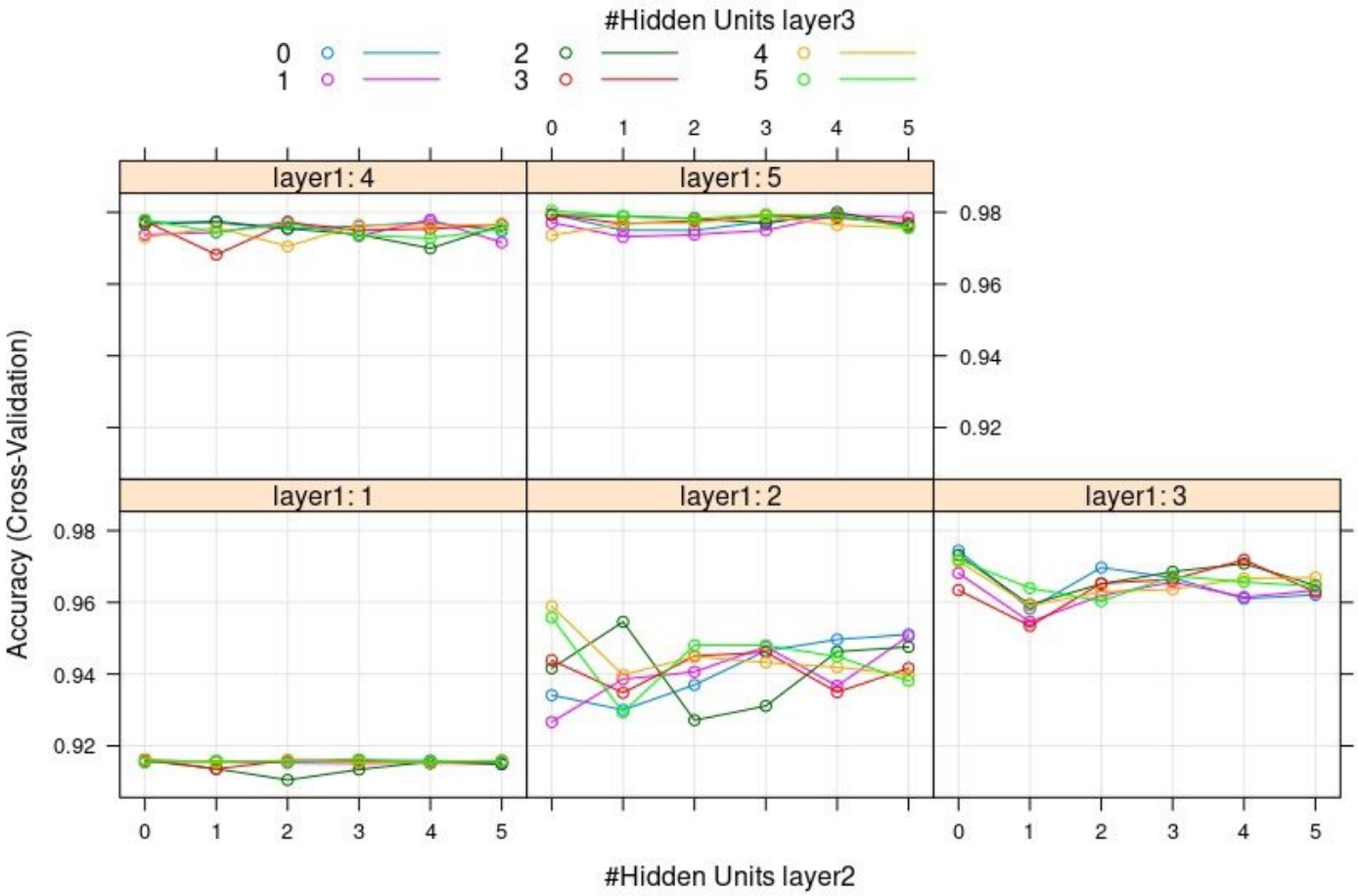

Figure 5

Results of Accuracy (Cross-Validation) in function of Hidden Units Layers parameters.

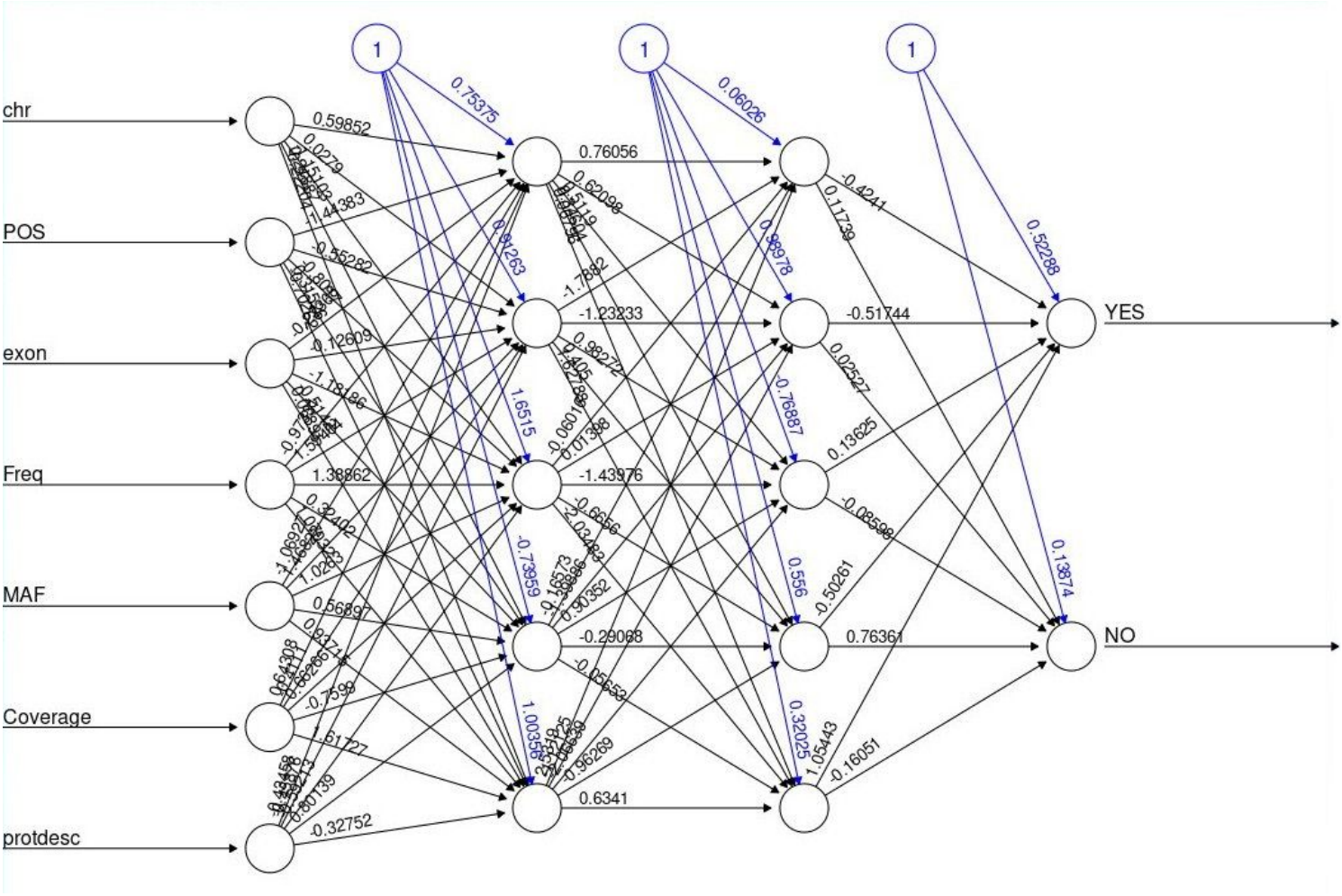

Figure 6

Our Artificial Neural Network for Onco-Somatic Variant.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SuppData.pdf