Supplementary A: Application Details for RF-LIME And RF-Faust

To compute RF-LIME the R package “lime published on CRAN and the R package “randomForestSRC” published on CRAN were used (https://CRAN.R-project.org/package=randomForestSRC and ” https://CRAN.R-project.org/package=lime). The Xgboost package is available on CRAN (https://CRAN.R-project.org/package=xgboost). The package “randomForestSRC is a fast OpenMP parallel computing of Breiman's random forests. The R package 'lime' is a port of the 'lime' 'Python' package.

The functionality of the package “lime” was applied in the default parameter setting. LIME explanation are based on a linear model with LASSO regularization. Although the value of penalty should influence the number of clusters in Tables 1-3, it was unfeasible to change this value in the R package due to a lack of documentation. Hence, this hypothesis could not be investigated further.

The computation random forests performed swiftly for the datasets. However, the computation of “lime” took a long time. We evaluated this explicitly with profiling (utils::Rprof) in the case of N=1500 of iris data with gaussian noise which yielded an overall computation time of 118 seconds of which the function lime::explain tool 114 seconds itself. In Figure 5 the facetted heatmap of RF-LIME is shown of the LIME algorithm [24] using the R package “lime”.

Faust (https://github.com/RGLab/FAUST) provides cell-counts per populations as an output. Each population in FAUST is described in form of a set of CDi+, CDj- conditions. Greene et al., provided several unsupervised and one supervised application based on a linear regression model for FAUST in their work [32]. Here, the

supervised application is adapted by using FAUST, as is proposed by the authors, as a feature extraction method and combining it with a typical random forest classifier because random forests are claimed to have the best performance [6].

Figure S1. Facetted heatmap of RF-LIME for the Iris data.