HHS Public Access

Author manuscript

SoftwareX. Author manuscript; available in PMC 2022 July 01.

Published in final edited form as:

SoftwareX. 2022 June ; 18: . doi:10.1016/j.softx.2022.101072.

TumorDecon: A digital cytometry software

Rachel A. Aronow,

Shaya Akbarinejad,

Trang Le,

Sumeyye Su,

Leili Shahriyari*

Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, 01003, USA

Abstract

There are many experimental methods for characterizing immune profiles of tumors, such as flow and mass cytometry. However, these approaches are time and resource intensive. Thus, several "digital cytometry" methods have been developed to extract cell frequencies from RNA-seq data. Here, we introduce TumorDecon, named for its potential to deconvolve the distribution of cells from the gene expression levels of a bulk of cells, such as a tumor. The Python package provides an accessible way of applying these methods. It includes four deconvolution methods as well as several gene sets, signature matrices, and functions for generating custom signature matrices.

Keywords

Deconvolution methods; Digital cytometry; Signature matrix; DeconRNASeq; CIBERSORT; ssGSEA; SingScore

Code metadata

Current code version	1.1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00035
Code Ocean compute capsule	https://codeocean.com/capsule/3124535/tree
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python ≥ 3.6; Linux, Mac OS X, Windows; Can be installed with pip (requirements provided in setup.py file).

*Corresponding author. lshahriyari@umass.edu (Leili Shahriyari).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

If available Link to developer documentation/manual	Link to User Manual available at https://github.com/ShahriyariLab/TumorDecon/blob/master/README.md
Support email for questions	lshahriyari@umass.edu raronow@umass.edu

1. Motivation and significance

Tumors consist of many different cell types, including various immune cells, fibroblasts, and epithelial cells. No two tumors are exactly alike, and thus each tumor may respond differently to identical treatments. Documenting these differences in tumor composition will help us to understand the process of tumorigenesis and to find ways to arrive at effective treatments. However, single cell analysis tools for documenting tumor immune infiltrates such as immunohistochemistry, flow cytometry, and mass cytometry, are time and resource intensive [1]. Recently, several new computational methods have therefore been developed to instead estimate the fraction of each cell type in a bulk of cells from gene expression data. Since RNA sequencing tools are cheaper and high-throughput, these “digital cytometry” methods show a lot of promise for efficiently determining the cellular make-up of tumors [2]. Four of the most commonly used digital cytometry methods are DeconRNASeq [3], CIBERSORT [4], ssGSEA deconvolution method (DM) [5], and SingScore DM [6]. Each of these methods has been coded individually in various programming languages, and a comprehensive review of them has been provided in [2]. However, there is currently no single platform in which all four methods can be run in a convenient and accessible manner. Thus, we introduce TumorDecon, a Python package which includes these four digital cytometry methods that have been used for “deconvolving” cellular frequencies from bulk gene expression data using reference signatures. This package also includes data pre-processing and a new method for creating the signature matrix required for DeconRNASeq and CIBERSORT methods. Furthermore, it provides informative plots of outputs that can be used in publications.

2. Software description

The TumorDecon package streamlines the process of getting the distribution of cell types in a bulk of cells, such as a tumor, from its gene expression profile (GEP), by providing a robust interface for running four of the most commonly-used digital cytometry methods. These methods can be categorized into two groups: linear models and rank-based methods. The two linear models included in TumorDecon are DeconRNASeq [3] and CIBERSORT [4]; these models require a “signature matrix” as additional input, in which each column is a reference expression vector of a specific cell type and each row is a different gene. The goal of these methods is to deconvolve the bulk gene expression data into a vector of cell frequencies and this signature matrix. Alternatively, rank-based methods such as single sample gene set enrichment analysis (ssGSEA) [5] and single-sample gene-set scoring (SingScore) [6] use reference sets of genes that are typically up-regulated (more highly expressed) and down-regulated (less expressed) in each cell type, in place of a signature matrix. These methods produce scores for each gene in the GEP, where the size of the score correlates with the relative frequency of a given cell type within the bulk sample.

To use the software, the user inputs GEP data from one or more patients and chooses which of the four methods to use. Next, depending on if the chosen method is linear or rank-based, the user selects from a list of commonly used signature matrices or gene sets, or chooses to upload or generate their own within TumorDecon. The output of TumorDecon is a spreadsheet of either the frequency (linear models) or score (rank-based methods) of each cell type for each patient in the original gene expression data. This output can be imaged in useful plots (functions provided in the TumorDecon package) or be directly incorporated into various applications, such as quantitative systems pharmacology (QSP) models for personalized cancer treatments.

2.1. Software architecture

TumorDecon follows a simple architecture, which makes it user-friendly for even beginner-level Python users. TumorDecon's pipeline consists of six stages, shown in Fig. 1. In the first stage, gene expression data is given to the program. GEPs can either be downloaded directly from cBioPortal [7] and UCSC Xena [8] within the package, or be in a spreadsheet specified by the user. Linear models further require a signature matrix. This signature matrix can be a pre-defined matrix like LM22, or any uploaded spreadsheet of reference gene expressions. Alternatively, the users can generate a customized signature matrix based on any single-cell reference profiles they have (using TumorDecon's `create_signature_matrix`() function) and use that matrix for deconvolution. Score-based methods require an up-regulated gene set in place of a signature matrix. Again, users can either choose from pre-defined gene sets, or upload their own.

In the third step of the pipeline, we preprocess the GEPs and signature matrix. This stage includes normalization, variance threshold, and converting between different symbols and IDs of genes. Two types of normalization are available: standard scaler (or "z-score") and min–max normalization. Furthermore, to reduce computational complexity and improve accuracy, the user can remove low variant genes by variance threshold. Lastly, the preprocessing step includes functions to address the challenge posed by incompatible gene names between various data sets and signature matrices. For example, the gene expression data might be labeled with HUGO symbols, while the signature matrix labels genes with Ensembl IDs. TumorDecon provides users with the option to convert the HUGO symbol to Ensembl ID and vice versa.

In the next step, one of the four deconvolution methods (DeconRNASeq, CIBERSORT, ssGSEA, or SingScore) is selected. Afterwards, in the post-processing step, there is an option to combine different stages or phases of a cell type. For example, one might be interested in the absolute abundance of mast cells, regardless of their activation status. However, since the LM22 signature matrix has separate values for activated and resting mast cells, linear models would treat these subgroups as separate cell types, and the output would not have a unified value for all mast cells. In this case, TumorDecon can automate the process of summing up the abundance of activated and resting mast cells, and report the total value in a new column named "Mast cells". Finally, the result can be visualized in box plots, pair plots, cluster maps, and bar charts to better represent the outcome.

2.2. Software functionalities

TumorDecon has two main functionalities: generating signature matrices from single-cell RNA-Seq profiles, and implementing four common digital cytometry methods (DeconRNASeq, CIBERSORT, ssGSEA DM, and SingScore DM) in a single and accessible platform.

2.2.1. Generating signature matrix—Our method for deriving a signature matrix consists of three main steps: (1) removing batch effects, (2) clustering within each cell type, and (3) differential expression analysis. Batch effects happen when multiple samples of a given cell type come from different experiments, resulting in variations among GEPs of these samples that are not related to inter- or intra-sample differences [9]. In other words, laboratory conditions can account for large variations in gene expression measurements. Batch effects can be severe and completely distort biological data [10], and therefore ruin the chance of deriving any meaningful inference from GEPs. TumorDecon utilizes the Python implementation of ComBat [11] to tackle batch effects. To do so, if there are multiple experiments for a cell type, the user simply needs to specify the name of all files that include the given cell, and TumorDecon will remove its batch effects. If all cells' expressions come from a single experiment, this step can be skipped.

Since there might be multiple sub-types (clusters) of each cell type, in the next step we try to determine all clusters and their respective gene expression profiles within a cell type. First, by using the Silhouette method, we find the optimal number of clusters (k) within a cell type. Then, we perform the K-means clustering algorithm on GEPs with k as the number of clusters. The gene expression value of each cluster is considered to be the mean expression of that gene among all samples of that cluster.

Next, for obtaining a signature matrix, we need to choose genes that are aberrantly expressed among one cluster compared to other clusters. To achieve this, TumorDecon first selects the genes that are significantly expressed in that cluster compared to others. Specifically, genes are selected such that their adjusted P-values (q-value) are less than 0.05. Among these significantly expressed genes, TumorDecon chooses the top 100 genes with the highest absolute log fold change compared to the mean expression of other clusters. This customized signature matrix can be used by the linear deconvolution methods included in TumorDecon (DeconRNASeq and CIBERSORT).

2.2.2. Performing digital cytometry—The second main functionality of TumorDecon is to provide a simple and accessible platform for running DeconRNASeq, CIBERSORT, ssGSEA DM, and SingScore DM in Python. A detailed example of this functionality is provided in Section 3.

3. Illustrative examples

Here, we demonstrate an example for how to utilize TumorDecon to get the distribution of cell types from bulk gene expression data. We use The Cancer Genome Atlas (TCGA) RNA sequence data provided by cBioPortal. This file contains gene expression values for 17,494 genes and 592 patients, as previewed in the sample below (Table 1).

We wish to know the distribution of cell types in each of these patient's tumors, but only have access to this bulk gene expression data. DeconRNASeq, CIBERSORT, ssGSEA DM, and SingScore DM each provide a different means of deconvolving bulk tumors into their relative cellular composition, and so we can run each method and compare the results. For the linear methods, we will use the default LM22 signature matrix, originally derived by [4]. This signature matrix contains 547 reference gene expression values for 22 cell types. For the rank-based methods, we will use the set of up-regulated genes provided by [2], which were derived from the raw data files used to generate the LM22 signature matrix, and therefore contain genes for the same set of 22 cell types. Here is the sample code for running these four methods.

```
import TumorDecon as td

# Download conlon cancer data from cBio Portal, and fetch Hugo IDs for any genes with only Entrez Gene IDs listed:

rna = td.download_by_name('cbio', 'Uveal Melanoma', fetch_missing_hugo=True)

# Drop any rows (genes) that contain a NaN value from rna expression data
rna.dropna(axis=0, inplace=True)

# Use default signature matrix file (LM22):
sig_mat = td.read_sig_file()
# (Can also pass in a filename of an alternative signature)

# Use default gene set (derived from single-cell reference profiles of LM22, in Le et al. (2020)):
gene_set = td.read_geneset()
# (Can also pass in a filename of an alternative gene set)

# Run TumorDecon on RNA data file, using each of the 4 methods:
decon = td.tumor_deconvolve(rna, 'DeconRNASeq', sig_matrix=sig_mat, args={'scaling':'minmax'})
cibersort = td.tumor_deconvolve(rna, 'cibersort', sig_matrix=sig_mat, args={'scaling':'minmax', 'nu':'best'})
ssgsea = td.tumor_deconvolve(rna, 'ssGSEA', up_genes=gene_set, args={'alpha':0.5, 'norm':True})
singscore = td.tumor_deconvolve(rna, 'singscore', up_genes=gene_set)
```

Note that our implementations of DeconRNASeq, CIBERSORT, and ssGSEA allow for additional customization arguments, as demonstrated in the code above. A full list of optional arguments is provided in the user manual that is available on the Github repository (Table 1).

The output of the `tumor_deconvolve()` function is a pandas dataframe where the rows are the patients from the original gene expression data file, and the columns are the 22 cell types from the signature matrix/gene set used in this example. For the linear models, the values are the percentage of the total number of cells made up by this cell type. These values can be used directly as input for initial conditions in a mathematical model of tumor growth, helping to personalize cancer models for individual patients. For the rank-based models, the values are an output score, where the relative size of the score reflects the frequency of a given cell type in the patient's tumor. These methods' outputs therefore need to be converted to percentages before used in mathematical models of tumor growth, but can still be used directly to draw qualitative conclusions about the prevalence of various cell types in a tumor.

TumorDecon also provides four different means for visualizing these results (Fig. 2). These include:

- **a.** Box plots that show immune cells frequencies in descending order, so that users can easily recognize the most frequent immune cells within a group of samples/patients.
- **b.** Pair plots that illustrate the correlation between different immune cells.
- **c.** Cluster heat maps that group cells based on their euclidean distance.
- **d.** Bar charts that show the estimated percentage of each immune cell within a group of individual samples.

These types of plots help users to summarize the essential takeaways of the results, and have been used in different studies such as immune cell classification of cancer types [12] and mathematical modeling of colon cancer [13].

4. Impact

We have developed a Python software package called TumorDecon that includes four methods (DeconRNASeq, CIBERSORT, ssGSEA DM, and SingScore DM) for performing digital cytometry. The only required input file to the software is a gene expression profile. Users can then choose from reference gene sets (for ssGSEA DM and SingScore DM) or signature matrices (for DeconRNASeq and CIBERSORT) that are included in the software. Alternatively, they can also upload their own signature matrices or reference gene sets, or generate custom ones tailored towards answering a specific research question. Importantly, within the TumorDecon program, users can directly perform digital cytometry on TCGA gene expression profiles of tumors available on cBioPortal [7] and UCSC Xena [8], as well as upload their own data.

The first step of digital cytometry, which is of utmost importance, is creating either a signature matrix or a reference gene set, depending on the method used. While there are a number of generic signature matrices provided in the package, such as LM22 [4] and LM6 [14], the user may wish to create their own signature matrix from single-cell RNA-Seq profiles for a specific purpose. For example, a researcher may want to determine the proportion of cells in a milieu that are not present in any of the generic signature matrices. In this case, they could of course estimate the proportion of an undefined cell by a closely related cell, however, this approach would be highly inaccurate. In TumorDecon, users have the additional option to generate and therefore fully customize a signature matrix for their desired cell types.

Separate implementations of DeconRNASeq, CIBERSORT, and ssGSEA currently exist for R, and are available through R Bioconductor. We adapt these functions for Python, and additionally restructure them so that input is entered in a streamlined and consistent manner for all four methods. We have also incorporated extra customization options not provided in the R implementations. While methods such as DeconRNASeq and CIBERSORT provide only z-score scaling options in their original implementations, we add the capability to scale the data with either min–max normalization or standardized scaling (or to proceed without

normalization). Further, the axis of normalization, i.e. whether to scale data by patient or by gene, can also be specified. This is of particular importance for GEP analysis, as a study of TCGA data sets demonstrated that the axis of normalization can drastically change the resulting distribution of cell counts [15]. For ssGSEA, we also introduce a parameter granting more control on how to handle ties in the ranking of genes. To incorporate the SingScore method into TumorDecon, we have simply built a wrapper function around the Python implementation, PySingscore [16], such that input is consistent among all four methods. This standardization permits a researcher to easily switch between methods without needing to restructure inputs or switch platforms.

Further, we provide useful functions for working within the tumor deconvolution problem. These include pre-processing functions such as reading gene expression data sets directly from cBioPortal and UCSC Xena, normalizing this data, and then eliminating genes that are not present in the reference gene sets or signature matrix (or exhibit variance below a certain threshold). We also provide functions for converting between Entrez Gene ID and Hugo Symbol, generating sets of up- or down-regulated genes from a given signature matrix, and deriving signature matrices from single-cell RNA-Seq profiles. These additions increase the flexibility of the program, allowing a user to easily tailor a tumor deconvolution problem to a specific data set or type of tumor. TumorDecon also provides helpful visualizations of its results, including box plots of the distribution of cells within a group of patients, bar charts of cell frequencies for individual patients, cluster maps for grouping cells and patients with similar cellular profiles, and pair plots for discovering correlations among cell types (Fig. 2).

To test the efficiency of the software, we ran an analysis of 544 unique tests. We considered 32 data sets of various cancer types from cBioPortal, and 36 data sets from UCSC Xena. All 4 digital cytometry methods were applied to each data set, using default values for optional arguments. We tested the linear methods with both the default LM22 signature matrix and with a custom signature matrix (generated via the methodology described in Section 2.2.1). This custom signature matrix is available on GitHub (see Table 1), and can also be recreated by following the `sig_matrix_tutorial.py` provided on GitHub. Rank-based methods were tested with both the up-regulated gene sets derived from LM22 [2], as well as a custom set of up-regulated genes derived from our custom signature matrix. All tests completed with no errors.

5. Conclusions

In this paper, we have described the TumorDecon software package for performing digital cytometry. It provides a faster and cheaper option than current experimental tools such as immunohistochemistry and flow cytometry for estimating the relative frequency of each cell type in a bulk of cells. Additionally, we provide a method of generating custom signature matrices, in order to tailor a tumor deconvolution problem to a specific set of cell types. TumorDecon has many applications, and can be easily integrated into other projects. For example, TumorDecon can be combined with quantitative systems pharmacology models to suggest an optimal treatment option for individual patients based on their tumors' characteristics.

Acknowledgments

This work was supported by the National Cancer Institute of the National Institutes of Health, USA [R21CA242933 to L.S.].

References

[1]. Heath JR, Ribas A, Mischel PS. Single-cell analysis tools for drug discovery and development. Nat Rev Drug Discov 2016;15(3):204–16. 10.1038/nrd.2015.16. [PubMed: 26669673]

[2]. Trang L, Rachel AA, Kirshtein A, Shahriyari L. A review of digital cytometry methods: estimating the relative abundance of cell types in a bulk of cells. Brief Bioinform 2021;22(4):bbaa219. 10.1093/bib/bbaa219. [PubMed: 33003193]

[3]. Abbas AR, Wolslegel K, Seshasayee D, Modrusan Z, Clark HF. Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus. PLoS One 2009;4(7):e6098. 10.1371/journal.pone.0006098. [PubMed: 19568420]

[4]. Newman AM, Liu CL, Green MR, Gentles AJ, Feng W, Xu Y, et al. Robust enumeration of cell subsets from tissue expression profiles. Nature Methods 2015;12(5):453–7. 10.1038/nmeth.3337. [PubMed: 25822800]

[5]. Senbabaoglu Y, Gejman RS, Winer AG, Liu M, Van Allen EM, de Velasco G, et al. Tumor immune microenvironment characterization in clear cell renal cell carcinoma identifies prognostic and immunotherapeutically relevant messenger RNA signatures. Genome Biol 2016;17:231. 10.1186/s13059-016-1092-z. [PubMed: 27855702]

[6]. Foroutan M, Bhuva DD, Lyu R, Horan K, Cursons J, Davis MJ. Single sample scoring of molecular phenotypes. BMC Bioinformatics 2018;6(19):404. 10.1186/s12859-018-2435-4.

[7]. Cerami E, Gao J, Dogrusoz U, Gross BE, Sumer SO, Aksoy BA, et al. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. Cancer Discov 2012;2(5):401–4. 10.1158/2159-8290.CD-12-0095. [PubMed: 22588877]

[8]. Goldman MJ, Craft B, Hastie M, Repecka K, McDade F, Kamath A, et al. Visualizing and interpreting cancer genomics data via the Xena platform. Nat Biotechnol 2020;38(6):675–8. 10.1038/s41587-020-0546-8. [PubMed: 32444850]

[9]. Goh WWB, Wang W, Wong L. Why batch effects matter in omics data, and how to avoid them. Trends Biotechnol 2017;35(6):498–507. 10.1016/j.tibtech.2017.02.012. [PubMed: 28351613]

[10]. Leek J, Scharpf R, Bravo HC, Simcha D, Langmead B, Johnson WE, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. Nat Rev Genet 2010;11:733–9. 10.1038/nrg2825. [PubMed: 20838408]

[11]. Leek JT, Johnson WE, Parker HS, Jaffe AE, Storey JD. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. Bioinformatics 2012;28(6):882–3. 10.1093/bioinformatics/bts034. [PubMed: 22257669]

[12]. Su S, Akbarinejad S, Shahriyari L. Immune classification of clear cell renal cell carcinoma. Sci Rep 2021;11:4338. 10.1038/s41598-021-83767-z. [PubMed: 33619294]

[13]. Kirshtein A, Akbarinejad S, Hao W, Le T, Su S, Aronow RA, et al. Data driven mathematical model of colon cancer progression. J Clinical Med 2020;9:3947. 10.3390/jcm9123947.

[14]. Chen B, Khodadoust MS, Liu CL, Newman AM, Alizadeh AA. Profiling tumor infiltrating immune cells with CIBERSORT. Methods Mol Biol 2018;1711:243–59. 10.1007/978-1-4939-7493-1_12. [PubMed: 29344893]

[15]. Shahriyari L Effect of normalization methods on the performance of supervised learning algorithms applied to HTSeq-FPKM-UQ data sets: 7SK RNA expression as a predictor of survival in patients with colon adenocarcinoma. Brief Bioinform 2019;20(3):985–94. 10.1093/bib/bbx153. [PubMed: 29112707]

[16]. Pysingscore Horan K.. 2019, URL https://github.com/DavisLaboratory/PySingscore. [Accessed: Feb 2021].

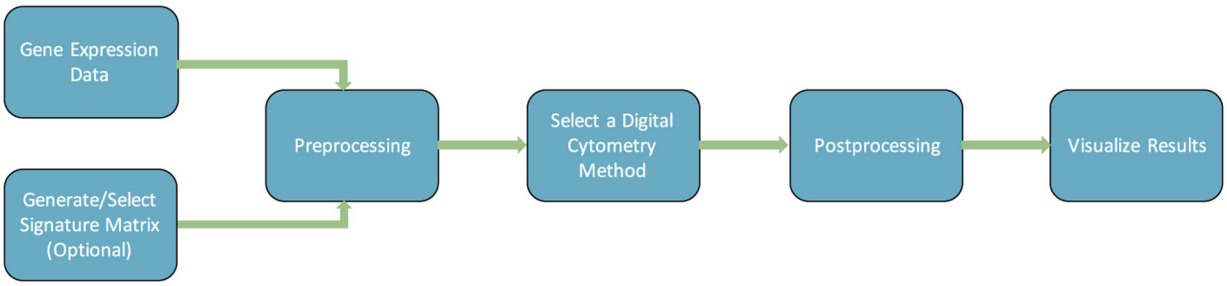

Fig. 1.
TumorDecon architecture.

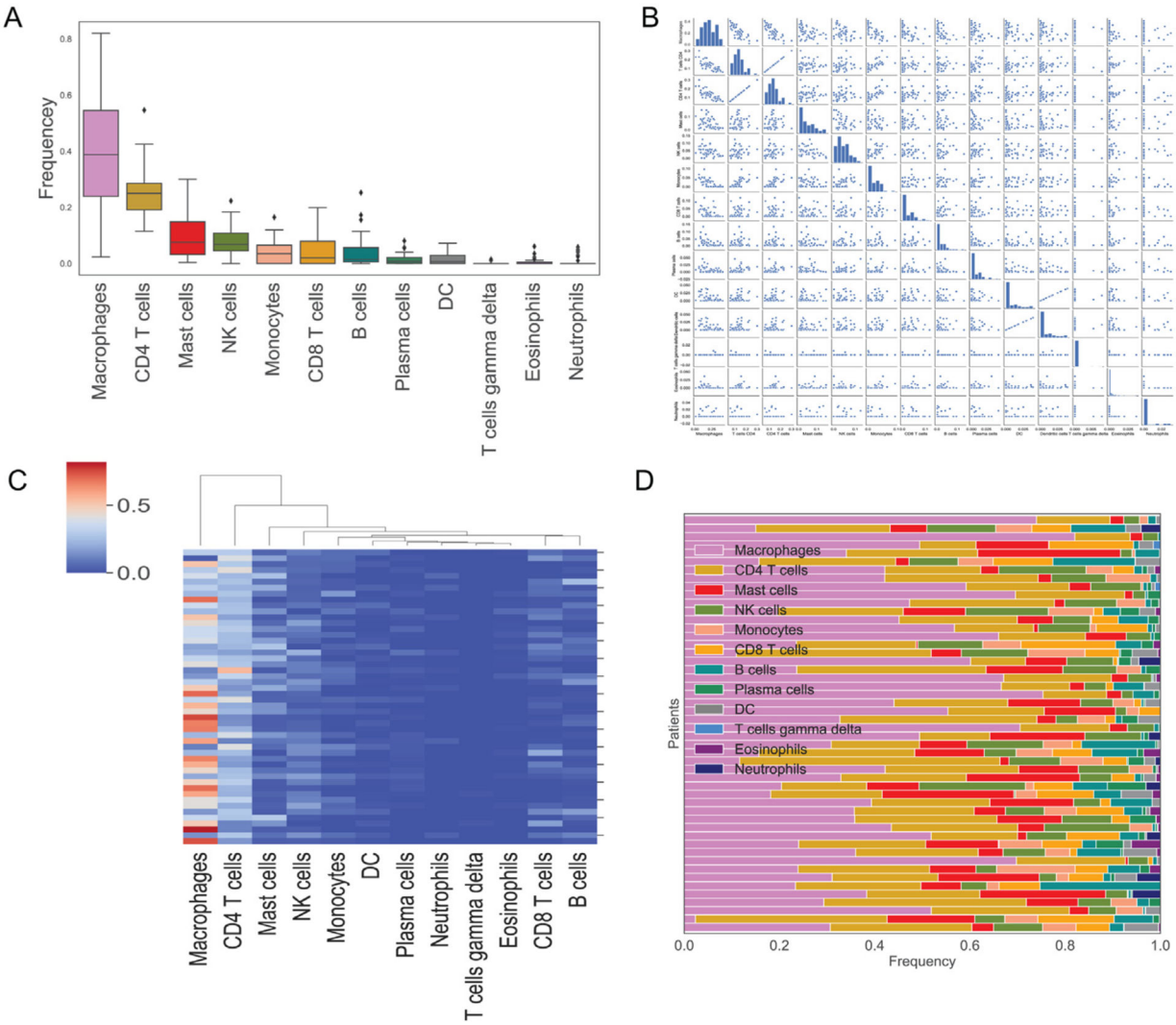

Fig. 2.

TumorDecon visualization examples using colon cancer data and CIBERSORT method. Output of TumorDecon includes (A) box plots, (B) pair plots, (C) cluster heat maps, and (D) bar charts of cell fractions in each tumor.

Table 1

A preview of a TCGA RNA Seq data.

Hugo_Symbol	Entrez_Gene_Id	TCGA-3L-AA1B-01	TCGA-4N-A93T-01	…	TCGA-AG-A032-01
A1BG	1	22.147	171.268	…	22.836
A1CF	29974	220.987	100.629	…	159.905
…	…	…	…	..	…
ZZZ3	26009	798.356	333.817	..	343.951