Combinatorial selection of biomarkers to optimize gene signatures in diagnostics and single cell applications

Ivan Ferrari [1], Saveria Mazzara [2], Sergio Abrignani [1,3], Renata Grifantini [1], Mauro Bombaci [1,*] & Riccardo Lorenzo Rossi [1,*]

[1] INGM, Istituto Nazionale Genetica Molecolare “Romeo ed Enrica Invernizzi”, Milan, Italy. [2] Division of Haematopathology, IEO, European Institute of Oncology IRCCS, 20141 Milan, Italy. [3] Department of Clinical Sciences and Community Health, Università degli Studi di Milano, Milan, Italy.

[*] Correspondence to: rossi@ingm.org, bombaci@ingm.org

Abstract

Biomarker selection is a critical step in research and diagnostics: here we present the newly developed “combiroc” R package, for fast and reproducible identification of diagnostic and/or research biomarkers. The package introduces new features for automatic assessment of signal thresholds, as well as functions for identification of unlabeled samples. We also show how combiroc leverages better and unambiguous cell type assignment in single cell RNA sequencing experiments through the combinatorial selection of highly specific gene sub-signatures.

Context and results

In diagnostic medicine, a biomarker is often used to identify subjects with a disease, or at high risk of developing it. Moreover, it can be used to foresee the disease’s outcome, monitor its progression and predict the response to a given therapy. Obtaining a reliable result with the lowest number of markers is important, since a smaller number of biological markers translates in easier applicability and lower cost per testing. Omics methods generate big data sets and long lists of biomarkers (signatures) that do not always fit practical or clinical purposes. We recently described the CombiROC method [1,2] which can be used to select subsets of biomarkers from relatively small signatures using ROC curves from combinations of predictors ranked by specificity (SP), sensitivity (SE) and AUC [3–5]. This first approach was based on Shiny [6]: though easy to use for non-coders but it cannot be customized and it is computationally limited. We then decided to develop a versatile, new open source R package to allow full customization of protocols and the analyses of much bigger marker signatures. We also implemented completely new functions for the identification of highly efficient signature subsets and we applied them on well characterized single cell RNA sequencing datasets: we showed that combinations of as few as three or four individual markers selected from much bigger, traditional gene signatures greatly improve the ability to discriminate between different cell type clusters.

The combiroc package is freely available from CRAN[7]; its input data are matrices of markers measurements (i.e. biochem assay, or gene expression values) per samples belonging to two different classes (e.g. healthy/disease, treated/untreated). The package allows for extensive re-formatting of data, and we implemented functions to transform the data in tidy format [8] for easier ggplot2 visualizations [9] and for further combinatorial analysis. In the context of omics methods, marker signatures (i.e. lists of markers/genes characteristically expressed in a specific cell, tissue or condition) are usually made of tens, if not hundreds, of features. Combiroc is agnostic to omics methods, as long as the input is a signal intensity value generated by markers, and it is meant to act as a signature reducer by identifying a subset of markers from the original full signature without impairing its discriminatory power (or having even a higher one). Finding such subsets is not a trivial task, in fact, even from a few tens of genes there is a huge number of combinations (**Fig. 1a**). Moreover, a critical step of this process is the choice of a specific signal threshold which is strictly dependent on the nature of the measurement method. If data are produced in house and the detection system is known, the user may be able to choose a threshold with knowledge, thus defining the positivity for markers whose signal is above it. The same procedure can be challenging, or even impossible, when handling third party data generated elsewhere. We solved this problem implementing and testing a function for the automatic evaluation of the signals' distributions from the two labelled sample classes: when knowledge of signal properties cannot be explicitly set, combiroc automatically evaluates and proposes a threshold value to be used in subsequent computations. This method automates a previously manual and somewhat arbitrary process, and it showed to be consistent with previously reported results on AIH datasets [1]. The density distributions of signals from both classes can be computed and plotted and their overlap visually inspected, along with the suggested signal threshold value (**Fig. 1b**). Once the signal threshold and the stringency (i.e. the minimum number of positive markers making the sample positive, defaulting at 1) are chosen, all marker combinations are computed with their accuracy values, rankings, as well as the ROC curves of single markers and/or their combinations. The steps of the whole workflow are detailed in the package's accompanying vignette (**Suppl. Material 1**) and can be found in our GitHub repository (ingmbioinfo.github.io/combiroc/). The prediction of the signal threshold and the possibility to classify unlabeled samples are new important features of the new package workflow (**Fig.1c**): the linear models obtained for each selected marker/combination with the linear regression method (glm) embedded in the combiroc's *roc_reports()* function were used to predict the labels of unlabeled samples. The unlabeled test dataset (a dataset in which samples are not labeled, i.e. without the "Class" column) must be of the same nature and with the same set of markers of the dataset used to obtain the model with the combinatorial analysis (training dataset) (**Fig. 1d**).

A major limitation in most scRNA-seq analyses is the manual annotation step which is performed looking at expression of marker genes and remains the gold standard procedure to determine cell populations identities. This step is highly time-demanding as it involves the manual inspection of a considerable amount of cluster-specific genes. Many methods for the automatic classification of cells exist[10], but the power of clustering-based methods in standard scRNA-seq differential expression analysis can be complemented by the search for the best performing combinations of markers among wider gene signatures. To this aim, it's not correct to just take the top few genes, since signatures are specific only if taken as a whole. Since scRNA-seq gene signatures are de-facto lists of biological markers, we believe that combinations of markers discriminating between phenotypes can be applied to this problem. This is why we decided to apply the combiroc procedure to a well-known scRNA-seq dataset, the human 68K peripheral blood mononuclear cell (Fresh PBMC-68K) datasets from 10X Genomics[11]: this dataset is FACS-sorted, labelled and belongs to a series of datasets from different healthy donors[12] (**Fig. 2a, Suppl. Material 2**). The sparsity of single cell RNAseq data renders many established marker genes individually unreliable for cell classification[13], and rare cell types often do not have any unique established marker that can be used to discriminate among cell populations. This is the case for NKG7 which is well known as specific to NK cells, but also highly expressed in CD8 T cells. CCL5, conversely, is specific for CD8 T cells and highly expressed in NK cells too, leading to an often overlapping classification of these two cell types [12–14] (**Fig. 2b**). We used as a training dataset the reduced PBMC-68K dataset consisting of 700 labelled cells classified in 10 different clusters and expression values of the original NK cell scRNA-seq signature made of 30 genes. Using combiroc in three increasing stringency modalities (called *default*, *hi-performance*, and *sniper*, see Methods) we screened the 174,436 combinations of up to 5 genes belonging to the 30 genes signature and we found combinations describing the NK cell populations with very high SE and SP values, with particularly high SPs for those found with the sniper analysis mode: **Fig. 2c** shows the top four gene combinations according to their AUC, SE and SP values for each stringency modality. To validate the models associated with the top combinations first we processed in the same way two other independent PBMC datasets, the frozen PBMC-3K[15] and the blood PBMCs from COVID-19 patients[16] (all cell classification labels were removed from these test datasets, only to be used later as ground truth to assess performance of combiroc-driven cell labelling); then we calculated the probabilities predicted by the trained models (see Methods). We refer to such probabilities as "combi-scores" as a thorough metric to assess discriminatory power given by each combination. The combi-score, which here measures the predicted probability of being identified as NK cells given by a specific combination, unequivocally identified NK cells in the PBMC-3K dataset (**Fig. 2d**) and all three types of NK cells (16hi, 56hi, prolif) annotated in PBMC-Covid19 dataset (**Fig. 2e**). Similarly, high combi-scores were also observed for cells labelled as Innate Lymphoid Cells (ILC1_3 and ILC2), which is not surprising since NKs belong to ILCs family[17] (**Fig. 2e**) further supporting the value of our approach in identifying relevant cell types since even if ILCs were not annotated in the coarse-grained training dataset, they were indeed picked as NK-like in the test dataset. A residual identification of CD8+ cells occurred in the PBMC-Covid19 dataset (not in the PBMC-3K dataset), but at a significantly lower level. The whole score distributions for individual combinations such as #470 and #5018 showed that the residual CD8+ signal is well below those from NK cells (**Fig. 2f,g** and **Suppl. Fig. 2**). Interestingly, the combi-score was highly efficient in identifying cell types, both delivering higher specificity and a strong background reduction when compared to a recently published gene expression score[18] (**Fig. 2f,g** and **Suppl. Fig. 3a**). Remarkably, calculating the gene expression score on the combiroc combinations, as opposed to the whole 30 genes signature, confirmed that the sniper combinations were as accurate as the whole signature in indicating the NK clusters and that the values were even less noisy, suggesting that these combination can be used as a refined version of the NK signature (**Suppl. Fig. 3b-d**).

Conclusions

In summary, a combination made of only three or four genes selected with combiroc from a much bigger (one order magnitude) original signature is able to discriminate NK cells with very high specificity and can be used to unambiguously identify cell clusters among diverse or even very heterogeneous cell populations (**Suppl. Fig. 4**). The ability of combiroc to efficiently select highly specific sub signatures, combined with the flexibility of open source code, has the potential to impact diagnostics markers applications and to widen the applicability of scRNAseq signatures for cell type discrimination.

Figure legends

Fig.1 a, Number of possible combinations (blue curve) with *k* distinct elements chosen from *n* markers and their cumulative sum (C_{tot}, red curve). For a signature of *n= 30* markers there are more than a billion combinations. **b**, Signal intensity distributions for both classes of labelled input data (classes A and B) processed with *combiroc_long()* and *markers_distribution()* functions; these functions allow to estimate the optimal signal intensity threshold which is displayed on the plot with a vertical dashed line and whose value can be used as argument for further computations. **c**, Analytical workflow of the combiroc package: red boxes highlight new features introduced with this package. If the signal threshold of input data is unknown, it can be predicted by inspecting the signal's distributions. **d**, Labels (classification) of an unlabelled test dataset (left, data without “class” column) can be inferred (right, data with green “class” column) using regression models and metrics obtained with *roc_reports()* function on a labeled training dataset (bottom, “class” column in yellow).

Fig. 2 a, PBMC-68K dataset from 10X Genomics in the reduced version clustered with UMAP: cell labeling from these clusters was used to train the models distinguishing NK cells (CD56+ NK) from all other cell types. **b**, Expression levels across the different clusters of the top three CD56+ NK markers (GNLY, NKG7, CD7) and the CD8+ Cytotoxic T cells marker CCL5; all these genes are significantly expressed in both NK and CD8+ T cells. **c**, Screening of all combinations of up to 5 markers belonging to the 30-genes signature of CD56+ NK cell cluster from the training dataset allowed us to select the top four gene combinations for each type of analysis (default, hi-perf and sniper) according to their AUC, SE and SP values. **d**, Heatmap of the combi-score (predicted probability of belonging to NK cell population) in the PBMC-3K test data using the top four combinations obtained from the training dataset. The plotted values are median combi-scores. **e**, Heatmap as in the previous panel for cells identified as NK cells in the PBMC-Covid19 dataset. **f,** NK-Combi-scores computed with combination #5018, i.e. probabilities of being a NK cell according to combination #5018 across the different clusters of PBMC-3K test dataset. **g**, NK-Combi-scores by combination #5018 for cell clusters of PBMC-Covid19 test dataset.

Methods

Extracting the NK markers

The *datasets.pbmc68k_reduced* object was downloaded from the Scanpy API webpage (https://scanpy.readthedocs.io/en/stable/api.html#module-scanpy.datasets): then the differentially expressed genes (AnnData.uns['ranked_genes_groups']) were saved as .csv file and the top 30 genes of the column 'CD56..NK' were selected.

Building the single-cell training dataset

To build the combiroc training dataset we started from the raw matrix of PMBC-68K dataset [12] (http://pklab.med.harvard.edu/peterk/review2020/examples/zhang_pbmc/). This matrix was then transposed and rescaled in order to have non-centred values ranging from 0 to 10. The matrix was then subset by selecting the 700 cells present in PBMC-68K reduced version (also available as *datasets.pbmc68k_reduced* from the Scanpy API, see above), and the 30 NK markers (in alphabetical order) in order to obtain a 700 x 30 matrix. Finally, two additional columns were added, one as ID using the barcodes (cells) in the rownames and the second as the "Class" for each cell, specifying if each one is a NK cell ('NK') or not ('Other').

Finding the best marker combinations and models

To find the best combinations we used the *combi()* function. This function works on the training dataset by computing the marker combinations and counting their corresponding positive samples for each class (once thresholds are set). A sample, to be considered positive for a given combination, must have a value higher than a given signal threshold (*signalthr*) for at least a given number of markers composing that combination (*combithr*).

As described in the combiroc's vignette for the standard workflow (**Suppl. Material 1** and GitHub at https://ingmbioinfo.github.io/combiroc/articles/combiroc_standard.html), the argument *signalthr* of the *combi()* function should be set according to the guidelines and characteristics of the methodology used for the analysis or by an accurate inspection of signal intensity distribution. If specific guidelines or knowledge are missing, one should set the value *signalthr* as suggested by the *distr$Density_plot* feature.

The screening of combinations was limited to those composed by no more than five markers (setting the *max_length = 5* in the *combi()* function). While this choice is arbitrary, the reason for this was double: first, it allows a decrease of the number of combinations to compute, making the analysis more manageable; second and more important, it fulfills the original aim of the package which is to trigger easier research and clinical applications, looking for combinations significantly *shorter* than the original gene expression signature. We chose not to set a default for this number since it can vary depending on the field of application and the experimental and/or clinical context.

Optimal signal threshold prediction

To predict the optimal signal threshold we used the *markers_distributions()* function, setting the argument *signalthr_prediction = TRUE*. In this way *distr$Density_plot* (see combiroc's vignette for the standard workflow, **Suppl. Material 1**) will compute the threshold and show it besides the distribution of the signal intensity values for both classes; the threshold is computed as the median of the signal threshold values in *distr$Coord* at which SE and SP are greater or equal to their set minimal values (*min_SE* and *min_SP*). The optimal threshold is added to the "Density_plot" object as a dashed black line and a number, which is being used as *signalthr* value for *combi()* function.

Three modalities of combi() calculation

Combinatorial analyses on the training dataset were performed with 3 degrees of stringency, thus defining three modalities (i.e. three sets of arguments) of the function *marker_distribution()* in order to obtain an optimal *signalthr*. These three modalities are characterized by increasing severity of SE and SP cutoffs and increasing minimum number of markers that need to be above such cutoffs. Specifically:

- **Default** mode: firstly we performed the combiroc analysis steps with default parameters to assess the performance of classification without fine tuning: setting (suggested default values) min_SE = 40 and min_SP = 80 in *markers_distributions()* as well as *combithr = 1* in *combi()*, which is the default value),
- **Hi performance** mode: we then increased min_SE at its maximum given the data at hand by maintaining min_SP at 80 (min_SE = 62, min_SP = 80, combithr = 1).
- **Sniper** mode: finally, to further increase stringency we increase also combithr to 2. (min_SE = 62, min_SP = 80, combithr = 2, the stringest mode with highest SP at the cost of SE)

For each modality, we selected the training models generated with the top four combinations ranked with *ranked_combs()* and we proceeded with test datasets classification. Picking only the top four combinations to be carried on in the analysis was an arbitrary choice, still sufficient to demonstrate the package usage and utility: the user can explore more combinations according to the obtained SE and SP values.

Training models on selected combination

Regression models on the selected combinations were trained using the function *roc_reports()*, which applies the Generalised Linear Model (*stats::glm()* with argument *family= binomial*) on each one. The equation used to compute the prediction is the following:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n$$

Where βn are the coefficients (being β_0 the intercept) determined by the model and x_n the variables (signal values associated to markers). The predicted probabilities have been calculated with the sigmoid function:

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

The performance of each model is internally evaluated in function of the cutoff (p(x) value above which an observation is positively classified) and an optimal cutoff is finally returned (cutoff at which occurs the least possible error of classification on the training dataset observations)

Preprocessing the test datasets

As independent validation to test the selected combination and models we are going to use two different single-cell RNA sequencing datasets to see if the obtained models are able to correctly identify NK cells, without having to rely on the original 30-genes NK-signature.

As already mentioned in the introduction, these datasets are:

- PBMC-3K from Satija et al. 2015 ([15])
- COVID-19 PBMC Ncl-Cambridge-UCL from Stephenson et al. 2021. ([16])

The PBMC-3K test dataset was directly installed and loaded from the SeuratData library. The dataset is immediately available as a Seurat object named pbmc3k.final. The PBMC-Covid19 dataset from Haniffa's lab was downloaded from the COVID-19 cell atlas in the .h5ad format (see "COVID-19 PBMC Ncl-Cambridge-UCL" section, then the "haniffa21.processed.h5ad" file among the available downloads). Both test datasets underwent the very same steps of the training dataset preparation (transposition, scaling values from 0 to 10, genes subsetting to the alphabetically ordered 30 marker genes and addition of 'ID' column) with the exception of the addition of ‘Class’ column which, obviously, was subsequently inferred in the end of the analysis by fitting the previously mentioned models.

Validation tests on unclassified data

Test datasets were classified by fitting each computed model with *classify()* function following this logic:

$$C(x) = \begin{cases} 1 & p(x) > opt.cutoff \\ 0 & p(x) \leq opt.cutoff \end{cases}$$

- Cells with *p(x)* higher than the optimal cutoff are classified as “NK” (= 1).
- Cells with *p(x)* lower or equal to the optimal cutoff are classified as “Other” (= 0)

The performances of classification of each combination model were obtained by comparing the inferred labels with the original cluster labels.

Combi-score

For each cell of the test dataset was computed a “combi-score” value (basically p(x)) using the standard *stats::predict()* method, specifying *type=’response’*. The combi score is, for each combination, the probability of the prediction of GLM fits on the scale of the response variable. This score was then used to assess the presence of cells classified as ‘NK’ in NK cells clusters: in this context, the combi-score is the probability of being a NK-cell given by a specific marker combination.

Gene expression signature score

For each cell of the test dataset was also computed a “gene signature score” to check the effect of using selected combinations on a different published score developed for whole genes signatures. The gene signature score is described in Della Chiara et al 2021 ([18]). It takes into account both the expression level and co-expression of genes within each single cell. Given a geneset, the increase of gene-signature-score is directly proportional to the number of expressed genes in the signature *and* to the sum of their level of expression. We internally reproduced the score computation with the R function signature_score.R available in the GitHub combiroc package repository.
(https://github.com/ingmbioinfo/combiroc/blob/master/inst/external_code/signature_score.R)

Acknowledgements

We would like to thank Jens Geginat, Davide Cittaro, Asia Zonca, Stefano Biffo and Francesco Marabita for fruitful comments; members of the Bioinformatics Unit at INGM for constructive discussions.

Author contributions

R.L.R. proposed the package project and coordinated the whole work; I.F. conceived new features; R.L.R, M.B, I.F. designed the analyses; I.F. and R.L.R. performed analyses, wrote code, compiled the package, wrote manuscript, produced figures and vignettes; S.M. drafted code parts; M.B. searched and selected datasets; R.L.R., S.M. reviewed and debugged code; S.A., R.G., secured funding; all authors reviewed the manuscript.

Code Availability

The complete code base for the combiroc package is available from CRAN at the url https://cloud.r-project.org/web/packages/combiroc/index.html.
The development version of combiroc package, as well as workflows, demo data and precomputed R objects for both the standard procedure and the application to single-cell datasets are available on GitHub at https://ingmbioinfo.github.io/combiroc/.

Data Availability

PBMC-68K reduced (training dataset): downloaded from the datasets.pbmc68k_reduced Scanpy object: https://scanpy.readthedocs.io/en/stable/api.html#module-scanpy.datasets, and from the Kharchenko Lab website at the Department of Biomedical Informatics of the Harvard Medical School: http://pklab.med.harvard.edu/peterk/review2020/examples/zhang_pbmc/.
PBMC3K (test dataset): installed from the SeuratData library:
https://github.com/satijalab/seurat-data.
COVID-19 PBMC Ncl-Cambridge-UCL from Haniffa lab (test dataset):
https://www.covid19cellatlas.org/index.patient.html.

References

1. Mazzara, S. *et al.* CombiROC: an interactive web tool for selecting accurate marker combinations of omics data. *Sci. Rep.* **7**, 45477 (2017).
2. Bombaci, M. & Rossi, R. L. Computation and selection of optimal biomarker combinations by integrative ROC analysis using combiroc. *Methods Mol. Biol.* **1959**, 247–259 (2019).
3. Bombaci, M. *et al.* Novel biomarkers for primary biliary cholangitis to improve diagnosis and understand underlying regulatory mechanisms. *Liver Int.* **39**, 2124–2135 (2019).
4. Sola, L. *et al.* Enhancing antibody serodiagnosis using a controlled peptide coimmobilization strategy. *ACS Infect. Dis.* **4**, 998–1006 (2018).
5. Cano-Rodriguez, D. *et al.* TCTN2: a novel tumor marker with oncogenic properties. *Oncotarget* **8**, 95256–95269 (2017).
6. Chang, W. *et al. shiny: Web Application Framework for R.* (RStudio, 2021).
7. The Comprehensive R Archive Network. https://cran.r-project.org/.
8. Wickham, H. Tidy Data. *J. Stat. Softw.* **59**, (2014).
9. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis (Use R)*. 276 (Springer, 2016).
10. Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol.* **20**, 194 (2019).
11. Datasets - 10x Genomics. https://www.10xgenomics.com/resources/datasets.
12. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
13. Grabski, I. N. & Irizarry, R. A. Probabilistic gene expression signatures identify cell-types from single cell RNA-seq data. *BioRxiv* (2020) doi:10.1101/2020.01.05.895441.
14. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888-1902.e21 (2019).
15. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
16. Stephenson, E. *et al.* Single-cell multi-omics analysis of the immune response in COVID-19. *Nat. Med.* (2021) doi:10.1038/s41591-021-01329-2.
17. Mariotti, F. R., Quatrini, L., Munari, E., Vacca, P. & Moretta, L. Innate Lymphoid Cells: Expression of PD-1 and Other Checkpoints in Normal and Pathological Conditions. *Front. Immunol.* **10**, 910 (2019).
18. Della Chiara, G. *et al.* Epigenomic landscape of human colorectal cancer unveils an aberrant core of pan-cancer enhancers orchestrated by YAP/TAZ. *Nat. Commun.* **12**, 2340 (2021).

Figure 1

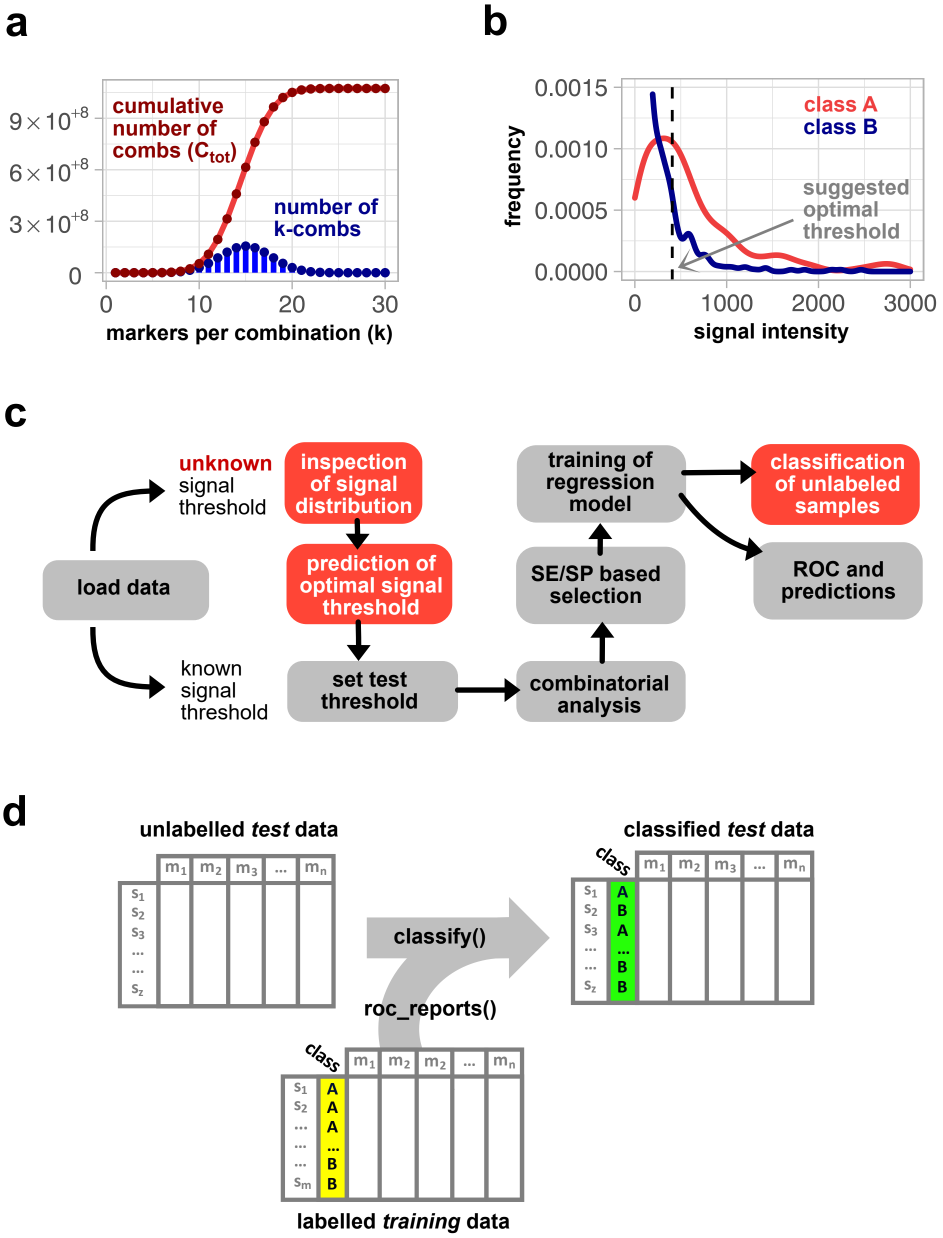
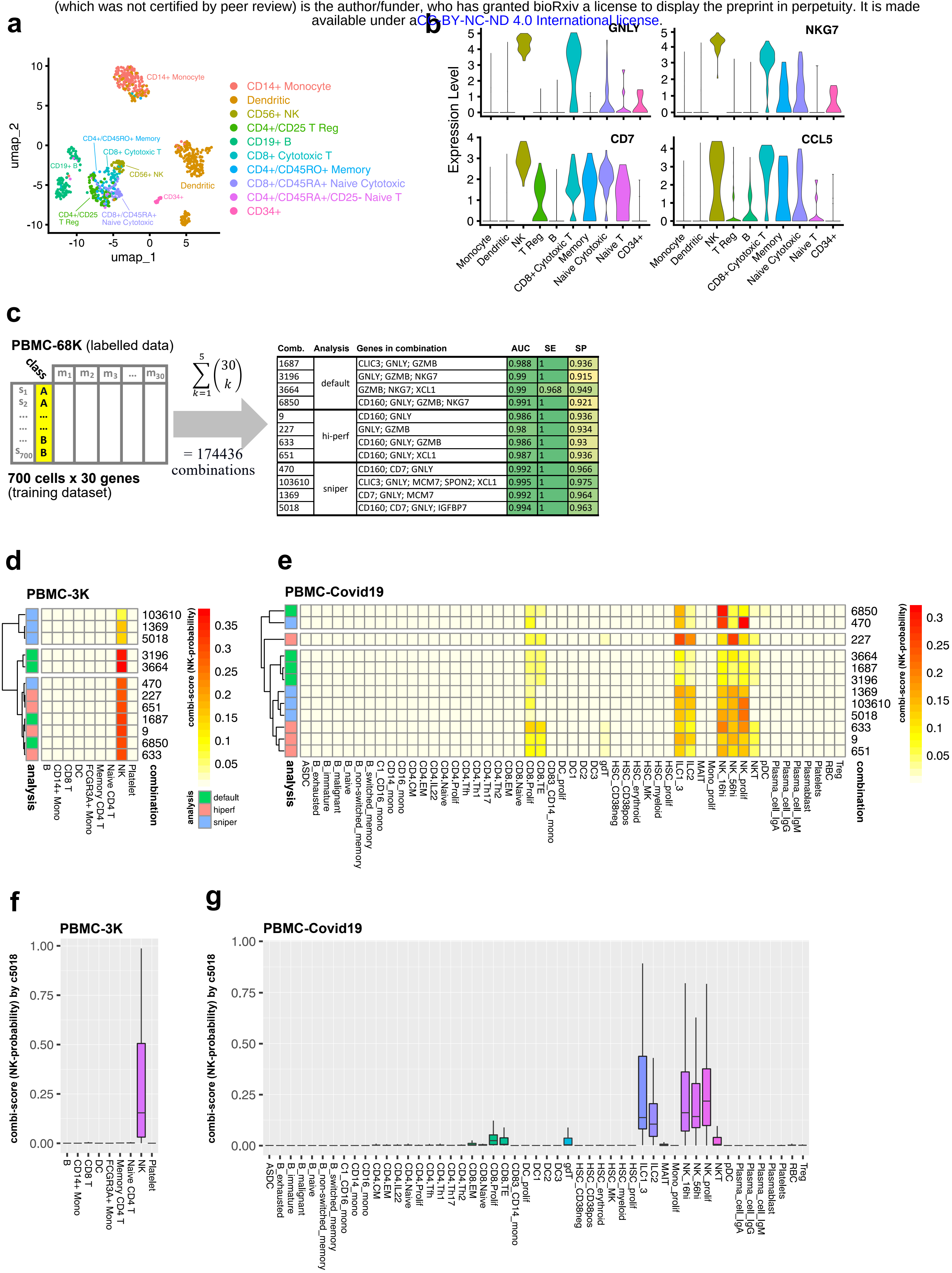

a
cumulative number of combs (C_{tot})
number of k-combs
$9\times10^{+8}$
$6\times10^{+8}$
$3\times10^{+8}$
0
0
10
20
30
markers per combination (k)
b
frequency
0.0015
0.0010
0.0005
0.0000
0
1000
2000
3000
signal intensity
class A
class B
suggested optimal threshold
c
load data
unknown signal threshold
known signal threshold
inspection of signal distribution
prediction of optimal signal threshold
set test threshold
combinatorial analysis
SE/SP based selection
training of regression model
classification of unlabeled samples
ROC and predictions
d
unlabelled *test* data
m_1 m_2 m_3 ... m_n
s_1 s_2 s_3 ... ... s_z
classify()
roc_reports()
classified *test* data
class
A B A ... B B
labelled *training* data
s_1 s_2 ... ... s_m
A A A ... B B

Figure 2



Comb.	Analysis	Genes in combination	AUC	SE	SP
1687	default	CLIC3; GNLY; GZMB	0.988	1	0.936
3196		GNLY; GZMB; NKG7	0.99	1	0.915
3664		GZMB; NKG7; XCL1	0.99	0.968	0.949
6850		CD160; GNLY; GZMB; NKG7	0.991	1	0.921
9	hi-perf	CD160; GNLY	0.986	1	0.936
227		GNLY; GZMB	0.98	1	0.934
633		CD160; GNLY; GZMB	0.986	1	0.93
651		CD160; GNLY; XCL1	0.987	1	0.936
470	sniper	CD160; CD7; GNLY	0.992	1	0.966
103610		CLIC3; GNLY; MCM7; SPON2; XCL1	0.995	1	0.975
1369		CD7; GNLY; MCM7	0.992	1	0.964
5018		CD160; CD7; GNLY; IGFBP7	0.994	1	0.963