YMTHE, Volume *29*

Supplemental Information

Predicting genotoxicity of viral vectors for stem cell gene therapy using gene expression-based machine learning

Adrian Schwarzer, Steven R. Talbot, Anton Selich, Michael Morgan, Juliane W. Schott, Oliver Dittrich-Breiholz, Antonella L. Bastone, Bettina Weigel, Teng Cheong Ha, Violetta Dziadek, Rik Gijsbers, Adrian J. Thrasher, Frank J.T. Staal, Hubert B. Gaspar, Ute Modlich, Axel Schambach, and Michael Rothe

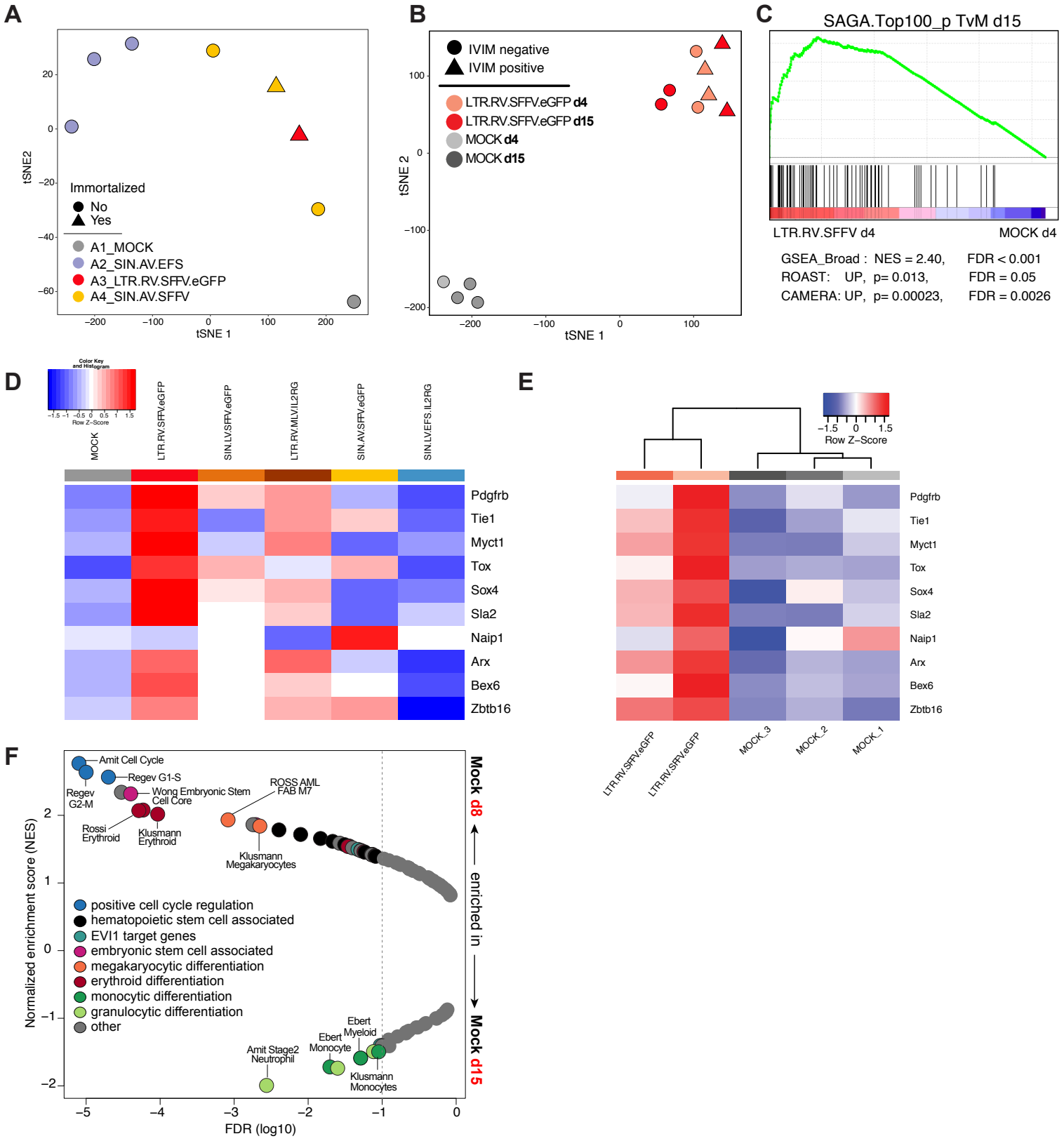

Figure S1. Transforming vectors impose specific gene expression changes in murine hematopoietic progenitors A) t-SNE representation of the gene expression profiles in samples transduced with SIN.AV.EFS, LTR.RV.SFFV and SIN.AV.SFFV. B) t-SNE representation of SAGA assays measured on day 4 (pale colors) and day 15 after batch correction using the assay date as a batch variable. C) GSEA-plot showing the enrichment of the Top 100 genes (upregulated on day 15 in transforming samples vs mock samples, rank based on p-values) in the day 4 samples transduced with LTR.RV.SFFV.eGFP compared to the MOCK control. Below the plot, the statistics for three different GSEA tests (Broad gene_set permutation; ROAST self-contained GSEA and CAMERA competitive GSEA test) are given as discussed in Materials and Methods. D) validation of gene expression changes by qPCR: row-scaled heatmap of qPCR based gene expression genes showing the highest $\log_2$FC by transforming vectors in the first three IVIM assays. e) validation of gene expression changes by RNASeq: row-scaled heatmap of RNA-Seq based gene expression (rlog transformed normalized counts) of the top-upregulated genes by transforming vectors. F) Gene set enrichment analysis of expression changes in 106 hematopoiesis-associated gene sets (Supplementary Table 3) in mock-samples from d8 versus mock-samples from day 15. Plotted are normalized enrichment scores (NES) against the false discovery rate (FDR) obtained by gene set permutation. Significant enrichment (FDR < 0.1) is indicated by the dashed line.

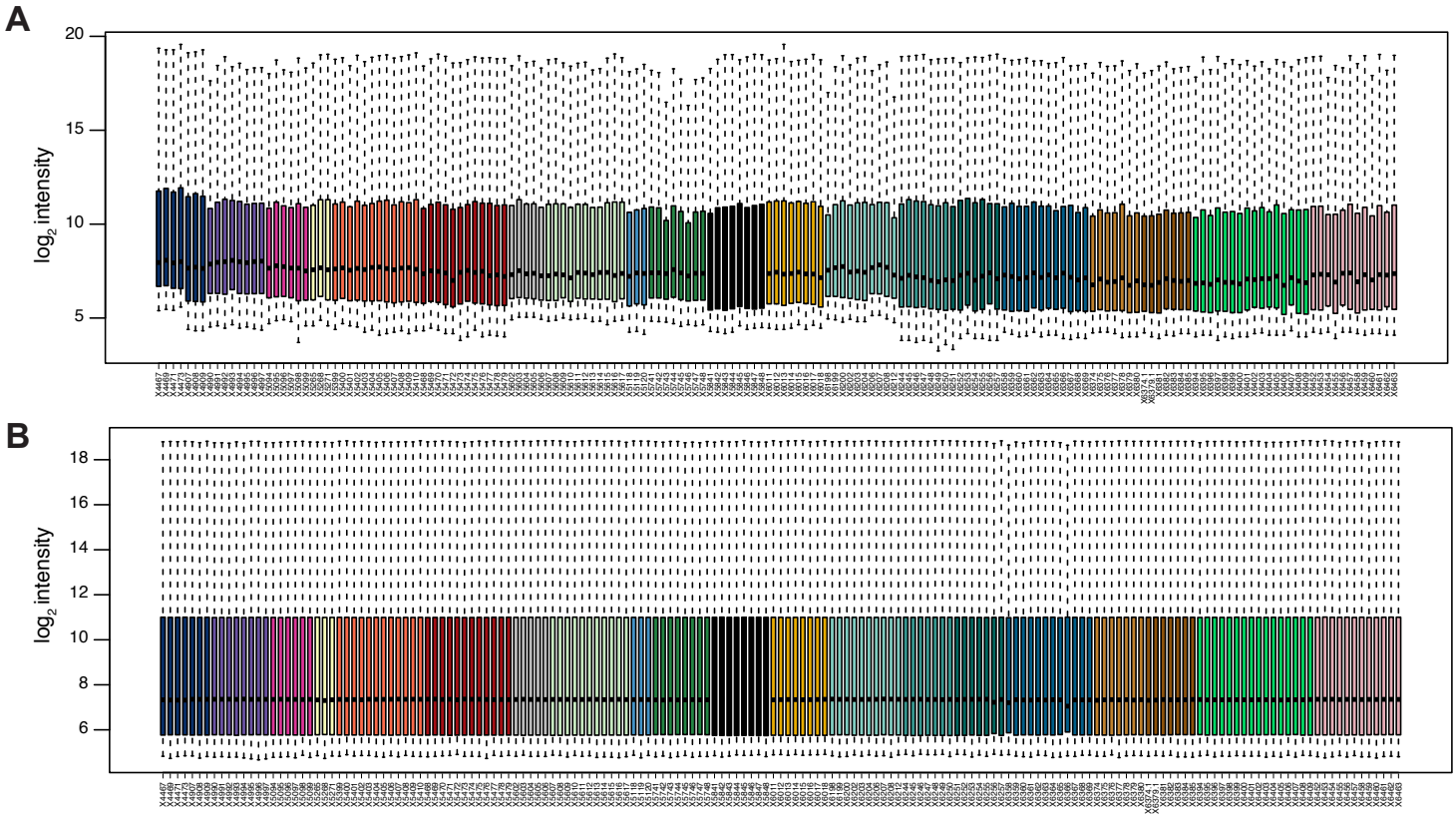

Figure S2 Quantile normalization of individual SAGA assays. A) Boxplot of $\log_2$ raw intensities of 169 SAGA samples (167 individual SAGA samples plus 2 mock duplicates from IVIM ID 180523; see Materials and Methods) hybridized to Agilent Microarrays. The coloring scheme denotes individual assays (batches). **B)** Boxplot of $\log_2$ intensities of 169 SAGA samples after quantile-normalization and averaging of quadruplicate probes

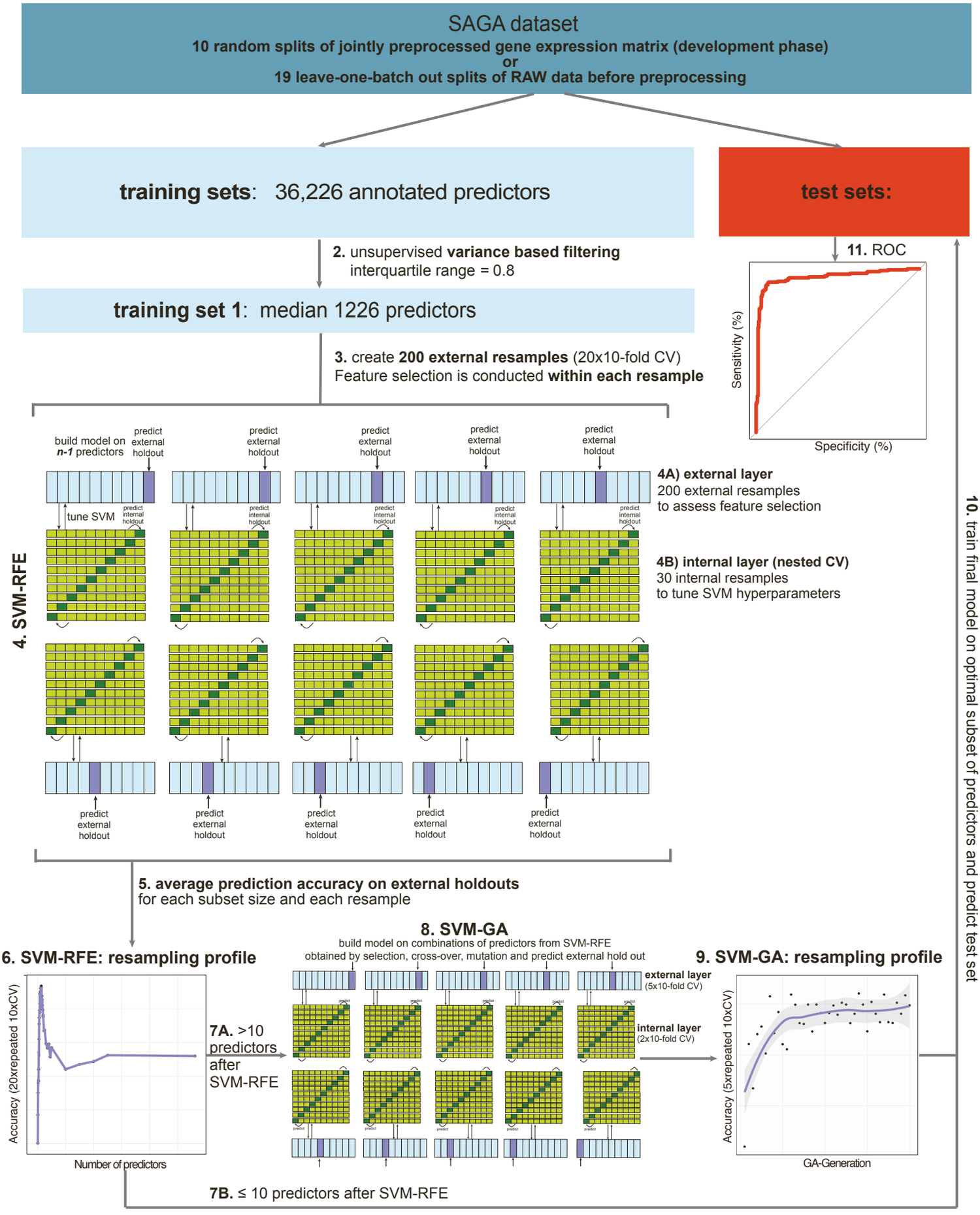

Figure S3. Training / Test set splitting and resampling during feature selection. 1) The dataset is split into training and test sets 2) An unsupervised filter is applied to the training set to filter out probes with low variance 3) 200 resamples of the training set are generated using 20 times repeated 10-fold CV. Each resample is comprised of 90 % of the training set (97 samples, light blue) for feature selection and a 10% hold-out sample (15 samples, purple) to assess prediction performance ("external layer"). 4A) Feature selection is performed within each resample by training/tuning the model using all n predictors and prediction of the external hold-out sample. Variable importance is calculated via AUC for each predictor and the process is repeated using the n-1 most important predictors 4B) Tuning of hyperparameters is performed at each iteration of feature selection using an "internal layer" that further splits each training sample from the outer loop using 3x10-fold CV. The optimal hyperparameter is passed to the outer loop to build the model. 5) results from the outer loop are aggregated into 6) a performance profile over the tested predictor subsets. 7) If SVM-RFE retains more than 10 predictors 8) a genetic algorithm is employed to find the best combination of retained features using a similar resampling scheme. 9) the resampling accuracy of each generation of the genetic algorithm is recorded and the optimal iteration is selected. 10) The final model is build using the optimal predictors on the complete training set and the test set is predicted. 11) The whole process is repeated for the remaining 9 training / test set splits and the prediction accuracies on the test sets are aggregated into a ROC statistic.

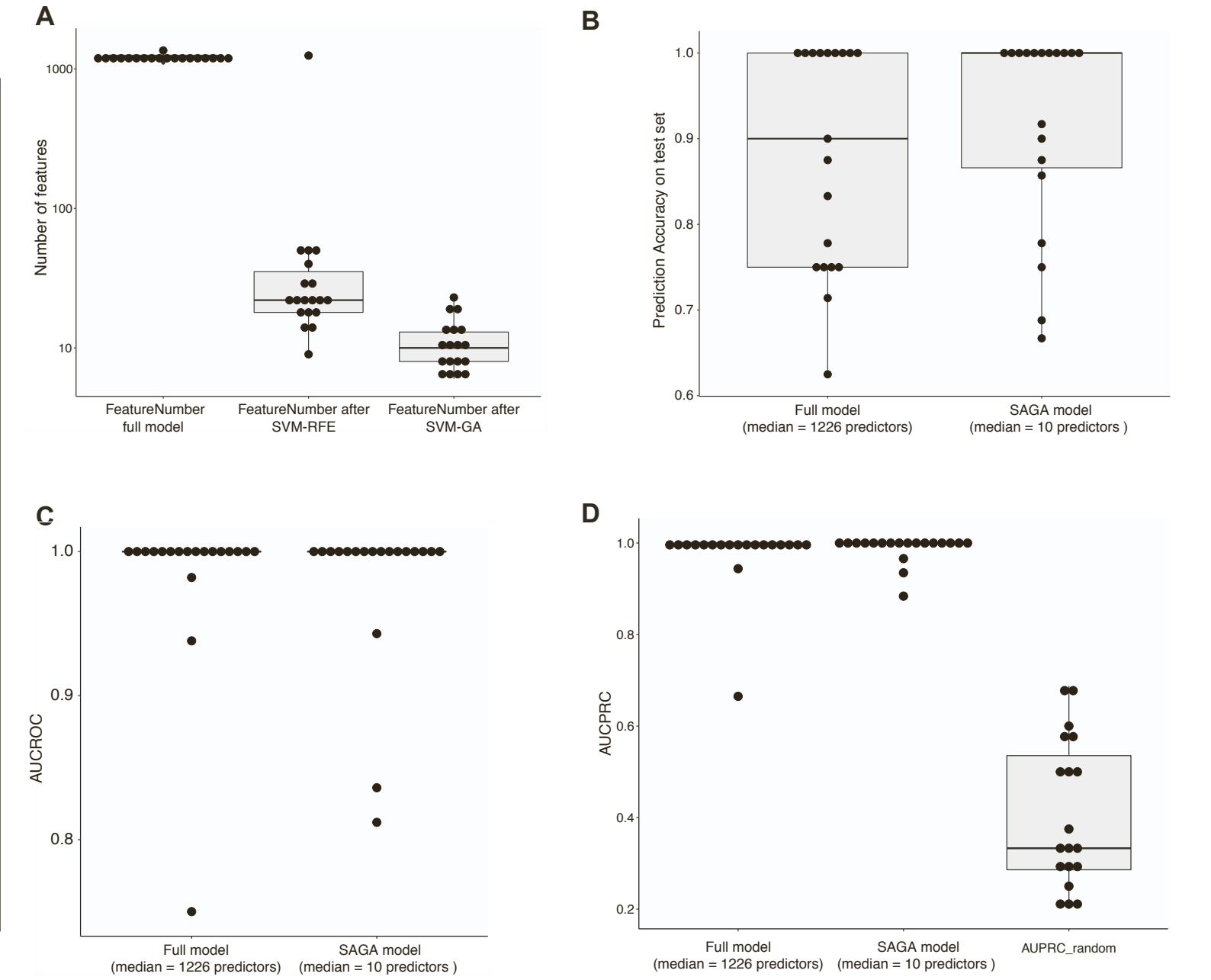

Figure S4 Classification performance metrics over the 19 iterations of the leave-one-batch-out approach. A) number of predictors used for the full model and as input for the feature selection routines (median = 1226), after SVM-RFE (median=22) and after SVM-GA (median = 10) **B)** boxplot of prediction accuracy on the 19 hold-out batches for the full model (mean prediction accuracy = 88.0 %) and SAGA (mean prediction accuracy = 91.7%, $P_{\text{Paired t-test}} = 0.242$) **C)** boxplot of AUC-ROC on the 19 hold-out batches for the full model (mean AUCROC = 0.98) and SAGA (mean AUCROC = 0.98) **D)** boxplot of AUC-PRC on the 19 hold-out batches for the full model (mean AUCPRC = 0.98) and SAGA (mean AUCPRC = 0.99)

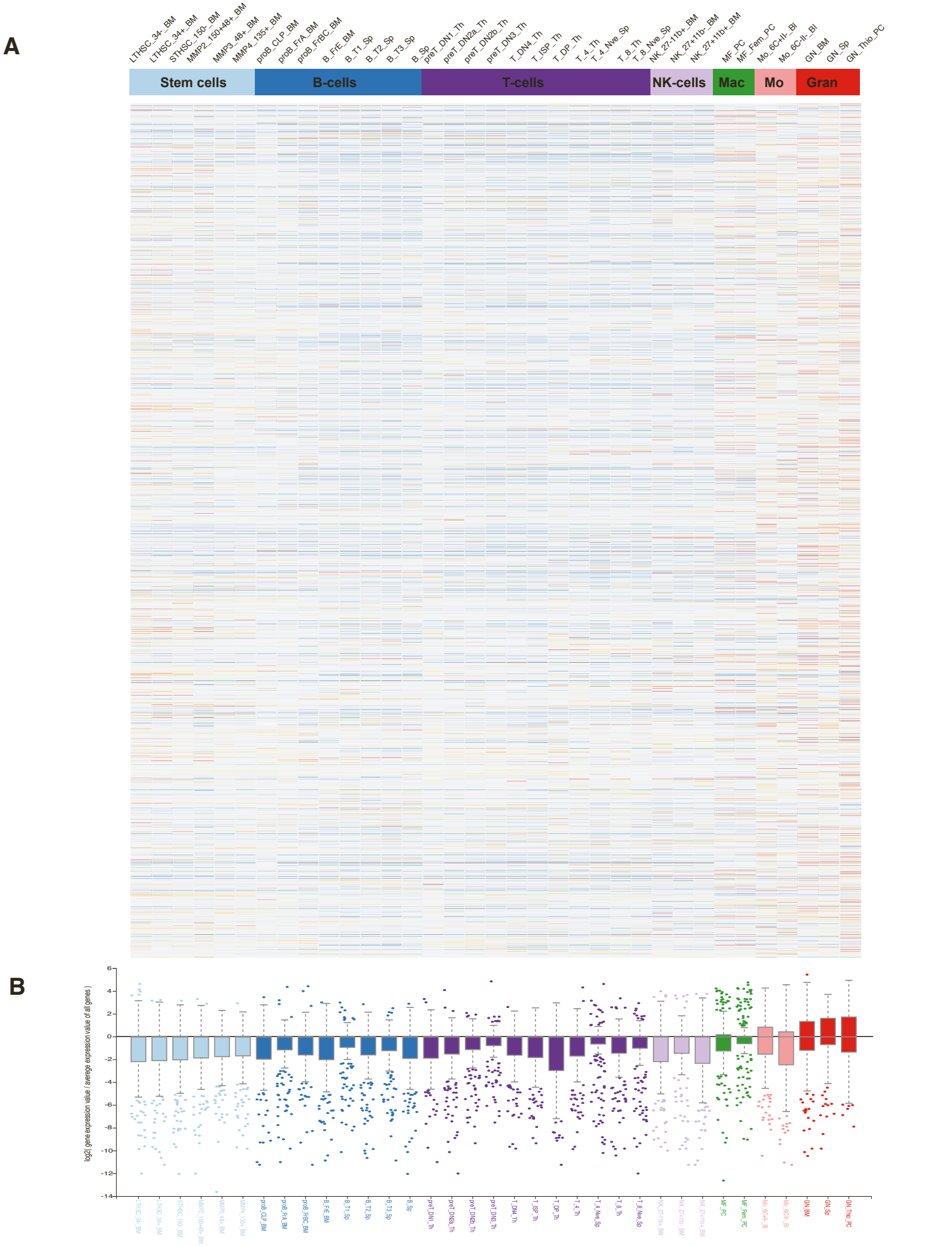

Figure S5 Expression of non-specifically filtered genes across the murine hematopoietic system. A) row- scaled heatmap of 1243 probes/genes retained after unsupervised filtering (IQR = 0.8) of the quantile normalized and batch-corrected expression matrix of n=152 SAGA samples. **B)** Boxplots of expression of genes in each column relative to the expression of all genes demonstrates relative enrichment of myeloid genes in the selection. Abbreviations: LT-HSC: long term-HSC, ST-HSC: short term-HSC, MPP: multipotent progenitors, Mac/MF: Macrophages, Mo: Monocytes, Gran/GN: Granulocytes.

Supplemental Methods and Materials

SAGA samples Table S8 tab 1 and tab 2 give an overview of all SAGA assays, experimental batches and the corresponding microarray samples used at each step. One SAGA assay consists of all SAGA samples generated in the same cell culture experiment. Each SAGA assay that was run independently is one experimental batch, with the following exemptions: SAGA assays with the IDs #160525 and #160706 were run in parallel and constitute one batch (batch 7). Due to severe class imbalance, one SAGA assay (ID #180523) had to be split into two separate batches (#180523A: batch 16, #180523B: batch 17) for normalization and batch correction. Each batch contained the two mock samples from assay #180523 and four or five LTR.RV.SFFV samples, respectively. **Table S8** lists all 179 SAGA samples used in this work, including 169 SAGA samples from day 15 (including the two mock duplicates X6374.1 and X6379.1 from assay #180523B), 5 SAGA samples from day 4, and 5 SAGA samples from day 8. For the computation of differentially expressed genes and pathways between mock, safe and transforming vectors, and for development of the SAGA classifier only samples from day 15 were used. These 169 samples were used as input into the microarray preprocessing pipeline, whose individual steps are visualized in the t-SNE plots of **Figures 3A-C** and **Figures S2A** and **2B** and described in detail in the paragraph "Microarray data processing". After preprocessing, the two mock duplicates (X6374.1 and X6379.1) and 15 samples for which the class label was unknown due to an insufficient number of IVIM assays or inconclusive IVIM results were removed from the analysis, resulting in a final dataset of 152 unique SAGA samples (65 transforming, 55 safe and 32 mock samples), which was used for differential expression, gene set enrichment analysis and development of the SAGA classifier. For the subsequent leave-one-batch-out approach, batch 17 was treated as independent test set with the two mock duplicates X6374.1 and X6379.1 included, resulting in a dataset of 19 test sets and 154 samples in total.

RNA isolation and microarray acquisition On day 15 p.t., cells from bulk cultures were pelleted (5 $x10^5$ to 2.5 $x10^6$ cells) and resuspended in 700 µl of RNAzol B reagent (WAK-Chemie Medical) and frozen at -80°C. Total RNA was isolated employing the Direct-Zol RNA MiniPrep Kit (Zymo Research) with on-column DNAse treatment. Four different microarray designs were used in this study, all representing a refined version of the Whole Mouse Genome Microarray 4x44K v2 (Design ID 026655, Agilent Technologies) comprised of all probes of this array in quadruplicates: (1) '026655AsQuadruplicatesOn4x180k' (Design ID 048306) was developed by the Research Core Unit Genomics (RCUG) of Hannover Medical School. Microarray design was created at Agilent's eArray portal using a 4x180K design format for mRNA expression as template. All non-control probes of design ID 026655 were printed four times within one 180K region. (2) '048306On1M' (Design ID 066423), (3) '048306On1M_V3' (Design ID 084107) and (4) '026655QM_RCUG_MusMusculus' (Design ID 084956) were also developed by RCUG, using a 1x1M design format for mRNA expression as template. All non-control probes of design ID 026655 were printed four times within a region comprising a total of 181560 features (probes) (170 columns x 1068 rows). Four of such regions were placed within one 1M region giving rise to four microarray fields per slide to be hybridized individually (Customer Specified Feature Layout). Control probes required for proper Feature Extraction software operation were determined and placed automatically by eArray using recommended default settings. 100 ng of total RNA was used to prepare Aminoallyl-UTP-modified (aaUTP)

cRNA (Amino Allyl MessageAmp™ II Kit; #AM1753; Thermo Fisher Scientific) applying one round of amplification as directed by the company, except for a two-fold downscaling of all reaction volumes. Prior to the reverse transcription reaction, 1 µl of a 1:5000 dilution of Agilent's One-Color spike-in Kit stock solution (#5188-5282, Agilent Technologies) was added to 100 ng of total RNA of each analyzed sample. Labeling of aaUTP-cRNA was performed with Alexa Fluor 555 Reactive Dye (#A32756; Thermo Fisher Scientific) as recommended in the manual of the Amino Allyl MessageAmp™ II Kit (two-fold downscaled reaction volumes). cRNA fragmentation, hybridization and washing steps were carried out as recommended in the 'One-Color Microarray-Based Gene Expression Analysis Protocol V5.7', except that 500 ng of each fluorescently labeled cRNA population were used for hybridization. Slides were scanned using the Agilent Micro Array Scanner G2565CA (pixel resolution 3 µm, bit depth 20). Data extraction was performed with the 'Feature Extraction Software V10.7.3.1' with the extraction protocol file 'GE1_107_Sep09.xml'.

Microarray annotation Since microarray probe annotation may change as the genome annotation advances, we re-annotated the 39,428 probes on the Agilent Whole Mouse Genome Oligo Microarray 4x44K v2 (Design ID 026655) by mapping the 60mer sequences to a recent release of the murine transcriptome (Gencode version M18[1], GRCm38.p6, release 07/2018). The transcript databases were downloaded as FASTA files for the 64,732 protein coding transcripts (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M18/gencode.vM18.pc_transcripts.fa.gz) and for all 136,535 coding and noncoding transcripts of the reference transcriptome (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M18/gencode.vM18.transcripts.fa.gz). The annotation was performed using R 3.5.1, Bioconductor 3.7[2] and the R package "Biostrings"[3]. Due to their higher expression compared to non-coding transcripts[4] we prioritized protein coding sequences by aligning the 60mers first to all protein coding transcripts of Gencode M18, and second to all transcripts of Gencode M18 allowing a maximum of 3 mismatches per 60mer. Using these parameters 33,361 out of 39,428 probes were successfully mapped to the Gencode M18 transcriptome. The mapping process retrieved an Ensembl-GeneID (e.g. "ENSMUSG00000020743") for each probe with a hit in the Gencode transcriptome. The Ensembl-GeneID was further annotated using the "BiomaRt" [5] R package to retrieve gene symbols, description and gene type from the Ensembl 94 database. For probes that could not be annotated by Gencode, annotation was taken from the latest annotation file for the Whole Mouse Genome Oligo Microarray 4x44K v2 downloaded from Agilent eArray web service (https://earray.chem.agilent.com/earray/, ID 026655, released October 2017) resulting in annotation of 2872 additional probes and 36,226 annotated probes in total. The R script for the annotation and all files used are available in the GitHub repository accompanying this manuscript.

Microarray data processing The data was analyzed using R 3.5.1 and Bioconductor 3.7[2]. Raw files were read in separately for each array design and a merged dataset was created by extracting all probes derived from the original Agilent Mouse Genome Oligo Microarray 4x44K v2 array from the four array platforms and combining them using the function "cbind.EList" from "limma". The probe with the ID "A_55_P2337033" interrogating the gene "2310065F04Rik" was excluded from the dataset since it strongly cross-reacted with the sequence of EGFP. Array quality was assessed by interrogation of probe intensity distributions and by principal component analysis of $\log_2$-transformed unprocessed data. The Raw data was $\log_2$-transformed and quantile-normalized using the "limma" package. The success of preprocessing was verified by inspection of probe intensity distributions before and after preprocessing (**Figure S2A** and **2B)**. The four within-array replicates of each probe were collapsed using the "avereps" function from the R package "limma" resulting in a dataset with 39,428 unique probes. Probes interrogating the same gene were not collapsed further since most genes were only interrogated by one probe on this platform. In the quantile-normalized data, a substantial batch effect between different SAGA assays was observed (**Figure 3A)**. Batch correction between different SAGA assays was performed on quantile-normalized $\log_2$-values using the parametric ComBat algorithm as implemented in the R package "sva" [6] (**Figures 3B** and **3C)** with the SAGA number as batch variable and all other parameters set to default.

t-SNE and PCA visualizations Two-dimensional representation of gene expression profiles was visualized by t-distributed stochastic neighbor embedding (*t*-SNE)[7]. The Barnes-Hut implementation of *t*-SNE from the "Rtsne"-package[8] without prior dimension reduction was used for all t-SNE representations. For each t-SNE plot, Barnes-Hut t-SNE was run 1000 times with different random seeds and the iteration with the lowest Kullback-Leibler divergence was selected for visualization as a 2D plot. For t-SNE visualizations of the whole dataset (**Figures 3A-C**), all 39,428 probes were used and the perplexity was set to 16, since this exceeded the average number of samples within each cluster/SAGA assay and is within the range of 5-50 proposed by the authors of t-SNE[7]. For the t-SNE plots in **Figures 2A**, **2C** and **Figure S1A**, 36,226 annotated probes were used and the perplexity was set to 2, which was the maximum value allowed for this sample size. The "prcomp" function from the R package "stats" was used to perform principal component analysis. The function "heatmap.2" from the R package "gplots" was used to generate heatmaps on the number of probes indicated in the figure legend. Heatmaps were row-scaled with the color key indicated below the heatmap. Variance-based filtering of probes for unsupervised analysis was performed using the interquartile range (IQR) function in the package "genefilter" resulting in the number of probes indicated in the figure legend.

Differential expression analysis Differentially expressed probes between the subgroups were computed on the quantile normalized and batch corrected expression matrix of 36,226 annotated probes using the moderated t-test of the "limma" package[9] with Benjamini-Hochberg multiple testing correction. We computed the Toplists (differentially expressed genes) for the following contrasts: "transforming – mock" (**Table S2 tab 3**) , "safe – mock" (**Table S2 tab 4**), "transforming – safe" (**Table S2 tab 5**) and "transforming – (mock+safe)/2" (**Table S2 tab 1**) for 152 SAGA samples with known IVIM properties (65 transforming, 32 mock and 55 safe).

Gene set enrichment analysis The quantile normalized and batch corrected SAGA expression matrix (36,226 annotated probes, 152 samples with known IVIM properties (transforming, mock and non-transforming ("safe")) was first filtered for gene symbols that appear at least once in the interrogated MSigDB.v6.2 (C2, C3, C5, C6, hallmark) gene set collections[10]. In cases with multiple probes per gene, the probe with the highest standard deviation across the samples was selected, resulting in a gene expression matrix consisting of 15,376 probes/rows interrogating 15,376 unique genes. From this matrix .gct files were generated containing all 65 transforming and 32 mock samples (contrast "transforming vs mock", **Table S3 tab 2 – tab 4**), 65 transforming and 55 safe samples (contrast "transforming vs safe", **Table S3 tab 7**), 55 safe and 32 mock samples (contrast "safe vs mock", **Table S3 tab 8 – tab 10**), 65 transforming, 32 mock and 55 safe samples (contrast "transforming vs mock and safe", **Table S3 tab 11 – tab 13**). For samples from day 4, cultures LTR.SF.EGFP (n=4) and one mock sample were used (contrast "transforming vs mock day 4", **Figure S1C**). For the comparison of day 8 and day 15, samples (**Table S3 tab 14 – tab 16**) were preprocessed together with all 169 SAGA samples and treated as a separated batch in COMBAT. For the GSEA contrast "d8 mock vs d15 mock", the two mock samples from day 8 were compared to 32 mock samples from day 15. The .gct files were used as input for the Broad GSEA software[10] together with a .chip file containing the annotation for the 15,376 probes. GSEA was performed with ranking the probes according to signal to noise ratio and the permutation type set to "gene_set" (10,000 permutations). First, we used 106 custom gene sets related to hematopoiesis and leukemia (**Table S3 tab 1**)[4]. In addition, 8286 gene sets were tested for enrichment from MSigDB.v6.2 (C2, C5, hallmark gene sets). The enrichment results were visualized by plotting the normalized enrichment score (NES) against the FDR (**Figures 2F-2H**). For visualization purposes, gene sets with a nominal FDR of zero were assigned a $\log_{10}$ FDR between -5 and -6 in **Figure 2F-2H** and **Figure S1F**. **Table S3 tab 2 – tab 16** contain all exact results of GSEA computations. Competitive gene set tests using permutation of genes assume statistical independence of genes in the gene sets, which is unrealistic in most cases. It has been shown that inter-gene correlation can lead to falsely significant P-values in these tests [11]. In contrast, permutation of the sample labels preserves inter-gene correlation, but requires a substantial number of samples in each group, suffers from low statistical power and inevitably alters the hypothesis being tested. Therefore, we additionally performed GSEA with ROAST (rotation gene set tests for complex microarray experiments[12]) and CAMERA (competitive gene set test accounting for inter-gene correlation[13]) from the limma package by applying both functions to the matrix of 15,376 probes and computing the same contrasts as with the Broad GSEA tool. The parameters for ROAST were set to 50,000 rotations and set.statistic="mean" (default value). For CAMERA the inter.gene.cor parameter was set to 0.01, as proposed by the authors[13]. CAMERA and ROAST allow for non-independence of genes by estimating the inter-gene correlation (CAMERA) or using rotation of residuals to generate a valid null distribution (ROAST)[12]. Importantly, both methods test different null hypotheses: whereas CAMERA is a competitive test that interrogates whether genes within the gene set of interest are significantly more often differentially expressed compared to genes outside of the gene set, ROAST is a self-contained test that tests whether a defined proportion of genes within the gene set is differentially expressed at all. However, while both methods have been shown to control the FDR correctly compared to methods based on gene permutation[12,13], they do not report a normalized enrichment score or a similar measure, making it difficult to assess how strong the gene set is enriched at the top or bottom of the ranked gene list. This also makes comparisons between different gene sets difficult. Therefore, we report both the results of GSEA with the intuitive and widely used NES (normalized enrichment score) and the results of CAMERA/ROAST based on a rigorous test statistic. All gene sets labeled in **Figures 2F-2H** and **Figure S1F** were found to be significantly enriched (FDR < 0.1) by at least one additional method (ROAST or CAMERA), whereas most of the gene sets were found by both additional methods (**Table S3 tab 2 – tab 16)**. For the enrichment map network shown in **Figure 2M**), the output from the GSEA analysis querying 8,286 gene sets from MSigDB.v6.2 was used as input for the Enrichment Map Tool[14] for Cytoscape 3.7.1. Gene sets with a nominal FDR < 0.05 were selected for visualization in the network graphs. The color of the nodes encodes normalized enrichment score as shown in the color key. A similarity cutoff of 0.375 (combined Jaccard and overlap) was used.

Classifier development phase The development of the predictive model was implemented using the R package "caret"[15] based on a support vector machine with a radial basis function kernel (method = "svmRadial"). Unless otherwise specified, all calls to functions mentioned in this paragraph belong to the "caret" package with key parameters specified in parentheses after the name of the function or directly discussed in the text. Computations allowing multiple cores, e.g. the feature selection routines, were run on a c5.18xlarge Amazon Web Service EC2 instance with 72 cores and 144 Gb RAM running RStudio 1.1.456 and R 3.5.1. The data splitting and resampling scheme to assess the performance of the models and control for overfitting is outlined in **Figure S3**. First, the

quantile normalized and batch corrected expression matrix (36,226 annotated probes, 152 samples with known IVIM properties) was partitioned into a training set comprised of 70% of the samples (107 samples) and an independent test set of 30% of the samples (45 samples). The test set was not used at any point for feature selection or model tuning. To allocate samples to the test or training set the caret function "createDataPartition" (p=0.7) was used, which performs stratified sampling based on the class labels to keep the distribution of transforming and nontransforming samples equal between the training and test sets. Since a single training / test set split can lead to a biased assessment of model building and feature selection[16] ten stratified random training / test set splits of the dataset were created and the complete model building pipeline was run for ten times for a more unbiased and reliable assessment of the predictive modeling process. Predictive performance of many models, especially support vector machines, can be significantly affected by large numbers of irrelevant predictors[17]. Furthermore, models using fewer predictors are quicker to compute, less prone to overfitting and generally better interpretable than models based on thousands of predictors[16]. Therefore, a combination of feature selection steps was performed to reduce the number of predictors as far as possible while maintaining or increasing predictive power. First, we applied an unsupervised filter to each training set to exclude probes interrogating genes that were not expressed at all or show only little variation in the dataset. This step helped to reduce computation time and avoided the selection of features by the subsequent *SVM-RFE* step that have a good discriminatory power between the classes based on their AUROC, but display only a small absolute fold-change between the different classes. The R-package "genefilter" was used to discard probes with an interquartile range (IQR) of $\log_2$-expression values less than 0.8 in the quantile-normalized and batch corrected training cohort, which retained a median of 1,195 out of 36,226 annotated probes (**Table S4 tab 1**). IQR = 0.8 was chosen empirically, since it consistently selected around 1,000 features in all test/training set splits. Setting the IQR lower (e.g. IQR= 0.5) retained too many features (median around 4,500), leading to a substantial increase in overall computation time as well as a failure to reduce the number of features in the subsequent *SVM-RFE* step in 3 out of 10 training/test splits. In contrast, setting IQR=1.2 selected on average around 250 features, which could be efficiently handled by *SVM-RFE*. However, at IQR=1.2 important predictors, such as A_55_P2077048/Itih5 (AUROC= 0.98) were already discarded before the actual feature selection step. The implementations using IQR 0.5 and IQR = 1.2 are available at GITHUB. Next, we performed recursive feature elimination (*SVM-RFE*) on the training set using the function "rfe". Since feature selection is part of the model building process, it needs to be conducted inside of a resampling layer ("external resampling layer", **Figure S3**) to assess the impact of the selection process on the model performance and to prevent overfitting of the model to the predictors. To establish the external resampling layer, 200 resamples of the training set were created by twenty times repeated 10-fold cross-validation using the function "createMultiFolds" (Parameters: k=10, times = 20). The function divides the entire training set (107 samples) into 10 subsets (folds) of equal size and the first fold (11 samples, "external holdouts") is predicted by a model fit to the remaining 9 folds (96 samples, "external training") of the data. This is repeated with the second fold after the first one has been returned to the training set and so on, resulting in 10 resamples for each of the twenty repeats of 10-fold CV. Importantly, the 200 identical resamples were used to fit the full models using all predictors, to allow a direct comparison of the *SVM-RFE* model and the full model using the resampling accuracies. The 200 resamples were submitted to the helper function "rfeControl", which controls the details of the external resampling process of the function "rfe". The feature selection process itself was carried out for each of the 200 resamples separately and computed in parallel by setting the "rfeControl" parameter: "allowParallel = TRUE". To ensure reproducibility of the analysis, a fixed set of random seeds that "rfe" uses at each resampling iteration was created and submitted to "rfeControl" via the "seeds" parameter. Within each resample, *SVM-RFE* ranks all predictors according to their individual receiver operating characteristic (ROC) on the 96 training samples. In each iteration, less important predictors are removed, the model is fitted to the 96 training samples and the 11 holdout samples are predicted. The metric to be maximized by "rfe" was set to "Accuracy". After initial inspection of the resampling profiles, we noted that accuracy peaked most often between 5-30 predictors. For maximum resolution within these ranges, all subset sizes from 1-40 predictors were tested. Outside of this range, wider intervals were used (45, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500 predictors), resulting in 52 subset sizes in total. For each tested subset within each resample of the external layer an additional "inner layer" of resampling had to be established to determine the tuning parameters of the SVM-model. The details of the inner resampling layer were specified by the helper function "trainControl" and set to three times repeated 10-fold cross-validation (30 resamples). To be precise, each training set from the external layer (96 samples) was partitioned further into 30 internal resamples comprised of 86 "internal training" and 10 "internal holdout" samples, respectively (**Figure S3)**. For each value of tuning parameters and each internal resample, the SVMrad model was fit to the 86 internal training samples and the remaining 10 internal holdout samples were predicted. The prediction accuracy from the 30 internal resamples over the different tested hyperparameter values was used to determine the optimal value for the tuning parameters and these parameters were passed to the external layer to fit the model and predict the external hold-outs. *SVM-RFE* with a radial basis function kernel has two tuning parameters: cost (penalty parameter) and sigma (inverse width of the gaussian kernel). For the cost parameter, the parameter "tuneLength" of the "rfe"- function was set to 20, resulting in cost values ranging from 2^{-2} - 2^{17}. For the sigma parameter an analytical estimate was used which is calculated by "rfe" internally by calling the function "sigest" from the R-package kernlab[18]. "sigest" uses the

methodology proposed by Caputo et al[19] to estimate a value for sigma which results in a good prediction performance when used with a radial kernel SVM. We validated this approach initially by a manual search for sigma over a wide range of values (2^{-15} - 2^{0}), but could not find substantially better solutions for our dataset than suggested by "sigest" (data not shown). Hence, using a fixed value for sigma estimated with "sigest" and tuning the SVM over the cost parameter only resulted in a substantially smaller hyperparameter space and reduced computation time for *SVM-RFE*. To find the best subset size for the entire training set, the prediction accuracy of the external holdout samples for each subset size and each resample was averaged into a resampling profile (**Figures 3E and 3F, Figure 5A**), which allowed to determine the best average subset size across all resamples. To generate the final set of predictors, "rfe" repeated the process on the complete training set with the optimal subset size determined from the resampling profile. The performance of the *SVM-RFE* model was compared to the full model using all predictors using the caret functions "resamples" and "diff", which compare resampling results of different models on a common data set comprised of identical resamples using a paired t-test[20]. The resampling-based results for the ten training / test set splits and the final model are tabulated in **Table S4 tab 1** (P-value_Resampling_full_vs_rfe). The GA procedure was implemented using the function "gafs" and its helper function "gafsControl" from the R package "caret". The gene expression matrix reduced to the probes found by the preceding *SVM-RFE* step was used as input into GA. Similarly to the *SVM-RFE* implementation, SVM-GA was conducted inside an external resampling layer to assess the performance of the GA-model over the generations (external resampling accuracy). 50 external resamples of the training sets or the final dataset were created with the function "createMultiFolds" (k=10, times = 5) and passed to the function "gafsControl", which controls the outer resampling process of the GA. The computational burden of *SVM-GA* is higher than for SVM-RFE, so only 50 external resamples were used to complete the analysis in a reasonable amount of time. The prediction performances on the external hold-out samples at each generation across all external resamples were averaged into the external resampling profile (**Figures 3G** and **5B),** which was used to determine the optimal number of iterations the algorithm should proceed **(Figure S3)**. To determine the final feature set, "gafs" applied the GA to the entire training set for the optimal number of generations from the resampling process. Further parameters of "gafsControl" were set to enable parallel computing for the external layer, to maximize the test statistic (accuracy) and to use fixed random seeds for reproducibility. In initial runs, using the default settings of "gaf" feature reduction was quite inefficient, leading to the removal of only 3-5 predictors on average. For a more effective reduction of feature numbers, the size of the initial predictor subsets (chromosomes) in the starting population was reduced. Therefore, the helper function of GA (caretGA$initial) that creates the initial population was modified to produce chromosomes comprised of a random 40% of predictors, instead of creating initial subsets ranging from 10% to 90% of predictors. The GA procedure itself was run for 40 generations, with a population size of 40, a crossover probability of 0.7, a mutation probability of 0.1. Elitism was set to 3, meaning that the best three solutions survive to the next generation. The metric to optimize was set to "accuracy", the classification method to "svmRadial". Similarly to the *SVM-RFE* process, the GA had an additional inner layer of resampling conducted at each generation within each resample and for each chromosome to tune the SVM. The inner resampling layer of GA was set to two times repeated 10-fold cross-validation (20 resamples) by the helper function "trainControl". For the cost parameter of the SVM, the parameter "tuneLength" was set to 12, for cost values between 2^{-2} – 2^{9}. The reduced tune length was chosen to save computation time after it had been determined from the preceding steps that the optimal cost parameter for the SVM was in the range of 2^{-2} - 2^{7}. For the sigma parameter, the estimate computed by "sigest" function from "kernlab" was used as described above. For the analysis of gene expression of the selected predictors across murine haematopoiesis (20 probes from *SVM-RFE* and 1243 probes after unsupervised filtering, **Table S6 tab 2,3 and Figures 5E** and **S5**), the online resource of the Immunological Genome Consortium[21] (http://rstats.immgen.org/MyGeneSet_New/index.html) was queried using the corresponding gene symbols of the probes as input.

Classifier performance metrics Samples in the test sets were predicted after training a support vector machine with radial kernel on the training set using all predictors (full model) or reduced to the optimal predictors found by *SVM-RFE* and *SVM-GA* (reduced models) by using the caret functions "train" and "predict", respectively. For training the full and the reduced SVM-models, identical parameters and resamples were specified in the "train" function (method = "svmRadial", metric = "Accuracy", tuneLength = 20, twenty repeats of 10-fold cross-validation). The function "predict" was used with the parameter "type" set to "prob", which computes the probability that a sample belongs to a given class. An unknown sample was considered belonging to the class "transforming" when the probability for class "transforming" was greater than 0.5. Performance estimates (sensitivity, specificity, accuracy, kappa) for the predicted test sets were computed using the function "confusionMatrix" on the predicted and the true class labels, respectively. For **Figures 3H-3J**, the resampling accuracies and their confidence intervals were determined using the function "resamples" for the full models, *SVM-RFE* and *SVM-GA* and plotted on the y-axis. The values on the x-axis represent the test set accuracies and the corresponding confidence intervals as output by the function "confusionMatrix". The "pROC" R-package[22] (v1.15.3) was used to compute and visualize the ROC curves for the test sets using the function "roc" on the probability for class "transforming" as output by the "predict" function. *P* values to compare the difference between the AUROC of two unpaired ROC curves were performed with the "roc.test" function using the "delong"

method and the alternative hypothesis set to "greater". Precision recall curves were generated using the R-package "PRROC" (v.1.3.1). As delineated in the main text, we defined SAGA as the compound model based on the predictions from *SVM-RFE* when this process yielded equal or less than 10 optimal predictors and from *SVM-RFE* followed by *SVM-GA* otherwise. For **Figures 4D-4I**, the prediction results for SAGA for all 19 independent test batches were aggregated and compared to the performance of the IVIM assay via AUROC, AUPRC and calculation of the confusion matrices and associated performance estimates (**Table S5)**.

Performance estimation via leave-one-batch-out approach Raw intensities of 169 arrays from 19 experimental batches were read in and combined into an "EListRaw" object without further modification. 15 samples with unknown ground truth were subsequently removed from the dataset, resulting in 154 assays including two mock duplicates (X6374.1, X6379.1 from batch 17, **Table S8**). For iteration 1, the raw data of batch 1 (IVIM #120411) was set aside as an independent test set, all other batches (2-19) were used as training set and were quantile normalized, averaged and batch corrected as described above. The preprocessed training set was subjected to *SVM-RFE* and *SVM-GA* using the same parameters as above, except for the numbers of subset sizes to assess during SVM-RFE, which were reduced to 1,2,3…,40,45,50, all predictors = 43 predictor subsets in total to limit computational costs. After having determined the optimal predictors in the training set, the raw training set was again quantile normalized and batch-corrected by the R package "bapred"[23], in order to estimate and store the parameters necessary for the later add-on correction of the test set. An SVM with radial kernel was trained on the bapred-adjusted training set reduced to the optimal predictors found by the feature selection routines. The hyperparameters of the SVM (sigma and cost) were determined by 20 times repeated 10-fold cross-validation as described above. At this point, the optimal features had been determined and the classifier had been trained and fixed using the training set only, whereas the test set had not been used. This was followed by add-on quantile normalization and add-on batch correction of the raw-test set using the bapred functions "qunormaddon" and "combatbaaddon", respectively. Add-on adjustment prevents the alteration of the training set by the addition of test set samples (information leakage) by applying the necessary adjustments to the test data using parameters estimated on the training data only[23]. The add-on adjusted test set was reduced to the optimal predictors determined on the training set (e.g. for the first iteration: 8 predictors) and predicted using the SVM trained before and the caret function "predict". The complete procedure was repeated 18 additional times with every available batch to be used one time as independent test set. The results from the 19 iterations of building SAGA and predicting the independent test batches are summarized in **Figure 4, Table S5** and **Figure S4.**

SAGA R package The R implementation of the SAGA classifier is available online (https://github.com/mytalbot/saga_package) and its functionality is described in detail in the package vignette. The SAGA package depends on R ≧ 3.6. The SAGA package expects data from microarrays based on Agilents Whole Mouse Genome 4x44K v2 platform as input. The Agilent Design IDs of compatible arrays are given in the section "*RNA isolation and microarray acquisition"* above. SAGA is a support vector machine with radial kernel that is trained on the complete SAGA dataset of 152 arrays reduced to the 11 optimal predictors derived from this dataset by applying the pipeline developed above (quantile normalization, batch correction and feature selection) to all 152 SAGA samples with known IVIM behavior (Table S6). The SVM is trained by using the "caret" function "train" with the following parameters: method = "svmRadial", metric = "Accuracy", tuneLength = 20 and five repeats of tenfold cross-validation for tuning the cost parameter, the sigma parameter is estimated internally by "train" as outlined before. The unknown samples are read in using the "limma" function "read.maimages" followed by add-on quantile normalization and add-on batch correction using the functions "qunormaddon" and "combatbaaddon" from the R package "bapred"[23]. Add-on adjusted test sets are then reduced to the 11 optimal SAGA predictors. Prediction of the unknown samples is performed by the function "predict" with the parameter "type" set to "prob" as described above. An unknown sample is considered belonging to the class "transforming" when the probability for class "transforming" is greater than 0.5. Prediction of unknown samples by *SAGA-GSEA* follows the procedure described in the paragraph *SAGA-GSEA*.

SAGA-GSEA For the implementation of *SAGA-GSEA*, complete assays were read in batch-wise, quantile-normalized, averaged and $\log_2$-transformed within each assay using the R package "limma". The preprocessed and unfiltered expression matrix with the Agilent ProbeIDs as row names was directly converted into an "epheno" object using the function "ExpressionPhenoTest" from the package "phenoTest"[24] with the phenotype variable ("Group") set to 1 for all mock samples in each assay and a unique value {2,3,…,n} for each of the samples to be tested against the mock samples. The normalized enrichment score, p-values and fdr were calculated for every sample against the mock samples using the function "gsea" from "phenoTest". During the leave-one-batch-out procedure, the optimal predictors found by *SVM-RFE* and *SVM-GA* for the training set of each iteration were used as geneset for GSEA. The raw data of the left-out test set was read in and preprocessed as described above followed by GSEA. IVIM #171102 was excluded from *SAGA-GSEA* since it had no mock samples available. The GSEA results were aggregated over the remaining 18 test sets. The ROC curve for *SAGA-GSEA* and the best NES cutoff were computed using the function "roc" on the normalized enrichment scores and the true class labels with the parameter "threshold" set to "best", which determines the NES associated with the point farthest to the diagonal

line[22]. A vector was assigned to the class "transforming" when its NES was greater than the optimal ROC-cutoff computed on the dataset after exclusion of the strongly transforming LTR.SFFV.eGFP samples (leave one-batch-out: NES>1.3, **Figure 6D**). For the final implementation of *SAGA-GSEA* to be used in the R-package, the 11 optimal predictors determined on the complete dataset for the final SAGA classifier (see above) are used as geneset. The optimal NES threshold for this geneset was determined by ROC-analysis after performing *SAGA-GSEA* on the 18 SAGA batches with mock controls available (NES > 1.0, **Figure 6G**).

Quantitative real-time PCR For quantitative real-time PCR (q-RT-PCR), 200 ng total RNA from day 15 samples were reverse transcribed with the QuantiTect Reverse Transcription kit (QIAGEN, Hilden, Germany). cDNA samples (20 µl reaction volume) were diluted with 20 µl water and 2 µl were used for each q-RT-PCR replicate. For quantification of gene expression in duplicate measurements, we used a mastermix of 7.5 µl 2x QuantiTect SYBR Green (QIAGEN), 0.75 µl 20x PrimeTime qPCR Assays (Mm.PT.39a.22214843.g, Mm.PT.56a.9170255, Mm.PT.58.10065691, Mm.PT.58.11560570, Mm.PT.58.5431010, Mm.PT.58.32478304.g, Mm.PT.58.41635140, Mm.PT.58.41288607, Mm.PT.58.12595646, Mm.PT.58.41494395, Mm.PT.58.5925960, all from Integrated DNA Technologies, Coralville, USA), 4.75 µl water and 2 µl diluted cDNA. The program in the StepOnePlus thermocycler (Thermo Fisher Scientific, Inc.) was 15 min 95°C, 50 cycles of 30 sec 94°C, 30 sec 60°C, 30 sec 72°C and a melt curve analysis with 15 min 95°C, 1 min 60°C and a gradual increase to 95°C for 15 min (2.3°C/min). Target gene expression was analyzed by the delta-delta Ct-method relative to *Actb*[25]. All transduced samples were compared to the mock control of the respective assay.

RNA-Seq RNA from three SAGA assays on day 15 was isolated as described above. RNA samples were sent for sequencing to Novogene Bioinformatics Technology Co., Hong Kong. The sample quality was verified with Agilent 2100. After mRNA enrichment with the NEBNext Poly(A) mRNA Magnetic Isolation Module, sequencing libraries were generated using the NEB Next® Ultra™ RNA Library Prep Kit from NEB. Samples were sequenced on an Illumina HiSeq2500. RNA-seq reads (on average 55 x 10^6 read counts per sample) were aligned to the Gencode mouse reference genome (GRCm38.p5) using Tophat2[26], which generated 44.5 x10^6 uniquely mapped reads on average. Count matrices were computed for Gencode defined transcripts and all reads that were unambiguously assigned to annotated exons were submitted to further expression analysis with DESeq2[27]. Heatmaps were generated by using rlog-transformed (Regularized logarithm transformation) values of normalized counts.

Supplemental References

1. Frankish, A., Diekhans, M., Ferreira, A.-M., Johnson, R., Jungreis, I., Loveland, J., Mudge, J.M., Sisu, C., Wright, J., Armstrong, J., et al. (2018). GENCODE reference annotation for the human and mouse genomes. Nucleic Acids Research *47*, D766–D773.

2. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., et al. (2004). Bioconductor: open software development for computational biology and bioinformatics. Genome Biology *5*, R80-16.

3. Pages, H., Aboyoun, P., Gentleman, R., and DebRoy, S. (2018). Biostrings: Efficient manipulation of biological strings. R package.

4. Schwarzer, A., Emmrich, S., Schmidt, F., Beck, D., Ng, M., Reimer, C., Adams, F.F., Grasedieck, S., Witte, D., Käbler, S., et al. (2017). The non-coding RNA landscape of human hematopoiesis and leukemia. Nature Communications *8*, 1–16.

5. Durinck, S., Spellman, P.T., Birney, E., and Huber, W. (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nature Protocols *4*, 1184–1191.

6. Leek, J.T., Johnson, W.E., Parker, H.S., Jaffe, A.E., and Storey, J.D. (2012). The sva package for removing batch effects and other unwanted variation in high-throughput experiments. Bioinformatics *28*, 882–883.

7. Maaten, L. van der, and Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 2579–2605.

8. Maaten, L. van der (2014). Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research *15*, 3221–3245.

9. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research *43*, e47–e47.

10. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences of the United States of America *102*, 15545–15550.

11. Gatti, D.M., Barry, W.T., Nobel, A.B., Rusyn, I., and Wright, F.A. (2010). Heading down the wrong pathway: on the influence of correlation within gene sets. BMC Genomics *11*, 574.

12. Wu, D., Lim, E., Vaillant, F., Asselin-Labat, M.L., Visvader, J.E., and Smyth, G.K. (2010). ROAST: rotation gene set tests for complex microarray experiments. Bioinformatics *26*, 2176–2182.

13. Wu, D., and Smyth, G.K. (2012). Camera: a competitive gene set test accounting for inter-gene correlation. Nucleic Acids Research *40*, e133–e133.

14. Merico, D., Isserlin, R., Stueker, O., Emili, A., and Bader, G.D. (2010). Enrichment Map: A Network-Based Method for Gene-Set Enrichment Visualization and Interpretation. PLoS ONE *5*, e13984-12.

15. Kuhn, M. (2018). caret: Classification and Regression Training.

16. Kuhn, M., and Johnson, K. (2013). Applied Predictive Modeling 1st ed.

17. Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. Bioinformatics *23*, 2507–2517.

18. Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab- An S4Package for Kernel Methods in R. journal of statistical software *11*, 1–20.

19. Caputo, B., Sim, K., Furesjo, F., and Smola, A. (2002). Appearance–Based Object Recognition Using SVMs: Which Kernel Should I Use? Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision.

20. Hothorn, T., Leisch, F., Zeileis, A., and Hornik, K. (2005). The Design and Analysis of Benchmark Experiments. Journal of Computational and Graphical Statistics *14*, 675–699.

21. Yoshida, H., Lareau, C.A., Ramirez, R.N., Rose, S.A., Maier, B., Wroblewska, A., Desland, F., Chudnovskiy, A., Mortha, A., Dominguez, C., et al. (2019). The cis-Regulatory Atlas of the Mouse Immune System. Cell, 1–37.

22. Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics *12*, 77–8.

23. Hornung, R. (2016). Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. BMC Bioinformatics, 1–19.

24. Planet, E. (2017). phenoTest Package. R Vignette, 1–15.

25. Pfaffl, M.W. (2001). A new mathematical model for relative quantification in real-time RT-PCR. Nucleic Acids Research *29*, e45.

26. Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., and Salzberg, S.L. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biology *14*, R36.

27. Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol *15*, 550.