CHAPTER TEN

Deconvoluting tumor-infiltrating immune cells from RNA-seq data using quanTIseq

Christina Plattner, Francesca Finotello, Dietmar Rieder*
Division for Bioinformatics, Biocenter, Medical University of Innsbruck, Innsbruck, Austria
*Corresponding author: e-mail address: dietmar.rieder@i-med.ac.at

Contents

Abstract

Tumor-infiltrating immune cells comprise various cells of the innate and the adaptive immune system, which influence tumor growth and response to immunotherapy by exerting anti- and protumorigenic functions. Therefore, the quantification of tumor immune infiltrates is of paramount importance for cancer immunology and immunotherapy.

We recently developed quanTIseq, a computational pipeline for the quantification of immune-cell fractions from bulk RNA sequencing (RNA-seq) data from blood or tumor samples. In this chapter, we show the capabilities of quanTIseq by analyzing two publicly available data sets. In the first example, we demonstrate how quanTIseq can be used to quantify circulating immune cells from preprocessed RNA-seq data and how to validate the results using matched flow cytometry data. In the second example, we analyze raw RNA-seq data from bulk tumor samples of melanoma patients collected before and on-treatment with kinase inhibitors to show how quanTIseq can be used to reveal the immunological effects of targeted and conventional drugs.

Methods in Enzymology, Volume 636
ISSN 0076-6879
https://doi.org/10.1016/bs.mie.2019.05.056

1. Introduction

Tumors do not only consist of malignant cells but also contain immune, stromal, and endothelial cells (Balkwill, Capasso, & Hagemann, 2012). It has been shown that the tumor microenvironment and the tumor immune contexture play a major role in tumor growth and response to tumor therapy (Chen & Mellman, 2017; Fridman, Pagès, Sautès-Fridman, & Galon, 2012; Galon et al., 2006). For instance, alternatively activated (M2) macrophages and regulatory $CD4^+$ T (T_{reg}) cells are known to contribute to tumor progression, whereas $CD8^+$ T cells can specifically recognize and kill tumor cells (Fridman et al., 2012).

Traditional methods to quantify cell fractions, like flow cytometry, immune fluorescence (IF) and immunohistochemistry (IHC) still suffer from important limitations, e.g., IHC/IF can only profile tumor-tissue slides that may not be representative of the whole tumor mass. Cutting-edge technologies like single-cell RNA sequencing or mass cytometry are currently too expensive to be used routinely and cannot be applied to archived samples. Moreover, cell-specific differences in the efficiency of single-cell dissociation can bias cell proportions (Finotello & Eduati, 2018). Therefore, there is need for computational methods to quantify the number of immune cells from bulk RNA-seq data.

We recently developed and extensively validated quanTIseq, a computational pipeline for the quantification of immune-cell fractions from blood or tumor samples through deconvolution of transcriptomic data (Finotello et al., 2019). As opposed to microarray-based methods, quanTIseq signature matrix was built from RNA-seq datasets from 10 different immune cell types (Table 1). Moreover, quanTIseq estimates immune-cell fractions referred to the total cellular content of the entire sample, allowing intra- and inter-sample comparison. In contrast to previous deconvolution approaches (e.g., CIBERSORT (Newman et al., 2015), EPIC (Racle, de Jonge, Baumgaertner, Speiser, & Gfeller, 2017); reviewed in Finotello and Trajanoski (2018)). quanTIseq implements a full pipeline for deconvolution of RNA-seq data, from data preprocessing and gene expression quantification, to cell-type deconvolution, simplifying its applicability and ensuring robustness of the full analytical procedure. Given that quanTIseq is available as a containerized software with all required dependencies, it is amenable to a broad user base by requiring only basic computer skills.

Table 1 Cell types quantified by quanTIseq.

Cell type	Cell ID in the output file
B cells	B.cells
Classically activated macrophages (M1)	Macrophages.M1
Alternatively activated macrophages (M2)	Macrophages.M2
Monocytes	Monocytes
Neutrophils	Neutrophils
Natural killer (NK) cells	NK.cells
Nonregulatory $CD4^+$ T cells	T.cells.CD4
$CD8^+$ T cells	T.cells.CD8
Regulatory $CD4^+$ (Treg) cells	Tregs
Dendritic cells	Dendritic.cells
Other uncharacterized cells	Other

In this chapter we will describe the quanTIseq pipeline, its implementation and its application using two example RNA-seq data sets from blood and tumor samples. In particular, we will show how to run the pipeline, specify relevant parameters for different sample or data types and, validate quanTIseq results using flow cytometry data.

quanTIseq is freely available at http://icbi.at/quantiseq.

2. quanTIseq deconvolution pipeline

quanTIseq is a deconvolution pipeline that quantifies the fractions and densities of 10 different immune cell types relevant for cancer immunology as well as the proportion of uncharacterized cells (e.g., malignant cells in bulk tumors). The cell types estimated by quanTIseq are listed in Table 1 where "T.cells.CD4" corresponds to nonregulatory $CD4^+$ T cells. The total $CD4^+$ T cell fraction can be obtained by summing up "T.cells.CD4" and "Tregs."

The quanTIseq pipeline consists of three main modules (see Fig. 1):

1. *Read preprocessing*: Trimmomatic (Bolger, Lohse, & Usadel, 2014) is used for raw RNA-seq reads (single- or paired-ends) preprocessing, to remove Illumina adapter sequences, trim low-quality read ends, crop long-reads to a maximum length, and discard short reads.

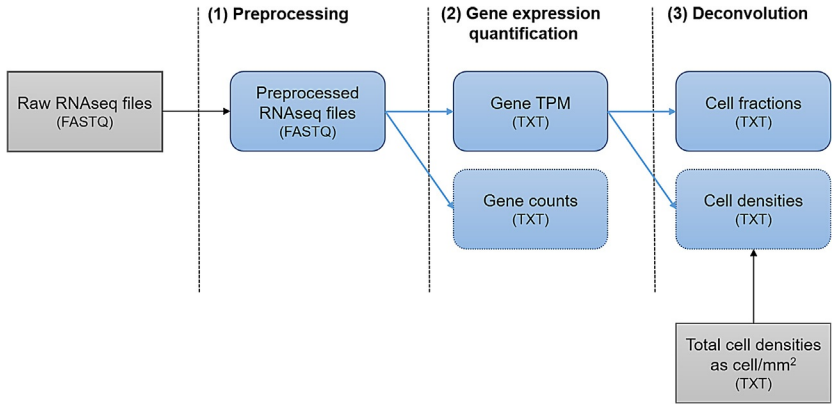

Fig. 1 The quanTIseq pipeline consists of three main modules. The pipeline can be started from any of these steps by specifying the parameter `--pipelinestart` ("preproc" to start with preprocessing, "expr" to start with quantifying the gene expression or "decon" for the deconvolution of cell fractions). Gene counts and cell densities can be obtained optionally by specifying the parameters accordingly.

To allow for flexibility in raw data preprocessing, different tools, such as Flexbar (Dodt, Roehr, Ahmed, & Dieterich, 2012) or CutAdapt (Martin, 2011), may be applied independently from the quanTIseq pipeline. If preprocessing was performed separately the pipeline can be started at the quantification step (see option `--pipelinestart`).

2. *Gene expression quantification*: Kallisto (Bray, Pimentel, Melsted, & Pachter, 2016) is used to quantify gene expression as transcripts per millions (TPM) and (optionally) as raw counts.
3. *Deconvolution*: this module performs expression normalization, gene reannotation, and deconvolution of cell fractions using constrained least squares regression. In addition, total cell densities per tumor can be used to optionally scale the cell fractions estimated by quanTIseq (see below).

The quanTIseq deconvolution algorithm models the expression of a bulk tumor mixture **M** as a linear combination of the expression of genes in different cell types, whose expression profiles are summarized in a signature matrix **S**, weighted by the relative cell fractions **f** (Fig. 2).

Given an input mixture M and a reference matrix S, quanTIseq estimates the unknown cell fractions **f** using least square regression to solve the following system of linear equations:

$$\boldsymbol{M} = \boldsymbol{S} x \boldsymbol{f}$$

Constrained least square regression is used to force the estimated cell fractions to be nonnegative and their sum to be lower of equal to one. As the

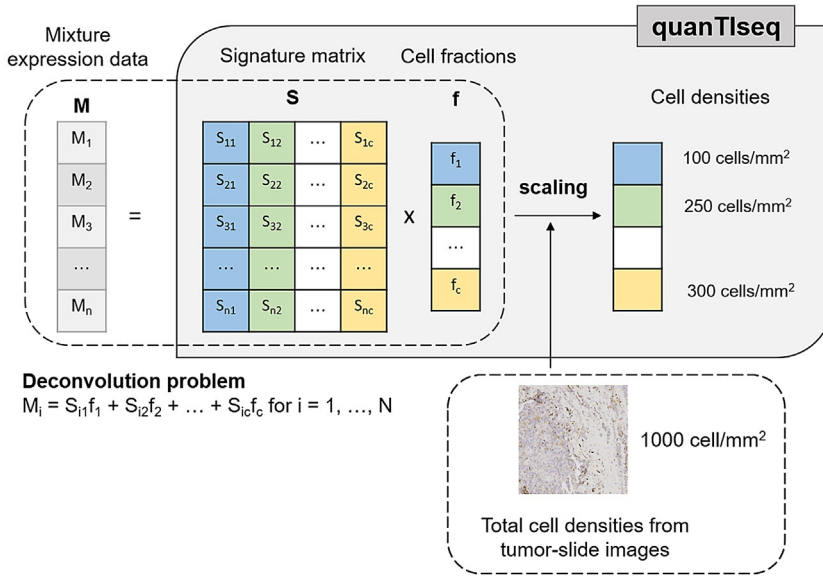

Fig. 2 Illustration of the deconvolution problem and of quanTIseq deconvolution approach. quanTIseq addresses the deconvolution problem by solving a system of linear equations modeling the expression of the mixture as the linear combination of the expression profiles of single immune cell types, weighted by their (unknown) fractions. Optionally the obtained cell fractions can be scaled to cells/mm^2 by using the total cell densities from tumor-slide images.

sum can be lower than one, the residual fraction represent the proportion of "other" uncharacterized cells, namely, cells that are present in a sample but are not represented in the signature matrix (e.g., cancer cells, Table 1). By quantifying the fraction of "other" cells, quanTIseq refers all immune cell fractions to the total cellular content of a sample, thereby allowing intra- and intersample comparison.

quanTIseq uses the "TIL10" signature matrix consisting of 153 genes for RNA-seq data and 170 genes for microarray data sets as they often lack some signature genes (see option `--rmgenes` below).

This signature matrix was built from a compendium of 51 RNA-seq data sets from 10 different immune cell types: B cells, CD4$^+$ and CD8$^+$ T cells, dendritic cells, M1 and M2 macrophages, monocytes, natural killer cells, neutrophils and T_{reg} cells (Finotello et al., 2019).

In case a dataset contains genes with aberrant high expression that would bias the results, it is possible to specify a customized file containing the name of genes which should be removed from the TIL10 signature matrix (see option `--rmgenes` below).

As noted above, quanTIseq can scale the estimated cell fractions to cell densities (e.g., cells/mm^2) by considering the density of total cells estimated from histological images. If this information is of interest and enough tissue material is available, a slice from the same tissue block consecutive to the slice that was used for RNA-seq may be used to perform hematoxylin and eosin (H&E) or immunofluorescence (IF) staining and manual or computational determination of total cells per area. For instance, IHCount (https://github.com/mui-icbi/IHCount), a publicly available semiautomated pipeline that was developed in conjunction with quanTIseq, can be used to obtain the overall cell densities as cells/mm^2 from H&E or IHC stained tumor tissue slides. IHcount can automatically preprocess whole-slide scans to create smaller image tiles and use a small subset of those to train a pixel classifier for nuclei detection with Ilastik (Sommer, Straehle, Köthe, & Hamprecht, 2011). This classifier is then applied to all tiles of the whole slide and the resulting probability maps together with the original tiles are used as input for a cellprofiler (Carpenter et al., 2006) pipeline. This cellprofiler pipeline, then, automatically segments the nuclei generates the count tables that may be used with quanTIseq.

3. Materials

3.1 Input files

QuanTIseq processes FASTQ files of raw RNA-seq reads when run from step 1 and 2, or a file containing gene expression levels in TPM units, when run from step 3 (Fig. 1).

For step 1 (`--pipelinestart=preproc`) and step 2 (`--pipelinestart=expr`), the `--inputfile` parameter must indicate the path and name of a tab-delimited text file containing information about the FASTQ files to be analyzed. The file consists of three columns without header (Table 2). For each sample, the first column specifies the sample identifier, the second column the path to the first FASTQ file, and the third column the path to the second FASTQ file, in case of paired-end samples. For single-end sequencing data, the third column must contain the string "None."

When starting the pipeline at step 3 (`--pipelinestart=decon`) the `--inputfile` parameter should specify the path and name of the tab-delimited text file with gene TPM (or microarray expression values) for all samples to be deconvoluted. The first column in this file should contain the gene symbols and the first row the sample identifiers (Table 3). The expression data should be on a nonlogarithmic scale.

Table 2 Input file format for the quanTIseq pipeline run on FASTQ data.

Sample 1	Input/rnaseq_sample1_1.fastq	Input/rnaseq_sample1_2.fastq
Sample 2	Input/rnaseq_sample2.fastq	None
…	…	…

Table 3 Input file format for the quanTIseq deconvolution module run on preprocessed expression data.

Gene	Sample 1	Sample 2	…
UBE2Q2P2	0	0	…
GTPBP6	75.06235	76.78617	…
…	…	…	…

Table 4 Input file format for the number of total cells from tumor images.

Sample 1	10767.0926258729
Sample 2	6871.40489414807
…	…

To also calculate densities of the deconvoluted cell types, the `--totalcell` option must be used to indicate the path and name of the file containing the number of total cells per mm^2 estimated from tumor-slide images of the samples under investigation. This file should contain the sample identifiers in the first column and the number of total cells per mm^2 in the second column (Table 4). The sample identifiers should match those of the input file.

Example files and other resources can be downloaded from the quanTIseq website: http://icbi.at/quantiseq.

3.2 System requirements

3.2.1 Operating system

The quanTIseq pipeline is available for Mac OS X Sierra or newer and for current Linux systems like CentOS 7, RedHat Linux 7, Fedora 28, Ubuntu 16.04 or later.

3.2.2 Software

To avoid software and library dependency issues and facilitate seamless integration into high performance computing environments, quanTIseq is containerized in Docker for Mac OS X or Singularity for Linux operating

systems. Depending on the operating system in use, either Docker or Singularity is required to run quanTIseq. We also recommend installing the R programming language for further analysis of the data as shown in our examples below.

3.2.3 Hardware

About 4 GB of free disk space is required to download the container image and extra disk space of about twice the data size of the input files should be available for the analyses. Additionally, a minimum of 8 GB RAM is needed to run the pipeline.

Mac hardware must be a 2010 or newer model and you might need to increase the amount of memory and CPUs (in case multithreading is considered to be used) in the MAC OS X Docker settings (Settings > Preferences > Advanced) according to the requirements above depending on the Docker version you are using.

4. Methods

4.1 Installation

quanTIseq is freely available and embedded in a Singularity image in case of Linux systems or in a Docker image in case of Mac OS X. The two-step installation procedure is explained in the following.

Mac OS X (based on Docker)

1. Install Docker (Instructions can be found on the Docker website: https://docs.docker.com/install/).
2. Download the "`quanTIseq_pipeline.sh`" script from the quanTIseq website: http://icbi.at/quantiseq

Linux (based on Singularity)

1. Install Singularity (Instructions can be found on the Singularity website: https://www.sylabs.io/docs/).
2. Download the "`quanTIseq_pipeline.sh`" script from the quanTIseq website: http://icbi.at/quantiseq.

The Docker/Singularity image includes all tools and dependencies that are required to run the analysis pipeline and is downloaded automatically the first time that the "`quanTIseq_pipeline.sh`" script is run.

4.2 Running quanTIseq

To run the quanTIseq pipeline, execute the following command:

- `bash quanTIseq_pipeline.sh --inputfile=path/to/input_file.txt \`
  `--outputdir=path/to/outputdirectory [options]`

Note that `--inputfile` and `--outputdir` are mandatory parameters.

The quanTIseq pipeline will automatically select Docker or Singularity depending on your operating system. If you are running the quanTIseq pipeline for the first time on Linux, quanTIseq saves the Singularity image file in the current directory ("quantiseq2.img"). All result files are saved in the directory specified by `--outputdir`.

4.3 Options

- *help*: prints instructions about how to run quanTIseq.
- *pipelinestart*: step from which the pipeline should be started as shown in Fig. 1: (1) "preproc," (2) "expr," or (3) "decon." Default: "preproc."
- If the option "decon" is set, the input file must be a tab-delimited text file with the gene TPMs (see Section 3).
- *tumor*: specifies whether expression data are from tumor or other samples. If this option is set to TRUE, signature genes with high expression in tumor samples are removed. Default: FALSE.
- We highly recommend setting "`--tumor=TRUE`" when analyzing tumor data. Very low fractions of other cells are estimated in tumor data using the "lsei" method can indicate a wrong setting of this parameter.
- *arrays*: specifies whether expression data are from microarrays (instead of RNA-seq). If TRUE, the "`--rmgenes`" parameter is automatically set to "none." Default: FALSE.
- *method*: deconvolution method to be used: "hampel," "huber," or "bisquare" for robust regression with Huber, Hampel, or Tukey bisquare estimators, respectively, or "lsei" for constrained least squares regression. The fraction of uncharacterized cells ("other") is computed only by the "lsei" method, which estimates cell fractions referred to the total cells in the sample under investigation, allowing intra- and inter-sample comparison. For all the other methods, the cell fractions are referred to immune cells considered in the signature matrix. Default: "lsei."
- All analyses and benchmarkings presented in the quanTIseq publication (Finotello et al., 2019) were run with the "lsei" algorithm.
- *mRNAscale*: specifies whether cell fractions must be scaled to account for cell-type-specific mRNA content. Default: TRUE.

- *totalcells*: path to a tab-separated text file containing the total cell densities (e.g., cells/mm^2) estimated from images of histological slides (e.g., H&E or IHC). This information will be used to scale the deconvoluted cell fractions, for the 10 immune cells and "other" cells, to cell densities (e.g., cells/mm^2). This parameter is optional and, if not set, only the cell fractions are returned.
- *rmgenes*: specifies which genes must be removed from the signature matrix before running deconvolution. This parameter can be equal to:
  - "none": no genes are removed
  - "default": A set of 17 genes with variable expression in RNA-seq data is removed from the TIL10 signature matrix (Finotello et al., 2019): *CD36*, *CSTA*, *NRGN*, *C5AR2*, *CEP19*, *CYP4F3*, *DOCK5*, *HAL*, *LRRK2*, *LY96*, *NINJ2*, *PPP1R3B*, *TECPR2*, *TLR1*, *TLR4*, *TMEM154* and *CD248*.
  - A path to a text file containing a list of genes to be removed. The text file must contain one gene symbol per line. This option can be used to specify noisy genes that might bias deconvolution results (e.g., immune genes with very high expression in the samples of interest).
  - Default: "default." (but set to "none" when **--arrays = TRUE**).
- *rawcounts*: specifies whether a file containing raw read counts per gene should be generated in addition to TPM. This file can be used for downstream analyses with third-party tools, like differential gene expression (e.g. with the R-packages edgeR (Robinson, McCarthy, & Smyth, 2010) or DESeq2 (Love, Huber, & Anders, 2014)). Default: FALSE.
- *prefix*: prefix of the output files. Default:"quanTIseq."
- *threads*: number of threads to be used. Default: 1.

  Kallisto results (gene counts and TPM) for paired end reads can differ slightly when using more than one thread. The estimate for the fragment length is computed from about 10,000 reads, which reads are used becomes nondeterministic in multithreading mode.
- *phred*: encoding of the RNA-seq quality scores for read preprocessing with Trimmomatic: "33" for Phred-33 or "64" for Phred-64. Default: 33.
- *adapterSeed*: maximum number of seed mismatches for the identification of adapter sequences by Trimmomatic. Default: 2.
- *palindromeClip*: threshold for palindrome clipping of adapter sequences by Trimmomatic. Default: 30.
- *simpleClip*: threshold for simple clipping of adapter sequences by Trimmomatic. Default: 10.

- *trimLead*: minimum Phred quality required by Trimmomatic to keep a base at the start of a read. Bases with lower quality are trimmed. Default: 20.
- *trimTrail*: minimum Phred quality required by Trimmomatic to keep a base at the end of a read. Bases with lower quality are trimmed. Default: 20.
- *minLen*: minimum read length required by Trimmomatic to keep a read. Reads shorter than this threshold are discarded. Default: 36.
- *crop*: maximum read length required by Trimmomatic. Longer reads are trimmed to this maximum length by removing bases at the end of the read. Default: 10000.
- *avgFragLen*: estimated average fragment length required by Kallisto for single-end data. Default: 50.
- *sdFragLen*: estimated standard deviation of fragment length required by Kallisto for single-end data. Default: 20.

4.4 Result files

Once the analysis is complete the output files will be stored in the specified output directory. According to the options specified, up to three text files will be created:

- *prefix_cell_fractions.txt*: a tab-delimited text file containing the cell fractions estimated by quanTIseq deconvolution module. The first column lists the sample identifiers and the header shows the cell types. The fraction of uncharacterized cells is reported only when `--method=lsei` option is specified.
- *prefix_gene_tpm.txt*: a tab-delimited text file containing gene expression values in TPM. Gene symbols are listed in the first column and the header contains the sample identifiers.
- *prefix_gene_count.txt*: a tab-delimited text file containing gene expression values as raw gene counts. The format is the same as for the TPM file. This file is generated only when the `--rawcounts=TRUE` option is specified.

5. Examples

The quanTIseq pipeline cannot only be used for tumor samples but can also be applied on RNA-seq data derived from human blood. Here, we will demonstrate two different applications of quanTIseq. The first example shows how to estimate the immune-cell fractions from preprocessed RNA-seq data (in TPM format) generated from blood

samples. The second example demonstrates how to run the full pipeline from raw RNA-seq data from bulk-tumor samples collected from melanoma patients.

Besides calculating the immune cell fractions, we will also show how to perform simple evaluations of the results using the R programming language.

5.1 Example 1

In this example, we show how quanTIseq can be used to deconvolute blood-derived immune-cell mixtures and compare them with matching flow cytometry data.

The example data set is available online at the Gene Expression Omnibus (GEO: https://www.ncbi.nlm.nih.gov/geo/) with the accession number GSE107572. This data set contains preprocessed RNA-seq data from blood-derived immune-cell mixtures from nine healthy donors. Flow cytometry estimates for the according immune subpopulations, except for macrophages which are not present in blood, are also available for the same samples.

5.1.1 Step 1—Download RNA-seq data

Download the data file of preprocessed RNA-seq data from the "Supplementary files" section on GEO (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE107572): "`GSE107572_tpm_PBMC_RNAseq.txt.gz`."

This file contains the preprocessed RNA-seq data in TPM units which can be used to run the quanTIseq pipeline starting at the deconvolution step (`--pipelinestart=decon`).

5.1.2 Step 2—Run quanTIseq

To calculate individual cell fractions the quanTIseq pipeline should be run with the following parameter settings:

```
• bash quanTIseq_pipeline.sh --inputfile=GSE107572_tpm_PBMC_RNAseq.txt\
    --outputdir=Output --prefix=GSE107572 --pipelinestart=decon
```

This command generates a file named "`GSE107572_cell_fractions.txt`" in the folder named "`Output`." The parameter `--prefix`, which specifies the prefix of the output files, is optional and can be changed.

5.1.3 Step 3—Compare results with flow cytometry data

Using the following R script (named "`GSE107572.R`"), the estimated cell fractions from quanTIseq can be compared with the true cell fractions measured by flow cytometry. The script uses the Bioconductor package "GEOquery" to download the flow cytometry data from the Gene

Expression Omnibus database (GEO), generates scatterplots and calculates Pearson's correlation coefficients between flow cytometry and quanTIseq deconvolution cell fraction estimates.

```
# Access flow cytometry cell fractions from GEO

args <- commandArgs(TRUE)
dir <- args[1]

library(GEOquery)

GEOid<-"GSE107572"
gds<-getGEO(GEOid)
GEOinfo<-pData(gds[[1]])

FACSdata<-data.frame(B.cells=GEOinfo$`b cells:ch1`,
  T.cells.CD4=GEOinfo$`cd4+ t cells:ch1`,
  T.cells.CD8=GEOinfo$`cd8+ t cells:ch1`,
  Monocytes=GEOinfo$`monocytes:ch1`,
  Dendritic.cells=GEOinfo$`myeloid dendritic cells:ch1`,
  NK.cells=GEOinfo$`natural killer cells:ch1`,
  Neutrophils=GEOinfo$`neutrophils:ch1`,
  Tregs=GEOinfo$`tregs:ch1`)
rownames(FACSdata)<-gsub("Blood-derived immune-cell mixture from
donor ", "pbmc", GEOinfo$title)

# Load deconvolution data obtained by running quanTIseq
(--pipelinestart=decon)

DCdata<-read.table(paste0(dir,"/GSE107572_cell_fractions.txt"),
header=TRUE, sep="\t", row.names=1)
rownames(DCdata)<-gsub("_.*$", "", sub("_", "", rownames(DCdata)))

ccells<-intersect(colnames(DCdata),colnames(FACSdata))
csbjs<-intersect(rownames(DCdata),rownames(FACSdata))
DCdata<-DCdata[csbjs,ccells]
FACSdata<-FACSdata[csbjs,ccells]

# Compare cell fractions for single cell types and all cell types
together

palette<-c("#451C87", "#b3b300", "#CE0648", "#2363C5", "#AB4CA1",
"#0A839B", "#DD8C24", "#ED6D42")
names(palette)<-c("T.cells.CD4", "Dendritic.cells", "Monocytes",
"T.cells.CD8", "Tregs", "B.cells", "NK.cells", "Neutrophils")

par(mfrow=c(3,3))
colall<-c()
```

```
for (i in 1:(ncol(DCdata)+1)) {

  if (i<=ncol(DCdata)) {

    x<-as.numeric(as.character(FACSdata[,i]))
    y<-DCdata[,i]
    ccell<-colnames(DCdata)[i]
    col<-palette[ccell]

} else {

    x<-as.numeric(as.vector(as.matrix(FACSdata)))
    y<-as.vector(as.matrix(DCdata))
    ccell<-"All cells"
    col<-colall
}

  res.cor<-cor.test(y,x)
  R<-round(res.cor$estimate, digits=2)
  p<-format.pval(res.cor$p.value, digits=2)
  RMSE<-round(sqrt(mean((y-x)^2, na.rm=TRUE)), digits=2)
  regl<-lm(y~x)

  ymax<-max(round(max(y),digits=2)*1.3,0.01)
  xmax<-max(round(max(x),digits=2),0.01)

  pdf("GSE107572_correlation.pdf")
  plot(x, y,
       main=gsub("(\\.)", " ", ccell),
       pch=19,
       xlab="Flow cytometry cell fractions",
       ylab="quanTIseq cell fractions",
       col=col,
       cex.main=1.3,
       ylim=c(0,ymax),
       xlim=c(0,xmax),
       las=1)
  abline(regl)

  text(0, ymax*0.98, cex=1, paste0("r = ", R, ", p = ", p), pos=4)
  text(0, ymax*0.9, cex=1, paste0("RMSE = ", RMSE), pos=4)

  colall<-c(colall, rep(col, length(x)))

dev.off()

}
```

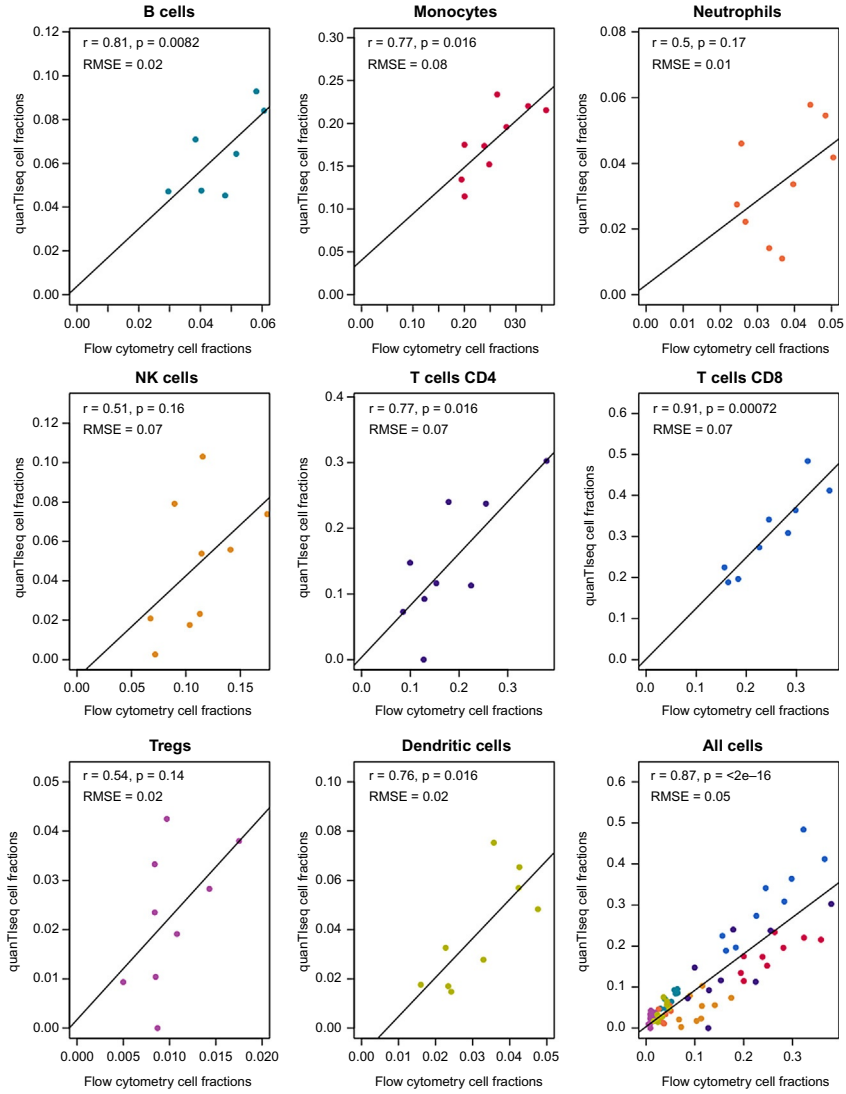

Fig. 3 Correlation plots comparing the estimated cell fractions from quanTIseq with cell fractions computed by flow cytometry for eight different immune cell types.

Run the R script with following command and specify the path to the quanTIseq output directory (e.g., "`Output`" from Step 2) as argument:

- `Rscript GSE107572.R path/to/directory`

The results of this analytical step are shown in Fig. 3. The overall Pearson's correlation coefficient for the different immune cell types estimated by quanTIseq is 0.87. The highest correlation ($r = 0.91$) is obtained for $CD8^+$ T cells.

The same type of analysis can be run on other datasets where RNA-seq data and matched ground-truth data (e.g., from immunohistochemistry or Coulter counter) are available.

5.2 Example 2

The second example analyzes gene expression data from seven melanoma patients who have been treated with kinase inhibitors. The immune cell fractions before and on-treatment with these inhibitors are then assessed and compared.

We first show how to download the publicly available RNAseq data from GEO as SRA files and convert them to FASTQ files. Then we show how to run the full quanTIseq pipeline to quantify the immune cell fractions. The list of samples is shown in Table 5.

Table 5 List of patients and kinase inhibitors.

GEO accession	Patient	Sample type	Inhibitor
GSM1949045	Pt1 baseline	None	None
GSM1949046	Pt1 D85	On-treatment	BRAFi
GSM1949047	Pt2 baseline	None	None
GSM1949048	Pt2 D726	On-treatment	BRAFi
GSM1949049	Pt3 baseline	None	None
GSM1949050	Pt3 D22	On-treatment	BRAFi + MEKi
GSM1949051	Pt4 baseline	None	None
GSM1949052	Pt4 D15	On-treatment	BRAFi + MEKi
GSM1949053	Pt4 RD261	On-treatment	BRAFi + MEKi
GSM1949054	Pt6 baseline	None	None
GSM1949055	Pt6 D6	On-treatment	BRAFi
GSM1949056	Pt6 D12	On-treatment	BRAFi + MEKi
GSM1949057	Pt6 D15	On-treatment	BRAFi + MEKi
GSM1949058	Pt7 baseline	None	None
GSM1949059	Pt7 D15A	On-treatment	MEKi
GSM1949060	Pt7 D15B	On-treatment	MEKi
GSM1949061	Pt7 D15C	On-treatment	MEKi
GSM1949062	Pt8 baseline	None	None
GSM1949063	Pt8 D22	On-treatment	BRAFi + MEKi

5.2.1 Step 1—Get list of files

The RNA-seq data from the seven patients is available on GEO with the accession number GSE75299 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75299). Open the link in a web browser and in the "Relations" section, follow the link to the sequence read archive (SRA) study with accession number SRP066571. Select the first entry of the resulting SRA page results and select "All runs" in the "Study" section (see Fig. 4).

You will be forwarded to a page with a table showing all runs and samples. All runs with source name "tumor biopsy" should be selected and the "Accession List" table saved as "`SRP066571_Acc_List.txt`". The file now contains the selected "Run" identifiers of the files to be downloaded. In total there should be 19 different identifiers (see Fig. 5).

Additionally, the "RunInfo Table" should also be saved as "`SraRunTable.txt`" which is needed for the annotation in a later step (Fig. 5).

5.2.2 Step 2—Download FASTQ files

There are different ways to download the SRA files. In this example we are using the **prefetch** program from the SRA Toolkit (https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/) together with the Aspera Connect Client (https://downloads.asperasoft.com/connect2/), for which three files are required, the ascp executable binary (ascp), the ascp ssh key (asperaweb_id_dsa.openssh) and the file containing your accession numbers from step 1. To download the SRA data files, locate these files on your system and run following bash command:

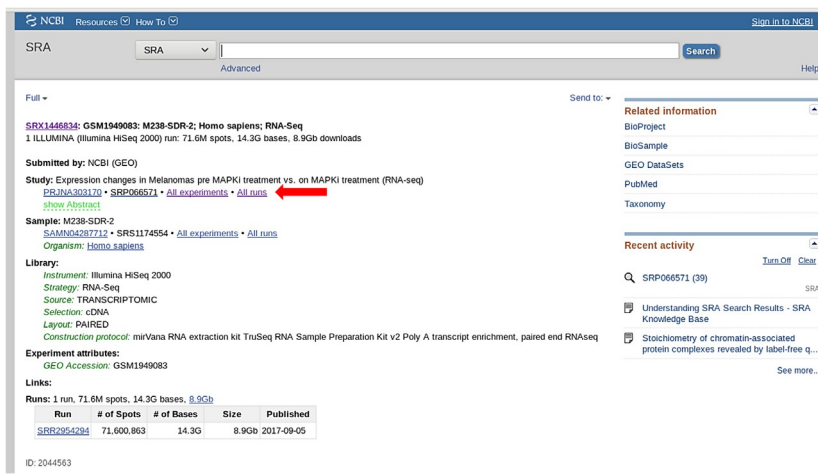

Fig. 4 Screenshot NCBI website to choose "All runs."

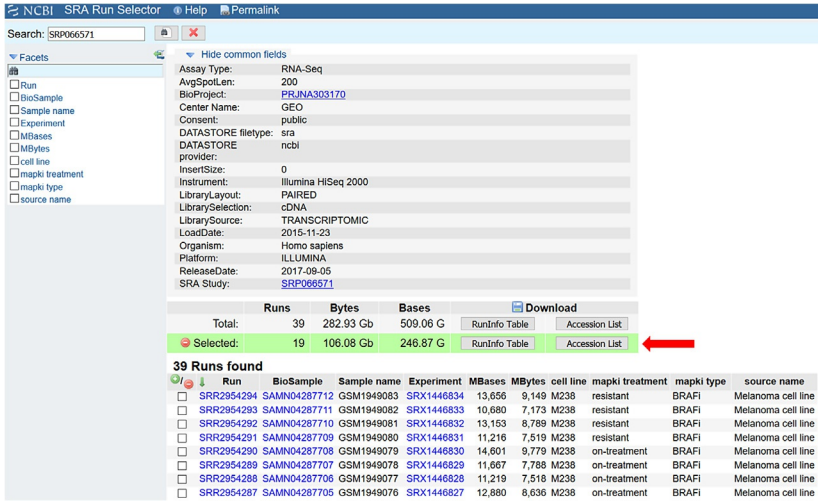

Fig. 5 Screenshot NCBI website to download "Accession List" and "RunInfo Table."

- `prefetch --ascp-path "path_to_aspera/connect/bin/ascp|path_to_aspera/connect/etc/asperaweb_id_dsa.openssh" -v --option-file SRP066571_Acc_List.txt`

After completion, the SRA files are available locally on the computer in the default folder named "`ncbi`." The **fasterq-dump** tool from the SRA Toolkit can then be used to convert the downloaded SRA files to FASTQ files. Since the sequencing data was generated by paired-end sequencing (see study run info), the reads need to be split into forward and reverse read files by using the `--split-files` option.

Extract and split all paired-end RNA-seq reads from the SRA files with the following bash command:

- `for s in` ` `ls -1 ~/path_to_files/ncbi/sra/*.sra` ` `; do fasterq-dump $s -p --split-files -O ~/path_to_output/directory; done`

To run this example, you need at least 1.7 TB of free disk space.

5.2.3 Step 3—Run quanTIseq pipeline

To run the full quanTIseq pipeline a tab-delimited text file containing the paths to the FASTQ files to deconvolute (see parameter `--inputfile`) needs to be created. Table 6 shows the input file where the first column specifies

Table 6 SRP066571 input file.

SRR2954274	SRR2954274.sra_1.fastq	SRR2954274.sra_2.fastq
SRR2954273	SRR2954273.sra_1.fastq	SRR2954273.sra_2.fastq
SRR2954272	SRR2954272.sra_1.fastq	SRR2954272.sra_2.fastq
SRR2954271	SRR2954271.sra_1.fastq	SRR2954271.sra_2.fastq
SRR2954270	SRR2954270.sra_1.fastq	SRR2954270.sra_2.fastq
SRR2954269	SRR2954269.sra_1.fastq	SRR2954269.sra_2.fastq
SRR2954268	SRR2954268.sra_1.fastq	SRR2954268.sra_2.fastq
SRR2954267	SRR2954267.sra_1.fastq	SRR2954267.sra_2.fastq
SRR2954266	SRR2954266.sra_1.fastq	SRR2954266.sra_2.fastq
SRR2954265	SRR2954265.sra_1.fastq	SRR2954265.sra_2.fastq
SRR2954264	SRR2954264.sra_1.fastq	SRR2954264.sra_2.fastq
SRR2954263	SRR2954263.sra_1.fastq	SRR2954263.sra_2.fastq
SRR2954262	SRR2954262.sra_1.fastq	SRR2954262.sra_2.fastq
SRR2954261	SRR2954261.sra_1.fastq	SRR2954261.sra_2.fastq
SRR2954260	SRR2954260.sra_1.fastq	SRR2954260.sra_2.fastq
SRR2954259	SRR2954259.sra_1.fastq	SRR2954259.sra_2.fastq
SRR2954258	SRR2954258.sra_1.fastq	SRR2954258.sra_2.fastq
SRR2954257	SRR2954257.sra_1.fastq	SRR2954257.sra_2.fastq
SRR2954256	SRR2954256.sra_1.fastq	SRR2954256.sra_2.fastq

the sample names and the second and third column contain the paths to the FASTQ files from the paired-end sequencing runs (forward and reverse reads, respectively). In this example the file names are listed without file directory, therefore the pipeline should be run in the same directory where the FASTQ files are located. Alternatively, full file paths can be provided.

This input file should be saved as "`SRP066571_rnaSeqInfoFile.txt`" in the same folder where also the FASTQ files, extracted with fasterq-dump in step 2, are located.

Next, the quanTIseq pipeline can be run with the following command:

- `bash quanTIseq_pipeline.sh --inputfile=SRP066571_rnaSeqInfoFile.txt\`
  `--outputdir=Output --prefix=SRP066571 --tumor=TRUE --threads=8`

In this example, data from melanoma patients are analyzed, therefore the parameter `--tumor=TRUE` needs to be specified, which will remove a set of immune-cell genes with aberrant expression in tumors from the signature matrix.

This command will create two result files, which are located in a folder named "`Output`":

- `SRP066571_cell_fractions.txt`
- `SRP066571_gene_tpm.txt`

The first file contains the cell fractions estimated by quanTIseq, which may be visualized using Excel of R to create stacked bar plots as those shown in Fig. 6. The individual fractions sum up to 1 and reflect their proportion with respect to the total cells in the sample. The second file reports the sample gene expression values in TPM format.

5.2.4 Step 4—Analysis/results

In order to compare the immune cell fractions from patients before and on-treatment with kinase inhibitors, the FASTQ files need to be matched with patient data. An example R script, which is using the Bioconductor package "GEOquery" to download the respective information from the

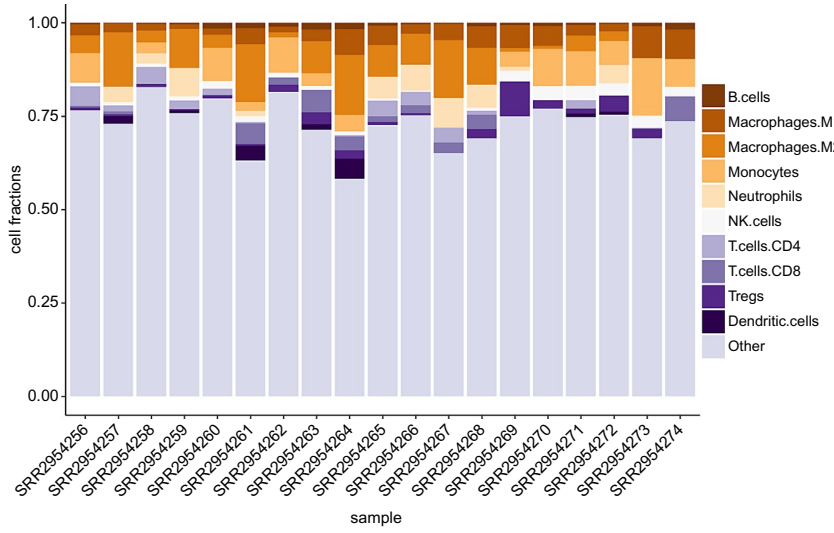

Fig. 6 Estimated immune cell fractions for all samples.

GEO database, is listed below. This R script (named "GSE75299.R") may be used to compare and statistically evaluate the pre- and on-treatment immune cell fractions. It generates boxplots of the cell proportions, uses Wilcoxon test to calculate the *p*-values, and can be run by using the following command:

- `Rscript GSE75299.R path/to/SRP066571_cell_fractions.txt \ path/to/SraRunTable.txt`

where the first argument is the file of cell type fractions obtained from step 2 and the second argument is the SRA run table info file created in step 1.

```
library(GEOquery)
library(ggplot2)
library(reshape2)
library(ggpubr)

args <- commandArgs(TRUE)
path <- args[1]
sra <- args[2]

### Load file with estimated cell fractions by quanTIseq:
cdata  <-  read.csv(paste0(path,"/SRP066571_cell_fractions.txt"),
header=TRUE, sep="\t", stringsAsFactors = FALSE, row.names=1)

### Annotation
gds<-getGEO("GSE75299")
pdata<-pData(gds[[1]])
sraInfo <- read.csv(paste0(sra,"/SraRunTable.txt"), header=TRUE,
sep="\t", stringsAsFactors = FALSE)

cdata <- cbind(cdata, sraInfo[, "Sample_Name"][match(rownames
(cdata), sraInfo$Run)])
colnames(cdata)[length(cdata)] <- "geo_accession"

pdata<-pdata[grep("Pt", pdata[,1]),]
cdata <- cbind(cdata, pdata[, "title"][match(cdata$geo_accession,
pdata$geo_accession)])
rownames(cdata) <- cdata[,length(cdata)]

### Remove other cells and the last two columns
cdata<-cdata[,-grep("Other", colnames(cdata))]
cdata<-cdata[,-grep("geo_accession", colnames(cdata))]
```

```
### Transpose
cdata<-t(cdata)

### Boxplots and wilcoxon's test pre/post
groups<-rep("POST", ncol(cdata))
groups[grep("baseline", colnames(cdata))]<-"PRE"
groups<-factor(groups, levels=c("PRE", "POST"))
names(groups)<-colnames(cdata)

dataset <- cdata
colnames(dataset) <- as.vector(groups)
dataset <- melt(dataset)

pdf("GSE75299_qTIs_boxplot.pdf")
ggplot(dataset, aes(Var1, value, fill=Var2))+
  geom_boxplot()+
  ylab("cell fractions") +
  xlab("")+
  theme(axis.text=element_text(size=17),
        axis.title=element_text(size=19))+
  theme(legend.text=element_text(size=17),
        legend.title=element_text(size=19))+
  scale_fill_discrete(name="Treatment")+
  theme(axis.text.x= element_text(angle=45, hjust = 1 ))+
  stat_compare_means(aes(group=Var2),
        symnum.args=list(cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
        symbols = c("****", "***", "**", "*", "ns")),
        label ="p.signif",
        hide.ns=TRUE, cex=10) #default: Wilcoxon Test
dev.off()
```

The resulting boxplots illustrate the cell-type fraction differences between pre and on-treatment with the kinase inhibitors (PRE vs. POST) for all analyzed samples. For this analytical step all samples were combined. Fig. 7 shows detection of significant increase in M2 macrophage and dendritic cells fractions (Wilcoxon test, P value < 0.1) after kinase treatment. All other immune cell types quantified by quanTIseq did not show any significant differences between on-treatment and before treatment samples.

Further analytical steps can be performed with the estimated immune cell fractions, for instance a comparison could be done between the single patients, to show how the immune infiltrates before and on-treatment differs between patients.

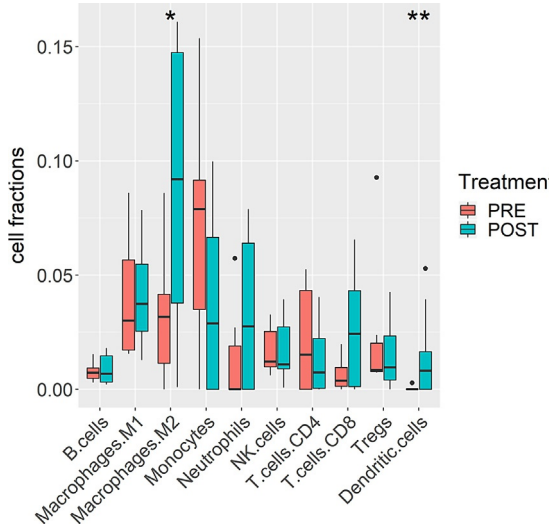

Fig. 7 Boxplots of the estimated cell fractions (pre vs. on treatment, $*P$ value < 0.1, $**P$ value < 0.05).

6. Concluding remarks

quanTIseq is a computational pipeline, developed for the analysis of bulk RNA-seq data that quantifies the fractions and densities of 10 different immune cell types relevant for cancer immunology. quanTIseq is specifically designed for and validated on RNA-seq data, allows full analysis of raw RNA-seq data, from pre-processing to cell fraction deconvolution, and can be used for either tumor or blood samples by simply setting the parameter `--tumor=TRUE/FALSE`.

The containerized pipeline of quanTIseq can estimate immune-cell fractions in a standardized manner across different platforms, enabling reproducible analysis of immune-cell fractions and densities. Hence, it is a useful tool to investigate the immunological effects of cancer drugs (Galluzzi, Buqué, Kepp, Zitvogel, & Kroemer, 2015) that can provide mechanistic insights for the design of combination therapies.

quanTIseq was primarily developed to estimate immune cell fractions relevant for cancer immunology, but it can be helpful also to investigate immune cell fractions and their modulation in other diseases, like autoimmunity, inflammation and infections.

Acknowledgments

C.P. is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Division for Bioinformatics. F.F. was supported by the Austrian Cancer Aid/Tyrol (project No. 17003, "quanTIseq: dissecting the immune contexture of human cancers") and by the Austrian Science Fund (FWF) (project No. T 974-B30).

References

Balkwill, F. R., Capasso, M., & Hagemann, T. (2012). The tumor microenvironment at a glance. *Journal of Cell Science*, *125*(23), 5591–5596. https://doi.org/10.1242/jcs.116392.

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*, *30*(15), 2114–2120. https://doi.org/10.1093/bioinformatics/btu170.

Bray, N. L., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, *34*(5), 525–527. https://doi.org/10.1038/nbt.3519.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., et al. (2006). Cell profiler: Image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10), R100. https://doi.org/10.1186/gb-2006-7-10-r100.

Chen, D. S., & Mellman, I. (2017). Elements of cancer immunity and the cancer–immune set point. *Nature*, *541*(7637), 321–330. https://doi.org/10.1038/nature21349.

Dodt, M., Roehr, J. T., Ahmed, R., & Dieterich, C. (2012). FLEXBAR—Flexible barcode and adapter processing for next-generation sequencing platforms. *Biology*, *1*(3), 895–905. https://doi.org/10.3390/biology1030895.

Finotello, F., & Eduati, F. (2018). Multi-omics profiling of the tumor microenvironment: Paving the way to precision immuno-oncology. *Frontiers in Oncology*, *8*, 430. https://doi.org/10.3389/fonc.2018.00430.

Finotello, F., Mayer, C., Plattner, C., Laschober, G., Rieder, D., Hackl, H., et al. (2019). Molecular and pharmacological modulators of the tumor immune contexture revealed by deconvolution of RNA-seq data. *Genome Medicine*, *11*(1), 34. https://doi.org/10.1186/s13073-019-0638-6.

Finotello, F., & Trajanoski, Z. (2018). Quantifying tumor-infiltrating immune cells from transcriptomics data. *Cancer Immunology, Immunotherapy*, *67*(7), 1031–1040. https://doi.org/10.1007/s00262-018-2150-z.

Fridman, W. H., Pagès, F., Sautès-Fridman, C., & Galon, J. (2012). The immune contexture in human tumours: Impact on clinical outcome. *Nature Reviews. Cancer*, *12*(4), 298–306. https://doi.org/10.1038/nrc3245.

Galluzzi, L., Buqué, A., Kepp, O., Zitvogel, L., & Kroemer, G. (2015). Immunological effects of conventional chemotherapy and targeted anticancer agents. *Cancer Cell*, *28*(6), 690–714. https://doi.org/10.1016/j.ccell.2015.10.012.

Galon, J., Costes, A., Sanchez-Cabo, F., Kirilovsky, A., Mlecnik, B., Lagorce-Pagès, C., et al. (2006). Type, density, and location of immune cells within human colorectal tumors predict clinical outcome. *Science*, *313*(5795), 1960–1964. https://doi.org/10.1126/science.1129139.

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12). https://doi.org/10.1186/s13059-014-0550-8.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. Journal*, *17*(1), 10–12. https://doi.org/10.14806/ej.17.1.200.

Newman, A. M., Liu, C. L., Green, M. R., Gentles, A. J., Feng, W., Xu, Y., et al. (2015). Robust enumeration of cell subsets from tissue expression profiles. *Nature Methods*, *12*(5), 453–457. https://doi.org/10.1038/nmeth.3337.

Racle, J., de Jonge, K., Baumgaertner, P., Speiser, D. E., & Gfeller, D. (2017). Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *eLife, 6*, e26476. https://doi.org/10.7554/eLife.26476.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616.

Sommer, C., Straehle, C., Köthe, U., & Hamprecht, F. A. (2011). Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE international symposium on biomedical imaging: From nano to macro* (pp. 230–233). https://doi.org/10.1109/ISBI.2011.5872394.