Community assessment of methods to deconvolve cellular composition from bulk gene expression

Brian S. White, Computational Oncology, Sage Bionetworks, Seattle, WA, USA
Aurélien de Reyniès, Centre de Recherche des Cordeliers, INSERM U1138, Université Paris Cité, Paris, France
Aaron M. Newman, Institute for Stem Cell Biology and Regenerative Medicine and Department of Biomedical Data Science, Stanford University, Stanford, CA, USA.
Joshua J. Waterfall, INSERM U830 and Translational Research Department, Institut Curie, PSL Research University, Paris, France.
Andrew Lamb, Computational Oncology, Sage Bionetworks, Seattle, WA, USA
Florent Petitprez, Programme Cartes d'Identité des Tumeurs, Ligue Nationale Contre le Cancer, Paris, France and MRC Centre for Reproductive Health, the Queen's Medical Research Institute, University of Edinburgh, United Kingdom
Alberto Valdeolivas, Heidelberg University, Faculty of Medicine, and Heidelberg University Hospital, Institute for Computational Biomedicine, Bioquant, Heidelberg, Germany
Yating Lin, Xiamen University, Xiamen, Fujian, China
Haojun Li, Xiamen University, Xiamen, Fujian, China
Xu Xiao, Xiamen University, Xiamen, Fujian, China
Shun Wang, Aginome Scientific, Xiamen, Fujian, China
Frank Zheng AmoyDx, Xiamen, Fujian, China
Wenxian Yang Aginome Scientific, Xiamen, Fujian, China
Rongshan Yu, Xiamen University, Xiamen, Fujian, China
Martin E Guerrero-Gimenez, Institute of Biochemistry and Biotechnology, School of Medicine, National University of Cuyo, Mendoza, Argentina
Carlos A Catania, Laboratory of Intelligent Systems (LABSIN), Engineering School, National University of Cuyo, Mendoza, Argentina
Benjamin J Lang, Department of Radiation Oncology, Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, USA
Sergii Domanskyi, Michigan State University, East Lansing, MI, USA
Thomas J Bertus, Michigan State University, East Lansing, MI, USA
Carlo Piermarocchi, Michigan State University, East Lansing, MI, USA
Gianni Monaco, BIOGEM Institute of Molecular Biology and Genetics, Ariano Irpino, AV, Italy
Francesca P Caruso, BIOGEM Institute of Molecular Biology and Genetics, Ariano Irpino, AV, Italy
Michele Ceccarelli, BIOGEM Institute of Molecular Biology and Genetics, Department of Electrical Engineering and Information Technology (DIETI), University of Naples "Federico II", Ariano Irpino, AV, Italy
Thomas Yu, Computational Oncology, Sage Bionetworks, Seattle, WA, USA
Xindi Guo, Computational Oncology, Sage Bionetworks, Seattle, WA, USA
John Coller, Stanford Functional Genomics Facility, Stanford University School of Medicine, Stanford, CA
Holden Maecker, Institute for Immunity, Transplantation, and Infection, Stanford University

School of Medicine, Stanford, CA
Caroline Duault, Institute for Immunity, Transplantation, and Infection, Stanford University School of Medicine, Stanford, CA
Vida Shokoohi, Stanford Functional Genomics Facility, Stanford University School of Medicine, Stanford, CA
Shailja Patel, Translational Applications Service Center, Stanford University School of Medicine, Stanford, CA
Joanna E Liliental, Translational Applications Service Center, Stanford University School of Medicine, Stanford, CA
Stockard Simon, Sage Bionetworks, Seattle, WA, USA
Tumor Deconvolution DREAM Challenge consortium
Julio Saez-Rodriguez, Heidelberg University, Faculty of Medicine, and Heidelberg University Hospital, Institute for Computational Biomedicine, Bioquant, Heidelberg, Germany
Laura M. Heiser, Department of Biomedical Engineering, Knight Cancer Institute, Oregon Health & Science University, Portland, OR, USA.
Justin Guinney, Computational Oncology, Sage Bionetworks, Seattle, WA, USA
Andrew J. Gentles, Departments of Medicine, and Biomedical Data Science, Stanford University School of Medicine, Stanford, CA, USA

Tumor Deconvolution DREAM Challenge consortium:
Aashi Jain, Department of computer science, Indraprastha Institute of Information Technology, Okhla Ph-3, New Delhi, India
Shreya Mishra, Department of computer science, Indraprastha Institute of Information Technology, Okhla Ph-3, New Delhi, India
Vibhor Kumar, Department of computer science, Indraprastha Institute of Information Technology, Okhla Ph-3, New Delhi, India
Jiajie Peng, School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
Lu Han, School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
Gonzalo H Otazu, Center for Biomedical Innovation, New York Institute of Technology, College of Osteopathic Medicine, New York, USA
Austin Meadows, Icahn School of Medicine at Mount Sinai, New York, USA
Patrick J Danaher, NanoString Technologies, Seattle, WA, USA
Maria K Jaakkola, Turku Bioscience Centre, University of Turku and Åbo Akademi University; Department of Mathematics and Statistics, University of Turku, Finland
Laura L Elo, Turku Bioscience Centre, University of Turku and Åbo Akademi University; Institute of Biomedicine, University of Turku, Finland
Julien Racle, Department of Oncology UNIL CHUV, Ludwig Institute for Cancer Research, University of Lausanne, Lausanne, Switzerland; Swiss Institute of Bioinformatics (SIB), Lausanne, Switzerland

David Gfeller, Department of Oncology UNIL CHUV, Ludwig Institute for Cancer Research, University of Lausanne, Lausanne, Switzerland; Swiss Institute of Bioinformatics (SIB), Lausanne, Switzerland
Dani Livne, The Mina & Everard Goodman Faculty of Life Sciences, Bar-Ilan University, Israel
Sol Efroni, The Mina & Everard Goodman Faculty of Life Sciences, Bar-Ilan University, Israel
Tom Snir, The Mina & Everard Goodman Faculty of Life Sciences, Bar-Ilan University, Israel
Oliver M Cast, Cancer Research UK, Cambridge Institute, University of Cambridge, UK
Martin L Miller, Cancer Research UK, Cambridge Institute, University of Cambridge, UK
Dominique-Laurent Couturier, Cancer Research UK, Cambridge Institute, University of Cambridge, UK
Wennan Chang, Electric Computer Engineering, Purdue University, West Lafayette, IN, US
Sha Cao, Department of Biostatistics, Indiana University, School of Medicine, Indianapolis, IN, US
Chi Zhang, Medical and Molecular Genetics, Indiana University School of Medicine, Indianapolis, IN, US
Dominik J Otto, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany; Institute for Clinical Immunology, Leipzig University, Johannisallee 30, 04103 Leipzig, Germany
Kristin Reiche, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany; Institute for Clinical Immunology, Leipzig University, Johannisallee 30, 04103 Leipzig, Germany
Christoph Kämpf, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany
Michael Rade, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany
Carolin Schimmelpfennig, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany
Markus Kreuz, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany
Alexander Scholz, Department of Diagnostics, Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany

Abstract

Deconvolution methods infer levels of immune and stromal infiltration from bulk expression of tumor samples. These methods allow projection of characteristics of the tumor microenvironment, known to affect patient outcome and therapeutic response, onto the millions of bulk transcriptional profiles in public databases, many focused on uniquely valuable and clinically-annotated cohorts. Despite the wide development of such methods, a standardized dataset with ground truth to evaluate their performance has been lacking. We generated and sequenced *in vitro* and *in silico* admixtures of tumor, immune, and stromal cells and used them as ground truth in a community-wide DREAM Challenge that provided an objective, unbiased assessment of six widely-used published deconvolution methods and of 22 new analytical approaches developed by international teams. Our results demonstrate that existing methods predict many cell types well, while team-contributed methods highlight the potential to resolve functional states of T cells that were either not covered by published reference signatures or estimated poorly by some published methods. Our assessment and the open-source implementations of top-performing methods will allow researchers to apply the deconvolution approach most appropriate to querying their cell type of interest. Further, our publicly-available admixed and purified expression profiles will be a valuable resource to those developing deconvolution methods, including in non-malignant settings involving immune cells.

Introduction

Tissues are comprised of multiple cell types that interact to confer diverse functions. Cells of the immune system are increasingly recognized for their critical function in both normal and diseased tissues, and many diseases have been linked to changes in the immune context of tissues, including cancer, Alzheimer's disease, arthritis and the recent SARS-CoV-2 virus, responsible for the COVID19 pandemic. In the field of oncology, the immune system has emerged as a critical factor in determining disease progression, patient survival and response to therapy. The tumor microenvironment (TME) also presents a number of actionable therapeutic targets.[1] Notably, immune checkpoint inhibitors, which are designed to re-potentiate cytotoxic T-cells to engage and kill malignant cells, have led to spectacular clinical outcomes for a subset of patients with previously dire prognoses. The precise reasons why some patients respond, but others do not, are poorly understood, indicating that a more detailed understanding of the TME and its cellular components is needed. A recent DREAM Challenge is aimed specifically at predicting response to anti PD-L1 therapy in lung cancer using demographic and gene expression data.[2]

Defining and characterizing the TME is the aim of numerous experimental and computational approaches.[3,4] Well-established techniques, such as flow cytometry and immunohistochemistry, can rapidly and accurately count cells of specific types from a tissue, but are limited by the number of markers (and therefore cell types) they can simultaneously assay. Single-cell proteomic technologies, including mass cytometry (CyTOF), can quantify ~100s of proteins in millions of cells at once, but require validated, high-quality antibodies for each marker of interest.[5] Single-cell RNA-seq (scRNA-seq) platforms are increasingly cost-effective and provide, in principle, an unbiased view of tissue content.[6] However, artifacts are introduced by tissue preparation steps including dissociation and other manipulations, leading to preferential loss of specific cell types, such as plasma cells and neutrophils, as well as perturbation of cell state, with attendant transcriptome and proteome changes.[7–11] Such methods also cannot currently be applied to archival tissues and instead require prospective sample collection. *In situ* molecular imaging platforms, such as cyclic immunofluorescence,[12] imaging mass cytometry,[13] CODEX,[14] and multiplex ion beam imaging,[15] can spatially resolve individual cells, but rely on predefined markers and appropriately prepared tissue. Newly emerging spatial transcriptomics technologies are able to measure expression for thousands of genes, are applicable to formalin fixed paraffin embedded as well as fresh frozen tissue, and are rapidly advancing to single-cell resolution.[16–18]

Beyond these experimental and analytical challenges, the number of clinically annotated single cell-based samples is dwarfed by those derived from bulk genomic and transcriptomic data. Consequently, to leverage the large databases of clinically-annotated samples with bulk data such as The Cancer Genome Atlas (TCGA) and the Gene Expression Omnibus (GEO), methods for computational "deconvolution" of bulk tissue transcriptomic profiles have been developed to estimate the relative amounts of specific cell types in an admixture leveraging both RNA-seq and legacy microarray platforms.[19–24]

Computational deconvolution approaches applied to bulk transcriptional profiles circumvent limitations of experimental approaches that could skew estimates of cell type frequencies. In particular they can be applied to expression profiles derived from intact fresh frozen *or* archival tissues, and do not perturb cell state or relative cell population abundances. Unsupervised (i.e., "complete" or "reference-free" deconvolution) methods dissect cellular composition without *a priori* information about the cell types such as marker genes or expression profiles.[25] Supervised approaches can be classified as "partial" deconvolution methods, which estimate the fractions of cell types in a mixture (more strictly, their RNA contribution to the total RNA of the mixture), or enrichment-/marker-based methods, which assign a per-cell-type score that can be used to compare the relative prevalence of a specific cell type across samples, but cannot compare different cell types. Enrichment-based methods can sensitively distinguish between "coarse-grained" cell types (e.g., B versus T cells), but often have low specificity in discriminating between "fine-grained" cell types (e.g. sub-populations of T cells such as central memory CD4+ T cells, effector CD8+ T cells, or T-regs).[26] Partial deconvolution methods are typically more specific than enrichment-based methods, but may be less sensitive. The tradeoff between these properties is important when considering their application to particular questions.

Several benchmarking efforts have used *in silico* simulation to evaluate factors that impact the accuracy of published deconvolution methods, including expression normalization,[27] technical noise,[28] and the specificity of marker genes.[26] Sturm and colleagues concluded one of these assessments with a call to refine population signatures.[26] In order to address some of these issues, we designed and ran a community-wide DREAM Challenge to encourage development of *novel* methods for deconvolving cellular composition from bulk gene expression and to objectively assess these methods alongside state-of-the-art published methods. We chose to focus on supervised methods, both partial deconvolution and enrichment-based, as these are most widely used in the cancer community. We use the overarching term "deconvolution" to refer to both. In contrast to prior benchmarking efforts, we simulated tumor expression profiles through *in vitro* mixing of RNA from different cell types in proportions intended to be representative of real solid tumors, then performed RNA-seq on the admixtures. Additionally, we generated *in silico* admixtures from the expression profiles of purified cell types. In both cases, the known mixing proportions were used as ground truth in assessing method predictions from the resulting bulk "tissue" expression using correlation as a metric. Simulating (*in vitro* or *in silico*) admixtures allowed us to define controlled ground truth in isolation from experimental biases due to technical and biological variability. Our findings, based on participation of 22 international teams, revealed that most methods were able to predict major or "coarse-grained" populations well, alongside pervasive difficulty in predicting a subset of "fine-grained" sub-populations accurately. The expression profiles of purified populations and *in vitro* admixtures generated for this Challenge are available as a resource for developing and training deconvolution methods in contexts where quantifying immune and stromal cells is of interest.

Results

Novel purified and admixed expression profiles enable unbiased assessment of deconvolution methods

Immune cell infiltration has prognostic significance across multiple levels of immune cell specialization and polarization. For example, CD8+ T cells, broadly encompassing memory and naïve compartments, have positive prognostic value in many cancer types, whereas regulatory T cells, a specific subset of CD4+ T cells, generally have negative prognostic associations.[29] To assess deconvolution across these levels, we divided the Tumor Deconvolution DREAM Challenge into "coarse-grained" and "fine-grained" sub-Challenges. In the coarse-grained sub-Challenge, participants predicted levels of eight major immune and stromal cell populations: B cells, CD4+ and CD8+ T cells, NK cells, neutrophils, cells of monocytic lineage (monocytes, macrophages, and dendritic cells), endothelial cells, and fibroblasts, whereas in the fine-grained sub-Challenge they further dissected these populations into 14 sub-populations, e.g., memory, naïve, and regulatory CD4+ T cells (**Fig. 1A**).

To facilitate benchmarking and create a ground truth dataset, we generated *in vitro* and *in silico* expression profiles of cell populations admixed at predefined ratios. We isolated immune cells from healthy donors and obtained stromal, endothelial, and cancer cells from cell lines (**Tables S1** and **S2**; Methods). Cell type-specific marker expression was confirmed on the purified cells through RNA sequencing (**Fig. S1**). To robustly test algorithm performance across diverse conditions, we defined mixing proportions and generated admixtures from them, grouped into one of six datasets according to whether they: (1) included breast or colon cancer cells; (2) had proportions that were unconstrained or constrained by biologically-reasonable expectations ("biological" distribution); or (3) were created *in silico* or *in vitro* (Methods; **Tables S3-S8**). This resulted in a total of 96 *in vitro* admixtures and 140 *in silico* admixtures, with at least 18 admixtures in each dataset to ensure adequate sample size (**Table S9**). We generated *in silico* admixtures as a linear combination of the mixing proportions and the purified expression profiles and *in vitro* admixtures by extracting RNA from the purified cells, mixing them at the specified proportions, and sequencing (**Fig. 1B**).

We provided participants with GEO microarray and RNA-seq sample identifiers curated as to inferred purified cell type for use in training (**Tables S10** and **S11**). Significantly, these were publicly available samples that did not include those used in generating our own admixtures. Methods were evaluated against the admixtures by correlating the predictions of cell type proportions with the predefined ("ground truth") proportions, independently for each cell type. Since the aim of the Challenge concerns the microenvironment populations, we only assessed participants' predictions based on inference of immune and stromal cell content, with the admixed cancer cells effectively treated as contaminating noise. In order to rank methods, we defined an aggregate score that averaged correlations across cell types and validation datasets (Methods). Methods were first assessed using a "primary" Pearson correlation-based score. Statistical ties were resolved relative to the top-performing method (as determined by a Bayes factor; Methods) by a "secondary" Spearman correlation-based score. To account for sampling variability, we reported these Pearson-based (*r*) and Spearman-based (ρ) scores as their means across bootstraps (Methods).

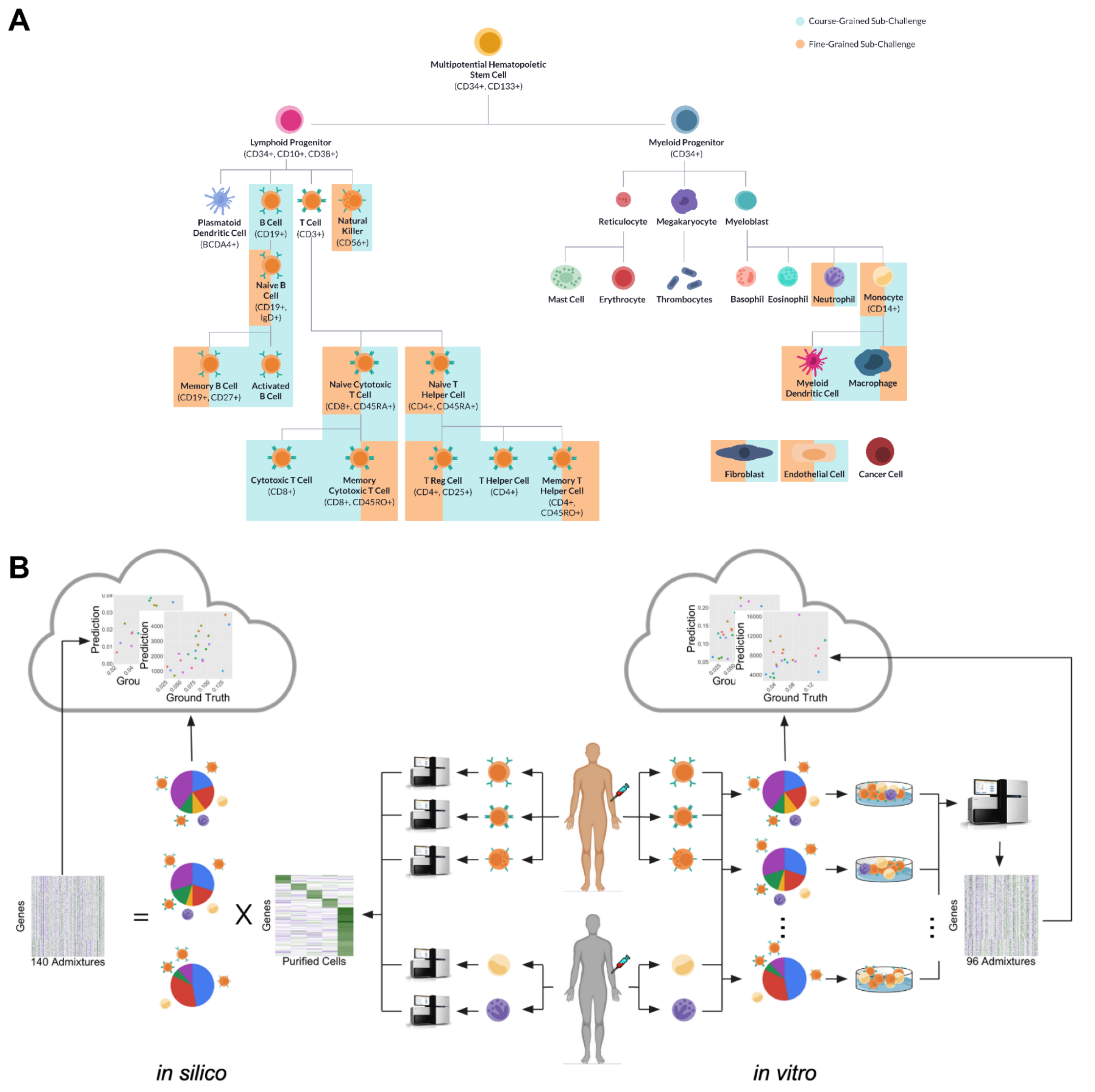


Fig. 1: Generation of *in silico* and *in vitro* admixtures of immune, stromal, and cancer cells and their use as validation data for a DREAM Challenge**.** (A) Cell populations predicted within Challenge. Populations used in coarse-grained sub-Challenge shaded in blue, in fine-grained sub-Challenge shaded in orange, and in both sub-Challenges shaded in both orange and

blue. Cell types aggregated together in coarse-grained sub-Challenge connected with blue shading (e.g., monocytes, myeloid dendritic cells, and macrophages were classified as monocytic lineage). Immune populations are depicted within the haematopoietic hierarchy. (B) Admixture generation and use for validation. (Left) Purified cell populations (from vendors, volunteers, and cell lines) were subjected to RNA-seq to define population-specific signatures. *In silico* admixtures were then defined as a linear combination of these signatures and specified ratios (unconstrained or constrained according to biologically reasonable expectation). (Right) *In vitro* admixtures were created by mixing mRNA from purified cell populations in specified ratios (unconstrained or biologically reasonable) and then subjected to RNA-seq. Deconvolution methods executed in the cloud against *in silico* and *in vitro* admixtures yielded predictions that were then compared to the input ratios using correlation.

As in previous DREAM Challenges,[30] participants submitted their methods as Docker containers, which were executed in the cloud against sequestered validation data. This "model-to-data"[31] approach ensured that data were not leaked to participants, prevented model over-fitting, and allowed complete reproducibility of the Challenge. Twenty-two teams contributed 39 submissions (i.e., unique methods) to the coarse-grained sub-Challenge, while 16 teams contributed 48 submissions to the fine-grained sub-Challenge. Additionally, we applied six widely-used published tools (CIBERSORT,[19] CIBERSORTx,[32] EPIC,[20] MCP-counter,[21] quanTIseq,[23] and xCell[24]) as baseline comparator methods (Methods). Comparator methods were run by Sage Bionetworks independent of method developers to ensure unbiased assessments, using default parameters and published reference signatures. No optimization was applied to tune their performance.

Diverse algorithmic approaches deconvolve immune populations well in aggregate across samples

We tested the performance of the six "comparator" methods, as well as Challenge participant methods. The median Pearson correlation-based score across participant and comparator methods was 0.75 [interquartile range (IQR): 0.43 - 0.81; **Fig. 2A**] for the coarse-grained sub-Challenge and 0.61 (IQR: 0.53 - 0.62; **Fig. 2B**) for the fine-grained sub-Challenge. Methods differed in their output. Some produce an arbitrary score proportional to the presence of a cell type, which can be used to compare the same cell type across samples, but not across cell types. Others generate normalized scores, non-negative proportions summing to one, or non-negative fractions that need not sum to one, which can be compared both across samples and cell types.

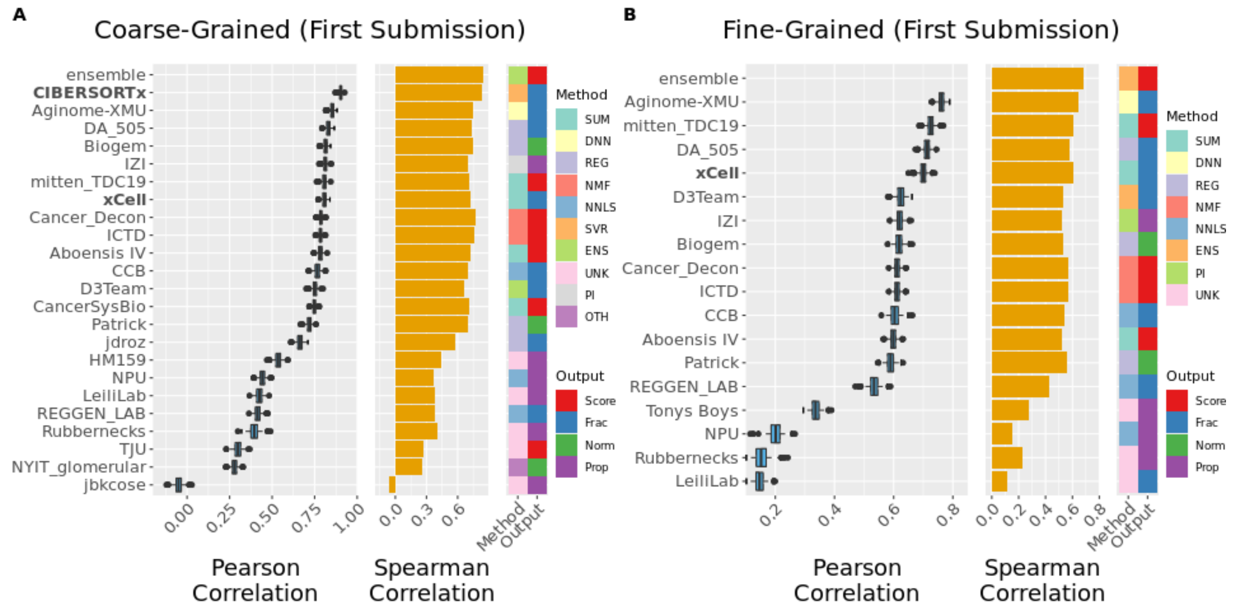


Fig. 2: Aggregate cross-sample performance of participant and comparator deconvolution methods. Aggregate score (primary metric: Pearson correlation; secondary metric: Spearman correlation) of participant (first submission only) and comparator methods in (A) coarse- and (B) fine-grained sub-Challenges over (*n*=1,000) bootstraps (Methods). Reported Spearman-based score is median over bootstraps. Comparator methods (bold) are shown only if their published reference signatures include all cell types in each respective sub-Challenge: CIBERSORTx (coarse-grained only) and xCell. DNN: deep neural network; ENS: ensemble; NMF: non-negative matrix factorization; NNLS: non-negative least squares; OTH: other; PI: probabilistic inference; REG: other regression; SUM: summary; SVR: support vector regression; UNK: unknown/unspecified; Frac: unnormalized fractions that need not sum to one; Norm: normalized scores (comparable across cell types and samples); Prop: proportions that sum to one

Across both comparator and participant approaches, CIBERSORTx was the top-performing method in the coarse-grained sub-Challenge according to both metrics (*r* = 0.90; ρ = 0.83; **Fig. 2A**). The next highest-scoring methods and the top-performing *participant* methods were Aginome-XMU (*r* = 0.85; https://doi.org/10.1016/j.patter.2022.100440) and Cancer_Decon (ρ = 0.79) according to the primary Pearson-based and secondary Spearman-based scores, respectively. Aginome-XMU was the top-performing method (participant or comparator) in the fine-grained sub-Challenge (*r* = 0.76; ρ = 0.64; **Fig. 2B**). Methods whose published reference signatures do not include all cell types in a sub-Challenge (e.g., five of six comparator methods, CIBERSORT, CIBERSORTx, EPIC, MCP-counter, and quanTIseq, in the fine-grained sub-Challenge) were not considered in that sub-Challenge's aggregate ranking. Additionally, there was broad consistency in method ranking across the two sub-Challenges, with the three top-ranked participant methods in the coarse-grained sub-Challenge (Aginome-XMU, DA_505, and Biogem) amongst the top seven evaluable methods in the fine-grained sub-Challenge.

Conversely, the three top-ranked evaluable teams in the fine-grained sub-Challenge (Aginome-XMU, mitten_TDC19, and DA_505) were amongst the top seven in the coarse-grained sub-Challenge.

We compared the performance of deconvolution methods to an ensemble combination of their outputs, since the latter has often been shown to outperform individual algorithms and thus represents a potential upper bound. As the scale of predicted values vary according to the type of method output (scores, normalized scores, or fractions/proportions), they cannot be combined by simple averaging. Instead, we defined an ensemble prediction as the consensus *rank* across individual methods (Methods). Though this is only evaluable for Spearman correlation, we found that overall it outperformed any individual method in both the coarse- (ρ = 0.84; **Fig. 2A**) and fine-grained (ρ = 0.67; **Fig. 2B**) sub-Challenges. However, there was only modest improvement of the ensemble method relative to the top-scoring individual methods by Spearman correlation in the coarse- (CIBERSORTx ρ = 0.83) and fine-grained (Aginome-XMU ρ = 0.64) sub-Challenges suggesting that individual methods are not leveraging independent or orthogonal signals in the data despite their diverse approaches.

Several core algorithmic approaches were common across submissions, including those based on non-negative least squares (NNLS; 6 in the coarse-grained and 6 in the fine-grained sub-Challenge, respectively; **Fig. 2A, B**) and summarization (SUM; 5 and 4). Nevertheless, there was wide methodological diversity amongst top performers. CIBERSORTx uses ν-SVR to simultaneously solve for all fractional abundances relating admixed and purified expression profiles using a signature matrix of ~525 differentially expressed genes spanning 22 immune cells types (LM22).[19,32] Aginome-XMU, published subsequent to the Challenge[33], utilizes a neural network composed of an input layer, five fully-connected hidden layers, and an output layer (Supplemental Methods; https://github.com/xmuyulab/DCTD_Team_Aginome-XMU; **Table S12**). The network effectively applies feature selection automatically and was trained here using synthetic admixtures. DA_505 applies a rank-based normalization, selects features by applying random forests to synthetic admixtures, and ultimately applies regression to predict abundance of each cell type *independently* (Supplemental Methods; https://github.com/martinguerrero89/Dream_Deconv_Challenge_Team_DA505; **Fig. S2**; **Table S13**). mitten_TDC19 calculates a summarization score as the sum of the expression of selected markers, with the cell type-specific markers first nominated from expression profiles in purified bulk data or identified[34,35] from single-cell data expression profiles and then prioritized according to their correlation with that cell type's proportion over synthetic admixtures (Supplemental Methods; https://github.com/sdomanskyi/mitten_TDC19; **Table S14**). Finally, Biogem, based on a previously published method,[36] uses robust linear modeling to perform deconvolution and differential expression-based feature selection to define the purified expression profiles (Supplemental Methods; https://github.com/giannimonaco/DREAMChallenge_Deconvolution; **Fig. S3**; **Table S15**). Hence, despite their algorithmic differences, three of the top-performing methods were trained using synthetic admixtures, generated *in silico* from publicly available purified expression profiles (**Table S16**). Importantly, the purified profiles that we created to generate Challenge admixtures were *not* made available to participants.

Method performance improved for most teams over three allowed submissions (**Fig. S4**), as they were permitted to revise their method with each submission. Since we provided teams with both aggregate and per-cell type scores following each submission, we can not exclude the possibility that these were used to tune methods between submissions. As this could result in over-fitting, we focused our analyses on the first submission, unless otherwise explicitly stated (**Fig. S5)**.

Deconvolution performance differs by cell type

In addition to a ranked average performance across cell types, we assessed methods in their ability to predict individual cell type levels within an admixture (**Fig. 3** and **Figs S6-S15**).
All major lineage cell types could be predicted robustly by at least one method (max row of **Fig. 3A** and **Figs S8A**, **S12A**, and **S15A**). For example, though CD4+ T cell levels were most difficult to predict on average, even these could be predicted with a Pearson correlation of 0.86 by CIBERSORTx and xCell. Neutrophil levels were predicted best on average, with 18 of 28 methods having a Pearson correlation of at least 0.90.

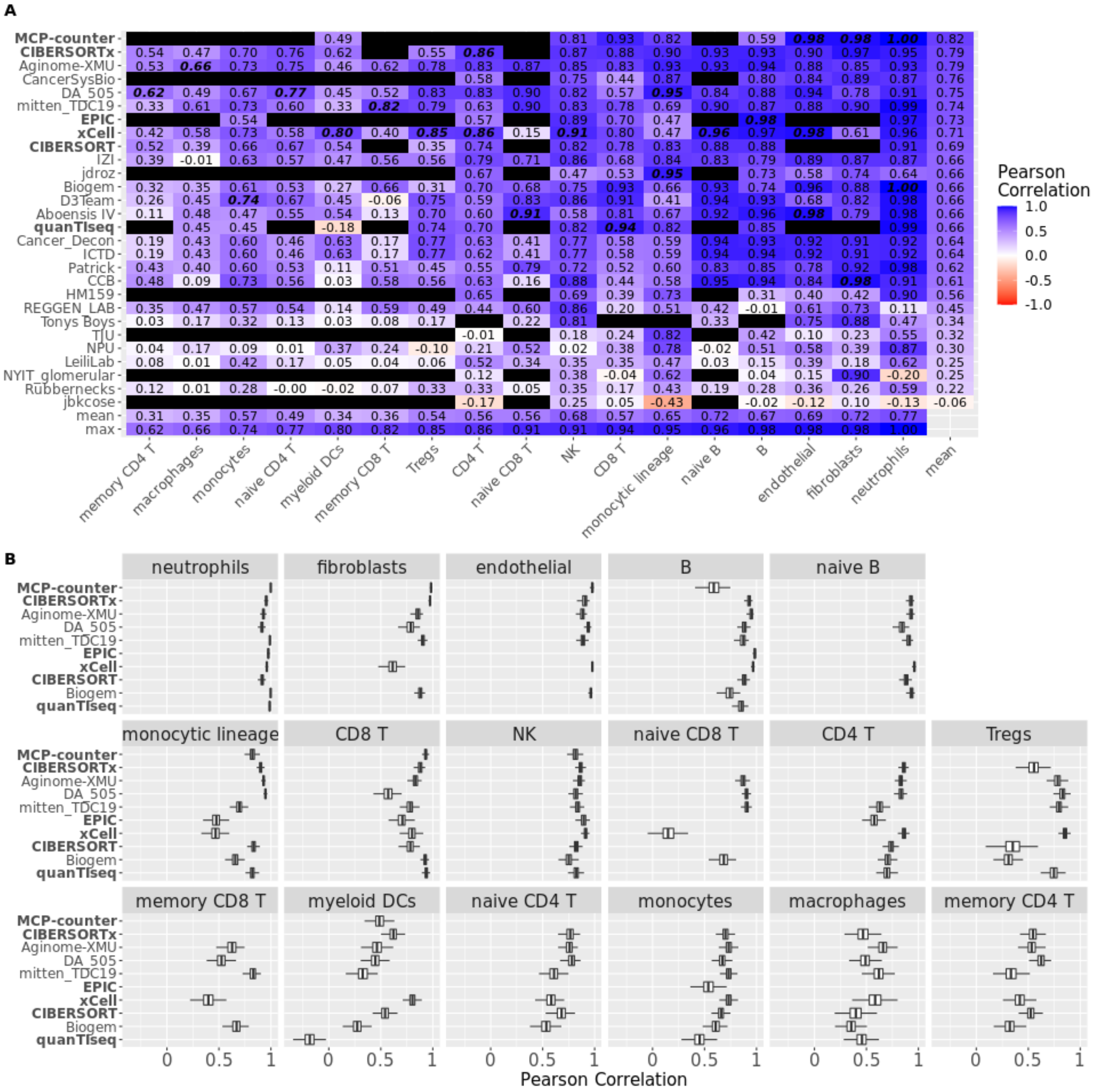


Fig. 3: Per-cell type performance of participant and comparator deconvolution methods. (A) Pearson correlation of method (left axis) prediction versus known proportion from admixture for each cell type (bottom axis). Pearson correlation is first averaged over validation dataset and then (*n*=1,000) over bootstraps (Methods) and subsequently averaged over coarse- and fine-grained sub-Challenges for cell types occurring in both. Black entry indicates cell type not predicted by corresponding method. Bottom two rows ("mean" and "max") are the mean and maximum correlation, respectively, for corresponding cell type across methods. Rightmost column ("mean") is mean correlation for corresponding method across predicted cell types. Highest correlations for each cell type highlighted in bold italics. (B) Performance (Pearson correlation; x axis) of comparator baseline methods and participant methods ranking within the top three in either or both sub-Challenges (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods), computed as average over validation datasets and subsequently over coarse- and fine-grained sub-

Challenges for cell types occurring in both. Blank row indicates cell type not reported by the corresponding method. Comparator methods in bold.

Baseline comparator methods, other than xCell, were not trained to predict all the fine-grained immune subtypes used in the Challenge. For example, none of quanTIseq, MCP-counter, or EPIC differentiate between memory and naïve CD4+ T cells; and only xCell differentiates between memory and naïve CD8+ T cells, though with low accuracy for both (**Fig. 3**; *r* <= 0.40). Participant submitted models showed potential at predicting these poorly covered cell types. For example, mitten_TDC19 dramatically improved upon comparator performance in predicting both naïve CD8+ T cells (*r* = 0.90 vs xCell *r* = 0.15) and memory CD8+ T cells (*r* = 0.82 vs xCell *r* = 0.40), with Aboensis IV outperforming both for naïve CD8+ T cells (*r* = 0.91). Further, Aginome-XMU performance on macrophages (*r* = 0.66 vs xCell *r* = 0.58) and DA_505 performance on memory CD4+ T cells (*r* = 0.62 vs CIBERSORTx *r* = 0.54) improved upon their respective best-performing comparator methods. In all other cases, participant methods showed at best only modest improvement (change in *r* < 0.05) relative to comparator methods. Notwithstanding these advances, the seven most difficult populations to predict were functional subsets of CD4+ and CD8+ T cells and sub-populations of the monocytic lineage (**Fig. 3**).

The methods that performed well in aggregate also performed well based on individual cell types, though none dominated across all populations. Nominally, 10 methods were the top performers across one or more of the 17 individual cell populations. For most of these populations, multiple methods achieved similar performance to the top-ranked one (**Fig. 3B** and **Figs S6**, **S7**, **S9**, **S11**, **S13**, **S14**). Exceptions in which the top method outperformed the nearest competitor (comparator or participant) by a large margin (change in *r* > 0.05) were: xCell on myeloid dendritic cells (*r* = 0.80 vs Cancer_Decon *r* = 0.63), mitten_TDC19 on memory CD8+ T cells (*r* = 0.82 vs Biogem *r* = 0.66), and DA_505 on memory CD4+ T cells (*r* = 0.62 vs CIBERSORTX *r* = 0.54). In all other cases, the top-performing method showed at best a marginal improvement (change in *r* < 0.05) relative to the next best-performing method (comparator or participant).

Intra-sample, inter-cell type deconvolution performance is worse than inter-sample, intra-cell type performance

We next assessed prediction performance across cell types *within* samples for those methods that produced normalized scores, proportions, and fractions (**Fig. 4**). To do so, we computed a correlation (Pearson and Spearman) and the root-mean-square error (RMSE) across cell types within a sample and then reported the median of these respective values across samples. Top-performing methods varied across sub-Challenge and metric (**Tables 1** and **S11**), though several methods performed well in both the above aggregate inter-sample/intra-cell type comparison and in this intra-sample/inter-cell type comparison: CIBERSORTx was amongst the top performers (i.e., having the highest score or showing no statistical difference from the method with the highest score) across all metrics in the coarse-grained sub-Challenge; DA_505 was a top performer based on RMSE and Spearman correlation in the coarse-grained sub-

Challenge; Biogem was a top performer based on Spearman correlation in the coarse-grained sub-Challenge; and Aginome-XMU was a top performer based on RMSE and and Pearson correlation in the fine-grained sub-Challenge. Additionally, several other methods were amongst the top performers across one or more metrics in one or both sub-Challenges, including: (1) xCell, which computes a score for each cell type by applying single sample gene set enrichment analysis (ssGEA) to a set of marker genes, transforms the scores to proportions using a calibration function, and finally compensates for spillover between similar cell types; (2) CCB, which extends the published NNLS-based EPIC method by applying ssGSEA to those populations not treated by EPIC and by relating those ssGSEA scores to proportions via a calibration function; and (3) Patrick, which uses excludes tumor-associated genes from the immune and stromal reference signatures and then performs constrained optimization in logarithmic space.

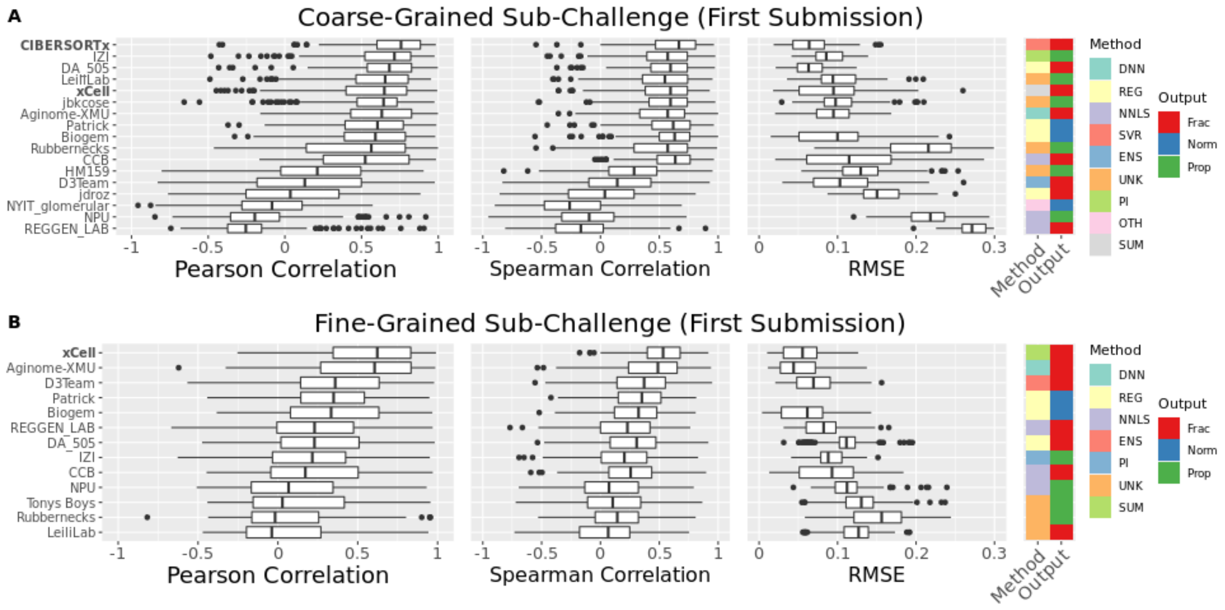


Fig. 4: Aggregate cross-cell type performance of participant and comparator deconvolution methods. Performance [Pearson correlation, Spearman correlation, and root mean square error (RMSE)] of methods capable of intra-sample, cross-cell type comparison to ground truth proportions in (A) coarse- and (B) fine-grained sub-Challenges. Distribution over *n*=166 samples (methods ordered by median Pearson correlation in respective sub-Challenge). Comparator methods in bold. DNN: deep neural network; ENS: ensemble; NNLS: non-negative least squares; OTH: other; PI: probabilistic inference; REG: other regression; SUM: summary; SVR: support vector regression; UNK: unknown/unspecified; Frac: fraction; Norm: normalized score (comparable across cell types and samples); Prop: proportion.

sub-Challenge	Metric	Top Performers (median)
Coarse	Pearson	CIBERSORTx (0.76)
	Spearman	CIBERSORTx (0.67), CCB (0.64), Biogem (0.63), Patrick (0.62), DA_505 (0.60),
	RMSE	DA_505 (0.06), CIBERSORTx (0.06)
Fine	Pearson	xCell (0.62), Aginome-XMU (0.61)
	Spearman	xCell (0.53)
	RMSE	Aginome-XMU (0.04)

Table 1: Top-ranked methods based on intra-sample performance. Median value (across samples) for corresponding metric provided in parentheses. Multiple top performing methods listed when they show no statistical evidence of difference (*p* > 0.05) from the method with largest median score. Several comparator methods were excluded from the fine-grained challenge since their published reference profiles did not include all of the evaluated cell types.

Deconvolution specificity is lower for T cells than for other cell types

Methods sometimes attribute signal from one cell type to a different cell type. This could particularly be the case for highly similar cell types such as sub-populations of CD4 T cells. To assess specificity, we quantified the "spillover" between cell types as a method's prediction for a particular cell type X within a sample purified for cell type $Y \neq X$ (**Fig. 5A** and **Fig. S16**). Based on median spillover, methods had greatest specificity for neutrophils. Expectedly, methods had greater specificity for the coarse- relative to the fine-grained populations (**Fig. 5B**): the second largest increase in median spillover separates a group enriched in major cell types [neutrophils, NK cells, naïve and "parental" (naïve and memory) B cells, endothelial cells, monocytes/monocytic lineage cells, and fibroblasts] from a group enriched in minor cell types (macrophages, memory/naïve/regulatory/parental CD4+ T cells, memory/naïve/parental CD8+ T cells, and myeloid dendritic cells). The single largest increase in median spillover separates memory CD4+ T cells, as the population for which methods have the worst specificity, from the remaining cell types. Across cell types, CCB had the lowest (median) spillover in both the coarse- (**Fig. 5C**) and fine-grained (**Fig. 5D**) sub-Challenges. In both cases, it was followed by

Aboensis IV, a summarization-based approach that defined robust marker genes within a cell type mutually correlated with one another. The top-performing methods (Aginome-XMU, Biogem, CIBERSORTx, DA_505, and mitten_TDC19) rank within the top half of methods in both sub-Challenges.

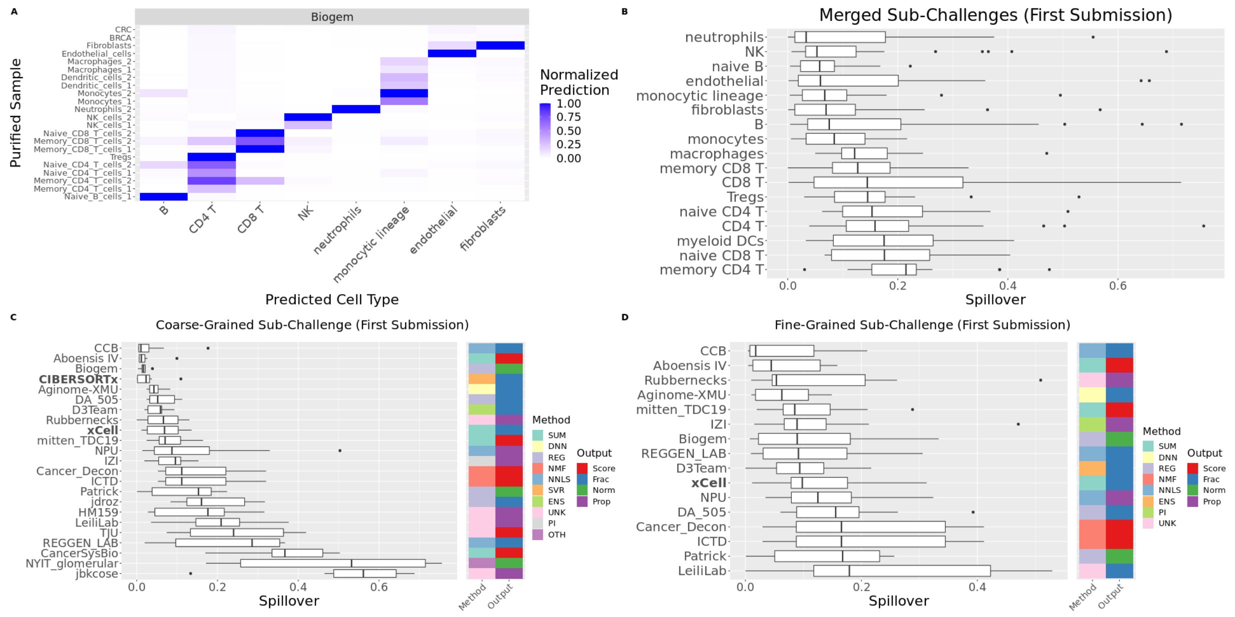


Fig. 5: Specificity of participant and comparator deconvolution methods. (A) Normalized prediction of cell type indicated on x axis in purified sample indicated on y axis. (B) Distribution over methods of spillover *into* cell type *c* indicated on y axis (averaged first over samples purified for any *other* cell type, then over sub-Challenges; Methods). (C, D) Distribution of spillover over cell types *c* for each method in (C) coarse- and (D) fine-grained sub-Challenges. Comparator methods in bold.

Deconvolution sensitivity is lower for CD4+ T cells than for other cell types

In real tumors, the representation of different cell types can range from only a fraction of a percent to a large proportion of the tissue. The limit of detection by deconvolution is likely to vary from cell type to cell type dependent on the uniqueness and strength of their transcriptional signal. We assessed deconvolution sensitivity using *in silico* spike-in experiments (Methods). We spiked each cell type at a given frequency (ranging from 0% to 40%) into an unconstrained admixture of all other cell types. We then determined the minimum frequency at *and* above which that cell type could be distinguished from the baseline (0% spike in; **Fig. 6A** and **Fig. S17**). The lowest limit of detection for any cell type was <0.2% (CIBERSORTx and Patrick for neutrophils and MCP-counter for CD8+ T cells), similar to that observed in prior microarray studies.[21] Seven methods showed similar mean limits of detection (3-4%) within the coarse-grained sub-Challenge (**Fig. 6A**), including top-ranked methods Biogem, CIBERSORTx, and Aginome-XMU. Neutrophils, fibroblasts, (naïve) B, and CD8+ T cells were sensitively identified by at least one method in both the coarse- (**Fig. 6B**) and fine-grained (**Fig. 6C**) sub-Challenges. All methods had low sensitivity in detecting CD4+ T cells (and their naïve and memory orientations) and macrophages, though several methods performed considerably better than others for both (CD4+ T cells: Aginome-XMU = 6%; D3Team = 7%; others >= 10% and macrophages: DA_505 = 8%; mitten_TDC19 = 8%; others >= 12%).

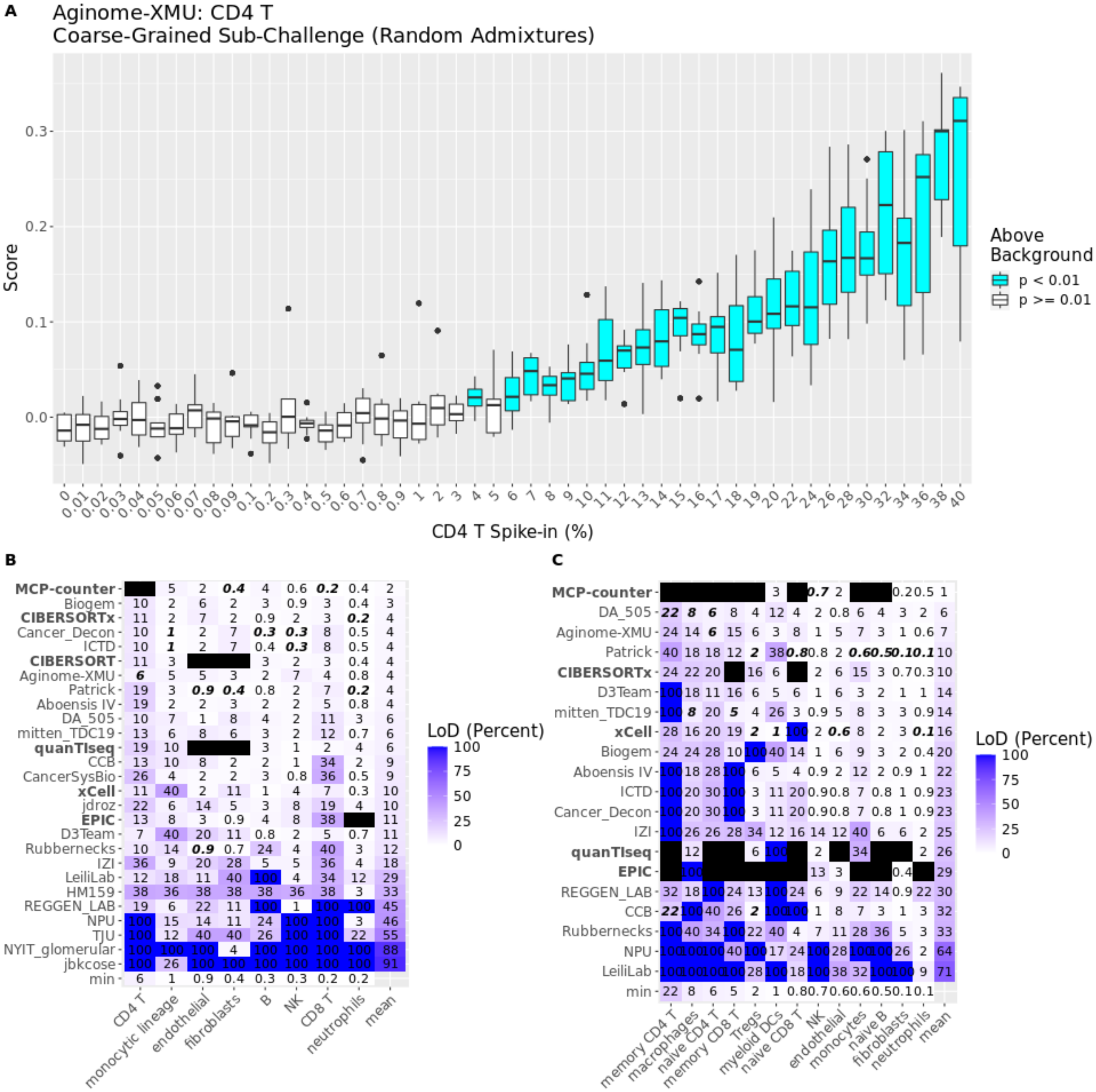


Fig. 6: Sensitivity of participant and comparator deconvolution methods. (A) Aginome-XMU predictions for CD4+ T cells (y axis) for unconstrained admixtures including the level of CD4+

T cells indicated (x axis). Limit of detection (LoD) is the least frequency at and above which all admixtures are statistically distinct from the baseline admixture (0% spike in), which is 6% in this case. (B, C) Limits of detection for indicated methods (rows) and cell types (columns) in the (B) coarse- and (C) fine-grained sub-Challenges. Best/lowest LoD for each cell type highlighted in bold italics. Comparator methods in bold.

Discussion

Existing (and growing) repositories of bulk gene expression data describe large cohorts of patients through rich annotations, making them invaluable in addressing questions across biological domains. Computational deconvolution methods for “unmixing” gene expression profiles from such bulk samples provide additional information – an estimation of cell type composition – shown to correlate with cancer phenotypes. However, benchmarking such methods has been hampered by the absence of “ground truth” data.

Here, we developed a novel resource and DREAM Challenge evaluation framework for a community assessment of deconvolution methods. We profiled major immune and stromal populations likely to be found in tumors using RNA-seq and derived both *in silico* and experimentally-generated *in vitro* admixtures against which to evaluate method performance. We objectively benchmarked six published and 22 novel methods and found that they could predict most major lineages well, with CIBERSORTx and Aginome-XMU narrowly emerging as leaders. As expected, prediction performance was lower for finer-grained dissections, particularly of the T cell compartment and of the monocytic lineage. Several novel methods showed potential performance gains in predicting memory CD4+ T cells and macrophages and in distinguishing between memory and naïve CD8+ T cells, which only one published method was trained to do. However, because comparator approaches were applied with their published reference signatures and default settings, we can not conclude whether performance gains were attributable to differences in algorithmic approach, reference signatures, or some combination. Overall, among methods formally evaluated on both coarse- and fine-grained sub-Challenges, Aginome-XMU, DA_505, and mitten_TDC19 had the best aggregate performance. That said, most comparator methods could not be included in the latter assessment since their published reference profiles did not include all the cell types required for scoring. Among such methods, CIBERSORTx was run on all cell types except naïve and memory CD8 T cells in the fine-grained sub-Challenge and exhibited comparable performance to other leading methods. MCP-counter could only predict half the specified cell types, but achieved high performance aggregated across those. Notably, no single method performed best on all cell types, nor did we observe that a particular high-level algorithmic approach dominated. This suggests that the “best” method may be problem specific and could be tailored to cell types of interest in a particular context. Our results provide the community cell type-level method performance to assist in making that decision, as well as novel datasets for benchmarking new approaches.

Prediction performance was lowest for memory and naïve CD4+ T cells, macrophages, and monocytes, despite potential strides made by novel Challenge methods for these sub-populations. We can partially diagnose potential causes of this reduced performance by leveraging the sequenced purified populations to assess specificity, i.e., "spillover" from the purified population to others, and sensitivity, assessed as the limit of detection of *in silico* spike ins. We observed that memory CD4+ T cells had poor specificity *and* sensitivity – ranking last in both metrics, with a median spillover across methods of 21%. Their best-case limit of detection was 22%, at which level they could be predicted above background by the *most* sensitive method. In contrast, other infiltrating immune cells had spillovers as low as ~5% for neutrophils and NK cells, and could be detected down to a threshold of ~0.2% (neutrophils for CIBERSORTx and Patrick and CD8+ T cells for MCP-counter). Naïve and parental CD4+ T cells also had poor sensitivity, with best-case limits of detection of 6%; and poor specificity, with spillovers of 15% and 16%, respectively. On one hand, poor prediction performance of macrophages may similarly be attributable to both low sensitivity, with a best-case limit of detection of 8%; and specificity indicated by 12% spillover. On the other hand, prediction performance of monocytes seems most likely due to low sensitivity (8% spillover) given the high specificity in detecting them, where the best-case limit of detection was ~1%. These difficulties in accurately predicting levels of CD4+ T cells and macrophages have important implications for tumor immunology given their importance as therapeutic targets.[37]

Our assessment has several limitations, including: (1) its focus on mRNA expression; (2) its use of immune cells from *healthy* donor PBMCs to simulate the tumor microenvironment; and (3) its admixing of cells (*in vitro* or *in silico*) in ratios that imperfectly represent *in vivo* distributions. With respect to the first of these, DNA methylation profiles have also proven to be a valuable source of information for tumor content deconvolution and it is possible to integrate multiple modalities to improve accuracy.[38] Nevertheless, the generated data here offer two principal advantages: they allow the objective definition of ground truth used in our evaluation and they can now be flexibly applied by others. In particular, the purified expression profiles can be used to simulate other admixtures *in silico* in proportions relevant to particular and diverse disease settings. Further, as highly-multiplexed imaging (or other technologies) elucidate cellular distributions of intact tissue, these may be used to simulate admixtures with our data. As such, we believe the (purified and admixed) expression profiles generated for this DREAM Challenge will be a valuable resource as the community continues to improve deconvolution methodology, including for CD4+ T cell and macrophage populations, in cancer and non-malignant, immune-involved diseases.

Our community-wide comparison of 28 novel and published deconvolution methods revealed that levels of most major immune and stromal lineages were well predicted by most approaches. As such, our assessment suggests they provide robust signals for downstream correlative analyses. We observed considerable variability in predictive performance for minor lineages across methods. Though finer dissection was difficult for most sub-populations and most methods, even levels of the most challenging cell type, memory CD4+ T cells, were predicted at an accuracy (r = 0.62) that may be sufficient for some applications. Hence, our results allow researchers to choose the most appropriate method for studying an individual cell type. Where greater accuracy is needed, the purified immune and stromal expression profiles we generated should be a useful resource to the community in refining marker genes and "signature matrices" for deconvolution of the tumor microenvironment or of non-malignant contexts with significant immune modulation.

Data and Code Availability:

The data discussed in this publication have been deposited in NCBI's Gene Expression Omnibus[39] and are accessible through GEO Series accession number GSE199324 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE199324). Additionally, they are hosted on the Synapse data-sharing platform at syn21557721 and syn21571479, respectively. Round one code implementations are available for Aginome-XMU (https://github.com/xmuyulab/DCTD_Team_Aginome-XMU), DA_505 (https://github.com/martinguerrero89/Dream_Deconv_Challenge_Team_DA505), mitten_TDC19 (https://github.com/sdomanskyi/mitten_TDC19), and Biogem (https://github.com/giannimonaco/DREAMChallenge_Deconvolution).

Acknowledgements:

B.S.W, A.L., X.G., T.Y., and J.G. were supported by an NCI Coordinating Center for the Cancer Systems Biology Consortium and the Physical Sciences-Oncology Network U24 grant (CA209923). A.J.G., A.M.N. (CA209971) and L.M.H (CA209988) were supported by NCI Cancer Systems Biology Consortium U54 grants, and J.S.R. by the German Federal Ministry of Education and Research (BMBF, Computational Life Sciences grant no. 031L0181B). The experimental data were generated under an administrative supplement from the CSBC to A.J.G. and J.G. S.D., T.B. and C.P. were supported by National Institutes of Health, grant no. R01GM122085. F.P and A.d.R were supported by the Programme Cartes d'Identité des Tumeurs (CIT) from the Ligue Nationale Contre le Cancer. A.J.G., H.T.M., and C.D. were supported by the Cancer Immune Monitoring and Analysis Consortium (CIMAC) through NIH grant 1U24CA224309. J.J.W. was supported by the SiRIC-Curie program (SiRIC grant INCa-DGOS- 4654).

Conflict of interests:

J.S.R. received funding from GlaxoSmithKline and Sanofi and consultant fees from Travere Therapeutics and Astex Therapeutic. A.V. is currently employed by F. Hoffmann-La Roche Ltd.

Online Methods

Cell isolation

Cells were obtained from four sources, StemExpress (Folsom, CA), AllCells (Alameda, CA), ATCC (Manassas, VA), and the Human Immune Monitoring Centering (HIMC) at Stanford University. Immune and stromal cells provided by StemExpress and AllCells were isolated according to vendor protocols (**Table S1**).

Neutrophils and CD8$^+$ memory T cells were isolated by HIMC as follows. Human whole blood samples were collected under informed consent in EDTA-coated tubes. After 2h resting, whole blood samples were split for neutrophils and CD8 memory T cell isolation.

Neutrophil isolation was performed with MACSxpress® Whole Blood Neutrophil Isolation Kit (Miltenyi Biotec) according to manufacturer instructions. Briefly, the whole blood samples were mixed with the appropriate amount of isolation mix buffer consisting of magnetically-coated beads conjugated to antibodies targeting all the immune populations in the peripheral blood except for the neutrophils. The cell suspension containing the isolation mix was incubated for 5 min at room temperature on a low-speed rotator. Then magnetic separation was performed for 15 minutes prior to collecting the untouched neutrophils in a clean tube.

For CD8$^+$ memory T cell isolation, PBMCs were first isolated by density gradient centrifugation using Ficoll-Paque™ Plus (Cytiva). After washes, cell counts were obtained using a Vi-Cell XR cell viability analyzer (Beckman Coulter). Actual isolation was performed using a CD8+ Memory T Cell Isolation Kit (Miltenyi Biotec) per the manufacturer's instructions. Briefly, PBMCs were incubated at 4°C for 10 min with a cocktail of biotin-conjugated monoclonal antibodies against CD4, CD11c, CD14, CD15, CD16, CD19, CD34, CD36, CD45RA, CD56, CD57, CD61, CD123, CD141, TCRgd and CD235a. After washing, cells were resuspended in a solution of anti-biotin magnetic microbeads and incubated for 15 min at 4°C. After another wash, magnetic separation was performed using LS columns (Miltenyi Biotec), and we collected the cell fraction corresponding to CD45RO$^+$CD45RA$^-$CD56$^-$CD57$^-$CD8$^+$ T cells.

Finally, isolated neutrophils and CD8$^+$ memory T cells were resuspended in RNAprotect Cell Reagent (Qiagen) for RNA extraction.

Library preparation, RNA sequencing, and data processing

Libraries were prepared using the Clontech SMARTer Stranded Total RNA-Seq v2 kit (Takara Bio) according to manufacturer instructions. Paired-end RNA sequencing of all *in vitro* admixtures and purified samples was performed by MedGenome Inc, by pooling the indexed libraries across four lanes of an Illumina NovaSeq S4 flowcell.

Estimated transcript read counts and transcripts per million (tpm) were generated via pseudo-alignment with Kallisto v0.46.0 to hg38 using Homo_sapiens.GRCh38.cdna.all.idx. A translation table of Ensembl transcript ID to Ensembl gene ID and gene symbol was derived using biomaRt and stored on the Synapse platform at syn21574276. Estimated gene read counts and counts per million were calculated as the sum of transcript counts and tpms, respectively, associated with the gene via the translation table.

Training data curation

Participants were provided with a curated list of purified expression profiles in GEO.[40] GEO annotations were queried using regular expressions corresponding to cell populations of interest (e.g., with patterns "T.reg", "regulatory", and "FOXP3" for regulatory T-cells). Specifically, GEO annotations for fields "source_name" or involving "characteristic" (e.g., "characteristics_ch1") were accessed via `GEOmetadb` in R.[41] Cell type patterns are available at: https://github.com/Sage-Bionetworks/Tumor-Deconvolution-Challenge/blob/master/scripts/training-data-curation/phenotypes-to-query.tsv. Matches identified via `grepl` were manually curated, resulting in tables associating cell populations with GEO samples. These were further summarized according to dataset, listing the cell populations and the number of cell populations represented by each dataset. This was intended to help participants prioritize datasets representing many or multiple cell types of interest. Per-sample and per-dataset tables were created separately for microarray (**Table S10**, https://www.synapse.org/#!Synapse:syn18728081, and https://www.synapse.org/#!Synapse:syn18728088) and RNA-seq (**Table S11**, https://www.synapse.org/#!Synapse:syn18751454, and https://www.synapse.org/#!Synapse:syn18751460) platforms. Microarray expression datasets were identified as those having 'Expression profiling by array' in 'type' field of the 'gse' SQLite table available in `GEOmetadb` and as being assayed in human [i.e., as having the pattern 'sapiens' in the 'organism' field of the 'gpl' (platform) SQLite table]. RNA-seq expression datasets were similarly identified as having 'Expression profiling by high throughput sequencing' in the 'gse' table and as being assayed in human. Additionally, RNA-seq datasets were limited to those generated on Illumina platforms (i.e., as having a pattern of 'illumina' in the 'title' field of the 'gpl' table), specifically the HiSeq (with pattern 'hiseq' in the 'title' field) or NextSeq (with pattern 'nextseq' in the 'title' field) platforms. Participants were provided only identifiers of GEO datasets (GSMxxx) and samples (GSExxx). In particular, cross dataset normalization to account for batch effects was not performed, but rather was left to participants.

Unconstrained admixture generation

Unconstrained admixtures were defined in two stages: (1) a broken-stick approach partitioned the entire admixture across *n* cell types and (2) the proportion of each cell type *c* was restricted to be within min_c and max_c. In particular, for *n* cell types contributing a proportion $p <= 1$ (i.e., 100%) of the admixture total, the range 0 to *p* was randomly broken into *n* segments by choosing *n-1* boundaries of those segments. The *n-1* boundaries were uniformly sampled between a minimum cell population size of 0.01, i.e., 1%, and *p* in fixed-sized steps (of 0.01 unless otherwise specified), thus ensuring that each of the *n* populations was represented at a frequency of at least 1%. The resulting candidate proportions were excluded if the proportion p_c for any of the *n* cell types *c* was outside the bounds [min_c, max_c]. min_c was set to 0 and max_c was set to 1 (i.e., 100%), unless otherwise specified. Setting $p < 1$ allows the remaining *1-p* proportion to be allocated to an $(n+1)^{st}$ cell type, e.g., fixing a spike in population at proportion *1-p*.

Biological admixture generation

Biologically-constrained admixtures were defined such that cell population proportions were within biologically-reasonable limits, in particular those detected by CyTOF in PBMCs and aggregated in the 10,000 Immunomes ("10KIP") database [42] (downloaded on July 9, 2018), observed in single-cell (sc)RNA-seq studies of breast cancer ("Azizi"[43]) or melanoma ("Tirosh"[44]), or inferred by CIBERSORT in the "Thorsson" TCGA pan-cancer study.[45]

As none of these resources included all (coarse- or fine-grained) cell types to be deconvolved, several were combined in a hierarchical fashion. Two such hierarchical models, one based on the Thorsson study and the second based on the Azizi study, were created. At each level of the hierarchy, the models defined a minimum and maximum proportion for each population relative to its parent population. The minimum and maximum proportions for a particular cell type in a particular dataset were defined as two standard deviations above (~98th percentile) or below (~2nd percentile), respectively, the mean of proportions observed for that cell type in that dataset. The root of the model corresponds to the admixture of *n* cell populations. In both hierarchical models, the entire admixture was partitioned into cancer cell, leukocyte, and non-leukocyte stromal compartments, with minimum and maximum proportions for each compartment defined using the Thorsson study. Specifically, from the stromal fraction (SF), or total non-tumor cellular fraction, and the leukocyte fraction (LF) defined by Thorsson, we define the cancer cell proportion as 1 - SF, the leukocyte proportion as LF, and the non-leukocyte stromal proportion as 1 - SF - LF. Both hierarchical models next subdivided the non-leukocyte stromal compartment into (cancer associated) fibroblasts and endothelial cells using proportions of single cells observed in the Tirosh study. The original publication noted that four samples (CY58, CY67, CY72, and CY74) were experimentally enriched for immune infiltrates (CD45+). As such, the proportions inferred from them would not have represented cellular proportions relative to the entire cellular population. Hence, we excluded from analysis these samples, as well as CY75, which also did not have any tumor cells.

The Thorsson-based hierarchical model subdivided the leukocyte component into those inferred using CIBERSORT in the original study. Specifically, the leukocyte fraction was subdivided into the following sub-compartments: memory CD4+ T (i.e., T.Cells.CD4.Memory.Activated + T.Cells.CD4.Memory.Resting in the original publication), naïve CD4+ T (i.e., T.Cells.CD4.Naive), regulatory CD4+ T (i.e., T.Cells.Regulatory.Tregs), CD8+ T (i.e., T.Cells.CD8), memory B (i.e., B.Cells.Memory), naïve B (i.e., B.Cells.Naive), natural killer (i.e., NK_cells), neutrophils (i.e., Neutrophils), dendritic cells (i.e., Dendritic.Cells.Activated + Dendritic.Cells.Resting), monocytes (i.e., Monocytes), and macrophages (i.e., Macrophages.M0 + Macrophages.M1 + Macrophages.M2). Finally, the CD8+ T cell proportion was subdivided into memory and naïve CD8+ T cells using the KIP database.

The Azizi-based hierarchical model subdivided the leukocyte component into those reported in the Azizi study, specifically: T (i.e., T.cell in the original study), B (i.e., B.cell), natural killer (i.e., NK.cell), neutrophils (i.e., Neutrophil), dendritic cells (i.e., DC), monocytes (i.e., Monocyte), macrophages (i.e., Macrophage). Using the KIP database, the T cell compartment was further subdivided into memory, naïve, and regulatory CD4 T cells and memory and naïve CD8+ T cells, while the B cell compartment was further subdivided into memory and naïve B cells.

A single, final model was created from the Thorsson and Azizi models with minimum proportion for cell type *c* as the maximum of 0.01 and the minimum of the two model minimums for cell type *c* and with the maximum proportion as the maximum of the two model maximums.

The biologically-constrained admixtures were generated using the “Hit and Run” Markov Chain Monte Carlo (MCMC) method for sampling uniformly from convex samples defined by linear (equality and inequality) constraints, as implemented in the `hitandrun` library in R (https://github.com/gertvv/hitandrun). The system of linear constraints included a variable for each of the *n* populations. As in the unconstrained admixtures, the corresponding *n* proportions sum to p <= 1, with $p < 1$ allowing the remaining $1-p$ proportion to be allocated to an $(n+1)^{st}$ cell type. The resulting equality constraint was passed to the `solution.basis` function, whose output was in turn passed to the `createTransform` function. $2n$ linear inequality constraints were defined from the minimum and maximum proportions of each of the *n* populations. These were passed along with the output of `createTransform` to the `transformConstraints` function. An initial guess was created by passing these transformed constraints to `createSeedPoint` along with arguments `homogeneous=TRUE` and `randomize=TRUE`. Admixtures were sampled by passing the resulting seed and the transformed constraints to the `har` function along with parameters `N`, the number of iterations to run, set to $1000n^3$ and `N.thin`, the thinning factor indicating how many iterations to skip between samples, set to n^3.

Selection of extremal candidate admixtures

Unless otherwise indicated, we ordered candidate admixtures so as to prioritize those most different from another. In particular, we select as the first two candidate admixtures those having maximum sum of squared (proportion) differences. Then, we greedily selected admixtures that maximized the minimal sum of squared differences to those admixtures already selected.

In vitro validation admixture generation

60 biological admixtures and 36 unconstrained admixtures were defined using the procedures described in “Biological admixture generation” and “Unconstrained admixture generation,” respectively, with the exceptions noted below. Admixtures were defined over the cell populations having samples with sufficient mass and high RNA integrity number upon first assessment (**Table S2**): breast or colorectal cancer, endothelial cells, neutrophils, dendritic cells, monocytes, macrophages, NK, regulatory T, naïve CD4+ T, memory CD4+ T, naïve CD8+ T, memory CD8+ T, naïve B, and memory B cells. Admixtures were designed so as to minimize batch effects across vendors, with half of the biological and half of the unconstrained admixtures assigned immune cells from Stem Express wherever availability allowed (**Tables S3** and **S4**, respectively) and the rest assigned immune cells from AllCells wherever availability allowed (**Tables S5** and **S6**). However, following subsequent experimental quantification, several cell populations (neutrophils, naïve CD8+ T cells, and memory B cells) did not have sufficient material for inclusion in the admixtures. As such, the final *in vitro* admixtures used during the Challenge validation phase included: breast or colorectal cancer, endothelial cells, fibroblasts, dendritic cells, monocytes, macrophages, NK, regulatory T, naïveCD4+ T, memory CD4+ T,

memory CD8+ T, and naïve B cells. The final relative concentrations were rescaled relative to those designed computationally after excluding neutrophils, naïve CD8+ T cells, and memory B cells. The final *in vitro* admixtures used during the Challenge validation phase are provided in **Tables S7** and **S8**.

Biological admixtures were generated with a fixed tumor proportion *1-p* in the range 0.2 to 0.8 in steps of 0.01 (i.e., such that the *n* populations excluding the tumor cells have proportions summing to *1-p*). This fixed tumor proportion overrode the tumor proportion bounds defined in the Thorsson-based and Azizi-based biological models.

To assess the ability of methods to differentiate between closely-related “signal / decoy” pairs of cell types (e.g., memory vs naïve CD4+ T cells) and to improve our sensitivity in measuring this ability, within each *unconstrained in vitro* admixture we included a signal cell type with a high proportion (min_c of 0.2 and max_c of 0.35) and we excluded the decoy cell type (min_c and max_c of 0). For all other non-cancer cell types *c*, min_c was set to 0.01 and max_c to 0.5. We considered three ranges of cancer cell proportions: min_{cancer} = 0.2 to max_{cancer} = 0.3, min_{cancer} = 0.4 to max_{cancer} = 0.5 and min_{cancer} = 0.6 to max_{cancer} = 0.7. For each combination of these three cancer ranges and 11 signal / decoy pairs, we generated 1,000 candidate admixtures. Finally, we applied the strategy described in ‘Selection of extremal candidate admixtures’ with a minor modification: in each selection round, we only considered candidate admixtures generated for a particular signal / decoy pair and we iterated through the list of pairs with each round (recycling pairs as necessary).

Table: Signal / decoy cell type pairs

Source	Decoy
Monocytes	Dendritic cells
Macrophages	Monocytes
Dendritic cells	Macrophages
Naïve CD4+ T cells	Memory CD4+ T cells
Memory CD4+ T cells	Naïve CD4+ T cells
Naïve CD8+ T cells	Memory CD8+ T cells
Memory CD8+ T cells	Naïve CD8+ T cells
Naïve B cells	Memory B cells
Memory B cells	Naïve B cells
Tregs	Naïve CD4+ T cells
Naïve CD4+ T cells	Tregs

Code to generate the (unconstrained and biological) *in vitro* validation admixtures is in `analysis/admixtures/new-admixtures/gen-admixtures-061819.R`. The admixture expression data (i.e., represented as TPM and as read counts) are in Synapse folder syn21821096 and the (ground truth) admixtures are in Synapse folder syn21820011. They are the datasets designated ‘DS1’, ‘DS2’, ‘DS3’, and ‘DS4.’ Participants were told the cancer cell type included in each dataset (BRCA or CRC), which was BRCA for DS1 and DS3 and CRC for DS2 and DS4.

In silico validation admixture generation

Insufficient RNA was available to include naïve CD8+ T cells and neutrophils in the *in vitro* admixtures. However, material was available to sequence the purified samples. This allowed us to generate *in silico* admixtures using the above biological and unconstrained procedures, such that the final *in silico* admixtures used during the Challenge validation phase included: breast or colorectal cancer, endothelial cells, fibroblasts, dendritic cells, monocytes, macrophages, neutrophils, NK, regulatory T, naïve CD4+ T, memory CD4+T, memory CD8+ T, naïve CD8+ T, and naïve B cells. Memory B cells were unavailable to be included in either the *in vitro* or *in silico* admixtures.

For each of the two cancers (breast or colorectal cancer) and each of the two vendor batches (i.e., Stem Express-enriched or AllCells-enriched, as described in “*In vitro* validation admixture generation”), we generated 15 coarse- and 15 fine-grained unconstrained admixtures and 20 coarse- and 20 fine-grained biological admixtures. Unconstrained admixtures were generated as described in “Unconstrained admixture generation,” except with a step size of 0.001. Further, we did not diversify admixtures by attempting to maximize the distance between them (as described in “Selection of extremal candidate admixtures”). We did diversify the biological admixtures, by generating each of the 20 coarse-grained admixtures (and similarly for the fine-grained admixtures) in five batches and by applying the distance maximization procedure to select the four most distant admixtures from those in each batch of MCMC samples.

The transcripts per million (TPM)-based expression of *in silico* admixtures were generated as the weighted sum of the purified TPM expression profiles. For counts-based expression of the admixtures, we first normalized the gene counts for each purified sample by the total counts for that sample, multiplied by the median across samples of sample total counts to obtain pseudo-counts on the same scale for each sample, and finally derived the admixtures as the weighted sum of the pseudo-counts.

Code to generate the (unconstrained and biological) *in silico* validation admixtures is in `analysis/validation_data/qc/generate-validation-in-silico-admixtures.R`. The admixture expression data (i.e., represented as TPM and as read counts) and (ground truth) admixtures are in the same Synapse folders as the corresponding *in vitro* data — i.e., syn21821096 and syn21820011, respectively. They are the datasets designated ‘AA’, ‘AB’, ‘AE’, and ‘AF.’ Participants were told the cancer cell type included in each dataset (BRCA or CRC), which was BRCA for AA and AE and CRC for AB and AF.

Comparator method evaluation

All comparator methods were executed by Sage Bionetworks (A. L. or B. S. W.) and without modification. All methods were passed expression in linear form.

CIBERSORT[19] was executed with arguments `abs_method = “sig.score”`, `absmean = TRUE`, `QN = FALSE`, and all other arguments default (including `absolute = FALSE`) via the script `CIBERSORT.R`. Outputs from CIBERSORT were translated into Challenge populations according to:

Challenge population	CIBERSORT output
B.cells	B cells naive + B cells memory
memory.B.cells	B cells memory
naive.B.cells	B cells naive
CD4.T.cells	T cells CD4 naive + T cells CD4 memory resting + T cells CD4 memory activated + T cells regulatory (Tregs) + T cells follicular helper
memory.CD4.T.cells	T cells CD4 memory activated + T cells CD4 memory resting
naive.CD4.T.cells	T cells CD4 naive
regulatory.T.cells	T cells regulatory (Tregs)
CD8.T.cells	T cells CD8 + T cells gamma delta
memory.CD8.T.cells	NA
naive.CD8.T.cells	NA
NK.cells	NK cells resting + NK cells activated
neutrophils	Neutrophils
monocytic.lineage	Monocytes + Macrophages M0 + Macrophages M1 + Macrophages M2 + Dendritic cells resting + Dendritic cells activated
macrophages	Macrophages M0 + Macrophages M1 + Macrophages M2
monocytes	Monocytes
myeloid.dendritic.cells	Dendritic cells resting + Dendritic cells activated
endothelial cells	NA
fibroblasts	NA

CIBERSORTx[32] was run in two phases: (1) The first phase separates immune cells (expressing CD45), endothelial cells (CD31), fibroblasts (CD10), and epithelial / tumor cells (EpCAM) using a signature matrix (Supplementary Table 2L of Ref[32]) derived from FACS purification of these four cell types within 26 surgically-resected primary non-small cell lung cancer biopsies. (2) The second phase further divides the immune compartment into the 22 immune sub-populations represented by the same LM22 signature matrix originally published with CIBERSORT.[19] In both cases, CIBERSORTx was executed using the cibersortx/fractions docker container obtained from https://cibersortx.stanford.edu/, with arguments `--rmbatchBmode TRUE --perm 1 --verbose TRUE --QN FALSE`. The `--sigmatrix` parameter was used to specify the appropriate signature matrix.

Outputs from the two phases of CIBERSORTx were translated into Challenge populations by scaling the output of LM22 phase by the CD45 output from the first phase according to:

Challenge population	CIBERSORTx output
B.cells	CD45 * [B cells naive + B cells memory]
memory.B.cells	CD45 * [B cells memory]
naive.B.cells	CD45 * [B cells naive]
CD4.T.cells	CD45 * [T cells CD4 naive + T cells CD4 memory resting + T cells CD4 memory activated + T cells regulatory (Tregs) + T cells follicular helper]
memory.CD4.T.cells	CD45 * [T cells CD4 memory activated + T cells CD4 memory resting]
naive.CD4.T.cells	CD45 * [T cells CD4 naive]
regulatory.T.cells	CD45 * [T cells regulatory (Tregs)]
CD8.T.cells	CD45 * [T cells CD8 + T cells gamma delta]
memory.CD8.T.cells	NA
naive.CD8.T.cells	NA
NK.cells	CD45 * [NK cells resting + NK cells activated]
neutrophils	CD45 * [Neutrophils]
monocytic.lineage	CD45 * [Monocytes + Macrophages M0 + Macrophages M1 + Macrophages M2 + Dendritic cells resting + Dendritic cells activated]
macrophages	CD45 * [Macrophages M0 + Macrophages M1 + Macrophages M2]
monocytes	CD45 * [Monocytes]
myeloid.dendritic.cells	CD45 * [Dendritic cells resting + Dendritic cells activated]
endothelial cells	CD31
fibroblasts	CD10

EPIC[20] was executed using the `EPIC` function from the `EPIC` R library, with the arguments `reference = "BRef"` and `mRNA_cell = FALSE`.

Outputs from EPIC were translated into Challenge populations according to:

Challenge population	EPIC output
B.cells	Bcells
memory.B.cells	NA
naive.B.cells	NA
CD4.T.cells	CD4_Tcells
memory.CD4.T.cells	NA
naive.CD4.T.cells	NA
regulatory.T.cells	NA
CD8.T.cells	CD8_Tcells
memory.CD8.T.cells	NA
naive.CD8.T.cells	NA
NK.cells	NKcells
neutrophils	Neutrophils
monocytic.lineage	NA
macrophages	Macrophages
monocytes	Monocytes
myeloid.dendritic.cells	NA
endothelial cells	Endothelial
fibroblasts	CAFs

MCP-counter[21] was executed using the `MCPcounter.estimate` function from the `MCPcounter` R library, with the argument `featuresType = 'HUGO_symbols'`.

Outputs from MCP-counter were translated into Challenge populations according to:

Challenge population	MCP-counter output
B.cells	B lineage
memory.B.cells	NA
naive.B.cells	NA

CD4.T.cells	NA
memory.CD4.T.cells	NA
naive.CD4.T.cells	NA
regulatory.T.cells	NA
CD8.T.cells	CD8 T cells
memory.CD8.T.cells	NA
naive.CD8.T.cells	NA
NK.cells	NK cells
neutrophils	Neutrophils
monocytic.lineage	Monocytic lineage
macrophages	NA
monocytes	NA
myeloid.dendritic.cells	Myeloid dendritic cells
endothelial cells	Endothelial cells
fibroblasts	Fibroblasts

quanTIseq[23] was executed using the `deconvolute_quantiseq` function implemented in the `immundeconv` R library.[26] `deconvolute_quantiseq` was passed the arguments `tumor = TRUE`, `arrays = FALSE`, and `scale_mrna = FALSE`. If parameterization of `deconvolute_quantiseq` returned any invalid (“not-a-number”) results, it was re-run with the additional argument `method = “huber”`.

Outputs from quanTIseq were translated into Challenge populations according to:

Challenge population	quanTIseq output
B.cells	B.cells
memory.B.cells	NA
naive.B.cells	NA
CD4.T.cells	T.cells.CD4
memory.CD4.T.cells	NA
naive.CD4.T.cells	NA
regulatory.T.cells	Tregs
CD8.T.cells	T.cells.CD8
memory.CD8.T.cells	NA

naive.CD8.T.cells	NA
NK.cells	NK.cells
neutrophils	Neutrophils
monocytic.lineage	Macrophages.M1 + Macrophages.M2 + Monocytes + Dendritic.cells
macrophages	Macrophages.M1 + Macrophages.M2
monocytes	Monocytes
myeloid.dendritic.cells	Dendritic cells
endothelial cells	NA
fibroblasts	NA

xCell (29141660) was executed using the `xCellAnalysis` function of the `xCell` R library, with the argument `rnaseq = TRUE` and the argument `cell.types.use` set to the corresponding cell types within each challenge [i.e., to `c(“B-cells”, “CD4+ T-cells”, “CD8+ T-cells”, “NK cells”, “Neutrophils”, “Monocytes”, “Fibroblasts”, and “Endothelial cells”)` in the coarse-grained sub-Challenge and to `c(“Memory B-cells”, “naive B-cells”, “CD4+ memory T-cells”, “CD4+ naive T-cells”, “Treg”, “CD8+ Tem”, “CD8+ naive T-cells”, “NK cells”, “Neutrophils”, “Monocytes”, “DC”, “Macrophages”, “Fibroblasts”, “Endothelial cells”)` in the fine-grained sub-Challenge].

Outputs from xCell were translated into Challenge populations according to:

Challenge population	xCell output
B.cells	B-cells
memory.B.cells	Memory B-cells
naive.B.cells	naive B-cells
CD4.T.cells	CD4+ T-cells
memory.CD4.T.cells	CD4+ memory T-cells
naive.CD4.T.cells	CD4+ naive T-cells
regulatory.T.cells	Tregs
CD8.T.cells	CD8+ T-cells
memory.CD8.T.cells	CD8+ Tem
naive.CD8.T.cells	CD8+ naive T-cells
NK.cells	NK cells

neutrophils	Neutrophils
monocytic.lineage	Monocytes
macrophages	Macrophages
monocytes	Monocytes
myeloid.dendritic.cells	DC
endothelial cells	Endothelial cells
fibroblasts	Fibroblasts

Deconvolution method scoring and comparison

Pearson and Spearman correlation-based scores were calculated hierarchically for a given method a: For each cell type c and validation dataset d (i.e., DS1, DS2, DS3, DS4, AA, AB, AE, and AF), the correlation between the values predicted by a and the ground truth was calculated. These correlations were then averaged over all cell types c to define the score of method a for dataset d. These dataset-level scores were finally averaged over all datasets d to define the aggregate score for method a.

To assess scoring differences in the primary metric between a top-performing method a and another method b, we computed a Bayes factor $K_{a,b}$ over 1000 bootstrap samples and considered $K_{a,b} > 3$ as indicating a significant difference. More specifically, we bootstrap sampled (i.e., sampled with replacement) prediction scores separately within each dataset (i.e., DS1, DS2, DS3, DS4, AA, AB, AE, and AF), calculated a Pearson correlation-based score S_i^a between the predictions in bootstrap sample i for method a and the corresponding ground truth values (and similarly for S_i^b and method b), and calculated $K_{a,b}$ as

$$K_{a,b} = \frac{\#\,\text{of bootstrap samples for which method } a \text{ outperforms method } b}{\#\,\text{of bootstrap samples for which method } b \text{ outperforms method } a} = \frac{\sum_{i=1}^{1000} \mathbf{1}\left(S_i^a > S_i^b\right)}{\sum_{i=1}^{1000} \mathbf{1}\left(S_i^b > S_i^a\right)} \ ,$$

where $\mathbf{1}(x)$ is the indicator function that equals 1 if and only if x is true and is 0 otherwise. A tie between methods a and b (i.e., $K_{a,b} \leq 3$) would have been resolved using the secondary Spearman correlation-based metric. However, this did not occur in the first submission results. Distributions, medians, and means over the S_i^a are reported for the Pearson correlation-based scores in the figures (e.g., **Fig. 2A**) and main text *in lieu of a single score on the original validation data*. Similar bootstrapped distributions, medians, and means were calculated for the Spearman correlation-based scores and are likewise reported.

Intra-sample deconvolution method assessment

We assessed prediction performance across cell types *within* samples for those methods outputting normalized scores (CCB, D3Team, NYIT_glomerular), proportions (Patrick), and fractions (Aginome-XMU, Biogem, CIBERSORTx, DA_505, HM159, IZI, jbkcose, jdroz, LeiliLab, NPU,REGGEN_LAB, Rubbernecks, Tonys Boys, xCell). We computed the Pearson correlation, Spearman correlation, and root-mean-square-error (RMSE) across cell types within a sample. To assess ties across teams, we fit a linear model whose response was the metric value and whose dependent variable was the team. The top-scoring team (based on ordering of the median value across samples) was used as the reference in the linear model, which was fit using `lm` in R. Teams were considered tied with the top performer if their corresponding *t*-statistic *p*-value was > 0.05, as computed from the model fit using `summary`.

Several outliers were excluded from the RMSE sub-plots of **Fig. 4** (Patrick from **Figs. 6A** and **6B** and NYIT_glomerular from **Fig. 6A**).

Deconvolution method specificity assessment

To assess deconvolution method specificity, we calculated the (min-max) normalized prediction for a cell type *c* in a sample *s'* purified for some cell type *c'*. These normalized predictions are displayed in the heatmap of **Fig. 7A**, with cell types as columns and samples as rows. Predictions were normalized so as to be comparable across methods independent of the scale of the prediction (e.g., both unnormalized scores comparable across samples and proportions comparable across samples and cell types). The min-max normalization of a prediction *pred(c', s', m)* for cell type *c',* method *m*, and purified sample *s'* was defined as $[\, pred(c,s',m) - min_{s''}\, pred(c,s'',m) \,] / [\, max_{s''}\, pred(c,s'',m) - min_{s''}\, pred(c,s'',m) \,]$. "Spillover" *into* (predicted) cell type *c* for method *m* was calculated as the above normalized prediction for cell type *c* and method *m* averaged over samples *s'* purified for some cell type *c'* ≠ *c* (i.e., the mean of the column corresponding to cell type *c* in **Fig. 7A** that excludes elements in which *c* is in the sample corresponding to the row). These spillovers were then averaged over sub-Challenges and the resulting distributions were plotted in **Fig. 7B**. Distributions of spillovers over cell types are plotted for each method in the coarse- (**Fig. 7C**) and fine-grained (**Fig. 7D**) sub-Challenges.

Code to generate the format the purified expression profiles is in `analysis/specificity-analysis/create-spillover-dataset.R`.
The processed expression data (i.e., represented as TPM and as read counts) and the (ground truth) admixtures are in Synapse folder syn22392130.

Deconvolution method sensitivity assessment

To assess deconvolution method sensitivity in detecting each cell type *c*, we generated *in silico* admixtures in which we computationally spiked in *c* at regular proportions. We considered 49 spike-in levels from 0% to 0.1% in increments of 0.01%, from 0.1% to 1% in increments of 0.1%, from 1% to 20% in increments of 1%, and from 20% to 40% in increments of 2%. Cancer cells were neither used as spike ins nor included within the admixtures. Otherwise, all cell types with available purified expression profiles were included, namely endothelial cells, fibroblasts, dendritic cells, monocytes, macrophages, neutrophils, NK, regulatory T, naïve CD4+ T, memory

CD4+ T, memory CD8+ T, naïve CD8+ T, and naïve B cells. Expression profiles of *in silico* admixtures were generated as described in "*In silico* validation admixture generation."

We defined the limit of detection (LoD) for cell type *c* and method *m* as the least frequency at and above which *m*'s prediction for *c* is statistically distinct from the baseline admixture (0% spike in). We assessed statistical significance using the Wilcoxon test as implemented in the `compare_means` function of the `ggpubr` library and using a raw (uncorrected) *p*-value cutoff of 0.01.

We generated both unconstrained and biological admixtures, using both fine- and coarse-grained populations. For unconstrained admixtures, we used the broken stick procedure described in "Unconstrained admixture generation," except with a step size of 0.001 and without diversifying admixtures as described in "Selection of extremal candidate admixtures." For each of the two vendor batches (i.e., Stem Express-enriched or AllCells-enriched, as described in "*In vitro* validation admixture generation") and each spike in level *s*, we generated five coarse- and five fine-grained unconstrained admixtures such that the proportions of the *n* populations summed to *1-s*. We used these same five admixtures for each of the spike-in experiments by simply assigning the population with fixed proportion *s* the name of the population to be spiked in.

For unconstrained coarse-grained populations, we wanted to fix the level of the parental population (e.g., CD8+ T cells) rather than the sub-populations comprising it (i.e., memory and naïve CD8+ T cells). We defined coarse-grained admixtures at the level of the coarse-grained populations, but to concretely instantiate them we distributed the proportion of each parental population into its corresponding sub-populations. We did so by randomly dividing the proportion into *m* sub-populations using a flat Dirichlet distribution (using the `rdirichlet` function in the `dirichlet` library) whose *m* parameters were set to *1/m*.

We generated biological spike-in admixtures using a similar approach as for the unconstrained spike-in admixtures, except using the procedure described in "Biological admixture generation." For each of the two vendor batches, we generated five coarse- and five fine-grained unconstrained admixtures such that the proportions of the *n* populations summed to *1-s*. We again used a Dirichlet distribution to distribute proportions from a parental population to sub-populations. For both coarse- and fine-grained admixtures, we diversified the admixtures as described in "Selection of extremal candidate admixtures."

Code to generate the *in silico* spike in admixtures is in `analysis/in-silico-admixtures/generate-in-silico-admixtures.R`.
The processed expression data (i.e., represented as TPM and as read counts) and the (ground truth) admixtures are in Synapse folder syn22361008.

Ensemble method

We sought to define an ensemble method to aggregate predictions across all participant and comparator methods. Since the scales of predicted values vary according to the type of method output (scores, normalized scores, or fractions), we decided to aggregate the *ranks* of the predicted values across methods rather than the predicted values themselves. This is an instance of the consensus ranking, or social choice, problem in which we seek a ranking that summarizes the individual rankings of *n* judges (or, in our case, methods) for *m* objects (here, samples). We could define a consensus rank-based ensemble method using `ConsRank`,[46] for example, which uses heuristic algorithms to define one or more consensus rankings. However, as the (approximate) solutions are not guaranteed to be unique, we decided instead to take the more straightforward and more computationally-efficient approach of simply defining the ensemble ranking as the mean of the individual rankings.

Aginome-XMU deconvolution method

Summary Sentence

We used a deep learning-based prediction model for cell fraction prediction from bulk RNA-seq data.

Introduction

Numerous computational deconvolution methods have been proposed to estimate the abundance of individual cell types from bulk RNA-seq data of heterogeneous tissues. Unfortunately, there are still many difficulties hindering the performance of these algorithms, such as the collinearity of expression from different cell types, lack of specific marker genes, and the difficulty in dealing with batch effects of technical variation of data from different platforms.

In recent years, we have witnessed a paradigm shift in the machine learning community. In contrast to traditional feature engineering approaches where feature extraction and classification are optimized separately, new techniques based on learning internal representations from data directly have been explored. These data-driven approaches, especially deep learning, have allowed significant research breakthroughs and have rapidly spread across multiple application domains such as computer vision, audio recognition, and human language processing.

In this work, we hypothesized that by using deep learning, we could capture the intrinsic relationship between cellular composition of tissue and its bulky gene expression profile without the need for manual feature selection or marker gene identification. To test this hypothesis, we developed a deep learning-based cell type fraction prediction model with in silico mixing training data from multiple microarray and RNA-seq (**Table S12**), as well as scRNA-seq, datasets. We tested its performance on our testing sets comprised of nine public datasets (GSE64385, GSE65133, GSE106898, GSE107011, GSE64655, GSE127813, SDY67, GSE59654, GSE107990) and datasets from three public leaderboard rounds. We found that the algorithm shows better accuracy and stability in comparison with existing methods on our testing sets.

Methods

sub-Challenge 1

Prediction method

We used a deep learning-based prediction model for cell fraction prediction from bulk RNA-seq data. Briefly, we trained deep feed-forward, fully connected neural networks (multilayer perceptron networks) on in silico mixed training data. The network consists of one input layer, five fully-connected hidden layers and one output layer and was implemented with the PyTorch framework (v1.0.1) in Python (v3.7.3). The detailed description of this deep learning-based model have been published.[33]

In addition to the deep learning-based method, in our submission we use an ensemble method to further improve the prediction results. Specifically, we noticed that summarization method (MCP-counter[21]) performs better on neutrophils, fibroblasts, and endothelial cells on our testing sets. Therefore, MCP-counter was used to predict the cell type proportion in our second submission to sub-Challenge 1.

Prediction output

Our deep learning-based method outputs the absolute fraction of target cell types. Hence, the results can be compared across both cell types and samples. The method can score all cell types present in the sample including immune, stroma and cancer. Non-negative or sum to one constraint were currently not enforced in our model. As the deep learning model produces absolute cell fraction prediction, the fraction of “other” cell types can be calculated by subtracting the total of the predicted fractions of all the supported cell types from 1.

Normalization/pre-processing data

We use min-max scaling to normalize the expression data. In the deep-learning model, we do not apply feature selection. Instead, all the available gene expression data, after intersection of the genes of the training data with the genes of the test data, were used as our input to the deep learning model.

Training models

We mixed the purified expressions of 8 cell types in random proportions, and at the same time mixed a part of the unknown cell lines (e.g., cancer cell line). These purified expression samples came from microarray, RNA-seq, scRNA-seq. The expression of a simulated sample e was calculated as

$$\boldsymbol{e} = \sum_{k=1}^{C} p_k \boldsymbol{\epsilon}_k,$$
,

where C is the number of cell types involved in mixing, p_k, $k = 1, \ldots, C$ are random variables with Dirichlet distribution that determined the fractions of different cells in the in silico mixed sample, and ϵ_k is the expression profile of a randomly selected purified sample of cell type k from the respective RNA-seq or microarray dataset.

For scRNA-seq dataset, $\boldsymbol{e}$ is given by

$$\boldsymbol{e} = \sum_{k=1}^{C} \sum_{j=1}^{n_k} \boldsymbol{\epsilon}_{kj},$$
,

where $n_k = 500 \cdot p_k$ is the number of cells of type k extracted randomly from scRNA-seq datasets for mixing, and ϵ_{kj} denote their expression profiles. Note that in this case e were further TPM-normalized before used as training data.

In this way, the expression of heterogeneous samples in the real-world tumor microenvironment is simulated, and large number of samples can be generated as training data.

In addition, we also used a previously-described[33] in silico mixing method to expand a large number of training samples through mixing RNA-seq dataset SDY67 with ground truth and scRNA-seq data.

We used bagging to select 30%-100% training datasets for model training. In model training, we used cross-validation through selecting 20% training data as a validation set to evaluate the training degree of the model. We used early stopping such that training is stopped when the loss on the validation set is not reduced after 10 epochs.

sub-Challenge 2

We used a similar approach as in sub-Challenge 1 to create our prediction model. The only exception is that in our third submission, xCell[24] was used to create prediction of cell type myeloid dendritic cells as it performs better than other methods on this cell type on our testing sets.

Results

Table: Pearson correlation results of the coarse-grained cell type deconvolution track

Team	Aginome-XMU	Aginome-XMU
objectId	9704292	9704388
submission	first submission	second submission

B.cells	0.894/DNN	0.894/DNN
CD4.T.cells	0.862/DNN	0.862/DNN
CD8.T.cells	0.896/DNN	0.896/DNN
monocytic.lineage	0.931/DNN	0.931/DNN
NK.cells	0.826/DNN	0.826/DNN
neutrophils	0.94/DNN	0.993/MCP-counter
fibroblasts	0.851/DNN	0.98/MCP-counter
endothelial.cells	0.864/DNN	0.98/MCP-counter
Grand mean	0.883	0.92

In the coarse cell type track, we made two submissions. In the first submission, we use a deep learning-based prediction model as described above on all cell types. For each cell type, we chose the best model that achieved the highest Pearson correlation results on our testing datasets. The grand mean of the Pearson correlation over all cell types was 88.3%.

We noticed that on our testing datasets, the deep learning-based model did not outperform marker gene-based models such as MCP-counter on some cell types, namely, neutrophils, fibroblasts, and endothelial cells. We believe that these cell types have highly specific marker genes, therefore it is relatively easy to predict using marker gene-based models. Therefore, we replaced the method with MCP-counter on these three cell types to profile the abundance of the cells in the second submission. The grand mean of the Pearson correlation over all cell types increased to 92% in this submission.

Table: Pearson correlation results of the fine-grained cell type deconvolution track

Team	**Aginome-XMU**	**Aginome-XMU**	**Aginome-XMU**
objectId	9704261	9704399	9704690
submission	first submission	second submission	third submission
naive.B.cells	0.939/DNN	0.939/DNN	0.939/DNN
naive.CD4.T.cells	0.771/DNN	0.771/DNN	0.771/DNN
memory.CD4.T.cells	0.523/DNN	0.523/DNN	0.523/DNN
regulatory.T.cells	0.8/DNN	0.8/DNN	0.8/DNN
naive.CD8.T.cells	0.875/DNN	0.868/DNN	0.875/DNN

memory.CD8.T.cells	0.645/DNN	0.573/DNN	0.645/DNN
NK.cells	0.91/DNN	0.91/DNN	0.91/DNN
monocytes	0.733/DNN	0.733/DNN	0.733/DNN
myeloid.dendritic.cells	0.468/DNN	0.468/DNN	0.815/xCell
macrophages	0.672/DNN	0.672/DNN	0.672/DNN
neutrophils	0.915/DNN	0.999/MCP-counter	0.999/MCP-counter
fibroblasts	0.928/DNN	0.981/MCP-counter	0.981/MCP-counter
endothelial.cells	0.95/DNN	0.979/MCP-counter	0.979/MCP-counter
Grand mean	0.779	0.786	0.819

In the fine cell type track, we made three submissions. In the first submission, we used a deep learning model to predict the fraction of 13 fine cell types. The grand mean of the Pearson correlation was 77.9%. To further improve the performance, we replaced deep learning-based model with MCP-counter for three cell types (neutrophils, fibroblasts, and endothelial cells) on which marker gene-based model performs better than deep learning-based model in the testing set, and obtained a grand mean of 78.6%. In the final submission, we further replaced the model for myeloid dendritic cells to xCell, which performed very well on this cell type in our test data. The final grand mean score was 81.9%.

Conclusion/Discussion

Looking forward, we believe that the following points can be further elaborated to improve the cell fraction deconvolution performance.

First, we used all the gene expression data for genes available in both training and testing datasets as the input of our prediction model. However, further analysis showed that different genes contribute differently to the prediction results. Therefore, it is possible to significantly reduce the number of genes used as prediction input by identifying a small subset of genes that contribute most to the prediction results.

Secondly, we found that although the deep learning-based model performs relatively well on all cell types, for some cell types, shallow models such as MCP-counter or xCell perform better. Therefore, it seems that both deep and shallow models provide complementary information to some extent, and a late fusion strategy to combine both models could potentially be used to further improve the performance. Note that although shallow models such as MCP-counter or xCell perform better on those cell types, they do not produce absolute cell fractions as the deep learning-based model does. Hence, the late fusion strategy could potentially bring additional advantage to unify different kinds of output scores to absolute scores that can be compared across cell types and samples.

Finally, as suggested in reported results,[33] it may be challenging to produce a "one-size-fits-all" prediction model that consistently performs well across datasets of different sources due to the existence of batch effects and technical variations from different experiment sites. A better solution seems to be training a "dataset-specific prediction model". Preliminary results on datasets with a small number of calibration samples with ground truth cell fractions are very promising in our experiments.

DA_505 deconvolution method

Summary Sentence

Our method utilizes Random Forest regression to select the most significant features associated with cell-type proportions and (support vector or penalized linear) regression to predict different cell type proportions in complex RNA admixtures using a novel approach to normalize and mix samples of purified cell types.

Background/Intro

Usually, RNA-Seq gene expression analysis is approached by applying a plethora of scaling and normalization methods to the sample's raw counts to apply different data analysis methods on the processed estimates. Although current techniques have proven to be solid and robust, many rely on inter-sample information gathering to perform their normalization procedures such as quantile normalization or using the library size to scale the expression counts (Voom normalization). This presents great obstacles when these methods are applied to samples that share the same biological origin but are from different technical contexts. These issues represent a great obstacle not only when we want to train a model using multiple datasets from different research centers, but also when we wish to apply our prediction models to new unseen samples from different centers or as isolated cases (i.e., when we desire to apply the model as a product to aid medical diagnosis and decide treatment).

Here we propose a normalization method to adjust count values measured on different scales into a common scale disregarding the sample counts' prior probability distribution or library size, utilizing a transformed ranking system under the assumption that the features expression order in a given sample would be unaltered under different scales and probability distributions. In particular, this approach allowed us to mix RNA-Seq experiments of purified cell lines into complex admixtures to train different machine learning models to predict cell-type proportions in new unseen cases.

To test our hypothesis we trained three mainstream Machine Learning models for each cell type using the admixtures created from 23 different datasets of bulk RNA-Seq experiments of purified cell types. Partial Least Squares (PLS), Penalized linear regression (GLMNET), and Support Vector Regressors (SVR) were used combined with a feature selection approach using Random Forest (RF) to reduce the dimensionality of the problem.

Methods

In both sub-Challenge 1 and sub-Challenge 2 the same approach was used with a variation in the creation of the admixtures used to train the models. In both cases, our approach performed well, which shows the versatility of the framework proposed.

Intuitively, the workflow can be divided into three phases: 1) Data curation, downloading, and pre-processing. 2) Data normalization, quality control and admixture creation, and 3) model training, hyperparameter tuning, and model selection. Since the SVR was the model with the best results, we will describe only that model here. The SVR model was used for our third submissions, whereas GLMNET was used for our first two submissions. The whole workflow is depicted in **Fig. S2** and each step will be explained in the subsequent sections.

Normalization/pre-processing data

We believe that the most important step in our framework was the data pre-processing and normalization.

To train our model 23 different sequencing datasets were manually curated and download from public repositories where purified immune, stromal, and cancer cell types from different origins and conditions were sequenced using different sequencing platforms (Genome Analyzer II, NextSeq500, Hiseq2000, 2500, 4000, Novaseq600, etc). The selection criteria to choose the different experiments were:

- Human isolated stromal, immune or cancer cell types
- No molecular modifications of the cells were performed on the samples
- The samples represent biologically plausible conditions

Once downloaded, samples were manually labeled in a unified coarse-grain or fine-grain classification. For the coarse-grain class the subtypes “cancer”, “fibroblast”, “endothelial”, “NK cells”, “Neutrophils”, “monocytic lineage”, “B cells”, “CD4 T cells”, “CD8 T cells” and “Adipocytes” were considered. Other cell types present in the used datasets were classified as “others”. For the fine-grain labels, immune cells were further categorized. “Monocytic lineage” was divided into “monocytes”, “macrophages” and “myeloid dendritic cells”. B, CD4, and CD8 cells were further classified into “naïve” and “memory”, and the “regulatory T cell” label was also included for CD4 cells. Also, the “others” label was used.

The metadata of the datasets is included in **Table S13.**

Once all the samples were downloaded, a quality control analysis was performed upon all available samples using FASTQC software.[47]

After performing the first fastqc analysis we noticed that some sample sets, e.g. ERP002049, had samples with reads of very poor quality which lead us to trial a series of trimmomatic settings to filter out poor quality reads.[48] Our strategy was to seek a balance between achieving the highest quality and retaining the most number of fragments possible for each sample. As the alignment software used (Kallisto) only looks for exact matches,[49] we placed strong value on high quality reads to reduce the possibility of misalignments. By default Kallisto uses 31 bp kmers to align the reads to features, considering this we opted for a MINLEN setting of 36 and used the trimmomatic MAXINFO option to perform an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. Using these settings mean fragment lengths for each sample set typically ranged from 40 - 49 bp. Eight samples from SRP031776 had read lengths of 27bp before trimming and were omitted from our analysis.

Kallisto pseudoalignment requires the strandedness of the library to be provided, which we verified prior to transcript count quantification using Kallisto’s default parameters. The transcript – count matrix was then collapsed to a gene – count matrix using the tximport R package.[50]

Once the count matrix was obtained, all samples were normalized using an ad-hoc developed normalization procedure we named ‘RHINO’. The rationale behind this approach is based on the observation of methods that use library size to scale counts, such as Voom transformation,[51] usually perform satisfactorily when comparing samples from the same origin, but fails to correct probability distribution differences between samples from different experimental contexts. Due to this observation, other widely used methods (i.e. quantile normalization) rely on taking all the samples to a single probability distribution under the assumption that all the sample counts come from the same underlying distribution. This second approach usually corrects satisfactorily the inequalities found from different sample centers, nevertheless, it fails to correct for systematic overdispersion of the counts from low expression genes (own observations, manuscript under preparation).

Taking these observations into consideration, we propose a new transformation method to normalize sample counts across datasets of different origins;

$$X\prime = -log_2\left[rank\left(-1X\right)\right] + log_2\left(n\right)$$

Where X is the count matrix of a given sample with dimensions $n * 1$, with n being the number of features in the count matrix. With this method, we only take into account the ranks of the expressed features using the ‘max’ form to account for ties in the rankings.

Once samples were collected and normalized, a sample selection procedure was performed to discard possible low quality or mislabeled purified samples. Briefly, a penalized multinomial classifier was trained with the complete dataset. Then, we selected the non-zero coefficients of the model and used this subset of features to calculate Pearson’s distance between samples.

Since some cell types might be similar or related in their origin (for example CD4 and CD8 T cells), we used a community detection clustering approach to detect communities of samples from the same origin. In particular, samples with Pearson's correlation coefficients greater than 0.8 were considered to be similar, information that was used to perform a Louvain clustering procedure that detected clearly defined communities of unique cell populations.[52] Samples belonging to these clearly defined clusters were used to create the admixtures while samples not falling into these groups were discarded.

Once the prototypic samples for each cell type were defined, 75% of the samples of each cell population were used in a training set, and the remaining 25% of the samples were used as a validation set to perform parameter tuning.

To create the admixtures to perform the model training, a data augmentation approach was used where samples from within the same cell type population were mixed to create 'new' purified samples, and these 'new' samples were mixed in silico with samples from other cell populations in known proportions to create a training and validation admixture set of 2000 and 500 samples respectively.

Using these two sets a Random Forest (RF) for feature selection was performed.[53]

The RF algorithm consists of a collection of tree-structured regressors. In general, each tree grows with respect to a random subset of the input dataset independent and identically distributed. The strategy known as Random Input Selection (RIS) is used to generate different trees. The algorithm chooses randomly a subset S with M features from the original set of n features and seeks within S the best feature to split a node of the tree according to some purity measure. Therefore a regression tree is found with M feature subset for each subset S. The final output for a given input data point is calculated using the average of all the regression trees predicted values. The parameters considered during the tuning process of the algorithm are the number of trees (ntree) and the number of features considered on each tree (mrty). The parameter ntree was fixed in 500 while the mrty was exponentially ranged between 2 and 200.

By using RF it is possible to obtain a rank of the most important features used for the deconvolution problem. By default, RF uses the mean decrease impurity and is related to the total decrease in node impurity averaged over all trees The use of RF for feature selection over linear regressors coefficients is preferred because of its capability of dealing with non-linear regression as well as a simple straightforward approach for measuring feature importance.

Prediction method

Once we obtained the training and validation In Silico admixtures we decided to use three classical regression methods to predict the different cell type proportions under the assumption that some features (genes) alone or in combination can act as specific cell subtype biomarkers that can be detected over biologically common or unspecific gene expression patterns. The three methods tested were Support Vector Regression (SVR), Partial Least Squares (PLS), and Penalized Linear regression (GLMNET) but other methods could also be applied. In particular, for each cell line, we trained a model that was fine-tuned using a validation set made out of a set of left out samples for this purpose and the decision of the best performing method was made by the comparison of the mean Pearson's correlation coefficient over all cell lines prediction in the test sets provided in the first three phases of the challenge.

Here we will further explain the SVR approach since it was the best performing submitted model in the final submission.

The results of our tested models for the coarse-grained sub-Challenge are here (https://rpubs.com/harpomaxx/devcon2020-5-newmix) and for the fine-grained sub-Challenge here (https://rpubs.com/harpomaxx/devcon2020-finegrain).

Support Vector Regression (SVR)[54] applies the same support-vectors based approach from Support Vector Machines to form a flexible tube of minimal radius around the estimated function. The points outside the tube suffer a penalization, but those within the tube will not be penalized. One of the advantages of SVR is that it has excellent generalization capability. Also, its computational complexity does not depend on the dimensionality of the input space. SVR is capable of dealing with non-linear data by simply transforming the data into a different feature space by the so-called kernel trick. Since in the deconvolution problem we assumed it can be explained by a non-linear regression function, we decided to apply a Gaussian/radial kernel to the epsilon-SVR variant: Four parameters were considered during the tuning process:

cost: cost of constraints violation

gamma: the inverse of the radius of influence of samples selected by the model as support vectors.

epsilon: Defines a margin of tolerance where no penalty is given to errors.

Nfeatures: the different number of features according to the rank provided by the Random Forest model.

Prediction output

The final model consists of a single sub-model for each cell type, so for instance, in both the coarse and fine grain sub-Challenges we used an SVR model for each of the cell lineages requested, other cell types can be easily added and detected if they are also included in the model. For example, some cell types not included in the challenge were also detected and estimated in our model, like cancer cells or other polymorphonuclear leukocytes.

The input to the prediction resulting model is a vector of 'Rhino' normalized gene expression values (or an expression matrix for multiple samples) and the result is a numeric vector (or matrix) with the estimated cell-type proportion of each cell type in the model. In these regards, it is important to notice that all the cell type models were trained with the same admixture sets, which were bound to contain known purified sample proportions that add up to 1. For instance, one training sample could have 0.2 fibroblast, 0.5 B cells and 0.3 of 'others' cells while another sample could have 0.5 of endothelial cell and 0.5 of cancer cells, which allowed our model to detect different cell type proportions independently one of the other, allowing the possibility of having unseen or unknown cell types in the admixtures without affecting the predicted output and making the predicted proportions comparable across different cell types.

This design should limit the results to have a maximum sum of 1 if there are not unknown cell types present in the mixture but the sum can be less than 1 if otherwise. Of note, no constraints were included in the submitted model but they could be included turning negative prediction to be 0 and rescaling the predictions to 1 if the sum of the predictions is greater than 1, although these alternatives have not been explored thoroughly.

Another important issue to address is that the normalization preprocessing of the training samples and that to perform predictions on new cases is done using the information within each sample independently, which allows the results to be comparable between samples even from different centers, which is of great importance when multi-institution studies are designed or when analyzing isolated samples for the decision-making process in a practical context (i.e. decide treatment in a patient)

These issues must be further explored and specifically addressed by additional research directed at this subject specifically, nevertheless, we believe the rationale of our method is supported by the results obtained using this approach within the current study.

Conclusion/Discussion

In conclusion, we believe our method yields a very flexible, yet robust, approach to the tumor deconvolution problem with several desired features that might be of interest for the community:

- It includes bulk RNA-seq experiments from a wide range of sequencing platforms and different research centers that were successfully used to train well-performing models with a reduced set of samples (843 purified samples in total). In these regards, the same approach could be applied with a bigger number of samples possibly improving the model accuracy or it also could be extended to single-cell RNA-Seq experiments, which might be able to detect 'pure' cell types that could be used for training the model.
- It relies on a within-sample normalization procedure that improves the generalization of our method and unlocks the prediction of cell-type proportions from single samples which could be of great importance in future health applications.
- It allows the possibility to include new cell types not studied here.
- It is able to account for unknown cell types in the sample.
- All the model training can be performed on a desktop computer and does not require exceptional computer resources, making it accessible for use by the community and for further research and development.

As future directions, we would like to test our method in real complex tissue samples like tumor samples to evaluate the performance of our method in a real context. As a limitation, we know that immune, cancer and stromal cells suffer from important changes in their phenotype and gene expression patterns in these complex scenarios and that the isolated samples might not represent the true gene expression patterns displayed by the different cell types when they interact with each other.

Also we would like to further validate our proposed normalization procedure “RHINO”, which we believe could be added to the transcriptomic analysis toolbox.

Authors Statement

M Guerrero-Gimenez designed the overall approach and methodology including the normalization procedure. B Lang and M Guerrero-Gimenez curated the datasets and metadata. B Lang downloaded the data, performed sample quality control, trimming and Kallisto aligning. C Catania aided in the overall design and led the machine learning process of the project performing all the model training, hyperparameter tuning, and testing.

mitten_TDC19 deconvolution method

Summary Sentence

We used a score calculated as the sum of the expression of a selected set of markers for each sample.

Background/Intro

The key idea of the method was to focus on identifying new markers that are predictive of the relative cell type fractions. This was achieved by first identifying bulk and single-cell datasets of pure cell populations. For single-cell data, we also used non-pure populations and extracted pure populations using previously-published methods.[34,35] We then used a Monte Carlo procedure to create random mixtures and identified genes that correlated better with the generated fractions. This took into account the overall distribution of each candidate marker in correlating with the cell fraction.

Methods

Fourteen pure cell type RNA-seq datasets were collected from GEO (**Table S14**). Only control samples, i.e., samples from healthy human donors without treatment, were retained, except for two datasets including samples with breast and colon cancer. Gene expression values were converted to TPM if not in TPM originally. To convert Raw data to TPM, gene length information was obtained from Gencode v31.

We used the above datasets of pure-cell types to create training datasets T_{jk} where $j = 1, ..., m$ are genes and $k = 1, ..., n$are cell types . We then constructed samples of randomly mixed pure cell type populations, $S_j^a = \sum_k T_{jk} R_k^a$ where a is the in silico sample index with fractions R_k^a

. The random fractions R_k^a were drawn from a non-uniform distribution enriched for smaller fractions.

Using the S^a ensemble, we calculated the Pearson correlation coefficient between the concentration of cell type k (i.e., R_k^a) in the M_{jk} ensemble and the expression of genes. We kept the most correlated genes as$+1$ in a signature matrix , and 0 otherwise.

In the deconvolution, each dataset X_{ij}, where i is a sample and j is a gene, was converted to a linear scale. Each sample$i$ was assigned a score for each cell type k according to $\beta_{ik} = \sum_j X_{ij} M_{jk}$, which was submitted as a prediction.

The same method was used for the two sub-Challenges.

Conclusion/Discussion

We focused on the selection of markers and used a basic algorithmic approach for the predictions. The fact that we still ended in the top three performing groups indicates that the selection of markers is probably the most important aspect of this challenge.

Authors Statement

SD, TB, and CP designed the algorithms. SD and TB performed calculations. TB submitted models to challenge.

Biogem deconvolution method

Summary Sentence

The method uses robust linear modeling from the R package MASS together with a signature matrix made of harmonized gene expression data normalized by mRNA abundance.

Background/Intro

The approach used in this challenge is based on work that was previously published.[36] The feature selection of the approach is done through differential expression and filtering for specific and low-noisy genes. The deconvolution method is robust linear model, which is robust to collinearity and outliers. The signature matrix was normalized by mRNA abundance to take into account the different mRNA yields of the various immune cell types.

The main variation in respect to the published paper[36] is the utilization of a large harmonized dataset for the generation of the signature matrix. The benefit of this approach over other popular approaches is that it allows to achieve top level results without the utilization of more complex algorithms based on machine and deep learning.

Methods

Data collection and preprocessing

The publicly available data used to generate the signature matrix were taken from the following datasets: PRJEB14751, PRJEB36933, PRJNA218851, PRJNA272556, PRJNA327180, PRJNA338944, PRJNA352224, PRJNA418779, PRJNA430418, PRJNA449980, PRJNA471906, PRJNA483877, PRJNA484735, PRJNA489270, PRJNA490870, PRJNA495625, PRJNA540256, PRJNA559359, PRJNA562113, PRJNA598222 (**Table S15**).

The samples were selected independently of the condition or tissue they were derived from and for some datasets only a subset of samples were downloaded and processed for this challenge.

Dataset	Cell type	Condition	Platform	Submissions
PRJEB14751	Endothelial cells	HUVEC cell line	Illumina HiSeq 2500	1, 2, 3
PRJEB36933	Tregs	healthy donors	Illumina HiSeq 4000	2, 3
PRJNA218851	CRC	primary tumor	Illumina HiSeq 2000	1, 2, 3
PRJNA272556	CRC	cell lines	Illumina HiSeq 1000	1, 2, 3
PRJNA327180	Fibroblasts	cell lines	Illumina HiSeq 2000	1, 2, 3
PRJNA338944	Fibroblasts	human cancer	Illumina HiSeq 2500	1, 2, 3
PRJNA352224	mDCs	healthy stimulated and non	Illumina HiSeq 2000	2, 3
PRJNA418779	various immune cells	healthy	Illumina HiSeq 2000	1, 2, 3
PRJNA430418	Endothelial cells	stimulated and non	Illumina HiSeq 2500	1, 2, 3
PRJNA449980	Macrophages	healthy simulated and non	Illumina HiSeq 2500	3
PRJNA471906	Endothelial cells	healthy	Illumina HiSeq 2500	1, 2, 3
PRJNA483877	Monocytes	breast cancer	Illumina HiSeq 2500	1, 2
PRJNA484735	Various immune cells	healthy stimulated and non	Illumina NovaSeq 6000	1, 2, 3
PRJNA489270	BRCA	cell lines	Illumina HiSeq 2000	1, 2, 3
PRJNA490870	CRC	primary tumor	Illumina HiSeq 4000	1, 2, 3
PRJNA495625	BRCA	cell lines	Illumina HiSeq 2000	1, 2, 3

PRJNA540256	Macrophages	from endometrium	Illumina HiSeq 4000	1, 2
PRJNA559359	Macrophages	healthy	Illumina HiSeq 2500	3
PRJNA562113	CRC	primary and metastatic tumor	Illumina HiSeq 2500	1, 2, 3
PRJNA598222	BRCA	primary tumor	Illumina HiSeq 2000	1, 2, 3

The fastq files were downloaded from the ENA archive and kallisto (version 0.46.2)[36,49] was used to obtain gene expression data. The pseudo-alignment was done against the GENCODE transcriptome (release 33, genome assembly GRCh38).

Feature selection

The first feature selection step consists in performing differential expression analysis between each cell type and remaining ones. This was done using the voom and limma methods.[51]

The second feature selection step consists in filtering out genes that are not beneficial for deconvolution. These are:

1. highly expressed genes -- any gene that has a TPM value > 3000 in at least one sample;
2. low expressed genes -- any gene with a summed up expression value < 5 across all samples;
3. specificity -- any gene *g* from the differential analysis between cell type *c* and the remaining cell types that has an effect size < 0.1 between its median $\log_2$ values in *c* and the cell type *c*' with the second highest expression (after *c*) of *g*.

The third feature selection step consists of a hard threshold filter to avoid the over-representation of cell types with many remaining genes. Hence, no more than 70 genes and 90 genes were kept for each cell type for the Coarse and Fine sub-Challenges, respectively.

Signature matrix

For the signature matrix, for each cell type the median value of counts per million (CPM) was generated. Next, the values were scaled by a factor that accounts for mRNA abundance.[36] The previously-published[36] scaling factors were only for cell types from Peripheral Blood Mononuclear Cells (PBMCs). Hence, the scaling factors for tumor cells, endothelial cells and fibroblasts were conventionally set to 1.

Prediction method

The prediction method used for deconvolution is the Robust Linear Model, which was implemented using the rlm function from the R package MASS. The method is robust to noise and collinearity and its performance has been shown to be comparable to support vector regression as used in CIBERSORT.[19,36]

The method expects CPM as input and the code includes a step that checks that the sum of every sample of the input data is 10^6.

Prediction output

The values generated by the robust linear model are interpretable as fractions.

The negative values are dealt with in the following two steps: 1) any value lower than -2 is set as -2; 2) scale values so that the minimum value is 0.

The output values can be used for both comparison between and within samples.

To allow comparison between samples, the approach does not implement any constraints. In this way, if the sample has an amount of unknown content that is not predicted by the method, there is no inflation of the fractions due to a method constraint.

To allow comparison within samples, the approach implements normalization of mRNA abundance to the signature matrix.[36] Normalization for mRNA abundance takes into consideration the higher mRNA content of certain cell types (such as monocytes) and thereby avoids inflation of their predicted fractions.

Therefore, this method allows the deconvolution of absolute abundance of immune cell types present in the mixture sample.

Variations in 2nd and 3rd submission

The variation included in the 2nd and 3rd submission do not affect the method, but only the implementation of batch effect correction and the inclusion and exclusion of various datasets.

For the 2nd and 3rd submission, the outlier samples to exclude were evaluated visually through a PCA analysis and were excluded from subsequent analyses. The data were also corrected for batch effects using Combat.

Fig. S3 shows that after batch effect correction of some immune cell datasets, the samples do not cluster according to dataset anymore.

Regarding specific datasets that allowed us to achieve better performance, we noticed that we could improve the scores obtained on T regulatory cells only after adding the dataset PRJEB36933. Moreover, to improve the results obtained for macrophages, we excluded the samples from the datasets PRJNA483877 and PRJNA540256, and we added samples from the datasets PRJNA449980 and PRJNA559359.

The table below shows the number of samples used for each cell type in the coarse-grained sub-Challenge:

Cell Type	Submission 1	Submission 2	Submission 3
B.cells	38	38	38
BRCA	10	10	10
CD4.T.cells	88	100	98
CD8.T.cells	51	51	51
CRC	20	20	20
endothelial.cells	19	19	19
fibroblasts	10	10	10
monocytic.lineage	58	61	53
neutrophils	4	4	4
NK.cells	15	15	15
SUM	313	328	318

The table below shows the number of samples used for each cell type in the fine-grained sub-Challenge:

Cell Type	Submission 1	Submission 2	Submission 3
BRCA	10	10	10
CRC	20	20	20
endothelial.cells	19	19	19
fibroblasts	10	10	10
macrophages	22	22	20
memory.B.cells	19	19	19
memory.CD4.T.cells	40	40	40
memory.CD8.T.cells	27	27	27
monocytes	29	29	18

myeloid.dendritic.cells	7	10	15
naive.B.cells	12	12	12
naive.CD4.T.cells	4	4	4
naive.CD8.T.cells	12	12	12
neutrophils	4	4	4
NK.cells	15	15	15
regulatory.T.cells	23	33	33
SUM	273	286	278

Through these variations, a substantial improvement of the approach was achieved for the fine challenge, although not much for the coarse challenge.

Variations between sub-Challenge 1 (coarse-grained sub-Challenge) and sub-Challenge 2 (fine-grained sub-Challenge)

The data processing and methodology used for the fine-grained sub-Challenge are identical to the ones for the coarse-grained sub-Challenge.

There are only these two consideration to make:

- regarding the normalization for mRNA abundance, for macrophages we used the same scaling factor calculated for classical monocytes as they belong to the same lineage.
- for the hard threshold of the number of genes to include for cell type, we kept no more than 70 genes for the coarse-grained sub-Challenge and no more than 90 genes for the fine-grained sub-Challenge.

Conclusion/Discussion

In conclusion, we believe we reached a good deconvolution performance using robust linear modeling on a signature matrix generated from a large harmonized dataset. We believe that more datasets are necessary to assess the variability across certain cell types and to establish which isolation strategies or biological conditions increase such variability. This is especially the case if one is trying to perform deconvolution on subtypes of the same lineage, such as memory T cell subtypes.

Authors Statement

GM collected, processed the samples, and developed the methodology. FPC contributed to data collection and data analysis. MC contributed in the interpretation of the results.

IZI deconvolution method

The method aims to utilize a probabilistic description of the cell-type specific transcription composition. We gathered thousands of replicates for gene expression measurements per cell type from 75,720 different samples in hundreds of publicly available GEO datasets. The replicates include single cell RNA-seq, bulk RNA-seq and bulk expression microarray data. To harmonize the datasets, we selected genes and samples such that the amount of recovered samples and genes that are shared between all samples are maximized. This resulted in a total of 21,992 harmonized RNA-expression vectors for the coarse-grained sub-Challenge and 13,317 for the fine-grained sub-Challenge. We then applied a pipeline of transformations and dimensional reduction that are both reversible and differentiable to the expression data. The goal was to find transcriptome representations such that the distribution per cell type can be described through normal distributions in a latent space where the Wasserstein distance between pairs of cell types is maximized. We assumed that the result of the RNA-seq experiment and counting process is a random process that is influenced by the composition of transcripts in the sequenced sample. Specifically, we assume that finding a read for a given sample is a Bernoulli experiment with a fixed probability for each sample. Bayes theorem gives rise to a probabilistic description of the transcriptome composition based on the raw observed counts through the Dirichlet distribution. The decomposition space can be transformed with an isometric log-ratio transformation to achieve approximate normal distributions. Through scaling such that the average covariance per cell type is the unit matrix and applying principal component analysis on the cell-type means, we found a dimensional reduction that approximately maximizes the Wasserstein distance between the cell-type specific distributions. A justification of this procedure has been previously described.[55] The resulting distributions in dimensionally reduced space serve as characterization of the respective cell types and define a distribution of transcriptome compositions in expression space. We used these characterizations to define a Bayesian model that mixes the cell-type-specific transcriptomes with Dirichlet distributed weights to describe the expression profile of the mixed samples that ought to be deconvolved. Through the application of automatic variational differential inference (ADVI)[56] in the pymc3 package[57] we performed Bayesian inference without exceeding the computation resource constraints and use the posterior mean of the mixture weights as an output for the deconvolution. The algorithm aims to make very little assumptions about the data and could potentially achieve absolute cell mass quantifications of individual samples without knowing the context of the cohort. The main limitations are the approximation of the posterior through ADVI and inaccuracies of the cell type characterization through mislabeling or lack of transcriptome samples, e.g., we used only 42 memory CD8 T-cells.

Supplemental Figures

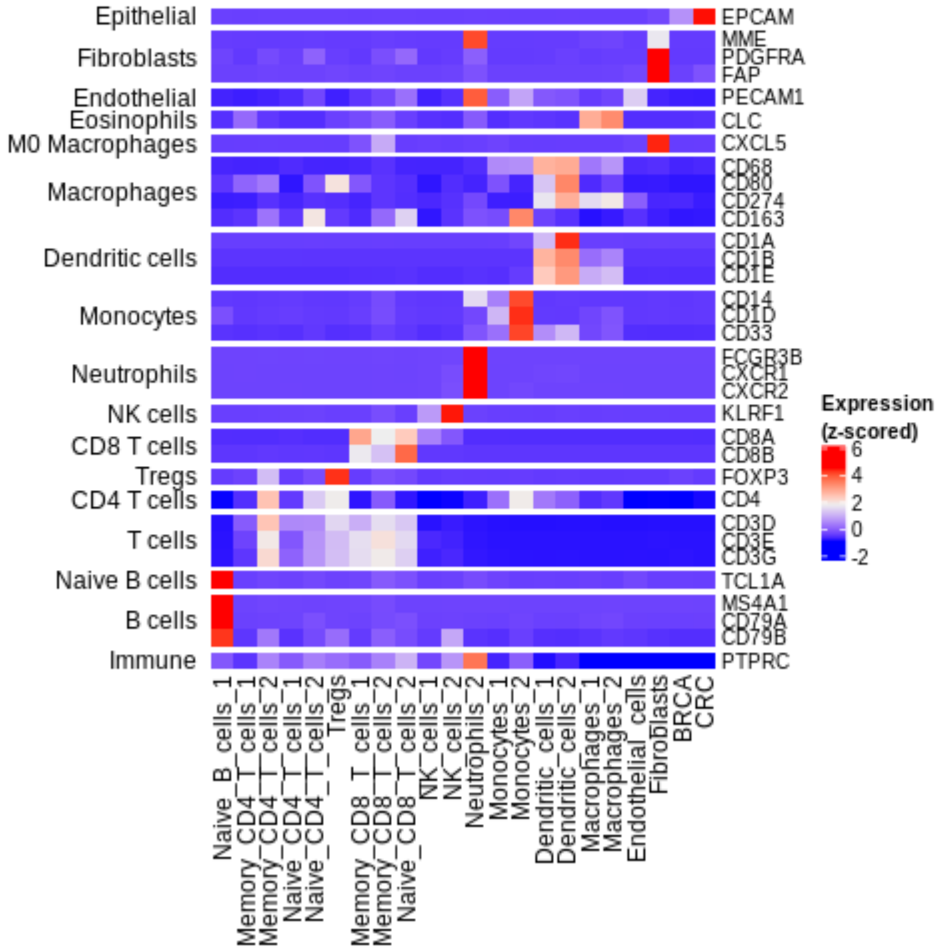


Fig. S1: Marker gene expression within purified populations. Expression of markers (right axis) group by corresponding cell population (left axis) within each purified sample (columns).

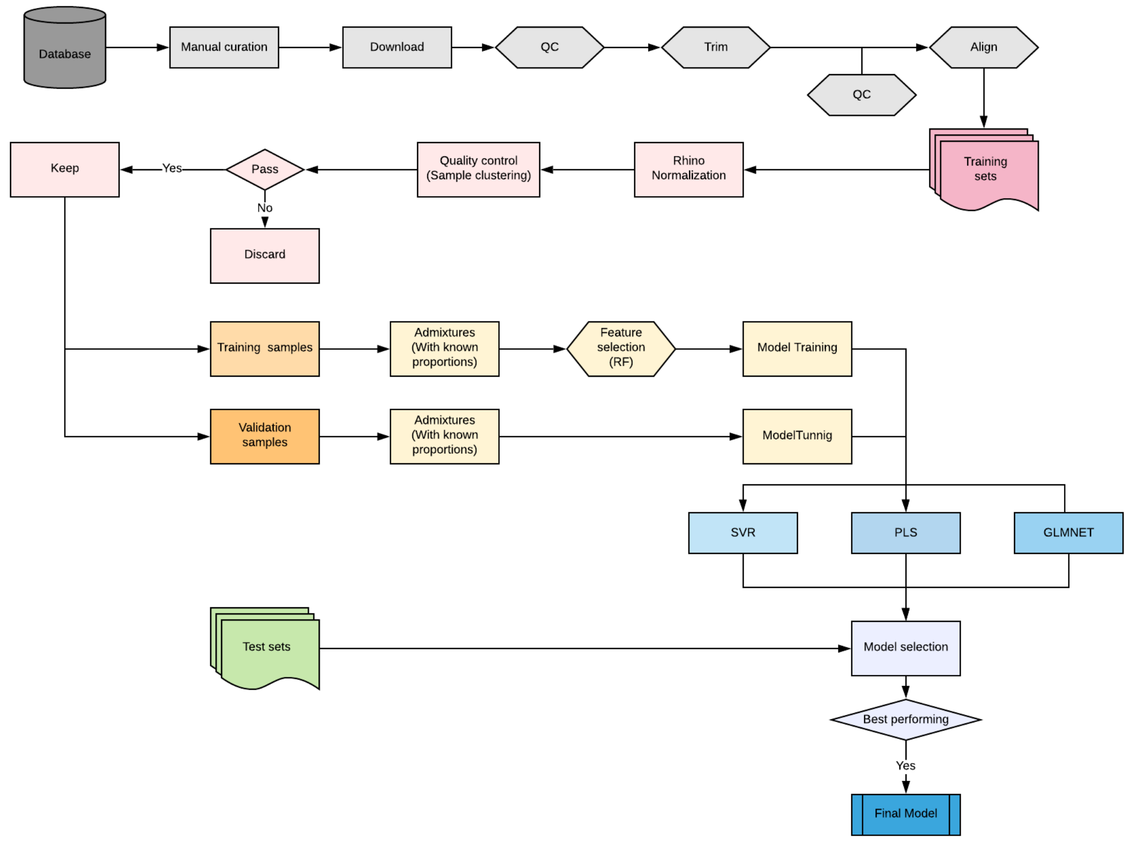


Fig. S2: Workflow of DA_505 deconvolution method.

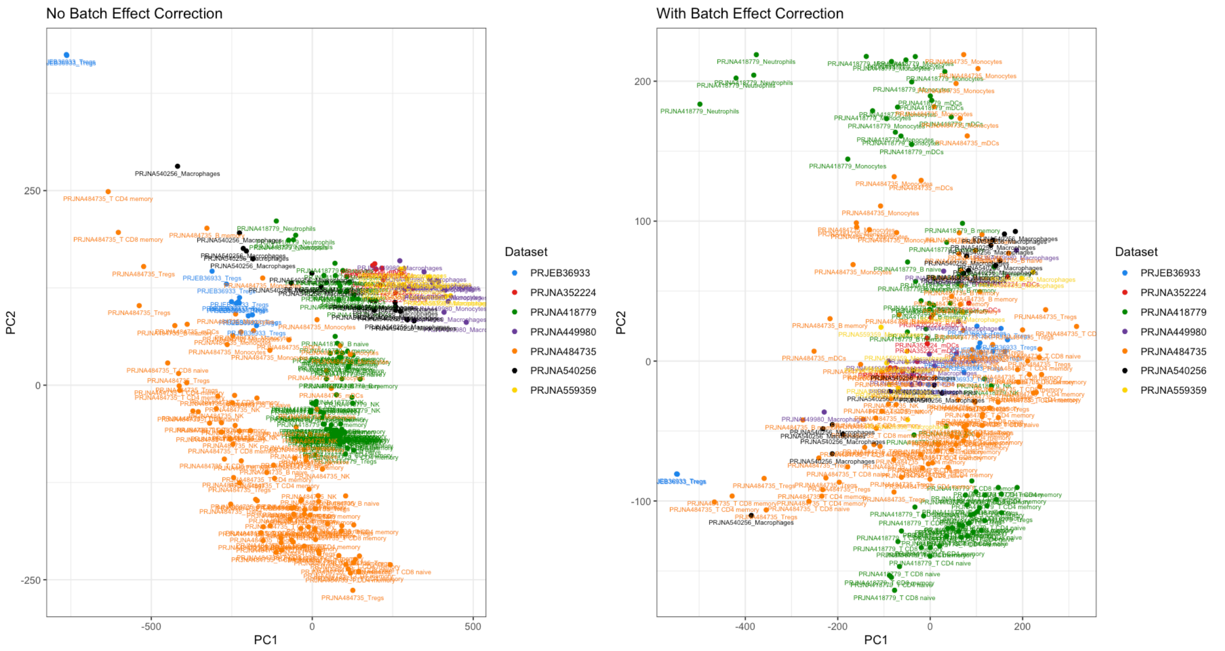


Fig. S3: Batch correction of data used to train Biogem deconvolution method.

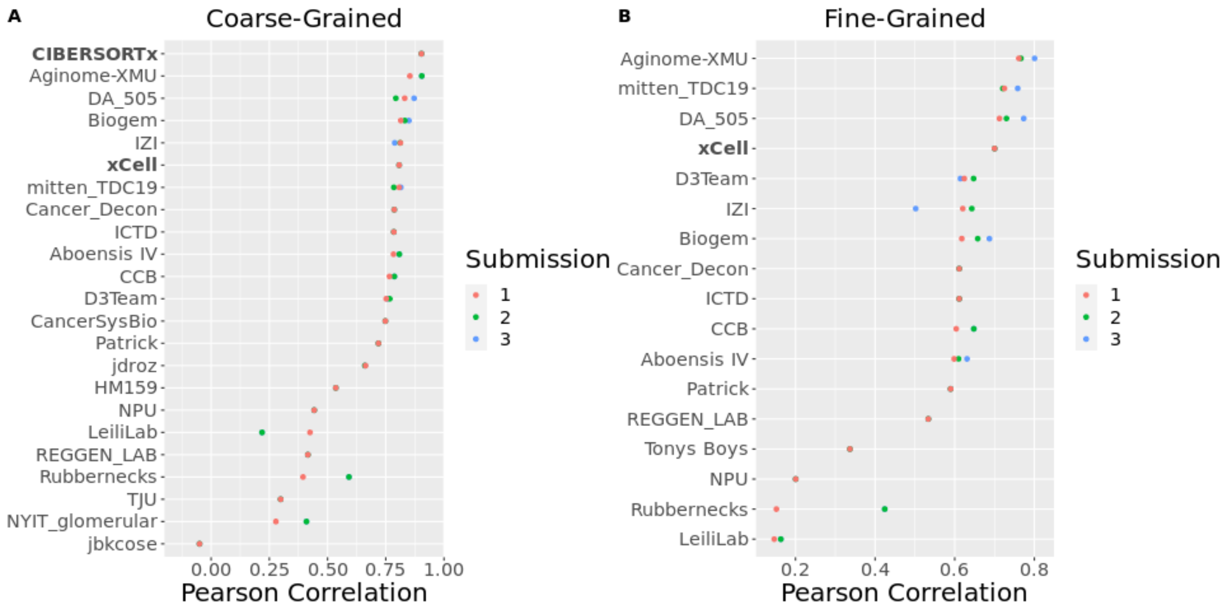


Fig. S4: Aggregate primary, Pearson-based score of participant methods over submissions and of comparator methods. (A, B) Aggregate Pearson-based score of methods in (A) coarse- and (B) fine-grained sub-Challenges over (*n*=1,000) bootstraps (Methods). Comparator methods (bold) are shown only if their published reference signatures include all cell types in each respective sub-Challenge: CIBERSORTx (coarse-grained only) and xCell.

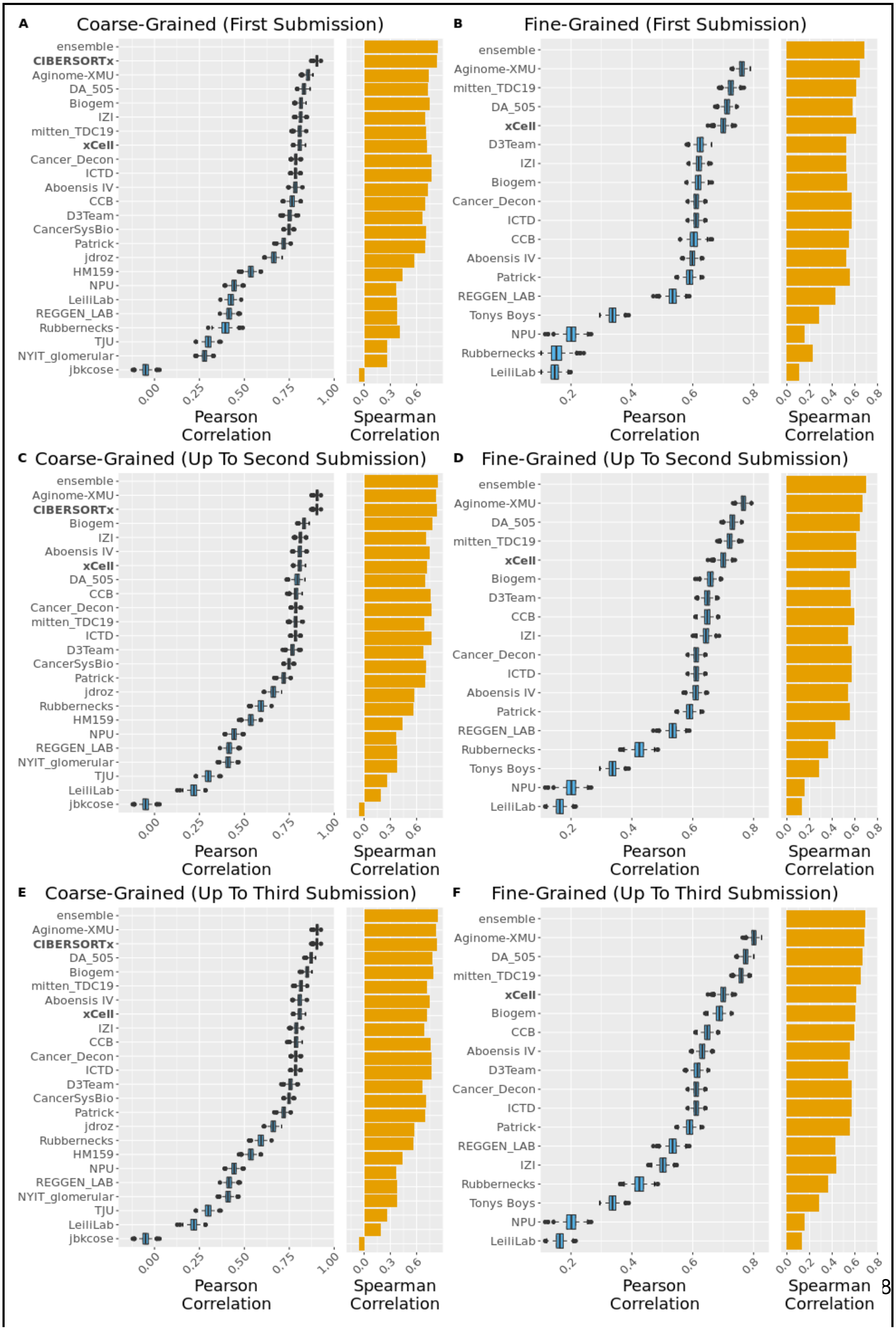

A
Coarse-Grained (First Submission)
ensemble
CIBERSORTx
Aginome-XMU
DA_505
Biogem
IZI
mitten_TDC19
xCell
Cancer_Decon
ICTD
Aboensis IV
CCB
D3Team
CancerSysBio
Patrick
jdroz
HM159
NPU
LeiliLab
REGGEN_LAB
Rubbernecks
TJU
NYIT_glomerular
jbkcose
0.00
0.25
0.50
0.75
1.00
Pearson Correlation
0.0
0.3
0.6
Spearman Correlation
B
Fine-Grained (First Submission)
ensemble
Aginome-XMU
mitten_TDC19
DA_505
xCell
D3Team
IZI
Biogem
Cancer_Decon
ICTD
CCB
Aboensis IV
Patrick
REGGEN_LAB
Tonys Boys
NPU
Rubbernecks
LeiliLab
0.2
0.4
0.6
0.8
Pearson Correlation
0.0
0.2
0.4
0.6
0.8
Spearman Correlation
C
Coarse-Grained (Up To Second Submission)
ensemble
Aginome-XMU
CIBERSORTx
Biogem
IZI
Aboensis IV
xCell
DA_505
CCB
Cancer_Decon
mitten_TDC19
ICTD
D3Team
CancerSysBio
Patrick
jdroz
Rubbernecks
HM159
NPU
REGGEN_LAB
NYIT_glomerular
TJU
LeiliLab
jbkcose
0.00
0.25
0.50
0.75
1.00
Pearson Correlation
0.0
0.3
0.6
Spearman Correlation
D
Fine-Grained (Up To Second Submission)
ensemble
Aginome-XMU
DA_505
mitten_TDC19
xCell
Biogem
D3Team
CCB
IZI
Cancer_Decon
ICTD
Aboensis IV
Patrick
REGGEN_LAB
Rubbernecks
Tonys Boys
NPU
LeiliLab
0.2
0.4
0.6
0.8
Pearson Correlation
0.0
0.2
0.4
0.6
0.8
Spearman Correlation
E
Coarse-Grained (Up To Third Submission)
ensemble
Aginome-XMU
CIBERSORTx
DA_505
Biogem
mitten_TDC19
Aboensis IV
xCell
IZI
CCB
Cancer_Decon
ICTD
D3Team
CancerSysBio
Patrick
jdroz
Rubbernecks
HM159
NPU
REGGEN_LAB
NYIT_glomerular
TJU
LeiliLab
jbkcose
0.00
0.25
0.50
0.75
1.00
Pearson Correlation
0.0
0.3
0.6
Spearman Correlation
F
Fine-Grained (Up To Third Submission)
ensemble
Aginome-XMU
DA_505
mitten_TDC19
xCell
Biogem
CCB
Aboensis IV
D3Team
Cancer_Decon
ICTD
Patrick
REGGEN_LAB
IZI
Rubbernecks
Tonys Boys
NPU
LeiliLab
0.2
0.4
0.6
0.8
Pearson Correlation
0.0
0.2
0.4
0.6
0.8
Spearman Correlation

Fig. S5: Aggregate method performance over rounds. Aggregate score (primary metric: Pearson correlation; secondary metric: Spearman correlation) of participant and comparator methods in (A, C, E) coarse- and (B, D, F) fine-grained sub-Challenges over (*n*=1,000) bootstraps (Methods). Scores reported from the (A, B) first submission, (C, D) second submission (or latest submission up to the second, if less than two submissions), or (E, F) third submission (or latest submission up to the third, if less than three submissions). Reported Spearman-based score is median over bootstraps. Comparator methods (bold) are shown only if their published reference signatures include all cell types in each respective sub-Challenge: CIBERSORTx (coarse-grained only) and xCell.

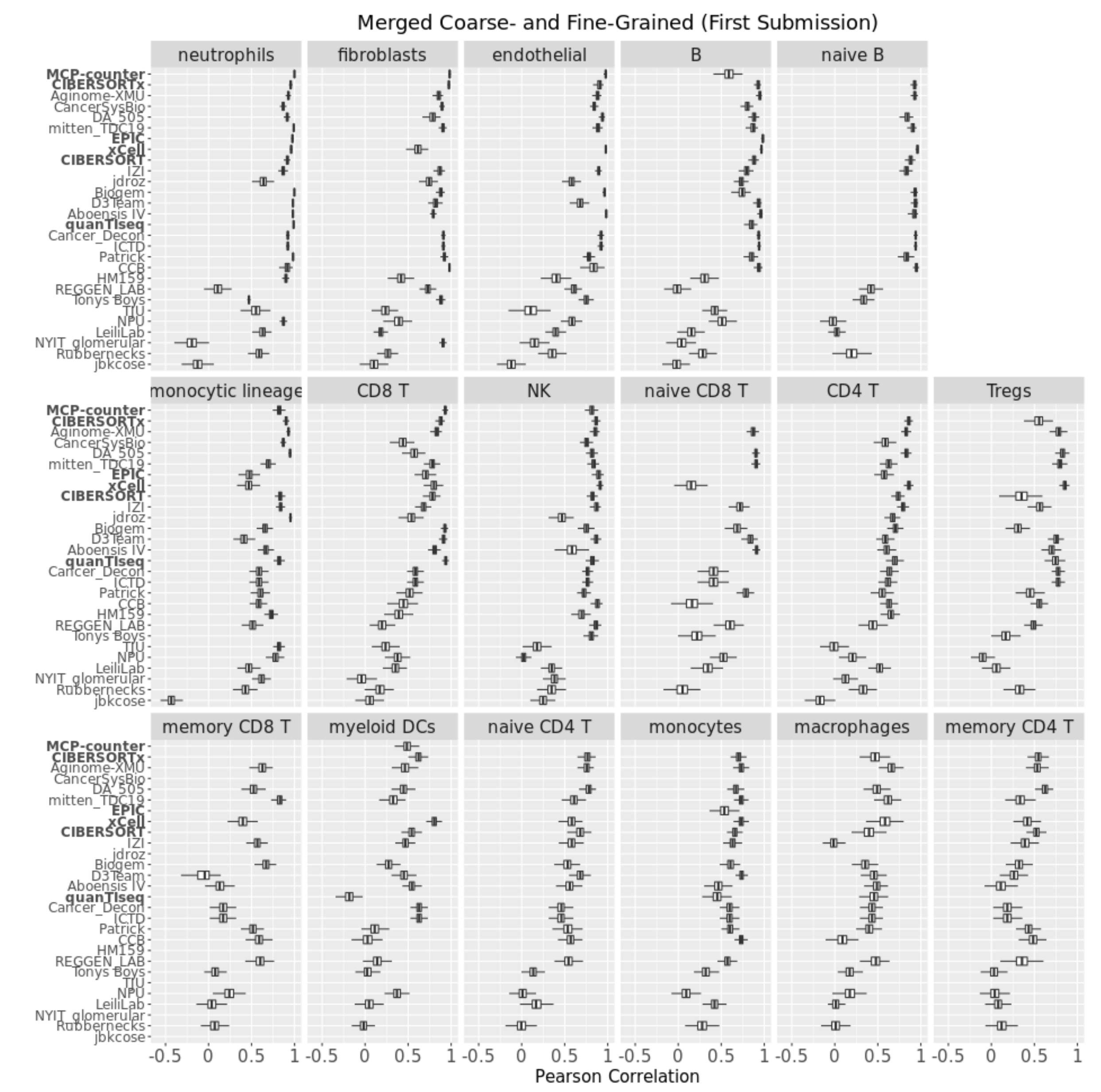


Fig. S6: Distribution of per-cell type method performance from first submission merged across coarse- and fine-grained sub-Challenges. Performance (Pearson correlation; x axis) of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods), computed as average over validation datasets and subsequently over coarse- and fine-grained sub-Challenges for cell types occurring in both. Blank row indicates cell type not reported by the corresponding method.

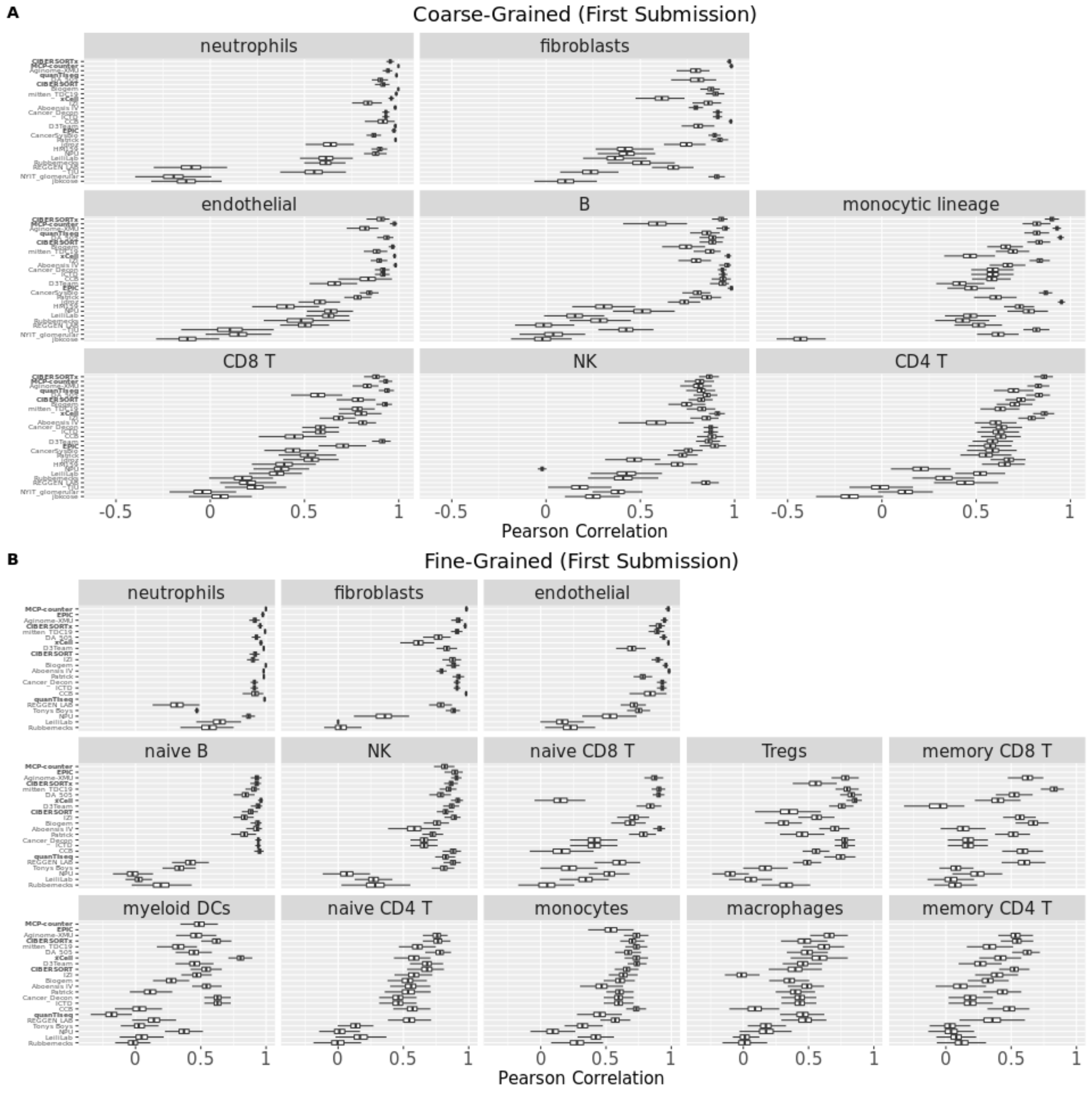
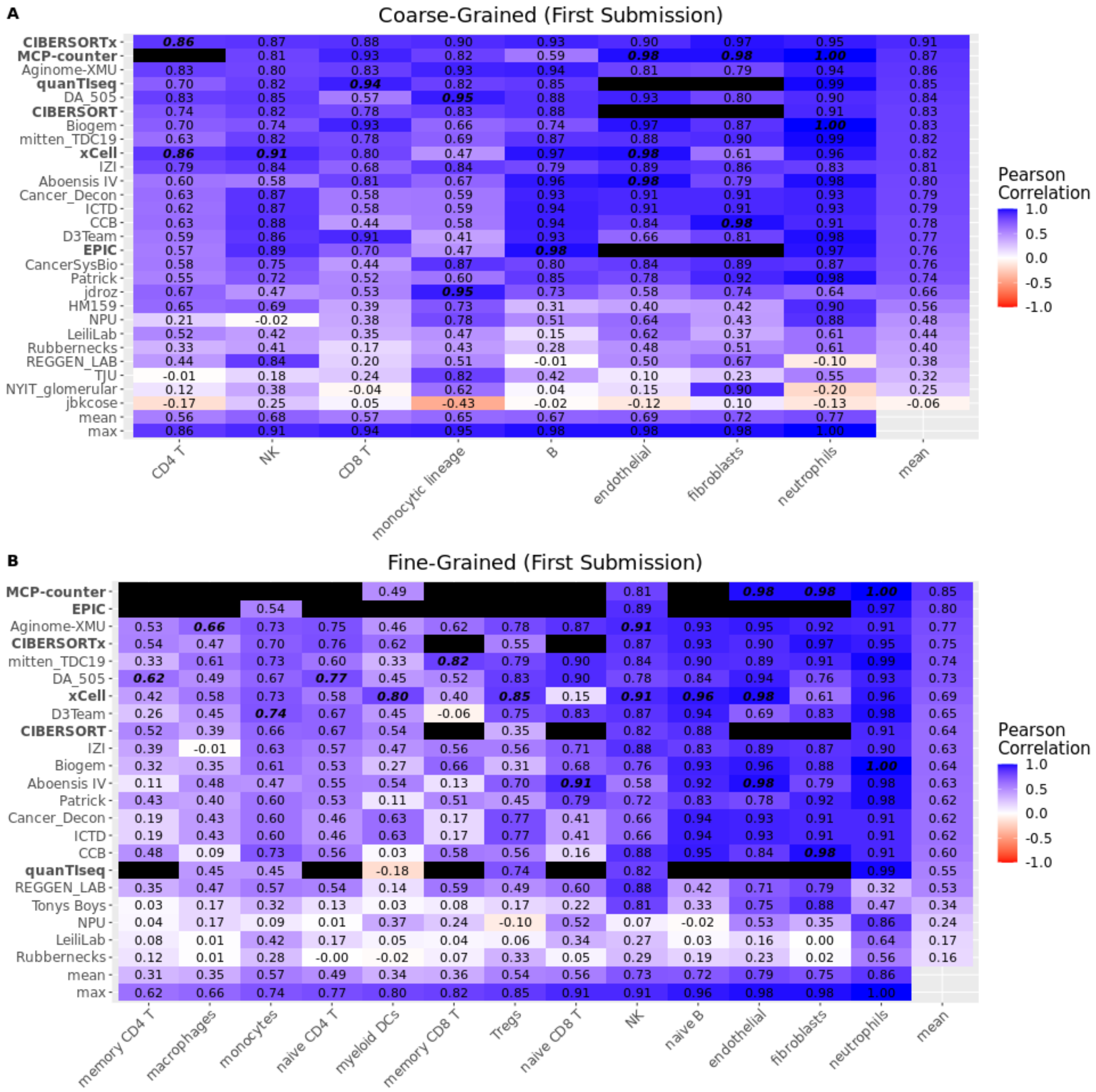


Fig. S7: Distribution of per-cell type method performance from first submission stratified by sub-Challenge. Performance (Pearson correlation; x axis) in (A) coarse- and (B) fine-grained sub-Challenges of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000;

Methods). Blank row indicates cell type not reported by the corresponding method.



A Coarse-Grained (First Submission)

	CD4 T	NK	CD8 T	monocytic lineage	B	endothelial	fibroblasts	neutrophils	mean
CIBERSORTx	***0.86***	0.87	0.88	0.90	0.93	0.90	0.97	0.95	0.91
MCP-counter		0.81	0.93	0.82	0.59	***0.98***	***0.98***	***1.00***	0.87
Aginome-XMU	0.83	0.80	0.83	0.93	0.94	0.81	0.79	0.94	0.86
quanTIseq	0.70	0.82	***0.94***	0.82	0.85			0.99	0.85
DA_505	0.83	0.85	0.57	***0.95***	0.88	0.93	0.80	0.90	0.84
CIBERSORT	0.74	0.82	0.78	0.83	0.88			0.91	0.83
Biogem	0.70	0.74	0.93	0.66	0.74	0.97	0.87	***1.00***	0.83
mitten_TDC19	0.63	0.82	0.78	0.69	0.87	0.88	0.90	0.99	0.82
xCell	***0.86***	***0.91***	0.80	0.47	0.97	***0.98***	0.61	0.96	0.82
IZI	0.79	0.84	0.68	0.84	0.79	0.89	0.86	0.83	0.81
Aboensis IV	0.60	0.58	0.81	0.67	0.96	***0.98***	0.79	0.98	0.80
Cancer_Decon	0.63	0.87	0.58	0.59	0.93	0.91	0.91	0.93	0.79
ICTD	0.62	0.87	0.58	0.59	0.94	0.91	0.91	0.93	0.79
CCB	0.63	0.88	0.44	0.58	0.94	0.84	***0.98***	0.91	0.78
D3Team	0.59	0.86	0.91	0.41	0.93	0.66	0.81	0.98	0.77
EPIC	0.57	0.89	0.70	0.47	***0.98***			0.97	0.76
CancerSysBio	0.58	0.75	0.44	0.87	0.80	0.84	0.89	0.87	0.76
Patrick	0.55	0.72	0.52	0.60	0.85	0.78	0.92	0.98	0.74
jdroz	0.67	0.47	0.53	***0.95***	0.73	0.58	0.74	0.64	0.66
HM159	0.65	0.69	0.39	0.73	0.31	0.40	0.42	0.90	0.56
NPU	0.21	-0.02	0.38	0.78	0.51	0.64	0.43	0.88	0.48
LeiliLab	0.52	0.42	0.35	0.47	0.15	0.62	0.37	0.61	0.44
Rubbernecks	0.33	0.41	0.17	0.43	0.28	0.48	0.51	0.61	0.40
REGGEN_LAB	0.44	0.84	0.20	0.51	-0.01	0.50	0.67	-0.10	0.38
TJU	-0.01	0.18	0.24	0.82	0.42	0.10	0.23	0.55	0.32
NYIT_glomerular	0.12	0.38	-0.04	0.62	0.04	0.15	0.90	-0.20	0.25
jbkcose	-0.17	0.25	0.05	-0.43	-0.02	-0.12	0.10	-0.13	-0.06
mean	0.56	0.68	0.57	0.65	0.67	0.69	0.72	0.77	
max	0.86	0.91	0.94	0.95	0.98	0.98	0.98	1.00	

B Fine-Grained (First Submission)

	memory CD4 T	macrophages	monocytes	naive CD4 T	myeloid DCs	memory CD8 T	Tregs	naive CD8 T	NK	naive B	endothelial	fibroblasts	neutrophils	mean
MCP-counter					0.49				0.81		***0.98***	***0.98***	***1.00***	0.85
EPIC			0.54						0.89				0.97	0.80
Aginome-XMU	0.53	***0.66***	0.73	0.75	0.46	0.62	0.78	0.87	***0.91***	0.93	0.95	0.92	0.91	0.77
CIBERSORTx	0.54	0.47	0.70	0.76	0.62		0.55		0.87	0.93	0.90	0.97	0.95	0.75
mitten_TDC19	0.33	0.61	0.73	0.60	0.33	***0.82***	0.79	0.90	0.84	0.90	0.89	0.91	0.99	0.74
DA_505	***0.62***	0.49	0.67	***0.77***	0.45	0.52	0.83	0.90	0.78	0.84	0.94	0.76	0.93	0.73
xCell	0.42	0.58	0.73	0.58	***0.80***	0.40	***0.85***	0.15	***0.91***	***0.96***	***0.98***	0.61	0.96	0.69
D3Team	0.26	0.45	***0.74***	0.67	0.45	-0.06	0.75	0.83	0.87	0.94	0.69	0.83	0.98	0.65
CIBERSORT	0.52	0.39	0.66	0.67	0.54		0.35		0.82	0.88			0.91	0.64
IZI	0.39	-0.01	0.63	0.57	0.47	0.56	0.56	0.71	0.88	0.83	0.89	0.87	0.90	0.63
Biogem	0.32	0.35	0.61	0.53	0.27	0.66	0.31	0.68	0.76	0.93	0.96	0.88	***1.00***	0.64
Aboensis IV	0.11	0.48	0.47	0.55	0.54	0.13	0.70	***0.91***	0.58	0.92	***0.98***	0.79	0.98	0.63
Patrick	0.43	0.40	0.60	0.53	0.11	0.51	0.45	0.79	0.72	0.83	0.78	0.92	0.98	0.62
Cancer_Decon	0.19	0.43	0.60	0.46	0.63	0.17	0.77	0.41	0.66	0.94	0.93	0.91	0.91	0.62
ICTD	0.19	0.43	0.60	0.46	0.63	0.17	0.77	0.41	0.66	0.94	0.93	0.91	0.91	0.62
CCB	0.48	0.09	0.73	0.56	0.03	0.58	0.56	0.16	0.88	0.95	0.84	***0.98***	0.91	0.60
quanTIseq		0.45	0.45		-0.18		0.74		0.82				0.99	0.55
REGGEN_LAB	0.35	0.47	0.57	0.54	0.14	0.59	0.49	0.60	0.88	0.42	0.71	0.79	0.32	0.53
Tonys Boys	0.03	0.17	0.32	0.13	0.03	0.08	0.17	0.22	0.81	0.33	0.75	0.88	0.47	0.34
NPU	0.04	0.17	0.09	0.01	0.37	0.24	-0.10	0.52	0.07	-0.02	0.53	0.35	0.86	0.24
LeiliLab	0.08	0.01	0.42	0.17	0.05	0.04	0.06	0.34	0.27	0.03	0.16	0.00	0.64	0.17
Rubbernecks	0.12	0.01	0.28	-0.00	-0.02	0.07	0.33	0.05	0.29	0.19	0.23	0.02	0.56	0.16
mean	0.31	0.35	0.57	0.49	0.34	0.36	0.54	0.56	0.73	0.72	0.79	0.75	0.86	
max	0.62	0.66	0.74	0.77	0.80	0.82	0.85	0.91	0.91	0.96	0.98	0.98	1.00	

Fig. S8: Per-cell type method performance from first submission stratified by sub-Challenge. Pearson correlation of method (left axis) prediction versus known proportion from admixture for each cell type (bottom axis) in (A) coarse- or (B) fine-grained sub-Challenge. Pearson correlation is first averaged over validation dataset and then (*n*=1,000) over bootstraps (Methods). Black entry indicates cell type not predicted by corresponding method. Bottom two rows (“mean” and "max") are the mean and maximum correlation, respectively, for corresponding cell type across methods. Rightmost column ("mean") is mean correlation for

corresponding method across predicted cell types. Highest correlation for each cell type highlighted in bold italics. Comparator methods in bold.

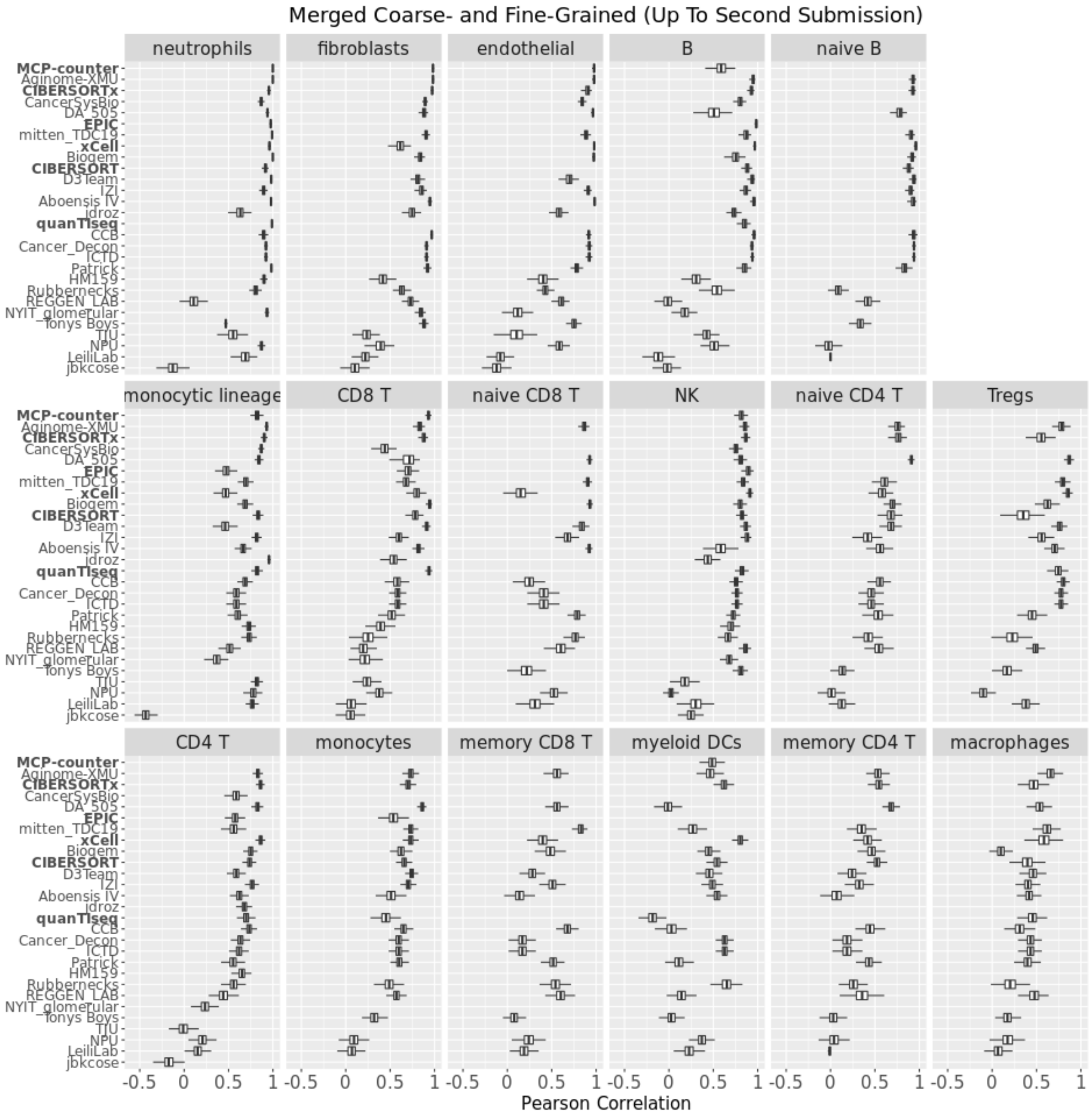


Fig. S9: Distribution of per-cell type method performance from second submission merged across coarse- and fine-grained sub-Challenges. Performance (Pearson correlation; x axis) of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods), computed as average over validation datasets and subsequently over coarse- and fine-grained sub-Challenges for cell types occurring in both. Results from latest submission up to the second, if less than two submissions for corresponding method. Blank row indicates cell type not reported by the corresponding method.

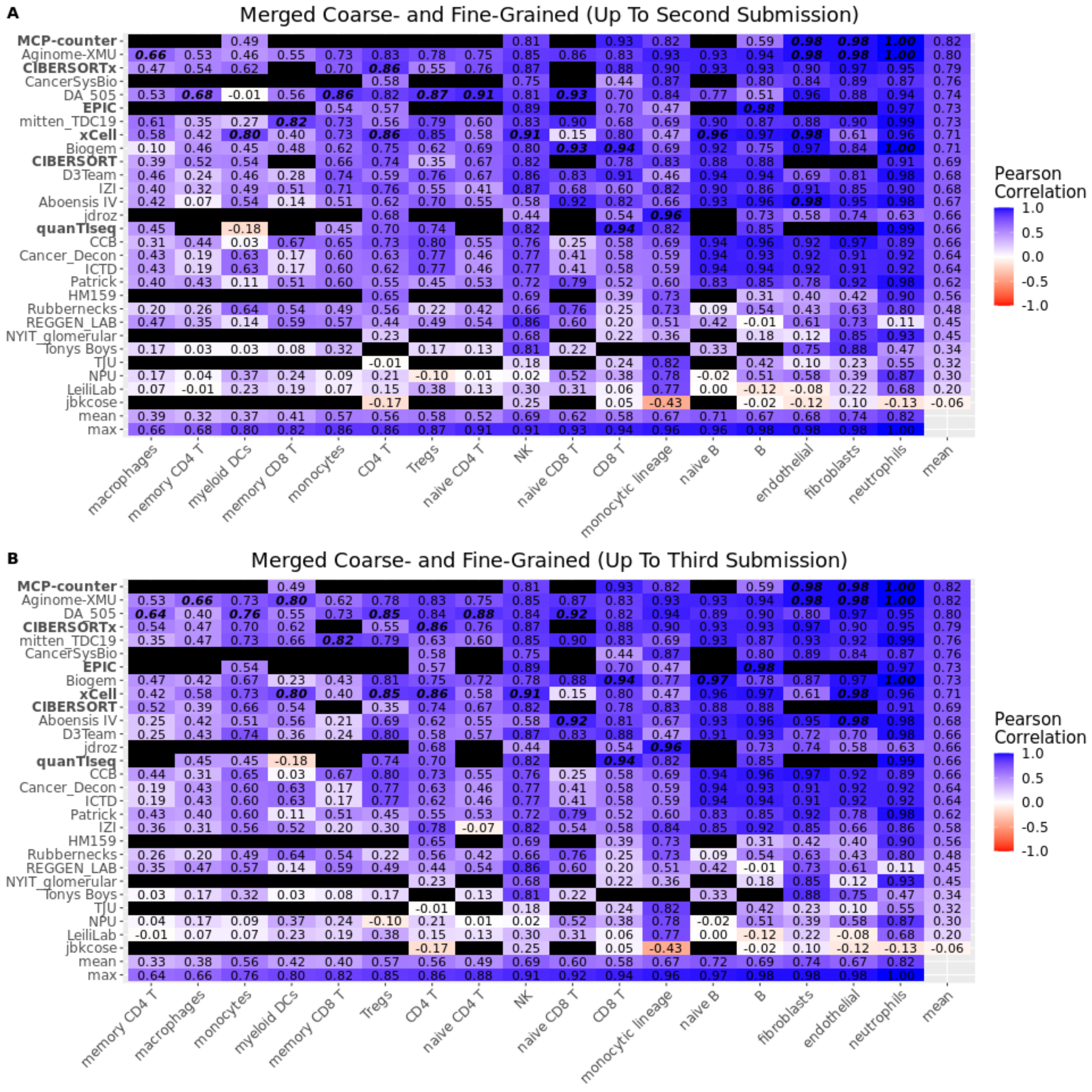


Fig. S10: Per-cell type method performance from second and third submissions merged across coarse- and fine-grained sub-Challenges. Pearson correlation of method (left axis) prediction versus known proportion from admixture for each cell type (bottom axis) for (A) second or (B) third submission. Pearson correlation is first averaged over validation dataset and then (*n*=1,000) over bootstraps (Methods). Results from latest submission up to (A) second or (B) third submission for those methods with fewer than two or three submissions, respectively. Black entry indicates cell type not predicted by corresponding method. Bottom two rows ("mean" and "max") are the mean and maximum correlation, respectively, for corresponding cell type across methods. Rightmost column ("mean") is mean correlation for corresponding method across predicted cell types. Highest correlation for each cell type

highlighted in bold italics. Comparator methods in bold.

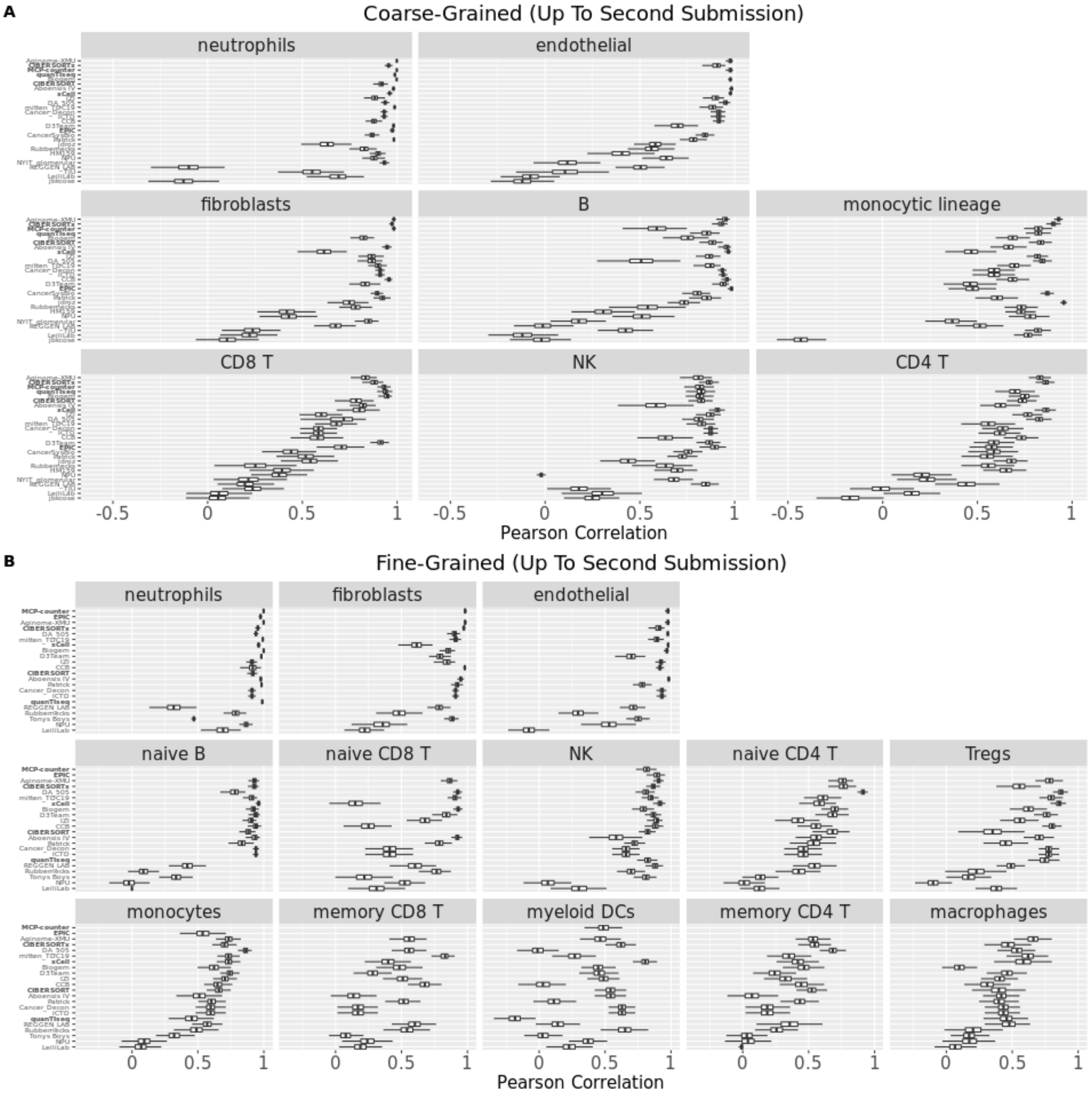


Fig. S11: Distribution of per-cell type method performance from second submission stratified by sub-Challenge. Performance (Pearson correlation; x axis) in (A) coarse- and (B) fine-grained sub-Challenges of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods). Results from latest submission up to the second, if less than two submissions for corresponding method. Blank row indicates cell type not reported by the corresponding method.

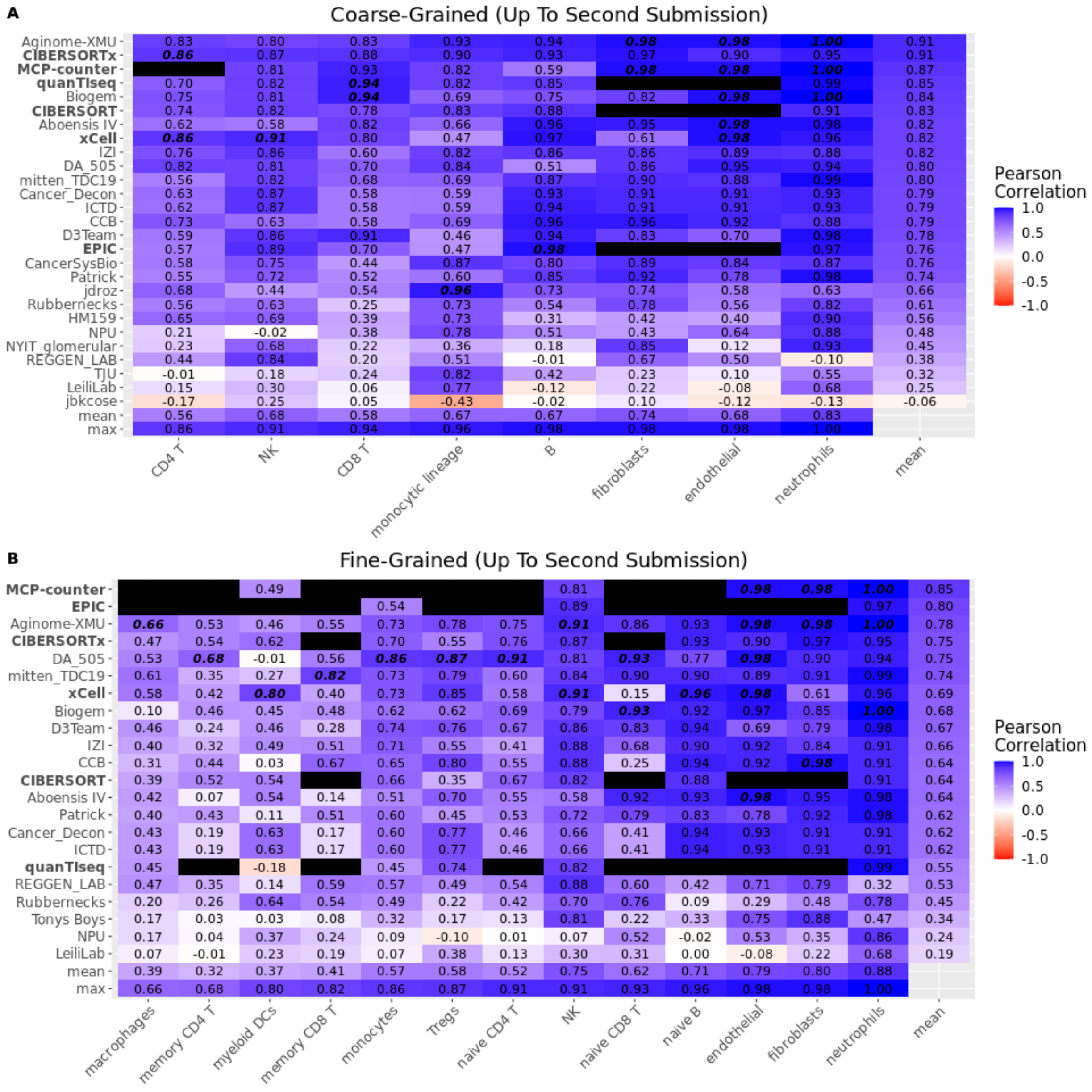


Fig. S12: Per-cell type method performance from second submission stratified by sub-Challenge. Pearson correlation of method (left axis) prediction versus known proportion from admixture for each cell type (bottom axis) in (A) coarse- or (B) fine-grained sub-Challenge. Pearson correlation is first averaged over validation dataset and then (*n*=1,000) over bootstraps (Methods). Results from latest submission up to second for those methods with fewer than two submissions. Black entry indicates cell type not predicted by corresponding method. Bottom two rows (“mean” and "max") are the mean and maximum correlation, respectively, for corresponding cell type across methods. Rightmost column ("mean") is mean correlation for corresponding method across predicted cell types. Highest correlation for each cell type highlighted in bold italics. Comparator methods in bold.

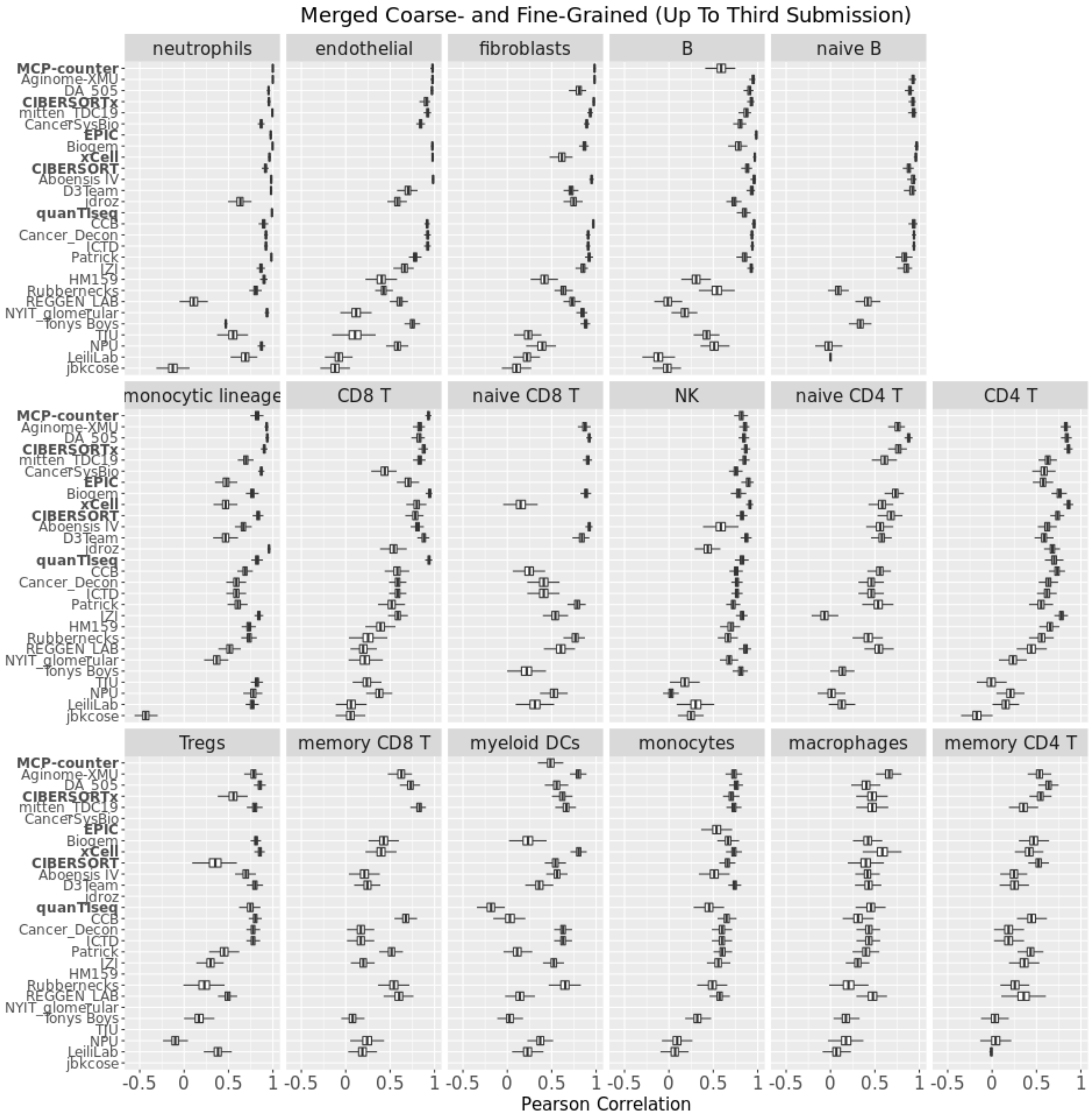


Fig. S13: Distribution of per-cell type method performance from third submission merged across coarse- and fine-grained sub-Challenges. Performance (Pearson correlation; x axis) of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods), computed as average over validation datasets and subsequently over coarse- and fine-grained sub-Challenges for cell types occurring in both. Results from latest submission up to the third, if less than three submissions for corresponding method. Blank row indicates cell type not reported by the corresponding method.

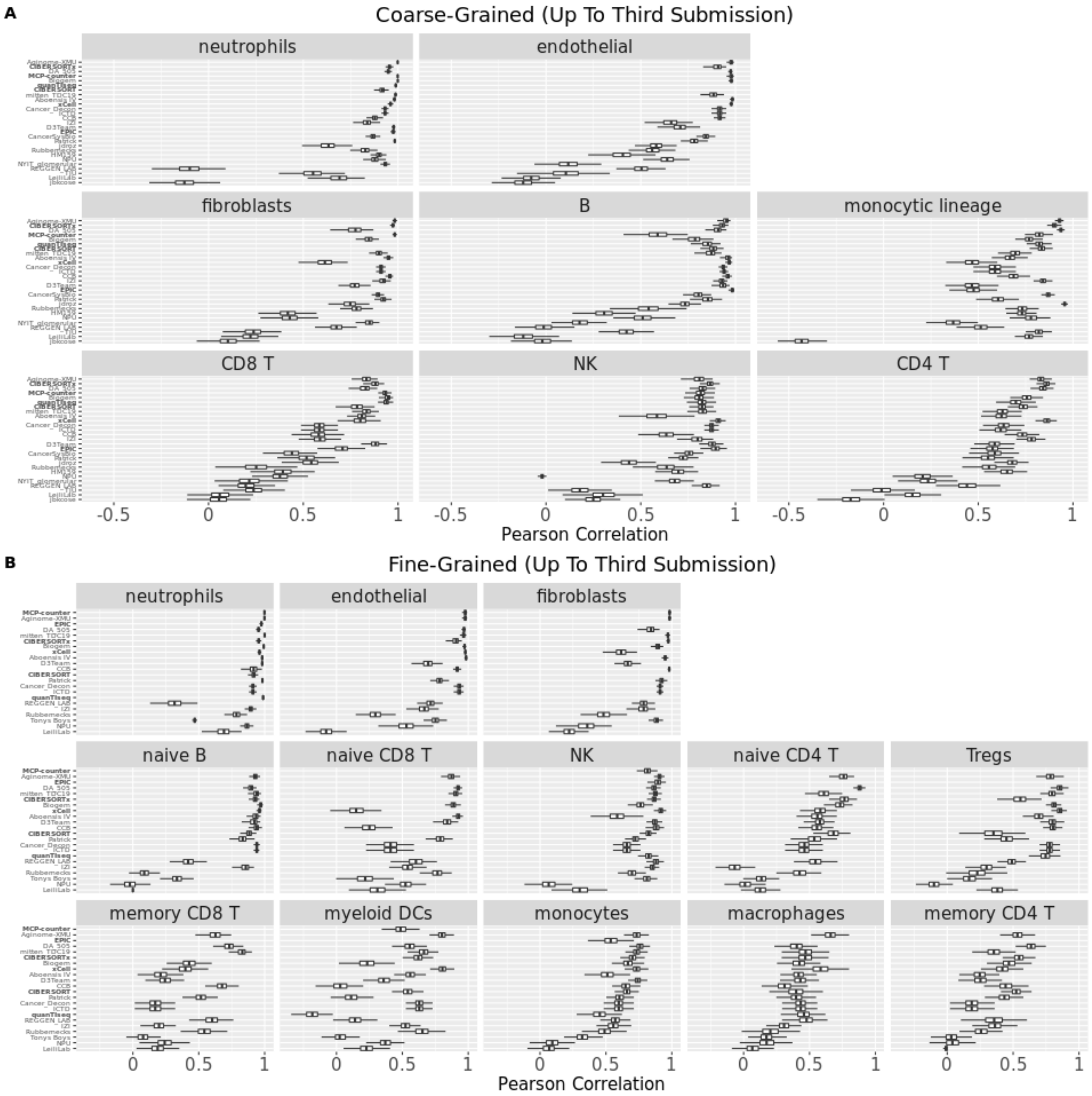


Fig. S14: Distribution of per-cell type method performance from third submission stratified by sub-Challenge. Performance (Pearson correlation; x axis) in (A) coarse- and (B) fine-grained sub-Challenges of comparator baseline methods (bold) and participant methods (y axis) for each cell type (facet label). Distribution of Pearson correlations over bootstraps (*n*=1,000; Methods). Results from latest submission up to the third, if less than three submissions for corresponding method. Blank row indicates cell type not reported by the corresponding method.

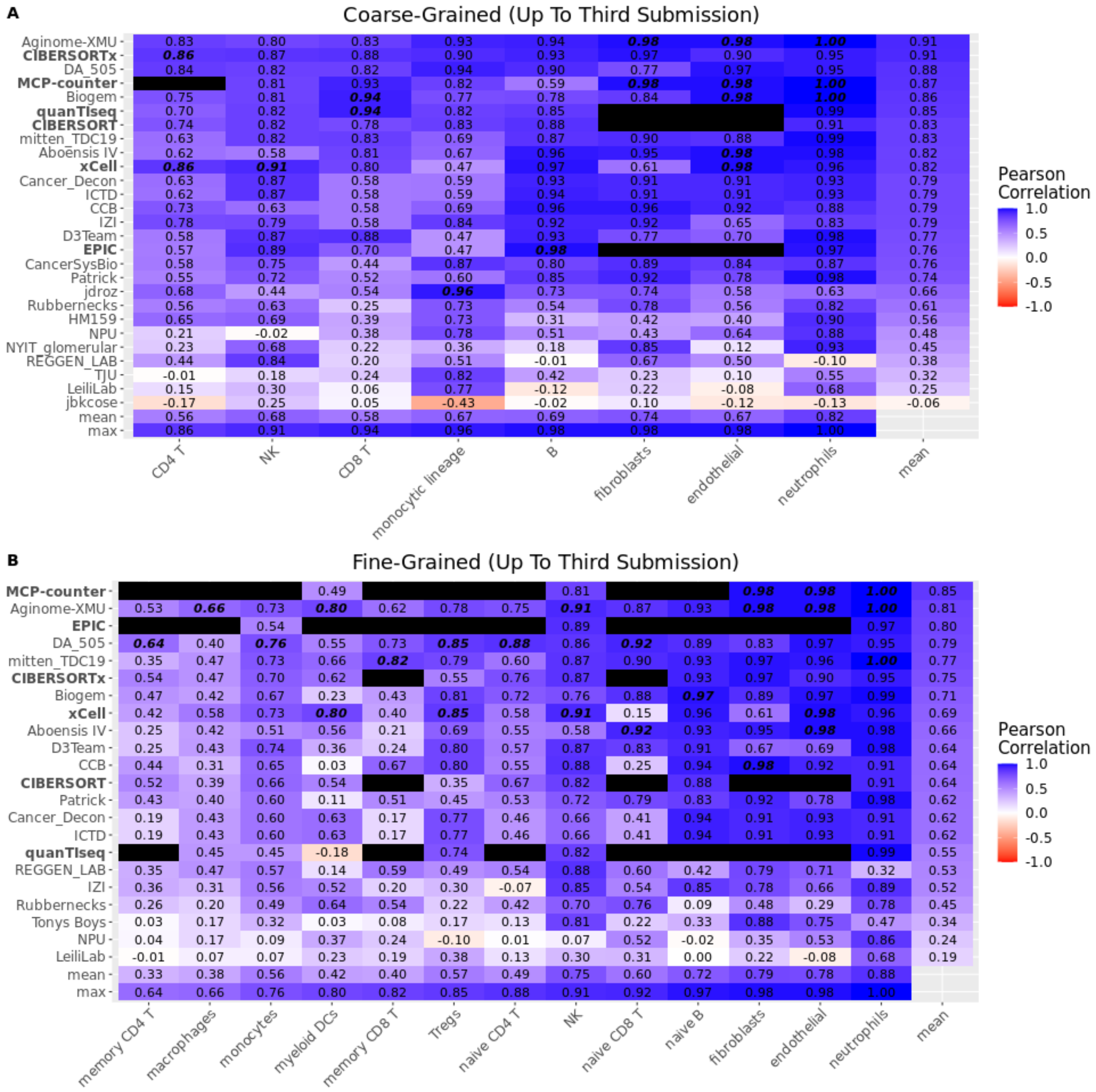


Fig. S15: Per-cell type method performance from third submission stratified by sub-Challenge. Pearson correlation of method (left axis) prediction versus known proportion from admixture for each cell type (bottom axis) in (A) coarse- or (B) fine-grained sub-Challenge. Pearson correlation is first averaged over validation dataset and then (*n*=1,000) over bootstraps (Methods). Results from latest submission up to third for those methods with fewer than three submissions. Black entry indicates cell type not predicted by corresponding method. Bottom two rows (“mean” and "max") are the mean and maximum correlation, respectively, for corresponding cell type across methods. Rightmost column ("mean") is mean correlation for corresponding method across predicted cell types. Highest correlation for each cell type highlighted in bold italics. Comparator methods in bold.

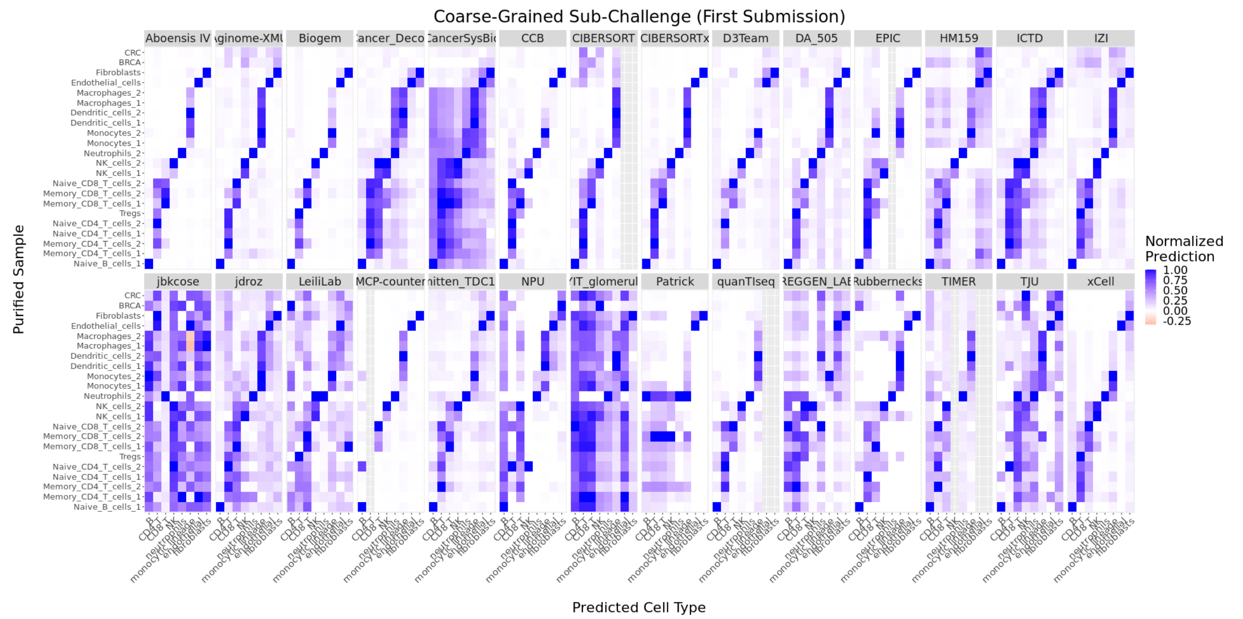


Fig. S16: Assessing specificity in coarse-grained sub-Challenge. Normalized prediction of cell type indicated on x axis in purified sample indicated on y axis.

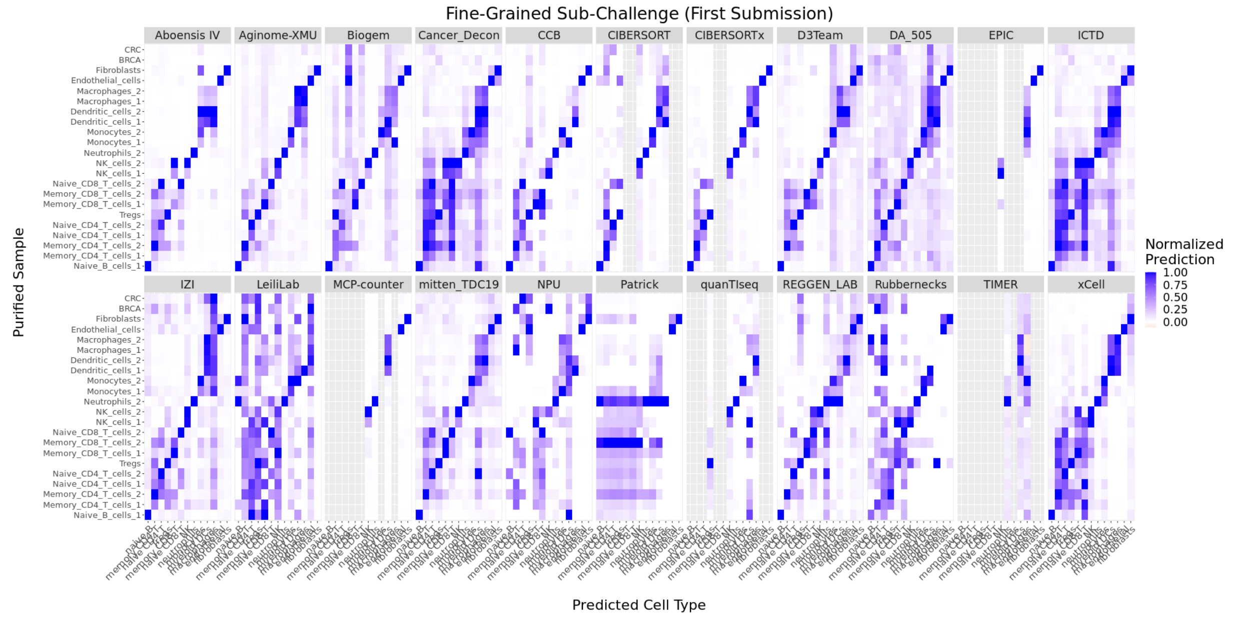


Fig. S17: Assessing specificity in fine-grained sub-Challenge. Normalized prediction of cell type indicated on x axis in purified sample indicated on y axis.

References

1. Petitprez, F. *et al.* Transcriptomic analysis of the tumor microenvironment to guide prognosis and immunotherapies. *Cancer Immunol. Immunother.* **67**, 981–988 (2018).
2. Vincent, B. G. *et al.* Pursuing Better Biomarkers for Immunotherapy Response in Cancer Through a Crowdsourced Data Challenge. *JCO Precis Oncol* **5**, 51–54 (2021).
3. Petitprez, F. *et al.* Quantitative Analyses of the Tumor Microenvironment Composition and Orientation in the Era of Precision Medicine. *Front. Oncol.* **8**, 390 (2018).
4. Lun, X.-K. & Bodenmiller, B. Profiling Cell Signaling Networks at Single-cell Resolution. *Mol. Cell. Proteomics* **19**, 744–756 (2020).
5. Bandura, D. R. *et al.* Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. *Anal. Chem.* **81**, 6813–6822 (2009).
6. Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* **6**, 377–382 (2009).
7. Denisenko, E. *et al.* Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus RNA-seq workflows. *Genome Biol.* **21**, 130 (2020).
8. van den Brink, S. C. *et al.* Single-cell sequencing reveals dissociation-induced gene expression in tissue subpopulations. *Nat. Methods* **14**, 935–936 (2017).
9. Slyper, M. *et al.* A single-cell and single-nucleus RNA-Seq toolbox for fresh and frozen human tumors. *Nat. Med.* **26**, 792–802 (2020).
10. Lambrechts, D. *et al.* Phenotype molding of stromal cells in the lung tumor microenvironment. *Nat. Med.* **24**, 1277–1289 (2018).
11. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome Res.* **25**, 1491–1498 (2015).

12. Lin, J.-R., Fallahi-Sichani, M. & Sorger, P. K. Highly multiplexed imaging of single cells using a high-throughput cyclic immunofluorescence method. *Nat. Commun.* **6**, 8390 (2015).

13. Giesen, C. *et al.* Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. *Nat. Methods* **11**, 417–422 (2014).

14. Goltsev, Y. *et al.* Deep Profiling of Mouse Splenic Architecture with CODEX Multiplexed Imaging. *Cell* **174**, 968–981.e15 (2018).

15. Angelo, M. *et al.* Multiplexed ion beam imaging of human breast tumors. *Nat. Med.* **20**, 436–442 (2014).

16. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).

17. Zollinger, D. R., Lingle, S. E., Sorg, K., Beechem, J. M. & Merritt, C. R. GeoMx^{TM} RNA Assay: High Multiplex, Digital, Spatial Analysis of RNA in FFPE Tissue. *Methods Mol. Biol.* **2148**, 331–345 (2020).

18. He, S. *et al.* High-plex Multiomic Analysis in FFPE at Subcellular Level by Spatial Molecular Imaging. *bioRxiv* 2021.11.03.467020 (2022) doi:10.1101/2021.11.03.467020.

19. Newman, A. M. *et al.* Robust enumeration of cell subsets from tissue expression profiles. *Nat. Methods* **12**, 453–457 (2015).

20. Racle, J., de Jonge, K., Baumgaertner, P., Speiser, D. E. & Gfeller, D. Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *Elife* **6**, (2017).

21. Becht, E. *et al.* Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression. *Genome Biol.* **17**, 218 (2016).

22. Li, B. *et al.* Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome Biol.* **17**, 174 (2016).

23. Finotello, F. *et al.* Molecular and pharmacological modulators of the tumor immune contexture revealed by deconvolution of RNA-seq data. *Genome Med.* **11**, 34 (2019).

24. Aran, D., Hu, Z. & Butte, A. J. xCell: digitally portraying the tissue cellular heterogeneity landscape. *Genome Biol.* **18**, 220 (2017).

25. Zaitsev, K., Bambouskova, M., Swain, A. & Artyomov, M. N. Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures. *Nat. Commun.* **10**, 2209 (2019).

26. Sturm, G. *et al.* Comprehensive evaluation of transcriptome-based cell-type quantification methods for immuno-oncology. *Bioinformatics* **35**, i436–i445 (2019).

27. Avila Cobos, F., Alquicira-Hernandez, J., Powell, J. E., Mestdagh, P. & De Preter, K. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nat. Commun.* **11**, 5650 (2020).

28. Jin, H. & Liu, Z. A benchmark for RNA-seq deconvolution analysis under dynamic testing environments. *Genome Biol.* **22**, 102 (2021).

29. Bruni, D., Angell, H. K. & Galon, J. The immune contexture and Immunoscore in cancer prognosis and therapeutic efficacy. *Nat. Rev. Cancer* **20**, 662–680 (2020).

30. Mason, M. J. *et al.* Multiple Myeloma DREAM Challenge reveals epigenetic regulator PHF19 as marker of aggressive disease. *Leukemia* **34**, 1866–1874 (2020).

31. Guinney, J. & Saez-Rodriguez, J. Alternative models for sharing confidential biomedical data. *Nat. Biotechnol.* **36**, 391–392 (2018).

32. Newman, A. M. *et al.* Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat. Biotechnol.* **37**, 773–782 (2019).

33. Lin, Y. *et al.* DAISM-DNN: Highly accurate cell type proportion estimation with data augmentation and deep neural networks. *Patterns (N Y)* **3**, 100440 (2022).

34. Domanskyi, S. *et al.* Polled Digital Cell Sorter (p-DCS): Automatic identification of hematological cell types from single cell RNA-sequencing clusters. *BMC Bioinformatics* **20**, 369 (2019).

35. Domanskyi, S., Hakansson, A., Bertus, T. J., Paternostro, G. & Piermarocchi, C. Digital Cell

Sorter (DCS): a cell type identification, anomaly detection, and Hopfield landscapes toolkit for single-cell transcriptomics. *PeerJ* **9**, e10670 (2021).

36. Monaco, G. *et al.* RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types. *Cell Rep.* **26**, 1627–1640.e7 (2019).
37. Duan, Z. & Luo, Y. Targeting macrophages in cancer immunotherapy. *Signal Transduct Target Ther* **6**, 127 (2021).
38. Decamps, C. *et al.* DECONbench: a benchmarking platform dedicated to deconvolution methods for tumor heterogeneity quantification. *BMC Bioinformatics* **22**, 473 (2021).
39. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**, 207–210 (2002).
40. Barrett, T. *et al.* NCBI GEO: archive for functional genomics data sets--update. *Nucleic Acids Res.* **41**, D991–5 (2013).
41. Zhu, Y., Davis, S., Stephens, R., Meltzer, P. S. & Chen, Y. GEOmetadb: powerful alternative search engine for the Gene Expression Omnibus. *Bioinformatics* **24**, 2798–2800 (2008).
42. Zalocusky, K. A. *et al.* The 10,000 Immunomes Project: Building a Resource for Human Immunology. *Cell Rep.* **25**, 513–522.e3 (2018).
43. Azizi, E. *et al.* Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment. *Cell* **174**, 1293–1308.e36 (2018).
44. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
45. Thorsson, V. *et al.* The Immune Landscape of Cancer. *Immunity* **48**, 812–830.e14 (2018).
46. Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. *Eur. J. Oper. Res.* **249**, 667–676 (2016).
47. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc.

48. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014).

49. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34**, 525–527 (2016).

50. Soneson, C., Love, M. I. & Robinson, M. D. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Res.* **4**, 1521 (2015).

51. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, R29 (2014).

52. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* vol. 2008 P10008 (2008).

53. Breiman, L. Random Forests. *Mach. Learn.* **45**, 5–32 (2001).

54. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing* vol. 14 199–222 (2004).

55. Otto, D. Computational Gene Expression Deconvolution. (2021).

56. Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. & Blei, D. M. Automatic Differentiation Variational Inference. (2016) doi:10.48550/ARXIV.1603.00788.

57. Salvatier, J., Wiecki, T. V. & Fonnesbeck, C. Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* **2**, e55 (2016).