PINSPlus: A tool for tumor subtype discovery in integrated genomic data

Supplementary Material

Hung Nguyen[1], Sangam Shrestha[1], Sorin Draghici[2], and Tin Nguyen[1,*]

[1]Department of Computer Science and Engineering, University of Nevada, Reno
[2]Department of Computer Science, Wayne State University, Detroit

December 1, 2018

Contents

*tinn@unr.edu

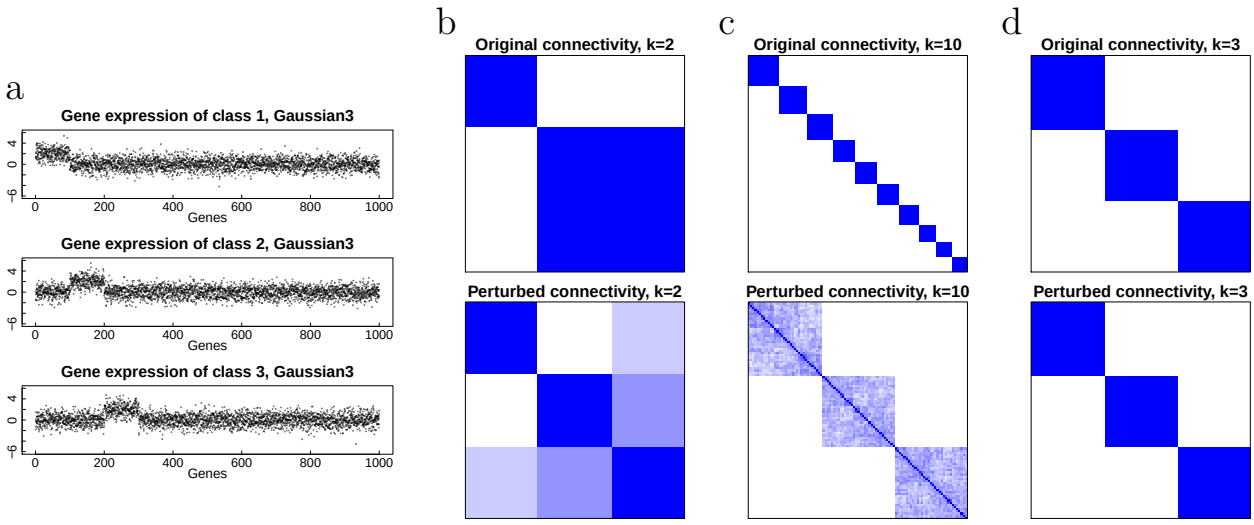

Figure S1: The resilience of pair-wise connectivity. (a) The dataset consists of three classes of patients: the first class has genes $1-100$ up-regulated, the second class has genes $101-200$ up-regulated, and the third class has genes $201-300$ up-regulated. (b) The original connectivity matrix (upper panel) and perturbed connectivity matrix (lower panel) for $k=2$ clusters. Despite setting the wrong number of subtypes ($k=2$), the perturbed connectivity matrix suggests that the data consists of three groups of samples, which is the true structure of the data. (c) The original (upper panel) and the perturbed connectivity (lower panel) matrices for $k=10$. Again, even if the number of clusters is incorrect ($k=10$ this time), the perturbed connectivity matrix still has three big blocks suggesting that the data consists of three groups of samples. (d) The original and perturbed connectivity matrices for $k=3$. The agreement between the original and perturbed connectivity strongly suggests the structure of the data.

1 Algorithm and implementation

1.1 Connectivity resilience

Our hypothesis is that if well-defined subtypes of a disease exist, these subtypes have to be stable with respect to small changes in the measured values. This is indeed the case and we will demonstrate that the pair-wise connectivity between patients that truly belong to the same subtype tends to be preserved when the data is perturbed (Figure S1). In this example, we have three distinct classes of patients (Figure S1a). We aim to discover the subtypes with an algorithm as simple as k-means. Assuming that we do not know the correct number of subtypes, we set the number of subtypes to $k=2$. The upper panel in Figure S1b shows the connectivity between patients after clustering: blue when they belong to the same cluster, and white otherwise. Now we perturb the molecular measurements and repeatedly perform clustering and partition the patients (with $k=2$). The lower panel in Figure S1b shows the combined connectivity of all perturbed connectivities between patients. The visualization of the perturbed connectivity matrix clearly suggests that the larger cluster is not stable. Similarly, we partition the patients using $k=10$ as the number of subtypes (Figure S1c). The discordant connectivity again states that this partitioning does not reflect the true structure of the data. More interestingly, the perturbed connectivity matrices for both cases (lower panels in Figure S1b,c) clearly suggest that there are three distinct classes of patients. Finally, when we set $k=3$ as the number of subtypes, the perturbed and the original connectivity matrices are identical (Figure S1d). This resilience of the patient connectivities occurs consistently regardless of the clustering algorithm being used (e.g., k-means, hierarchical clustering, partitioning around medoids, etc.), or the distribution of the data.

1.2 Perturbation clustering and stopping criterion

In the perturbation clustering algorithm proposed by Nguyen et al. [11], for each number of cluster $k \in \{2, 3, ...10\}$, perturbation process perturbs the original data then performs clustering on perturbed data in a finite number of times n, for example, $n = 200$, to generate the perturbed connectivity matrices. The algorithm then calculates the difference between the original and the perturbed connectivity matrices and computes the empirical cumulative distribution functions of the difference matrix (CDF-DM). The area under the CDF-DM curve AUC_k is used to assess the stability of the partitioning. In the ideal case when the original and the perturbed connectivity matrices are identical, the difference matrix consists of only zero values, yielding a CDF-DM that jumps from 0 to 1 at the origin, and an AUC value of 1.

The perturbation clustering is very robust against noisy high-throughput data. However, the algorithm is slow due to the large number of perturbations needed to obtain the optimal k and AUC_k. For example, it takes 25 minutes to analyze mRNA, methylation, and miRNA data of the kidney renal clear cell carcinoma (KIRC) dataset with 124 patients. Here we optimize the algorithm to significantly reduce the analysis time. For the same dataset (KIRC, 124 patients), the running time is reduced to less than a minute.

Figure 1C in the main text shows the AUC values after each iteration for mRNA and methylation data of the KIRC dataset. For each data type, the AUC values tend to converge after a certain number of iterations, which means that at some point, additional iterations are not necessary. PINSPlus makes use of this advantage in order to determine an early stopping point for the perturbation clustering. As a result, the iteration can stop much earlier before it reaches the maximum number of iterations but still guarantees the quality of perturbed connectivity matrices. More specifically, the perturbation process will stop if: i) after the first 20 iterations, there exists a k for which $AUC_k = 1$, or ii) within all values of k, the variance of the last 20 iterations is smaller than 10^{-6}, i.e., $\frac{\sum_{i-20}^{i}(AUC_i-\mu)^2}{20} < 10^{-6}$ where $\mu = \frac{\sum_{i-20}^{i}(AUC_i)}{20}$. Figure 1C1 shows the first scenario, for which all perturbation processing for every k stops when the number of iterations $i = 20$ because $AUC_2 = 1$. Figure 1C2 shows the second scenario for which the AUC values barely change after 20 iterations before the stopping points (triangle symbols).

1.3 Parallel programming

PINSPlus makes use of multi-core processing to speed up the perturbation processing. The iterations in the perturbation processing are now assigned for different cores of the CPU. Many existing clustering approaches are sensitive to the number of threads being used, leading to different results with different numbers of threads. PINSPlus implements multi-core feature in a way such that the result is stable regardless of the number of cores being used.

1.4 Customizable algorithm

By default, PINSPlus uses k-means as the basic clustering algorithm and Gaussian noise as the method of perturbation. To make PINSPlus more flexible, we also implemented hierarchical clustering and partitioning around medoids [7] as built-in alternatives to k-means. For advanced users, PINSPlus allows passing any customized clustering function as a parameter. For data perturbation, we also implemented a subsampling approach as an alternative method to Gaussian noise. Advanced users can also pass a customized perturbation function as a parameter.

1.5 Cluster ensemble and two-stage clustering

Let us consider T data types from N patients. In the first stage, PINSPlus works with each data type to build T connectivity matrices, one for each data type. A connectivity matrix can be represented as a graph, with patients as nodes, and connectivity between patients as edges. Our goal is to identify subgraphs that are strongly connected across all data types. We merge the T connectivity matrices into a combined similarity matrix that represents the overall connectivity between patients. This matrix is used as an input for similarity-based clustering algorithms, such as hierarchical clustering and partitioning around medoids [7]. We then choose the partitioning most agrees with the partitionings of individual data types [13]. This completes Stage I.

In Stage II, we consider each group one at a time and decide whether to split it further. We expect the splitting algorithm to work effectively when the data has a hierarchical structure, i.e., there are subgroups of patients within discovered subtypes. Since our method is an unsupervised approach, we do not have prior information to take into account important covariates, such as gender, race, or demographic. If these signals are predominant, we are likely to miss the real subtypes. Another motivation is that there are often heterogeneous subgroups of patients that share clinically relevant characteristics even within a subtype. One example is that Luminal A and Luminal B are both estrogen receptor positives. If the data follows a hierarchical structure, the distances between subgroups at the second level are smaller than those between groups at the first level. Therefore, one-round clustering would likely overlook the subgroups within the groups identified in Stage I. To avoid over-splitting the subtypes, we impose some conditions before proceeding to Stage II. First, Stage I clustering has to be extremely imbalanced. Second, the splitting must be supported by a strong signal across all data types. In both cases, it is worth reviewing the data to see if each of the discovered groups can be further split. The software returns the result of both rounds, so users can investigate both groupings for discovery. Figure S2 demonstrates and example using the dataset KIRC.

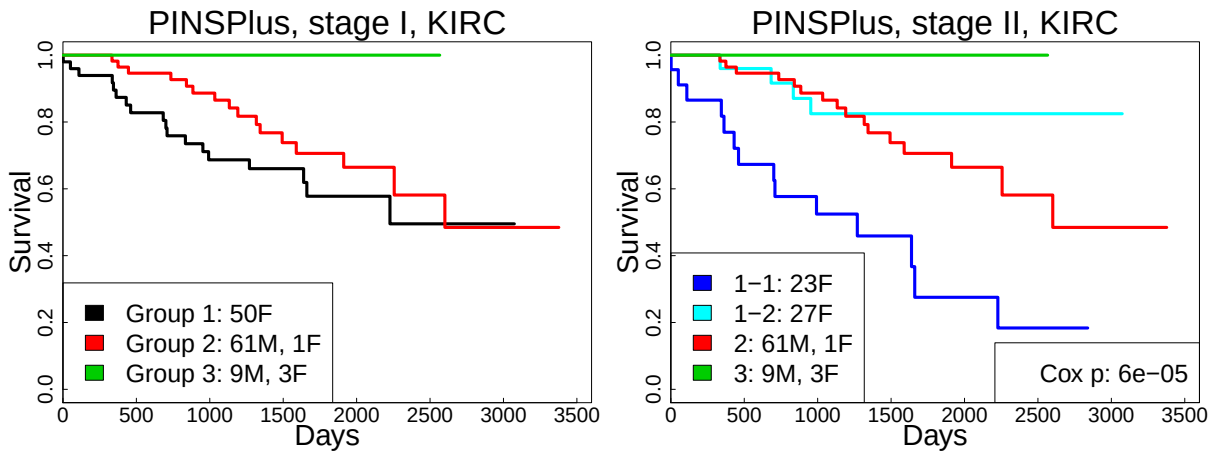

Figure S2: Kaplan-Meier survival analysis for kidney renal clear cell carcinoma (KIRC). The horizontal axes represent the time passed after entry into the study while the vertical axes represent estimated survival percentage. The left and right panels show subtypes discovered by PINS in stage I and stage II, respectively.

1.6 Choosing a suitable clustering method

PINSPlus uses k-means as default, and it has been shown to work well in our analysis of 8 mRNA and 36 omics datasets. However, in theory, k-means is not without flaws. For example, k-means might be sensitive to outliers and is not designed to discover hierarchical data structures. Therefore, we provide pam and hclust as alternative build-in algorithms. We will show examples in which one method performs well in one scenario might not be the best choice in another scenario. We note that these examples are not supposed to provide a thorough comparison between the three methods (k-means, hclust, and pam), but to provide some guidance for a better use of PINSPlus.

Generally, if the groups are well separated, any clustering algorithm would perform well. This ideal scenario is shown in Figure S3a. In this example, there are 3 groups of samples with different sets of up-regulated genes. The expression values of the up-regulated genes are very different from those of un-regulated genes. As shown in the first 2 principal components, the groups are well separated. All of the three methods perform well in this ideal case.

When the distances between the groups decrease, we notice that k-means is a more robust choice. As shown in Figure S3b, k-means performs very well even when the three groups are close to one another. One likely reason is that the cluster centers are very stable to data perturbation. When the data is perturbed, each data point moves around its original position. However, these random effects from multiple data points are canceled out and the cluster centers do not vary drastically, leading to a very stable k-means grouping.

Since hclust tries to force the data into a hierarchy, the structure changes every time the data is perturbed. Therefore, hclust tends to increase the number of clusters to seek for stability. The algorithm pam differs from k-means in the way that it uses medoids to represent clusters (instead of arithmetic centers). When the data is perturbed, the medoids move around and are unstable, leading to unstable pam groupings.

In some cases, when the data has a hierarchical structure, hclust is expected to perform better than k-means. If the data follows a hierarchical structure, the distances between subgroups at the second level are smaller than the distances between groups at the first level. Therefore, k-means probably can only identify the groups at the first level. Figure S3c shows an example in which the distance between groups 2 and 3 and between groups 4 and 5 are much smaller than the distance between group 1 and the rest. As shown in the principal components, the difference between groups 2 and 3 are not distinguishable when we look at the data altogether. In this case, both k-means and pam are unable to discover the true structure of the data. On the contrary, hclust perfectly separates the groups.

Figure S3d shows an example in which pam is the best choice. Note that in this scenario, the data are well separated and each group has approximately the same number of data points. We added some outliers in order to test the robustness of each clustering method. In this case, pam provides a perfect grouping while k-means and hclust are sensitive to outliers and are unable to identify the correct number of groups.

2 Data processing

2.1 Gene expression data

For this single data type analysis, we download 8 gene expression datasets, from a variety of human cancers with known classes (subtypes). Details of the 8 datasets are described in Table S1. The 5 datasets GSE10245, GSE19188, GSE43580, GSE15061, and GSE14924 were downloaded from Gene Expression Omnibus (`www.ncbi.nlm.nih.gov/geo/`). The other three datasets were downloaded from the Broad Institute: Lung2001 (`www.broadinstitute.org/mpr/lung/`), AML2004 (`www.broadinstitute.org/cancer/pub/nmf`), and Brain2002 (`www.broadinstitute.org/MPR/CNS/`). The dataset AML2004 was already processed and normalized and thus no further data processing was needed. For the other 7 datasets, Affymetrix *CEL* files containing raw expression data were downloaded and processed and normalized using the *threestep* function from the package *affyPLM version 1.38.0* [2].

Table S1: Description of the eight mRNA datasets used in our analysis. The top five datasets were downloaded from the Gene Expression Omnibus. The bottom three datasets were downloaded from the Broad Institute website.

Datasets	#Class	#Sample	#Feature	Platform	Description
GSE10245 [8]	2	58	19851	hgu133plus2	40 adenocarcinomas and 18 squamous cell carcinomas
GSE19188 [6]	3	91	19851	hgu133plus2	45 adenocarcinomas, 19 large cell carcinomas, and 27 squamous cell carcinomas
GSE43580 [14]	2	150	19851	hgu133plus2	77 adenocarcinomas and 73 squamous cell carcinomas
GSE14924 [9]	2	20	19851	hgu133plus2	10 acute myeloid leukemia CD4 T cell and 10 CD8 T cell
GSE15061 [10]	2	366	19851	hgu133plus2	202 acute myeloid leukemia samples and 164 myelodyplastic syndrome samples
Lung2001 [1]	4	237	8641	hgu95a	190 adenocarcinomas, 21 squamous cell carcinomas, 20 carcinoid, and 6 small-cell lung carcinomas
AML2004 [5, 3]	3	38	5000	hgu6800	11 acute myeloid leukemia, 19 acute lymphoblastic leukemia B cell, and 8 T cell
Brain2002 [12]	5	42	5299	hgu6800	10 meduloblastomas, 10 malignant gliolas, 10 atypical teratoid/rhaboid tumors, 4 normal cerebellums, and 8 primitive neuroectodermal tumors

2.2 TCGA and METABRIC data

We analyzed 34 different types of cancer with curated level three data, available at The Cancer Genome Atlas datasets (TCGA) website (`cancergenome.nih.gov` and `firebrowse.org`): Kidney renal clear cell carcinoma (KIRC), Glioblastoma multiforme (GBM), Acute Myeloid Leukemia (LAML), Lung squamous cell carcinoma (LUSC), Bladder Urothelial Carcinoma (BLCA), Head and Neck squamous cell carcinoma (HNSC),

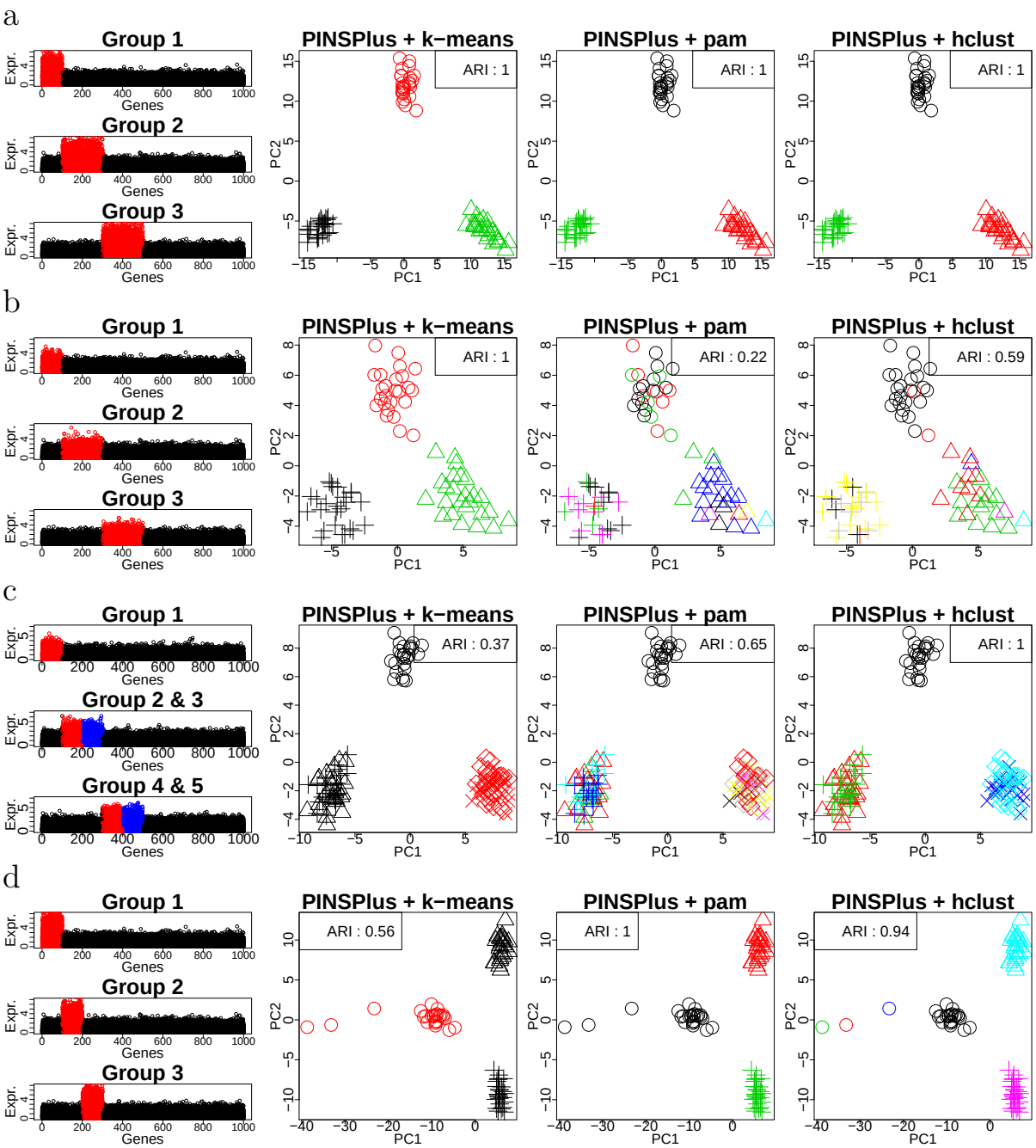

Figure S3: Examples to demonstrate the strength and weakness of each clustering method. In each row, the most left panel shows the data while the three remaining panels show the clustering results of PINSPlus in conjunction with k-means, pam, and hclust, respectively. (a) All clustering methods perform well when the clusters are well-separated. (b) k-means outperforms other methods when the clusters are close to one another. (c) When the data has a hierarchical structure, hclust should be the best choice. (d) In presence of outliers, pam outperforms k-means and hclust.

Liver hepatocellular carcinoma (LIHC), Stomach adenocarcinoma (STAD), Thymoma (THYM), Glioma (GBMLGG), Brain Lower Grade Glioma (LGG), Pancreatic adenocarcinoma (PAAD), Skin Cutaneous Melanoma (SKCM), Colorectal adenocarcinoma (COADREAD), Uterine Corpus Endometrial Carcinoma (UCEC), Cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC), Colon adenocarcinoma (COAD), Breast invasive carcinoma (BRCA), Stomach and Esophageal carcinoma (STES), Kidney renal papillary cell carcinoma (KIRP), Kidney Chromophobe (KICH), Uveal Melanoma (UVM), Adrenocortical carcinoma (ACC), Sarcoma (SARC), Mesothelioma (MESO), Rectum adenocarcinoma (READ), Uterine Carcinosarcoma (UCS), Ovarian serous cystadenocarcinoma (OV), Esophageal carcinoma (ESCA), Paraganglioma (PCPG), Lung adenocarcinoma (LUAD), Prostate adenocarcinoma (PRAD), Thyroid carcinoma (THCA), and Testicular Germ Cell Tumors (TGCT). We used mRNA expression, DNA methylation, and miRNA expression data for each of the 34 cancers. Table S3 shows the details of each dataset.

The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) data [4] consists of a discovery cohort (997 patients) and a validation cohort (983 patients). For each of these patients, matched DNA and RNA were subjected to copy number analysis and transcriptional profiling on the Affymetrix SNP 6.0 and Illumina HT 12 v3 platforms, respectively. We downloaded the mRNA and copy number variation (CNV) data from the European Genome-Phenome Archive (`www.ebi.ac.uk/ega/`) and high quality follow up clinical data from cBioPortal (`www.cbioportal.org`). There are patients that were followed up upon for almost 30 years. The only preprocessing done was mapping CNVs to genes using the CNTools package [15].

3 Experimental results

The data analysis is done on a Debian Linux server that has 376GB of RAM, and multi-core CPU (32 cores, 2 sockets, 16 cores/socket, 2 threads/core, Intel Xeon Gold 6130, 2.10GHz). The *R* session and packages information is presented as below:

- R version 3.4.3 (2017-11-30), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 16.04.4 LTS`
- Matrix products: default
- BLAS: `/usr/lib/libblas/libblas.so.3.6.0`
- LAPACK: `/usr/local/lib/R/lib/libRlapack.so`
- Base packages: base, datasets, graphics, grDevices, grid, methods, parallel, stats, stats4, utils
- Other packages: cluster 2.0.7-1, ConsensusClusterPlus 1.46.0, doParallel 1.0.11, entropy 1.2.1, flexclust 1.3-5, foreach 1.4.4, future 1.8.0, iClusterPlus 1.18.0, iterators 1.0.9, lattice 0.20-35, modeltools 0.2-21, pbmcapply 1.2.4, PINSPlus 1.0.2, SNFtool 2.3.0, survival 2.42-3
- Loaded via a namespace (and not attached): Biobase 2.38.0, BiocGenerics 0.24.0, codetools 0.2-15, compiler 3.4.3, digest 0.6.15, globals 0.11.0, heatmap.plus 1.3, listenv 0.7.0, Matrix 1.2-14, splines 3.4.3, tools 3.4.3

3.1 Gene expression data

In order to validate PINSPlus with single data type analysis, we first tested it using eight real datasets with known subtypes from Gene Expression Omnibus and Broad Institute. Table S4 presents the results

Table S2: Description of the 34 datasets from The Cancer Genome Atlas (TCGA)

Dataset	#Sample	mRNA	Methylation	miRNA
KIRC	124	HiSeq RNASeq	Methylation27	GASeq miRNASeq
GBM	273	HT HG-U133A	Methylation27	HiSeq miRNASeq
LAML	164	GASeq RNASeq	Methylation27	GASeq miRNASeq
LUSC	110	HT HG-U133A	Methylation27	GASeq miRNASeq
BLCA	404	HiSeq RNASeq v2	Methylation450	GASeq miRNASeq
HNSC	228	HiSeq RNASeq	Methylation450	HiSeq miRNASeq
LIHC	366	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
STAD	362	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
THYM	654	HiSeq RNASeq v2	Methylation450	GASeq miRNASeq
GBMLGG	654	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
LGG	510	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
PAAD	178	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
SKCM	439	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
COADREAD	294	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
UCEC	234	GASeq RNASeq v2	Methylation450	HiSeq miRNASeq
CESC	304	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
COAD	220	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
BRCA	622	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
STES	545	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
KIRP	271	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
KICH	65	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
UVM	80	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
ACC	79	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
SARC	257	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
MESO	86	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
READ	74	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
UCS	56	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
OV	286	HiSeq RNASeq v2	Methylation27	HiSeq miRNASeq
ESCA	183	HiSeq RNASeq	Methylation450	HiSeq miRNASeq
PCPG	179	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
LUAD	428	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
PRAD	493	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
THCA	499	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq
TGCT	134	HiSeq RNASeq v2	Methylation450	HiSeq miRNASeq

Table S3: Description of the 2 datasets from The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC): METABRIC discovery and METABRIC validation.

Dataset	#Sample	mRNA	CNV
Discovery	997	Illumina HT 12 v3	Affymetrix SNP 6.0
Validation	983	Illumina HT 12 v3	Affymetrix SNP 6.0

produced from PINSPlus, CC, SNF, and iClusterPlus. We note that for *iClusterPlus* datasets, only the top 4000 components were used due to its time complexity.

Table S4: The performance of PINS, PINSPlus, Consensus Clustering (CC), Similarity Network Fusion (SNF), and iClusterPlus in discovering subtypes from gene expression data. For each dataset (row), cells highlighted in green have the highest Rand Index (RI), and Adjusted Rand Index (ARI). For all 8 datasets, PINSPlus outperforms its competitors by having the highest RI and ARI. SNF produced an error for GSE14924, and iClusterPlus produced an error for AML2004, shown as an NA value.

Dataset			PINS/PINS+			CC			SNF			iCluster+		
Name	Samples	#Class	k	RI	ARI	k	RI	ARI	k	RI	ARI	k	RI	ARI
GSE10245	58	2	2	0.90	0.80	6	0.64	0.32	2	0.69	0.38	4	0.58	0.22
GSE19188	91	3	3	0.84	0.66	4	0.82	0.6	4	0.61	0.12	9	0.67	0.19
GSE43580	150	2	2	0.72	0.44	3	0.68	0.37	2	0.58	0.15	5	0.61	0.21
GSE15061	366	2	2	0.83	0.65	6	0.72	0.43	2	0.53	0.05	10	0.57	0.15
GSE14924	20	2	2	1.00	1.00	7	0.64	0.25	NA	NA	NA	3	0.87	0.73
Lung2001	237	4	2	0.82	0.54	8	0.46	0.11	3	0.62	0.28	7	0.45	0.11
AML2004	38	3	4	0.85	0.65	5	0.81	0.56	2	0.59	0.17	NA	NA	NA
Brain2002	42	5	7	0.89	0.61	5	0.8	0.46	2	0.57	0.13	4	0.74	0.32

3.2 TCGA and METABRIC data

To validate PINSPlus using multi-omics data, we tested it using 34 TCGA datasets and two METABRIC datasets. The results are reported in Table 1 of the main text. There are 9 datasets for which no method is able to identify subtypes with significantly different survival (READ, UCS, OV, ESCA, PCPG, LUAD, PRAD, THCA, TGCT). For the remaining 27 datasets, PINSPlus has significant p-values in all of them whereas CC, SNF, and iClusterPlus has significant p-values in only in 8, 14, and 9 datasets, respectively. More importantly, PINSPlus has the most significant p-values in 23 datasets (out of 27).

3.3 Running time

Table S5 shows the running time of each method for the 34 datasets. For gene expression data, PINSPlus, CC, and SNF can finish each analysis in less than a minute while it takes iClusterPlus several hours. The gap in running time is much larger for data integration. PINSPlus, CC, and SNF can integrate omics data and partition hundreds of patients in minutes while iClusterPlus (with 60 cores) takes up to many hours to analyze large datasets.

Table S5: *Running time of each subtyping method. The time is rounded to minutes (min). CC and SNF can only run on 1 core while PINSPlus and iClusterPlus allow for parallel computing.*

Consortium	Dataset	#Patient	PINS 1 core	PINS+ 2 cores	CC 1 core	SNF 1 core	iCluster+ 60 cores
GEO&Broad	GSE10245	58	<1m	<1m	<1m	<1m	19m
	GSE19188	91	1m	<1m	<1m	<1m	29m
	GSE43580	150	2m	<1m	<1m	<1m	50m
	GSE15061	366	12m	<1m	<1m	<1m	100m
	GSE14924	20	<1m	<1m	<1m	<1m	9m
	Lung2001	237	5m	<1m	<1m	<1m	58m
	AML2004	38	<1m	<1m	<1m	<1m	NA
	Brain2002	42	<1m	<1m	<1m	<1m	16m
TCGA	KIRC	124	6m	<1m	<1m	<1m	95m
	GBM	273	53m	1m	<1m	<1m	190m
	LAML	164	10m	<1m	<1m	<1m	123m
	LUSC	110	5m	<1m	<1m	<1m	59m
	BLCA	404	112m	6m	3m	3m	433m
	HNSC	228	32m	4m	3m	2m	101m
	LIHC	366	96m	5m	4m	3m	263m
	STAD	362	97m	5m	4m	3m	299m
	THYM	119	6m	1m	2m	1m	95m
	GBMLGG	510	192m	7m	7m	4m	392m
	LGG	510	188m	12m	8m	6m	274m
	PAAD	178	20m	3m	2m	1m	176m
	SKCM	439	144m	8m	3m	3m	202m
	COADREAD	294	61m	5m	4m	3m	157m
	UCEC	234	34m	4m	4m	2m	201m
	CESC	304	60m	7m	5m	2m	203m
	COAD	220	30m	3m	3m	2m	126m
	BRCA	622	236m	16m	10m	5m	285m
	STES	545	171m	12m	14m	5m	324m
	KIRP	271	33m	3m	3m	1m	184m
	KICH	65	4m	1m	1m	<1m	58m
	UVM	80	3m	1m	1m	1m	71m
	ACC	79	3m	1m	1m	<1m	63m
	SARC	257	43m	5m	3m	1m	201m
	MESO	86	4m	1m	2m	<1m	72m
	READ	74	3m	1m	2m	<1m	52m
	UCS	56	2m	1m	1m	<1m	32m
	OV	286	52m	2m	2m	1m	188m
	ESCA	183	23m	5m	5m	2m	204m
	PCPG	179	16m	2m	3m	1m	244m
	LUAD	428	128m	8m	5m	3m	233m
	PRAD	493	205m	11m	10m	5m	276m
	THCA	499	213m	10m	5m	3m	251m
	TGCT	134	9m	2m	2m	1m	105m
METABRIC	Discovery	997	1153m	9m	15m	2m	350m
	Validation	983	581m	8m	14m	2m	348m

References

[1] A Bhattacharjee, WG Richards, J Staunton, C Li, S Monti, P Vasa, C Ladd, J Beheshti, R Bueno, M Gillette, M Loda, G Weber, EJ Mark, ES Lander, W Wong, BE Johnson, TR Golub, DJ Sugarbaker, and M Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–5, Nov. 2001.

[2] Benjamin Milo Bolstad. *Low-level analysis of high-density oligonucleotide array data: background, normalization and summarization*. PhD thesis, University of California, 2004.

[3] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, March 2004.

[4] Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, Stefan Graf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

[5] Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, Clara D Bloomfield, and Eric S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[6] Jun Hou, Joachim Aerts, Bianca Den Hamer, Wilfred Van Ijcken, Michael Den Bakker, Peter Riegman, Cor van der Leest, Peter van der Spek, John A Foekens, Henk C Hoogsteden, Frank Grosveld, and Sjaak Philipsen. Gene expression-based classification of non-small cell lung carcinomas and survival prediction. *PLoS ONE*, 5(4):e10312, 2010.

[7] Leonard Kaufman and Peter Rousseeuw. Clustering by Means of Medoids. In Yadolah Dodge, editor, *Statistical Data Analysis Based on the L1–Norm and Related Methods*, pages 405–416. North-Holland, Amsterdam, 1987.

[8] Ruprecht Kuner, Thomas Muley, Michael Meister, Markus Ruschhaupt, Andreas Buness, Elizabeth C Xu, Phillipp Schnabel, Arne Warth, Annemarie Poustka, Holger Sultmann, and Hans Hoffmann. Global gene expression analysis reveals specific patterns of cell junctions in non-small cell lung cancer subtypes. *Lung Cancer*, 63(1):32–38, 2009.

[9] Rifca Le Dieu, David C. Taussig, Alan G. Ramsay, Richard Mitter, Faridah Miraki-Moud, Rewas Fatah, Abigail M. Lee, T. Andrew Lister, and John G. Gribben. Peripheral blood T cells in acute myeloid leukemia (AML) patients at diagnosis have abnormal phenotype and genotype and form defective immune synapses with AML blasts. *Blood*, 114(18):3909–3916, October 2009.

[10] Ken I Mills, Alexander Kohlmann, P Mickey Williams, Lothar Wieczorek, Wei-min Liu, Rachel Li, Wen Wei, David T Bowen, Helmut Loeffler, Jesus M Hernandez, Wolf-Karsten Hofmann, and Torsten Haferlach. Microarray-based classifiers and prognosis models identify subgroups with distinct clinical outcomes and high risk of AML transformation of myelodysplastic syndrome. *Blood*, 114(5):1063–1072, 2009.

[11] Tin Nguyen, Rebecca Tagett, Diana Diaz, and Sorin Draghici. A novel approach for data integration and disease subtyping. *Genome Research*, 27(12):2025–2039, 2017.

[12] SL Pomeroy, P Tamayo, M Gaasenbeek, LM Sturla, M Angelo, ME McLaughlin, JY Kim, LC Goumnerova, PM Black, C Lau, JC Allen, D Zagzag, JM Olson, T Curran, C Wetmore, JA Biegel, T Poggio, S Mukherjee, R Rifkin, A Califano, G Stolovitzky, DN Louis, JP Mesirov, ES Lander, and TR Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, January 2002.

[13] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[14] Adi L Tarca, Mario Lauria, Michael Unger, Erhan Bilal, Stephanie Boue, Kushal Kumar Dey, Julia Hoeng, Heinz Koeppl, Florian Martin, Pablo Meyer, Preetam Nandy, Raquel Norel, Manuel Peitsch, Jeremy J Rice, Roberto Romero, Gustavo Stolovitzky, Marja Talikka, Yang Xiang, Christoph Zechner, and IMPROVER DSC Collaborators. Strengths and limitations of microarray-based phenotype prediction: lessons learned from the IMPROVER diagnostic signature challenge. *Bioinformatics*, 29(22):2892–2899, 2013.

[15] Jianhua Zhang. *CNTools: Convert segment data into a region by sample matrix to allow for other high level computational analyses*, 2014.