This tutorial describes the output produced by the Classification module and gives general guidelines on how to interpret the results therein. As a recap, the Classification module performs a binary classification on the count files input as control or treatment by applying a Random Forest classifier. The output includes different types of plots: data exploration, classifier performance and feature importance plots, along with an interactive web report of known and predicted micro RNAs (miRNAs) that allows you to subsequently perform functional analyses. The analysis example throughout this tutorial is the classification output of the Psoriasis demo dataset (Joyce et al., 2011).

This tutorial aims at helping the user interpret the results of the Classification module. Further information on how to submit data to the Oasis Classification module can be found on the classification tutorial page.

Part 1: Overview

The first part of the output contains an overview table of the job carried out. First of all, the parameters \texttt{mtry} and \texttt{n_tree} used to train the random forest classifier are shown (where the former is by default automatically calculated). The next two numbers are performance measures for area under the ROC curve (A.U.C) and out-of-bag (O.O.B) error. The number of samples or instances in each group (control and treatment) used to train the random forest are also included. Finally, the table contains the number of sRNAs initially input into the classifier (before filtering) and the number of sRNAs effectively used to train the random forest (after filtering) (Fig. 1).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{overview_table.png}
\caption{Overview of the Classification output module}
\end{figure}
Some clarifications on the overview table:

1. **Random forest parameters**: as it is presented later in this document, one way to see if the parameterisation of the forest is correct is to check the OOB error plot (Fig. 8). If the error lines converge, it means the number of trees in the forest was correctly set.

2. **Performance measures**: the A.U.C. is an important measure of classifier importance, because it encapsulates in a single number the ability of the classifier to separate the data. While the A.U.C. measures classifier importance, the O.O.B. error is an estimate of the expected classification error you will get if you classify a completely new sample.

**Part 2: Data exploration plots**

In addition to the overview table the Classification module returns principle component analysis (PCA) plot (Fig. 2) and Outlier plot (Fig. 3) for the initial exploratory analysis of your data.

PCA plots are useful to understand whether control and treatment groups cluster separately, but are also good for understanding if other noticeable features in the data are present. The psoriasis data, for example, clusters into two biological conditions, where control samples appear in red and treatment samples appear in blue (Fig. 2). The distinction between the samples based on their group is poor, which can have many reasons, most notably technical problems or problems with the biological specimen.

![Fig. 2: PCA plot](image)
Technically, PCA plots show the two principal components of the samples used to train the classifier. The x- and y-axes indicate the level of variance "explained" by each principal component.

A complementary form of understanding how data is organized is given by the Outlier plot (Fig. 3). While the PCA shows a multitude of misclassified samples, here samples 14 and 20 clearly show suspiciously high scores. Technically, the Outlier plot shows the modified Z-scores on the proximities reported by the random forest. In random forests, proximities are a measure of how similar samples are to each other (Breiman, 2001), so they can be used as input for methods designed to detect outliers. The modified Z-score has been proposed as a measure for outlier detection and is a function of the sample's median absolute deviation (Iglewicz & Hoaglin, 1993).

Please note that for ease of visualization, the outlier plot in Oasis shows your samples with colors associated to the class they belong to: red for controls and blue for treatments. However, do not confuse this plot with a depiction of how your samples separate.

Some notes about outlier detection:

1. **How to detect an outlier**: as suggested by (Breiman, 2001), modified Z-score values greater than 10 could be potential outliers. In the example of
Fig. 3, sample 20 is a strong outlier candidate whereas sample 14 is a borderline case.

Part 3: Performance plots

While the PCA and Outlier plots should be used to assess if your samples cluster into their biological conditions, ROC (receiver operating characteristic) (Fig. 4) and precision-recall plots (Fig. 5) inform you about the actual classification performance.

A ROC curve shows the true-positive rate as a function of the false-positive rate. In a nutshell, the TPR (true-positive rate) is the rate of predicted true positive (treatment) samples over all positive (treatment) samples. The FPR (false-positive rate) is the rate of predicted false positives (control) samples over all negative (control) samples. An ideal classifier would have an ROC curve covering the whole "ROC space" and reach the upper-left corner of the space, corresponding to the point (0,1). A classifier that reaches such point would correctly predict all positive observations without incorrectly labeling any negative observation. In simple terms, the closer the red line to the upper left corner of the plot, the better is your classifier performance. The closer the red line to a diagonal line from the bottom left to the upper right corner of the plot, the worse is your classification performance (a diagonal line would signify random classification, i.e. the samples are randomly assigned as treatment or control, and not based on some sRNA measurements).
The ROC space is normalized and covers an area of 100%, so one very simplified way to represent classifier performance is to use the area under the ROC, or A.U.C., reported in the Overview section and on top of the ROC plot. As a rule of thumb, A.U.C. values above 0.9 signify good classification performance.

Possibly a more intuitive way to graphically assess your classification performance is the precision-recall plot (Fig. 5). Recall is equivalent to the TPR (see above), indicating how many of all treatment samples classify correctly. The precision is the rate of true positive (treatment) samples over the sum of all classified samples (true-positive and false-positive). Given a good classification, a precision-recall plot will pair high precision with high recall.

More information on ROC plots can be found at (Fawcett, 2003).

**Part 4: Feature importance plots**

Biomarker identification is about selecting the most important set of sRNAs that might be useful to make a classification. Oasis uses the gini index as a measure of variable importance with higher indexes indicating higher importance of the sRNA. Oasis’ variable importance plot reports the gini indices for the 10 most important sRNAs found by the random forest classifier, in order of decreasing importance (Fig. 6).
The variable importance can be interpreted as follows:

1. **Gini indices**: as a ‘rule of thumb’, a break in the variable importance plot will indicate the set of most importance features. For example, in Fig. 6, the two most important breaks occur between hsa-miR-21-3p and hsa-miR-203a-5p, and between hsa-miR-574-3p and hsa-miR-21-5p.

A valuable plot for the identification of the ‘optimal’ biomarker set of sRNAs is the feature selection plot (Fig. 7). It reports the cross-validated prediction error of random forest models trained by increasingly adding sRNAs, with the order of additions being determined by the sRNA gini indices. This feature selection strategy has been applied in previous studies (Ashlock & Datta, 2012; Erho et al., 2013).
Notes on feature selection:

1. It has been suggested in (Svetnik, Liaw, & Tong, 2004) that the optimal set of features is given by the point in which the curve of feature selection plot is minimized. For the psoriasis example, it seems that the top 57 sRNAs obtain optimal classification results (10% OOB error, 1:57 features).

2. This plot is obtained by applying the strategy of (Svetnik et al., 2004) and consists of computing the cross-validated prediction error obtained from training different random forest models by iteratively adding features, with the order of additions being defined by the ranking of features obtained from the random forest trained with the full feature set. The number of folds used to partition the data is set to k=10.

**Part 5: Error rate plot**

The last plot that Oasis’ Classification module provides is the error rate plot (Fig. 8). It shows the Out-of-bag (O.O.B) error of the random forest model trained with the full set of features. This plot is useful to let you know whether the parameter NTREE of the forest has been correctly set. A sufficiently large NTREE will be reflected by OOB error rates for treatments, controls, and both classes that are parallel to the x-axis for higher NTREE values (green, black and red). Parallel
error lines indicate constant classification performance for increasing number of trees.

![OOB error rate vs Number of Trees](image)

**Fig. 8: Out-of-bag error plot**

Notes on the OOB error plot:

1. If there is no convergence on any of the lines at the right hand side of the plot, then the parameter NTREE should be increased.

**Part 6: Interactive web report**

Apart from the statistical plots, the Oasis Classification module returns an interactive web table that contains detailed information on each sRNA, and allows for the functional enrichment analysis of target sRNAs. To access this information, press the option “Classification results” at the top of the HTML page (Fig. 1, red box). A new web page will open (Fig. 9), containing two functional parts: functional analysis options (Fig. 9, blue box) and a table of all classification analysis results for the sRNAs (Fig. 9, red box).
Within the table, columns “mature”, “structure”, “position” and “strand” indicate the name of the reported sRNA, the sRNA species it belongs to (for elaboration on sRNA species, see the sRNA Output Tutorial) and its position and strand, respectively. The column “feature importance” is based on the gini index indicating the importance of each sRNA to the clustering of the samples. Finally, “Validated Targets” and “Predicted Targets” indicate how many validated and predicted gene targets are linked to the sRNA. Clicking on one of the target entries will show a list of the gene targets as gene names.

Part 4: Enrichment analysis

The enrichment analysis of Oasis allows the user to select specific miRNAs (either manually or by filtering based on adjusted p-value threshold), and use the associated genes to find which gene ontology (GO) categories, KEGG pathways or protein-protein interactions are enriched for them.

1. Select the type of miRNA targets you would like to analyse. The “Targets type” selector box gives you three options: Only validated targets (miRTarBase (Hsu et al., 2014) and miRecords (Xiao et al., 2009)); Predicted targets (miRanda (Betel, Wilson, Gabow, Marks, & Sander, 2008)), or Both. While there might be less validated targets for a given miRNA, we recommend using only validated targets, as target predictions
have a notoriously high false-positive rate.

2. Select the enrichment analysis tool(s) you may want to use. At the moment, you are free to choose from the following web-services: gProfiler (Reimand, Arak, & Vilo, 2011), Genemania (Warde-Farley et al., 2010), DAVID (Huang, Lempicki, & Sherman, 2009), STRING (Snel, Lehmann, Bork, & Huynen, 2000), and STITCH (Kuhn et al., 2014). A brief description of each of these tools can be found at the end of this document. Check the boxes to select the enrichment analysis you would like to perform.

3. Lastly, you will have to select the miRNAs you want to analyse for functional enrichment. You can do this by checking the corresponding boxes on the column 'Select'.

Finally, submit your enrichment analysis by clicking the 'Enrichment analysis' button (Fig. 9, green box). You may submit jobs to multiple enrichment tools at once, but consider that a new browser window will open for each tool you use, and it will take longer to run. If the links do not open, try disabling your popup blocker.

Notes:

1. There is a limit of 1000 miRNA targets allowed in an automatic submission, so if you need to use more than 1000 targets for your enrichment analysis, we recommend you copy the list of targets and directly use it on the software of your preference.

2. In order to reset miRNA selections, click on the button “Clear Selection” (Fig. 9, yellow box).

3. The table can be sorted by any of its columns by clicking on the column header.

4. A click on the sRNA ID will redirect you to a detailed annotation on mirbase (miRNA), UCSC genome browser tracks (novel miRNA) or genecards (Rappaport et al., 2014) (other sRNAs).

5. In case you want to analyse the results manually, you can download Excel tables of all analysed sRNAs (by clicking on the link 'Open Total Results in excel’) or of sRNAs that passed the filtering stage (by clicking the link 'Open Filtered Results in excel').

Enrichment programs provided:

When having a set of DE genes associated with particular miRNAs, the most direct way to derive some biological context from them is to run an enrichment analysis. This analysis finds particular biological terms or pathways which contain have a strong representation of DE genes within them. There are various
tools to do such test, and here are several which are available through Oasis:

1. **gProfiler**: returns enriched GO categories, KEGG and REACTOME pathways, TRANSFAC regulatory motifs, miRBase miRNAs, CORUM protein complexes, Human Phenotype Ontologies and BioGRID protein-protein interactions.

2. **Genemania**: returns a protein-protein network, showing the protein products of the selected gene targets and how they associate with each other, as well as significantly enriched GO categories.

3. **STITCH** and **STRING**: returns a single image of a protein-protein network. On top of having protein-protein interactions, STITCH includes small molecules, drugs and ATPs associated with the target proteins as well.

4. **DAVID**: runs an enrichment test for all target genes, using various functional annotations (GO categories, KEGG pathways, BIOCARD pathways, protein domains and so on).

**References:**


