Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Robust cell subsets decomposition from tissue expression profiles for biomarker identification

Bc. Evžen Šírek

Supervisor: doc. Ing. Jiří Kléma, Ph.D.
Field of study: Open Informatics
Subfield: Bioinformatics
May 2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šírek** Jméno: **Evžen** Osobní číslo: **434672**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Bioinformatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Učení rozkladu komplexních tkání z expresních profilů a jeho využití při vyhledávání biomarkerů

Název diplomové práce anglicky:

Robust cell subsets decomposition from tissue expression profiles for biomarker identification

Pokyny pro vypracování:

1. Seznamte se s existujícími metodami dekonvoluce profilů genové exprese.
2. Proveďte rešerši výše uvedených metod, navrhněte základní parametry pro hodnocení výše uvedených metod, včetně dostupnosti.
3. Vybrané metody aplikujte na data dodaná vedoucím práce, porovnejte je.
4. Empiricky zhodnoťte použitenost dekonvoluce pro zkvalitnění vyhledávání biomarkerů v uvedených datech. Zaměřte se na množství nalezených markerů a jejich známou anotaci.

Seznam doporučené literatury:

[1] Newman, Aaron M., et al. "Robust enumeration of cell subsets from tissue expression profiles." Nature methods 12.5 (2015): 453.

[2] Schölkopf, B., Smola, A.J., Williamson, R.C. & Bartlett, P.L. New support vector algorithms. Neural Comput. 12, 1207–1245 (2000).

[3] Shen-Orr, S.S. & Gaujoux, R. Computational deconvolution: extracting cell type-specific information from heterogeneous samples.. Curr. Opin. Immunol. 25, 571–578 (2013).

[4] Abbas, A.R., Wolslegel, K., Seshasayee, D., Modrusan, Z. &amp; Clark, H.F. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. PLoS ONE 4, e6098 (2009).

[5] Gong, T. et al. Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples. PLoS ONE 6, e27156 (2011).

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Jiří Kléma, Ph.D., Intelligent Data Analysis FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **11.02.2020** Termín odevzdání diplomové práce: **22.05.2020**

Platnost zadání diplomové práce: **30.09.2021**

doc. Ing. Jiří Kléma, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.

Datum převzetí zadání

Podpis studenta

Acknowledgements

I would like to thank doc. ing. Jiří Kléma Ph.D. for the valuable comments and remarks he has given me during the creation of this master thesis.

I also want to thank my beloved Domi for her endless words of support and patience during the difficult times of writing this thesis.

Finally, I thank my family for supporting me during the studies.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 22. May 2020

Abstract

RNA sequencing (RNA-seq) is a widely used technology used for measuring the gene expression and consequently, for the differential gene expression analysis. The sequencing is usually performed on bulk mixture samples and is thus not able to reveal the cell type composition of the sample. It is, however, possible to infer this composition in silico from the measurements of bulk samples — the class of methods, performing this task, is commonly referred to as *gene expression profile deconvolution* methods.

We give a brief introduction to the RNA-seq technology and describe the basic statistical properties of the RNA-seq count data, mainly in the context of various normalization methods. We formalize the problem of deconvolution, perform research of deconvolution methods available in the literature, and compare them based on proposed metrics. We select 10 of these methods and apply them in 18 various setups to RNA-seq count data. The deconvolution results are then compared based on Pearson and Spearman correlations, revealing clusters of methods performing similarly.

We then introduce ways of incorporating these results into differential gene expression (DGE) analysis. We show that incorporating deconvolution into the DGE pipeline produces results different from DGE with no such information. Although the benefit of such differences could not be directly evaluated, this opens the door to future research of these differences on datasets with well-defined ground truth.

Keywords: RNA sequencing, deconvolution, gene expression profiles, differential gene expression, biomarkers

Supervisor: doc. Ing. Jiří Kléma, Ph.D.

Abstrakt

Sekvenování RNA je běžně používaná technologie. Často slouží zejména pro měření genové exprese, a následně pro analýzu diferenciální genové exprese. RNA sekvenování se většinou provádí na vzorcích z komplexních tkání, u kterých není známo buněčné složení. RNA sekvenování tak nedokáže rozpoznat rozdíly v genové expresi na úrovni buněčných typů. Existují však metody, zaměřené na rozložení naměřených dat z komplexních tkání do jednotlivých buněčných typů — nazývají se metody *dekonvoluce (rozkladu) expresních profilů*.

V této práci stručně představujeme technologii RNA sekvenování a popisujeme základní statistické vlastnosti jí produkovaných dat, zejména z pohlednu normalizace těchto dat. Dále popisujeme formalizaci problému dekonvoluce expresních profilů, představujeme rešerši dekonvolučních metod v literatuře a porovnáváme je z pohledu navrhnutých metrik. Následně jsme vybrali 10 těchto metod, a v 18 různých konfiguracích jsme je aplikovali na poskytnutá data genové exprese. Výsledky dekonvoluce porovnáváme na základě Pearsonovy a Spearmanovy korelace, což odhalilo skupiny metod, které produkovaly podobné výsledky.

Prezentujeme různé způsoby použití těchto výsledků v analýze DGE vedoucí k odlišným signifikantním biomarkerům. To dává podnět k budoucímu výzkumu a ověření přínosu těchto odlišností na cíleně připravených datasetech.

Klíčová slova: RNA sekvenování, dekonvoluce, expresní profily, diferenciální genová exprese, biomarkery

Překlad názvu: Učení rozkladu komplexních tkání z expresních profilů a jeho využití při vyhledávání biomarkerů

Contents

Figures

Tables

Acronyms

DE Diferentially Expressed.

DEG Diferentially Expressed Gene.

DGE Diferential Gene Expression.

DGEA Diferential Gene Expression analysis.

FPKM Fragments Per Kilobase of transcript per Million mapped reads.

GE Gene Expression.

GEP Gene Expression Profile.

GSEA Gene Set Enrichment Analysis.

LFC Logarithmic Fold Change.

LLSR Linear Least Squares Regression.

PBMC Peripheral Blood Mononuclear Cell.

QP Quadratic Programming.

RBCs Red Blood Cells.

RNA-seq RNA-sequencing.

RPKM Reads Per Kilobase of transcript per Million mapped reads.

scRNA-seq Single-cell RNA sequencing.

TPM Transcripts Per Million.

WBCs White Blood Cells.

Chapter 1

Introduction

RNA sequencing is a rapidly developing technology, which allows for the simultaneous sequencing of RNA present in cells or tissues. It provides a valuable insight into the transcriptome, and thanks to its continuously lowering prices, it is becoming more and more available. Data coming from the RNA sequencing are commonly used for differential gene expression analysis, for example, to identify biomarkers that could serve as a predictor of a condition, as an indicator of a successful treatment, or as in our case for treatment selection.

As most RNA sequencing experiments are run on bulk data of complex tissues, the underlying cell type composition is often unknown. Different cell types in the tissue can exhibit different gene expression profiles, which the bulk RNA sequencing cannot capture, as it measures only the average gene expression in the sample. This can lead to misleading results when interpreting changes in gene expression between conditions as caused by some biological mechanism — the true underlying reason for the change can be caused by change in cell type composition of the sample.

This is a problem, which is being solved by various techniques and methods of gene expression profile deconvolution. They are used for determining the proportions (or even expression profiles) of different cell types present in the bulk sample.

In this thesis, we explore the wide range of deconvolution methods, and we compare them according to their properties. They are compared with the respect to their applicability to RNA-seq data, which were provided by the thesis supervisor.

During the differential gene expression analysis of RNA-seq count data, complex models (e.g., using the generalized linear models) are used for the modeling of counts. These models allow for complex designs, which enables the incorporation of additional independent variables into the model. This is commonly used for control of various batch effects, which could cause undesirable false detection of differentially expressed genes.

We explore the possibility of applying the results of deconvolution of gene expression profiles to the differential expression analysis, with the desired outcome of eliminating the presence of significantly differentially expressed genes/biomarkers, whose change in expression was caused by the underlying change in cell type composition of the sample.

During this, the whole pipeline of processing the RNA-seq reads data was performed, and finally, several ways of incorporating the deconvolution results to differential gene expression analysis were evaluated.

Chapter 2
Background

In the following sections, we present definitions and explanations for terms and concepts needed for the understanding of gene expression profiles deconvolution. We start by introducing the molecular biology background, followed by the exploration of gene expression data and, finally, a formal definition of deconvolution in the context of gene expression profiles. It serves as a rather compact summary of concepts that one can frequently come across in bioinformatics articles and tools dealing with the differential gene expression and deconvolution. The following chapters are based mainly on three books, Molecular Cell Biology by Berk et al. [14], Molecular Biology of the Cell by Alberts et al. [3] and The Cell: A Molecular Approach by Cooper and Hausman [29]. Exceptions from this are explicitly marked in the text. Also please note, that as this thesis deals with data coming from the sequencing of human blood, the following sections thus address eukaryotes only; therefore, some of the statements do not generally apply to bacteria and archaea.

2.1 'Central dogma' of molecular biology

To better understand the origin of our data, let us first revisit the so-called 'Central dogma' of molecular biology. The dogma describes the flow of information from genetic material (DNA) to the resulting functional product. The process of transcription from DNA to RNA and subsequent translation of RNA to functional product is called **gene expression**. The functional product can be a protein, and RNA, which encodes such a protein is called the mRNA (messenger RNA). But the functional product can be the RNA itself. To name a few: tRNA (transfer RNA, serves as an adapter between amino acids and mRNA during protein synthesis), rRNA (ribosomal RNA, which form ribosome and participate in protein synthesis), snRNA (small nuclear RNA, playing a role in gene expression regulation) or lncRNA (long noncoding RNA, regulating various cell processes). The gene expression can be described (from a very high-level point of view) as a process consisting of two main steps in series, the transcription, and translation.

2.1.1 Transcription

Transcription is the first step in expressing the information encoded in DNA. The sequence of DNA—a gene—is transcribed into RNA sequence. A small part of DNA double helix is separated into strands, and one of them serves as a *template strand.* This template strand is used for the synthesis of the RNA chain, see Figure 2.1. This transcription is done by enzymes called RNA polymerases. Specifically, there are RNA polymerases I, II, and III, each transcribing different genes. As an example, protein-coding genes and most snRNA genes are transcribed by RNA polymerase II, tRNA genes by RNA polymerase III.

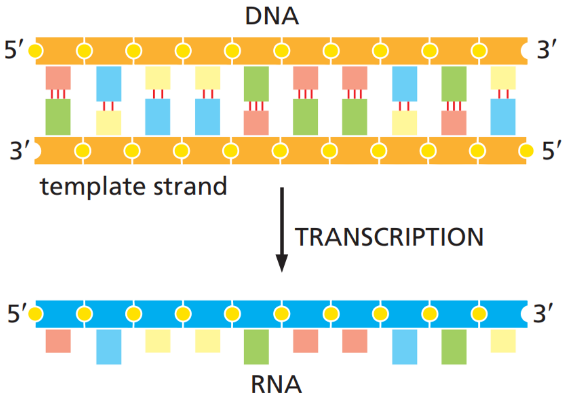


Figure 2.1: Transcription example, taken from Alberts et al. [3]

The process, however, does not end here. Before reaching the mature mRNA, several necessary steps have to happen. The unmodified RNA, immediately resulting from the transcription, is commonly called the precursory mRNA or pre-mRNA. Both ends of the pre-mRNA are modified — a cap consisting of modified guanine nucleotide is added to the 5′ end. To the 3′ end, an enzyme called poly-A polymerase adds approximately 200 A nucleotides. This sequence is then called poly-A tail (this is commonly encountered in bioinformatics tools dealing with quality control of sequencing reads, where poly-A tails are detected and usually trimmed).

Another pre-mRNA processing step is called the splicing. Genes of eukaryotic DNA was found to consists of coding regions (called *exons*) and non-coding regions (called *introns*). The introns have to be cut out — this is done by a complex assembly of RNA and proteins, called the spliceosome. Another important concept, prevalent in RNA sequencing analysis, is *alternative splicing.* During the splicing of introns and exons, some exons can be skipped or cut in half. This results in the fact that many different mRNA transcripts can come from a single gene. This concept is illustrated in Figure 2.2.

As a final note, the three above mentioned modifications to pre-mRNA do not happen after the whole pre-mRNA transcript is created. The 5′ end capping happens shortly after the start of the transcription, splicing

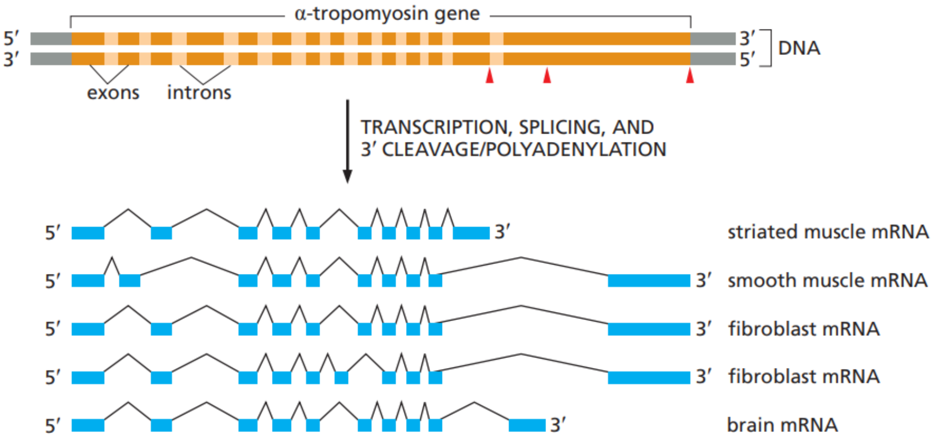


Figure 2.2: Splicing example—one gene can be transcribed into several different mRNA transcripts. Taken from Alberts et al. [3].

occurs during, and addition of poly-A tail happens with the termination of transcription. The 5′ end cap and poly-A tail thus mark a fully transcribed mRNA transcript.

2.1.2 Translation

After successful transcription of the gene into the mRNA molecule is the mRNA molecule transported into the cytoplasm. There, with the help of ribosomes (mostly made up of rRNAs), is the mRNA decoded, and protein is synthesized. It consists of amino acids connected into a chain. The amino acids are interestingly encoded in the mRNA. They are encoded by triplets of RNA nucleotides, called *codons*. This means that there are 64 possible triplets of nucleotides; there are, however, only 20 amino acids commonly used in proteins. Several codons, therefore, code some amino acids, but no codon codes for multiple amino acids — this mapping is called the *genetic code*. The mapping is shown in Figure 2.3.

GCA GCC GCG GCU	AGA AGG CGA CGC CGG CGU	GAC GAU	AAC AAU	UGC UGU	GAA GAG	CAA CAG	GGA GGC GGG GGU	CAC CAU	AUA AUC AUU	UUA UUG CUA CUC CUG CUU	AAA AAG	AUG	UUC UUU	CCA CCC CCG CCU	AGC AGU UCA UCC UCG UCU	ACA ACC ACG ACU	UGG	UAC UAU	GUA GUC GUG GUU	UAA UAG UGA
Ala	Arg	Asp	Asn	Cys	Glu	Gln	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val	stop
A	R	D	N	C	E	Q	G	H	I	L	K	M	F	P	S	T	W	Y	V	

.

Figure 2.3: The mapping of codons onto amino acids. Note how are amino acids encoded by several different codons. Taken from Alberts et al. [3]

The translation does not play a significant role in RNA sequencing, DGE, and GEP deconvolution, and therefore it was not covered in substantial detail here. The most important observation is discussed in Section 2.2 on RNA-seq — which uses RNA sequencing to infer gene expression rates and, in the case of proteins, omitting the step of translation.

2.1.3 Gene expression

The whole process of information flow from gene to functional product is also called **gene expression**. By functional products, we mean not only proteins but also many other constructs having a function, like rRNAs, sn-RNAs, tRNAs, lnRNAs, etc. The gene expression is different in each cell type; it is different in each cell cycle and can be influenced by, e.g., treatment or medical condition.

It is, therefore, of great interest to, in some way, quantify and measure the gene expression, as it can provide valuable insights into the understanding of cellular function. One can, for example, observe and measure the reaction of different cells for treatment or derive and predict the response of patients to different drugs. The area of application of gene expression data is exhaustively large [45]. One possible approach to gene expression measurement and quantification is the **RNA sequencing**, RNA-seq in short. We describe RNA-seq in more detail in Section 2.2. With the basic description of gene expression, we can now describe differential gene expression analysis and biomarkers.

Differential gene expression analysis

The analysis focused on identifying genes having different expression levels between several condition or groups is called the *Differential Gene Expression* (DGE) analysis. Numerous tools have been developed for this problem. The tools are specialized on deciding (by performing suitable statistical tests), whether a gene's expression varies between conditions or groups. Due to the nature of available data, which is most often characterized by very low number of replicated in each group, the tool need to account for that and employ proper statistical models, approaches and tests.

Examples of commonly used DGE tools are DESeq2 [65], edgeR [95][72] and limma [94] (top 3 DGE tools based on number of downloads on Bioconductor stats website [36], all implemented in R language.

Biomarkers

One of the goals of this thesis is to use gene expression profiles data for biomarker detection. It is, therefore, necessary to explain what we mean by **biomarker**, specifically in the context of differential gene expression analysis. One common definition describes biomarker as '*an indicator of normal biological processes, pathogenic processes or pharmacological response to a therapeutic intervention*' [9], also cited, e.g., by Sidefrow et al. [102]. As is the nature of DGE analysis, several samples (or groups of samples, samples being given some treatment, etc.) are compared to each other. A natural candidate for biomarker would, therefore, be a gene, uniquely (or, in general, differentially) expressed in some group of samples.

As an example of another closely related biomarker usage, consider the area of gene expression profiles deconvolution (further explored in detail in Section 2.3). For the needs of distinguishing different cell types between each other, Venet et al. [115] defines biomarker (or marker in short) as a gene, which is expressed only in one cell type. However, this definition proved to be too restrictive, and it was relaxed to '*gene, which is being expressed mostly in one cell type*' [115].

2.2 RNA-seq

RNA sequencing (RNA-seq) is a technology of high throughput sequencing (in other words, a technology able to sequence many sequences in parallel). RNA-seq is mainly used for the analysis of transcriptome and subsequently for the differential gene expression analysis [119]. The goal of RNA-seq is to sequence and quantify RNA present in a sample — and although sequencing of RNA molecules itself is possible [84], it is usually not done [48]. RNA molecules are very unstable, and the sequencing is thus more technically demanding. Most RNA-seq experiments are done by converting the RNA molecules into complementary DNA (cDNA) first. During this, the original RNA pool can undergo a selection process, for example, a selection of transcripts with poly-A tail (described in Section 2.1.1) or rRNA depletion (as it forms a huge part of transcriptome and there is usually no interest in it). After this, the RNA is fragmented into sequences of a certain size (or rather a distribution of sizes, as the fragmentation is not an exact process). There is also a possibility of reverse transcribing the whole RNA transcript, and fragment the resulting cDNA [48][119].

The resulting pool of fragmented cDNA is then ligated with special DNA adapters. Adapters serve many purposes; they can, for example, mark the $5'$ and $3'$ ends of the fragment or include a barcode or index for sample identification. They also serve as primer binding sites used for amplification, which is the next step in RNA-seq experiment [25]. The pool of prepared fragmented and amplified cDNA molecules with adaptors is called the **library**. After the amplification, finally, the sequencing reads are produced. One read or two reads (one from each end of the fragment) can be produced — they are called single-end and paired-end reads, respectively [119]. These reads are then aligned to the reference genome or transcriptome with the help of various bioinformatics tools.

It is clear that the library preparation plays an important role in the whole RNA-seq experiment and basically determines its results. As an example, consider the RNA-seq experiment as described above, with the poly-A end selection. This is a common setup for inferring the gene expression in a tissue or heterogeneous cell populations. This, however, omits the translation step of gene expression, i.e., the synthesis of proteins from mRNAs. And in general, the abundance of mRNA does not need to match the protein abundance, as many regulatory steps can happen during and before the translation.

This is the reason, why ribosomal sequencing [52][44] (Ribo-seq) was developed. The library preparation for Ribo-seq is focused on the selection of RNAs, which are bound to ribosomes, as those can better indicate the RNAs, which are in the translation process. Ribo-seq is not subject of this theses, but it serves as a reminder, that RNA-seq measures the transcriptome rather than the gene expression in general, and that the library preparation is of huge importance in interpreting the resulting data.

2.2.1 Fragments, reads, inserts, adapters

Most of the bioinformatics tools working with RNA-seq data use common terminology, which is, however, often a source of confusion. Below we give a common understanding of those terms as encountered during writing of this theses, although some of them (notably **fragment**) can have different meaning in different sequencing platforms (fragments can be considered with or without adapters). **Adapters** are constant sequences, which are added to the 5' and 3' ends of the sequence resulting from cDNA fragmentation. By **fragment**, we mean the sequence of fragmented cDNA with added adapters (this is rather counter-intuitive, as the result of cDNA fragmentation is not called fragment). The sequence of fragmented cDNA, which is between both adapters, is called the **insert**, with the length referred to as **insert length**. The **insert** is also sometimes referred to as **template DNA** [24]. In the case of paired-end sequencing, two reads from both sides of the **insert** are sequenced. In the case of the sum of the length of both reads being shorter than insert length, the length of unsequenced **insert** is called **inner distance**.

The mentioned terms are visualized in Figure 2.4.

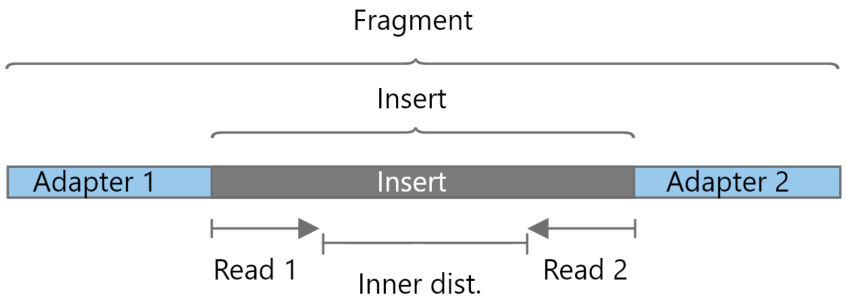


Figure 2.4: Visualization of fragments, reads, adapters, inserts and inner size in paired-end sequencing.

Count matrix as an RNA-seq output

It is now an appropriate time to visualize the final output of RNA-seq experiment, commonly referred to as *count matrix* or *raw count matrix* (*raw*, because the data did not undergo any normalization so far). It is a product of aligning RNA-seq reads to genome/transcriptome and counting how many times does the read aligns with some feature (usually transcript or gene). There are many problems connected with this counting (for example, how to deal with reads mapping to several genes?), which are being dealt with by appropriate tools. The typical count matrix is shown in Figure 2.5, with samples in columns and features (in this case, genes) in rows. This is an almost canonical arrangement, as it was used in all methods discussed in the following Section 2.4.

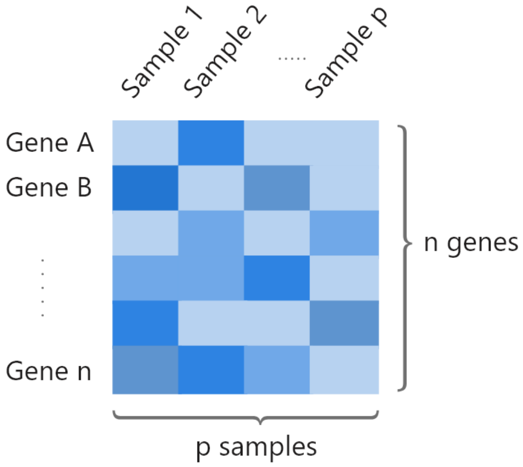


Figure 2.5: Typical representation of result of RNA-seq experiment, the count matrix of n genes and p samples. Darker and lighter colour represents higher and lower counts, respectively.

We should also mention how are the features (in Figure 2.5 genes) described, i.e., what naming standards are used. For human genes, typically symbols by HUGO Gene Nomenclature Committee (HGNC) [124], which sets standards for human gene nomenclature, are used. Other common options are Entrez Gene Id [70] and Ensembl Id [97]. All of these are used in literature and in RNA-seq tools, which sometimes results in need of conversion between those systems — this can be, for example, done by R Bioconductor package [112].

2.2.2 Paired-end vs single-end sequencing

Along with the previous section, where we described commonly used terms, we should elaborate more on the differences between single-end and paired-end sequencing. As mentioned before, paired-end reads produce reads from both ends of the sequenced inserts, single-end from one end only [119]. Although the paired-end sequencing is more costly and more time requiring [32], it allows for much more accurate alignment to the genome or transcriptome, or a discovery of novel transcripts [86].

The main reason for this is the fact, that when fragmenting the sequencing library, we expect some distribution of the length fragmentation, which can be subsequently used in the process of alignment (i.e., we know that the two reads should be in aligned to specific range from each other). Single-end reads do miss this information, and they cannot deal, for example, with the alignment to repeated sequences in the genome in an unambiguous way. The situation is depicted in Figure 2.6.

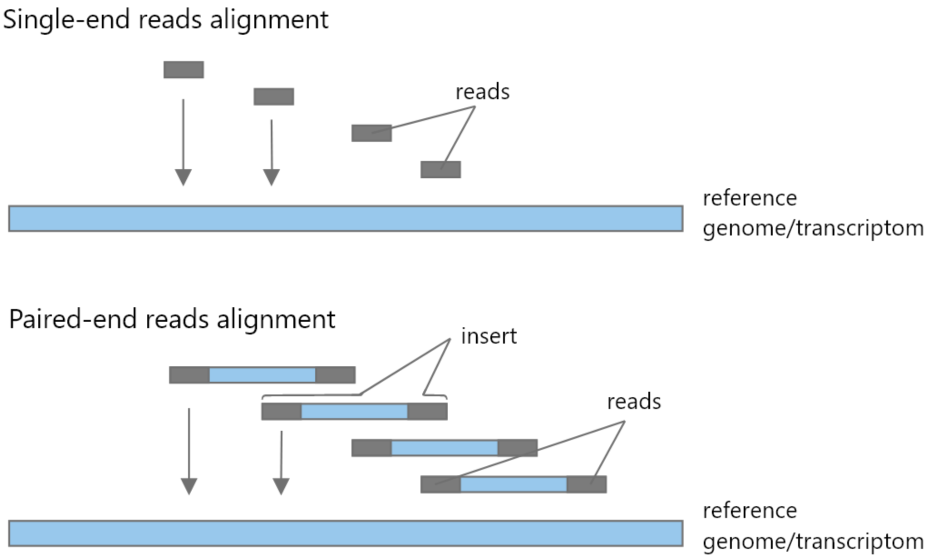


Figure 2.6: Alignment of single-end and paired-end reads. The paired-end reads can use the additional information if insert size distribution and thus better solve the ambiguities in alignment.

2.2.3 RNA-seq raw count normalization

There are specifics of RNA-seq methodology which require normalization of the raw count data, as usually outputted from the alignment tools. Samples having varying sequencing depth (number of reads produced per sample) and longer genes having a higher chance of producing fragment being read are two most notable aspects it needs to be accounted for. The normalization method is selected based on the intended use of the count data - mostly the within or cross-sample comparisons.

Several normalization methods (and resulting measurement units) are currently being used, each having its pros and cons. There is, unfortunately, some confusion of differences between those methods and their intended use [121][16]. Below we present a brief overview and explanations for most commonly used normalization methods. Understanding these methods is important for correct usage of some deconvolution methods discussed in the following sections - some of them require input to be normalized by a specific method.

At first, lets introduce the notation and conventions used in the following definitions, based on [121] and [62]. The RNA-seq sample consists of a pool of transcripts. Let M denote the distinct number of those transcripts and c_i the actual number of transcript i in the sample. Total number of transcripts in a sample is then computed as $\sum_{i=1}^{M} c_i$. From this pool of transcripts, the RNA-seq fragments are sampled — and this has serious consequences. Consider transcripts of lengths 100bp and 1000bp, and RNA-seq producing fragments of mean length 50bp. The 1000bp long transcript has a much higher chance of generating fragments than the 100bp one, even though there might be the same number of transcripts in our pool. Therefore, **the relative fractions of transcripts in the sample are (generally) not proportional to the number of their fragments being sequenced**.

To capture the difference, let us properly define those quantities. By *fraction of transcript*, we mean the fraction of transcript i present in the sample, i.e., $\tau_i = \frac{c_i}{\sum_{j=1}^{M} c_j}$. As mentioned before, this is not generally proportional to the fraction of fragments of this transcript being sequenced. The *fraction of fragment* is the fraction of fragments coming from transcript i in the pool of all fragments, denoted by $\eta_i = \frac{c_i \hat{l}_i}{\sum_{j=1}^{M} c_j \hat{l}_j}$. The $\hat{l}_i$ denotes the *effective length* of transcript i. It is usually defined [110] as $\hat{l}_i = l_i - \mu^{l_i} + 1$, where l_i is length of transcript, μ^{l_i} is the mean of empirical length of fragments shorter or equal to l_i in the sample.

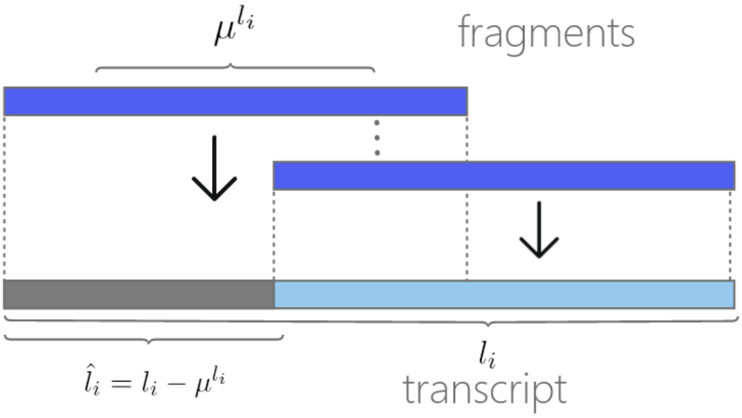


Figure 2.7: Visualization of effective length for a transcript, given empirical mean of fragment length

The *effective length* can be understood as the number of possible starting points, from which the fragment can align to the transcript and still fully fit inside it. Assuming uniform distribution, the probability of sampling specific fragment is $1/(\hat{l}_i = l_i - \mu^{l_i} + 1)$ The $+1$ in the expression accounts for the situation when $\mu^{l_i} = l_i$, so the effective length is 1, rather than 0. The meaning of *effective length* is depicted in Figure 2.7. This term models the transcript-length bias — longer transcripts having a higher chance of producing a fragment.

Let X_i denote the expected number of fragments coming out of the transcript i (which is, in fact, the count value outputted from the quantification or alignment tools). And finally, let N be the total number of mapped reads in a sample.

With these quantities defined, we can now explain the commonly used metrics and units used for transcript abundance quantification.

2.2.4 CPM

CPM (Counts Per Million) is one of the simplest units of expression measurements. It simply divides the counts for a feature by the total number of counts and scales it by ‘million’ constant.

$$\mathrm{CPM}_i = \frac{X_i}{N} \times 10^6 \tag{2.1}$$

The CPM normalizes the counts for the library size (total number of mapped reads or, i.e., the sequencing depth). It does not take into account the length-bias. It is therefore related to the *fragment fraction* η_i (expected ratio of sampled fragments coming from transcript i) rather than the τ_i, the true ratio of transcripts present in the sample. CPM is sometimes still mentioned and used [60][95].

2.2.5 RPKM and FPKM

RPKM (Reads Per Kilobase Million or *Reads per Kilobase of exon Per Million reads mapped* in full) and FPKM (Fragments Per Kilobase Million) are two closely related units of transcript abundance measurement. The FPKM is more general, as it takes into account the differences between single-end and paired-end RNA-seq. Specifically, it deals with the fact that in paired-end sequencing, two reads are produced from one fragment, so it does not make sense to count them twice (also, two reads do not necessarily map to the same transcript, for example, due to read quality). In general, for single-end sequencing, the FPKM = RPKM, for paired-end, they differ, with FPKM $\leq$ RPKM. Below we show the formula for FPKM (RPKM is essentially the same, with reads instead of fragments) because we defined X_i above as an expected number of fragments:

$$\mathrm{FPKM}_i = \frac{X_i}{(\frac{N}{10^6})(\frac{\hat{l}_i}{10^3})} = \frac{X_i}{N \cdot \hat{l}_i} \cdot 10^9 \tag{2.2}$$

The RPKM was probably first introduced by Mortazavi in 2008 [78]. Since then, it became popular and is used in tools in the RNA-seq pipelines, for example, in assembly and gene differential expression tool Cufflinks [111]. The RPKM/FPKM aims to be a τ_i estimator, i.e., estimate the true relative abundance of transcripts in a sample. However, it was found that RPKM is inconsistent between samples and is overall not very suitable as a normalization method [116][128].

The reason for this is in detail, explained in [116]. The main argument is that when using RPKM for normalization of samples with the same transcript set and different sequencing depth, the RPKM produces different average transcript abundance in each sample and thus causing interpretability issues when performing between-sample comparisons.

Example 2.1. Consider the following simplified situation. We have two samples, for each of which the counts were counted, mapping to three transcripts of given effective lengths.

Sample	Total counts	Tx A (100 kbp)	Tx B (40 kbp)	Tx C (20 kbp)
S1	200	20	110	70
S2	400	40	50	310

Table 2.1: Situation for Example 2.1

Now, let us compute RPKM values using the Equation 2.2. The results are presented in the Table 2.2

Sample	Tx A (100 kbp)	Tx B (40 kbp)	Tx C (20 kbp)	Sum of RPKMs
S1	1000	137500	31819	170319
S2	1000	31250	310000	342250

Table 2.2: RPKMs computed for values in Table 2.1

In table 2.2, we can see that the sum of all RPKM values in each sample differs. This complicates the interpretation of the RPKM values between samples. Consider transcript A in samples S1 and S2. The values are the same, implying equivalent relative (as RNA-seq is by itself relative measurement method) values of transcript A in both samples. However, when taking the proportion of RPKM to the sum of all RPKMs in the sample ($\frac{1000}{170319}$ versus $\frac{1000}{342250}$), we see that the actual relative abundances of transcript A in samples are different.

The reason for this inconsistency was explained by Wagner [116], who advocated for another unit of transcript abundance measurement, the TPM.

2.2.6 TPM

TPM (Transcripts Per Million) is a transcript abundance measurement unit, introduced by Li et al. [62][61]. It was meant to deal with the between-sample inconsistency of RPKM/FPKM and is an estimate of τ_i. It is now the recommended measurement unit to be used [116]. The TPM is computed as follows:

$$\mathrm{TPM}_i = \left(\frac{\frac{X_i}{\hat{l}_i}}{\sum_{j=1}^{M} \frac{X_i}{\hat{l}_i}} \right) \cdot 10^6 \propto \tau_i \tag{2.3}$$

To further emphasize the advantage of TPM to RPKM/FPKM, let us continue with computing TPM for count values from Table 2.1 in Example 2.2.

Example 2.2. The TPM values, along with their sum in a sample, are computed for counts from Table 2.1.

Sample	Tx A	Tx B	Tx C	TPMs
	(TPM)	TPM	(TPM)	sum
S1	31008	426357	542635	10^6
S2	23323	72886	903791	10^6

Table 2.3: TPMs for count values from Table 2.1

The vital thing to notice is the fact that the sum of TPMs in each sample is the same; it is always 10^6 — this follows from the Equation 2.3. Thus, when interpreting the TPM value between samples, the value always represents the fraction of transcript abundance from the same size — 10^6. This is, however, not what we usually want — we are more interested in the absolute abundance value. Even with TPM, the same gene with the exact same expression value can have different TPM value in two different samples, even from the same experiment and sequencing depth. For example, this happens when the distribution of other transcript abundances in the samples are different. The nominator in Equation 2.3 stays the same, the denominator can, however, be different.

As a result, **none of the above-mentioned units can be properly used for comparison across experiments** (comparison between samples coming from the same condition/experiment might be somehow reasonable if assuming similar RNA-distribution in the samples) [121].

Relation of TPM and RPKM/FPKM

It is interesting to see the relationship between TPM and RPKM/FPKM, to emphasize the difference. This relationship was derived first by Pachter et al. [85], even before the introduction of TPM. The Pimentel [121] recognized this and recited the derivation in the following form:

$$\begin{aligned} \mathrm{TPM}_i &= \left(\frac{\frac{X_i}{\hat{l}_i}}{\sum_{j=1}^{M} \frac{X_i}{\hat{l}_i}} \right) \cdot 10^6 \\ &\propto \left(\frac{\frac{X_i}{\hat{l}_i \cdot N}}{\sum_{j=1}^{M} \frac{X_i}{\hat{l}_i \cdot N}} \right) \\ &\propto \frac{X_i}{N \cdot \hat{l}_i} \cdot 10^9 = \mathrm{FPKM}_i \end{aligned} \tag{2.4}$$

Note that when using the *proportional to* symbol, it is in the sense of *proportional to in a given sample.* There is also another equation showing the TPM/FPKM relationship. Having the FPKMs for all genes in a sample, we are able to infer the TPM value for each gene [121]:

$$\mathrm{TPM}_i = \left(\frac{\mathrm{FPKM}_i}{\sum_{j=1}^{M} \mathrm{FPKM}_j} \right) \cdot 10^6 \tag{2.5}$$

2.2.7 Between-sample normalisation methods

Due to the above-mentioned problems of common normalization techniques and units, another class of normalization methods can be recognized. They are sometimes referred to as between-sample normalization (or BSN) techniques [38][51]. These are not that prominent in deconvolution methods (which usually perform deconvolution for samples independently) but are crucial for the performance of differential gene expression analysis. Therefore, we provide a short summary and explanations of several such methods.

2.2.8 Quantile normalization

Quantile normalization [17] is a method, which was commonly used in the area of microarray analysis [38]. It tries to enforce a similar distribution of gene counts in all samples. The algorithm can be summarized by three following steps [38]:

1. Sort counts for genes in all samples (assuming the same set of genes in all samples), thus having the same quantiles of all samples in the same position.

2. For each quantile, compute its mean over all samples. In all samples, replace each quantile with the computed mean of that quantile. (Other measures, e.g., median, can be used instead of the mean).

3. Revert the sorting of all samples to the previous state.

After applying this procedure, each sample has the same distribution of counts. Despite its simplicity, the method is reported to produce reasonable results in DE analysis [38].

TMM (edgeR)

Trimmed Mean of the M-values (TMM) [96] is a method of between-sample normalization introduced by authors of edgeR [95][72], in which it is commonly used. TMM tries to compute the normalization factor for each sample. It does so by setting one sample as a reference, and for each remaining sample, it estimates *log-fold change* and *absolute expression levels*. These quantities are formally defined as follows (we extend the notation from the previous section, X_{ij} and N_j now means expected number of counts for transcript i in sample j and the total number of counts in sample j, respectively):

$$\begin{aligned} M_{ij}^r &= \log_2 \frac{X_{ij}/N_j}{X_{ir}/N_r}, \\ A_{ij}^r &= \frac{1}{2}\log_2 \left(\frac{X_{ij}}{N_j} \cdot \frac{X_{ir}}{N_r} \right), \end{aligned} \tag{2.6}$$

where r is the reference sample, i is the transcript, and j is the sample, which is being compared. M_{ij}^r is the *log-fold change* and A_{ij}^r is the *absolute expression level* of transcript i between reference sample r and sample j. The M and A values are independently trimmed (in the original paper, 30% is trimmed for M and 5% for A [96]). Then a set of transcripts, for which neither the A or M value were trimmed, are used for computation of weighted mean. As weights, the inverses of the approximate asymptotic variances are used [96]. The mean is then used as a normalization factor, with which each sample is divided.

Median of ratios (DESeq2)

Normalization method introduced in DESeq [4], and further used in next iterations DEXSeq [5] and DESeq2 [65]. This method computes geometric mean of counts of a transcript across all samples and then computes ratios of these counts to this geometric mean. This is done for all transcripts. For each sample, the median of ratios of all transcripts is selected as a normalization factor. Note that this method assumes that at least half of transcripts in the sample s NOT deferentially expressed. The Equation 2.7 formally describes the computation of normalization factor s_j for sample j [65]:

$$s_j = \underset{i:X_i^R \neq 0}{\text{median}} \frac{X_{ij}}{X_i^R}, \text{ where } X_i^R = \left(\prod_{j=1}^{p} X_{ij} \right)^{1/p} \text{ and } p = \text{number of samples.} \tag{2.7}$$

The normalized values for the sample are then obtained by dividing all counts in the sample by the computed normalization factor. Note, that this formalization uses one normalization factor per sample; however, the actual DESeq2 implementation allows for different normalization factors for each transcript, based on, e.g., the effective transcript length.

2.2.9 Normalization methods summary

The final summary of the discussed units of measurement is in Table 2.4. The table was inspired by materials of Harvard Chan Bioinformatics Core (HBC) [73] and Evant et al. [38]. The quantile normalization was left out, because it is a method usually applied to microarray analysis and therefore it is difficult to compare it with the methods originally meant for RNA-seq (it is for example hard to say, whether it takes into account the transcript length bias or library size — it certainly has an effect on these problems, but it does not work with them explicitly).

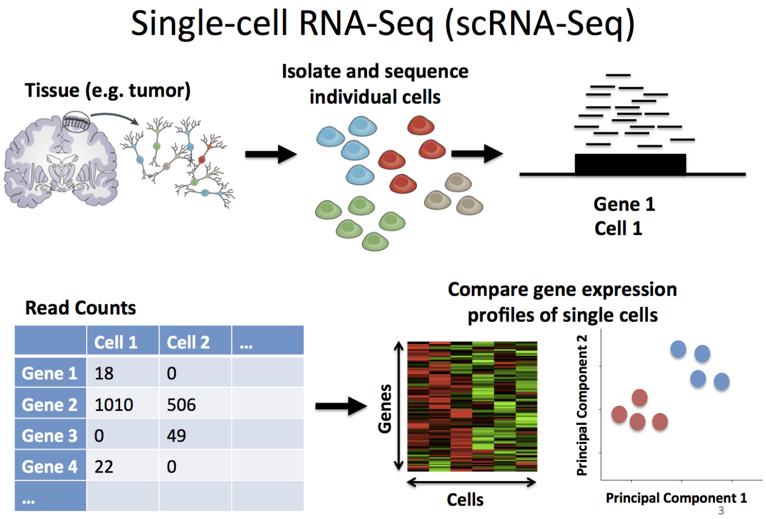
Unit/ method	Factors accounted for:	Usable for:	NOT usable for:
CPM	library size	comparison of counts between samples (from the same condition group)	within sample count comparison, DGE analysis
RPKM/ FPKM	library size, transcript lengths	within sample comparison, between sample comparison (from the same condition)	between sample count comparison, DGE analysis
TPM	library size, transcript lengths	within sample comparison, between sample comparison (from the same condition group)	DGE analysis
TMM (edgeR)	library size, RNA composition	between sample comparison, DGE analysis	within sample comparison
DESeq2 (median of ratios)	library size, RNA composition, transcript lengths (optionally)	between sample comparison, DGE analysis	within sample comparison

Table 2.4: Comparison of mentioned methods. The quantile normalization methods is left out, as it is usually applied to microarray data and the comparison is difficult.

2.2.10 Single-cell RNA sequencing

Although the high-throughput sequencing methods gave rise to a new range of possibilities of transcriptome analysis and allowed for a detailed study of gene expression in general, they still have several caveats. The main disadvantage is probably the fact that RNA-seq measures gene expression of a bulk sample, averaging the expression levels across a population of different cells (which is one of the reasons why the techniques of GE deconvolution, the topic of this thesis, were developed). Quite recently, in 2009 [108], a new technology was introduced. Single-cell RNA sequencing (scRNA-seq) [83][49] is a method (or rather class of methods), which allows for sequencing of RNA present in single cell. As thus, it allows for a detailed study of cell-specific changes of the transcriptome, identification of different cell types and subtypes, or stochasticity of gene expression in cells (which is considered as noise in traditional bulk RNA-seq [103]).

We will not go into details of the technology, as it is not the main topic of this thesis. The steps of the analysis are similar to the bulk RNA-seq, with the main difference in the cell dissociation, isolation before the sequencing, or different types of normalizations. Also, the tools for downstream analysis had to be developed specifically for scRNA-seq [46], e.g., tools for differential expression, clustering, cell sub-populations detection, etc. In Figure 2.8, the basic steps of scRNA-seq experiments are outlined.


Single-cell RNA-Seq (scRNA-Seq)
Tissue (e.g. tumor)
Isolate and sequence individual cells
Gene 1
Cell 1
Read Counts
Compare gene expression profiles of single cells
Genes
Cells
Principal Component 2
Principal Component 1
3

	Cell 1	Cell 2	...
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
...			

Figure 2.8: Usual workflow in the scRNA-seq experiment, taken from Hicks [105]

One of the common goals of scRNA-seq analysis is the detection of new cell types or differentiating known cell types into subtypes. This can be done, for example, by hierarchical clustering, or dimensionality reduction of cell expression profiles (using, e.g., PCA or t-SNE) and subsequent clustering. Typical example (taken from PanglaoDB [39], database of scRNA-seq experiments) is shown in Figure 2.9. These results are interesting in the context of bulk samples deconvolution, as they can be used as reference profiles of cell types underlying the bulk sample. This is further explored in the chapter on deconvolution.

2.3 Deconvolution

Informally speaking, the goal of deconvolution of bulk tissue gene expression profile is to estimate the amount/abundance/proportions of cell types (or cell lines, i.e., sets of cell types) in given bulk tissue gene expression profile. The individual GE profiles of those cell types can also be estimated (or given as input). This can be done in several ways, mainly depending on the data and its type available before performing the deconvolution. This means that the GEP *deconvolution* problems consist of several different problems, mostly determined by the input data available at hand. For this reason, it is very important to properly formalize the GEP deconvolution problem and specify different 'types' of deconvolution problems.

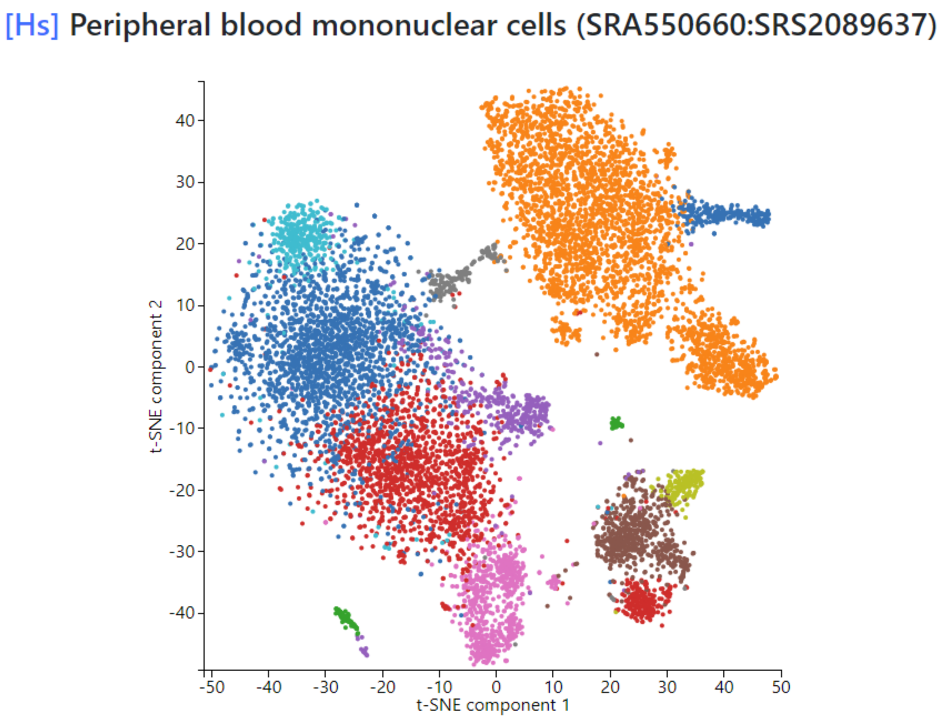


Figure 2.9: Typical t-SNE clustering of scRNA-seq results from PBMC. Different identified cell types are colored. Taken from PanglaoDB [39]

2.3.1 Deconvolution formalization

There are several approaches to comprehensive problem formalization, notably by Venet et al. [115], which was adopted e.g., by Avilla et al. [10]. Below we use the notation used by Mohammedi et al. [75], as it is slightly more thorough (detailed and rigorous matrix notation) and general (it allows for replication in reference profiles).

The notation works with following constructs [75]:

Definition 2.3. $M \in \mathbb{R}^{n \times p}$: A **mixture matrix**, where $M(i, j)$ represents the expression of gene i in sample j, $1 \leq i \leq n$ and $1 \leq j \leq p$. In other words, it is a **mixture** matrix with n genes in rows and p samples on columns. This usually corresponds to the gene expression profile coming from bulk heterogeneous tissue.

$H \in \mathbb{R}^{n \times r}$: **Expanded signature matrix**. The number of rows is the same as in M and also corresponds to the same list of genes. The columns represent reference expression profiles of cell types in question. One cell type can have several reference profiles. That is why the matrix is called *expanded*. There exists a grouping of these reference profiles/columns separating columns belonging to the same type. This matrix is not used in the notation by Venet et al. [115] and allows for the description of deconvolution using, for example, multiple scRNA-seq reference profiles.

$G \in \mathbb{R}^{n \times q}$: **Signature matrix** with reference profiles for each of q cell types in question. The gene list in rows is again the same as in matrices M and H. One column represents the reference profile of cell type, aggregated in some way from all reference profiles belonging to this cell type in matrix H. (Mohammedi et al. [75] explicitly define the aggregation as averaging, but let's not restrict us to this — more complicated solutions are explored further.) From the above it follows, that $q \leq r$ — r is the number of total reference profiles for cell types with replicates (biological or technical), q is the number of cell types.

$C \in \mathbb{R}^{q \times p}$: Matrix of **relative proportions** of cell types from signature matrix G in mixture matrix M. Rows correspond to the cell types and columns to the samples.

With the definitions of the above constructs, we can describe the deconvolution problem as follows: Deconvolution of p samples with measured expression on n genes into q cell types. We can now describe a model of deconvolution, while assuming linearity. I.e., the measures expression, expression matrix, is a linear combination of reference cell type profiles with the proportion coefficients. A visual example of this is shown in Figure 2.10. This is model and assumption is one of three main groups of deconvolution methods [78][78]. The second group consists of methods based on probabilistic models of gene expression, e.g., Bayesian model or Latent Dirichlet Allocation (LDA) [75], and the methods from the third group can be described as enrichment based methods, examples of which are given later.

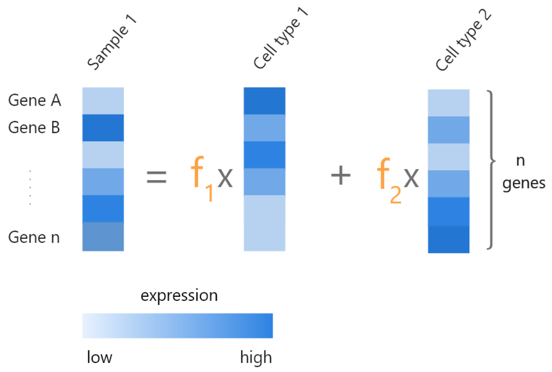


Figure 2.10: The linearity assumption visualized for one sample, consisting of two different cell types, combined in f_1 and f_2 proportions.

As mentioned above, the first group of methods employs the linear model. This can be written with the help of above-defined notation as a matrix equation:

$$\mathbf{M} = \mathbf{G} \cdot \mathbf{C} \tag{2.8}$$

For a better insight into the equation, see Figure 2.11. The figure shows an example of the Equation 2.8 with number of reference cell types $q = 2$,number of samples $p = 5$ and number of genes $n = 6$.

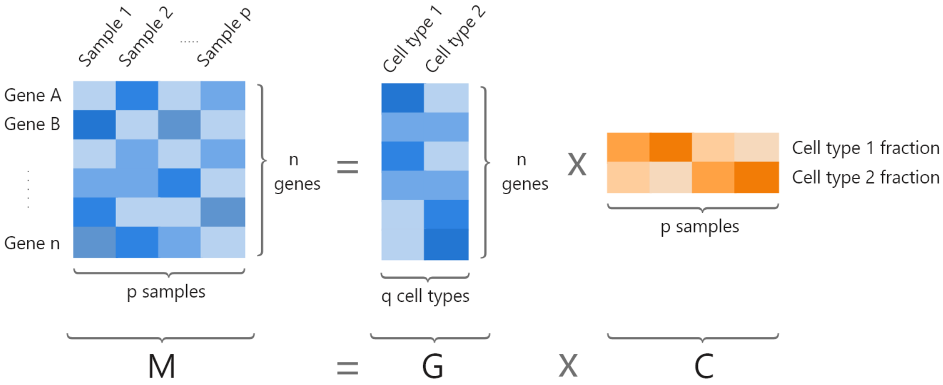


Figure 2.11: Visualisation of the Equation 2.8, where number of reference cell types $q = 2$,number of samples $p = 5$ and number of genes $n = 6$.

Using the Equation 2.8, the goal of deconvolution is to find estimates of $\mathbf{G}$ and $\mathbf{C}$, $\hat{G}$ and $\hat{C}$, respectively, and solve the following optimization problem [75]:

$$\min_{0 \leq \hat{\mathbf{G}}, \hat{\mathbf{C}}} \delta(\hat{\mathbf{G}}\hat{\mathbf{C}} - \mathbf{M}), \tag{2.9}$$

where δ is a loss function. The analysis of the usage of different loss functions used in various deconvolution techniques is explored in detail by Mohammadi et al. [75].

Based on the availability of $\mathbf{G}$ and $\mathbf{C}$ prior to deconvolution (usually their approximation based on previous knowledge or some measurement), we can state two more different optimization problems:

$$\min_{0 \leq \hat{\mathbf{G}}} \delta(\hat{\mathbf{G}}\mathbf{C} - \mathbf{M}) \tag{2.10}$$

$$\min_{0 \leq \hat{\mathbf{C}}} \delta(\mathbf{G}\hat{\mathbf{C}} - \mathbf{M}) \tag{2.11}$$

To recapitulate, this gives us three possibilities of objective minimization based on the available input data (and this applies not only to the linear models but to probabilistic models too — only the optimized objective will be different):

1. There is no prior information, and only the mixture matrix $\mathbf{M}$ is available. Both $\mathbf{C}$ and $\mathbf{G}$ have to be estimated by minimizing the objective 2.9 (in linear model approach). Further, in text, we call this the **complete deconvolution**.

2. Along with the mixture matrix $\mathbf{M}$, we have $\mathbf{G}$, the reference expression profiles of cell types possibly present in the mixture. These reference profiles may be obtained beforehand, e.g., by flow cytometry, scRNA sequencing, or sequencing of isolated those cell types in general. With

the two available matrices, we estimate the **C** matrix, the proportions of cell types in individual samples. We call this type of deconvolution the **signature matrix based** deconvolution.

3. Along with the mixture matrix **M**, we have **C**, estimated proportions of cell types presented in the samples, possibly coming from some measurement or expert knowledge. From these, we estimate GEPs of the cell types. We refer to this type of deconvolution as **proportion based** deconvolution.

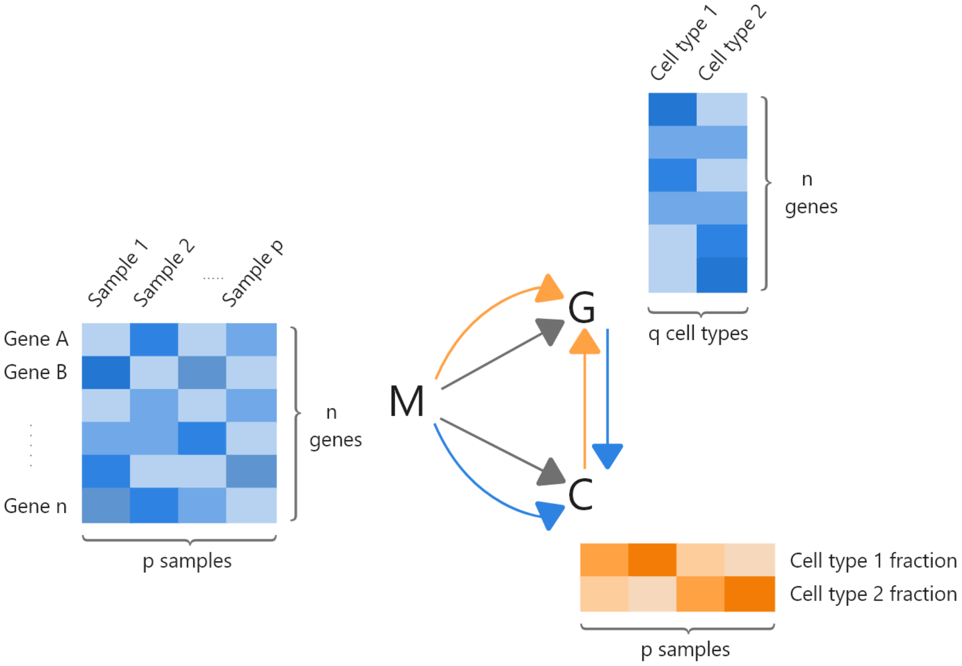


Figure 2.12: There are 3 possible situation based on the availability of input data. **1.** When having only the mixture **M** and estimating both **G** and **C**, depicted by grey arrows. **2.** Having the signature/reference matrix **G** and estimating proportions, depicted by blue arrows. **3.** Sometimes, the proportion matrix **C** is known before and the gene expression profiles of individiual cell types are estimated, depicted by orange arrows.

In literature, the two latter methods are also commonly referred to as **partial deconvolution** methods.

Some methods performing the above types of deconvolution can additionally employ a set of **markers**. These are usually set of genes (or features, in general), which are candidates for differentiating between cell types — this usually means that they are mainly expressed in one or more cell types. This is slightly more general than the previous description of biomarkers. Informally, it can be understood as a set of genes, which the tool should pay special attention in differentiating the cell types — although the exact definitions of *markers* may differ tool by tool. Sometimes only one set of markers is required; other time, a set of markers for each cell is.

To further recapitulate the mentioned types of deconvolution along with the models used, see Figure 2.13. Deconvolution, in general, can be separated into groups based on what are the required inputs and outputs. In Figure 2.13, we show this division. Methods based both on linear and probabilistic model generally perform complete, signature based and proportion based deconvolution. Methods, employing the enrichment approach, can be loosely described as performing the signature based deconvolution, in the sense of estimating proportions (or enrichment scores) — although not based on the signature matrix, but some previous knowledge, commonly sets of genes connected with the presence of some cell type.

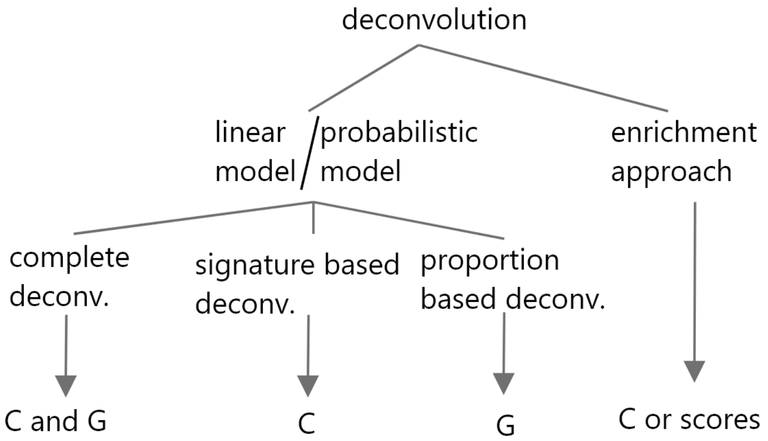


Figure 2.13: Hierarchy of discussed deconvolution types and approaches. On the bottom of the picture, the estimated outputs are shown. Methods from each group occasionally employ information about markers or estimate them.

2.4 Review of available deconvolution methods

The research area of GEP deconvolution is rather a new one. It gained interest with the arrival and wide availability of microarray technology, which produces data with the expression of thousands or even more genes simultaneously. The existence of such data has given impulse to the research of deconvolution methods, as the proper data were available.

In further sections, we show an overview of deconvolution methods published in the literature, with a brief description. At this point, it is also useful to compare these methods from the 'user' point of view, i.e., check the code availability, programming language choice, supported tissues and cell types, etc.

In the review of these methods, we focus mainly on the following metrics/questions:

- Is the implementation of the method publicly available? How? (Web application, code, library. . .)

  - Unavailability of the implementation brings the need for creating own implementation, with uncertain results. Web applications may be unsuitable for repeated computation using code. Implementation might be written in an uncommon language, thus complicating the usage.

- For what data is it intended? (I.e., microarray or RNA-seq data)
  - The underlying statistical properties of data coming from microarray and RNA-seq are different and require different normalizations and assumptions. Some older tools may not be updated to deconvolve RNA-seq data properly.
- What type of method is it? (Complete deconvolution, signature matrix based method, enrichment method. . .)
  - The differences are explained in Section 2.3
- On what datasets was it trained, tested, and validated?
  - Some methods are tested on real, some on artificial datasets. It is also important to know, from which tissue are the datasets coming, as the method can be overfitted to them.
- What cell types does it distinguish?
  - The number of recognized cell types differs between methods. For example, in the signature matrix method, it is usually the q parameter from Equation 2.8, or in the complete deconvolution methods, the number of cell types can be set by parameter by a user or fixed, usually to some low number.
- In the case of *signature matrix* methods: Does it provide the signature matrix? What cell types and how many genes are in the matrix? What is the gene nomenclature?
  - Data inputted to the method need to match the metadata (gene names or symbols) of the internally used signature method (if there is such).
- What are the input data requirements? Is specific gene nomenclature required? Do the data have to be normalized? If so, how?
  - Tools may require data to be in a specific format; common is, for example, tab-delimited text file. Tools may also internally use some gene sets with specific nomenclature (e.g. Hugo or EnsemblID), which is used internally and provided data must be accustomed to that.
- Does it provide some statistical tests or guarantees?
- And finally, is the method applicable to obtained data?

- For this, we need to specify our data in detail, which is done in Chapter 3, where we describe our data in detail and select methods, which will be applied to them.

2.5 Deconvolution methods in literature

In this section, we explore the deconvolution methods described in the literature. We split the methods in sections based on the type of deconvolution (see Section 2.3) they aim to solve.

As an introduction, we show one of the earliest experiments with the deconvolution of GE profiles done by Lu et al. [68]. Using the microarray GE profiles of yeast culture cells at specific points of their cell cycle, the proportion of cells in different phases was predicted in heterogeneous populations, see Figure 2.14. This used the formulation of *signature matrix* deconvolution problem.

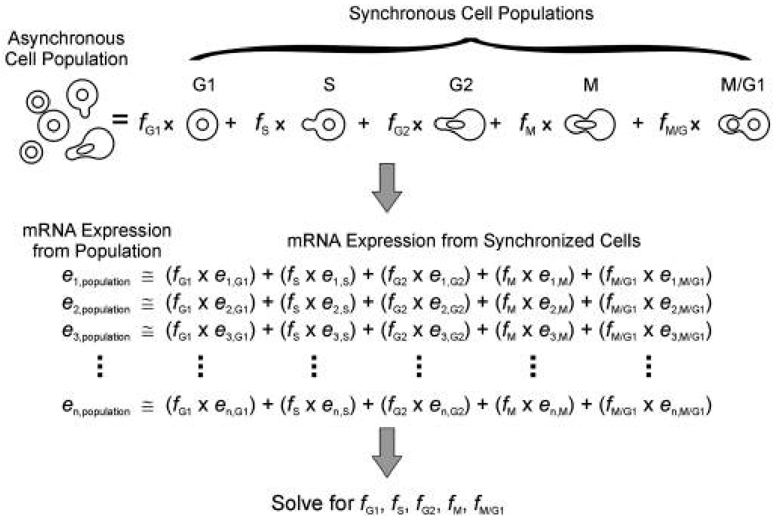

Synchronous Cell Populations
Asynchronous Cell Population
G1
S
G2
M
M/G1
$= f_{G1} \times$ + $f_S \times$ + $f_{G2} \times$ + $f_M \times$ + $f_{M/G} \times$
mRNA Expression from Population
mRNA Expression from Synchronized Cells
$e_{1,population} \cong (f_{G1} \times e_{1,G1}) + (f_S \times e_{1,S}) + (f_{G2} \times e_{1,G2}) + (f_M \times e_{1,M}) + (f_{M/G1} \times e_{1,M/G1})$
$e_{2,population} \cong (f_{G1} \times e_{2,G1}) + (f_S \times e_{2,S}) + (f_{G2} \times e_{2,G2}) + (f_M \times e_{2,M}) + (f_{M/G1} \times e_{2,M/G1})$
$e_{3,population} \cong (f_{G1} \times e_{3,G1}) + (f_S \times e_{3,S}) + (f_{G2} \times e_{3,G2}) + (f_M \times e_{3,M}) + (f_{M/G1} \times e_{3,M/G1})$
$e_{n,population} \cong (f_{G1} \times e_{n,G1}) + (f_S \times e_{n,S}) + (f_{G2} \times e_{n,G2}) + (f_M \times e_{n,M}) + (f_{M/G1} \times e_{n,M/G1})$
Solve for $f_{G1}, f_S, f_{G2}, f_M, f_{M/G1}$

Figure 2.14: Yeast culture cells deconvolution formulation, taken from Lu et al. [68]. The proportions of cells in G1, S, G2, M and M/G1 cell cycle are computed.

2.5.1 Signature/reference matrix based methods

This class of deconvolution methods benefits from the availability of the reference expression profiles of cell types possibly present in the mixture. These profiles then help to deliver more informed proportions of cell types in individual samples.

LLSR

Linear Least Squares Regression (LLSR) [1], is a commonly used name for a method, which solved Equation 2.11 by least squares regression. As the least square regression can produce negative values in matrix C, which is not desirable, the LLSR [1] deals with that by removing cell type, whose fraction was negative, and running the least squares regression again. Method was originally validated for microarray data, but the assumptions of linearity probably allow for application on RNA-seq data [129].

Implementation of LLSR method is available in CellMix R library [41] and by itself, it does not provide any signature matrix.

QP

QP is a method developed by Gong et al. [43]. It performs deconvolution by optimizing Equation 2.11, with L2 loss. It does so by using the quadratic programming, with imposed constraint on non-negativity of the C matrix and sum-to-one constraint on the columns of matrix C. This is very similar to the previously described LLSR, but with the constraints specified explicitly. An implementation of QP method is also available in CellMix R library [41], and there is no signature matrix explicitly provided by the authors of the method.

CIBERSORT

Cell-type Identification By Estimating Relative Subsets Of RNA Transcripts (CIBERSORT) [80] is a *signature matrix based* method, based on optimizing the criterion from Equation 2.11 using the linear nu-support vector regression (ν-SVR) [98]. CIBERSORT provides a method for the construction of signature matrices, based on gene filtering and minimization of the condition number of the matrix [120]. The condition number is used as a measure of the stability of the system of linear equations to input variation and noise. It is, therefore, possible to create own signature matrices, although the GEPs of individual cell types have to be available, which is often not the case.

That is why the authors of CIBERSORT provide a signature matrix called LM22. The matrix describes 22 leukocyte cell subtypes, using 547 genes. The matrix was built using the CIBERSORT's approach to signature matrix creation and is available to reproduction on the CIBERSORT website [26]. Four years after the CIBERSORT publication, its authors presented a new signature matrix, LM6, designated for use with RNA-seq data [23]. It is, again, a signature matrix consisting of leukocyte cell types, this time, it is made up of 6 of them. It is available to download on the CIBERSORT website [26].

CIBERSORT also produces empirical p-values, for the null hypothesis of '*no cell types from the signature matrix are present in the mixture sample.*' This means that no p-values are reported for the individual cell type's proportions or even their presence in the sample.

The authors tested the method mostly on tumor and PBMCs (both simulated and real datasets, which were tested against ground truth obtained by flow cytometry), but they state that it is applicable to 'nearly any tissue'[80]. They also report partial results of the deconvolution of whole blood, with promising results. It is also tested only on microarray datasets, although the authors state, that the assumptions made by their method should hold for RNA-seq data too, and for practical usage, they recommend to skip the quantile normalization step (as it is generally used as a microarray normalization method, see Section 2.2.8).

The tool is written in R and Java, and is available as a web tool [26] or R package. The web tool and R package are available only after registration, which has to be confirmed by the authors/providers.

EPIC

EPIC [92] is another signature matrix based tool, optimizing the Equation 2.11. It uses its internally built signature matrix, which used publicly available datasets of immune cells from peripheral blood, along with the reference profiles of tumor-infiltrating cells — the user can choose which one of them to use.

The main purpose of this tool is to deconvolve bulk RNA-seq data of tumor tissues; however, the authors report that the method was validated both on PBMCs and whole blood. The tool's focus on tumor tissues results in an implicit assumption of the presence of unknown cell types in samples. EPIC, therefore, reports a fraction of *unknown* cell types — this is a unique feature, not present, e.g., in the CIBERSORT method. EPIC does not report any p-values.

EPIC is available web tool [91], python wrapper [81] and R package [90].

DeMixT

DeMixT [118] is a method focused on heterogeneous tumor tissues deconvolution. It differs from the previous methods, as it only performs deconvolution of the bulk sample into 2 or 3 components. For it to do so, it needs expression data of the 1 or 2 components (for 2- and 3-component deconvolution, respectively) present in the sample. It estimates proportions and expression profiles of the one not provided component. From this point of view, it could also be considered a *complete deconvolution* method, but due to the dependence on provided expression profiles of sorted samples, we include it in the *signature based* methods.

The method is applicable both to the microarray and RNA-seq data, and its implementation is available on GitHub [31]. It was tested on both microarray and RNA-seq data on publicly available datasets with known ground truth. The method is solely aimed at tumor deconvolution; it does not provide any statistical guarantees or p-values.

ABIS

ABIS [77] is a recently published method. It uses robust linear regression for optimization of the objective from Equation 2.11. It is available as a web tool [30], with source code of a R library available on GitHub [76].

Although applicable both to RNA-seq and microarray data, it is meant for PBMC samples only. It also does not report any results of statistical testing on the obtained results. It requires the data to be TPM normalized.

CIBERSORTx

We put CIBERSORTx [79] in the signature matrix based methods, but it is rather a collection of different methods, allowing for a variety of computation. It is developed on the basis of CIBERSORT [80] by the same authors, and it greatly extends the tool functionality. Similarly to the CIBERSORT, it provides a web-based interface [27], which is accessible after registration and confirmation of the authors of the tool.

The main advantage of CIBERSORTx is the integration of scRNA-seq data in the workflow of deconvolution while controlling for the unwanted effects of the origin of technology of the data, i.e., cross-platform variation. The tool allows for the usage of microarray, RNA-seq, and scRNA-seq data. CIBERSORTx consists of different modules based on the desired goal. It can be used for signature based deconvolution, individual cell types GEPs profiling, and custom signature matrix creation. The custom signature matrix can be created from sorted bulk RNA-seq data, sorted microarray data, or scRNA-seq data. With the increasing availability of scRNA-seq datasets from various tissues in online databases [39], this promises the possibility of creating precise custom signature matrices for specific needs. This process is shown in Figure 2.15 and is further explored in Chapter on deconvolution.

The underlying deconvolution computation is the same as in CIBERSORT [80], with the difference being in the incorporation of batch correction mechanisms. Two batch correction methods were developed, *B-mode* and *S-mode* [79]. The former dealing with the technical differences between the sequencing of sorted and bulk RNA-seq data (i.e., the signature matrix and mixture matrix) and the latter with excessive cross-platform variation (for example, signature matrix coming from the scRNA-seq dataset).

Statistical properties of the deconvolution in CIBERSORTx are the same as described in CIBERSORT. The methods were thoroughly tested both on (sc)RNA-seq and microarray data, as well as on both external and own patient samples. Those methods and datasets are described in detail in the original CIBERSORTx article [79].

At the time of writing, the tool was available mainly as an online tool [27], with the possibility of downloading the tool in a docker image upon request and subsequent permission from authors of the tool. This slightly complicates the usage compared to CIBERSORT, which was available to download as an R library. The documentation to the tool is very good and contains many examples, with the availability of reproducing results described in the original article.

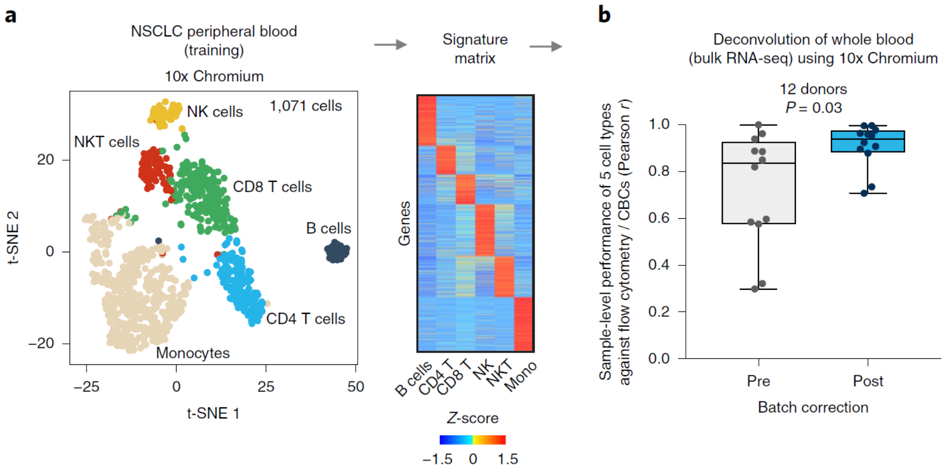


Figure 2.15: The process of creating custom signature matrix based on scRNA-seq datasets, and subsequent application to bulk RNA-seq data prior and after batch correction. Taken from Newman et al. [79].

DeconRNASeq

DeconRNASeq [42] is a method specifically focused on bulk RNA-seq mRNA deconvolution. It solves the Equation 2.11 by non-negative least-squares constraint problem with quadratic programming. The method is mainly focused on tissues; i.e., it is not meant for blood samples. It accepts user-supplied signature matrices; one signature matrix is part of the R package, which consists of reference profiles of, e.g., brain, muscle, lung, liver, and heart tissues. In other words, by default, it is not able to deconvolve in a heterogeneous sample in cell types fractions, only in tissue fractions.

In the manual, it leaves the input data normalization to the user, with the only restriction being that the same normalization of both mixture and signature matrix is selected. The data should also be in a non-log space. In provided examples, DeconRNASeq uses RPKM units, which are no longer recommended, see the section on normalization.

Other signature matrix based deconvolution methods

There are also many other deconvolution methods, some of them being published even during writing of this thesis, namely SCDC [35][21], Bisque [53], MuSiC [117], Bseq-SC [11][20], Deblender [33][58] and DWLS [113][37]. All of the above mentioned methods (except for Deblender) are focused on integrating scRNA-data into the pipeline of bulk RNA-seq data deconvolution. They deal with various problems arising when working with scRNA-seq datasets, such as cross-platform variations, combining multiple scRNA-seq datasets etc. As the general assumption of these methods is that the RNA-seq dataset comes from the same tissue, which was not the case in our situation, we do not explore them further in detail, keeping the CIBERSORTx as the main representative of this class of methods.

Overview of discussed signature matrix based deconvolution methods is shown in Table 2.5.

	Signature based methods							
	Sig. matrix	cells	p-values	data origin	norm. req.	applicable to	code	web tool
CIBERSORT	yes	6/22	yes	RNA-seq/microarray	no, raw counts	PBMC, tumors, tissues	yes	yes
LLSR	no	-	no	microarray***	no	-	yes**	no
QP	no	-	no	microarray***	no	-	yes**	no
DeconRNASeq	yes	16	no	RNA-seq	yes	tissues	yes	no
EPIC	yes	6/7 + 1	no	RNA-seq	FPKM/TPM	tumors, blood*	yes	yes
ABIS	yes	17	no	RNA-seq/microarray	TPM	PBMC	yes	yes
QuantiSeq	no	10(11)	no	RNA-seq/ microarray	TPM	tumors, blood*	yes	no

Table 2.5: Comparison of selected signature matrix based deconvolution methods, **(*)** not meant to be directly used for blood, but some validation or testing was done on blood sample, **(**)** code available only upon request, **(***)** developed for microarray data, but could possibly be applicable to RNA-seq too [129].

2.5.2 Proportion based methods

Proportion based methods, as described in the previous section on deconvolution method types, are not very commonly used, as the proportions are often not known and it is expensive to measure them — therefore, the other mentioned types of deconvolution were introduced. But it can happen, that the proportions of all cell types are known (for example, by flow cytometry), and the gene expression profiles of individual cell types is not. Then, these methods would allow for computation of those GEPs for individual cell types in each sample, and, for example, the differential gene expression analysis could be performed between those GEPs instead of bulk GEPs, probably leading to better results.

One method, employing such approach, is cs-SAM [100]. It was originally used on microarray measurements of patients with kidney transplants. Whole blood samples from patients witch acute rejection and those with stabilized post-transplant states were collected. Then, using Coulter counter, fractions of white blood cells subtypes were obtained, and individual GEPs of those subtypes were deconvolved. This allowed for more precise evaluation differentially expressed genes, which were not discovered in the heterogeneous samples.

The method was tested only on microarray data, no p-values are presented.

2.5.3 Complete deconvolution methods

The idea of complete deconvolution arose along with the *signature matrix* methods. One of the first attempts to properly formalize the problem was done by Venet *et al.* [115]. This article describes a solution to the complete deconvolution by using non-negative least squares and non-negative matrix factorization.

LinSeed

LinSeed [125] is a method developed for complete deconvolution. It introduces a concept of identifying genes specific to one cell type by their *mutual linearity*, i.e., the ability of two gene expression to follows a $y = k \cdot x$ relationship. Linseed does this by first identifying one set of all pairs of genes, which are mutually linear and clustering this set into subsets, based on collinearity networks. The basic idea is illustrated in Figure 2.16.

Authors further explore the space of mixed gene expression profiles, i.e., the columns of mixture matrix, and the space of proportion vector, i.e., the proportion matrix. Informally, they identify a common linear subspace, to which they project both the point in gene expression space and proportion space (after various normalization and transformation steps). They argue that the projected points form a simplex, whose corners can be identified as pure cell types, and points closest to the simplex corner are genes, mainly expressed in those cell types. After identification of those genes, authors

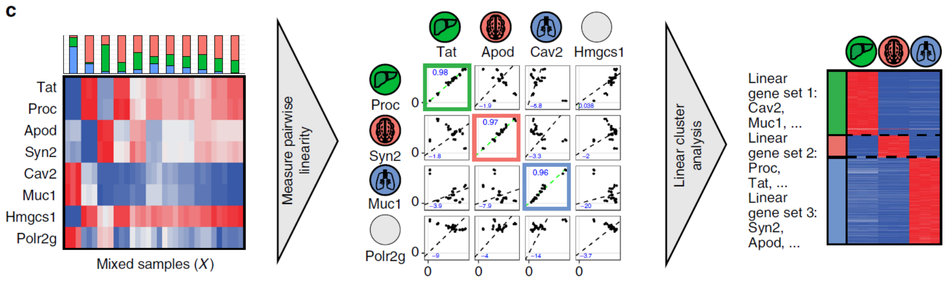

c
Tat
Proc
Apod
Syn2
Cav2
Muc1
Hmgcs1
Polr2g
Mixed samples (X)
Measure pairwise linearity
Tat
Apod
Cav2
Hmgcs1
Proc
Syn2
Muc1
Polr2g
0.98
0.97
0.96
0
Linear cluster analysis
Linear gene set 1: Cav2, Muc1, ...
Linear gene set 2: Proc, Tat, ...
Linear gene set 3: Syn2, Apod, ...

Figure 2.16: The idea behind Linseed — identify set of genes, which are mutually linear and use those as markers for gene deconvolution. Taken from Zaitsev et al. [126].

deduce the actual cell type by gene enrichment analysis (this is, however, not a part of the publicly available implementation).

Additionally, the method explores the consequences of different amounts of RNA molecules in different cell types and develops a procedure for dealing with it. It also provides a recommendation for a way of selecting the optimal number of underlying cell types (which is usually not known in complete deconvolution) based on SVD, although the number is not automatically inferred and the user has to choose it based on the provided plot of *variance explained vs. the number of cell types.*

The method was validated and tested on both microarray data and RNA-seq data, with publicly available, artificial, and author-prepared datasets from human and mouse. The method can also be used for any tissue, as it is a complete deconvolution method. The method by itself does not produce any values on the statistical significance on the computed values; there are, however, steps in the algorithm, where statistical testing for the mutual linearity of genes is performed, and the user can select the desired significance level.

The implementation of the LinSeed method in R is publicly available on GitHub [64], along with the manual on usage.

CDSeq

CDSeq (Complete Deconvolution for Sequencing data) [55] is another recently developed complete deconvolution method (2019). Contrary to the previous method, which used the projection of points on simplex in linear subspace, this method is based on the probabilistic model, specifically Latent Dirichlet allocation (LDA) [15]. This is a probabilistic model, originating from the field of natural language processing, where it is used for inferring topics present in the corpus of texts. This can be understood as an analogy for searching for individual cell types present in mixture bulk RNA-seq data. Several other deconvolution methods are based on LDA, for example, PERT [89], but they are not complete deconvolution methods.

CDSeq extends the LDA model in order to include the dependence of cell-type-specific GEPs on the gene length and to account for the varying amount of RNA produced by cells of a particular type, which is usually influenced by current cell cycle and cell size [71].

One feature of CDSeq, uncommon in other complete deconvolution methods, is its ability to estimate the number of underlying cell types constituting the bulk sample (but if known beforehand, the user can set the correct number by itself). It does so by maximizing the posterior distribution of the model [55].

The method was also intensively tested against the CIBERSORT's LM22 signature matrix. The CDSeq was used to deconvolve the LM22 matrix itself — the expected result would be to identify 22 cell types correctly, and for each sample (or, in this case, the cell type) predict 1.0 proportion of that cell type. The authors report this value to be over 0.9 for all cell types. This is connected to an interesting concept presented in the article, the *quasi-unsupervised* strategy. In order to improve deconvolution results, one can append known GEPs of pure cell types to the mixture matrix, i.e., adding one pure sample. This is reported to 'guide' CDSeq to the correct estimation of the underlying cell type's GEPs. To our knowledge, this was not tested in other complete deconvolution methods, although it could potentially have a similar effect.

The method was tested on both RNA-seq and microarray data, although the authors describe the tool as focused on RNA-seq deconvolution. It is also not meant to be used for a specific tissue. The method implementation in Matlab and Octave is available on GitHub [54] with a demo of usage.

2.5.4 Enrichment based approach

Some of the first approaches to GEP deconvolution were made using the Gene Set Enrichment Analysis (GSEA) [107]. GSEA is a knowledge-based method, using pre-compiled sets of genes sharing some property, such as location on a chromosome, biological function, or cell types of origin, in which they are deferentially expressed [2]. The GSEA then tests whether the distribution of genes from those gene sets differs from the uniform distribution when sorted with respect to a ranked list of genes from GEP (many ranking metrics are possible and used [130]). The basic idea is illustrated in Figure 2.17 [107]. It is important to notice that this is, in fact, not a 'true' deconvolution method, as it computes only an enrichment score (ES) for given cell types. This score then does not correspond to the proportion of cell type in the sample, i.e., percentages. It is not possible to compare ES of different cell types in one sample; this method server more for the detection and presence of certain cell types. Comparison of cell type ES might be possible; however, the scores are in general, not on a linear scale, so the interpretation might be difficult.

Although this approach seems to be not directly useful for the deconvolution, there are steps in deconvolution (as mentioned before in complete deconvolution), where they prove to be very helpful in enriching and correlating latent cell types with true types.

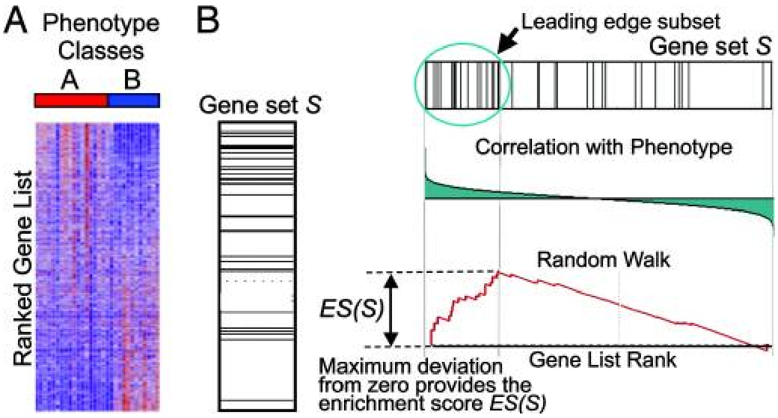


Figure 2.17: An example of GSEA method, with ranking based on phenotype correlation [107]

xCell

There is xCell [7], a method developed using this approach that provides results resembling the deconvolution (mainly by transforming the enrichment scores to linear scale and thus allowing for interpretable inter-sample comparison). It is therefore comparable with other fraction-producing methods and allows for easier interpretability. The method is applicable to both microarray and RNA-seq datasets, and recognizes 64 cell types. It was tested on simulated and real datasets, validated by cytometry. In the method's manuscript, the usage of the method both on whole blood and PBMC is reported. It is available as a web tool or R library.

MCP-counter

Microenvironment Cell Populations (MCP)-counter [12] could be understood as a method using both the signature matrix and enrichment based approach. From publicly available datasets, it identified the so-called Transcriptomic Markers (TMs). These are defined as *gene expression features expressed in one and only one cell population* [12]. The expression of these markers is then used for the computation of scores for given cell populations. The method is mainly intended for tumor tissue deconvolution; it was, however, validated on PBMCs, which suggests that it can be used for blood deconvolution. It is applicable both to RNA-seq and microarray data, and it does not provide any p-values for the computed values.

2.5.5 Available deconvolution methods reviews

The problem of mapping the situation in this young, yet somehow complicated and chaotic area of bulk tissue deconvolution methods has been tackled by several works already, which aim to provide summarized and compact overview of this area [10][75][122][99]. Recently (2020), a book providing [114] exhausting overview of available deconvolution methods was published. To get an idea of the vast number of deconvolution methods, see Figure 2.18, which shows the number of deconvolution methods covered in the mentioned review articles.

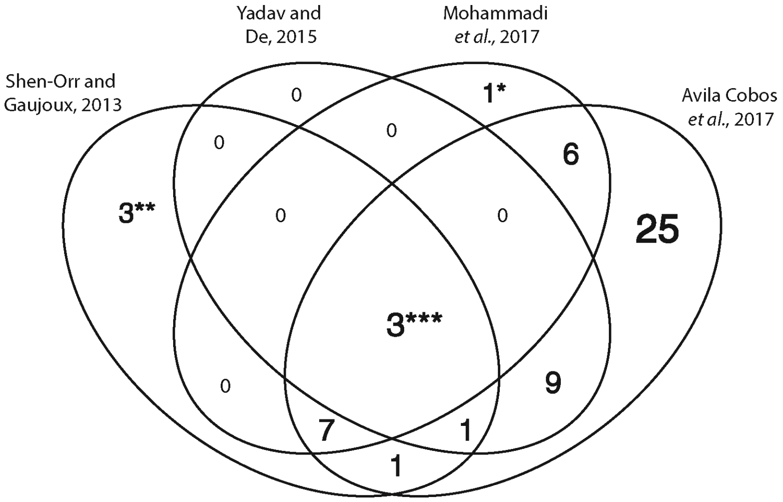


Figure 2.18: Number of deconvolution methods covered in shown articles/publications, taken from [10]

Chapter 3

Data and methods

In this chapter, we describe the provided data and perform preprocessing steps necessary for their deconvolution and differential gene expression analysis. Based on the properties of data, we discuss and select appropriate deconvolution methods, which are applicable to them.

3.1 Data

In the next two section, we briefly describe the origin of the data, followed by the description of the data itself.

3.1.1 Data origin

The data provided for this thesis come from a real medical research experiment. The transplantation often results in rejection by the immune system of the recipient, which results in the need for immunosuppressive drugs to prevent rejection. The usage of these drugs is, however, associated with various negative side effects [22]. Some patients, who discontinued the use of immunosuppressive drugs (for example, by non-adherence to the usage), were surprisingly not experiencing the rejection. This state of the patient is termed the *operational tolerance* [22].

The data come from a research experiment, whose goal was to identify biomarkers able to predict the *operational tolerance* of kidney transplant recipients. Whole blood samples were collected from kidney transplant patients in various states, along with the samples of healthy patients, used as the control.

3.1.2 Data description

The provided data can be divided into 3 parts:

1. **Count matrix** (see Section 2.2.1) of 80 samples, externally prepared by SEQme [34] company. The count matrix recognizes features at the transcript level, which are described by the Ensembl Id [112]. There are 187626 transcripts, with 161734 transcripts having non-zero expression in at least one sample.

2. **Raw paired-end reads**(see Figure 2.6) as resulting from the RNA-seq, presented in FASTQ [28] data format. Two reads per sample, 160 FASTQ files in total. The sequencing was done by SEQme [34].

3. Sample's **metadata**, containing information about the group, to which the sample belongs, and what immunosuppressives does the corresponding patient take. There are 5 groups of samples; OT, CR, CyA, STA and HC.

3.2 Deconvolution methods selection

For the method selection, we need to specify the data properties so that the appropriate methods can be selected. The main defining feature of the data is the origin of the samples — peripheral blood. The samples are also said to be globin depleted. This is not a very common source of data for most of the deconvolution methods, which are usually (as explored in previous chapter) performed on peripheral mononuclear blood cells (PBMCs) separated from the whole blood. It is therefore necesarry to decide, whether whole blood samples with globin depletion can be deconvolved using the PBMC validated methods.

The cells present in whole blood consists mainly of red blood cells (RBCs, or erythrocytes), while blood cells (WBCs or leukocytes) and platelets. The question is how much of the RNA present in the samples comes from the PBMCs — if it is a high fraction, the method could probably be used and the remaining RNA could be considered as a noise. This is, however, not the case – residual RNA from reticulocytes (immature erytrhocytes, that still posses a nucleus) contributes to up to 70% of the RNA in blood, most of it being the globin mRNA [18]. Fortunately, it seems that the globin depletion procedure seems to deal with that, and allows for the study of the transcriptome, including the PBMCs [101] [59]. We therefore suppose that the PBMCs, which are recognized by most of the deconvolution tools, are present at reasonable level in the samples.

It is also unclear, if the methods, originally intended for microarray data, are applicable to our RNA-seq data. It is somehow possible to augment the RNA-Seq data for usage with methods aimed for microarray data, that is for example explored in *voom* paper [60], and there is a study by Zhong et al. [129], which reasons, that the linearity assumption used for microarray data also hold for RNA-seq data. Therefore, we do not rule out the usage of methods originally used only for microarray data, even though we recommend caution when interpreting them.

In conclusion, we select any method, which was tested or is intended for use with blood or PBMCs, is intended for microarray or RNA-seq data, and has an available implementation, either as a library or a web tool. In case of signature matrix based method, the signature matrix should be provided — but this is not strictly necessary, as signature matrices from other methods can be used.

Based on the previous points, the following methods were selected for the gene expression deconvolution:

- **CIBERSORT**, although generally meant for microarray data, the authors did not rule out its application to RNA-seq data, even with the LM22 signature matrix. It also provides the LM6 signature matrix, developed specifically for usage with LM6. The authors also mention testing CIBERSORT on whole blood samples.
- **CIBERSORTx**, with the custom signature matrix based on scRNA-seq data. This matrix can be used with the standard CIBERSORT deconvolution framework.
- **EPIC**, can be used for RNA-seq data, provides the signature matrix for cell types present in the blood.
- **xCell**, as a representative of enrichment based methods, which provides scores resembling percentages. This makes the comparison with other methods easier. The method's manuscript has also reported and validated usage on whole blood.
- **Linseed**, as a representative of a complete deconvolution method. We will try to map the identified latent cell types to the known types using an enrichment approach.
- **QP** and **LLSR** are both methods originally intended for microarray deconvolution. Their implementation is also readily available in R package CellMix [41]. Both methods do not provide their own signature matrices; therefore, others (LM22 and LM6, specifically) will be used.
- **ABIS**, which is applicable to RNA-seq data, and its signature matrix contains immune cell types present in the blood. Implementation available as a web tool or R library.
- **QuantiSeq**, with reported usage on blood, implementation available as R library and recognized immune cell types present in blood.
- **MCP-Counter**, although mainly intended for tumor tissue deconvolution, it was validated on PBMCs separated from blood, promising applicability to our data.

For more information about selected methods, refer to Section 2.5 with description of various deconvolution methods available in literature, including the above-selected methods.

3.3 Data preparation

As all of the selected methods work on the gene level (except for the complete deconvolution methods, which could probably be used on the transcript level, but authors report validation only on gene level), we need to prepare our data in this format. The count matrix provided for this thesis consists of count data on the transcript level. It is therefore necessary to somehow aggregate the counts from transcripts to genes.

Problem of transcript to gene aggregation

Number of rather naive approaches of the aggregation comes to mind; for example summation of all transcripts belonging to one gene, computing the average or taking the maximal value of all those transcripts and setting it as the gene count value. We decided to explore the idea, that the the choice of this aggregation approach will not have a significant impact on the deconvolution results. We therefore applied the these three mentioned aggregation approaches (summation of transcripts, average of transcripts and max of transcripts) and applied CIBERSORT with LM6, as described in the next chapter on deconvolution.

In Figure 3.1 and Figure 3.2 we show a results of this deconvolution on samples OT05 and CR01. It is clear, that results are significantly different, where, e.g., cell type T1 is not even recognized in the averaging aggregation for sample OT05.

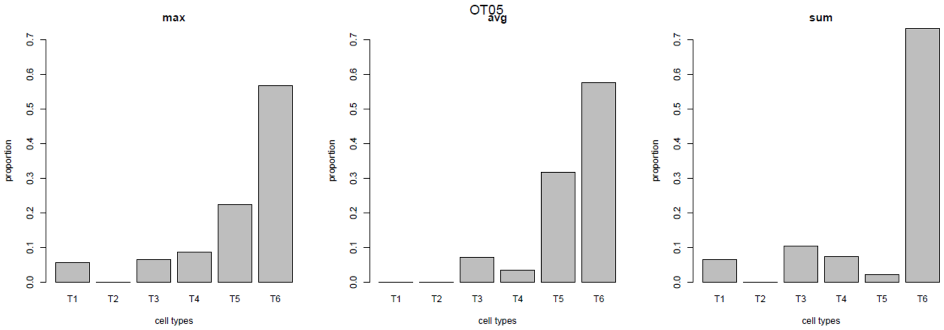


Figure 3.1: Results of CIBERSORT LM6 deconvolution on sample OT05 from gene count matrix obtained by aggregation from transcripts. The results vary significantly between different aggregation approaches.

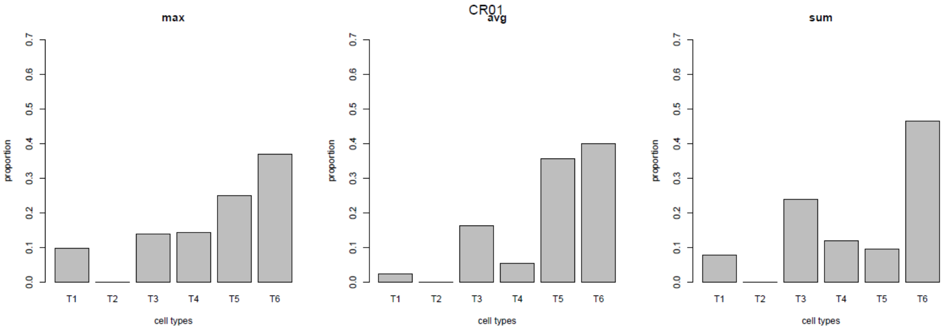


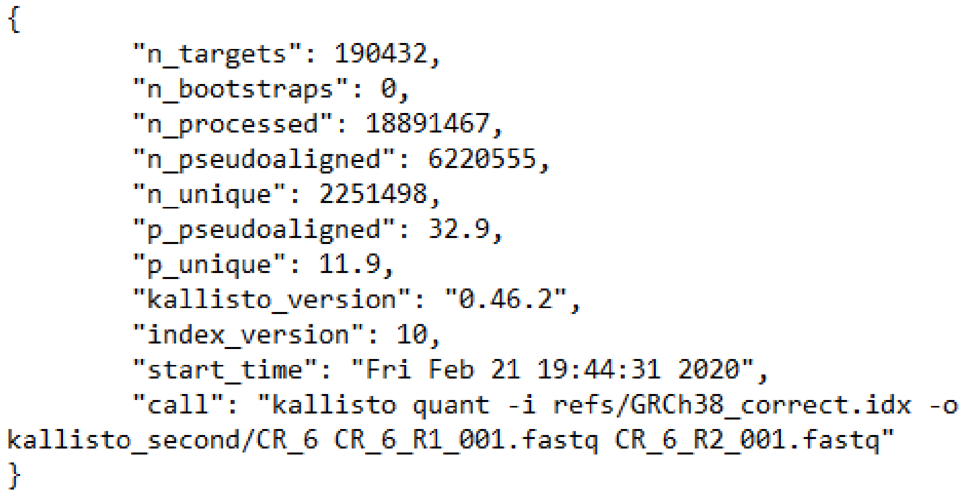
Figure 3.2: Results of CIBERSORT LM6 deconvolution on sample CR01.

We therefore concluded, that the aggregation approach is of great importance, and has to be performed properly. It came out, that to correctly aggregate transcript counts to gene level, one has to use information gained from the process of alignment of RNA-seq reads to genome/transcriptome [104], specifically the effective length of transcripts, see Section 2.2.3. This approach has been implemented in the R library *tximport* [67], which is also available of directly importing the data to the DGE tools, like DESeq2, and is the author-recommended tool-of-choice for this task. Sadly, the *tximport* is not directly available to the this transcript count matrix, as no data gained from the alignment step are present. Also, *tximport* only accepts files as outputted from alignment tools as an input. For this reason, we decided to perform the alignment of RNA-seq reads by ourselves, and subsequently aggregate them to gene level using the *tximport* library.

3.3.1 RNA-seq reads alignment

For the alignment of reads, we decided to use the *kallisto* tool [19]. It is a tool performing the so-called *pseudo-alignment* [47]. This means, that instead of aligning to the whole genome, the tool accepts a set of transcripts, transcriptome, and classified all reads as belonging to one or more transcripts (or to no transcript at all). This is much faster and less memory-intensive, compared to traditional alignment tools [19]. This is particularly important in our case, as the FASTQ read files occupy almost 2TBs of space.

Kallisto requires an index of reference transcriptome to be build. For this the *Homo_sapiens.GRCh38.cdna* reference from Ensembl [123] website was used. All 80 samples were aligned to this reference. However, the tool reported surprisingly low mapping rate was reported for all 80 samples (between 25% and 40% for all samples). Example of such report is shown in Figure 3.3.



```
{
        "n_targets": 190432,
        "n_bootstraps": 0,
        "n_processed": 18891467,
        "n_pseudoaligned": 6220555,
        "n_unique": 2251498,
        "p_pseudoaligned": 32.9,
        "p_unique": 11.9,
        "kallisto_version": "0.46.2",
        "index_version": 10,
        "start_time": "Fri Feb 21 19:44:31 2020",
        "call": "kallisto quant -i refs/GRCh38_correct.idx -o
kallisto_second/CR_6 CR_6_R1_001.fastq CR_6_R2_001.fastq"
}
```

Figure 3.3: Output from the *kallisto* alignment for the CR06 sample. Only 32.9% of all reads were aligned to the reference.

The low-mapping rate gave rise to doubts about correct usage of the tool and the correctness of aligned data. Therefore, we decided to verify the results by using another pseudo-alignment tool, the *Salmon* [87]. *Salmon* works on the same principle as *kallisto*, so it should reveal whether the low mapping is caused by data or incorrect usage of the *kallisto*. Again, the index was build, with the same reference file. Because of the memory requirements and alignment being time consuming, only one sample was selected, the CR06, because its relatively small size compared to other samples. After the alignment, the mapping rate of 26.66% was reported. That is even lower than the 32.9% mapping rate of *kallisto*.

3.3.2 Problem of low mapping rate

The low mapping rate of both kallisto and salmon (about 30% of successfully mapped reads) is not a good sign. To resolve, whether is it caused by incorrect usage of *kallisto* and *Salmon*, or by the reads itself, we performed alignment by *hisat2* [57] to the whole human genome. This alternative alignment approach can reveal, whether the low mapping is caused by reads coming from regions, which were not covered in the transcriptome, used in the *kallisto* and *salmon*.

We performed the alignment by *hisat2* only on the CR_06 sample, same sample as in the case of *salmon* test, and for the same reasons — relatively smaller size, which is even more important, as the *hisat2* is very memory demanding tool. We aligned the CR_06 to the GRCh38.p13 genome assembly, downloaded from Ensembl [123].

The *hisat2* produces a SAM file, which was then sorted to BAM file by *samtools* [63]. The resulting BAM file was them examined using the *QualiMap* tool [82][40], which can produce various information about the alignment, with the respect to provided genome annotation file (*Homo_sapiens.GRCh38.99.gtf* from Ensembl was used). The *QualiMap* reports the overall mapping rate of reads, and is able to infer the genomic origins of most of the reads. Note, that in comparison to *Salmon* and *kallisto*, *Hisat2* aligned reads to the whole genome, and *QualiMap* has therefore information about intronic and intergenic regions, which were not considered in the previously mentioned tools.

QualiMap reported overall mapping rate of 90.7% for the CR06 sample, with the mapped reads being divided based on the genomic origin, as shown in Figure 3.4 and Table 3.1.

Genomic origin	Number of reads	% of total reads
Exonic:	8,553,590	29.78%
Intronic:	17,926,307	62.4%
Intergenic:	2,246,591	7.82%

Table 3.1: Results of QualiMap [82][40] RNA-seq mapping quality analysis on the genomic origin of mapped reads.

The percentage of reads mapping to exonic region is 29.78%. This is approximately the same value as reported by *kallisto* and *Salmon*. This seems as an evidence, that the mapping rate of those two tools was reasonable, and

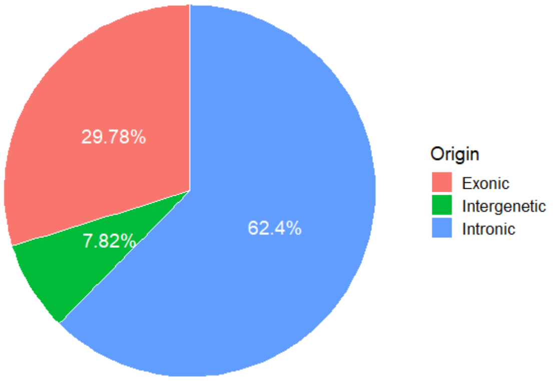


Figure 3.4: Genomic origin of RNA-seq reads, as reported by QualiMap [82][40]

most of the reads are coming from intronic and intergenic regions. Based on this evidence, we continue with the usage of measures count data as coming from the alignment performed with *kallisto*.

3.3.3 Comparison of DESeq2 results with transcripts and genes

As all of the above mentioned deconvolution tools work with expression data on the gene level, and the provided RNA-seq data have resolution on the transcript level, it is necessary to somehow aggregate the data to the gene level. In the previous section, we showed some rather naive approaches to this, which showed great variation in the data when performing the deconvolution.

There is a study by Soneson et al. [104], which argues that the differential analysis performed at the gene level gives results "appealing in terms of robustness, statistical performance and interpretation" and that "taking advantage of transcript-level abundance estimates when defining or analyzing gene-level abundances leads to improved DGE results compared to simple counting for genes exhibiting DTU (Differential Transcript Usage)" [104].

Now, with the help of *tximport* R package [104] by the authors of previously mentioned study, we can perform the aggregation of transcript-level estimates as outputted by *kallisto*. To assess the difference in differential expression analysis, we perform DGE analysis between CR and OT groups both on transcript and gene level (as obtained by *tximport).*

To compare the results, we will explore following properties of the DESeq2 (for more detailed description of DESeq2, refer to Section 5.1) results:

1. The number of found differentially expressed features (i.e., transcripts or genes) with multiple testing corrected p-value (or adjusted p-value) lower then set significance level of 0.1.

2. The distribution of p-values of differentially expressed features. This is shown in Figure 3.6.

3. The intersection of found features — for this, the transcripts are mapped to their corresponding genes. We compare top 100 vs 100, 500 vs 500, and 100 genes vs 500 top transcript with lowest p-values, as 36575 genes and transcripts 176126 were present in the differential expression analysis. This means, that there are approximately 5 transcripts per 1 gene. Venn diagram of the intersection is shown in Figure 3.7 and 3.8.

Discussion of results

The simplest metric of comparison is the number of significantly differentially expressed features. We set the significance level of 0.1 and count the number of features with adjusted p-values lower then that level. 71 transcripts and 48 genes were found significant. Note that this can't be interpreted as transcript level being more sensitive or accurate without any additional information. This can be merely used as an indicator of not equal results — 48 significantly DE genes should probably consist of more than 71 significantly DE transcripts (with average of 5 transcripts per gene). It is necessary to explore the difference further.

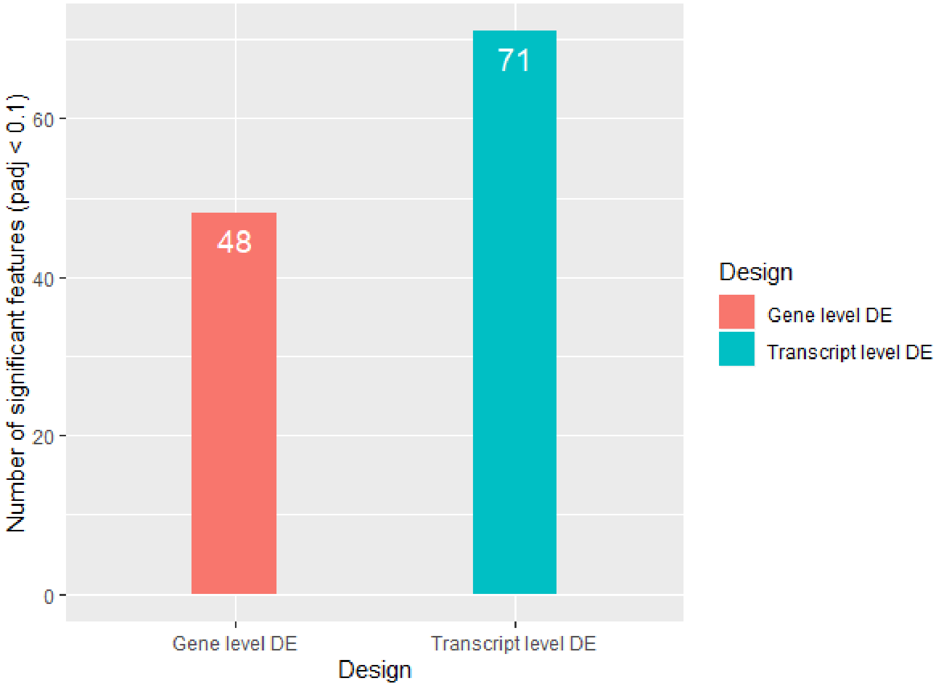


Figure 3.5: The number of significant DE features (with adjusted p-value <0.1).

We can take a look at the distribution of p-values. There are two distinct shapes of distributions of p-values in Figure 3.6. There is recognizable peak in the p-values close to zero n the gene's p-values distribution, with the transcript's p-values distribution seems to follow the same trend, except that in reverse — recognizable peak is at the highest p-values. This could lead to conclusion that gene level DE analysis produced more significant genes, but that would be misleading (as shown in Figure 3.5). The information about different number of transcripts and genes present in the analysis is omitted, so even there is visually more significant peak in the gene's distribution, it represents part of a much smaller set of genes.

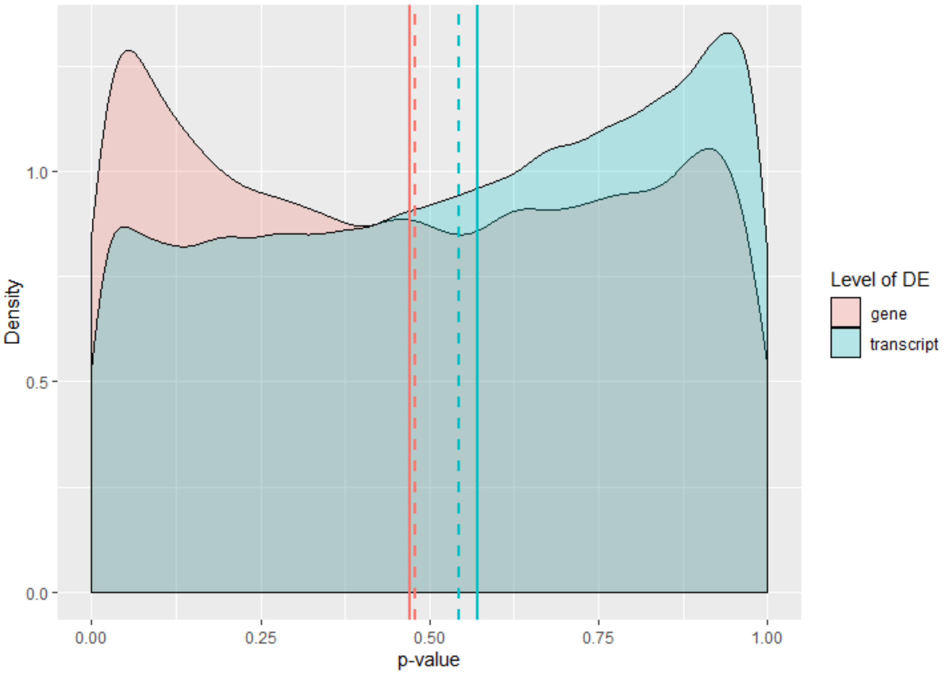
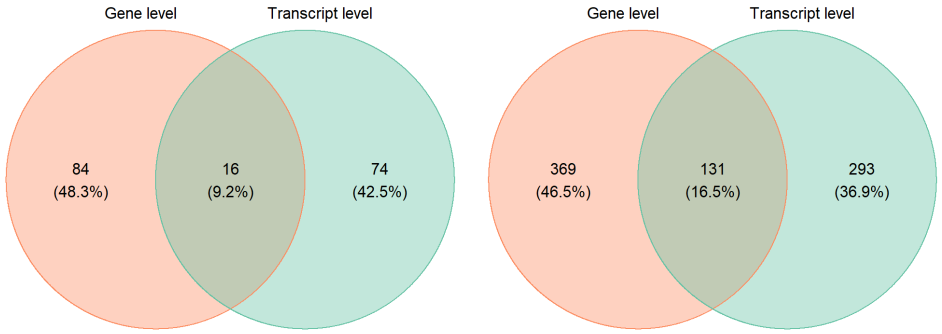


Figure 3.6: The distribution of p-values in DESeq2 output on transcript and gene level analysis. Dashed line marks the mean, and solid line the median of corresponding distributions.

In Figure 3.7, we explore the number of identically recognized significantly differentially expressed genes. The transcripts were mapped to genes and sorted by the p-value. When choosing top 100 genes from both approaches, only 16 were found identically. When 500 top genes were taken, 131 were found identically. Note, that when mapping transcripts to their corresponding genes, several transcripts could map to the same gene. In that case, out of the top 100 or 500 genes, only unique ones were chosen (that is why the sum of numbers for transcript levels does not sum to 100 and 500, respectively).

Surprisingly, the number of identically identified genes is quite low. This might be caused by the fact, that transcripts of one gene can be significantly differentially expressed, but when aggregated to the gene level, the total expression levels out and the gene by itself is not differentially expressed. On the other hand, several not DE transcripts may aggregate to gene, which itself is differentially expressed.

In Figure 3.8, we show comparison of top 100 genes and 500 transcripts mapped to genes, as in previous examples. The reason for the sizes of 100 and 500 is that there are approximately five times more transcripts than genes, i.e., five transcripts per gene in general — and in ideal situation, when a gene is DE and all its transcripts are too, would result in 5-to-1 ratio of



(a) : Top 100 DE genes and transcripts

(b) : Top 500 DE genes and transcripts

Figure 3.7: The comparison of uniquely and identically identified differentially expressed genes and transcripts. The transcripts were mapped to their corresponding genes. Both transcripts and genes were sorted by their p-value, lowest first.

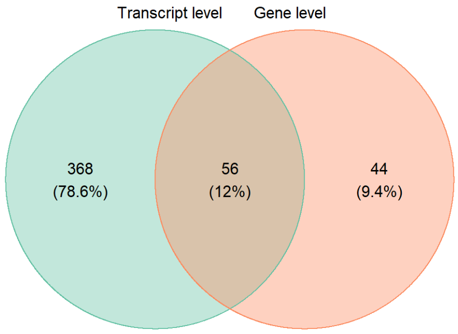


Figure 3.8: The comparison of uniquely and identically identified differentially expressed genes and transcripts. The transcripts were mapped to their corresponding genes. Both transcripts and genes were sorted by their p-value, lowest first. Please note the mirrored ordering of the transcript and gene groups compared to previous Figure 3.7.

significantly DE features. There are, however, still only 56 identical genes.

In conclusion, based on this simple comparison, we see that the results of DGE analysis differs greatly when performed on transcript and gene level. This example only illustrated and confirmed findings described by Soneson et al. [104]. As we are not able to asses the true underlying expressions and true sets of differentially expressed genes (they are not known for this dataset), we will perform the DGE analysis on the gene level, as recommended in *tximport*[104] and DESeq2 vignette [65]. The ideal result of this little experiment would be for the result on both levels to be the same — we would have an affirmation, that we do not lose information any information in the aggregation step. But it was not the case, and we resort to the procedures recommended and researched in literature.

Chapter 4

Deconvolution of gene expression profiles

In this chapter, we perform the deconvolution of 80 provided samples, using the methods selected in the previous Section 3.2. Below, we specify the setups and settings of methods used and give them unique identifying names for later reference in text and plots.

4.1 Application of deconvolution methods

For the **MCP-Counter**, **QuantiSeq** and **EPIC**, the R library *immunedeconv* [50] was used. For the **CIBERSORT**, the author's R implementation, which is available upon request, was used. CIBERSORT was run in two setups, one with the LM22 signature matrix and the second with the LM6 signature matrix. Both are available on the website of CIBERSORT [26]. We will refer to CIBERSORT with LM22 and LM6, as **CIBERSORT LM22** and **CIBERSORT LM6**, respectively. We also perform CIBERSORT deconvolution using the absolute mode, which measures the overall abundance of each cell type (as opposed to standard, which measures the fraction of cell type from cell types present in the signature matrix only [80]). This mode is marked as an experimental and was tested with LM22 [80], so we used it with LM22 and referred to it by **CIBERSORT LM22 (abs)**. All CIBERSORT related methods were run with the *quantile normalization* option turned off, as authors recommended for the RNA-seq data.

The **EPIC** was used with three different settings; the first, as implemented by *immunedeconv*, further called **EPIC**. Second, with default settings as present in EPIC R package [91], named **EPIC default**, and third, again using the R package, this time with manually setting the method to recognize blood immune cells as reference — further called **EPIC bref**. The **ABIS** deconvolution was performed using the web interface [76], downloaded and processed in R language.

The xCell deconvolution was performed by available R library [6] in two different modes — the first with the default settings, called **xCell**, and the second, which makes use of the xCell's ability to indicate cell types present in the mixture in advance. For this, we indicated the presence of cell types present in LM22, if possible (as LM22 consists of immune cell types present in the blood, and all of them can be expected to be present in our samples).

This setup of xCell will be referred to as **xCell guided**.

The QP and LLSR deconvolution was done using the implementation provided by *CellMix* R library [41]. Both of them were used with LM6 and LM22 signature matrices; we refer to these setups as **QP LM6, QP LM22, LLSR LM6, and LLSR LM22**. Additionally, we used LLSR with the gene expression profiles of immune cells as prepared by Abbas et al. [2], which were also present in the *CellMix*. This is further called **LLSR Abbas**.

All of the above-mentioned deconvolution methods were done on the gene level, and input gene expression data were normalized using the TPM method, as outputted by *tximport*.

Two deconvolution methods deserve to be described in more detail, as their application was not as straightforward as simply using the library, the **CIBERSORTx**, and **LinSeed**.

Deconvolution by CIBERSORTx

CIBERSORTx offers several modules based on the intended goal of the user. Similarly to the older CIBERSORT, it provides a module for custom signature matrix creation, but CIBERSORTx can create a signature matrix from scRNA-seq datasets (CIBERSORT could previously only use sorted RNA-seq datasets). Currently, scRNA-seq datasets are becoming widely available in online databases, for example in PanglaoDB [39] or Human Cell Atlas [93].

We used CIBERSORTx for the creation of a custom signature matrix. For this, we selected scRNA-seq dataset found in PanglaoDB [39]. The dataset is accessible under the NCBI SRA submission code of SRA749327 and NCBI SRS sample identifier of SRS3693911. The PanglaoDB offers the preprocessed dataset for downloading, along with the clustering results and cluster identification. All these were downloaded. The dimensional reduced (using t-SNE) and clustered dataset with identified cell types is shown in Figure 4.1.

The CIBERSORTx requires five gene expression reference profiles at a minimum for each cell type. For each of the nine cell types present in the datasets, we randomly selected ten replicates (note that this corresponds to the H matrix, as presented in the chapter on deconvolution formalization — each cell type is represented by several gene expression profiles). The data was further prepared for the format required by CIBERSORTx and uploaded to the website. The *create signature matrix* module was used at first, and the resulting signature matrix was used for deconvolution of the mixture data (the same as in the previous methods). The signature matrix visualisation is shown in Figure 2.15. The red color shows highly expressed genes, and the figure illustrates how the CIBERSORTx restricted the number of genes to include mostly these, which are significantly expressed in some cell type only — the red clusters could be interpreted as marked genes for given cell type. In the formalized deconvolution framework from Chapter 2.3, we can describe this CIBERSORTx feature as a transformation of the expanded signature matrix H to signature matrix G.

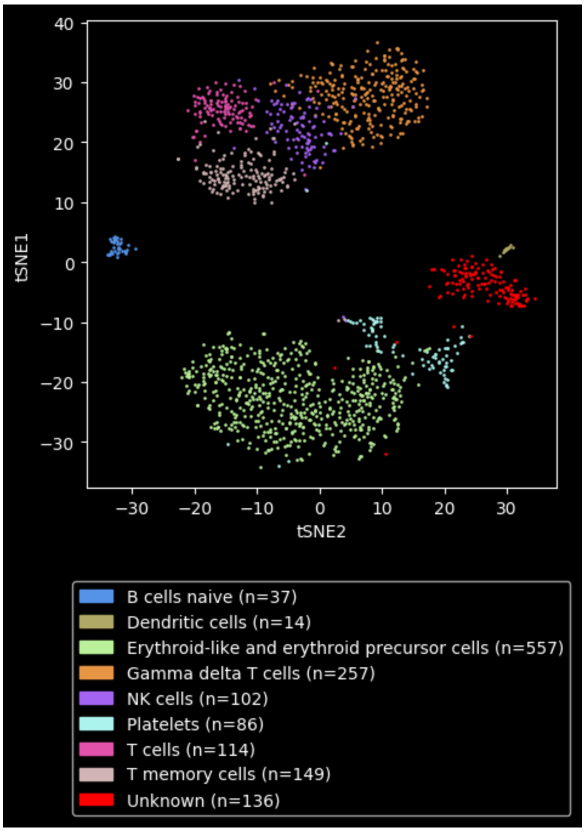


Figure 4.1: scRNA-seq dataset visualized after dimensional reduction and clustering, with eight identified cell types. This dataset was used for custom signature matrix creation by CIBERSORTx.

For the deconvolution, the *Impute Cell Fractions* module available on the website was used. The *S-mode* of batch correction was used (it corrects for the technical variances of RNA-seq input data and scRNA-seq based signature matrix). Deconvolution results were downloaded and processed using the R language.

Deconvolution using LinSeed

For the LinSeed usage, we used the R implementation available on GitHub [64]. The number of cell types present in the mixture was set to eight (based on the plot of explained variance provided by the tool, where eight cell types explained 90% of the variance). As the tool performs complete deconvolution, see Equation 2.9, both the cell-type-specific gene expression profiles and their corresponding proportions in the mixture are estimated. The problem lies in identifying the true cell types belonging to the reference GEPs. We employed two approaches for this:

- Correlating unknown GEPS to known profiles from signature matrices, namely LM22 and LM6.

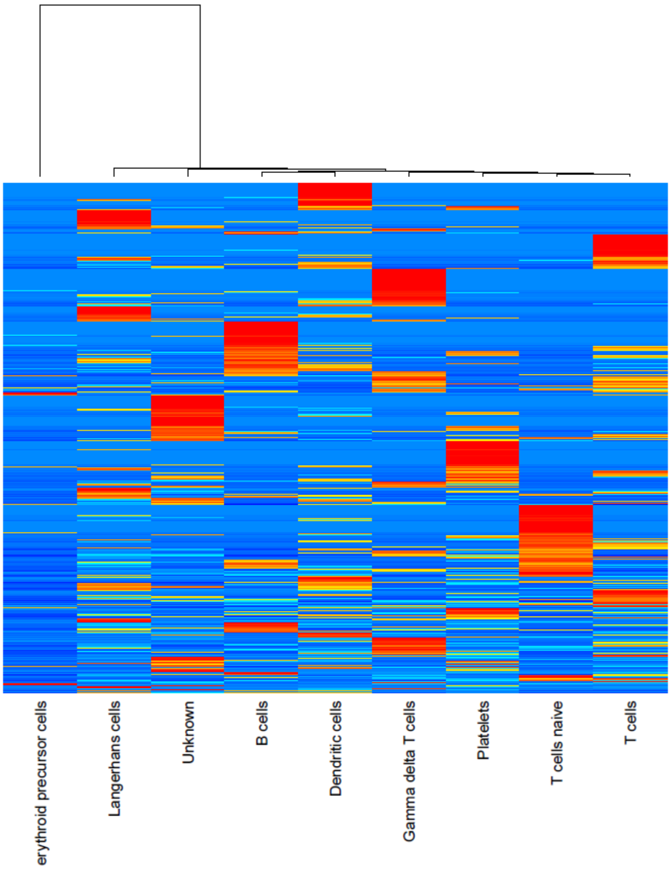


Figure 4.2: Visualization of custom signature matrix, as produced by the web interface of CIBERSORTx [27]. The recognized cell types were determined by the used scRNA-seq dataset from PanglaoDB [39]. Notice the red clusters, corresponding to probable markers for given cell types.

- Extracting marker genes for each unknown GEP and use these marker genes for identification.

For the correlation of GEPs, we choose two metrics; the Pearson correlation and Spearman rank correlation coefficient. Both Pearson and Spearman correlation coefficients of LinSeed cell types and LM22 and LM6 are reported in Figure A.2 and Figure A.1, enclosed in the Appendix A.

In Figure A.1, we see several candidates for mapping of LinSeed cell types to LM6 cell types. There is moderate positive Pearson correlation between *Cell type 2* and *NK.cells* ($r = 0.55, p\text{-value} \leq .001$) and between *Cell type 3* and *B.cells* ($r = 0.50, p\text{-value} \leq .001$). *Cell type 8* seems to correlate with both CD4+ ($r = 0.54, p\text{-value} \leq .001$) and CD8+ cell types ($r = 0.46, p\text{-value} \leq .001$). It might therefore represent T cells in general, with not enough resolution to distinguish between more T cell subtypes. Similar trend is present in the plot of Spearman correlation coefficients, with neutrophils having high positive correlation ($\rho = 0.81, p\text{-value} \leq .001$).

The evidence of this probable mapping is further strengthened when looking at Figure A.2, where Pearson's coefficient reveals moderately high correlation of *Cell type 3* with all subsets of B cells, *Cell type 2* with NK cells, *Cell type 8* moderate to high correlation with T cell subsets and *Cell type 7* with neutrophils and monocytes (exact values not shown, available in Figure A.2). This would indicate that *Cell type 7* might represent their nearest common ancestor, myeloid cells, see Figure E.1.

For the second approach to mapping, we employed a simple strategy: for each cell type, the gene expression profile was sorted from mostly expressed to the lowest expressed genes. We selected the top five genes, having at least five times higher expression than any other cell type. Those five selected genes were then compared with cell marker genes from *CellMarker* [127]. For the three above suggested mappings (to B cells, T cells, and NK cells), all five of the top genes were reported as markers for these cell types.

Therefore, we mapped three cell types GEPs found by **LinSeed**, which we now assume to represent B cells, T cells, and NK cells.

We also used the above-mentioned method of cell markers identification to check whether one of these GEPs could represent erythrocytes or platelets, which could possibly constitute to a large part of the mRNA present in the blood samples and confound the deconvolution results (this concern were discussed before in Chapter 3). We used markers, having at least two evidences, from PanglaoDB: CD235a, CD24, CD45, GlyA, and Ter119 for erythrocytes and CD41, CD61 and CD62P for platelets. None of these were even present in the found GEPs, which seems to support the claim, that cell types other than PBMC do not play a significant role in gene expression in these samples.

4.2 Results of deconvolution

In total, eighteen methods were applied to the data. As we do not know the ground truth of fractions of cell types present in the sample, we can only compare the obtained results between each other. There are, however, problems with comparison and interpretability of the results. To be specific, two main problems complicate the comparison:

1. Each method has a different set of recognized cell types (see Table E.1). For example, we cannot compare results on erythrocytes proportion in samples, as only xCell provides these estimates. Connected with this issue is the fact that some method provides results on the different detail levels, for example, LM22 recognizing seven types of T cells, while LM6 only two.

2. The results are not reported in the same units. Some methods report results as percentages, some as scores. They are thus not directly comparable.

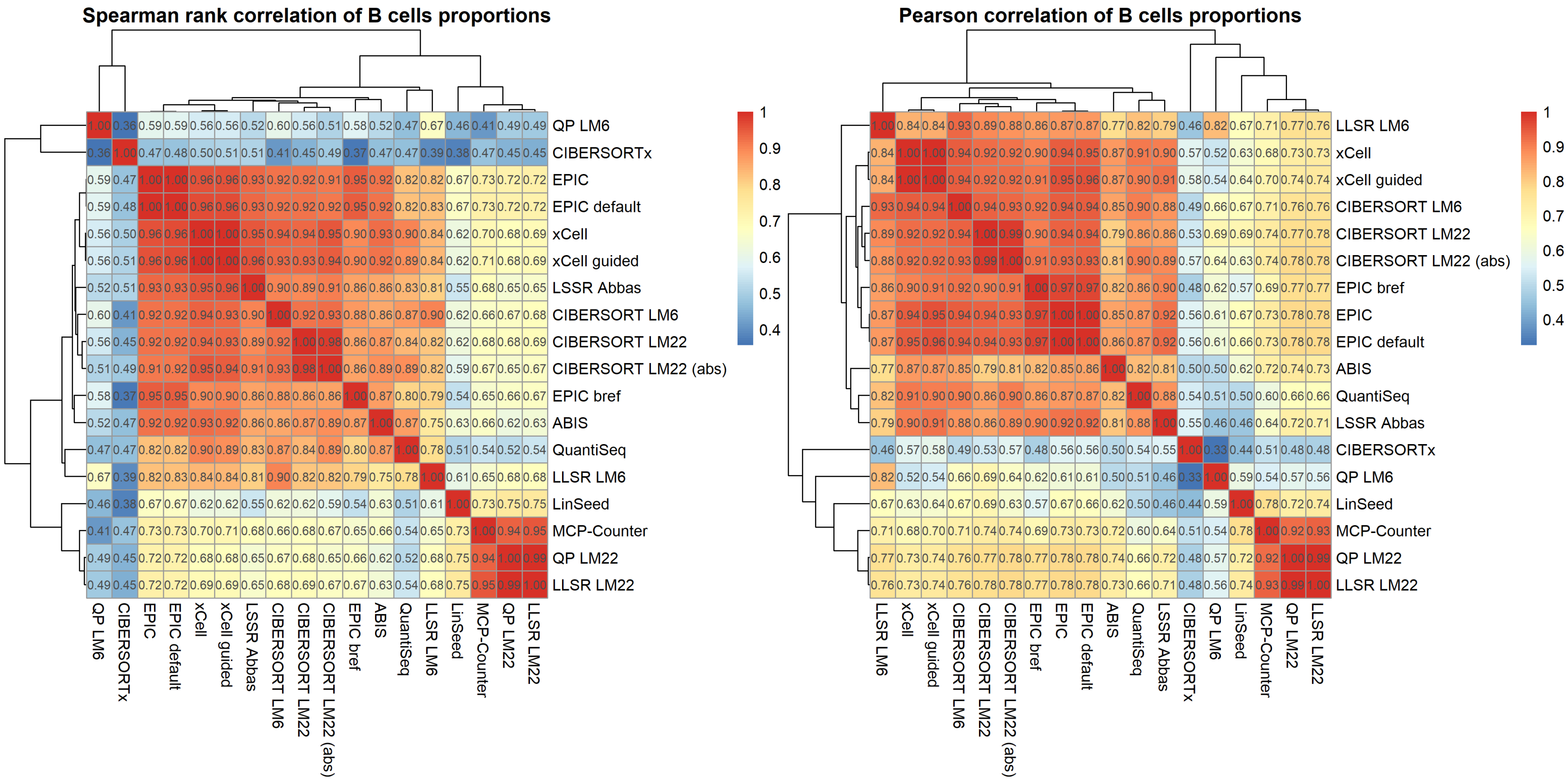
To deal with the problems mentioned above, we employed the following approaches. The first problem of different recognized cell types is dealt with by the help of R package *immunedeconv* [50]. Other than providing a unified interface to some deconvolution methods, it presents a complex hierarchical characterization of different cell types present in these methods, for more details see Appendix E.1 and Table E.1. It further implements an algorithm for summarizing deconvolution results to a given level of cell-type resolution. In other words, it can produce a score or fraction of T cells present in a sample for a method, even though the method provides scores or fractions for different subtypes of T cells.

Based on this tool, we compare results of deconvolution on the level of cell-type resolution, which is ideally present in all methods. This is restricted by the signature matrix LM6, which has the lowest number of recognized cell types. Thus, we compare the deconvolution results of fractions and scores of **B cells, NK cells, CD4+ T cells, CD8+ T cells, monocytes, and neutrophils**. In methods where higher cell type resolution is available, the summarized value is obtained by *immunedeconv* or, if not applicable, by summing all values of cell subtypes belonging to the parent cell type, based on the hierarchical tree defined by *immunedeconv*, see Figure E.1. If some of the above-mentioned cell types are not available in a method or signature matrix used by the method, the corresponding method is left out of the comparison for given cell type.

The second problem of different used units is solved on two levels and is based on the fact that all of the methods produce results, which are comparable across samples. On the visualization level, we scale all the values for one cell type to range $[0; 1]$. This helps with the visual comparison of found fractions and scores in different samples, as it preservers the ranking order of one cell type across samples. These scaled values are shown in Figure B.1, B.5, B.8, B.16, B.13 and B.10. We also show the same plots for methods, which produce values resembling percentages and are thus not adjusted in any way. These plots are shown in Figures B.2, B.4, B.7, B.17, B.14 and B.11. All of the above mentioned figures are enclosed in Appendix B.

For the comparison of results for given cell types, we compare used methods by the Pearson correlation, but as some methods do not guarantee that the values are on a linear scale, we use the Spearman rank correlation as well. We show an example of such comparison in Figures 4.3a and 4.3b. It is presented as a heatmap, along with a hierarchical clustering using the euclidean distance. This representations helps to reveal similarly performing methods, both visually (by clusters of similar colors) and by hierarchical clusters. For example, based on Figure 4.3a, we see that all CIBERSORT versions, version of xCell and EPIC form a cluster.

Complete results are enclosed in attached supplementary materials, available as high-resolution images and interactive HTML plots, which allow for more clear inspection of computed results.



(a) : Spearman rank correlation of found fraction of B cells by selected deconvolution methods

(b) : Pearson correlation of found fraction of B cells by selected deconvolution methods

Figure 4.3: Spearman's and Pearson's correlation coefficients for the computed fractions of B cells.

Chapter 5

Incorporating the deconvolution results to DGE analysis

Having the deconvolution results of our 80 samples, the question of incorporating them into the differential gene expression analysis process arises. Usual input to the DEG analysis consists of a matrix of count data and metadata table — with sample features, such as grouping, condition, age. . . .

5.1 DESeq2

DESeq [65] is a tool used for differential gene expression analysis. It models RNA-seq counts with the negative binomial (NB) distribution. It allows for complex design specifications and is accompanied by rich documentation, which is why we selected it for the following experiments. It is available as an R library.

DESeq2 model

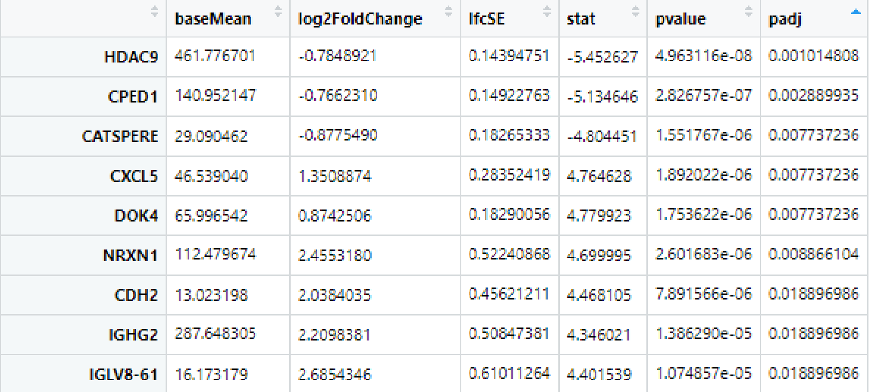
To get a brief insight into the working of DESeq2, we give a short description of the DESeq2's model. The tool models the RNA-seq count by a negative binomial generalized linear model (GLM). To be precise (and using the notation from DESeq2 paper [65]), the RNA-seq read counts K_{ij} for gene i in sample j is modeled by GLM from the negative binomial family (parametrized by mean and dispersion parameter), with a logarithmic link [88]:

$$K_{ij} \sim NB(\mu_{ij}, \alpha_i), \quad \mu_{ij} = s_{ij} q_{ij}, \quad \log q_{ij} = \sum_r x_{jr} \beta_{ir}, \tag{5.1}$$

where μ_{ij} is the fitted mean for gene i and sample j, α_i is the gene-specific dispersion. The s_{ij} is normalization factor, as described in Section 2.7 on DESeq normalization, $q_i j$ is a value proportional to the true concentration of fragments from gene i in sample j (which is closely related to the ν quantity, described in Section 2.2.3), x_{jr} and βir are explanatory variables and their coefficients, respectively. Finally, r is the number of covariates in the design matrix of the model [65].

DESeq2 output description

To properly interpret obtained results from DESeq2, we need to describe the results as outputted from the tool. In Figure 5.1 we show an example of the tool's output for DGE analysis between two conditions, lets call them A and B, where A is the reference condition. The output consists of a row for each gene, which is present in DGE analysis. The *baseMean* is the mean of normalized counts in reference condition. The *log2FoldChange* is the estimated log-2 fold change between conditions, computed as $\log_2\left(\frac{m(A)}{m(B)}\right)$, where $m(A)$ and $m(B)$ are the means of normalized counts in condition A and B, respectively. The *lfcSE* gives the standard error for log-2 fold change. The *stat* is by default the value of Wald statistic (which is, in this case, *log2FoldChange* divided by *lfcSE*. The *stat* value is then compared to standard Normal distribution to produce *pvalue*. The *padj* is *pvalue* corrected for multiple testing by default by Benjamini & Hochberg (BH) method [13].



	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
HDAC9	461.776701	-0.7848921	0.14394751	-5.452627	4.963116e-08	0.001014808
CPED1	140.952147	-0.7662310	0.14922763	-5.134646	2.826757e-07	0.002889935
CATSPERE	29.090462	-0.8775490	0.18265333	-4.804451	1.551767e-06	0.007737236
CXCL5	46.539040	1.3508874	0.28352419	4.764628	1.892022e-06	0.007737236
DOK4	65.996542	0.8742506	0.18290056	4.779923	1.753622e-06	0.007737236
NRXN1	112.479674	2.4553180	0.52240868	4.699995	2.601683e-06	0.008866104
CDH2	13.023198	2.0384035	0.45621211	4.468105	7.891566e-06	0.018896986
IGHG2	287.648305	2.2098381	0.50847381	4.346021	1.386290e-05	0.018896986
IGLV8-61	16.173179	2.6854346	0.61011264	4.401539	1.074857e-05	0.018896986

Figure 5.1: Example output of DESeq2, sorted by *padj*.

Later, when comparing results coming from different models, we will be interested mainly in the number of genes under some selected threshold for adjusted *p*-values *padj* and the overall distribution of all p-values.

5.1.1 Incorporating deconvolution results to DESeq2

DESeq2 is ready to incorporate any factors (or variables) into the experimental design. In the vignette of DESeq2 R library [66], several examples and general recommendations are given. In general, the same approach as when including batch control variables can be used. For the further description, we will use the R Formula, a compact symbolic form of a model specification. The R formula uses the general form of "$y \sim model$", with the meaning of response variable y being modeled by *model*, where *model* is a series of terms (in our case, independent or explanatory variables) joined by plus (+) signs. Terms (or variables) can also be joined by a colon (:), a symbol with a special meaning of *interaction* of all interaction between joined variables.

In the context of DESeq2, which requires for its design to be specified by formula, the typical design would be "$sample \sim cond$". This design specifies that the samples are modeled using *cond* categorical variable, which represents various conditions, in which the samples were taken. DESeq2 allows for more complex designs, for example "$sample \sim tech + cond + tech{:}cond$". This design would model the samples by categorical variable *tech* (having two values, e.g., depending on the technology of sample collection) and *cond* categorical variable. The *tech:cond* models the possibility that the condition effect is different across sampling technology. We further adapt the DESeq2 notation of specifications of models — the left-hand side of the formula is not written out, e.g., only "$\sim tech + cond + tech{:}cond$" is written instead.

Using the above-described terminology, we can explore the possibilities of including deconvolution results in the DESeq2 design. In the R library vignette, authors explicitly mention the possibility of adding an arbitrary number of variables to the design, both categorical and continuous (although authors recommend converting the continuous variable to categorical by 'cutting' it into a small number of bins, ideally 3-5).

As a reminder, by performing the deconvolution, we got several additional values (based on the number of recognized cell types), representing the fraction (or score) of specific cell types present in the sample. These can be understood as a realization of a continuous random variable. For each sample, we, therefore, have the group information and, e.g., six continuous values of cell type fractions from LM6.

5.2 Experiment setup

We have several options when preparing the DGE experiment. At first, we have to decide between which groups of samples will the analysis be performed. There are five groups to which the samples are distributed — OT, CR, CyA, STA, and HC. Based on the discussion with supervisor of this thesis, the DGE analysis between CR and OT groups were deemed interesting from the biological point of view; therefore all the following experiments (if not stated otherwise) are performed as DGE analysis between those two groups, with the intention to discover biomarkers specific to those groups. Note, that all 80 samples are inputted in the DESeq2 tool, as recommended by authors in the R library vignette [66] — the model can benefit from these data, even though they are not directly used for the resulting comparison.

It is important to note that there is no ground truth available to our data, so we cannot compare the results absolutely (i.e., answer the questions like 'Does the method reveal true biomarkers?' or 'Does this method find more true biomarkers?'). We can compare different methods and the design relatively, between each other, with the intention of examining the effect which different designs have on the result. We will also compare the results to the 'baseline' result, DESeq2 with model $\sim group$. In other words, the model without any additional information coming from deconvolution.

5.2.1 Experiment results comparison

With the previous remarks in mind, we compare the results in the following ways:

- By the number of found significantly DE genes, based on adjusted p-value threshold, see Figure 5.2.
- By the number of uniquely and identically found genes in the top N genes (sorted by p-values). This is visualized by Venn diagrams, see Figure 5.3.
- By similarity of the total ordering of genes, sorted by p-values. This was measured by the Spearman rank correlation coefficient, see Figure 5.4.
- To assess the differences of results from the point of known gene annotation, we perform pathway analysis and gene set enrichment analysis. We focus on differences between significantly enriches terms, eventually on differences of their significance level.
- By comparing the overall distribution of p-values computed for genes by DESeq2, similarly to the experiments performed in Section 3.3.3. The comparison of absolute p-values would not make sense, as different models have different null hypotheses, for which is the p-value computed. Note that the comparison using these plots can also misleading, for example, as shown in the section on gene vs. transcript level DGE analysis. It does not take into account the overall number of filtered genes, computation of adjusted p-values, and independent filtering procedure of DESeq2.

5.2.2 First experiment

When designing a model for DESeq2, we have to make several decisions:

1. Which deconvolution method's results to use.
2. Which and how many cell types proportions should we use in the design.
3. How to deal with the variable representing the proportions — leave it as it is (continuous), or 'cut' it into a categorical variable with some particular number of levels.
4. How to combine variables in the design, i.e., how many of them, with or without interaction, etc.

To illustrate the previous point, consider the following description of designs, which incorporates the B cell proportion coming from xCell. The xCell is selected as a representative method, because it is commonly in the biggest cluster of similarly performing methods (see Chapter 4.2) and at the same time, it is a method fulfilling assumption for deconvolution of our data. We describe four different models, using different forms of the xCell proportion variable, along with the basic design of $\sim group$. The ideas described in these models (specifically the reasoning behind 'cutting' the continuous variables into categorical) will be applied to further experiment designs. The five models are:

- The baseline design, without any additional information from deconvolution. DESeq2 is supplied with "$\sim group$" model.

- Using the "$\sim bcell\_2 + group$" model, where *bcell_2* is a variable taking two values. The original output from *xCell* was 'cut' into two parts based on a threshold, so both groups are of equal size. The two values of the variable can be understood as an indication of either *low* or *high* proportion of B cells in the sample. Two groups were selected because DESeq2 internally uses different methods for category variables taking only two values [65].

- Using the "$\sim bcell\_3 + group$" model, where *bcell_3* is a variable taking three values. The *bcell_3* was obtained in an analogous way to the previous design. The number of groups is selected on the basis of the recommendation of DESeq2 authors, where three is the lower bound on recommended values the transformed categorical value should take.

- Using the "$\sim bcell\_5 + group$" model, where *bcell_5* is a variable taking five values. Analogous to the previous two transformations, five is the upper bound on recommended values the transformed categorical value should take.

- Using the "$\sim bcell\_cont{+}group$" model, where *bcell_cont* is a continuous variable, taking values from the $[0; 1]$ range. This corresponds to the complete information, as obtained from *xCell*.

The notation of *celltype_K* will further be used for a continuous variable, transformed to categorical by dividing its values into K equally sized groups, e.g., *cd4tcell_3* will be variable, representing the CD4+ T cell proportion in the sample, with three-level resolution. By *celltype_cont*, we mean the original continuous variable, as outputted by deconvolution method. The particular method will follow from the context.

Let us discuss the results obtained from this first example. In Figure 5.2, the numbers of significantly differentially expressed genes are shown. As to the matter of significance, we consider genes with adjusted p-value <0.1 significant. We can see that compared to the baseline method, the additional B cell fraction information factored to two levels lowered the number of significant DE genes. However, while adding more resolution, the number of significantly DE genes seems to be getting higher, with the higher number being 66 for unfactored continuous variable. It might be interesting to find out if this trend will be observed in general, with proportions of other cell types added.

For the comparison of the actual set of genes found and their intersection between methods, we use the Venn diagram, as seen in Figure 5.3. Here we

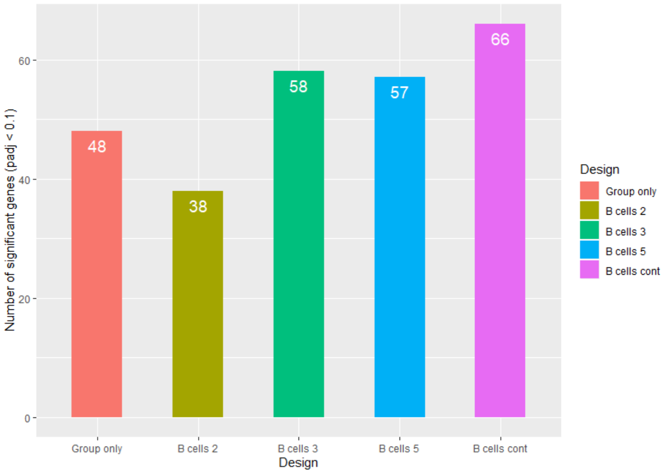


Figure 5.2: The number of significant DEGs (with adjusted p-value <0.1) for five different DESeq2 designs.

compare the top 100 most significant genes (when sorted by p-value). Exactly 50 out of the 100 genes were identified by all methods, with each method having found some genes, which were not identified by any other method.

5.2.3 Experiments

So far, we have performed an addition of xCell's B cell proportions to the model. Let us now properly define a series of experiments. We will examine the following questions:

- **Will the results differ for B cell proportions coming from different methods?**
- **Will the trend in the number of found signature genes be the same when using other cell type's proportions?**
- **What is the effect of including more additional variables with different cell type's proportions?**
- **Is it possible to include all variables coming from one deconvolution method at the same time?**

For each of these questions, we carry out a single experiment focused on answering the question.

Experiment 1 — B cells proportions coming from different methods

We examined the effect of including B cell proportions coming from xCell in various ways to the DESeq2 design. We perform the same experiments with B cell fraction coming from different methods for the crude evaluation of the robustness of obtained results. We selected results coming from **CIBERSORT LM6** and **EPIC** methods, based on Figures B.3a and B.3b.

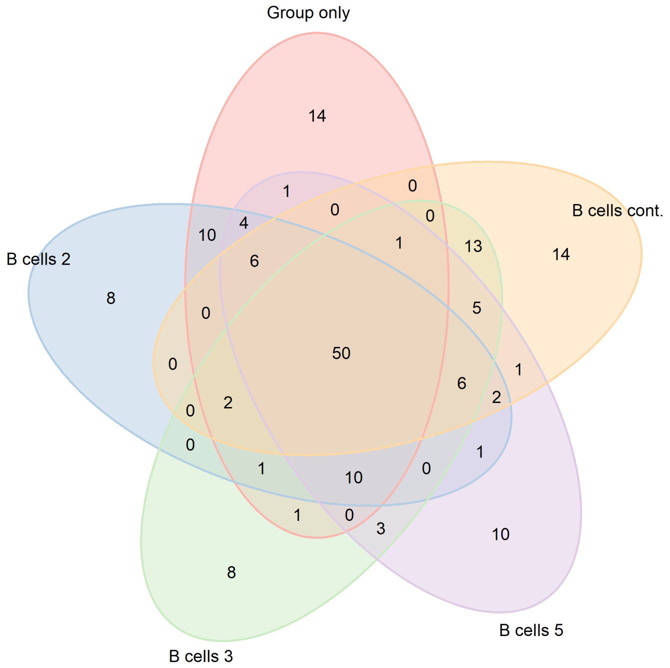


Figure 5.3: Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by different methods

The **xCell**, **CIBERSORT LM6**, and **EPIC** seems to be in the same cluster of similarly performing deconvolution methods (with respect to the B cells proportions). Also, the focus on B cells in this example stems from the reported significance of B cells in the kidney transplantation research [56][74].

The setup of the experiment and way of incorporating the deconvolution is the same as in the introductory example, see Section 5.2.2. Proportions of both methods are used in four different ways, three of them based on the number of levels in the transformed categorical variable, one with the original continuous values.

We start the comparison by comparing number of significant DE genes for all three origins of B cell proportion results (Figures 5.2, C.2a and C.1a). The results of **xCell** and **CIBERSORT LM6** seems to follow similar trend, with the *b_cells_2* variable even having the same number of genes. However, with the **EPIC** origin, results look different — there seems to be a very small difference in terms of significant DE genes amount when including the B cell fraction in the categorical variable. Only when including it in unchanged continuous form, the number goes up. Some similarities can be observed in the number of DE genes when including the variable in the continuous form — all three data origins, it is the highest number of all designs, and the values are relatively close to each other (66, 74, 75).

When comparing the results in terms of uniquely and identically identified genes, the results are mostly different too. This time, the **EPIC** and **xCell** have the same number of identically identified genes (Figures 5.3 and C.1b), with the numbers varying for the rest of designs. In conclusion, even with

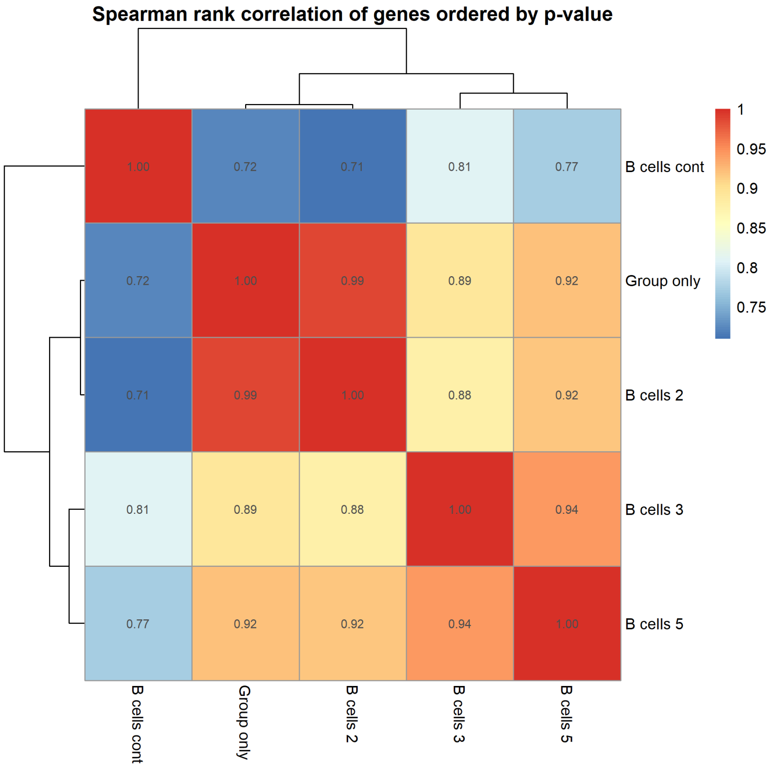


Figure 5.4: Spearman's rank correlation of gene ordering for five different DESeq2 designs, with B cell proportions coming from **xCell**.

taking fractions of B cells in samples from those methods that generally perform similarly (based on Pearson's and Spearman's correlation), the results are not very consistent. There is a similarity in the fact that the highest number of significantly DE genes is highest when using the fraction variable in its continuous form. Also, the number of uniquely identifies DE genes is highest for the models with a continuous fraction variable. This might indicate that the deconvolution information is most informative to the DESeq2 model when incorporated as a continuous variable. This is, however, in contrast with the recommendations of DESeq2's authors.

Experiment 2 — Proportion of other cell types

We will now focus on incorporating cell types other than B cells to the DESeq2 design. For simplification of the experiment, we will consider results coming only from one deconvolution method. Based on the results and figures from Chapter 4, we select **CIBERSORT LM6** as this method. For all cell types, this method is part of the biggest cluster (confirmed only visually) of similarly performing deconvolution methods, which makes it a good candidate for a representative method.

We will perform the experiment on three additional cell types: **CD4+ T cells, CD8+ T cells, and NK cells**. These three cell types were selected randomly out of five remaining not yet examined cell types from **CIBERSORT LM6**, as the examination of all cell types computed by all methods is not easily feasible. For each of them, we will explore all four options of the variable handling, as shown in previous experiments, and compare it to the baseline design, with no additional variables.

We report the results using the same types of plots, as in previous examples. All the plots are reported in Appendix C.

The results for **CD4+ T cells** are shown in Figures C.3a, C.3b and C.3c. Surprising result are seen in Figure C.3a, where the same number (172) of significant genes was identified for both for models with *cd4cell_3* and *cd4cell_cont*, even though the lists of sorted genes and p-values are different.

The results for **CD8+ T cell** are shown in Figures C.4a, C.4b and C.4c, results for **NK cells** are shown in Figures C.5a, C.5b and C.5c. The highest number is observed in *celltype_3* or in *celltype_cont* designs, although there seems to be no clear trend in the number of significant DEGs — for example, for the NK cell variable, the lowest number of significant genes was found for the *nkcell_3* design. We again emphasize that the the number of significantly DEGs without any ground truth does not assess the design as better or worse compared to baseline design, i.e., we are not able to assess the true sensitivity and specificity of the method.

We also point out a similar number of identically identified genes (42,38 and 44) for all models of given variables, which is also relatively consistent with the values obtained in Experiment 2. Overall, apart from the identically identified genes, there seems to be no clear trend or consistency when using the B cell fractions from similarly performing methods.

Experiment 3 — Proportion of multiple cell times at the same time

For the third experiment, we will explore the effect of combining several cell type's proportions at the same time. This is not a very rigorous experiment, as the previous experiments showed very varying results, and therefore this should be understood mainly as an exploratory experiment. For the designs, we selected data from **CIBERSORT LM6**, as in the previous examples. Because the designs with continuous variables seemed to provide a high number of significant DE genes (but that does not automatically mean it was a better result, as the null hypothesis of the model might be different from other designs), we decided to use this form of fraction variable.

As it is not feasible to try out all possible combinations of variables from **CIBERSORT LM6**, we restricted ourselves to the following combinations (all combinations of size two from the cell types presented in Experiment 2 in continuous form):

- " $\sim$ *nk_cont* + *cd4+_cont* + *group*"
- " $\sim$ *nk_cont* + *cd8+_cont* + *group*"
- " $\sim$ *cd4+_cont* + *cd8+_cont* + *group*"

We compare DGE analysis results of these designs between each other and with the baseline " $\sim$ *group*" design. The results are presented in Figure 5.6. There are visible differences, for example, in the number of significant DEGs, where the combination of NK and CD4+ T cells created 625, and NK and CD8+ T cells 77 genes. Also, when looking at the Venn diagram in Figure 5.6b, we can see that there was no overlap in the top 100 most significant DEGs for all compared designs.

Experiment 4 — Full model, all continuous variables from LM6 added

For this experiment, we will include the full deconvolution information, as available from the CIBERSORT LM6. This means, that we will model the expression with the rather complicated DESeq2 design of " $\sim$ *bcell_cont + cd4+_cont + cd8+_cont + nk_cont + mono_cont + neutro_cont + group*".

When attempting to run the DESeq2 with the design, the error message "Model matrix not full rank", which in the documentation of DESeq2 is described as "One or more variables or interaction terms in the design formula are linear combinations of the others and must be removed". This is somehow surprising, as all variables are continuous, and the presence of linear combinations is highly unlikely. There is probably some inner reason in the DESeq2 for this not to be applicable. Further experimenting with the design reveals that omitting any of the added continuous terms from the design above allows for the DESeq2 to run. However, the tool reports that the fitting of the NB model did not converge. It is also possible that the number of parameters to be estimated is larger than the number of samples, and the system is under-determined or that there is some possible near-collinearity of added variables, which complicates the model fitting.

When inspecting the model's design matrix, it was found why the matrix was not full rank. There was a linear dependence — the implicit intercept term (represented by columns of ones) was a linear combination of all the continuous variables. As they are all coming from CIBERSORT LM6 and represent fractions, they sum up to one for each sample. Therefore, the sum of columns of the design matrix belonging to these variables equals to column composed of ones, which is the implicit intercept term.

To complete the experiment, we build one more model, including all the variables from LM6. However, this time, in the form of a binary variable, where the values are divided into two equally sized groups based on a threshold (as described in the first experiment). In other words, we use the " $\sim$ *bcell_2 + cd4+_2 + cd8+_2 + nk_2 + mono_2 + neutro_2 + group*" design. This time, the DESeq2 was able to run without any problems, and all coefficients of the model converged.

The results are compared to the baseline model of "$\sim$ *group*". We compare them in the same manner as in the previous experiments; the number of significantly DE genes was 48 and 91 for the basic and full model, respectively. The Spearman's correlation rank between ordered genes was 0.51. Out of the 100 top genes from both designs, 35 were identically identified. We also show the distribution of p-values for all genes, as reported by DESeq2. It seems like there is a lower number of genes with low p-value in the full model, which seems to not go well with the reported number of significant DE genes. This can probably be explained by the fact that the significance of genes is based on adjusted p-value, which is also influenced by DESeq2's independence filtering [65]. This is another reason why this kind of comparison is hard to interpret.

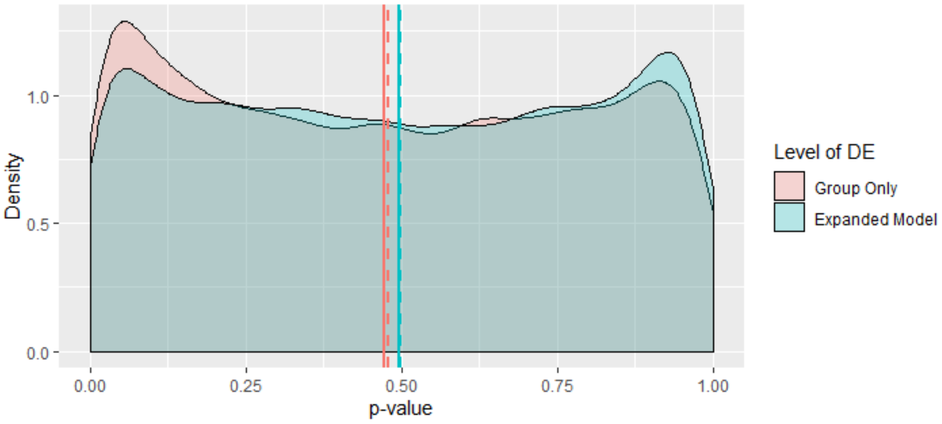


Figure 5.5: Comparison of distributions of p-values of results from both the basic model (with only group variable) and the full model (called Expanded Model in Figure 5.5) with all LM6 variable added in binary form. Means are marked by dashed, and medians by solid line.

5.2.4 GSEA of DESeq2 results

Apart from the described measures of comparison, we employed one more way of looking at the results — from the point of gene set enrichment analysis (GSEA). We are mainly interested in whether the designs used in the previous experiments—which often produced very different results when comparing them based on our proposed metrics —will produce similarly varying results when interpreting the results using the GSEA.

There are many ways of employing the GSEA, but we decided to use it for the enrichment of GO terms, using the *gage* R library [69] (details in the usage are reported in the Appendix D). We performed the GO terms enrichment on the results obtained by DESeq2 on designs used in Experiment 1 and Experiment 2. For the GSEA, the genes were sorted by p-value, and the $\log_2$-fold change was used as input metric.

The results of the GSEA are reported in Appendix D. Results are reported in tables in the following way: for each design used, the top six over-represented GO terms are reported along with their p-value and adjusted p-value, corrected for multiple testing. We do not report the under-represented GO terms, in order to make the resulting tables more comprehensible — the results and following discussion might thus not be that comprehensive, although the methodology for obtaining under-represented GO terms is the same and the result should follow similar trends.

The results for different designs are presented in Table D.2 for B cells from CIBERSORT LM6, in Table D.3 for B cells from EPIC, in Table D.1 for B cells from xCell, in Table D.4 for CD4+ T cells from CIBERSORT LM6, in Table D.5 for CD8+ T cells from CIBERSORT LM6, and in Table D.6 for NK cells from CIBERSORT LM6.

When examining the tables, the first thing to notice is that the found significant GO terms almost do not differ between different models. There is some occasional position switch or an introduction of new GO term to the top 6. A more substantial trend, present in results for all methods, is in the changes of p-value and adjusted p-value. Given a cell type, with the increasing complexity of the model (with the increasing number of recognized levels of the categorical variable, up to the continuous variable), the p-values and adjusted p-values increase up to a point, that except *CD8+cell_2* from CIBERSORT LM2, all the GO terms would not be found significant on the significance level of 0.1.

In conclusion, it seems that the top found GO terms are not that much influenced by the incorporation of additional variables, although the p-values and adjusted p-values seem to increase dramatically, deeming the found terms insignificant.

5.2.5 Summary of experiments

In conclusion, we showed several ways of incorporating the deconvolution results into the DESeq2 design. When describing the setup of the experiments, we have stated several questions, we can approximately answer now. We took B cell proportions from three different deconvolution methods, which were found to provide highly correlated results. We applied the B cell fractions to the design of DESeq2 in a way analogous to including the batch correction variables. Even when incorporating results from different deconvolution methods in exactly the same way, we obtained very different results in the DGE analysis. This supports the thesis that the DGEA is very sensitive to the additional variables incorporated in the design; therefore, the correctness of used deconvolution results is of high importance. This place demands on the well-founded and supported deconvolution approaches.

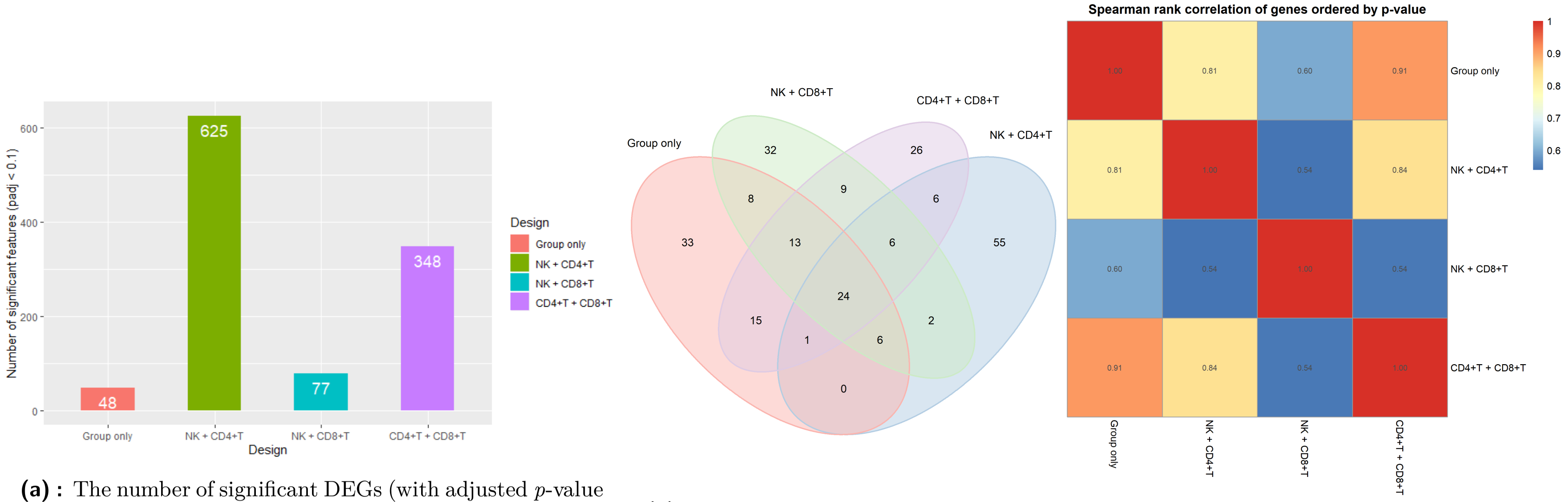
We further explored if adding fractions of three other cell types other than B cells will reveal some general principles or trends in the observed comparison metrics. It seems that in general the *celltype_3* or *celltype_cont* provides the highest number of significant DEGs. Also, in this and the previous Experiment 1, the ratio of identically recognized significant DEGs stayed between 38-61%. This creates an opportunity for selecting those overlapping genes and using them for further analysis as a robustly selected set, identified by multiple different designs at the same time.

Including more variables at the same time in the design showed to produce very diverse results. We explored three designs incorporating combinations of two continuous fraction variables. The number of significant DEGs varied greatly, and there was no overlap in the lists of the top 100 most significantly DE genes for these designs.

Including full information from CIBERSORT LM6 has shown unexpected problems with the design matrices, specifically the problem of linear dependence between variables, which has to be controlled for, especially when including continuous variables resembling percentages. This also shows that there has to be some mechanism for controlling for linear dependence of the incorporated variables, e.g., in the case of implementing an algorithm for automatic incorporation of the deconvolution results to the DGEA.

We also evaluated the stability of the DESeq2's results with different designs with respect to the GSEA of GO terms, indicating that the results are very robust, showing minimal differences in the lists of most significant GO terms. This would support the hypothesis that although the different designs produce highly varying results on the gene level, in the "bigger" picture of GO terms, the results remain quite similar. However, there is a substantial increase in the *p*-values and adjusted *p*-values, causing the identified enriched GO terms to be insignificant.

In conclusion, it was not easy to compare and evaluate the actual benefit of added deconvolution variables without ground truth. We cannot assess the actual added benefit of added deconvolution, but the reported differences and results create an opportunity for future research focused on evaluating the nature of the differences with respect to the known underlying truth.



(a) : The number of significant DEGs (with adjusted p-value <0.1) for three different designs incorporating combinations of two fraction variables in continuous form, and the baseline design with no additional variables.

(b) : Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by four different designs, three of them with combinations of shown cell type fraction variables.

(c) : Spearman's rank correlation of gene ordering for four different DESeq2 designs, three of them designs incorporating combinations of two fraction variables in continuous form, and one serving as a baseline, with no additional variables.

Figure 5.6: Comparison of four different DESeq2 designs, three of them incorporating combinations (of size 2) of variables with fractions of CD4+ T cells, CD8+ T cells and NK cells coming from **CIBERSORT LM6**. We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.

Chapter 6
Conclusion

In this master thesis, we presented an introduction to the RNA sequencing, firstly by introducing the necessary biological background, followed by a high-level description of the sequencing technology. After describing the RNA-seq pipeline, we explored in detail the statistical properties and handling of the RNA-seq count data, specifically the normalization methods, with the description of their recommended usage.

We further described a problem of deconvolution of gene expression profiles, described several types of tasks, which are commonly understood under this term, and showed a proper mathematical formalization of the problem. In the context of this formalization, we performed comprehensive literature research, describing both commonly used and state-of-the-art methods. These methods were described in detail, put in the framework of previously introduced formalization, and evaluated based on proposed metrics. These metrics were carefully selected based on the needs and specifics of our data and detailed research in the literature.

Out of the described methods, eighteen variants and setups coming from ten of these deconvolution methods were selected to be applied to the data provided by the thesis supervisor. The provided data consisted of RNA-seq data, coming from eighty whole blood samples. Although the so-called count matrix resulting from RNA-seq reads was provided too, it was not suitable for proper deconvolution and differential gene expression analysis, mainly because of missing information from the alignment to genome/transcriptome. Therefore, the complete pipeline of RNA-seq FASTAQ read files alignment to the transcriptome and feature counting had to be performed. This was done using several tools, in order to deal with the uncertainty coming from a low mapping rate, which was further explored with RNA-seq quality control tools and plausibly explained. During the data preprocessing, we also briefly examined the differences in performing the differential gene expression analysis on gene and transcript levels, showing significant differences.

The results of deconvolution methods applied to the data were examined, compared on the basis of Pearson and Spearman correlation coefficients, and visualized within a common framework, where possible. This revealed clusters of methods, reaching similar results, even though employing different approaches to deconvolution. We also showed a method for identifying unknown cell types detected by complete deconvolution method, and successfully used it for mapping of unknown gene expression profiles to known cell types, defined in other signature matrices.

Finally, we applied the results obtained by deconvolution to DGE analysis (in order to detect biomarkers) and explored the DGE results. We proposed several ways of including the deconvolution in the design of DESeq2 and compared the results based on the number of significantly DE genes, the number of uniquely and identically found genes, distribution of p-values and overall ranking of the found genes. We also compared the results using the gene set enrichment analysis of GO terms.

We showed that those results differ, sometimes even greatly, and no consistent effect of incorporating the additional information was observed. Results obtained by GSEA were found to be robust in terms of the order of identified GO terms but generally losing significance with the incorporation of deconvolution information. Although the actual evaluation of the added benefit is not easily possible (as no ground truth of the actual biomarkers present is available), the mere presence of such differences gives an opportunity for additional research, which would focus in detail on the nature of these differences and their relationship to the underlying truth.

Future work should be aimed at establishing a pipeline for performing deconvolution, well tested and validated both on simulated and real datasets, with known ground truth. The experiments on the identification of biomarkers would also greatly benefit from a carefully prepared dataset, consisting of both bulk RNA-seq and scRNA-seq datasets, coming from the same tissue of interest, ideally with measured proportions of present cell types.

Bibliography

[1] Alexander R. Abbas et al. “Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus”. In: *PloS One* 4.7 (July 1, 2009), e6098. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0006098.

[2] Alexander R. Abbas et al. “Immune response in silico (IRIS): immune-specific genes identified from a compendium of microarray expression data”. In: *Genes and Immunity* 6.4 (June 2005), pp. 319–331. ISSN: 1466-4879. DOI: 10.1038/sj.gene.6364173.

[3] Bruce Alberts. *Molecular Biology of the Cell*. Google-Books-ID: 2xIwDwAAQBAJ. Garland Science, Aug. 7, 2017. 3413 pp. ISBN: 978-1-317-56374-7.

[4] Simon Anders and Wolfgang Huber. “Differential expression analysis for sequence count data”. In: *Genome Biology* 11.10 (Oct. 27, 2010), R106. ISSN: 1474-760X. DOI: 10.1186/gb-2010-11-10-r106. URL: https://doi.org/10.1186/gb-2010-11-10-r106.

[5] Simon Anders, Alejandro Reyes, and Wolfgang Huber. “Detecting differential usage of exons from RNA-seq data”. In: *Genome Research* 22.10 (Oct. 2012), pp. 2008–2017. ISSN: 1549-5469. DOI: 10.1101/gr.133744.111.

[6] Dvir Aran. *xCell*. original-date: 2017-03-28T16:03:47Z. May 8, 2020. URL: https://github.com/dviraran/xCell (visited on 05/13/2020).

[7] Dvir Aran, Zicheng Hu, and Atul J. Butte. “xCell: digitally portraying the tissue cellular heterogeneity landscape”. In: *Genome Biology* 18 (Nov. 15, 2017). ISSN: 1474-7596. DOI: 10.1186/s13059-017-1349-1. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5688663/.

[8] Michael Ashburner et al. “Gene Ontology: tool for the unification of biology”. In: *Nature genetics* 25.1 (May 2000), pp. 25–29. ISSN: 1061-4036. DOI: 10.1038/75556. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3037419/.

[9] Arthur Atkinson et al. “Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework”. In: *Clinical Pharmacology & Therapeutics* 69.3 (2001), pp. 89–95. ISSN: 1532-6535. DOI: 10.1067/mcp.2001.113989. URL: https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1067/mcp.2001.113989.

[10] Francisco Avila Cobos et al. “Computational deconvolution of transcriptomics data from mixed cell populations”. In: *Bioinformatics* 34.11 (June 1, 2018). Publisher: Oxford Academic, pp. 1969–1979. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty019. URL: https://academic.oup.com/bioinformatics/article/34/11/1969/4813737.

[11] Maayan Baron et al. “A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure”. In: *Cell Systems* 3.4 (2016), 346–360.e4. ISSN: 2405-4712. DOI: 10.1016/j.cels.2016.08.011.

[12] Etienne Becht et al. “Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression”. In: *Genome Biology* 17.1 (Oct. 20, 2016), p. 218. ISSN: 1474-760X. DOI: 10.1186/s13059-016-1070-5. URL: https://doi.org/10.1186/s13059-016-1070-5.

[13] Yoav Benjamini and Yosef Hochberg. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995). Publisher: [Royal Statistical Society, Wiley], pp. 289–300. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2346101.

[14] Arnold Berk et al. *Molecular Cell Biology*. Macmillan Learning, 2016. ISBN: 978-1-4641-8339-3. URL: https://books.google.cz/books?id=HjEVswEACAAJ.

[15] David M. Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[16] RNA-Seq Blog. *RPKM, FPKM and TPM, clearly explained | RNA-Seq Blog*. Library Catalog: www.rna-seqblog.com. URL: https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/ (visited on 04/19/2020).

[17] Ben M. Bolstad et al. “A comparison of normalization methods for high density oligonucleotide array data based on variance and bias”. In: *Bioinformatics (Oxford, England)* 19.2 (Jan. 22, 2003), pp. 185–193. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/19.2.185.

[18] Jürgen Borlak. *Handbook of Toxicogenomics: A Strategic View of Current Research and Applications*. Google-Books-ID: z1fbymw_NvUC. John Wiley & Sons, Mar. 6, 2006. 708 pp. ISBN: 978-3-527-60451-7.

[19] Nicolas L. Bray et al. "Near-optimal probabilistic RNA-seq quantification". In: *Nature Biotechnology* 34.5 (May 2016). Number: 5 Publisher: Nature Publishing Group, pp. 525–527. ISSN: 1546-1696. DOI: 10.1038/nbt.3519. URL: https://www.nature.com/articles/nbt.3519.

[20] *bseqsc*. original-date: 2016-06-28T10:13:54Z. Apr. 21, 2020. URL: https://github.com/shenorrLab/bseqsc (visited on 05/08/2020).

[21] *Bulk RNA-seq deconvolution by scRNA-seq with multi-reference datasets*. Library Catalog: meichendong.github.io. URL: https://meichendong.github.io/SCDC/index.html (visited on 05/08/2020).

[22] Deepak Chandrasekharan, Fadi Issa, and Kathryn J. Wood. "Achieving operational tolerance in transplantation: how can lessons from the clinic inform research directions?" In: *Transplant International* 26.6 (2013). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tri.12081, pp. 576–589. ISSN: 1432-2277. DOI: 10.1111/tri.12081. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/tri.12081.

[23] Binbin Chen et al. "Profiling tumor infiltrating immune cells with CIBERSORT". In: *Methods in molecular biology (Clifton, N.J.)* 1711 (2018), pp. 243–259. ISSN: 1064-3745. DOI: 10.1007/978-1-4939-7493-1_12. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5895181/.

[24] Shifu Chen et al. "AfterQC: automatic filtering, trimming, error removing and quality control for fastq data". In: *BMC Bioinformatics* 18.3 (Mar. 14, 2017), p. 80. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1469-3. URL: https://doi.org/10.1186/s12859-017-1469-3.

[25] Jared M. Churko et al. "Overview of High Throughput Sequencing Technologies to Elucidate Molecular Pathways in Cardiovascular Diseases". In: *Circulation research* 112.12 (June 7, 2013). ISSN: 0009-7330. DOI: 10.1161/CIRCRESAHA.113.300939. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3831009/.

[26] *CIBERSORT*. URL: https://cibersort.stanford.edu/ (visited on 05/04/2020).

[27] *CIBERSORTx*. URL: https://cibersortx.stanford.edu/ (visited on 05/04/2020).

[28] Peter J. A. Cock et al. "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". In: *Nucleic Acids Research* 38.6 (Apr. 2010), pp. 1767–1771. ISSN: 0305-1048. DOI: 10.1093/nar/gkp1137. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2847217/.

[29] Geoffrey M. Cooper and Robert E. Hausman. *The Cell: A Molecular Approach*. Google-Books-ID: brOXjgEACAAJ. Sinauer, Nov. 19, 2015. 500 pp. ISBN: 978-1-60535-563-4.

[30] *Deconvolution of ABsolute Immune Signal*. URL: https://giannimonaco.shinyapps.io/ABIS/ (visited on 05/07/2020).

[31] *DeMixT*. original-date: 2018-08-24T19:21:55Z. Feb. 1, 2020. URL: https://github.com/wwylab/DeMixTallmaterials (visited on 05/09/2020).

[32] *Designing Next-Generation Sequencing Runs*. URL: https://genohub.com/next-generation-sequencing-guide/ (visited on 05/02/2020).

[33] Konstantina Dimitrakopoulou et al. "Deblender: a semi-/unsupervised multi-operational computational method for complete deconvolution of expression data from heterogeneous samples". In: *BMC Bioinformatics* 19 (Nov. 7, 2018). ISSN: 1471-2105. DOI: 10.1186/s12859-018-2442-5. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6223087/.

[34] *DNA sekvenování & Real-Time PCR*. URL: https://www.seqme.eu/cs/ (visited on 05/10/2020).

[35] Meichen Dong et al. "SCDC: bulk gene expression deconvolution by multiple single-cell RNA sequencing references". In: *Briefings in Bioinformatics* (). DOI: 10.1093/bib/bbz166. URL: https://academic.oup.com/bib/advance-article/doi/10.1093/bib/bbz166/5699815.

[36] *Download stats for Bioconductor software packages*. URL: https://bioconductor.org/packages/stats/ (visited on 05/17/2020).

[37] *DWLS*. URL: https://bitbucket.org/yuanlab/dwls/src/default/ (visited on 05/08/2020).

[38] Ciaran Evans, Johanna Hardin, and Daniel M Stoebel. "Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions". In: *Briefings in Bioinformatics* 19.5 (Feb. 27, 2017), pp. 776–792. ISSN: 1467-5463. DOI: 10.1093/bib/bbx008. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6171491/.

[39] Oscar Franzén, Li-Ming Gan, and Johan L. M. Björkegren. "PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data". In: *Database: The Journal of Biological Databases and Curation* 2019 (2019). ISSN: 1758-0463. DOI: 10.1093/database/baz046.

[40] Fernando García-Alcalde et al. "Qualimap: evaluating next-generation sequencing alignment data". In: *Bioinformatics (Oxford, England)* 28.20 (Oct. 15, 2012), pp. 2678–2679. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bts503.

[41] Renaud Gaujoux and Cathal Seoighe. "CellMix: a comprehensive toolbox for gene expression deconvolution". In: *Bioinformatics (Oxford, England)* 29.17 (Sept. 1, 2013), pp. 2211–2212. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btt351.

[42] Ting Gong and Joseph D. Szustakowski. "DeconRNASeq: a statistical framework for deconvolution of heterogeneous tissue samples based on mRNA-Seq data". In: *Bioinformatics* 29.8 (Apr. 15, 2013), pp. 1083–1085. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt090. URL: https://academic.oup.com/bioinformatics/article/29/8/1083/229442.

[43] Ting Gong et al. "Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples". In: *PloS One* 6.11 (2011), e27156. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0027156.

[44] Huili Guo et al. "Mammalian microRNAs predominantly act to decrease target mRNA levels". In: *Nature* 466.7308 (Aug. 12, 2010), pp. 835–840. ISSN: 1476-4687. DOI: 10.1038/nature09267.

[45] Yixing Han et al. "Advanced Applications of RNA Sequencing and Challenges". In: *Bioinformatics and Biology Insights* 9 (Suppl 1 Nov. 15, 2015), pp. 29–46. ISSN: 1177-9322. DOI: 10.4137/BBI.S28991. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4648566/.

[46] Tallulah Andrews Hemberg et al. *2 Introduction to single-cell RNA-seq | Analysis of single cell RNA-seq data*. URL: http://hemberg-lab.github.io/scRNA.seq.course/.

[47] *How "Pseudoalignments" Work in kallisto*. URL: https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html (visited on 05/18/2020).

[48] Radmila Hrdlickova, Masoud Toloue, and Bin Tian. "RNA-Seq methods for transcriptome analysis". In: *Wiley interdisciplinary reviews. RNA* 8.1 (Jan. 2017). ISSN: 1757-7004. DOI: 10.1002/wrna.1364. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5717752/.

[49] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-cell RNA sequencing technologies and bioinformatics pipelines". In: *Experimental & Molecular Medicine* 50.8 (Aug. 7, 2018). Number: 8 Publisher: Nature Publishing Group, pp. 1–14. ISSN: 2092-6413. DOI: 10.1038/s12276-018-0071-8. URL: https://www.nature.com/articles/s12276-018-0071-8.

[50] *immunedeconv*. May 12, 2020. URL: https://github.com/icbi-lab/immunedeconv (visited on 05/13/2020).

[51] *In RNA-Seq, 2 != 2: Between-sample normalization*. The farrago. Library Catalog: haroldpimentel.wordpress.com. Dec. 8, 2014. URL: https://haroldpimentel.wordpress.com/2014/12/08/in-rna-seq-2-2-between-sample-normalization/ (visited on 04/25/2020).

[52] Nicholas T. Ingolia. "Genome-wide translational profiling by ribosome footprinting". In: *Methods in Enzymology* 470 (2010), pp. 119–142. ISSN: 1557-7988. DOI: 10.1016/S0076-6879(10)70006-9.

[53] Brandon Jew et al. "Accurate estimation of cell composition in bulk expression through robust integration of single-cell information". In: *bioRxiv* (June 15, 2019). Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 669911. DOI: 10.1101/669911. URL: https://www.biorxiv.org/content/10.1101/669911v1.

[54] Kai. *CDSeq*. original-date: 2018-11-29T15:43:02Z. Dec. 9, 2019. URL: https://github.com/kkang7/CDSeq (visited on 05/07/2020).

[55] Kai Kang et al. "CDSeq: A novel complete deconvolution method for dissecting heterogeneous samples using gene expression data". In: *PLoS computational biology* 15.12 (2019), e1007510. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1007510.

[56] Gonca E. Karahan, Frans H. J. Claas, and Sebastiaan Heidt. "B Cell Immunity in Solid Organ Transplantation". In: *Frontiers in Immunology* 7 (Jan. 10, 2017). ISSN: 1664-3224. DOI: 10.3389/fimmu.2016.00686. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5222792/.

[57] Daehwan Kim et al. "Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype". In: *Nature Biotechnology* 37.8 (Aug. 2019). Number: 8 Publisher: Nature Publishing Group, pp. 907–915. ISSN: 1546-1696. DOI: 10.1038/s41587-019-0201-4. URL: https://www.nature.com/articles/s41587-019-0201-4.

[58] kondim1983. *Deblender*. Oct. 6, 2018. URL: https://github.com/kondim1983/Deblender (visited on 05/09/2020).

[59] Kaarel Krjutškov et al. "Globin mRNA reduction for whole-blood transcriptome sequencing". In: *Scientific Reports* 6.1 (Aug. 12, 2016). Number: 1 Publisher: Nature Publishing Group, p. 31584. ISSN: 2045-2322. DOI: 10.1038/srep31584. URL: https://www.nature.com/articles/srep31584.

[60] Charity W. Law et al. "voom: precision weights unlock linear model analysis tools for RNA-seq read counts". In: *Genome Biology* 15.2 (Feb. 3, 2014), R29. ISSN: 1474-760X. DOI: 10.1186/gb-2014-15-2-r29. URL: https://doi.org/10.1186/gb-2014-15-2-r29.

[61] Bo Li and Colin N. Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome". In: *BMC Bioinformatics* 12.1 (Aug. 4, 2011), p. 323. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-323. URL: https://doi.org/10.1186/1471-2105-12-323.

[62] Bo Li et al. "RNA-Seq gene expression estimation with read mapping uncertainty". In: *Bioinformatics* 26.4 (Feb. 15, 2010). Publisher: Oxford Academic, pp. 493–500. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp692. URL: https://academic.oup.com/bioinformatics/article/26/4/493/243395.

[63] Heng Li et al. “The Sequence Alignment/Map format and SAMtools”. In: *Bioinformatics (Oxford, England)* 25.16 (Aug. 15, 2009), pp. 2078–2079. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btp352.

[64] *LinSeed*. original-date: 2018-03-13T21:35:38Z. Nov. 21, 2019. URL: https://github.com/ctlab/LinSeed (visited on 05/07/2020).

[65] Michael I. Love, Wolfgang Huber, and Simon Anders. “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”. In: *Genome Biology* 15.12 (Dec. 5, 2014), p. 550. ISSN: 1474-760X. DOI: 10.1186/s13059-014-0550-8. URL: https://doi.org/10.1186/s13059-014-0550-8.

[66] Michael I. Love et al. *DESeq2: Differential gene expression analysis based on the negative binomial distribution*. Version 1.28.1. 2020. DOI: 10.18129/B9.bioc.DESeq2. URL: https://bioconductor.org/packages/DESeq2/ (visited on 05/15/2020).

[67] Michael I. Love et al. *tximport: Import and summarize transcript-level estimates for transcript- and gene-level analysis*. Version 1.16.0. 2020. DOI: 10.18129/B9.bioc.tximport. URL: https://bioconductor.org/packages/tximport/ (visited on 05/18/2020).

[68] Peng Lu, Aleksey Nakorchevskiy, and Edward M. Marcotte. “Expression deconvolution: A reinterpretation of DNA microarray data reveals dynamic changes in cell populations”. In: *Proceedings of the National Academy of Sciences of the United States of America* 100.18 (Sept. 2, 2003), pp. 10370–10375. ISSN: 0027-8424. DOI: 10.1073/pnas.1832361100. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC193568/.

[69] Luo et al. “GAGE: generally applicable gene set enrichment for pathway analysis”. In: *BMC Bioinformatics* 10 (2009), p. 161.

[70] Donna Maglott et al. “Entrez Gene: gene-centered information at NCBI”. In: *Nucleic Acids Research* 39 (Database issue Jan. 2011), pp. D52–D57. ISSN: 0305-1048. DOI: 10.1093/nar/gkq1237. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013746/.

[71] Samuel Marguerat and Jürg Bähler. “Coordinating genome expression with cell size”. In: *Trends in Genetics* 28.11 (Nov. 1, 2012). Publisher: Elsevier, pp. 560–565. ISSN: 0168-9525. DOI: 10.1016/j.tig.2012.07.003. URL: https://www.cell.com/trends/genetics/abstract/S0168-9525(12)00100-X.

[72] Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. “Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation”. In: *Nucleic Acids Research* 40.10 (May 1, 2012). Publisher: Oxford Academic, pp. 4288–4297. ISSN: 0305-1048. DOI: 10.1093/nar/gks042. URL: https://academic.oup.com/nar/article/40/10/4288/2411520.

[73] Meeta Mistry, Radhika Khetani, Mary Piper. *Count normalization with DESeq2*. Introduction to DGE. Library Catalog: hbctraining.github.io. Apr. 26, 2017. URL: https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html (visited on 04/23/2020).

[74] Anita Mehrotra and Peter S. Heeger. "B Cells and Kidney Transplantation: Beyond Antibodies". In: *Journal of the American Society of Nephrology : JASN* 25.7 (July 2014), pp. 1373–1374. ISSN: 1046-6673. DOI: 10.1681/ASN.2014020132. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4073443/.

[75] Shahin Mohammadi et al. "A Critical Survey of Deconvolution Methods for Separating cell-types in Complex Tissues". In: *Proceedings of the IEEE* 105.2 (Feb. 2017), pp. 340–366. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2016.2607121. arXiv: 1510.04583. URL: http://arxiv.org/abs/1510.04583.

[76] Gianni Monaco. *ABIS*. original-date: 2018-08-14T12:39:48Z. Apr. 28, 2020. URL: https://github.com/giannimonaco/ABIS (visited on 05/07/2020).

[77] Gianni Monaco et al. "RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types". In: *Cell Reports* 26.6 (2019), 1627–1640.e7. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2019.01.041.

[78] Ali Mortazavi et al. "Mapping and quantifying mammalian transcriptomes by RNA-Seq". In: *Nature Methods* 5.7 (July 2008). Number: 7 Publisher: Nature Publishing Group, pp. 621–628. ISSN: 1548-7105. DOI: 10.1038/nmeth.1226. URL: https://www.nature.com/articles/nmeth.1226.

[79] Aaron M. Newman et al. "Determining cell type abundance and expression from bulk tissues with digital cytometry". In: *Nature Biotechnology* 37.7 (July 2019), pp. 773–782. ISSN: 1546-1696. DOI: 10.1038/s41587-019-0114-2. URL: https://www.nature.com/articles/s41587-019-0114-2.

[80] Aaron M. Newman et al. "Robust enumeration of cell subsets from tissue expression profiles". In: *Nature Methods* 12.5 (May 2015), pp. 453–457. ISSN: 1548-7105. DOI: 10.1038/nmeth.3337.

[81] Stephen C. Van Nostrand. *epicpy*. original-date: 2018-07-03T18:46:07Z. July 23, 2018. URL: https://github.com/scvannost/epicpy (visited on 05/07/2020).

[82] Konstantin Okonechnikov, Ana Conesa, and Fernando García-Alcalde. "Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data". In: *Bioinformatics (Oxford, England)* 32.2 (Jan. 15, 2016), pp. 292–294. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btv566.

[83] Thale Kristin Olsen and Ninib Baryawno. “Introduction to Single-Cell RNA Sequencing”. In: *Current Protocols in Molecular Biology* 122.1 (2018), e57. ISSN: 1934-3647. DOI: 10.1002/cpmb.57. URL: https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/cpmb.57.

[84] Fatih Ozsolak et al. “Direct RNA sequencing”. In: *Nature* 461.7265 (Oct. 8, 2009), pp. 814–818. ISSN: 1476-4687. DOI: 10.1038/nature08390.

[85] Lior Pachter. “Models for transcript quantification from RNA-Seq”. In: *arXiv:1104.3889 [q-bio, stat]* (May 12, 2011). arXiv: 1104.3889. URL: http://arxiv.org/abs/1104.3889.

[86] *Paired-End vs. Single-Read Sequencing Technology*. Library Catalog: www.illumina.com. URL: https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/paired-end-vs-single-read.html (visited on 05/02/2020).

[87] Rob Patro et al. “Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference”. In: *Nature methods* 14.4 (Apr. 2017), pp. 417–419. ISSN: 1548-7091. DOI: 10.1038/nmeth.4197. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5600148/.

[88] Stano Pekar and Marek Brabec. *Modern Analysis of Biological Data. Generalized Linear Models in R*. Jan. 2016. ISBN: 9788021080195.

[89] Wenlian Qiao et al. “PERT: a method for expression deconvolution of human blood samples from varied microenvironmental and developmental conditions”. In: *PLoS computational biology* 8.12 (2012), e1002838. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002838.

[90] Julien Racle and David Gfeller. *EPIC*. Apr. 16, 2020. URL: https://github.com/GfellerLab/EPIC (visited on 05/07/2020).

[91] Julien Racle and David Gfeller. *EPIC*. URL: https://gfellerlab.shinyapps.io/EPIC_1-1/ (visited on 05/07/2020).

[92] Julien Racle et al. “Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data”. In: *eLife* 6 (Nov. 13, 2017). Ed. by Alfonso Valencia. Publisher: eLife Sciences Publications, Ltd, e26476. ISSN: 2050-084X. DOI: 10.7554/eLife.26476. URL: https://doi.org/10.7554/eLife.26476.

[93] Aviv Regev et al. “The Human Cell Atlas”. In: *eLife* 6 (2017). ISSN: 2050-084X. DOI: 10.7554/eLife.27041.

[94] Matthew E. Ritchie et al. “limma powers differential expression analyses for RNA-sequencing and microarray studies”. In: *Nucleic Acids Research* 43.7 (Apr. 20, 2015), e47. ISSN: 0305-1048. DOI: 10.1093/nar/gkv007. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4402510/.

[95] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”. In: *Bioinformatics* 26.1 (Jan. 1, 2010). Publisher: Oxford Academic, pp. 139–140. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp616. URL: https://academic.oup.com/bioinformatics/article/26/1/139/182458.

[96] Mark D. Robinson and Alicia Oshlack. “A scaling normalization method for differential expression analysis of RNA-seq data”. In: *Genome Biology* 11.3 (2010), R25. ISSN: 1474-760X. DOI: 10.1186/gb-2010-11-3-r25.

[97] Magali Ruffier et al. “Ensembl core software resources: storage and programmatic access for DNA sequence and genome annotation”. In: *Database: The Journal of Biological Databases and Curation* 2017.1 (2017). ISSN: 1758-0463. DOI: 10.1093/database/bax020.

[98] Bernhard Scholkopf et al. “New support vector algorithms”. In: *Neural Computation* 12.5 (May 2000), pp. 1207–1245. ISSN: 1530-888X. DOI: 10.1162/089976600300015565.

[99] Shai S. Shen-Orr and Renaud Gaujoux. “Computational deconvolution: extracting cell type-specific information from heterogeneous samples”. In: *Current Opinion in Immunology* 25.5 (Oct. 2013), pp. 571–578. ISSN: 1879-0372. DOI: 10.1016/j.coi.2013.09.015.

[100] Shai S. Shen-Orr et al. “cell type–specific gene expression differences in complex tissues”. In: *Nature methods* 7.4 (Apr. 2010), pp. 287–289. ISSN: 1548-7091. DOI: 10.1038/nmeth.1439. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3699332/.

[101] Heesun Shin et al. “Variation in RNA-Seq Transcriptome Profiles of Peripheral Whole Blood from Healthy Individuals with and without Globin Depletion”. In: *PLOS ONE* 9.3 (July 3, 2014). Publisher: Public Library of Science, e91041. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0091041. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0091041.

[102] Andrew Siderowf et al. “Biomarkers for cognitive impairment in Lewy body disorders: Status and relevance for clinical trials”. In: *Movement Disorders* 33.4 (2018), pp. 528–536. ISSN: 1531-8257. DOI: 10.1002/mds.27355. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mds.27355.

[103] *Single cell RNA sequencing*. NGS Analysis. Jan. 11, 2018. URL: https://learn.gencore.bio.nyu.edu/single-cell-rnaseq/ (visited on 04/26/2020).

[104] Charlotte Soneson, Michael I. Love, and Mark D. Robinson. “Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences”. In: *F1000Research* 4 (2015), p. 1521. ISSN: 2046-1402. DOI: 10.12688/f1000research.7563.2.

[105] Stephanie Hicks. *Welcome to the World of Single-Cell RNA-Sequencing.* Speaker Deck. Library Catalog: speakerdeck.com. URL: `https://speakerdeck.com/stephaniehicks/welcome-to-the-world-of-single-cell-rna-sequencing?slide=3` (visited on 04/26/2020).

[106] Gregor Sturm et al. "Comprehensive evaluation of transcriptome-based cell-type quantification methods for immuno-oncology". In: *Bioinformatics* 35.14 (July 2019), pp. i436–i445. ISSN: 1367-4803. DOI: `10.1093/bioinformatics/btz363`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6612828/`.

[107] Aravind Subramanian et al. "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences of the United States of America* 102.43 (Oct. 25, 2005), pp. 15545–15550. ISSN: 0027-8424. DOI: `10.1073/pnas.0506580102`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1239896/`.

[108] Fuchou Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature Methods* 6.5 (May 2009). Number: 5 Publisher: Nature Publishing Group, pp. 377–382. ISSN: 1548-7105. DOI: `10.1038/nmeth.1315`. URL: `https://www.nature.com/articles/nmeth.1315`.

[109] "The Gene Ontology Resource: 20 years and still GOing strong". In: *Nucleic Acids Research* 47 (D1 Jan. 8, 2019). Publisher: Oxford Academic, pp. D330–D338. ISSN: 0305-1048. DOI: `10.1093/nar/gky1055`. URL: `https://academic.oup.com/nar/article/47/D1/D330/5160994`.

[110] *The RNA-seq abundance zoo | RoBlog.* Library Catalog: robpatro.com. URL: `http://robpatro.com/blog/?p=235` (visited on 04/20/2020).

[111] Cole Trapnell et al. "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation". In: *Nature Biotechnology* 28.5 (May 2010). Number: 5 Publisher: Nature Publishing Group, pp. 511–515. ISSN: 1546-1696. DOI: `10.1038/nbt.1621`. URL: `https://www.nature.com/articles/nbt.1621`.

[112] Johannes Rainer Triche et al. *ensembldb: Utilities to create and use Ensembl-based annotation databases.* Version 2.12.1. 2020. DOI: `10.18129/B9.bioc.ensembldb`. URL: `https://bioconductor.org/packages/ensembldb/` (visited on 05/09/2020).

[113] Daphne Tsoucas et al. "Accurate estimation of cell-type composition from gene expression data". In: *Nature Communications* 10.1 (2019), p. 2975. ISSN: 2041-1723. DOI: `10.1038/s41467-019-10802-z`.

[114] *Tumor Immunology and Immunotherapy - Integrated Methods Part B.* Academic Press, Mar. 15, 2020. 406 pp. ISBN: 978-0-12-820668-3.

[115] David. Venet et al. “Separation of samples into their constituents using gene expression data”. In: *Bioinformatics (Oxford, England)* 17 Suppl 1 (2001), S279–287. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/17.suppl_1.s279.

[116] Günter P. Wagner, Koryu Kin, and Vincent J. Lynch. “Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples”. In: *Theory in Biosciences* 131.4 (Dec. 1, 2012), pp. 281–285. ISSN: 1611-7530. DOI: 10.1007/s12064-012-0162-3. URL: https://doi.org/10.1007/s12064-012-0162-3.

[117] Xuran Wang et al. “Bulk tissue cell type deconvolution with multi-subject single-cell expression reference”. In: *Nature Communications* 10.1 (2019), p. 380. ISSN: 2041-1723. DOI: 10.1038/s41467-018-08023-x.

[118] Zeya Wang et al. “Transcriptome Deconvolution of Heterogeneous Tumor Samples with Immune Infiltration”. In: *iScience* 9 (Nov. 2, 2018), pp. 451–460. ISSN: 2589-0042. DOI: 10.1016/j.isci.2018.10.028. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6249353/.

[119] Zhong Wang, Mark Gerstein, and Michael Snyder. “RNA-Seq: a revolutionary tool for transcriptomics”. In: *Nature reviews. Genetics* 10.1 (Jan. 2009), pp. 57–63. ISSN: 1471-0056. DOI: 10.1038/nrg2484. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2949280/.

[120] *What is the Condition Number of a Matrix? » Cleve’s Corner: Cleve Moler on Mathematics and Computing - MATLAB & Simulink*. URL: https://blogs.mathworks.com/cleve/2017/07/17/what-is-the-condition-number-of-a-matrix/ (visited on 05/08/2020).

[121] *What the FPKM? A review of RNA-Seq expression units*. The farrago. Library Catalog: haroldpimentel.wordpress.com. May 8, 2014. URL: https://haroldpimentel.wordpress.com/2014/05/08/what-the-fpkm-a-review-rna-seq-expression-units/ (visited on 04/19/2020).

[122] Vinod Kumar Yadav and Subhajyoti De. “An assessment of computational methods for estimating purity and clonality using genomic data derived from heterogeneous tumor tissue samples”. In: *Briefings in Bioinformatics* 16.2 (Mar. 1, 2015). Publisher: Oxford Academic, pp. 232–241. ISSN: 1467-5463. DOI: 10.1093/bib/bbu002. URL: https://academic.oup.com/bib/article/16/2/232/246006.

[123] Andrew D. Yates et al. “Ensembl 2020”. In: *Nucleic Acids Research* 48 (D1 Jan. 8, 2020). Publisher: Oxford Academic, pp. D682–D688. ISSN: 0305-1048. DOI: 10.1093/nar/gkz966. URL: https://academic.oup.com/nar/article/48/D1/D682/5613682.

[124] Bethan Yates et al. “Genenames.org: the HGNC and VGNC resources in 2017”. In: *Nucleic Acids Research* 45 (D1 2017), pp. D619–D625. ISSN: 1362-4962. DOI: 10.1093/nar/gkw1033.

[125] Konstantin Zaitsev et al. “Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures”. In: *Nature Communications* 10.1 (May 17, 2019), pp. 1–16. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09990-5. URL: https://www.nature.com/articles/s41467-019-09990-5.

[126] Konstantin Zaitsev et al. “Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures”. In: *Nature Communications* 10.1 (May 17, 2019), pp. 1–16. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09990-5. URL: https://www.nature.com/articles/s41467-019-09990-5.

[127] Xinxin Zhang et al. “CellMarker: a manually curated resource of cell markers in human and mouse”. In: *Nucleic Acids Research* 47 (D1 2019), pp. D721–D728. ISSN: 1362-4962. DOI: 10.1093/nar/gky900.

[128] Shanrong Zhao, Zhan Ye, and Robert Stanton. “Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols”. In: *RNA* (Apr. 13, 2020). Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, rna.074922.120. ISSN: 1355-8382, 1469-9001. DOI: 10.1261/rna.074922.120. URL: http://rnajournal.cshlp.org/content/early/2020/04/13/rna.074922.120.

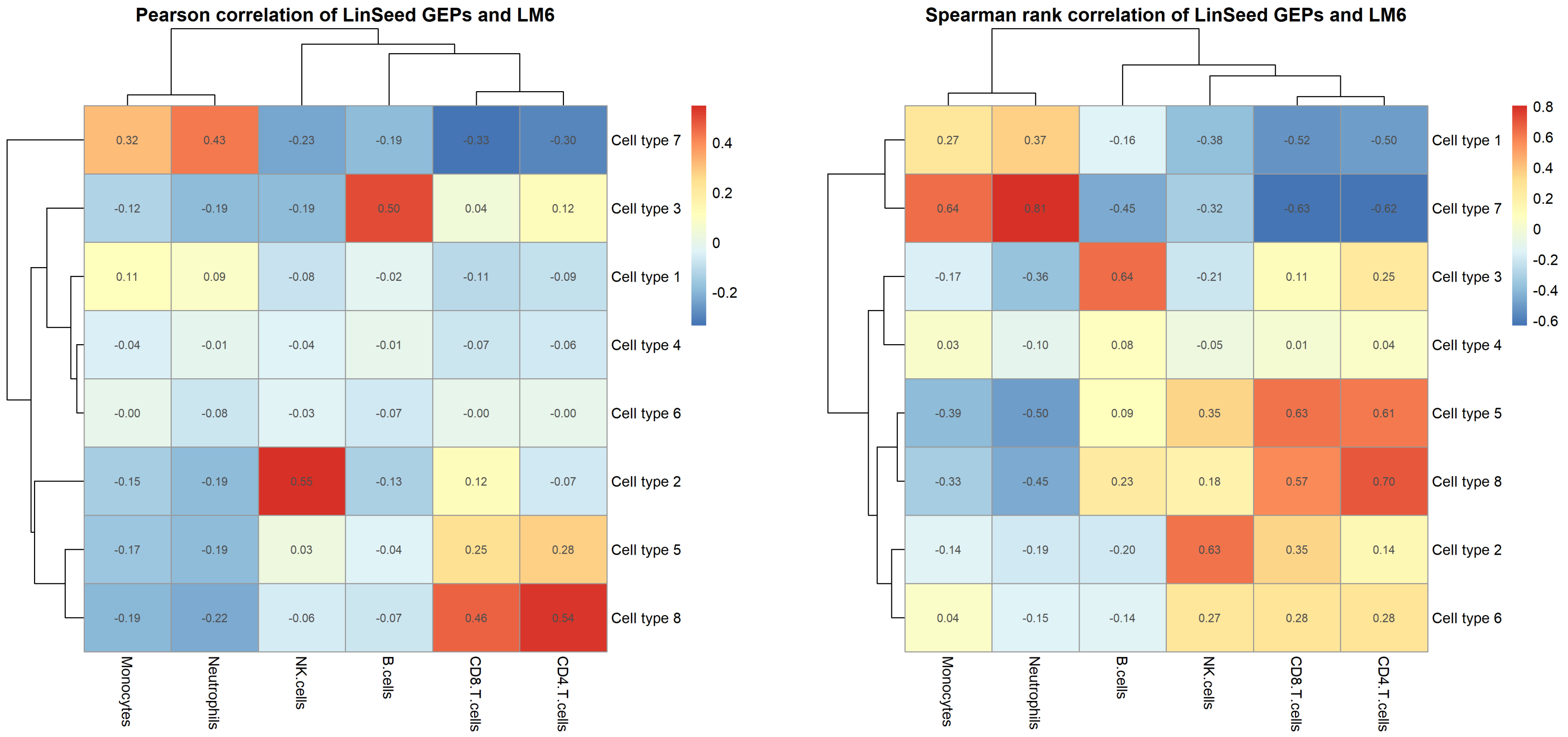
[129] Yi Zhong et al. “Digital sorting of complex tissues for cell type-specific gene expression profiles”. In: *BMC bioinformatics* 14 (Mar. 7, 2013), p. 89. ISSN: 1471-2105. DOI: 10.1186/1471-2105-14-89.

[130] Joanna Zyla et al. “Ranking metrics in gene set enrichment analysis: do they matter?” In: *BMC Bioinformatics* 18.1 (May 12, 2017), p. 256. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1674-0. URL: https://doi.org/10.1186/s12859-017-1674-0.

Appendix A

LinSeed deconvolution — figures

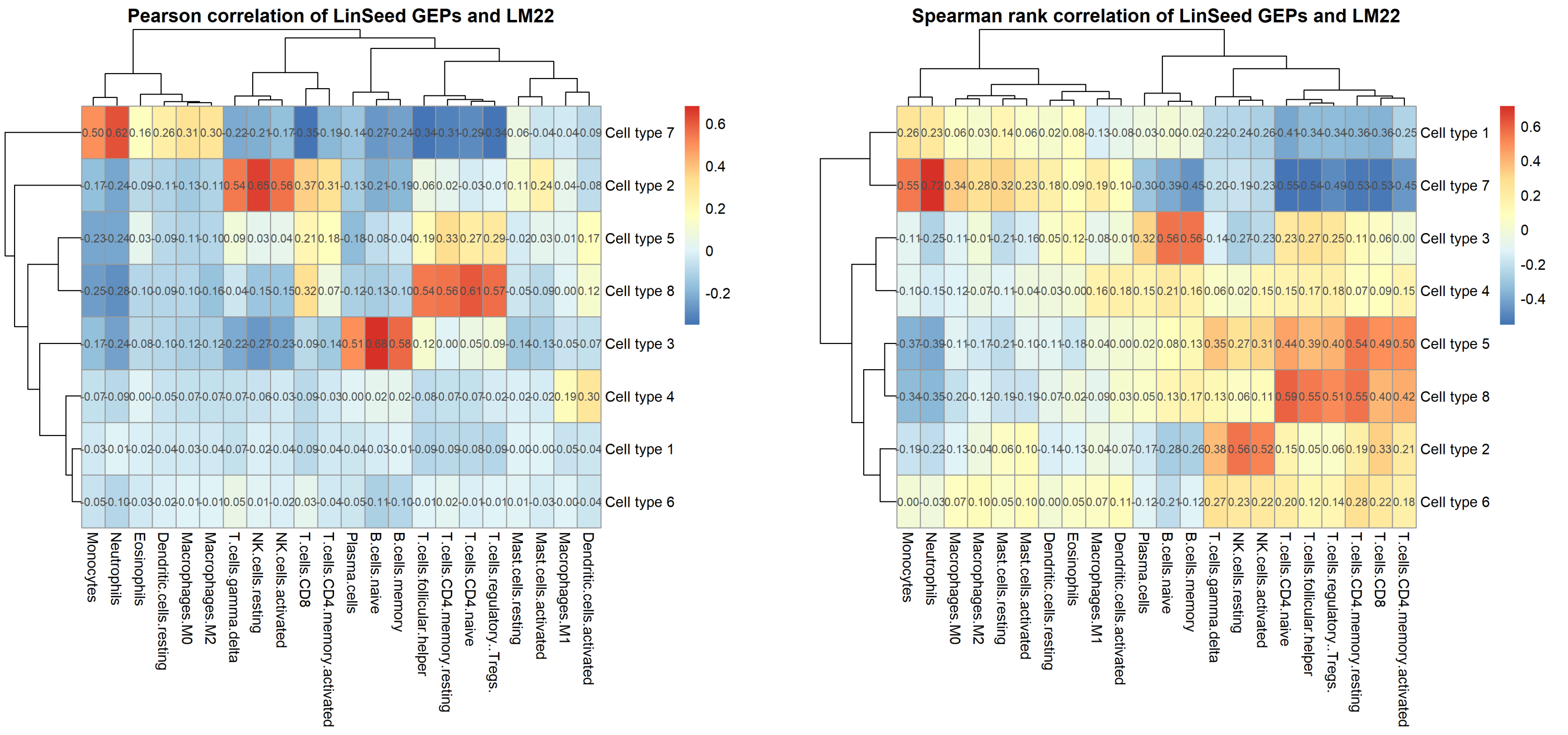
In this appendix, the accompanying figures for the Section 4.1 on deconvolution using the LinSeed are enclosed.



(a) : Pearson

(b) : Spearman

Figure A.1: Pearson's and Spearman's correlation of LM6 reference GEPs and GEPs of unknown cell types, as found by LinSeed. Note that the ordering of unknown cell types is different between these two plots — this is caused by different hierarchical clustering of unknown cell types.



(a) : Pearson correlation

(b) : Spearman correlation

Figure A.2: Pearson's and Spearman's correlation of LM22 reference GEPs and GEPs of unknown cell types, as found by LinSeed. Note that the ordering of unknown cell types is different between these two plots — this is caused by different hierarchical clustering of unknown cell types.

Appendix B

Deconvolution results — figures

In this appendix, we present figures of deconvolution results using different methods. The results are reported by cell type, **B cells, NK cells, CD4+ T cells, CD8+ T cells, monocytes and neutrophils**. For each cell type, four Figures are given. The first, showing proportions of given cell types scaled to $[0; 1]$ range in all samples, the second, showing the original proportions as reported (with methods, not producing values resembling percentages, omitted). Then, the proportions for given cell types from different methods are compared based on Pearson's and Spearman's correlation, which are reported in the third and fourth figures, respectively.

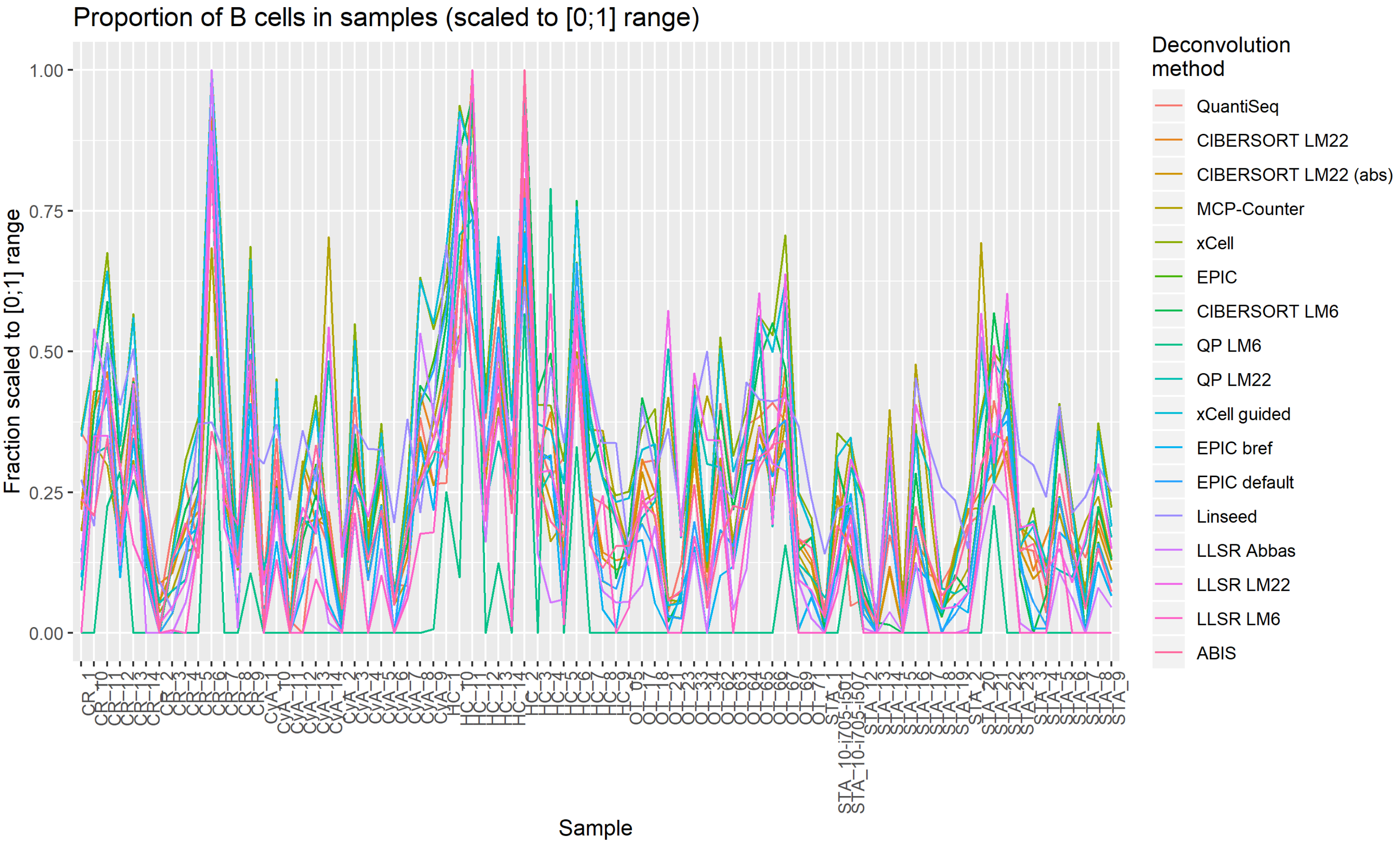


Figure B.1: Proportions or scores of B cells in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

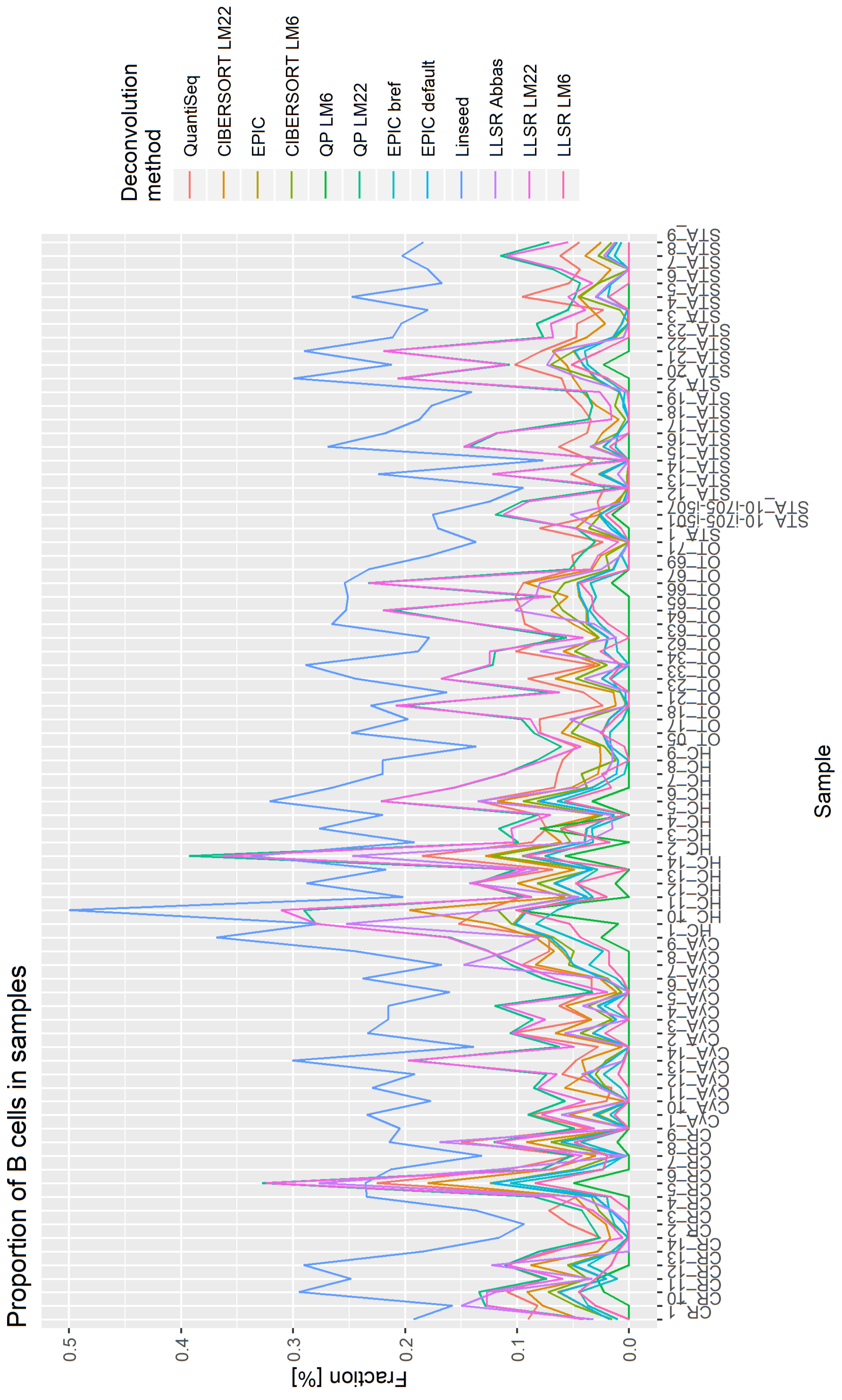
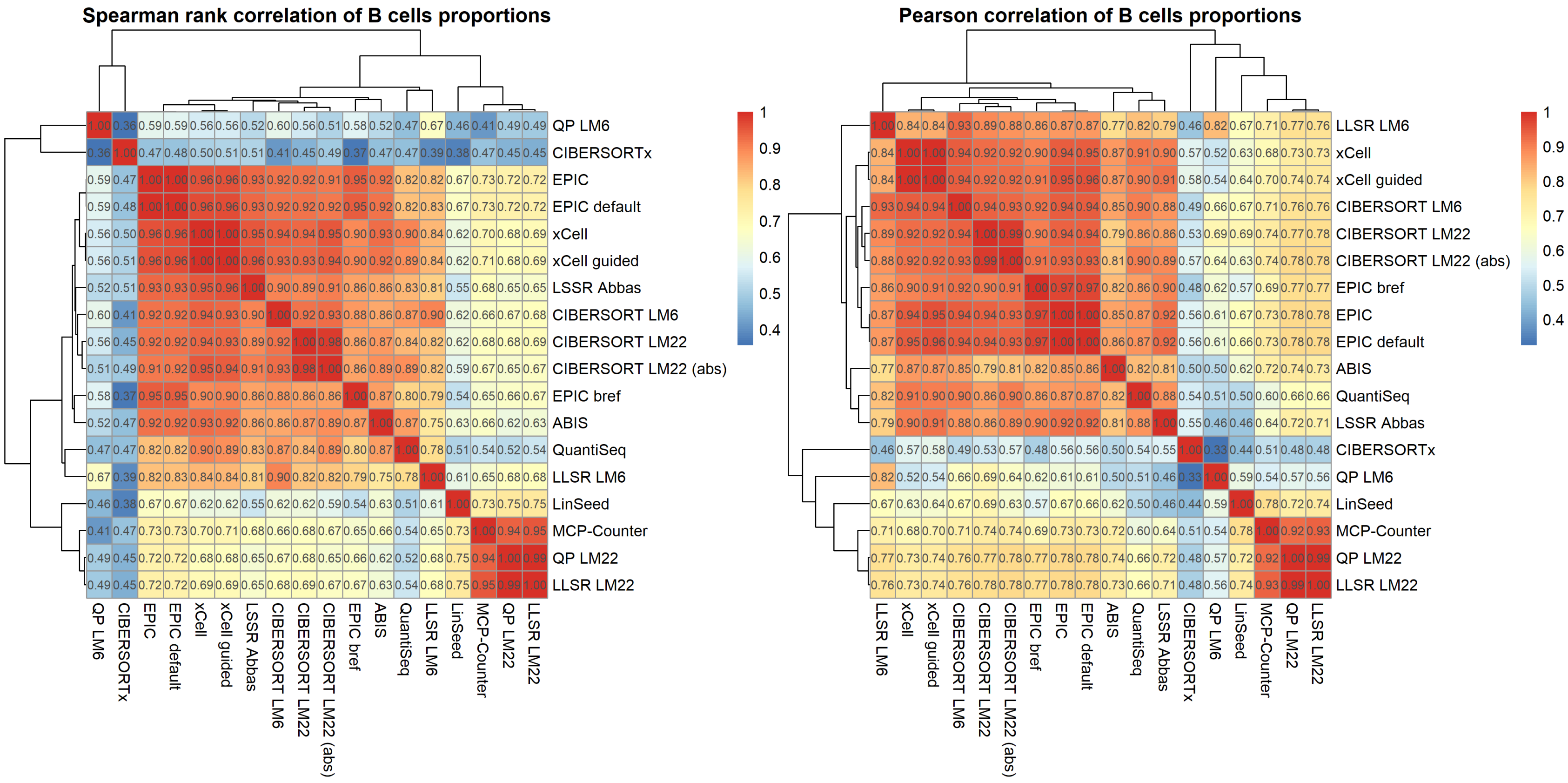


Figure B.2: Proportions of B cells in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



(a) : Spearman rank correlation of found fraction of B cells by selected deconvolution methods

(b) : Pearson correlation of found fraction of B cells by selected deconvolution methods

Figure B.3: Spearman's and Pearson's correlation coefficients for the computed fractions of B cells.

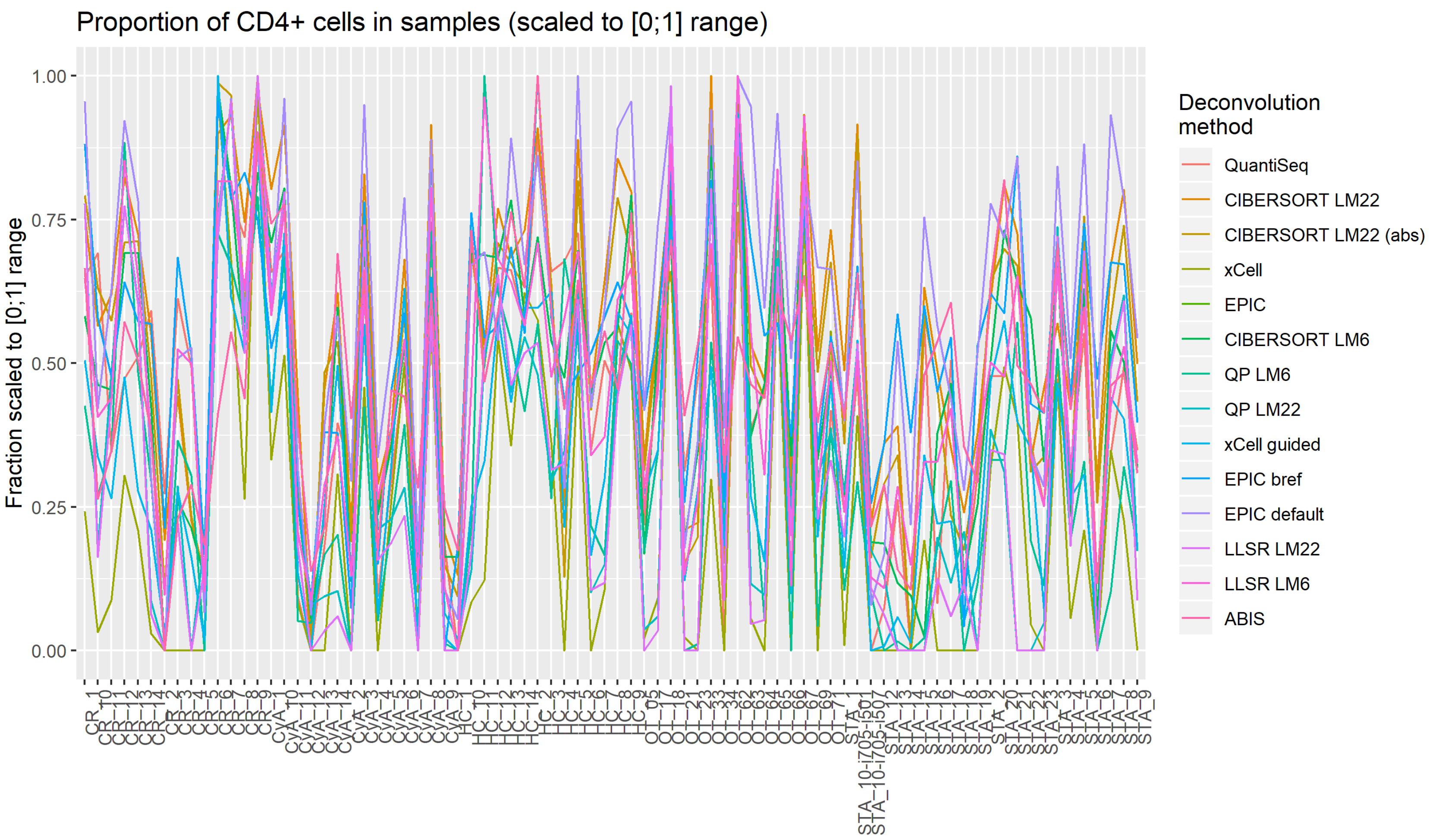


Figure B.4: Proportions or scores of CD4+ T cells in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

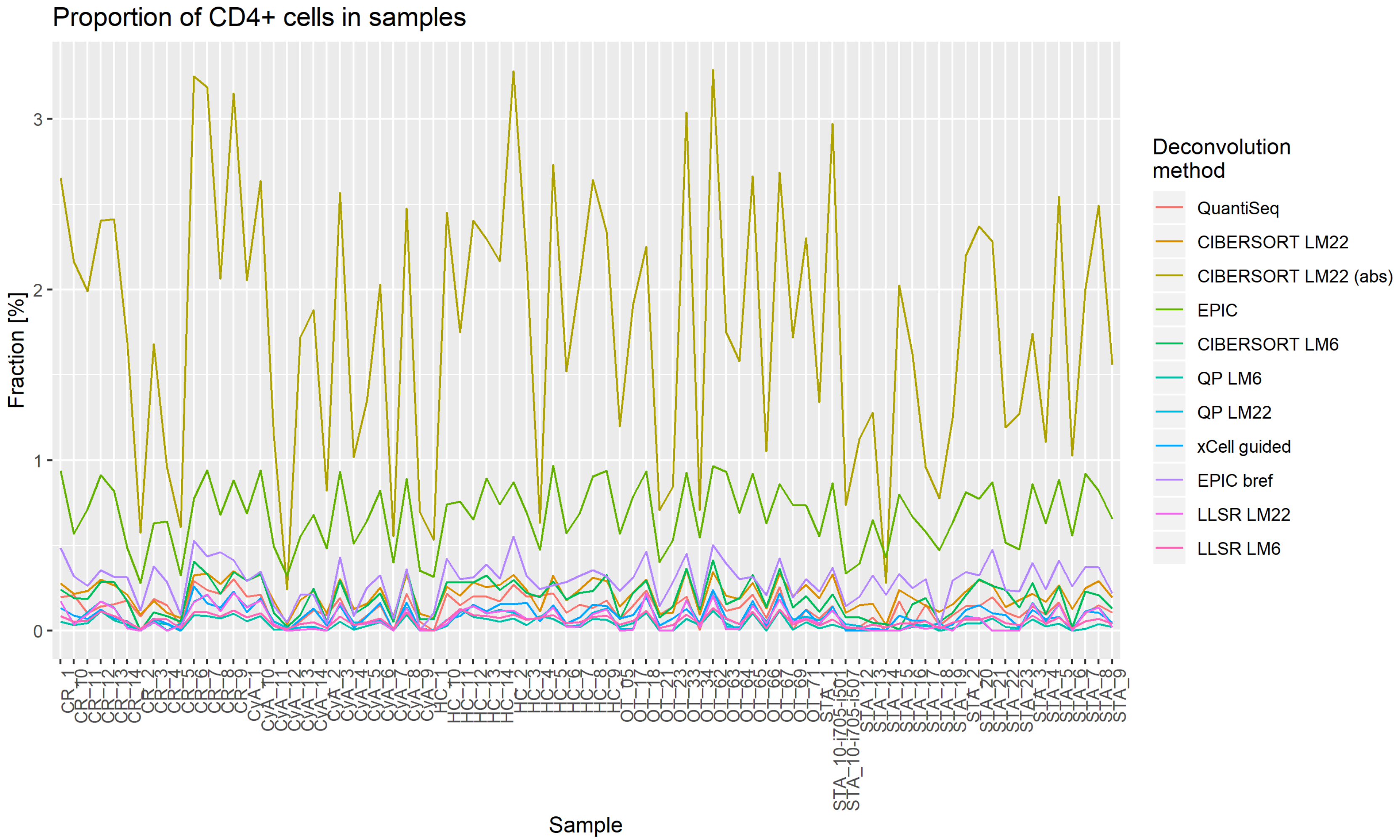
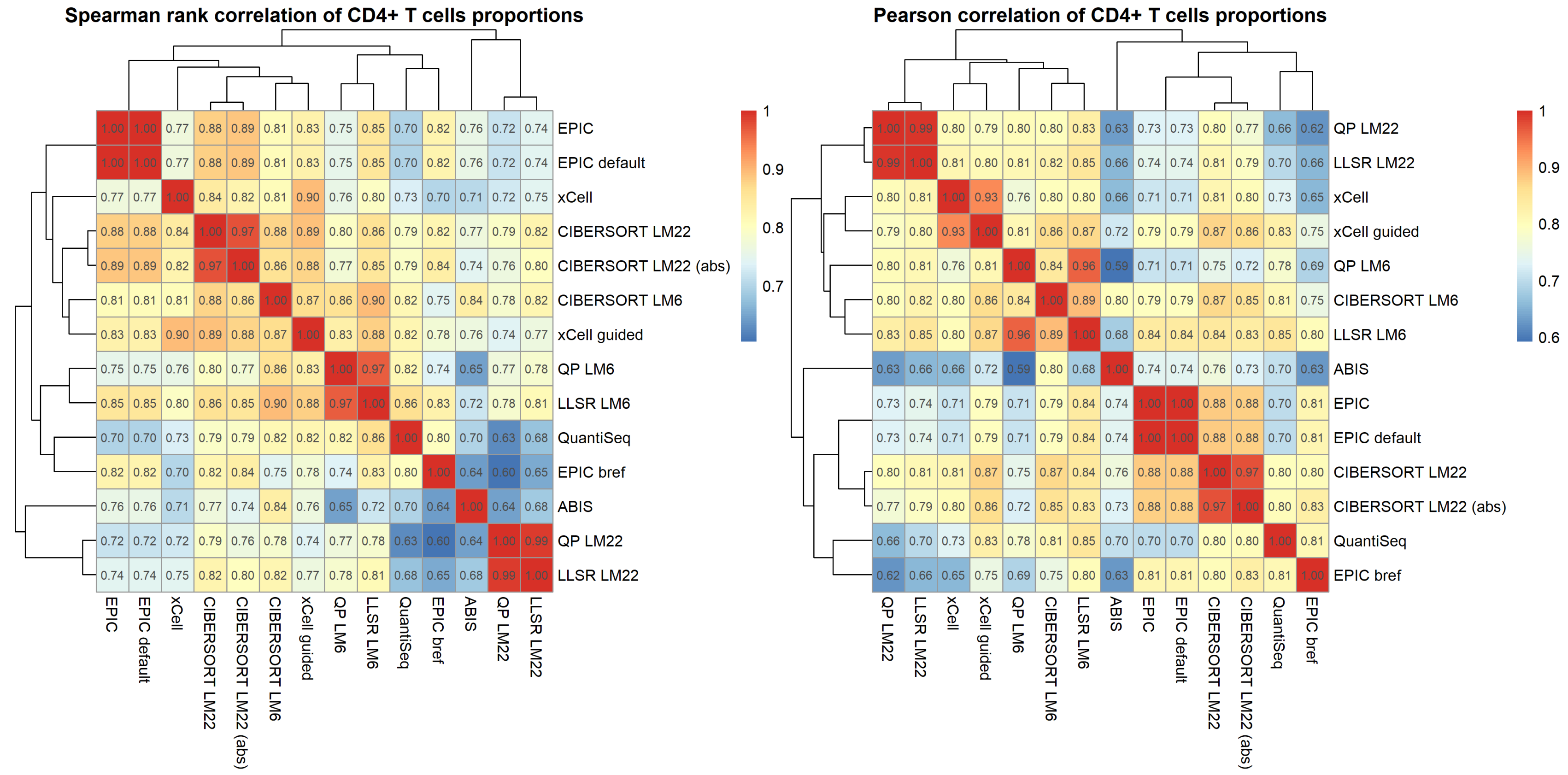


Figure B.5: Proportions of CD4+ T cells in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



(a) : Spearman rank correlation of found fraction of CD4+ T cells by selected deconvolution methods

(b) : Pearson correlation of found fraction of CD4+ T cells by selected deconvolution methods

Figure B.6: Spearman's and Pearson's correlation coefficients for the computed fractions of CD4+ T cells.

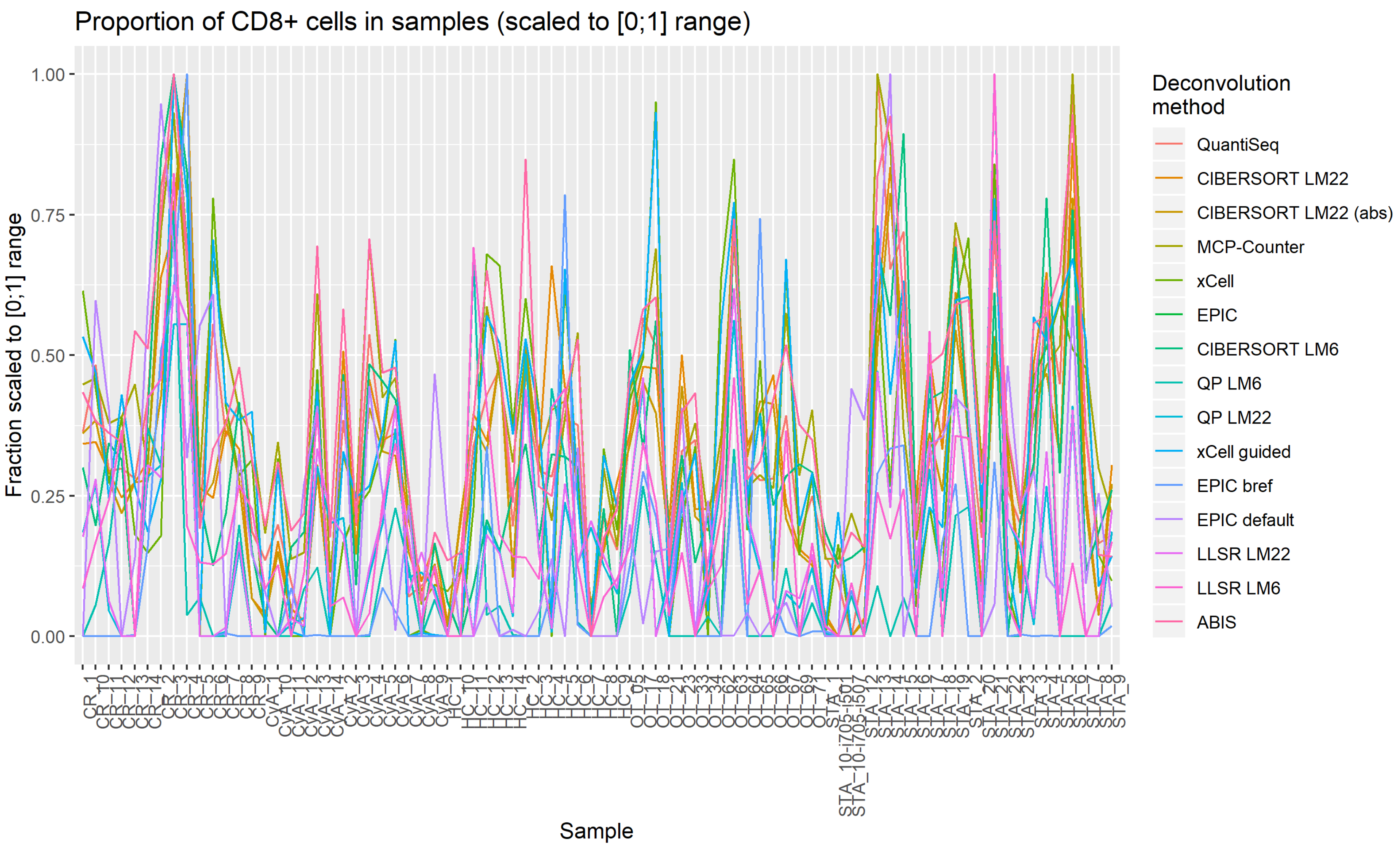


Figure B.7: Proportions or scores of CD8+ T cells in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

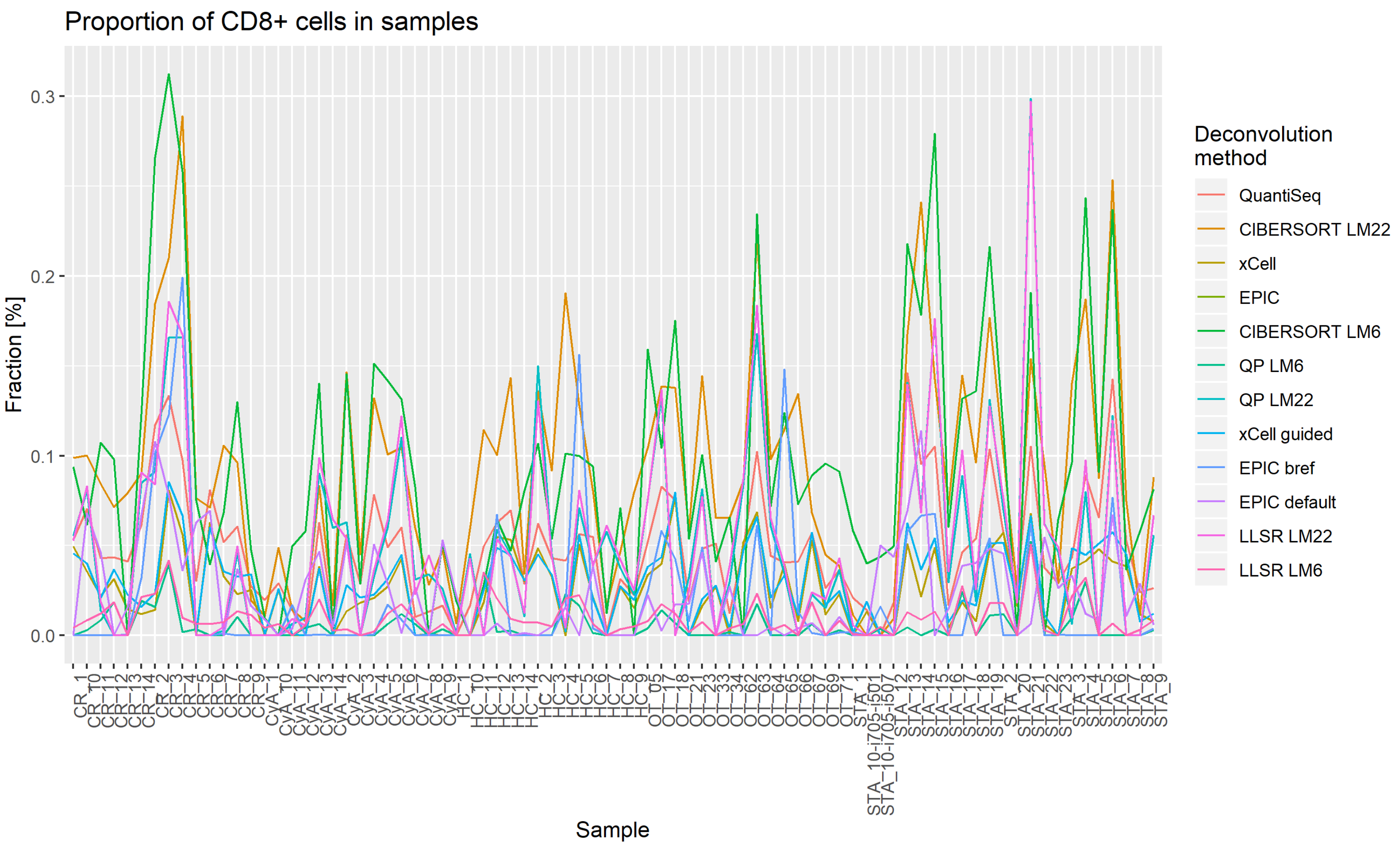
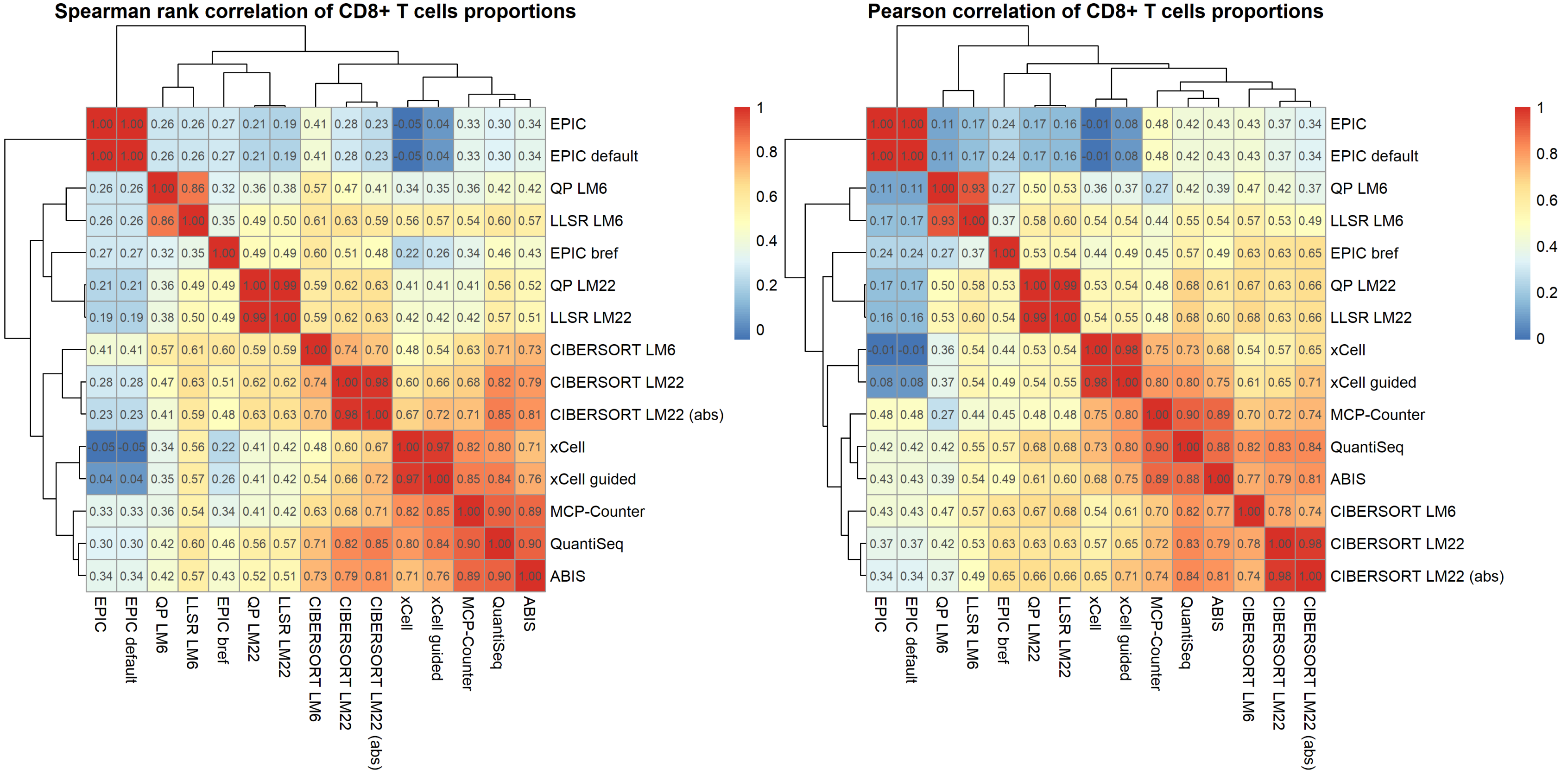


Figure B.8: Proportions of CD8+ T cells in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



(a) : Spearman rank correlation of found fraction of CD8+ T cells by selected deconvolution methods

(b) : Pearson correlation of found fraction of CD8+ T cells by selected deconvolution methods

Figure B.9: Spearman's and Pearson's correlation coefficients for the computed fractions of CD8+ T cells.

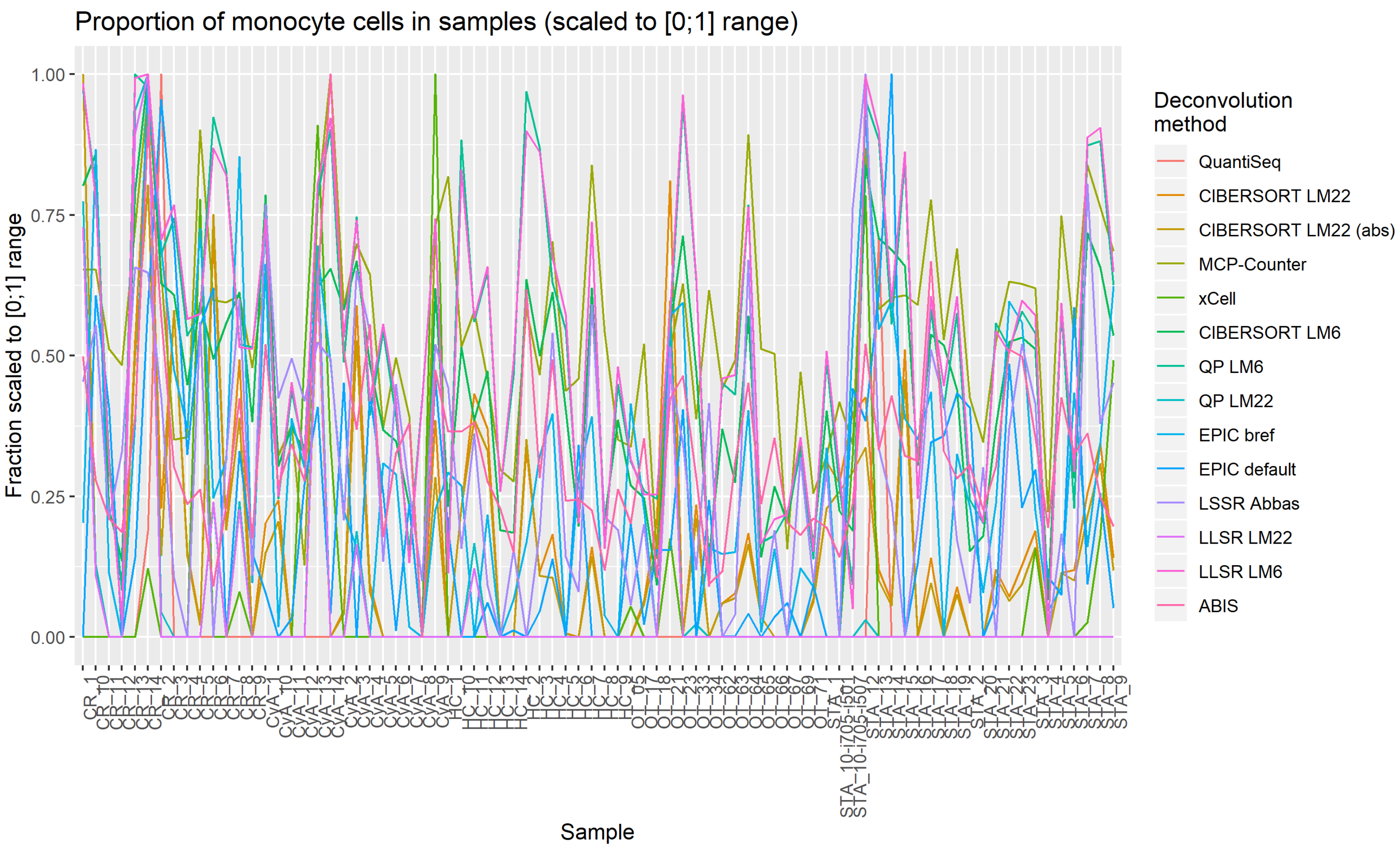


Figure B.10: Proportions or scores of monocytes in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

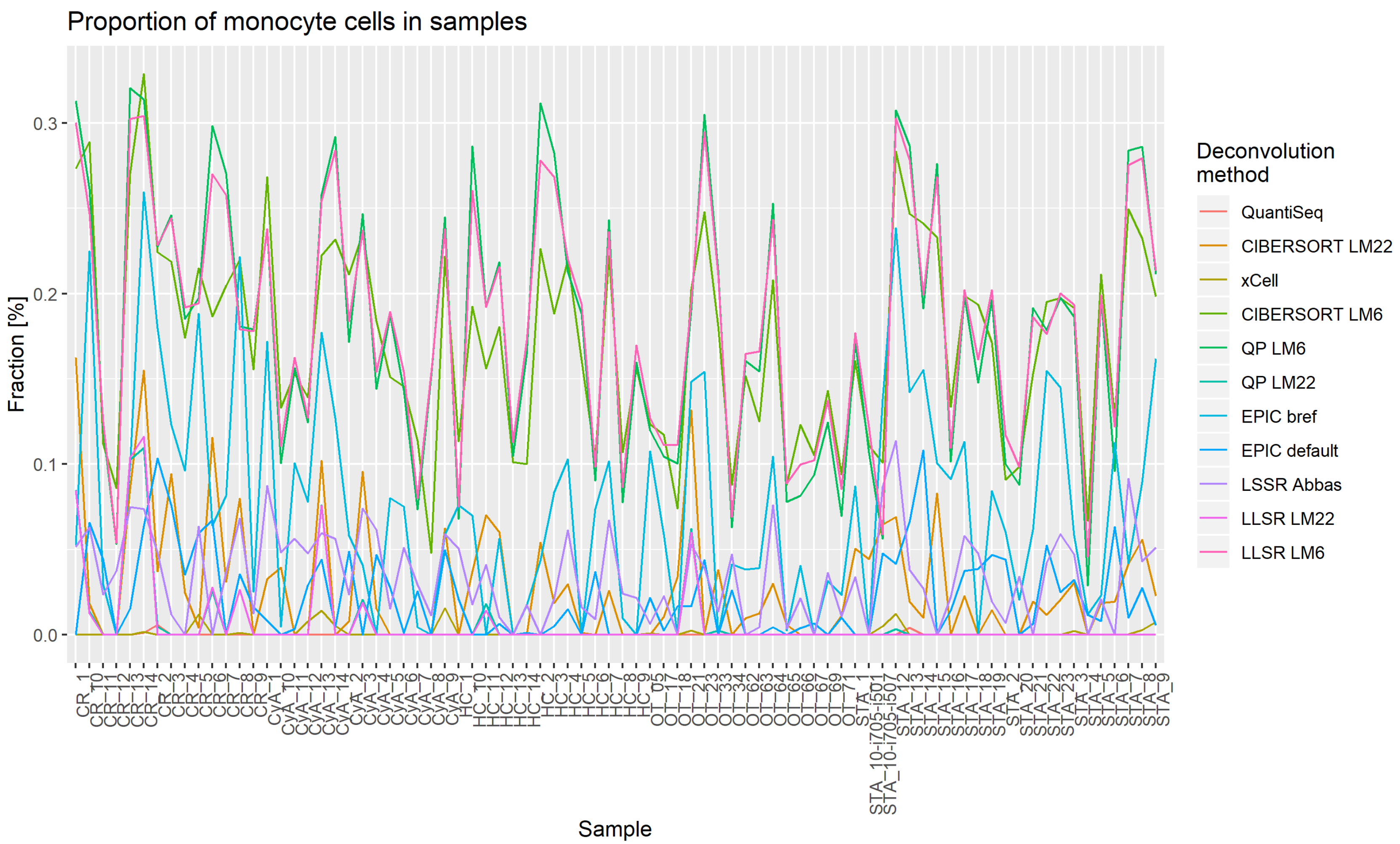
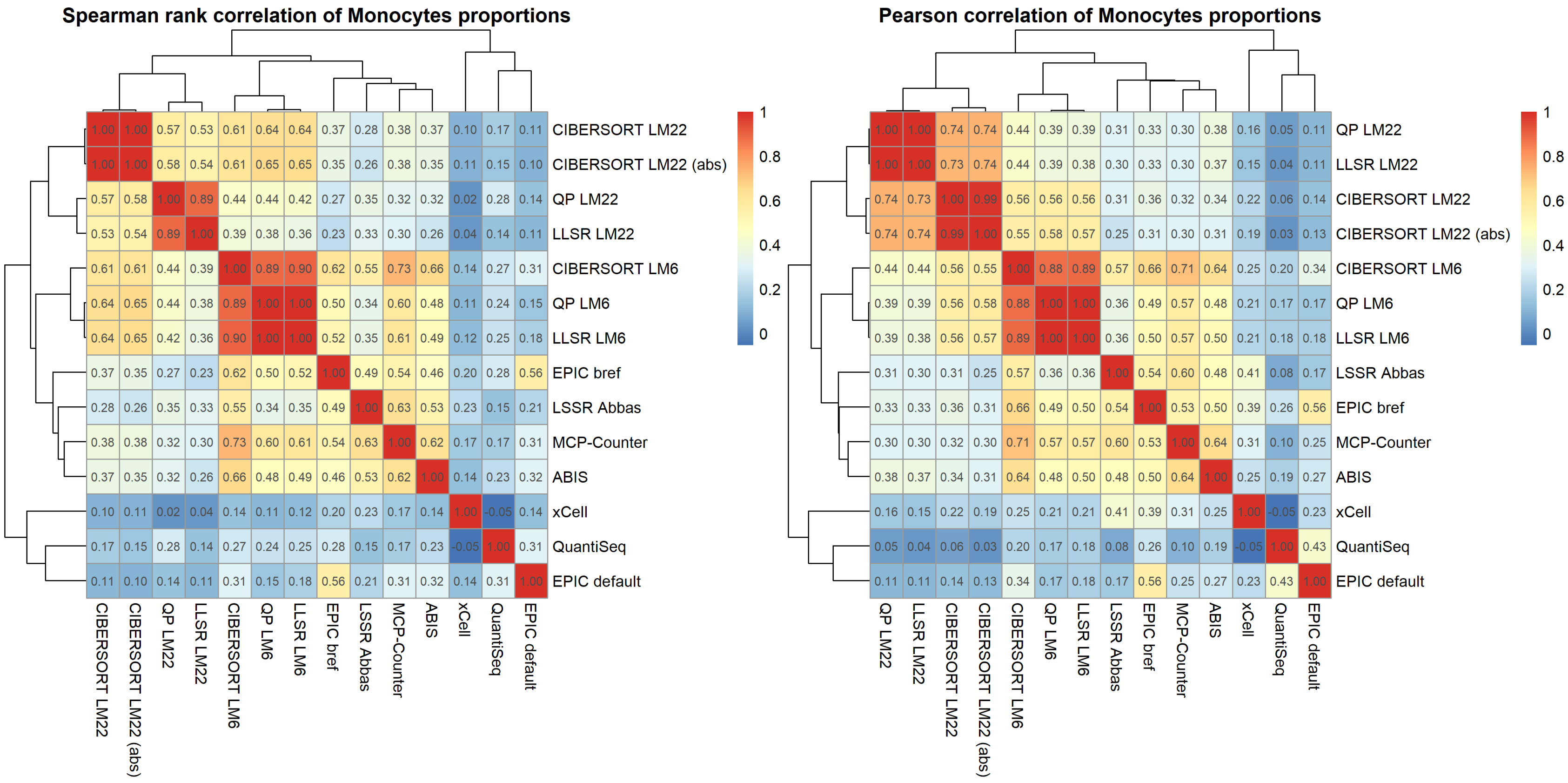


Figure B.11: Proportions of monocytes in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



	CIBERSORT LM22	CIBERSORT LM22 (abs)	QP LM22	LLSR LM22	CIBERSORT LM6	QP LM6	LLSR LM6	EPIC bref	LSSR Abbas	MCP-Counter	ABIS	xCell	QuantiSeq	EPIC default
CIBERSORT LM22	1.00	1.00	0.57	0.53	0.61	0.64	0.64	0.37	0.28	0.38	0.37	0.10	0.17	0.11
CIBERSORT LM22 (abs)	1.00	1.00	0.58	0.54	0.61	0.65	0.65	0.35	0.26	0.38	0.35	0.11	0.15	0.10
QP LM22	0.57	0.58	1.00	0.89	0.44	0.44	0.42	0.27	0.35	0.32	0.32	0.02	0.28	0.14
LLSR LM22	0.53	0.54	0.89	1.00	0.39	0.38	0.36	0.23	0.33	0.30	0.26	0.04	0.14	0.11
CIBERSORT LM6	0.61	0.61	0.44	0.39	1.00	0.89	0.90	0.62	0.55	0.73	0.66	0.14	0.27	0.31
QP LM6	0.64	0.65	0.44	0.38	0.89	1.00	1.00	0.50	0.34	0.60	0.48	0.11	0.24	0.15
LLSR LM6	0.64	0.65	0.42	0.36	0.90	1.00	1.00	0.52	0.35	0.61	0.49	0.12	0.25	0.18
EPIC bref	0.37	0.35	0.27	0.23	0.62	0.50	0.52	1.00	0.49	0.54	0.46	0.20	0.28	0.56
LSSR Abbas	0.28	0.26	0.35	0.33	0.55	0.34	0.35	0.49	1.00	0.63	0.53	0.23	0.15	0.21
MCP-Counter	0.38	0.38	0.32	0.30	0.73	0.60	0.61	0.54	0.63	1.00	0.62	0.17	0.17	0.31
ABIS	0.37	0.35	0.32	0.26	0.66	0.48	0.49	0.46	0.53	0.62	1.00	0.14	0.23	0.32
xCell	0.10	0.11	0.02	0.04	0.14	0.11	0.12	0.20	0.23	0.17	0.14	1.00	-0.05	0.14
QuantiSeq	0.17	0.15	0.28	0.14	0.27	0.24	0.25	0.28	0.15	0.17	0.23	-0.05	1.00	0.31
EPIC default	0.11	0.10	0.14	0.11	0.31	0.15	0.18	0.56	0.21	0.31	0.32	0.14	0.31	1.00

(a) : Spearman rank correlation of found fraction of monocytes by selected deconvolution methods

	QP LM22	LLSR LM22	CIBERSORT LM22	CIBERSORT LM22 (abs)	CIBERSORT LM6	QP LM6	LLSR LM6	LSSR Abbas	EPIC bref	MCP-Counter	ABIS	xCell	QuantiSeq	EPIC default
QP LM22	1.00	1.00	0.74	0.74	0.44	0.39	0.39	0.31	0.33	0.30	0.38	0.16	0.05	0.11
LLSR LM22	1.00	1.00	0.73	0.74	0.44	0.39	0.38	0.30	0.33	0.30	0.37	0.15	0.04	0.11
CIBERSORT LM22	0.74	0.73	1.00	0.99	0.56	0.56	0.56	0.31	0.36	0.32	0.34	0.22	0.06	0.14
CIBERSORT LM22 (abs)	0.74	0.74	0.99	1.00	0.55	0.58	0.57	0.25	0.31	0.30	0.31	0.19	0.03	0.13
CIBERSORT LM6	0.44	0.44	0.56	0.55	1.00	0.88	0.89	0.57	0.66	0.71	0.64	0.25	0.20	0.34
QP LM6	0.39	0.39	0.56	0.58	0.88	1.00	1.00	0.36	0.49	0.57	0.48	0.21	0.17	0.17
LLSR LM6	0.39	0.38	0.56	0.57	0.89	1.00	1.00	0.36	0.50	0.57	0.50	0.21	0.18	0.18
LSSR Abbas	0.31	0.30	0.31	0.25	0.57	0.36	0.36	1.00	0.54	0.60	0.48	0.41	0.08	0.17
EPIC bref	0.33	0.33	0.36	0.31	0.66	0.49	0.50	0.54	1.00	0.53	0.50	0.39	0.26	0.56
MCP-Counter	0.30	0.30	0.32	0.30	0.71	0.57	0.57	0.60	0.53	1.00	0.64	0.31	0.10	0.25
ABIS	0.38	0.37	0.34	0.31	0.64	0.48	0.50	0.48	0.50	0.64	1.00	0.25	0.19	0.27
xCell	0.16	0.15	0.22	0.19	0.25	0.21	0.21	0.41	0.39	0.31	0.25	1.00	-0.05	0.23
QuantiSeq	0.05	0.04	0.06	0.03	0.20	0.17	0.18	0.08	0.26	0.10	0.19	-0.05	1.00	0.43
EPIC default	0.11	0.11	0.14	0.13	0.34	0.17	0.18	0.17	0.56	0.25	0.27	0.23	0.43	1.00

(b) : Pearson correlation of found fraction of monocytes by selected deconvolution methods

Figure B.12: Spearman's and Pearson's correlation coefficients for the computed fractions of neutrophil cells.

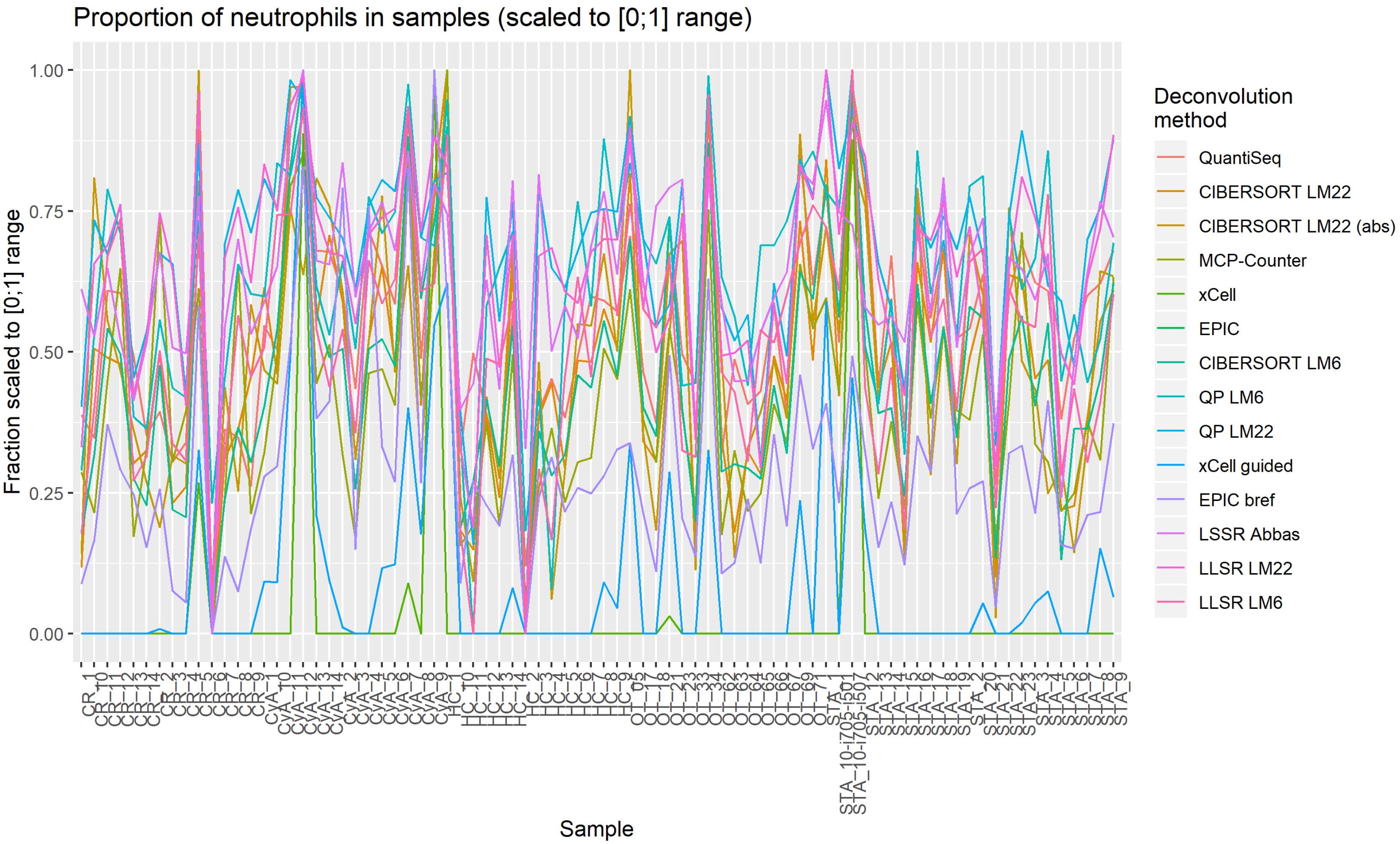


Figure B.13: Proportions or scores of neutrophils in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

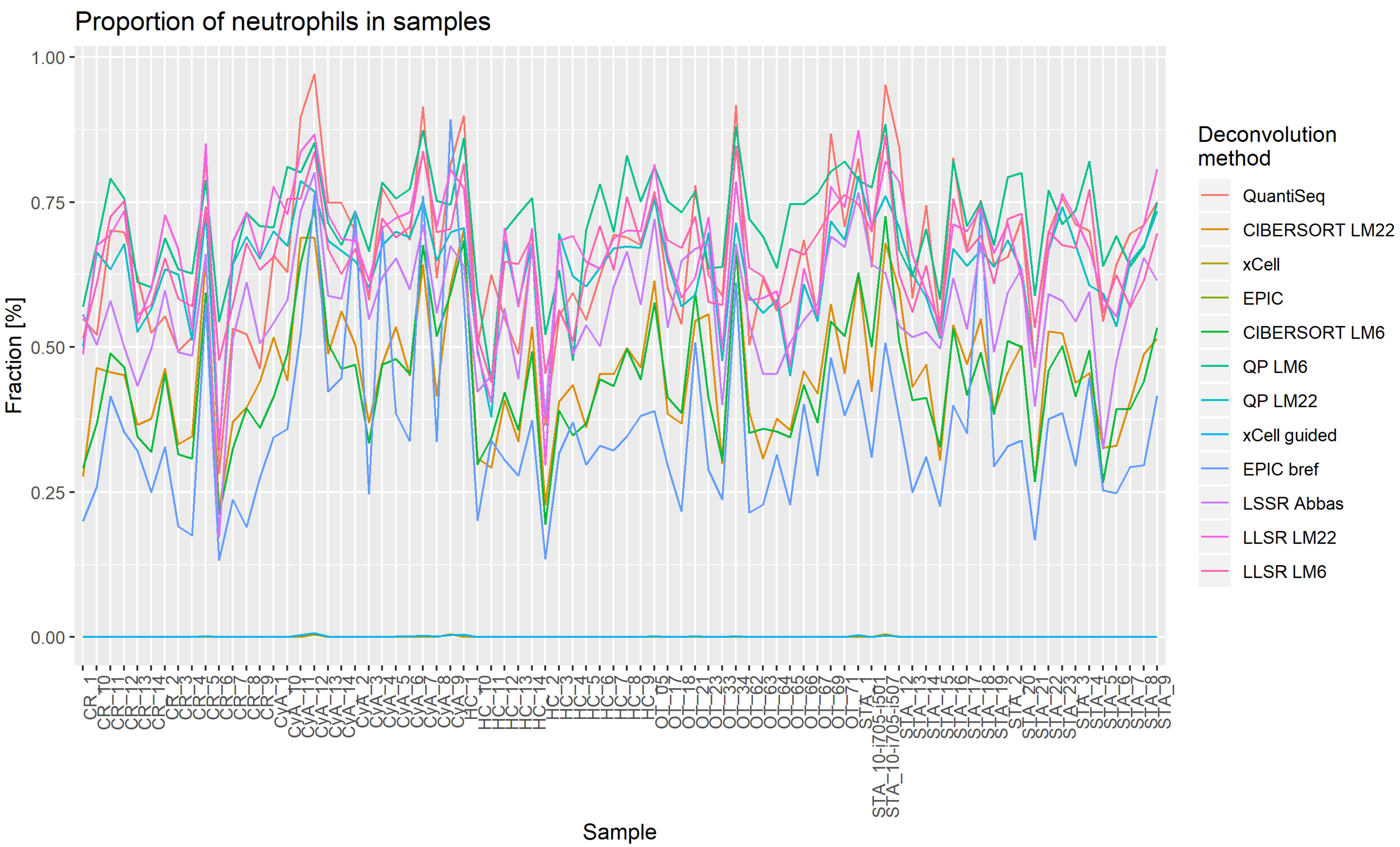
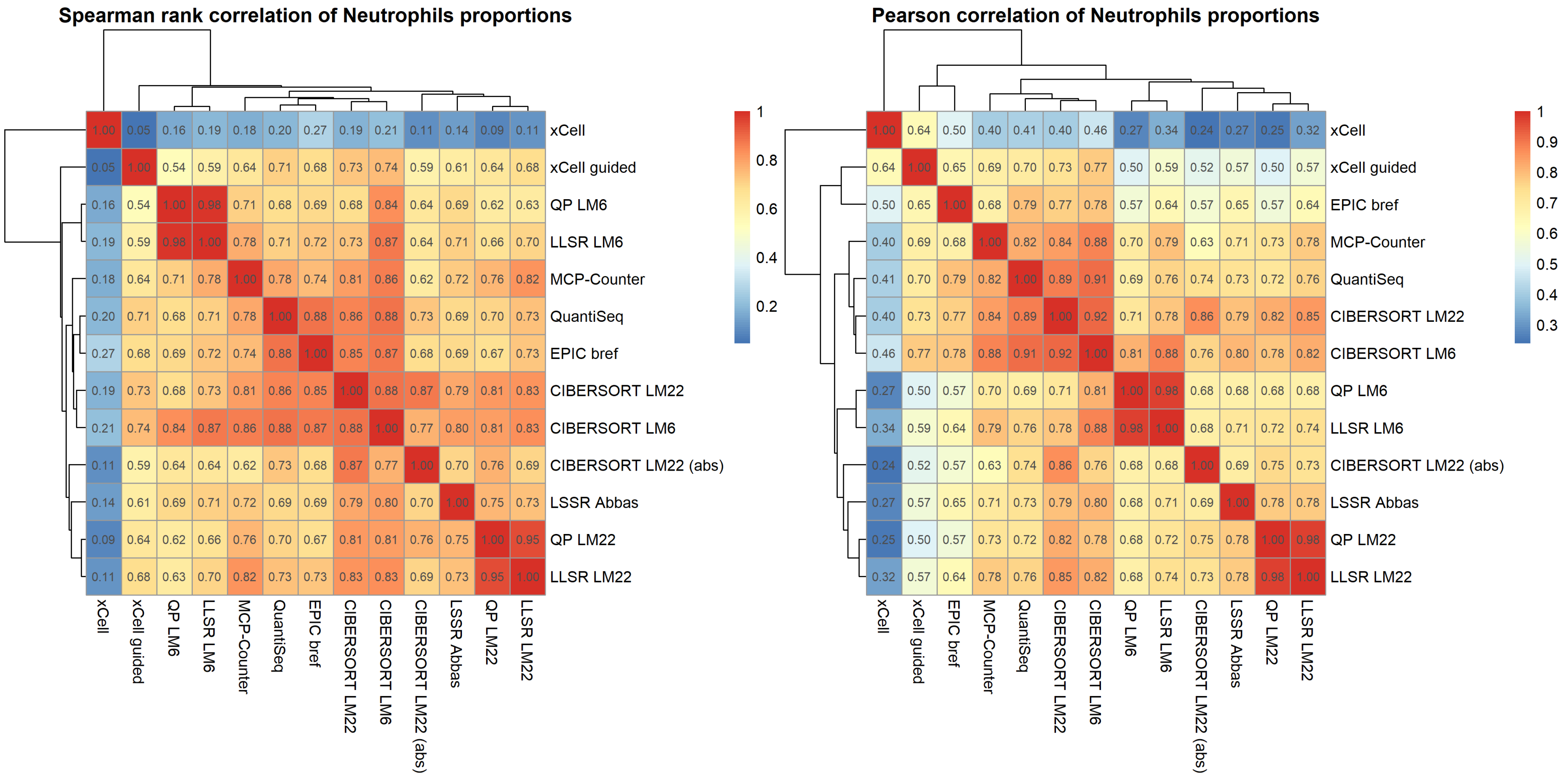


Figure B.14: Proportions of neutrophils in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



(a) : Spearman rank correlation of found fraction of neutrophil cells by selected deconvolution methods

(b) : Pearson correlation of found fraction of neutrophil cells by selected deconvolution methods

Figure B.15: Spearman's and Pearson's correlation coefficients for the computed fractions of neutrophil cells.

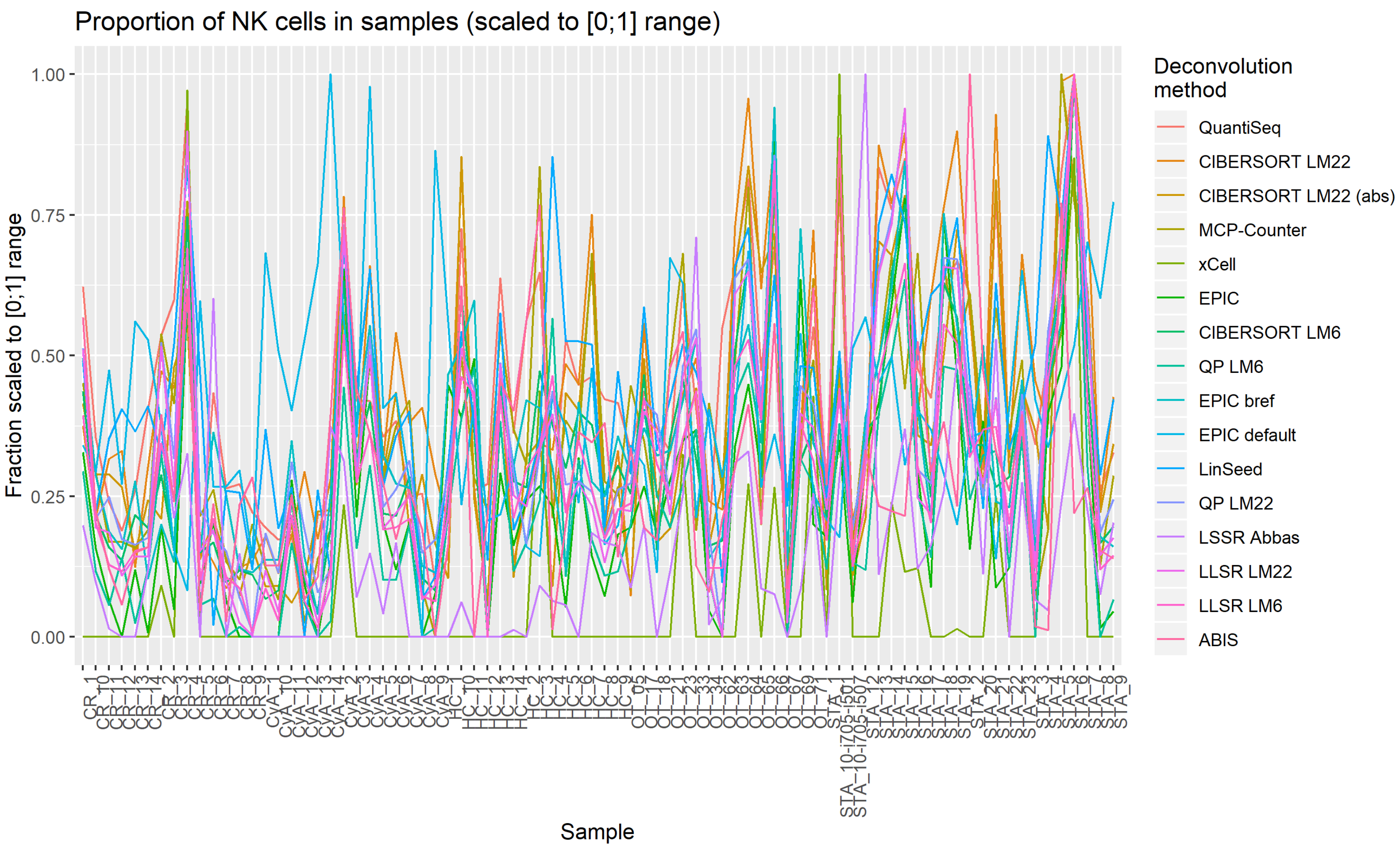


Figure B.16: Proportions or scores of NK cells in all samples as reported by different deconvolution methods. For each method, all the values were scaled to range [0;1].

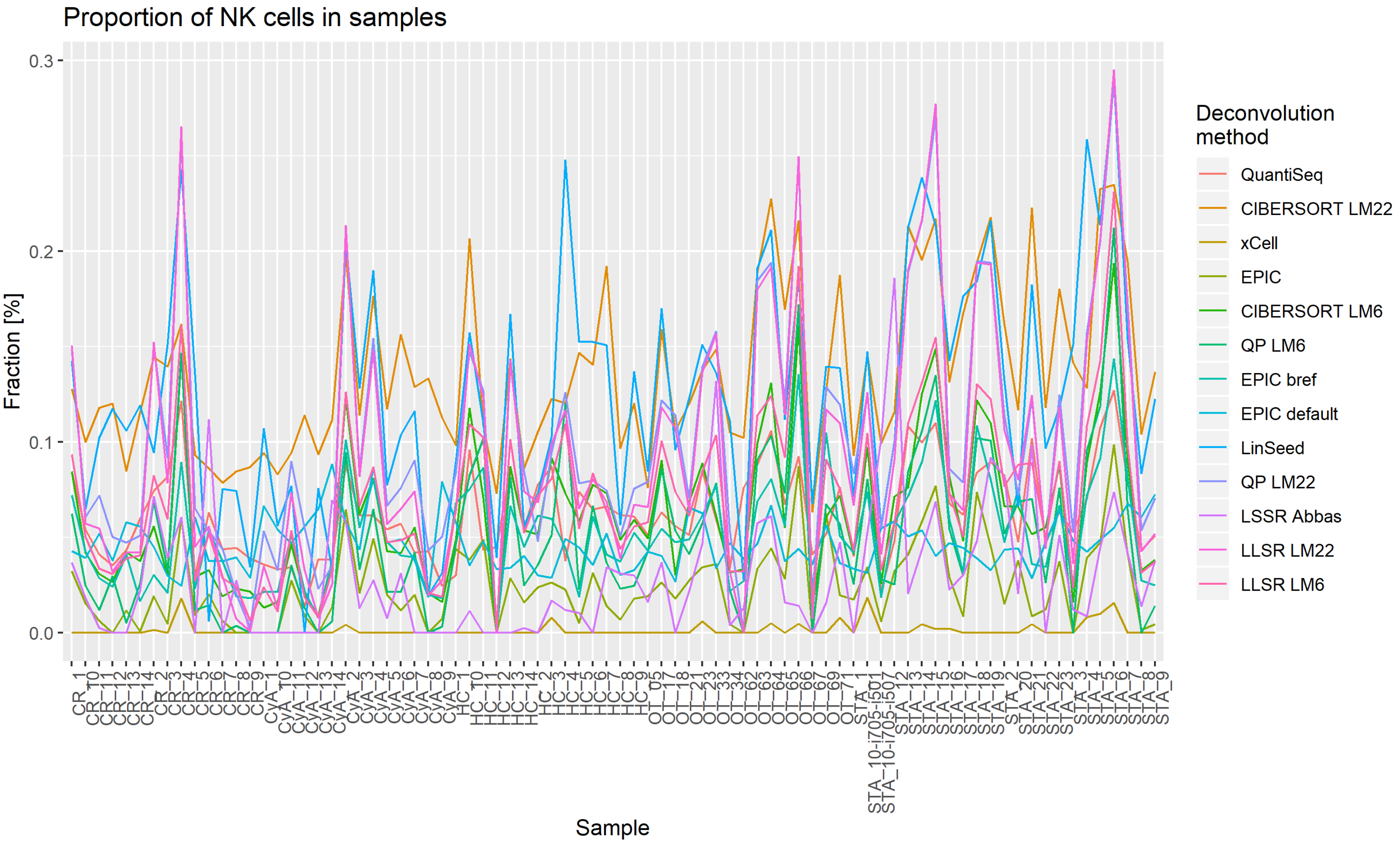
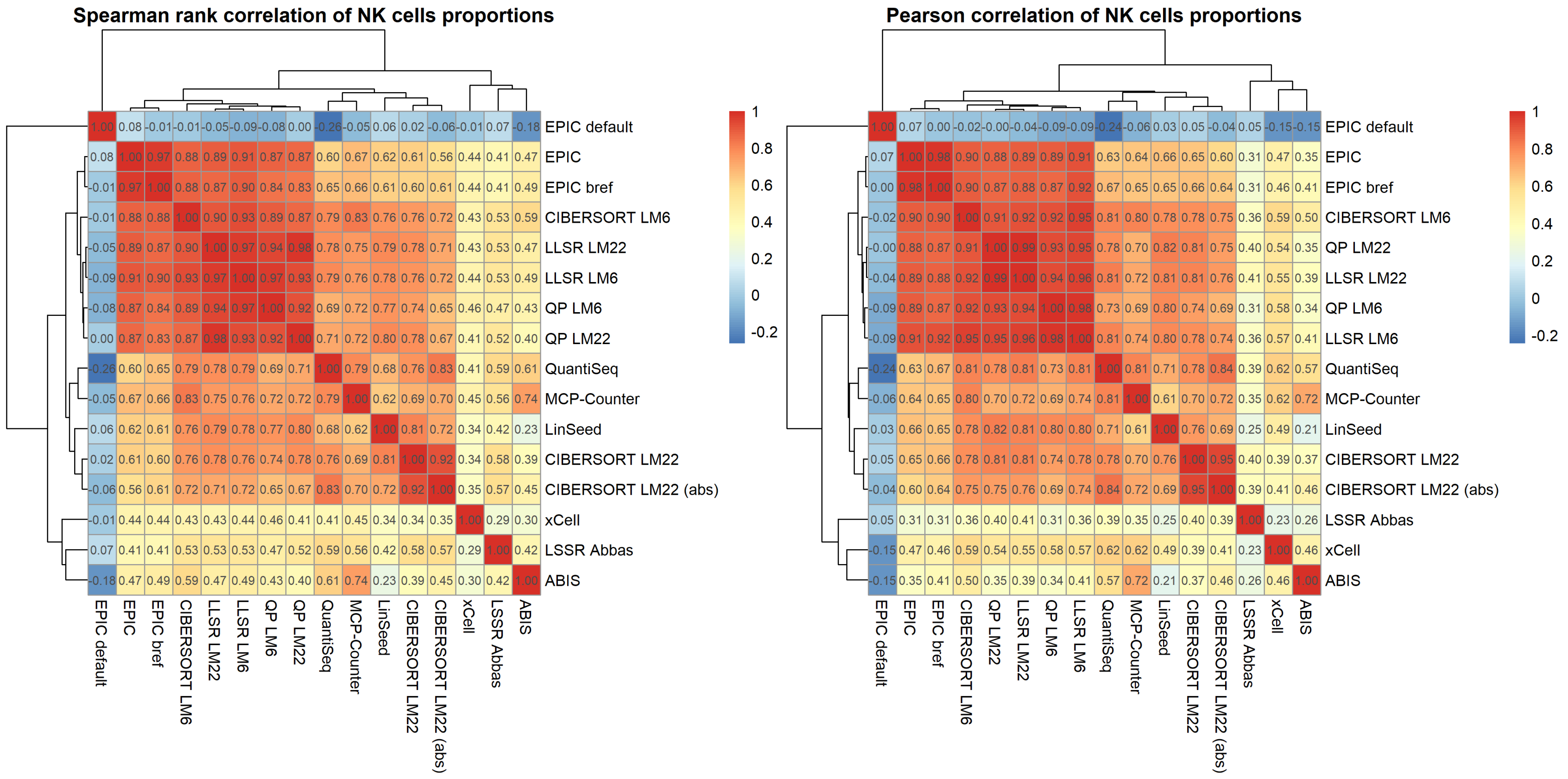


Figure B.17: Proportions of NK cells in all samples as reported by different deconvolution methods. Methods not producing values resembling percentages are omitted.



(a) : Spearman rank correlation of found fraction of NK cells by selected deconvolution methods

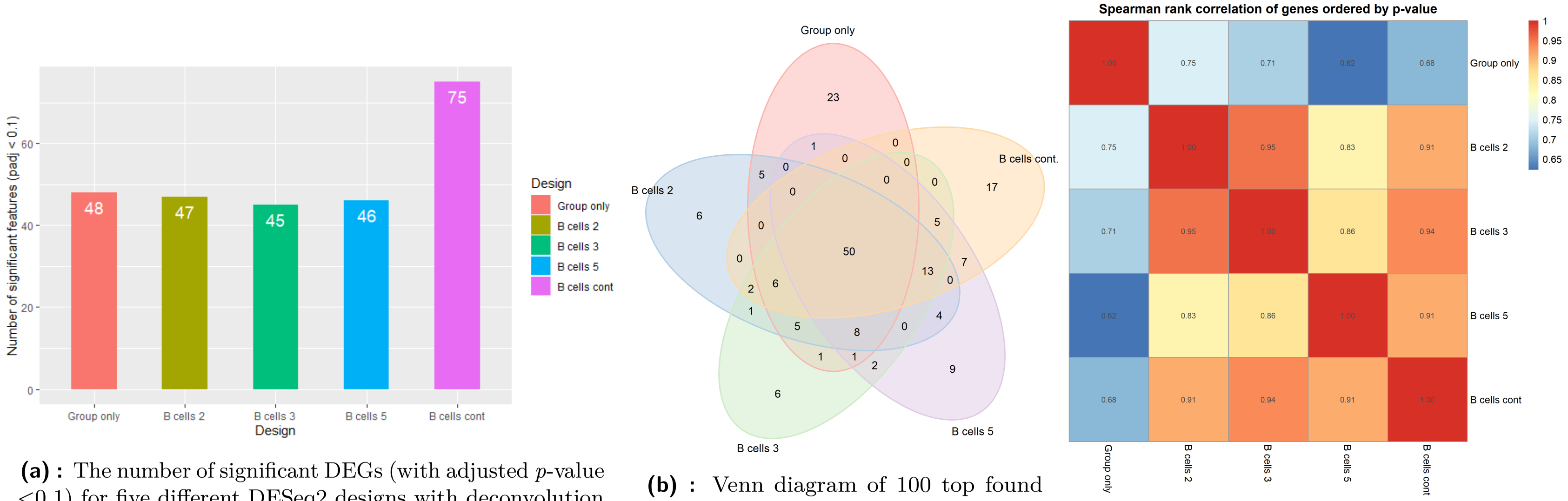
(b) : Pearson correlation of found fraction of NK cells by selected deconvolution methods

Figure B.18: Spearman's and Pearson's correlation coefficients for the computed fractions of NK cells.

Appendix C

DESeq2 experiments — figures

This appendix contains of figures described and discussed in the Section 5.2.3 in the main text of the thesis.

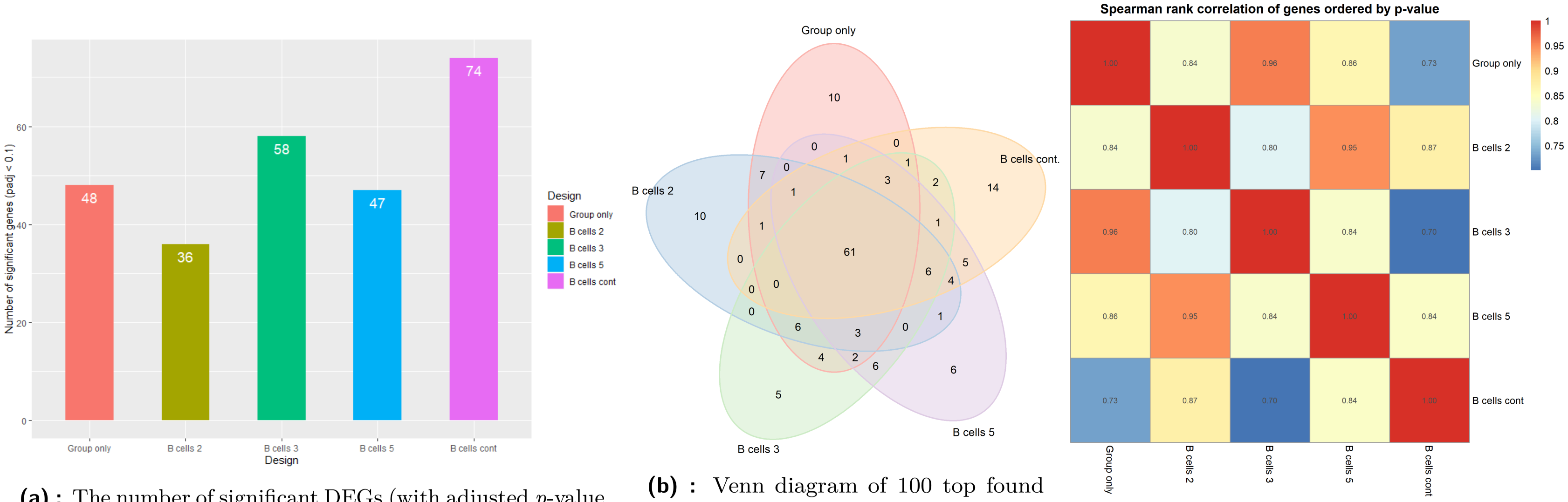


(a) : The number of significant DEGs (with adjusted p-value <0.1) for five different DESeq2 designs with deconvolution results from **EPIC**

(b) : Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by five different designs with data from **EPIC**

(c) : Spearman's rank correlation of gene ordering for five different DESeq2 designs, with B cell proportions from **EPIC**.

Figure C.1: Comparison of four different DESeq2 designs, incorporating the B cell fractions computed by **EPIC**. The designs are compared to each other and to baseline design with no additional B cell information added (marked as *Group only*). We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.

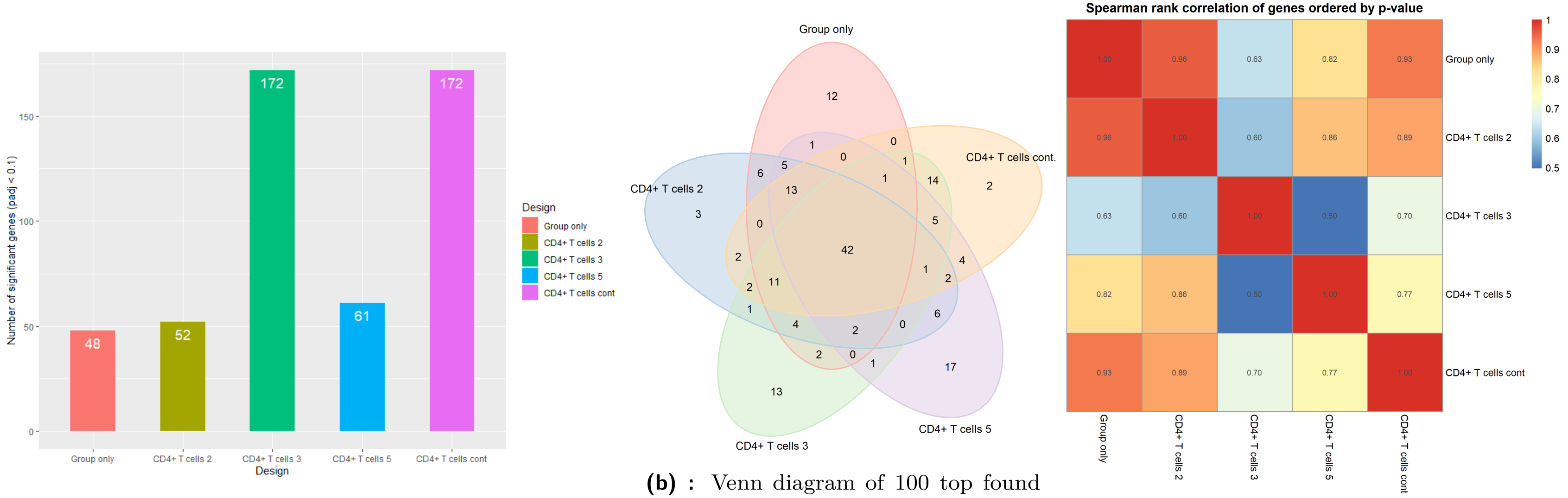


(a) : The number of significant DEGs (with adjusted p-value <0.1) for five different DESeq2 designs with deconvolution results from **CIBERSORT LM6**

(b) : Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by five different designs with data from **CIBERSORT LM6**

(c) : Spearman's rank correlation of gene ordering for five different DESeq2 designs, with B cell proportions from **CIBERSORT LM6**.

Figure C.2: Comparison of four different DESeq2 designs, incorporating the B cell fractions computed by **CIBERSORT LM6**. The designs are compared to each other and to baseline design with no additional B cell information added (marked as *Group only*). We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.

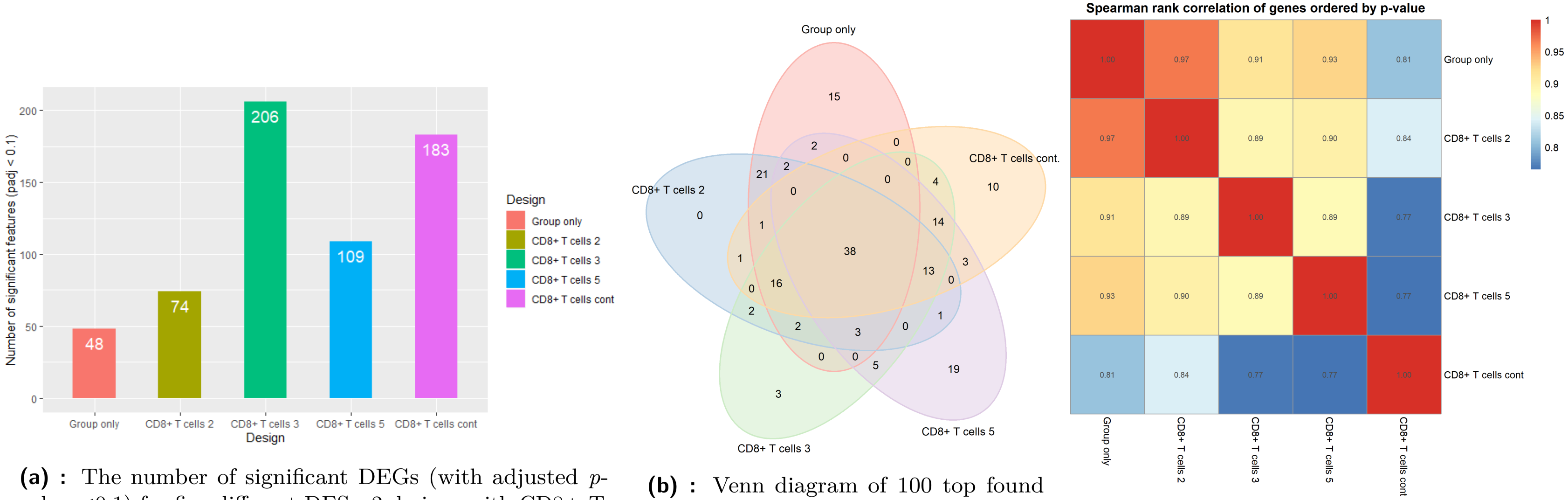


(a) : The number of significant DEGs (with adjusted *p*-value <0.1) for five different DESeq2 designs with CD4+ T cells data from **CIBERSORT LM6**

(b) : Venn diagram of 100 top found genes, sorted by *p*-value, uniquely and identically found by five different designs with CD4+ T cells data from **CIBERSORT LM6**

(c) : Spearman's rank correlation of gene ordering for five different DESeq2 designs, with CD4+ T cell proportions from **CIBERSORT LM6**.

Figure C.3: Comparison of four different DESeq2 designs, incorporating the CD4+ T cell fractions computed by **CIBERSORT LM6**. The designs are compared to each other and to baseline design with no additional CD4+ T cell information added (marked as *Group only*). We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.

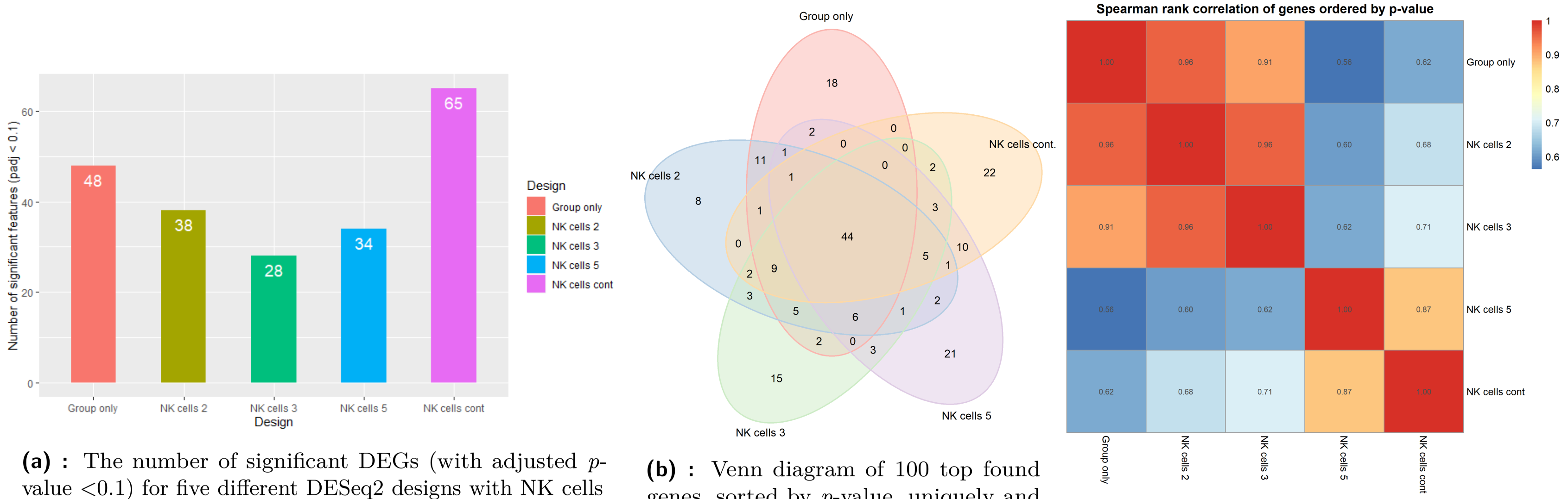


(a) : The number of significant DEGs (with adjusted p-value <0.1) for five different DESeq2 designs with CD8+ T cells data from **CIBERSORT LM6**

(b) : Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by five different designs with CD8+ T cells data from **CIBERSORT LM6**

(c) : Spearman's rank correlation of gene ordering for five different DESeq2 designs, with CD8+ T cell proportions from **CIBERSORT LM6**.

Figure C.4: Comparison of four different DESeq2 designs, incorporating the CD8+ T cell fractions computed by **CIBERSORT LM6**. The designs are compared to each other and to baseline design with no additional CD8+ T cell information added (marked as *Group only*). We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.



(a) : The number of significant DEGs (with adjusted p-value <0.1) for five different DESeq2 designs with NK cells data from **CIBERSORT LM6**

(b) : Venn diagram of 100 top found genes, sorted by p-value, uniquely and identically found by five different designs with NK cells data from **CIBERSORT LM6**

(c) : Spearman's rank correlation of gene ordering for five different DESeq2 designs, with NK cell proportions from **CIBERSORT LM6**.

Figure C.5: Comparison of four different DESeq2 designs, incorporating NK cell fractions computed by **CIBERSORT LM6**. The designs are compared to each other and to baseline design with no additional NK cell information added (marked as *Group only*). We compare the designs based on the number of significantly DE genes, number of uniquely and identically identified genes and by Spearman's correlation.

Appendix D

Gene set enrichment analysis

The gene set enrichment analysis (GSEA) was performed using the *gage* R library [69]. The *gage* was used for enrichment of Gene Ontology (GO) terms [109] [8], specifically with the *go.sets.hs* dataset, provided by *gage*, consisting of 17202 GO terms. Out of those, the GO terms belonging to the *biological process* subtree were selected.

In the following tables, top 6 up-regulated GO terms are shown for each specified design. The reported p-values and multiple testing corrected p-values (marked as **p adj**) are also shown.

The motivation for GSEA and discussion of the obtained results is present in the main text, in Section 5.2.4.

Group only			Bcell_2			Bcell_3			Bcell_5			Bcell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	0.0001445995	0.2870989	GO:0006613	0.0001370592	0.2050701	GO:0006613	0.002151692	0.9168501	GO:0006613	0.003293386	0.9025416
GO:0006614	1.652514e-05	0.02482969	GO:0006614	0.0001931417	0.2870989	GO:0006614	0.0001770897	0.2050701	GO:0006614	0.002547295	0.9168501	GO:0006614	0.003771361	0.9025416
GO:0045047	1.954147e-05	0.02482969	GO:0045047	0.0002183183	0.2870989	GO:0045047	0.0002025142	0.2050701	GO:0045047	0.002792644	0.9168501	GO:0045047	0.004097574	0.9025416
GO:0072599	3.003291e-05	0.02482969	GO:0072599	0.0003144061	0.2870989	GO:0006415	0.0002657643	0.2050701	GO:0006415	0.002923561	0.9168501	GO:0006415	0.00409896	0.9025416
GO:0006415	3.115809e-05	0.02482969	GO:0006415	0.0003403556	0.2870989	GO:0006414	0.0002989729	0.2050701	GO:0072599	0.003453224	0.9168501	GO:0072599	0.005094614	0.9025416
GO:0006414	3.560332e-05	0.02482969	GO:0006414	0.0004000449	0.2870989	GO:0072594	0.0003118358	0.2050701	GO:0006414	0.003985478	0.9168501	GO:0006414	0.005177704	0.9025416
GO:0072594	4.036411e-05	0.02482969	GO:0072594	0.0005252381	0.2971532	GO:0072599	0.0003333698	0.2050701	GO:0000184	0.005237939	0.9168501	GO:0072594	0.007247416	0.9025416
GO:0000184	6.852138e-05	0.03688163	GO:0000184	0.0005520728	0.2971532	GO:0000184	0.000527298	0.2609794	GO:0070972	0.006173833	0.9168501	GO:0000184	0.007339644	0.9025416

Table D.1: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with B cells proportions coming from **xCell** incorporated. The models are described in Experiment 5.2.2.

Group only			Bcell_2			Bcell_3			Bcell_5			Bcell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	3.017246e-05	0.05598808	GO:0006613	0.0007821995	0.9128093	GO:0006613	0.002013009	0.8989793	GO:0006613	0.007438125	0.8824686
GO:0006614	1.652514e-05	0.02482969	GO:0006614	4.142393e-05	0.05598808	GO:0006614	0.0009400447	0.9128093	GO:0006614	0.002371582	0.8989793	GO:0006614	0.008450313	0.8824686
GO:0045047	1.954147e-05	0.02482969	GO:0045047	4.97404e-05	0.05598808	GO:0045047	0.0010446	0.9128093	GO:0045047	0.002614635	0.8989793	GO:0045047	0.009068521	0.8824686
GO:0072599	3.003291e-05	0.02482969	GO:0072599	7.067148e-05	0.05598808	GO:0006415	0.00115502	0.9128093	GO:0006415	0.002734989	0.8989793	GO:0006415	0.009954641	0.8824686
GO:0006415	3.115809e-05	0.02482969	GO:0006415	7.527019e-05	0.05598808	GO:0006414	0.001423517	0.9128093	GO:0072599	0.003391553	0.8989793	GO:0072599	0.01065188	0.8824686
GO:0006414	3.560332e-05	0.02482969	GO:0006414	8.879198e-05	0.05598808	GO:0072599	0.001473927	0.9128093	GO:0006414	0.003489936	0.8989793	GO:0006414	0.01180976	0.8824686
GO:0072594	4.036411e-05	0.02482969	GO:0072594	9.101639e-05	0.05598808	GO:0072594	0.001725563	0.9128093	GO:0072594	0.004314127	0.8989793	GO:0072594	0.01192783	0.8824686
GO:0000184	6.852138e-05	0.03688163	GO:0000184	0.0001625702	0.08750339	GO:0000184	0.002184264	0.9128093	GO:0000184	0.004907455	0.8989793	GO:0000184	0.01481349	0.8824686

Table D.2: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with B cells proportions coming from **CIBERSORT LM6** incorporated. The models are described in Experiment 5.2.2.

Group only			Bcell_2			Bcell_3			Bcell_5			Bcell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	7.7702e-07	0.001988701	GO:0006613	0.0002592072	0.4378213	GO:0006613	0.002786883	0.9026611	GO:0006613	0.008444617	0.8698404
GO:0006614	1.652514e-05	0.02482969	GO:0006614	1.164929e-06	0.001988701	GO:0006614	0.0003424451	0.4378213	GO:0006614	0.003438135	0.9026611	GO:0006614	0.009653455	0.8698404
GO:0045047	1.954147e-05	0.02482969	GO:0045047	1.497185e-06	0.001988701	GO:0045047	0.0003920344	0.4378213	GO:0072594	0.003512372	0.9026611	GO:0045047	0.01033455	0.8698404
GO:0072599	3.003291e-05	0.02482969	GO:0006415	2.525448e-06	0.001988701	GO:0072594	0.0005041506	0.4378213	GO:0045047	0.003737362	0.9026611	GO:0006415	0.01189403	0.8698404
GO:0006415	3.115809e-05	0.02482969	GO:0072599	2.531886e-06	0.001988701	GO:0072599	0.000523622	0.4378213	GO:0072599	0.004251593	0.9026611	GO:0072599	0.01195869	0.8698404
GO:0006414	3.560332e-05	0.02482969	GO:0006414	2.771065e-06	0.001988701	GO:0006415	0.0006261378	0.4378213	GO:0006415	0.005186914	0.9026611	GO:0072594	0.01271518	0.8698404
GO:0072594	4.036411e-05	0.02482969	GO:0072594	3.619895e-06	0.002226752	GO:0006414	0.0007117393	0.4378213	GO:0006414	0.005899974	0.9026611	GO:0006414	0.01390253	0.8698404
GO:0000184	6.852138e-05	0.03688163	GO:0000184	7.085776e-06	0.003813919	GO:0000184	0.001015652	0.5466745	GO:0000184	0.006946327	0.9026611	GO:0000184	0.01685091	0.8698404

Table D.3: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with B cells proportions coming from **EPIC** incorporated. The models are described in Experiment 5.2.2.

Group only			CD4+cell_2			CD4+cell_3			CD4+cell_5			CD4+cell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	0.0001676895	0.3090815	GO:0006613	0.0005612675	0.7493705	GO:0006613	0.01542153	0.8896787	GO:0006613	0.001416317	0.9135998
GO:0006614	1.652514e-05	0.02482969	GO:0006614	0.000219226	0.3090815	GO:0006614	0.0006594868	0.7493705	GO:0006614	0.01756337	0.8896787	GO:0006614	0.001635127	0.9135998
GO:0045047	1.954147e-05	0.02482969	GO:0045047	0.0002488351	0.3090815	GO:0006415	0.0007289062	0.7493705	GO:0045047	0.01852788	0.8896787	GO:0045047	0.001795796	0.9135998
GO:0072599	3.003291e-05	0.02482969	GO:0072599	0.0003482488	0.3090815	GO:0045047	0.0007333985	0.7493705	GO:0006415	0.02085	0.8896787	GO:0006415	0.001865355	0.9135998
GO:0006415	3.115809e-05	0.02482969	GO:0006415	0.0003588964	0.3090815	GO:0006414	0.0009062099	0.7493705	GO:0042742	0.02192362	0.8896787	GO:0006414	0.002353585	0.9135998
GO:0006414	3.560332e-05	0.02482969	GO:0006414	0.0004398842	0.3121214	GO:0072599	0.001044176	0.7493705	GO:0072599	0.02297034	0.8896787	GO:0072599	0.002441686	0.9135998
GO:0072594	4.036411e-05	0.02482969	GO:0072594	0.0005073967	0.3121214	GO:0000184	0.001587032	0.9088111	GO:0006414	0.02462622	0.8896787	GO:0072594	0.003191103	0.9135998
GO:0000184	6.852138e-05	0.03688163	GO:0000184	0.0006428217	0.3459988	GO:0072594	0.001688455	0.9088111	GO:0072594	0.02492623	0.8896787	GO:0000184	0.003506459	0.9135998

Table D.4: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with **CD4+ T** cells proportions coming from **CIBERSORT LM6** incorporated

Group only			CD8+cell_2			CD8+cell_3			CD8+cell_5			CD8+cell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	4.529762e-05	0.08249787	GO:0006613	0.001515869	0.9042089	GO:0006613	0.02431412	0.8990949	GO:0006613	0.002835031	0.9025299
GO:0006614	1.652514e-05	0.02482969	GO:0006614	5.808516e-05	0.08249787	GO:0006614	0.001734421	0.9042089	GO:0006614	0.02495792	0.8990949	GO:0006614	0.003175331	0.9025299
GO:0045047	1.954147e-05	0.02482969	GO:0045047	6.72596e-05	0.08249787	GO:0045047	0.00193619	0.9042089	GO:0045047	0.02625063	0.8990949	GO:0045047	0.003478937	0.9025299
GO:0072599	3.003291e-05	0.02482969	GO:0006415	9.743179e-05	0.08249787	GO:0006415	0.002195423	0.9042089	GO:0006415	0.03075278	0.8990949	GO:0006415	0.003903349	0.9025299
GO:0006415	3.115809e-05	0.02482969	GO:0072599	0.0001018422	0.08249787	GO:0072599	0.002954206	0.9042089	GO:0034660	0.03080869	0.8990949	GO:0072599	0.004897199	0.9025299
GO:0006414	3.560332e-05	0.02482969	GO:0006414	0.0001163072	0.08249787	GO:0006414	0.00297778	0.9042089	GO:0072594	0.03093121	0.8990949	GO:0006414	0.004905904	0.9025299
GO:0072594	4.036411e-05	0.02482969	GO:0072594	0.0001341117	0.08249787	GO:0000184	0.003432781	0.9042089	GO:0050922	0.03102301	0.8990949	GO:0072594	0.005697376	0.9025299
GO:0000184	6.852138e-05	0.03688163	GO:0000184	0.0001956746	0.1053218	GO:0072594	0.003678052	0.9042089	GO:0000184	0.03450036	0.8990949	GO:0000184	0.006041963	0.9025299

Table D.5: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with **CD8+ T** cells proportions coming from **CIBERSORT LM6** incorporated.

Group only			NKcell_2			NKcell_3			NKcell_5			NKcell_cont		
GO term	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**	**GO term**	**p-value**	**p adj**
GO:0006613	1.189391e-05	0.02482969	GO:0006613	0.0004025308	0.4729602	GO:0006613	3.421546e-05	0.04967458	GO:0006613	0.01275847	0.9050324	GO:0006613	0.01418561	0.8836663
GO:0006614	1.652514e-05	0.02482969	GO:0006614	0.000486016	0.4729602	GO:0006614	4.143496e-05	0.04967458	GO:0006614	0.01433141	0.9050324	GO:0006614	0.01525831	0.8836663
GO:0045047	1.954147e-05	0.02482969	GO:0045047	0.0005342703	0.4729602	GO:0045047	4.530645e-05	0.04967458	GO:0006412	0.01441533	0.9050324	GO:0045047	0.01567108	0.8836663
GO:0072599	3.003291e-05	0.02482969	GO:0006415	0.0005870856	0.4729602	GO:0072599	4.999332e-05	0.04967458	GO:0045047	0.01467966	0.9050324	GO:0006415	0.01613662	0.8836663
GO:0006415	3.115809e-05	0.02482969	GO:0072599	0.0006424661	0.4729602	GO:0006414	6.068357e-05	0.04967458	GO:0006415	0.01689604	0.9050324	GO:0072599	0.01678237	0.8836663
GO:0006414	3.560332e-05	0.02482969	GO:0006414	0.0006590249	0.4729602	GO:0006415	7.495457e-05	0.04967458	GO:0072599	0.01699317	0.9050324	GO:0006414	0.01680409	0.8836663
GO:0072594	4.036411e-05	0.02482969	GO:0072594	0.0009269406	0.5702009	GO:0072594	8.075292e-05	0.04967458	GO:0006414	0.01734079	0.9050324	GO:0072594	0.02162098	0.8836663
GO:0000184	6.852138e-05	0.03688163	GO:0000184	0.001281064	0.6895329	GO:0070972	0.0001463158	0.0787545	GO:0072594	0.02156878	0.9050324	GO:0070972	0.02341897	0.8836663

Table D.6: Top 6 up-regulated GO terms with reported *p*-values and multiple testing corrected *p*-values (marked as **p adj**). Results for 5 different DESeq2 models are shown, 4 of them with **NK** cells proportions coming from **CIBERSORT LM6** incorporated.

Appendix E

Recognized cell types in presented signature matrices and methods

In Table E.1, we present recognized cell types as presented in different signature matrices and methods used in this thesis. The terminology or naming of the different cell types differs greatly between methods, sometimes resulting in ambiguous meaning. This complicates the comparison of results coming from different methods, as it is not clear, which categories are equal. This lead to the need of formalizing the relationship between naming of cell types between methods and signature matrices. One such is shown in Figure E.1, presented by Sturm et al. [106]. The hierarchy of cell types is represented by tree, where value of a node is computed as sum of its children. This is implemented in R package *immunedeconv* [50], which provides a mapping of results between CIBERSORTs LM22, Quantiseq, EPIC, TIMER, xCell and MCPcounter results.

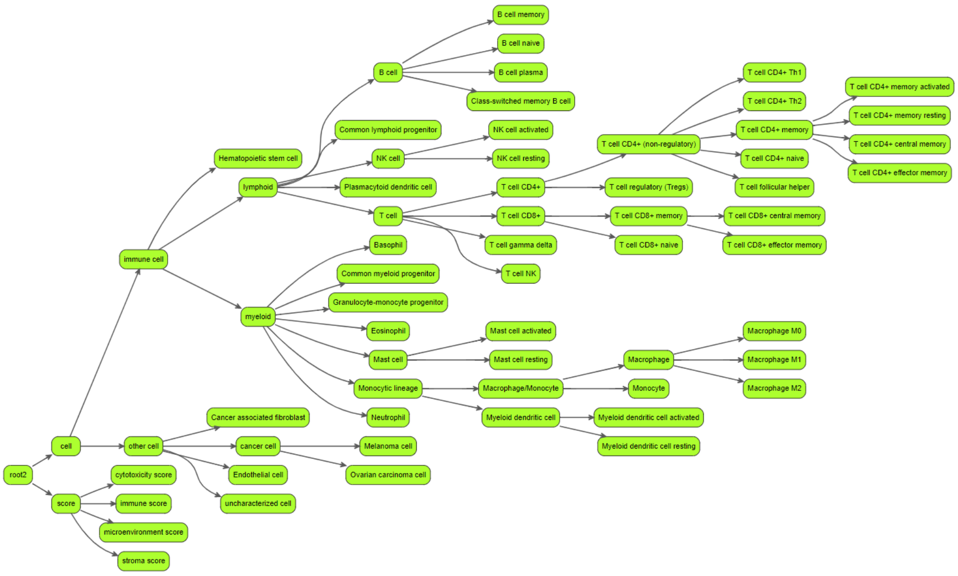


Figure E.1: Immune cell types hierarchy, as defined by Sturm et al. [106]

MCP-counter	**LM22**	**LM6**	**ABIS**	**xCell**	**QuantiSeq**	**Abbas**	**EPIC**
T cell	B.cells.naive	B.cells	Monocytes	Myeloid dendritic cell activated	B cell	Th	Bcells
T cell CD8+	B.cells.memory	CD8.T.cells	NK	B cell	Macrophage M1	Th act	CAFs
cytotoxicity score	Plasma.cells	CD4.T.cells	T CD8 Memory	T cell CD4+ memory	Macrophage M2	Tc	CD4_Tcells
NK cell	T.cells.CD8	NK.cells	T CD4 Naive	T cell CD4+ naive	Monocyte	Tc act	CD8_Tcells
B cell	T.cells.CD4.naive	Monocytes	T CD8 Naive	T cell CD4+ (non-regulatory)	Neutrophil	B	Endothelial
Monocyte	T.cells.CD4.memory.resting	Neutrophils	B Naive	T cell CD4+ central memory	NK cell	B act	Macrophages
Macrophage/Monocyte	T.cells.CD4.memory.activated		T CD4 Memory	T cell CD4+ effector memory	T cell CD4+ (non-regulatory)	B aIgM	NKcells
Myeloid dendritic cell	T.cells.follicular.helper		MAIT	T cell CD8+ naive	T cell CD8+	Mem IgG	otherCells
Neutrophil	T.cells.regulatory..Tregs.		T gd Vd2	T cell CD8+	T cell regulatory (Tregs)	Mem IgM	
Endothelial cell	T.cells.gamma.delta		Neutrophils LD	T cell CD8+ central memory	Myeloid dendritic cell	PC	
Cancer associated fibroblast	NK.cells.resting		T gd non-Vd2	T cell CD8+ effector memory	uncharacterized cell	NK	
	NK.cells.activated		Basophils LD	Class-switched memory B cell		NK act	
	Monocytes		Monocytes NC+I	Common lymphoid progenitor		mono	
	Macrophages.M0		B Memory	Common myeloid progenitor		mono act	
	Macrophages.M1		mDCs	Myeloid dendritic cell		DC	
	Macrophages.M2		pDCs	Endothelial cell		DC act	
	Dendritic.cells.resting		Plasmablasts	Eosinophil		neutro	
	Dendritic.cells.activated			Cancer associated fibroblast			
	Mast.cells.resting			Granulocyte-monocyte progenitor			
	Mast.cells.activated			Hematopoietic stem cell			
	Eosinophils			Macrophage			
	Neutrophils			Macrophage M1			
				Macrophage M2			
				Mast cell			
				B cell memory			
				Monocyte			
				B cell naive			
				Neutrophil			
				NK cell			
				T cell NK			
				Plasmacytoid dendritic cell			
				B cell plasma			
				T cell gamma delta			
				T cell CD4+ Th1			
				T cell CD4+ Th2			
				T cell regulatory (Tregs)			

Table E.1: List of recognized cell types in individual signature matrices or methods. LM22 and LM6 are signature matrices as provided by CIBERSORT. Abbas is signature matrix extracted from the CellMix package, originally described by Abbas et al. [2]. The xCell does not use signature matrices, the list represents cell types that xCell has reference gene datasets for enrichment. Other signature matrices are referred to by the method, which employs them, as the matrix. All cell types names are formatted as outputted from corresponding methods, no post processing or unification was done.