Applications of community detection algorithms to large biological datasets

Itamar Kanter[1], Gur Yaari[1,*,+], and Tomer Kalisky[1,*,+]

[1]BIU, Department of Bioengineering, Bar-Ilan University, Ramat Gan, 5290002, Israel
[*]corresponding. tomer.kalisky@gmail.com, gur.yaari@biu.ac.il
[+]These authors share equal senior authorship

ABSTRACT

Recent advances in data acquiring technologies in biology have led to major challenges in mining relevant information from large datasets. For example, single-cell RNA sequencing technologies are producing expression and sequence information from tens of thousands of cells in every single experiment. A common task in analyzing biological data is to cluster samples or features (e.g. genes) into groups sharing common characteristics. This is an NP-hard problem for which numerous heuristic algorithms have been developed. However, in many cases, the clusters created by these algorithms do not reflect biological reality. To overcome this, a Networks Based Clustering (NBC) approach was recently proposed, by which the samples or genes in the dataset are first mapped to a network and then community detection (CD) algorithms are used to identify clusters of nodes.

Here, we created an open and flexible python-based toolkit for NBC that enables easy and accessible network construction and community detection. We then tested the applicability of NBC for identifying clusters of cells or genes from previously published large-scale single-cell and bulk RNA-seq datasets.

We show that NBC can be used to accurately and efficiently analyze large-scale datasets of RNA sequencing experiments.

Introduction

Advances in high-throughput genomic technologies have revolutionized the way biological data is being acquired. Technologies like DNA sequencing (DNA-seq), RNA sequencing (RNA-seq), chromatin immunoprecipitation sequencing (ChIP-seq), and mass cytometry are becoming standard components of modern biological research. The majority of these datasets are publicly available for further large-scale studies. Notable examples include the Genotype-Tissue Expression (GTEx) project[1], the cancer genome atlas (TCGA)[2], and the 1000 genomes project[3]. Examples of utilizing these datasets include studying allele-specific expression across tissues[4,5], characterizing functional variation in the human genome[6], finding patterns of transcriptome variations across individuals and tissues[7], and characterizing the global mutational landscape of cancer[8]. Moreover, some of these genomic technologies have recently been adapted to work at the single-cell level[9]. While pioneering single-cell RNA sequencing (scRNA-seq) studies were able to process relatively small numbers of cells (42 cells in[10] and 18 cells in[11]), recent single-cell RNA-seq studies taking advantage of automation and nanotechnology were able to produce expression and sequence data from many thousands of individual cells ($\sim$1,500 cells in[12] and $\sim$40,000 cells in[13]). Hence, biology is facing significant challenges in handling and analyzing large complex datasets[14,15].

Clustering analysis

One of the common methods used for making sense of large biological datasets is cluster analysis: the task of grouping similar samples or features[16]. For example, clustering analysis has been used to identify subtypes of breast tumors[17,18] with implications to treatment and prognosis. More recently, clustering analysis was used to identify and characterize cell types in various tissues and tumors in the colon[19], brain[20], blood[12], and lung[21], with the overall aim of finding key stem and progenitor cell populations involved in tissue development, repair, and tumorigenesis. Another application is to find sets of coordinately regulated genes in order to find gene modules[11,22,23]. Such clusters of genes (or other features such as single-nucleotide polymorphism (SNPs)[24]) can be further analyzed by gene set enrichment approaches to identify gene annotations[25] (e.g. GO[26], KEGG[27], and OMIM[28]) that are over-represented in a given cluster, and thus shed light on their biological functionalities[29]. Two of the most common clustering methods used in biology are K-means clustering, which groups data-points into K prototypes (where K is a predetermined number of clusters), and hierarchical clustering, which builds a hierarchy of clusters from all data points[30]. Other methods for clustering include self-organizing map (SOM)[31], spectral clustering[32], and density based methods[33] (for a comprehensive review on clustering see[30,34,35]).

Networks and community detection

Another way to model biological systems is through network science. Networks (also known as graphs) are structures composed of nodes that are connected by edges, which may be weighted and/or directional. Networks have been used for modeling interactions between components of complex systems such as users in social media platforms (Facebook[36] and Twitter[37]) or proteins[38] and genes[39] in a cell. Often, networks contain communities of nodes that are tightly interconnected with each other, which is an indication of common features. For example, communities in social networks[40] are composed, in most cases, of people with common interests or goals that are interacting with each other. Likewise, proteins with related functionalities interact with each other, and as a result form close-knit communities in protein-protein interactions (PPI) networks[41]. Similarly, web pages with similar content in the World Wide Web[42] usually have links to each other.

The problem of community detection (CD), which can be viewed as a network's equivalent for clustering, is not rigorously defined. As a consequence, there are numerous CD algorithms that solve this problem reasonably well using different strategies[43]. An intuitive way to define communities in a network is to divide the nodes into groups that have many in-group edges and few out-group edges. This can be achieved by maximizing the network modularity - a measure that quantifies edge density within communities compared to edge sparseness between communities[44,45] (see Methods for formal definition).

Numerous community detection algorithms were developed during the last two decades[43,46]. Newman defined a measure for the modularity of a weighted network[45]. Clauset et al. developed a fast greedy algorithm to partition the nodes of the network in a way that maximizes the modularity by hierarchical agglomeration of the nodes[47]. Reichardt et al. proposed an approach based on statistical mechanics. Their algorithm models the network as a spin glass and aims to find the partition of nodes with maximal modularity by finding the minimal energy state of the system[48]. Rosvall et al. and Yucel et al. proposed methods to find the community structure by approximating the information flow in the network as a random walk[49,50]. Jian et al. developed SPICi - a fast clustering algorithm for large biological networks based on expanding clusters from local seeds[51].

A popular community detection algorithm called the *Louvain* algorithm was proposed by Blondel et al.[52] (see Methods for details). The *Louvain* algorithm starts by defining each node as a separate community and then performs modularity optimization by an iterative heuristic two-step process. In the first step, the algorithm goes over all nodes of the network and checks, for each individual node, if the network modularity can be increased by removing it from its present community and joining it to one of its neighboring communities. The process is repeated until no further increase in modularity can be achieved. This approach is called the "local moving heuristic". In the second step, a new meta-network, whose nodes are the communities identified by the first step, is constructed. The two steps are repeated until maximum modularity is attained. It was shown that this algorithm can be improved further by modifying the "local moving heuristic". SLM, for example, attempts to split each community into sub-communities prior to construction of the meta-network. In this way, communities can be split up and sets of nodes can be moved from one community to another for improving the overall modularity score[53].

Recently, several networks-based clustering algorithms were developed specifically for single-cell gene expression datasets (see Table 1). Pe'er and colleagues developed PhenoGraph[54]. This method first builds a k-nearest neighbors (KNN) network of cells, where each cell is connected to its K nearest cells in Euclidean space. In order to better resolve rare or non-convex cell populations, the algorithm then constructs a second, shared nearest neighbors (SNN) network, in which the similarity between every two nodes is determined by the number of neighboring nodes that are connected to both of them. Finally, the *Louvain* community detection algorithm is used to find groups of cells with similar gene expression profiles. Applying their method to mass cytometry data from 30,000 human bone marrow cells, they were able to cluster single-cell expression profiles into different immune cell types. Su and Xu developed another algorithm called SNN-cliq[55]. This algorithm also constructs a KNN network, and then constructs a SNN network in which the weight between every two nodes is determined not only by the number of shared nearest neighbors, but also their distances to the two nodes. Communities then are detected using a quasi-clique-based clustering algorithm. When applying their method to several single-cell transcriptomics datasets, they found it to be more robust and precise than traditional clustering approaches.

In this manuscript, we introduce an accessible and flexible python-based toolkit for Networks Based Clustering (NBC) for large-scale RNA-seq datasets in the form of an *IPython* notebook with a self-contained example (Supplementary File S1 online). This toolkit allows the user to follow and modify various aspects of the algorithms detailed above[52–56], that is, to map a given dataset into a KNN network, to visualize the network, and to perform community detection using a variety of similarity measures and community detection algorithms of his choice. This flexibility is important since, from our experience, different parameters and algorithms might work best for different datasets according to their specific characteristics. Using this toolkit, we tested the performance of NBC on previously published large-scale single-cell and bulk RNA-seq datasets.

Results

A workflow for Networks Based Clustering (NBC)

A typical Networks Based Clustering workflow can be divided into four steps (Fig 1). The given dataset, in the form of a matrix of N samples (e.g. cells) by P features (e.g. genes), is first preprocessed by normalizing the samples to each other[57,58] and

filtering less informative samples or features[58,59]. This step is especially important in single-cell data since it is typically noisier than bulk samples. Likewise, in meta-analysis, it is important to normalize samples obtained from different sources in order to to mitigate bias due to batch effects[60,61]. In our *IPython* notebook we took a dataset that was collected and pre-filtered by Patel et al.[62] and normalized each sample such that it will have zero mean and unit length (L2 normalization).

Next, a similarity measure is defined between the samples (or alternatively, features) and a KNN (K-nearest neighbors) network is constructed as follows: First, each sample (or feature) is represented as a node. Then each node is linked to its K nearest neighboring nodes. Constructing a KNN network using the naïve algorithm has a complexity of $O(N^2)$ which is slow for large N. We therefore use the more efficient ball tree algorithm[63], whose complexity scales as $O(N\log(N))$ if supplied with a true distance metric that satisfies the triangle inequality[64]. There are various such distance-like measures (each based on a similarity measures) that can be used[65] to construct the network, for example, the Euclidean distance, the Minkowski distance, and the cosine distance. In our solved example we used the cosine similarity (see Methods). Note that the popular correlation distance does not satisfy the triangle inequality[64] and hence is not a true metric. Following network construction, a community detection (CD) algorithm is performed, resulting in an assignment of each node (sample or feature) to a distinct community.

Once communities have been identified, each community can be characterized to infer its biological meaning. For example, communities of cells may represent cell sub-populations in complex tissues or tumors and can be identified using previously known markers[66,67]. Similarly, the biological functionality of communities of genes (or of gene sets that are over-expressed in specific cell communities) can be inferred using enrichment analysis at the gene and gene-set levels[29,68–72].

NBC accurately resolves seven cell types from a glioblastoma single-cell RNA-seq dataset

To test the performance of NBC, we analyzed single-cell RNA-seq datasets originally published by Patel *et al.*[62], for which the biological interpretation was known a-priori. These datasets were previously obtained from five glioblastoma patients and two gliomasphere cell lines and were found to contain 7 biologically distinct cell types. A 2D representation of the data by PCA and tSNE can be found in Supplementary File S1 online. We first calculated the distance between individual cells according to the cosine similarity and then constructed a KNN network with $K = 40$ (for details see Methods). We applied the *Louvain* algorithm[52], detected communities of cells, and used the $F-measure$ (the harmonic mean between precision and sensitivity) to check the degree to which the inferred communities reproduce the known cell types from the original publication. We found that NBC resolves the original seven cell types with high precision and sensitivity (Fig 2a and b, $F-measure = 0.93$). Constructing the network with $K = 10$ resulted in a slightly lower $F-measure$ (Fig 2c and d, $F-measure = 0.81$), mainly due to the separation of one original cluster (indicated by light blue in C) into two inferred clusters (indicated by light blue and orange in D). Evaluating the precision and sensitivity of NBC for a wide range of K's shows that, for this dataset, NBC is quite robust to the choice of K for values larger than $K \approx 18$ (Fig 2e). Using the correlation similarity for constructing the KNN network results in a similar performance in terms of the $F-measure$ (Fig 2e). In this dataset, NBC outperformed other common clustering methods (Table 2).

We performed a similar analysis on another single-cell RNA-seq dataset published by Klein *et al.*[73] containing 2,717 mouse embryonic stem cells collected from four consecutive developmental stages. We found that also for this dataset, NBC performs well relative to other common clustering methods (Fig 3 and Table 2). However, here we found that, for sufficiently large K ($K > 50$), correlation similarity had better performance than cosine similarity ($F-measure = 0.90$ for correlation similarity and 0.81 for cosine similarity, in both cases using the *Louvain* algorithm, Fig 3e and Table 2).

Comparing NBC with other common clustering methods

We compared NBC with three widely used clustering algorithms: K-means, hierarchical clustering, and spectral clustering. As a reference dataset, we used 3,174 human tissue-specific gene expression samples from the GTEx project[1] that were collected from 21 tissue types. Based on the number of tissue types in the original samples, the number of clusters was set to 21 for all clustering algorithms (see Methods). A KNN network was constructed with $K = 50$ and the *Louvain* algorithm was applied to infer sample communities. In order to compare the algorithms and test their robustness to different levels of "noise", a fraction of the genes was randomly permuted by shuffling their sample labels (Fig 4a), thereby maintaining the original distribution for each gene. This actually mimics different levels of non-informative features. We observed that for this data set, NBC out-performs K-means, spectral, and hierarchical clustering in terms of the *F-measure* over a wide range of noise levels (Fig 4a).

We performed a similar comparison of the CPU time required to detect communities using a single-cell RNA-seq dataset that was published by Macosko et al.[13]. We randomly chose subsets of varying sizes of samples (cells) and genes in order to test the dependency of the running-time on the size of the dataset. We found that NBC falls in-between K-means, Hierarchical clustering, and spectral clustering for a wide range of samples and genes (Fig 4b and c).

NBC can be used to resolve tissue-specific genes

NBC can also be used to detect communities of genes from large gene expression datasets. To demonstrate this, we analyzed a dataset composed of 394 GTEx samples collected from three tissues: pancreas, liver, and spleen. First, a KNN network with

$K = 10$ was constructed for the 394 samples. The resulting network contained three perfectly separated components, each corresponding to one of the three tissue types (Fig 5b-d, $F - measure = 1$). We then constructed another KNN network for the 27,838 genes with $K = 200$. The *Louvain* community detection algorithm was applied and 11 gene communities were detected.

In order to explore the biological meaning of these 11 NBC-derived gene communities, we used three independently derived tissue-specific "reference lists" of genes from the Human Protein Atlas[74] that were found to be over-expressed in the pancreas, liver, and spleen. We compared these three reference lists to the 11 NBC-derived communities and found that each reference list was found predominantly in a single community (Fig 5a). In community #1, 200 of the 210 spleen-specific reference genes were found, in community #2, 82 of the 87 pancreas-specific reference genes were found, and in community #3, 397 of the 403 liver-specific reference genes were found. On the contrary, a reference list of "House Keeping Genes" (HKG) was found to be distributed relatively uniformly among the different communities.

Another helpful feature of NBC is that it can be used to visualize families of genes within the network of samples. To demonstrate this, we measured the relative gene expression levels of genes from NBC-derived community no. 1 (enriched for spleen related genes) in each node (=sample) of the network and used this information to determine the size of that node (Fig 5b). It can be seen that the average expression level of NBC-derived community #1, that is enriched for spleen specific genes, was indeed much higher in the spleen samples compared to pancreas and liver samples (Table 3, one side t-test p-value=2×10^{-61}). We observed similar results when we repeated this analysis for NBC-derived communities 2 and 3 (pancreas-enriched and liver-enriched, Fig 5c and d, Table 3).

Discussion

To date, genomic datasets typically contain hundreds of samples with thousands of features each. FACS datasets may contain millions of samples with 10-30 features each. Improvements in single-cell processing techniques (e.g. droplet-based[13] or magnetic-beads based methods[75]) are further increasing the number of samples in single-cell RNA-seq data. Therefore, tools for genomic data analysis need to perform efficiently on very large datasets. In this aspect, the major bottleneck of our toolkit is the KNN network construction step for which we use the ball tree algorithm[63]. Although efficient, this algorithm does not scale well with respect to memory usage and query time when the number of features increases. One possible solution is to use methods for approximating KNN networks, which might be utilized for this step after careful exploration of the error they introduce[76,77]. Another possibility is to use parallel architectures to accelerate KNN network construction[78].

Moreover, when the number of features P is larger than the number of samples N ($P > N$), network construction can be made more efficiently by projecting the original matrix (embedded in $\mathbb{R}^P$) into a lower dimension space ($\mathbb{R}^N$) using PCA[79]. This projection preserves the original Euclidean distances between the samples. Other possible projections into lower dimensions are truncated SVD or approximated PCA methods[80]. However, these do not preserve the original Euclidean distances between the samples[80]. The original matrix can also be projected into lower dimensions by non-linear transformations like tSNE[81]. tSNE captures much of the local structure of the data and has been widely used for single-cell expression analysis[21,82].

Note that Network-based methods themselves can be used for dimensionality reduction. Isomap[83], for instance, constructs a KNN graph that is used to approximate the geodesic distance between data points. Then, a multi-dimensional scaling is applied, based on the graph distances, to produce a low-dimensional mapping of the data that maintains the geodesic distances between all points.

NBC has much in common with the widely used density-based clustering method DBSCAN[33]. Although both methods explore the local structure of the data, NBC uses the K nearest neighbors, while DBSCAN defines clusters by their local densities. However in NBC, as opposed to DBSCAN, no minimal distance is required to define two samples as neighbors. In addition, DBSCAN does not produce the network explicitly, but rather just the connectivity component of each sample. This is in contrast to NBC that provides an explicit representation of the underlying weighted network that can be analyzed with different CD algorithms.

NBC requires the user to specify the following parameters: a similarity measure, a community detection algorithm, and the number of nearest neighbors K. For the datasets that we checked we found that NBC is not very sensitive to the choice of K given sufficiently large values ($K > 18$) (e.g. Fig 2e); however, we found that the choice of the community detection algorithm and especially the similarity measure may significantly influence its performance (e.g. Fig 3e and Fig 4a). Hence, these parameters should be chosen carefully when applying NBC to other data types. Similar to other machine learning approaches, NBC parameters can be optimized using a labeled training dataset prior to application on unlabeled data.

We created an open and flexible python-based toolkit for Networks Based Clustering (NBC) that enables easy and accessible KNN network construction followed by community detection for clustering large biological datasets, and used this toolkit to test the performance of NBC on previously published single-cell and bulk RNA-seq datasets. We find that NBC can identify communities of samples (e.g. cells) and genes, and that it performs better than other common clustering algorithms over a wide range of parameters.

In practice, given a new dataset, we recommend to carefully test different alternatives for network construction and community detection since results may vary among different datasets according to their unique characteristics. We believe that the open and flexible toolkit that we introduced here can assist in rapid testing of the many possibilities.

Methods

single-cell and "bulk" RNA sequencing datasets

We used Four datasets in this study:

I) Single-cell RNA-seq data from Patel et al.[62] containing single-cell gene expression levels from five patients with glioblastoma and two gliomasphere cell lines that were acquired using the SMART-SEQ protocol. We downloaded the preprocessed data from GEO[84]. Altogether, this dataset contains 543 cells by 5,948 genes.

II) Single-cell RNA-seq data from Klein et al.[73] containing single-cell gene expression levels from mouse embryonic stem cells at different stages of differentiation that were acquired using the inDrop protocol. In that experiment, samples were collected along the differentiation timeline by sequencing single cells at 0, 2, 4, 7 days after withdrawal of leukemia inhibitory factor (LIF). We downloaded the preprocessed data from GEO[85] and removed genes with zero expression levels, resulting in a dataset of 8,669 cells by 24,049 genes. For the analysis presented in Fig 3, we first removed technical replicates and data from a control cell line, resulting in a total number of 2,717 cells.

III) "Bulk" RNA sequencing datasets from the Genotype-Tissue Expression (GTEx) database[1]. We downloaded the data from the GTEx website[86] version 6. This dataset includes 8,555 samples taken from 30 tissue types (according to the SMTS variable) of 570 individuals. Gene names were translated from Ensemble gene ID into HGNC symbols using the *BioMart* Bioconductor package[87]. In cases where we found multiple matches of the Ensemble gene ID's corresponding to a single HGNC symbol, the Ensemble gene ID with maximum average intensity across all samples was chosen. To compare different clustering methods (Fig 4a) we chose only samples originating from a single tissue type. Moreover, we omitted tissues having multiple detailed tissue types (according to the SMTSD variable) even if they had a single tissue type (indicated by the SMTS variable). Likewise, genes with zero expression were omitted, resulting in a dataset of 3,174 samples by 33,183 genes from 21 tissue types.

IV) Single-cell RNA-seq data from Macosko et al.[13] containing single-cell gene expression levels from a P14 mouse retina that were acquired using the Drop-Seq protocol. We downloaded the preprocessed data from GEO[88] and removed genes with zero expression, resulting in a dataset of 49,300 cells by 24,071 genes. This dataset was used to compare the performance, in terms of CPU time, of NBC, K-means, and hierarchical clustering as shown in Fig 4b-c.

KNN network construction and visualization

A KNN network with cosine similarity was constructed using the *scikit-learn* package[89] for machine learning in Python. Since the cosine distance was not directly available in the *scikit-learn* BallTree() function, we used a two-step implementation as follows: First, each sample was mean-centered and standardized such that it will have zero mean and unit length (L2 normalization). Next, the ball tree algorithm[63] was applied with Euclidean distance to find the K nearest neighbors of each sample and construct a KNN network. Then, the Euclidean distances between the nodes (=samples) were transformed to cosine similarities that were used as the edges weights for community detection.

We calculated the cosine similarity from the Euclidean distance as follows. The cosine similarity between two vectors A and B is defined as:

$$sim_{cos}(A,B) \equiv 1 - \frac{\theta(A,B)}{\pi},$$

where $\theta(A,B)$ is the angle between A and B. If A and B are also of unit length (L2 normalized) then this angle is related to the Euclidean distance $D_{euc}(A,B)$ according to:

$$\theta(A,B) = \cos^{-1}[1 - \frac{D_{euc}(A,B)^2}{2}]$$

or: $D_{euc}(A,B) = \sqrt{2 - 2\cos\theta(A,B)}$.

Network layouts for visualization were created by the *fruchterman-reingold* algorithm[90] as implemented in the *igraph* Python and R packages[91]. For correlation similarity we calculated the full *spearman* correlation matrix $\rho(A,B)$ between any two vectors A and B using the *corr* function in R.

Community detection algorithms

In this manuscript we generally used the *Louvain*[52] algorithm for community detection as implemented by the *igraph* Python and R packages for network analysis[91], apart from Fig 3 in which we used the *fast greedy*[47] algorithm.

The *Louvain* method partitions the nodes of the network into communities $c1, c2, c3, \ldots$, such that network modularity score

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i \cdot k_j}{2m} \right] \delta(c_i, c_j),$$

is maximized. In the above formula, A_{ij} is the edge weight between nodes i and j, k_i is the degree of node i (that is, the sum of the weights of all the links emanating from node i), m is the overall sum of weights, $m = \frac{1}{2} \sum_{ij} A_{ij}$, and $\delta(c_i, c_j)$ is the Kronecker delta function. The network modularity score is actually the difference between the number of links connecting nodes within the same community and the expected number of links in a network with randomly shuffled links.

Briefly, the algorithm starts by assigning a separate community to each node. Then, the algorithm iterates between two steps: In the first step, the modularity is maximized by repeatedly iterating over all nodes in the network. For each node, we evaluate the gain in modularity that will take place by removing it from its present community and assigning it to one of its neighboring communities. If the overall modularity can be improved, the community of the node is reassigned accordingly. This process is repeated until a local maximum is reached. In the second step, the algorithm constructs a meta-network in which the nodes are communities from the first step and the edges are the edges between the communities. At this point, the first step is repeated on the nodes of the new meta-network in order to check if they can be merged into even larger communities. The algorithm stops when there is no more improvement in the modularity score.

Statistical measures for comparing NBC and other common clustering algorithms

To compare the performance of NBC, hierarchical clustering, K-means, and spectral clustering, we used the *F-measure*, which is the harmonic mean between the precision P and sensitivity (recall) R:

$$F \equiv \frac{2}{\frac{1}{P} + \frac{1}{R}} = 2 * \frac{P * R}{P + R}$$

where $P \equiv Precision \equiv \frac{TP}{TP+FP}$, and $R \equiv Recall \equiv \frac{TP}{TP+FN}$ (TP- true positive, FP- false positive, FN- false negative). To calculate precision and sensitivity for each clustering algorithm, we also used the R package *clusterCrit*[92] that compares the labels from the original publication to the labels inferred by the algorithm.

Another requirement for evaluating and comparing the different clustering algorithms was to require all of them to find the same number of clusters. Therefore, the number of required clusters was set to the number of distinct groups from the original publication (7 clusters in Fig 2, 4 clusters in Fig 3, 21 clusters in Fig 4a, etc.). We used the *stat* package in R[93] to run hierarchical clustering and K-means clustering with default parameters (Euclidean distance), apart from the linkage in hierarchical clustering which was set to *average* linkage. For spectral clustering we used the *specc* function from the *kernlab* R package[94] with default parameters. Generally, all parameters were chosen as default unless otherwise specified.

All computations were done on a standard PC with i7-4600 CPU with 2.10 GHz and 16 GB of RAM memory.

Tissue-specific reference genes lists from the Human Protein Atlas

Tissue-specific reference lists of genes were obtained from the Human Protein Atlas[74] version 14[95]. Altogether, the housekeeping genes (HKG) reference list is composed of 8,588 genes, and the liver-specific, pancreas-specific, and spleen-specific reference genes lists are composed of 436, 234, and 95 genes respectively. Genes that do not appear in the dataset or genes that appear in more than one tissue-specific list were removed, resulting 8331, 403, 210, and 87 genes in the HKG, liver-specific, pancreas-specific, and spleen-specific reference genes lists respectively.

References

1. The GTEx Consortium. The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science* **348**, 648–660, DOI: 10.1126/science.1262110 (2015).
2. Network, T. C. G. A. R. *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet.* **45**, 1113–1120 (2013).
3. Durbin, R. M. *et al.* A map of human genome variation from population-scale sequencing. *Nature* **467**, 1061–1073, DOI: 10.1038/nature09534 (2010).
4. Baran, Y. *et al.* The landscape of genomic imprinting across diverse adult human tissues. *Genome Res.* **25**, 927–936, DOI: 10.1101/gr.192278.115 (2015).
5. Pirinen, M. *et al.* Assessing allele-specific expression across multiple tissues from RNA-seq read data. *Bioinformatics* **31**, 2497–2504, DOI: 10.1093/bioinformatics/btv074 (2015).

6. Lappalainen, T. *et al.* Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501**, 506–511, DOI: 10.1038/nature12531 (2013).

7. Mele, M. *et al.* The human transcriptome across tissues and individuals. *Science* **348**, 660–665, DOI: 10.1126/science.aaa0355 (2015).

8. Leiserson, M. D. M. *et al.* Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.* **47**, 106–114, DOI: 10.1038/ng.3168 (2014).

9. Nawy, T. Single-cell sequencing. *Nat. Methods* **11**, 18–18, DOI: 10.1038/nmeth.2771 (2013).

10. Ramsköld, D. *et al.* Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nat. biotechnology* **30**, 777–82, DOI: 10.1038/nbt.2282 (2012).

11. Shalek, A. K. *et al.* Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature* **498**, 236–40, DOI: 10.1038/nature12172 (2013).

12. Jaitin, D. A. *et al.* Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types. *Science* **343**, 776–779, DOI: 10.1126/science.1247651 (2014).

13. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202–1214, DOI: 10.1016/j.cell.2015.05.002 (2015).

14. Stephens, Z. D. *et al.* Big Data: Astronomical or Genomical? *PLOS Biol.* **13**, e1002195, DOI: 10.1371/journal.pbio.1002195 (2015).

15. Marx, V. Biology: The big challenges of big data. *Nature* **498**, 255–260, DOI: 10.1038/498255a (2013).

16. Daxin Jiang, Chun Tang & Aidong Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowl. Data Eng.* **16**, 1370–1386, DOI: 10.1109/TKDE.2004.68 (2004).

17. Sørlie, T. *et al.* Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. United States Am.* **98**, 10869–74, DOI: 10.1073/pnas.191367098 (2001).

18. Kapp, A. V. *et al.* Discovery and validation of breast cancer subtypes. *BMC genomics* **7**, 231, DOI: 10.1186/1471-2164-7-231 (2006).

19. Rothenberg, M. E. *et al.* Identification of a cKit(+) colonic crypt base secretory cell that supports Lgr5(+) stem cells in mice. *Gastroenterology* **142**, 1195–1205.e6, DOI: 10.1053/j.gastro.2012.02.006 (2012).

20. Pollen, A. A. *et al.* Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat. Biotechnol.* **32**, 1053–1058, DOI: 10.1038/nbt.2967 (2014).

21. Treutlein, B. *et al.* Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq. *Nature* 1–15, DOI: 10.1038/nature18323 (2016).

22. Kolodziejczyk, A. A. *et al.* Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation. *Cell stem cell* **17**, 471–85, DOI: 10.1016/j.stem.2015.09.011 (2015).

23. Wang, J. *et al.* Single-Cell Co-expression Analysis Reveals Distinct Functional Modules, Co-regulation Mechanisms and Clinical Outcomes. *PLoS computational biology* **12**, e1004892, DOI: 10.1371/journal.pcbi.1004892 (2016).

24. Wills, Q. F. *et al.* Single-cell gene expression analysis reveals genetic associations masked in whole-tissue experiments. *Nat. biotechnology* **31**, 748–52, DOI: 10.1038/nbt.2642 (2013).

25. Hung, J.-H., Yang, T.-H., Hu, Z., Weng, Z. & DeLisi, C. Gene set enrichment analysis: performance evaluation and usage guidelines. *Briefings Bioinforma.* **13**, 281–291, DOI: 10.1093/bib/bbr049 (2012).

26. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29, DOI: 10.1038/75556 (2000).

27. Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28**, 27–30, DOI: 10.1093/nar/28.1.27 (2000).

28. Hamosh, A. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.* **33**, D514–D517, DOI: 10.1093/nar/gki033 (2004).

29. Huang, D. W., Sherman, B. T. & Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. protocols* **4**, 44–57, DOI: 10.1038/nprot.2008.211 (2009).

30. Jain, A. K., Murty, M. N. & Flynn, P. J. Data clustering: a review, DOI: 10.1145/331499.331504 (1999).

31. Kohonen, T. The self-organizing map. *Proc. IEEE* **78**, 1464–1480, DOI: 10.1109/5.58325 (1990).

32. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416, DOI: 10.1007/s11222-007-9033-z (2007). arXiv:0711.0189v1.

33. Martin, E., Hans-Peter, K., Jörg, S. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD-96 Proc.* 226–231, DOI: CiteSeerX:10.1.1.121.9220 (1996).

34. Xu, R. & Wunsch, D. C. Clustering algorithms in biomedical research: a review. *IEEE reviews biomedical engineering* **3**, 120–154, DOI: 10.1109/RBME.2010.2083647 (2010).

35. Berkhin, P. A Survey of Clustering Data Mining Techniques. In Kogan, J., Nicholas, C. & Teboulle, M. (eds.) *Grouping Multidimensional Data: Recent Advances in Clustering*, 25–71, DOI: 10.1007/3-540-28349-8_2 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006).

36. Lewis, K., Kaufman, J., Gonzalez, M., Wimmer, A. & Christakis, N. Tastes, ties, and time: A new social network dataset using Facebook.com. *Soc. Networks* **30**, 330–342, DOI: 10.1016/j.socnet.2008.07.002 (2008).

37. Ediger, D., Jiang, K., Riedy, J., Bader, D. A. & Corley, C. Massive Social Network Analysis: Mining Twitter for Social Good. In *2010 39th International Conference on Parallel Processing*, 583–593, DOI: 10.1109/ICPP.2010.66 (IEEE, 2010).

38. Jeong, H., Mason, S. P., Barabási, A.-L. & Oltvai, Z. N. Lethality and centrality in protein networks. *Nature* **411**, 41–42, DOI: 10.1038/35075138 (2001).

39. Shen-Orr, S. S., Milo, R., Mangan, S. & Alon, U. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat. Genet.* **31**, 64–68, DOI: 10.1038/ng881 (2002).

40. Papadopoulos, S., Kompatsiaris, Y., Vakali, A. & Spyridonos, P. Community detection in Social Media. *Data Min. Knowl. Discov.* **24**, 515–554, DOI: 10.1007/s10618-011-0224-z (2012).

41. Chen, J. & Yuan, B. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* **22**, 2283–2290, DOI: 10.1093/bioinformatics/btl370 (2006).

42. Dourisboure, Y., Geraci, F. & Pellegrini, M. Extraction and Classification of Dense Communities in the Web. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, 461–470, DOI: 10.1145/1242572.1242635 (ACM, New York, NY, USA, 2007).

43. Fortunato, S. Community detection in graphs. *Phys. Reports* **486**, 75–174, DOI: 10.1016/j.physrep.2009.11.002 (2010).

44. Newman, M. E. J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582, DOI: 10.1073/pnas.0601602103 (2006).

45. Newman, M. E. J. Analysis of weighted networks. *Phys. Rev. E* **70**, 056131, DOI: 10.1103/PhysRevE.70.056131 (2004).

46. Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113, DOI: 10.1103/PhysRevE.69.026113 (2004).

47. Clauset, A., Newman, M. E. J. & Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111, DOI: 10.1103/PhysRevE.70.066111 (2004).

48. Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110, DOI: 10.1103/PhysRevE.74.016110 (2006).

49. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**, 1118–1123, DOI: 10.1073/pnas.0706851105 (2008).

50. Yucel, M., Muchnik, L. & Hershberg, U. Detection of network communities with memory-biased random walk algorithms. *J. Complex Networks* **5**, 48–69, DOI: 10.1093/comnet/cnw007 (2016).

51. Jiang, P. & Singh, M. SPICi: a fast clustering algorithm for large biological networks. *Bioinforma. (Oxford, England)* **26**, 1105–11, DOI: 10.1093/bioinformatics/btq078 (2010).

52. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008, DOI: 10.1088/1742-5468/2008/10/P10008 (2008).

53. Waltman, L. & van Eck, N. J. A smart local moving algorithm for large-scale modularity-based community detection. *The Eur. Phys. J. B* **86**, 471, DOI: 10.1140/epjb/e2013-40829-0 (2013).

54. Levine, J. *et al.* Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell* **162**, 184–197, DOI: 10.1016/j.cell.2015.05.047 (2015).

55. Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980, DOI: 10.1093/bioinformatics/btv088 (2015).

56. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. biotechnology* **36**, 411–420, DOI: 10.1038/nbt.4096 (2018).

57. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinforma.* **11**, 94, DOI: 10.1186/1471-2105-11-94 (2010).

58. Diaz, A. *et al.* SCell: integrated analysis of single-cell RNA-seq data. *Bioinformatics* **32**, 2219–2220, DOI: 10.1093/bioinformatics/btw201 (2016).

59. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. *PLOS Comput. Biol.* **11**, e1004575, DOI: 10.1371/journal.pcbi.1004575 (2015).

60. Li, P., Piao, Y., Shon, H. S. & Ryu, K. H. Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-Seq data. *BMC Bioinforma.* **16**, 347, DOI: 10.1186/s12859-015-0778-7 (2015).

61. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods* **14**, 565–571, DOI: 10.1038/nmeth.4292 (2017).

62. Patel, A. P. *et al.* Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Sci. (New York, N.Y.)* **344**, 1–9, DOI: 10.1126/science.1254257 (2014).

63. Omohundro, S. M. *Five balltree construction algorithms* (International Computer Science Institute Berkeley, 1989).

64. van Dongen, S. & Enright, A. J. Metric distances derived from cosine similarity and Pearson and Spearman correlations. *arXiv preprint arXiv:1208.3145* (2012). 1208.3145.

65. Jaskowiak, P. A., Campello, R. J. G. B. & Costa, I. G. On the selection of appropriate distances for gene expression data clustering. *BMC bioinformatics* **15 Suppl 2**, S2, DOI: 10.1186/1471-2105-15-S2-S2 (2014).

66. Heng, T. S. P., Painter, M. W. & Immunological Genome Project Consortium. The Immunological Genome Project: networks of gene expression in immune cells. *Nat. immunology* **9**, 1091–4, DOI: 10.1038/ni1008-1091 (2008).

67. Harding, S. D. *et al.* The GUDMAP database–an online resource for genitourinary research. *Dev. (Cambridge, England)* **138**, 2845–53, DOI: 10.1242/dev.063594 (2011).

68. Ritchie, M. E. *et al.* limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **43**, e47–e47, DOI: 10.1093/nar/gkv007 (2015).

69. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci.* **102**, 15545–15550, DOI: 10.1073/pnas.0506580102 (2005).

70. Yaari, G., Bolen, C. R., Thakar, J. & Kleinstein, S. H. Quantitative set analysis for gene expression: a method to quantify gene set differential expression including gene-gene correlations. *Nucleic acids research* **41**, e170, DOI: 10.1093/nar/gkt660 (2013).

71. Dalerba, P. *et al.* Single-cell dissection of transcriptional heterogeneity in human colon tumors. *Nat. biotechnology* **29**, 1120–7, DOI: 10.1038/nbt.2038 (2011).

72. Huang, D. W., Sherman, B. T. & Lempicki, R. a. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* **37**, 1–13, DOI: 10.1093/nar/gkn923 (2009).

73. Klein, A. M. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161**, 1187–1201, DOI: 10.1016/j.cell.2015.04.044 (2015).

74. Uhlen, M. *et al.* Tissue-based map of the human proteome. *Science* **347**, 1260419–1260419, DOI: 10.1126/science.1260419 (2015).

75. Fan, H. C., Fu, G. K. & Fodor, S. P. A. Combinatorial labeling of single cells for gene expression cytometry. *Science* **347**, 1258367–1258367, DOI: 10.1126/science.1258367 (2015).

76. Andoni, A. & Indyk, P. Near-optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. *Commun. ACM* **51**, 117–122, DOI: 10.1145/1327452.1327494 (2008).

77. Bawa, M., Condie, T. & Ganesan, P. LSH Forest: Self-tuning Indexes for Similarity Search. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, 651–660, DOI: 10.1145/1060745.1060840 (ACM, New York, NY, USA, 2005).

78. Wang, M. *et al.* Parallel Clustering Algorithm for Large-Scale Biological Data Sets. *PLoS ONE* **9**, e91315, DOI: 10.1371/journal.pone.0091315 (2014).

79. Hastie, T. & Tibshirani, R. Efficient quadratic regularization for expression arrays. *Biostatistics* **5**, 329–340, DOI: 10.1093/biostatistics/kxh010 (2004).

80. Halko, N., Martinsson, P. G. & Tropp, J. A. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Rev.* **53**, 217–288, DOI: 10.1137/090771806 (2011).

81. van der Maaten, L. & Hinton, G. E. Visualizing High-Dimensional Data Using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

82. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196, DOI: 10.1126/science.aad0501 (2016). arXiv:1011.1669v3.

83. Tenenbaum, J. B., de Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Sci. (New York, N.Y.)* **290**, 2319–23, DOI: 10.1126/science.290.5500.2319 (2000).

84. Series GSE57872. ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE57nnn/GSE57872/suppl/GSE57872_GBM_data_matrix.txt.gz. Accessed 7 Sep 2017.

85. Series GSE65525. http://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE65525&format=file. Accessed 7 Sep 2017.

86. GTEx Portal. http://www.gtexportal.org/home/datasets. Accessed 7 Sep 2017.

87. Durinck, S. *et al.* BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinforma. (Oxford, England)* **21**, 3439–40, DOI: 10.1093/bioinformatics/bti525 (2005).

88. Series GSE63472. ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE63nnn/GSE63472/suppl/GSE63472_P14Retina_merged_digital_expression.txt.gz. Accessed 7 Sep 2017.

89. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

90. Fruchterman, T. M. J. & Reingold, E. M. Graph Drawing by Force-directed Placement. *Softw. Pract. Exper.* **21**, 1129–1164, DOI: 10.1002/spe.4380211102 (1991).

91. Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Syst.* **1695**, 1695 (2006).

92. Desgraupes, B. *clusterCrit: Clustering Indices* (2018).

93. R Core Team. R: A Language and Environment for Statistical Computing (2016).

94. Karatzoglou, A., Smola, A., Hornik, K. & Zeileis, A. kernlab - An S4 Package for Kernel Methods in R. *J. Stat. Softw.* **11**, DOI: 10.18637/jss.v011.i09 (2004).

95. Human Protein Atlas Version 14. http://v14.proteinatlas.org. Accessed 6 August 2018.

96. PhenoGraph repository. https://github.com/jacoblevine/PhenoGraph. Accessed 3 May 2018.

97. SNN-Cliq repository. http://bioinfo.uncc.edu/SNNCliq/. Accessed 3 May 2018.

98. Seurat repository. http://satijalab.org/seurat/. Accessed 3 May 2018.

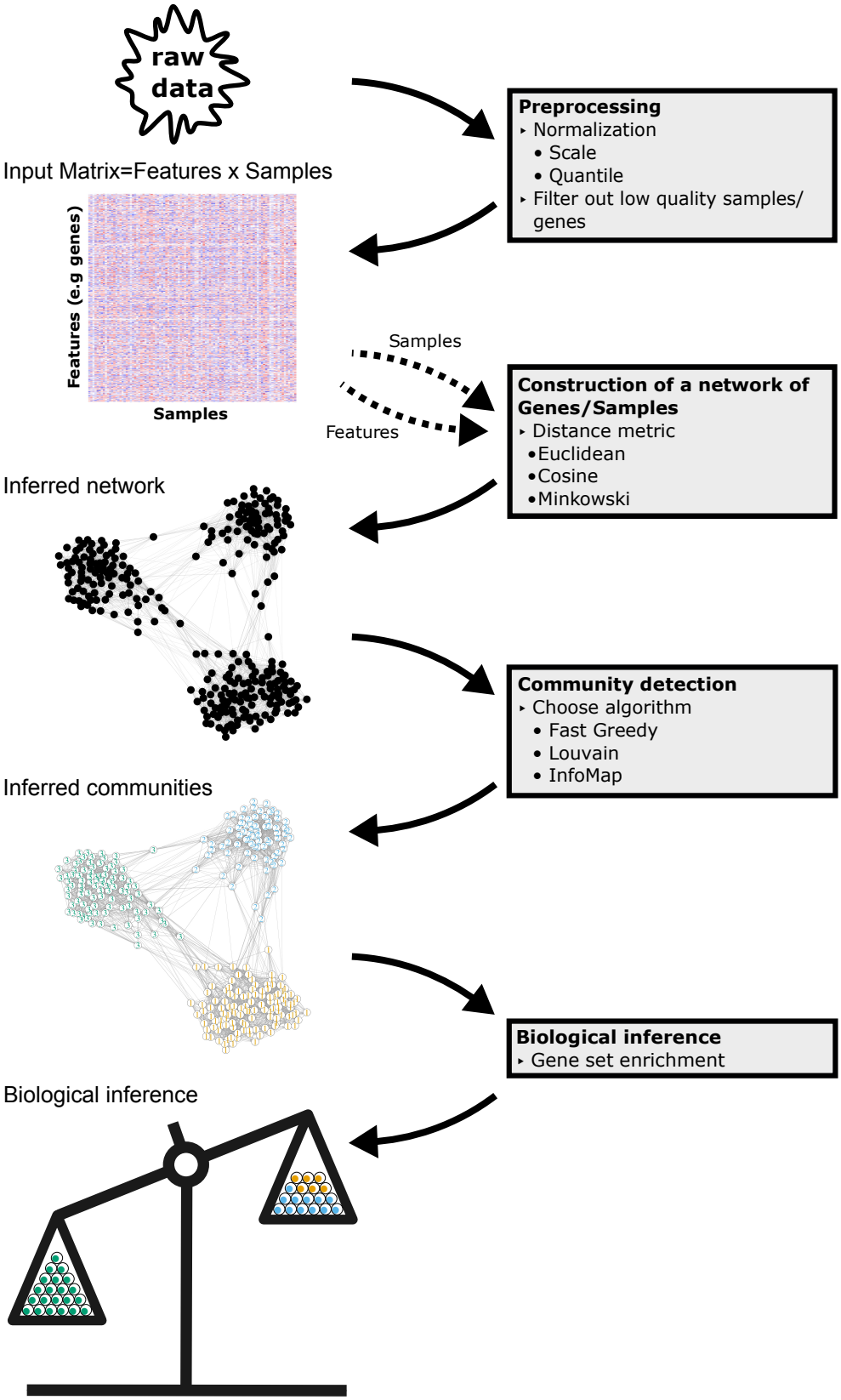

Figure 1. A typical workflow for Networks Based Clustering (NBC). The raw data is first preprocessed to form a gene expression matrix. From this matrix, a weighted KNN network is constructed, in which each node represents a sample (e.g a single cell) or a feature (e.g. a gene). Then, a community detection algorithm is applied to partition the nodes into closely-knit communities, which can be characterized using enrichment strategies.

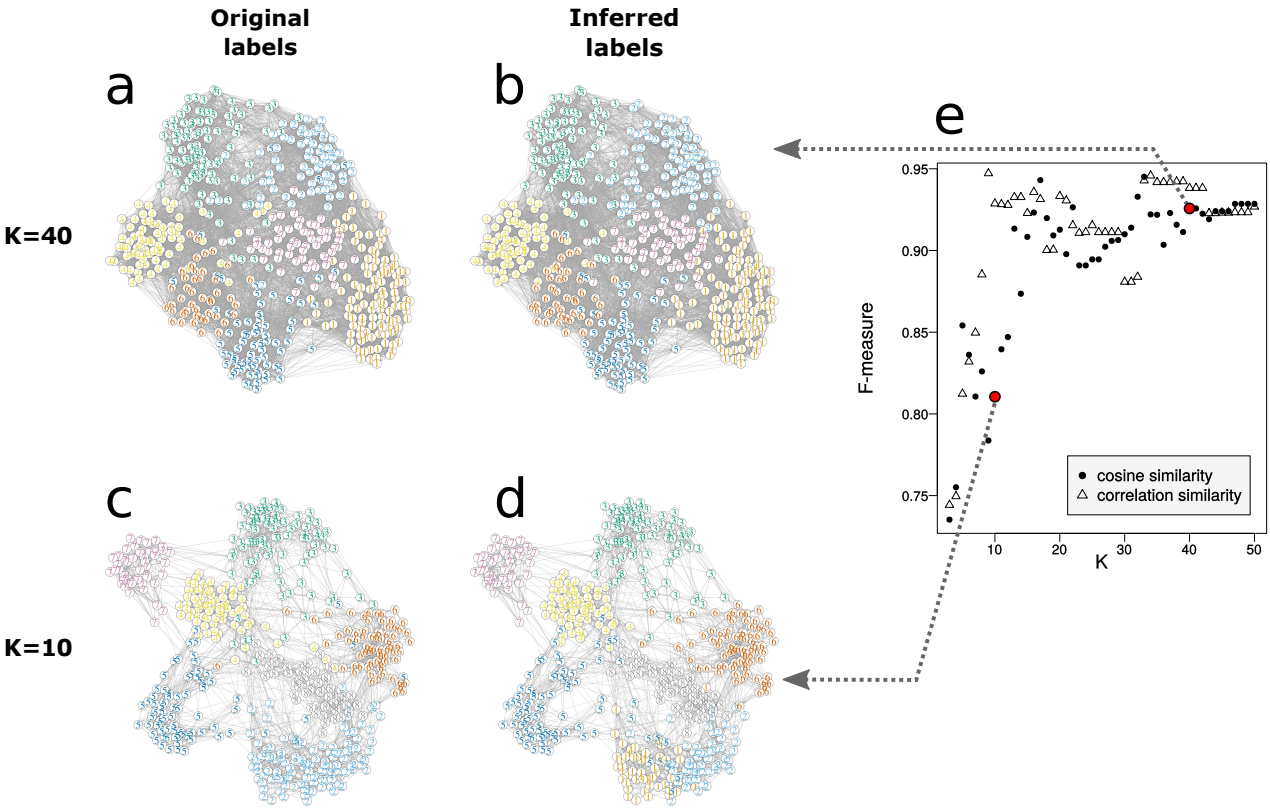

Figure 2. NBC accurately resolves seven cell types from a glioblastoma single-cell RNA-seq dataset. We applied NBC to single-cell RNA-seq data published by Patel *et al.*[62] containing single cells collected from five patients with glioblastoma and two gliomasphere cell lines. A KNN network constructed using cosine similarity with $K = 40$ is shown in (a) and (b). A similar KNN network with $K = 10$ is shown in (c) and (d). Nodes shown in (a) and (c) are color-coded according to cell types reported by the original publication, while nodes in (b) and (d) are color-coded according to communities inferred by the *Louvain* algorithm. (e) The *F-measure*, which is the harmonic mean of precision and recall, is plotted against K for networks constructed with cosine (full circles) and correlation (empty triangles) similarities. Specific points corresponding to $K = 10$ and $K = 40$ are highlighted in red.

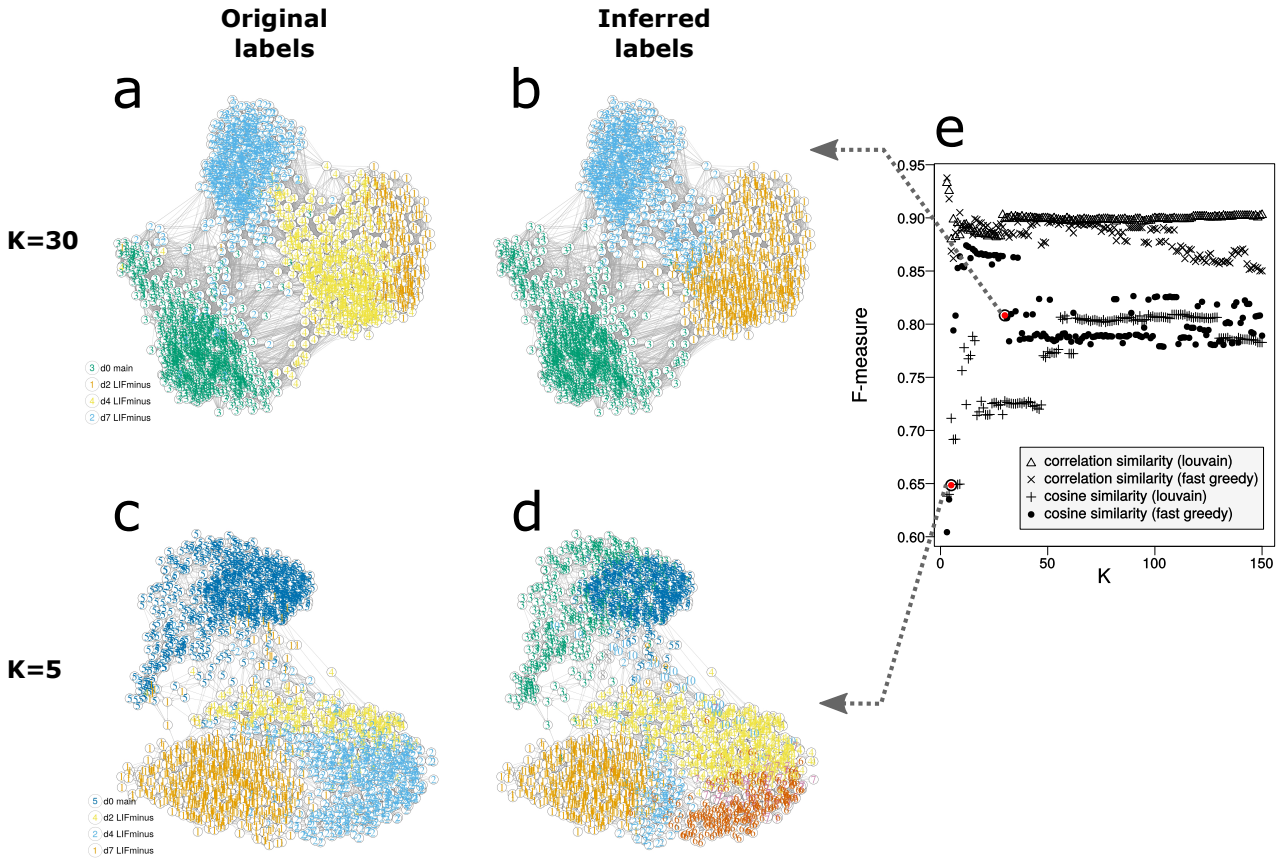

Figure 3. NBC resolves four differentiation stages in a single-cell RNA-seq dataset from mouse embryonic stem cells. We applied NBC to single-cell RNA-seq data published by Klein *et al.*[73] containing single mouse embryonic stem cells collected from four consecutive developmental stages. A KNN network constructed using cosine similarity with $K = 30$ is shown in (a) and (b). A similar KNN network with $K = 5$ is shown in (c) and (d). Nodes shown in (a) and (c) are color-coded according to the differentiation stage reported by the original publication, while nodes in (b) and (d) are color-coded according to communities inferred by the *fast greedy* algorithm. (e) The *F-measure*, which is the harmonic mean of precision and recall, is plotted against K for networks constructed with cosine and correlation similarities and communities inferred by the *fast greedy* and *Louvain* algorithms. Specific points corresponding to $K = 5$ and $K = 30$ are highlighted in red.

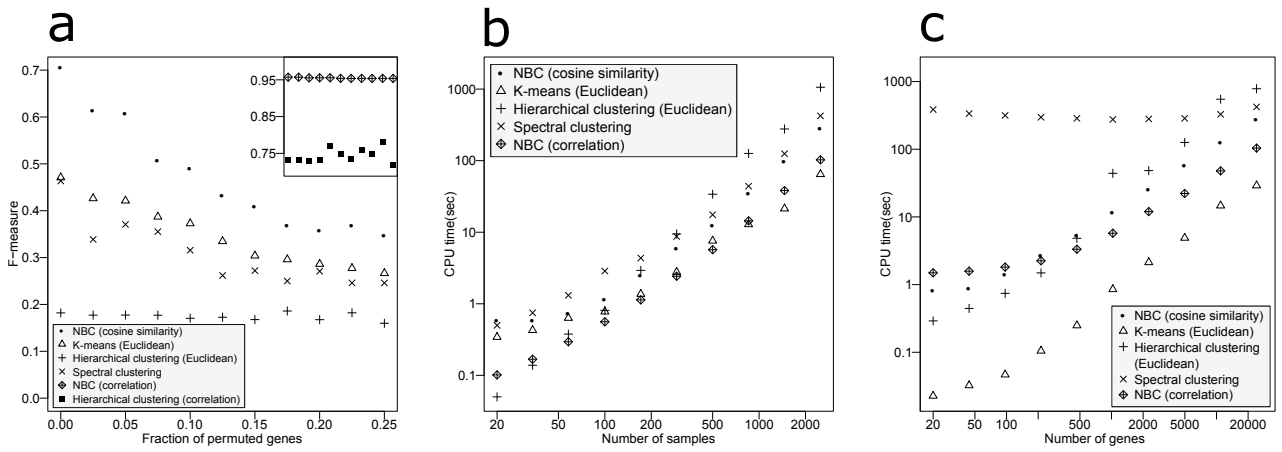

Figure 4. Comparing NBC with other common clustering methods. (a) Shown is a comparison between NBC, K-means, hierarchical clustering, and spectral clustering in terms of the *F-measure* as a function of the effective number of genes. We effectively reduced the number of genes by randomly permuting the sample labels for a fraction of the genes. The inset shows results for NBC and hierarchical clustering with correlation similarity. (b,c) Shown is a comparison of the CPU times required by the five clustering methods, once as a function of the number of randomly chosen samples while keeping the number of genes fixed to 24,071 (b), and once as a function of the number of randomly chosen genes while keeping the number of samples fixed to 2,500 (c). Each point shown is an average of three iterations. Data used in (a) was downloaded from the GTEx consortium[1], and data used in (b) and (c) was taken from a single-cell RNA-seq dataset published by Macosko et al.[13].

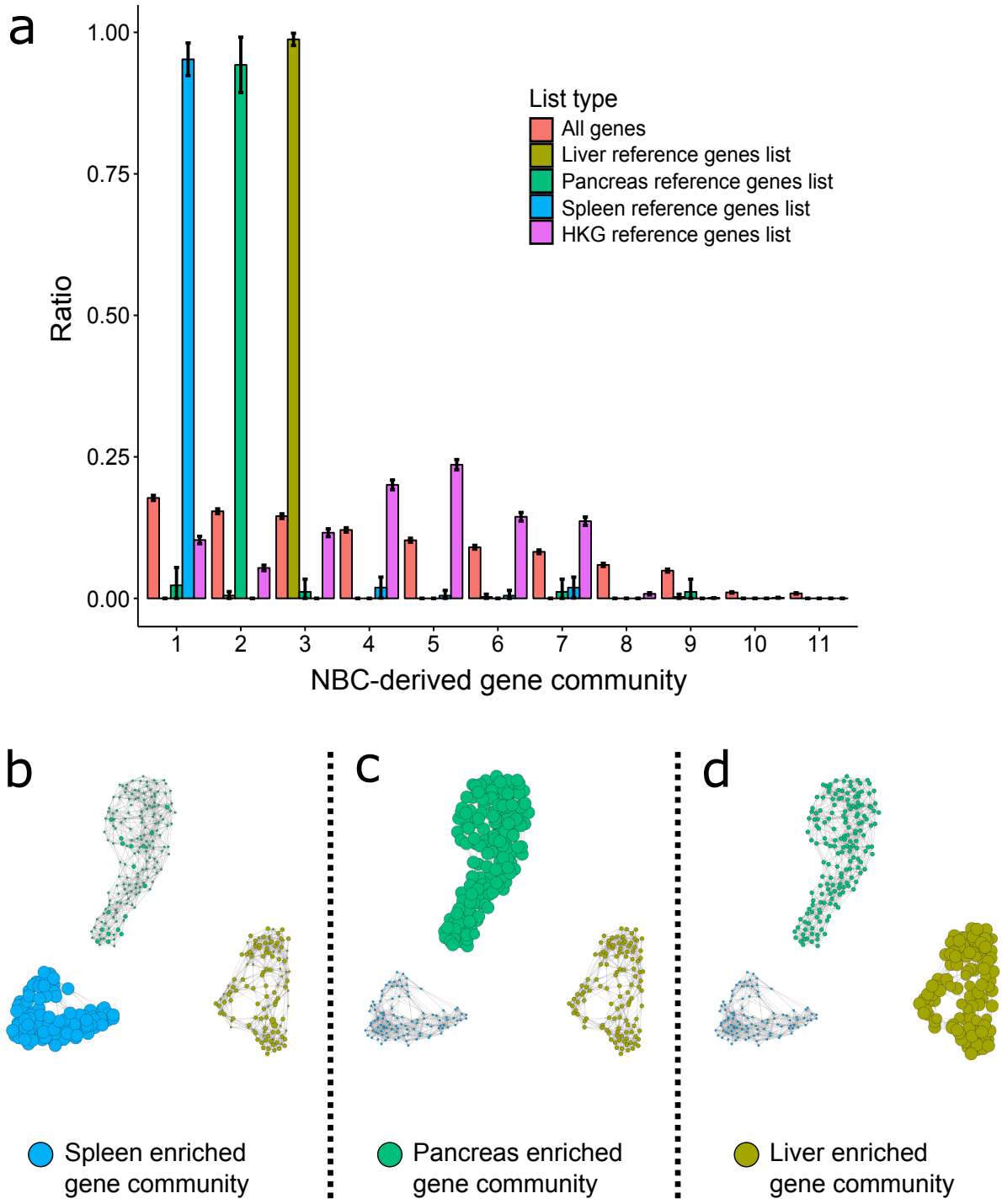

Figure 5. NBC can be used to resolve tissue-specific genes. (a) NBC was applied to a mix of RNA-seq expression profiles from "bulk" samples of the human liver, pancreas, and spleen, that were obtained from the GTEx project[1]. Eleven communities of genes were detected by NBC using the *Louvain* algorithm. For each NBC-derived gene community, the relative fractions of all genes, housekeeping genes (HKG), and three tissue-specific reference genes lists are shown. The tissue-specific reference genes lists were downloaded from the Human Protein Atlas[74]. The NBC-derived gene communities are ordered according to their relative sizes, which is represented by the fraction of total genes that belong to each community (light red bars). (b-d) Shown is the network of samples, where in each panel the node size is proportional to the average log-transformed expression of the genes from NBC-derived community #1 (spleen-enriched, panel b), NBC-derived community #2 (pancreas-enriched, panel c), and NBC-derived community #3 (liver-enriched, panel d). The nodes are color-coded according to their respective tissue type.

Table 1. Examples of previously published methods for NBC.

Name	Reference	Description
PhenoGraph	Manuscript:[54] Software:[96]	A KNN network is constructed using Euclidean distance. Then, a SNN network is constructed by using the number of shared neighbors between every two nodes as a new similarity measure between them. Communities are found using the *Louvain* algorithm.
SNN-Cliq	Manuscript:[55] Software:[97]	A KNN network is constructed using Euclidean distance (or similar). Then, a SNN network is constructed using the number of shared neighbors between every two nodes, as well as their distances to the two nodes, as a new similarity measure. Communities are found using a heuristic approach to find "quasi-cliques" and merge them.
Seurat	Manuscript:[56] Software:[98]	The algorithm constructs a KNN network and then a SNN network. Communities are found using the *Louvain*[52] or the SLM[53] algorithms.

Table 2. Comparison between NBC, K-means, hierarchical clustering, and spectral clustering in terms of the *F-measure* for two single-cell RNAseq datasets. The *F-measure*, which is the harmonic mean between precision and sensitivity, measures the degree to which the inferred communities reproduce the known cell types from the original publication.

Method	Patel et al.[62] dataset (Fig 2)	Klein et al.[73] dataset (Fig 3)
NBC (Cosine similarity, *Louvain* CD)	0.93 ($K = 40$)	0.81 ($K = 30$)
NBC (Correlation similarity, *Louvain* CD)	0.94 ($K = 40$)	0.90 ($K = 30$)
K-means (Euclidean similarity)	0.76	0.84
Hierarchical clustering (Euclidean similarity)	0.86	0.44
Hierarchical clustering (Correlation similarity)	0.79	0.44
Spectral clustering	0.47	0.80

Table 3. Summary statistics for NBC-derived gene communities.

NBC-derived gene community no.	Total number of genes in community	Annotation of enriched tissue-specific "reference" genes list (from Human Protein Atlas)	Enrichment - number of genes from tissue-specific "reference" list (from the Human Protein Atlas) found in this NBC-derived community	p-value for over-expression in corresponding tissue-specific samples (one side t-test with unequal variance)
1	4940	Spleen	200 of 210	2×10^{-61}
2	4284	Pancreas	82 of 87	2×10^{-250}
3	4043	Liver	397 of 403	2×10^{-124}