Software Manual for iOmicsPASS: integrative -Omics Tool for Predictive Analysis with Subnetwork Signatures

Version 1.0

Authors

Hiromi W.L. Koh & Hyungwon Choi

Last Modified

June 13, 2018

Mantainer: Hiromi W.L. Koh
E-mail: hiromikwl@gmail.com

Contents

1 Installation **2**
- 1.1 Installing and setting up R . . . 2
- 1.2 Installing iOmicsPASS . . . 2

2 Introduction to iOmicsPASS **3**

3 Workflow **3**
- 3.1 Data Filtering Step . . . 4
- 3.2 Data Imputation . . . 4
- 3.3 Integration of Multiple -Omics Datasets . . . 4
- 3.4 Subnetwork Discovery Module . . . 6
  - 3.4.1 Modifications to PAM method for predictive analysis . . . 6
  - 3.4.2 Cross-validations . . . 7
- 3.5 Pathway Enrichment Module . . . 8

4 Structuring of Input Datasets **8**
- 4.1 Molecular Data . . . 9
- 4.2 Sample Group Information . . . 9
- 4.3 Network Data . . . 9
- 4.4 Pathway Information . . . 10

5 Specification for Input Parameter file **11**

6 Illustration of iOmicsPASS **13**

7 Output from iOmicsPASS **14**
- 7.1 Subnetwork Discovery results . . . 16
  - 7.1.1 EdgesSelected_minThres.txt . . . 16
  - 7.1.2 AttributesTable.txt . . . 17
  - 7.1.3 Node_Neighbors.txt . . . 17
  - 7.1.4 SampleClass_Probabilities.txt . . . 17
- 7.2 Pathway Enrichment . . . 18
- 7.3 Datasets Reported by iOmicsPASS . . . 18

8 Visualisation of Analysis Output **18**
- 8.1 CVplot_Penalty.pdf . . . 19

1 Installation

The software requires a **gcc** compiler and makes use of the **boost** library (available at `http://www.boost.org/users/download/`), which is distributed along with the software package under the Boost Software License (`https://www.boost.org/LICENSE_1_0.txt`). Additionally, a programming software **R** [1] is required for graphical visualization of the results.

1.1 Installing and setting up R

To install **R**, go to the website (`https://cran.r-project.org`) and download the right version (Linux, Mac OS X or Windows) for installation. The latest version of **R** is version 3.4.3. After that, add the directory where **R** is installed to the system **Path** environment variable by specifying the following:

For Windows users

Navigating through the following:

"My Computer > Control Panel > Systems > Advance system settings > Environment variable"

and click on **edit** to add `"C:/Program Files/R/R-3.4.3/bin"` manually. Or use command line to create/set a variable permanently (as Administrator), use:

```
> setx PATH "C:/Program Files/R/R-3.4.3/bin/"
```

For Mac OS X/Linux users

Make sure that `"usr/local/bin"` is already added in **Path**, else type the following in the "Terminal":

```
> PATH=$PATH:/usr/local/bin/
```

1.2 Installing iOmicsPASS

After downloading the zip folder for the software package from github (`https://github.com/cssblab/iOmicsPASS`), uncompress the folder on your local directory. Then, type "`make`" in the directory to install all the components of the software. Users can add the directory of `iOmicsPASS` to their PATH variable by specifying:

For example, if the program is installed in `"/Users/hiromi/Desktop/"`,

For BASH/ksh/sh shell users,

```
>PATH=$PATH:/Users/hiromi/Desktop/iOmicsPASS_v1.0/bin/
```

To make these changes permanent,

For BASH shell users,

```
>export PATH=$PATH:/Users/hiromi/Desktop/iOmicsPASS_v1.0/bin/ >> ~/.bash_profile
```

For ksh/sh shell users,

```
>export PATH=$PATH:/Users/hiromi/Desktop/iOmicsPASS_v1.0/bin/ >> ~/.profile
```

For Mac OS X/Linux users

Full installation of `Xcode` is required in Mac OS X users and all other instructions are the same as linux.

For Windows users

Executables for 64-bit Windows are included in the zip folder. Installation of `Cygwin` [2] is required (download available at `https://www.cygwin.com/`) to run `iOmicsPASS` and after installation, ensure that the directory is added to the system **Path** environment variable by navigating through the following:

"My Computer > Control Panel > Systems > Advance system settings > Environment variable"

and click on **edit** to add `"C:/cygwin64/bin"` manually. Or use command line to create/set a variable permanently (as Administrator), use:

```
> setx PATH "C:/cygwin64/bin"
```

2 Introduction to iOmicsPASS

`iOmicsPASS` is a powerful bioinformatics tool, developed in `C++` language, for network-based data integration and predictive feature selection. The tool integrates multiple -omics datasets utilizing biological network information and identifies important biological interactions in the network as signatures to distinguish between different sample groups (i.e. phenotypes). It is a supervised method where the classification of the samples is known in the training data. The tool can be applied in clinical settings, to better understand the biological mechanisms underlying the differences across groups of samples (e.g. when comparing disease versus non-disease group) in studies with multiple related -omics data.

3 Workflow

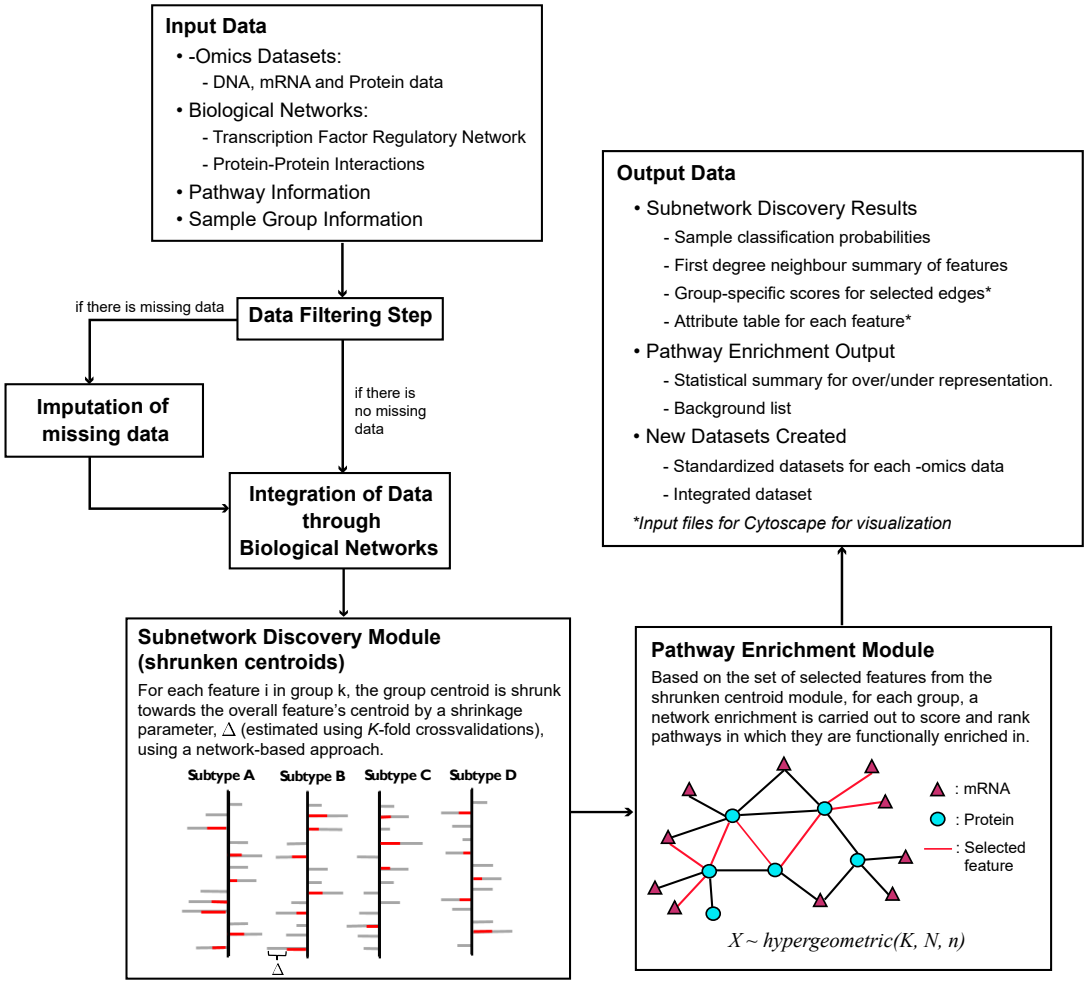

Figure 1: Workflow of iOmicsPASS

Figure 1 above shows the overview of the workflow in `iOmicsPASS`. The tool first applies data filtering with user-specified criteria on each -omics dataset to ensure that molecules with too many missing data points are removed from the data. Next, if there are still missing data, the user can choose to use the K-nearest neighbour (KNN) method to impute the data [3]. The multi-omics datasets are then integrated into a single dataset over the user-provided biological network to form an edge-level feature dataset (nodes and edges are used following the nomenclature of network biology). Next, a subnetwork discovery module, adapted from the Prediction Analysis of Microarray method by Tibshirani's et al. [4], is carried out on the transformed data, shrinking each group's centroid towards the overall average centroid by using a soft-thresholding method. K-fold cross-validation is used to optimize the shrinkage parameter, which minimizes the overall misclassification error rate. Finally, a novel method for pathway-level scoring, in a network setting, is carried out for the set of selected features from the previous steps.

Further details regarding the method can be found in Koh *et al.* [5]

3.1 Data Filtering Step

To filter out molecules with too many missing data points, each molecule should have non-missing measurements in (1) at least x number of samples or (2) at least x/n samples with non-missing data in each sample group, where x and n are integers and $x \leq n$.

3.2 Data Imputation

To impute missing data, weighted K-nearest neighbors (KNNimpute) method proposed by Schwender *et al.* [6] is used. For every molecule i with a missing value in sample j, the algorithm searches for K other molecules most similar to the given molecule but with a value present in sample j. A weighted average of the values for the K closest molecules in the same sample j is used as an estimate for the missing value, with the weights proportional to the similarity of each of the K molecules and molecule i. The similarity is determined by the Euclidean distance, or L_2-norm.

Specifically, assuming $\mathcal{L}_K$ is a set consisting of K molecules with the smallest Euclidean distances to molecule i with a value present in sample j. The missing value x_{ij} is replaced by

$$x_{ij} = \sum_{\ell \in \mathcal{L}_K} w_{i\ell} x_{\ell j} / \sum_{\ell \in \mathcal{L}_K} w_{i\ell}$$

where the weight $w_{i\ell}$ is the reciprocal of the Euclidean distance between feature i and feature ℓ.

3.3 Integration of Multiple -Omics Datasets

To integrate multiple -omics datasets, we create a measurement of "co-variation" for every pair of interacting molecules in the given biological network. When modelling the expression profiles across samples at the DNA, mRNA and protein level, two types of networks are relevant for linking the information from different molecular levels: protein-protein interaction (PPI) network and transcription factor (TF) regulatory network. The former is a network of physical binding between protein molecules, and the latter is a network between protein molecules of TF genes and mRNA molecules of target genes. In the latter network, DNA copy number can also be incorporated as a normalizing value for mRNA abundance, since

the ratio mRNA / DNA copy number can be considered as the "output" of gene transcription. Here we form two types of edges, where we consider the case with DNA copy number and the case without separately.

Each edge, $e_{i,j,t}$ for i=1,..., p, j=1,..., n and t=1 or 2, is defined as follows:

(1) when DNA copy number data is not available,

$$e_{i,j,1} = z_{prot_A,j} + z_{mrna_B,j}$$
$$e_{i,j,2} = z_{prot_A,j} + z_{prot_B,j}$$

(2) when DNA copy number data is available,

$$e_{i,j,1} = z_{prot_A,j} + (z_{mrna_B,j} - z_{dna_B,j})$$
$$e_{i,j,2} = z_{prot_A,j} + z_{prot_B,j}$$

where z represents the standardized log-scale (base 2) measurement of each molecule in the different -omics datasets, p denotes the total number of edges, n denotes the total number of samples, and t represent the type of data, where t=1 denotes the TF edge and t=2 denotes the PPI edge. For t=1, gene A is the TF and gene B is the target of A.

Here, $mrna_B$ and dna_B are the mRNA expression and DNA copy number of gene B, respectively. It is assumed to be a target of the transcription factor protein $prot_A$. Also, $prot_A$ and $prot_B$ are assumed to be any two different interacting proteins.

These new values $\{e\}$ are considered as **co-variation** between two or three related molecules because consistently high or low values of the molecules involved in a biological interaction (PPI or TF regulation) suggests increased or decreased chance of molecular interaction in a given biological sample.

Generalizing to other -omics datasets

The same data transformation approach can also be used to integrate other types of data such as mRNA expression and DNA methylation data (e.g. DNA methylation array with probe sets located in various genomic regions). Similar to protein-mRNA integration, a gene-to-gene interaction network and a network of methylation probes to their nearest gene(s) can be used for integration. It is commonly reported in literature [7, 8, 9] that methylation probes are usually negatively correlated with the gene target's expression values if the methylation site is located in the upstream of transcription start site (TSS) of a gene, and are positively correlated if the site is located within a gene body. In this case, we multiply a signed constant (-1 or 1) to the standardized measurements and allow for positive and negative interactions, as directed by the user.

The edge data are calculated as follows:

$$e_{i,j,1} = z_{gene_A,j} + sign(z_{methylprobe_B,j}) * z_{methylprobe_B,j}.$$

The same logic can be applied for the integration of negative interactions for microRNAs and protein and mRNA molecules of their target genes based on a target scan map of conserved sequences of miRNAs and target genes.

3.4 Subnetwork Discovery Module

In this module, the integrated data is put into a shrunken-centroid algorithm that we modified from the traditional PAM methodology to identify edges in a network that is predictive of the sample groups. This involves testing and training of data using cross-validations to build the best model that can be used to predict group membership of samples. And by doing so, we also obtain the set of edges or biological interactions in the overall network constructed, that best characterizes each of the sample groups that are called **signatures**.

3.4.1 Modifications to PAM method for predictive analysis

Tibshirani's *et al.* [4] developed a method for identifying important gene signatures across multiple subtypes of cancers using gene expression microarray data, called Prediction of Microarray (PAM) which is a nearest shrunken centroid method (NSC). The method computes a group centroid for each feature (gene) with a soft-thresholding method, and the shrinkage parameter is optimized using cross-validation. However, the PAM method was designed for a single -omic dataset and the method was shown to be biased towards subgroups with larger sample sizes [10, 11]. As part of `iOmicsPASS`, we made changes to the PAM method to account for the network-based data integration. The modifications are as follows:

(1) Computation of centroids for each feature (edge-level data)

The overall centroid for feature i, or interaction i, is weighted by sample group size and is calculated as:

$$\bar{x}_i = \frac{1}{K}\sum_{k=1}^{K}\bar{x}_{ik} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{n_k}\sum_{j\in C_k} x_{ij}\right)$$

where C_k represents the set of indices of the n_k samples in group k and K represents the number of sample groups.

(2) Computation of test statistics for features

In PAM, d_{ik} is defined to be the t statistics for each gene i, comparing sample group k to the overall centroid:

$$d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k.(s_i + s_0)}$$

where $m_k = \sqrt{1/n_k + 1/n}$ and s_i is the pooled within-class standard deviation of feature i and s_0 is a positive constant, the median value of s_i across all features.

Here, we add an additional term in the score to reward or penalize d_{ik} depending on the quantitative levels of the "neighbor" edges that share common molecules. We define d^*_{ik} as the new test statistics for each edge i, comparing sample group k to the overall centroid:

$$d^*_{ik} = d_{ik} + \left(\psi_k * \frac{|N_{e_{i,1}}|\sum_{s\in N_{e_{i,1}}} d_{sk} + |N_{e_{i,2}}|\sum_{r\in N_{e_{i,2}}} d_{rk}}{|N_{e_i}|}\right)$$

where N_{e_i} represents the set of edges which are connected to at least one of the nodes of edge e_i. N_{e_i} can be further partitioned into two subsets, $N_{e_{i,1}}$ and $N_{e_{i,2}}$, representing the set of TF and PPI edges, respectively.

Furthermore, we set

$$\psi_k = \frac{2e^{5(p_{ik}-0.5)}}{1+e^{5(p_{ik}-0.5)}}$$

where

$$p_{ik} = \frac{|\sum_{j \in N_{e_i}, i \neq j} \text{sign}(d_{ik}) = \text{sign}(d_{jk})|}{|N_{e_i}|}.$$

Here, ψ_k acts as a multiplicative factor based on the proportion of agreement, p_k, between the edge and its direct neighbor edges sharing at least one node of the edge i in sample group k.

Note. Since the set of neighbor edges sharing node(s) of a given edge is invariant across samples, e_{ijt} is referred to as e_i (e_{i1} for TF edges and e_{i2} for PPI edges) below.

(3) Group-specific thresholds for soft-thresholding

For each sample group k, a group-specific threshold Δ_k is derived from the shrinkage parameter, Δ.

$$\Delta_k = \frac{\Delta}{\Delta_{max}} * \max d_{ik}$$

where

$$\Delta_{max} = \frac{1}{K}\sum_{k=1}^{K} \max d_{ik}$$

Δ is defined to be over a grid of 30 equally-spaced values arranged in an increasing order from zero to some constant (i.e. $\Delta \in \{0, \Delta_1, \Delta_2, ..., \Delta_{max}\}$). Using a soft-thresholding method, each feature's distance measure d_{ik} is re-computed by reducing it incrementally by an absolute shrinkage amount and it will be set to zero if it falls below zero:

$$d'_{ik} = sign(d^*_{ik})(|d^*_{ik}| - \Delta)_+$$

The optimal value for Δ is chosen based on K-fold cross-validation, to yield a set of features, or subnetwork signatures that can best classify samples to their respective groups.

3.4.2 Cross-validations

By default, a 10-fold cross-validation is used to estimate the shrinkage parameter for soft-thresholding. The data is first split into 10 mutually exclusive parts where each part is used as test dataset and the rest of te 9 parts are used as training datasets. The training dataset is used to build a discriminant model that separates the sample groups based on a certain Δ value. Then the model is used to predict the membership of the test samples and the mis-classification error is recorded. This is repeated over a grid of increasing values for Δ starting from 0 to a value large enough that all the features in the model are reduced towards the group centroids (i.e d'_{ik} values become zero) and this is known as soft-thresholding method. The process is repeated for each of the 10 folds and the mis-classification error recorded will be averaged to give the overall mis-classifcation error.

The test samples will be classified to the nearest shrunken centroid of each group. Suppose a test sample with expression values $x^* = (x^*_1, x^*_2, ..., x^*_p)$, the discrimminant score for class k is defined to be:

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \bar{x}'_{ik})^2}{(s_i + s_0)^2} - 2log(\pi_k)$$

where π_k is the k^{th} class prior probability and estimated using an equal prior $\pi_k = 1/K$.

We then assign each of the test sample to the class with the smallest discriminant score, using the classification rule: $C(x^*) = \ell$ where $\delta_\ell(x^*) = \min_k \delta_k(x^*)$. Using the discriminant scores, we can then construct the estimates of the class probabilities as follows:

$$\hat{p}_k(x^*) = \frac{e^{-\frac{1}{2}\delta_k(x^*)}}{\sum_{\ell=1}^{K} e^{-\frac{1}{2}\delta_\ell(x^*)}}.$$

3.5 Pathway Enrichment Module

From the previous step, a set of edges with non-zero centroids are obtained. These edges can be used to perform enrichment of biological pathways in each of the subgroup. The edges are first separated into two groups with positive and negative d'_{ik} scores for each subgroup. Edges which have been shrunken towards zero (i.e. $d'_{ik} = 0$) are removed. Then, hypergeometric test will be applied to compute the probability of over-representation in a particular pathway, separately for each subgroup.

Given a graph $G = (V, E)$ where V is the set of all possible nodes (i.e. mRNA and Proteins available in the data) and E is the set of all possible edges (i.e. TF and PPI interactions from network information) in the graph. We can construct smaller subgraphs, $G_p = (V_p, E_p)$ to represent every pathway where V_p (i.e. $V_p \subset V$) is the set of nodes representing all genes in pathway p and E_p (i.e. $E_p \subset E$) is the set of edges that exist in the graph. Using the set of edges in the enrichment list for each sample group k, we create an induced subgraph of graph G_p, $G_k = (V_k, E_k)$, where V_k (i.e. $V_k \subseteq V_p$) is the set of nodes in the enrichment list that represent all genes in pathway p, and E_k (i.e. $E_k \subseteq E_p$) is the set of edges in the enrichment list that exist in the subgraph.

To find the probability of over-representation of edges in a particular pathway, we perform a hypergeometric test. Let X be a random variable representing the number of edges in a graph and define the following:

$$X \sim Hypergeometric(K, N, n)$$

$$Pr(overrepresentation) = Pr(X > x) = 1 - Pr(X \leq x)$$

where $|E_k| = K$ is the number of edges in the enrichment list, $|E| = N$ is the total number of possible TF and PPI edges, and n is the number edges that can be constructed with the vertices representing genes that belong to the pathway of interest.

4 Structuring of Input Datasets

`iOmicsPASS` takes several types of input files and the following sections give an overview of the required structure of the input datasets. All input files should be formatted in tab-delimited format. The labels in

the header do not matter but the ordering of the information in each column for the different input files should be strictly adhered to the format as described below.

4.1 Molecular Data

The molecular data should be in the format of a p by n matrix, where the rows represent the p molecules and the columns represent the samples. The first column in the data is the gene/protein identifier, followed by quantitative values in each sample. The column headers for the samples should be unique and have to correspond to the sample IDs provided by user in the sample group information.

Table 1: Example of a molecular dataset structure

Gene_identifier	**Sample_1**	**Sample_2**	**...**	**Sample_n**
$gene_1$	1.032	1.423	$\cdots$	0.894
$gene_2$	2.355	0.983	$\cdots$	1.188
$gene_3$	0.783	1.216	$\cdots$	0.730
$gene_4$	2.892	1.822	$\cdots$	0.741
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$gene_p$	0.283	1.024	$\cdots$	0.762

Note. The ordering of the samples does not matter and data will only be extracted from samples that have a sample ID provided in the sample group information. The type of gene identifiers should be uniform across the different -omics datasets. For example, if gene symbols are used in RNA-level data, it should also be used in Protein-level data.

4.2 Sample Group Information

The sample group information should contain two columns. The first column is the sample identifiers/IDs and the second column is the sample group information (class).

Table 2: Example of a Sample Information structure

Sample_ID	**Classification**
Sample_1	Class_1
Sample_2	Class_3
Sample_3	Class_2
Sample_4	Class_1
$\vdots$	$\vdots$
Sample_n	Class_k

4.3 Network Data

Up to two types of network files can be supplied by the user in the current version. One of the network files will be used to link features from the Protein-level input data to the RNA-level input data. The other

network file will be used to link features within the Protein-level input data.

In the TF network file, there should be two columns containing headers where each row shows the interaction between two molecules: the molecule in the first column has a directed edge pointing or targeting the corresponding molecule in the second column (see Table 3A).

In the PPI network file, there should be two columns containing headers where each row shows the interaction between two features: the node in the first column has an undirected edge with the corresponding node in the second column (see Table 3B).

For implementing both positive and negative interactions in a single network, a third column should be included in the network file with values 1 or -1, to denote positive and negative interactions, respectively (see Table 3C).

Table 3A: Example of a TF network file structure

TF	**TF_target**
$Protein_a$	$mRNA_1$
$Protein_a$	$mRNA_2$
$Protein_b$	$mRNA_3$
$Protein_b$	$mRNA_4$
$Protein_c$	$mRNA_5$
$\vdots$	$\vdots$

Table 3B: Example of a PPI network file structure

Protein_A	**Protein_B**
$Protein_a$	$Protein_b$
$Protein_a$	$Protein_c$
$Protein_b$	$Protein_d$
$Protein_c$	$Protein_d$
$Protein_c$	$Protein_e$
$\vdots$	$\vdots$

Table 3C: Example of a network file with signed interactions

Gene	**MethylationProbe**	**InteractionSign**
$gene_a$	$methyprobe_1$	1
$gene_a$	$methyprobe_2$	-1
$gene_b$	$methyprobe_3$	-1
$gene_b$	$methyprobe_4$	1
$gene_c$	$methyprobe_5$	1
$\vdots$	$\vdots$	$\vdots$

4.4 Pathway Information

The pathway information file should have three columns containing headers. The first is the gene identifier, the second column is the pathway identifier that the gene belongs to and the third column denotes the pathway information (e.g. name of pathway).The type of gene identifier used should be consistent with that used in the input data. For example, if gene symbols are used in the input for molecular datasets, then it should also be used here in the pathway information.

Table 4: Example of a Pathway module file structure

Gene_Symbol	**Pathway_ID**	**Pathway_Function**
SOX9	GO:0042981	regulation of apoptotic process
MAP3K8	GO:0042981	regulation of apoptotic process
ALX4	GO:0042981	regulation of apoptotic process
UBC	GO:0042981	regulation of apoptotic process
BCL6	GO:0000902	cell morphogenesis
FRYL	GO:0000902	cell morphogenesis
SOX6	GO:0000902	cell morphogenesis
⋮	⋮	⋮

5 Specification for Input Parameter file

The table below (Table 5) gives a short description of each input parameter offered in `iOmicsPASS`.

Table 5: Description of input parameters for iOmicsPASS

Parameter Label	**Variable Type**	**Description**
	Input data files	
`DNA_FILE`	*string*	Filename of DNA copy number dataset (optional).
`RNA_FILE`	*string*	Filename of RNA-level expression dataset.
`PROT_FILE`	*string*	Filename of Protein-level expression dataset.
`PPI_NETWORK`	*string*	Filename of Protein-Protein interaction network file.
`TF_NETWORK`	*string*	Filename of Transcription-factor regulatory network file.
`MODULE_FILE`	*string*	Filename of pathway information for enrichment.
`SUBTYPE_FILE`	*string*	Filename of group information.
	Parameters for data manipulation and filtering step	
`LOG_TRANSFORM_DNA`	*boolean* <true/false>	If true, the data will be treated as intensities (before log) and log (base 2) transformation will be carried out. If false, no log-transformation will be carried out.
`LOG_TRANSFORM_RNA`	*boolean* <true/false>	If true, the data will be treated as intensities (before log) and log (base 2) transformation will be carried out. If false, data will be assumed to be log-transformed (default is true).
`LOG_TRANSFORM_PROT`	*boolean* <true/false>	If true, the data will be treated as intensities (before log) and log (base 2) transformation will be carried out. If false, data will be assumed to be log-transformed (default is true).
`ZTRANS_DNA`	*boolean* <true/false>	Whether or not to perform standardization to the DNA-level data before integration (default is true).

Continued on next page

Table 5 – continued from previous page

Parameter Label	Variable Type	Description
ZTRANS_RNA	*boolean* <true/false>	Whether or not to log-transform the RNA-level data before integration (default is true).
ZTRANS_PROT	*boolean* <true/false>	Whether or not to log-transform the Protein-level data before integration (default is true).
MIN_OBS	*integer*	Minimum number of samples in each sample group with complete data that is used in the analysis (default is 1).
MIN_PROP	*double* [0-1]	Minimum proportion of samples in each sample group with complete data that is used in the analysis (default is 0.5).
KNN_IMPUTE	*boolean* <true/false>	Whether or not to perform K-nearest neighbor imputation. Should be set to true if dataset has missing cells (default is true).
MAX_BLOCKSIZE	*integer*	Maximum number of samples used to infer the imputation in KNN (default is 1500).
Parameters for iOmicsPASS		
ANALYSE_DNA	*boolean* <true/false>	Whether or not to include DNA in the integration analysis. If true, a filename has to be provided in DNA_FILE.
INTERACTION_SIGN	*boolean* <true/false>	Whether or not to account for different types of interactions in network. If true, a third column with values 1 or -1 should be included in TF_NETWORK (default is false).
CROSS_VALIDATION	*boolean* <true/false>	Whether or not to carry out cross-validation to select optimal threshold cut-off. If true, the minimum threshold will be automatically selected and used as a cut-off. If false, a threshold value should be provided in MIN_THRES.
CV_FOLD	*integer*	Number for K in K-fold cross-validation (default is 10).
MIN_THRES	*double*	Threshold to be used as the final cut-off in scoring algorithm. If not specified, this value will be automatically selected to be the threshold at which the lowest misclassification error occurs from the cross-validations.
Specifications for subnetwork enrichment module		
BACKGROUND_PROP	*double*	Minimum proportion of genes in the pathway that are also in the background list (default = 0.5).
MINBG_SIZE	*integer*	Minimum number of edges constructed from the set of genes in a given pathway (default is 1).
MINSIG_SIZE	*integer*	Minimum number of selected edges from the network constructed in a given pathway to be reported (default is 3).

Choice of threshold through cross-validation:

The tool, by default, will use the threshold that yields the smallest misclassification error to select features if no threshold is specified. At times, this may not be the most desirable method. For instance, there are many thresholds with similar misclassification error rates, yet the numerically optimal threshold leads to selection of too many features (i.e. there is an alternative threshold with far sparser and more interpretable size of networks).

A separate R-code (`PerformancePlot.R`) is provided in the tool to assist visualization of the misclassification error plot for the entire range of possible thresholds. In the plot, a line indicating one standard deviation above the minimum misclassification error is drawn. Users can select a threshold that produces a more

sparse network which maintains the core constituents of predictive network.

More generally, the user can then make an informed decision as to where to choose the optimal threshold where the trade-off between misclassification error rate and the number of selected features is balanced. Then, users can rerun `iOmicsPASS` by turning the option `CROSS_VALIDATION` off and specifying the preferred threshold in `MIN_THRES`.

6 Illustration of iOmicsPASS

To illustrate the use of `iOmicsPASS`, we utilize the breast cancer (BRCA) dataset from the Cancer Genome Atlas (TCGA) [12]. The gene expression data was quantified using RNA sequencing in 1,098 tumor samples collected from patients and quantification of proteins using iTRAQ was carried out on a subset of 105 breast cancer patients by the Clinical Proteomic Tumor Analysis Consortium (CPTAC) [13]. Of those, 103 subjects had subtype classification using mRNA-based PAM50 signature [14]. into 4 breast cancer subtypes: Basal-like, Her2-enriched, Luminal A and Luminal B subtype. By performing this analysis, we aim to identify subnetwork signatures of transcriptional regulation network and protein-protein interaction network which are predictive of each breast cancer subtype.

To reduce computational time, we randomly selected 1000 transcription factor proteins and 3000 mRNAs from the gene targets of the selected transcription factors. Then, a smaller subset of the TF and PPI network was derived to make up the datasets in the example folders. Out of the 103 individuals with group information, 24 were classified as Basal-like, 18 were Her2-enriched, 24 were Luminal A and 32 were Luminal B subtype.

Then, `iOmicsPASS` was executed to integrate the proteomics and mRNA datasets using the input parameters shown in Figure 2. Here, we performed log-transformation on the mRNA data as the values are fragments per kilobase of transcript per million mapped reads (FPKM) and *KNN* imputation has to be set to `true` as there are missing values in both datasets. We also require that for each gene/protein in the input data should have at least 10 or 80% (which ever is larger) non-missing observations across the samples in each sample group. For parameters in the enrichment module, we will require pathways with at least 50% of the genes in the background list and report pathways with a size of 3 edges and have at least 1 enriched edge.

In the directory of the example folder, type the following command to run the tool:

```
> ../bin/iOmicsPASS <input_example>
```

If running on Windows, users can use **Command Prompt** and go to the directory `"Windows Binary"` to use the executable (`iOmicsPASS.exe`). Note that another input parameter file —`input_windows`, slightly modified from `input_example`, was created as the directories of the input files changed since `iOmicsPASS.exe` has to be executed within `"Windows Binary"` folder.

Then, type the following command to run the tool:

```
> iOmicsPASS.exe <input_windows>
```

Note: The lines in the input parameter file commented out by hash # symbol will not be read in. The input parameter and the data files should be placed are in the same working directory. Otherwise, include the full path to the files in the input parameter.

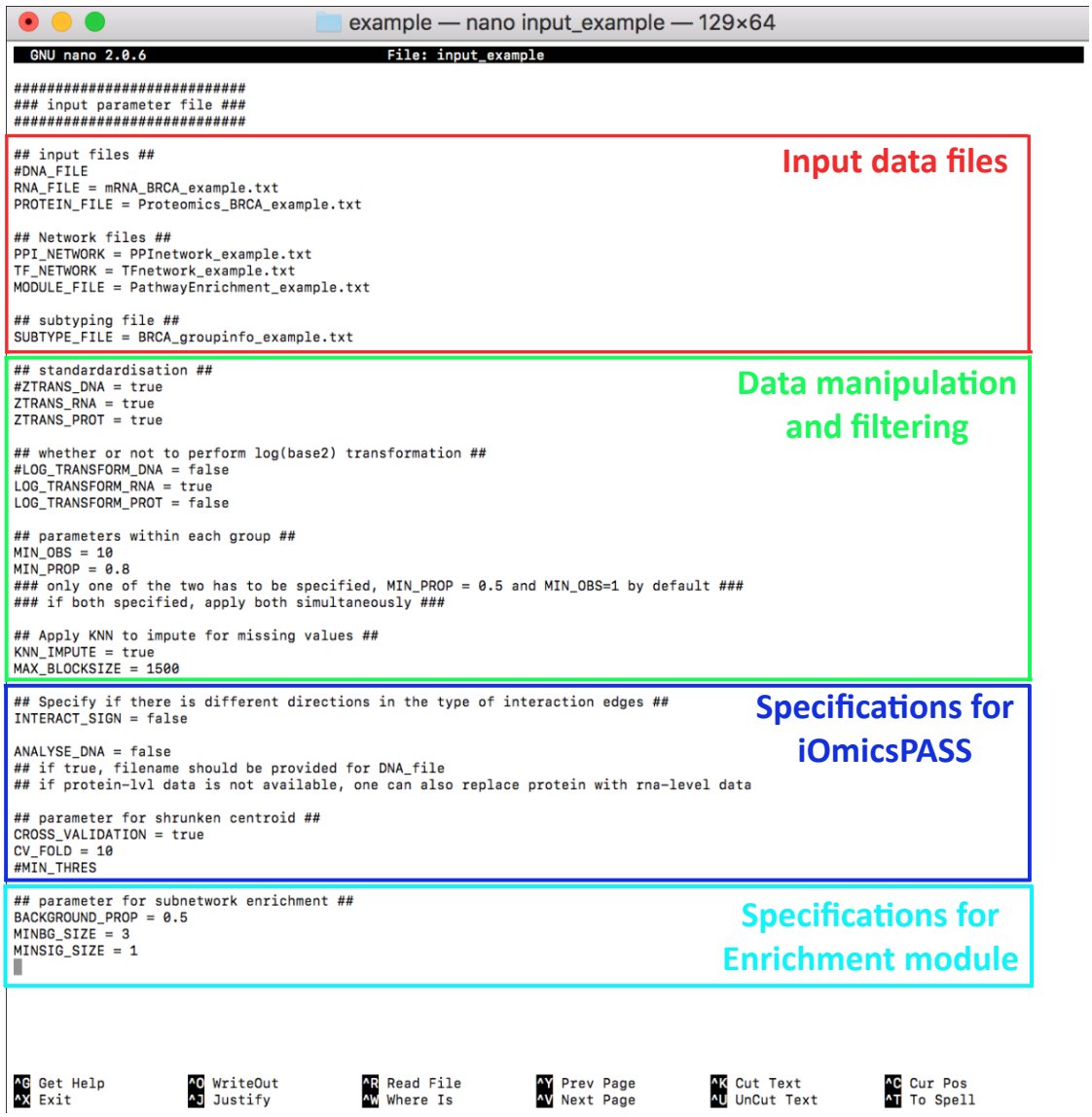

Figure 2: Screen shot of input parameter

7 Output from iOmicsPASS

Executing `iOmicsPASS` may take some time depending on the size of the input data (number of molecules) and the density of the network data. The runtime is the longest when the network density is very high, i.e. many interactions share common nodes (molecules). During the runtime, the progress of the software will be printed on the screen. Figure 3 shows an example of a log file when executing the tool on an example dataset.

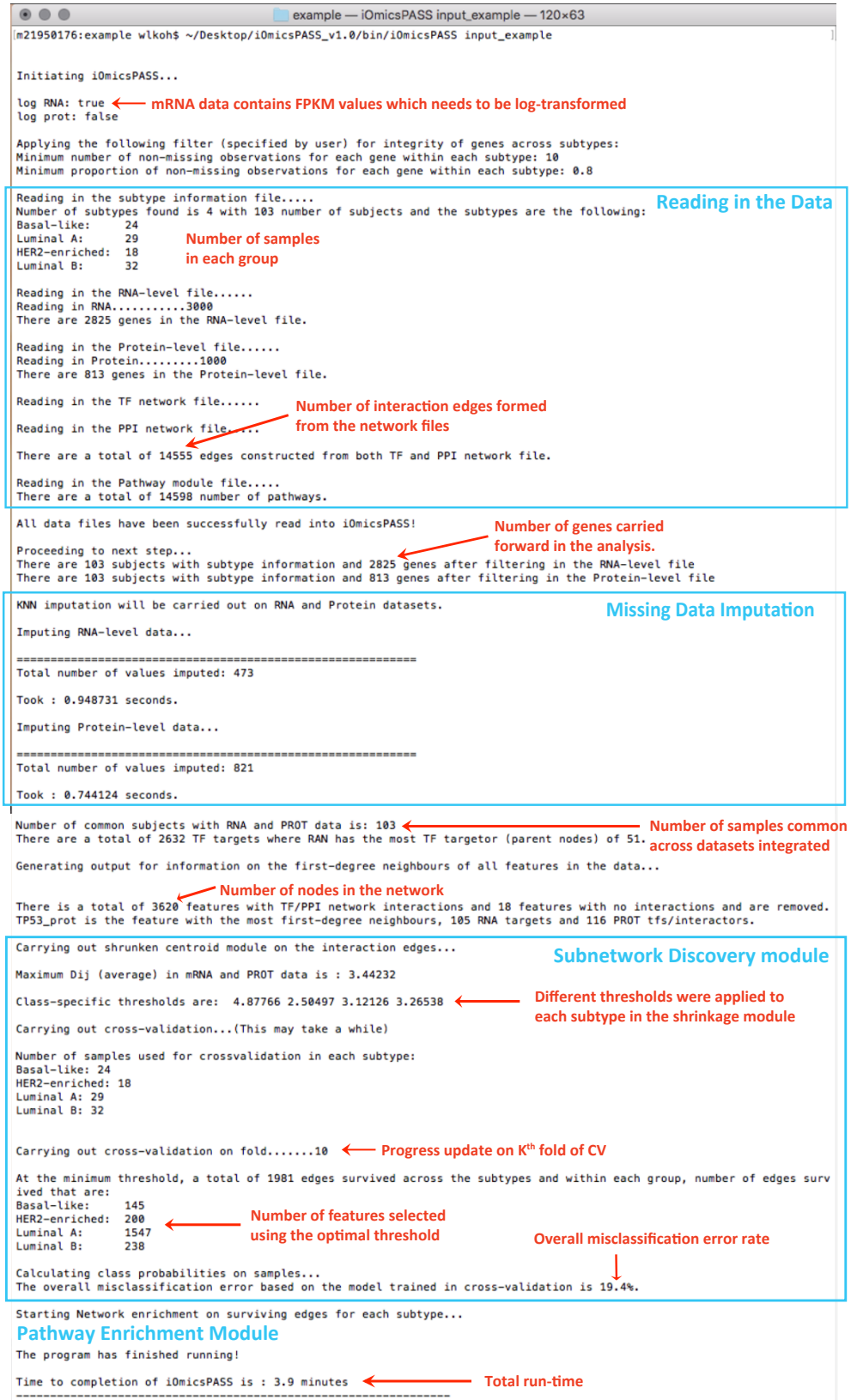

Figure 3: Screen shot of executing iOmicsPASS on the example datasets, with annotations.

The total runtime of `iOmicsPASS` on the example dataset was about 4 minutes. A total of 14,555 edges with quantitative data were constructed and 1,981 features were selected across the four subtypes by the cross-validation.

Notes: The number of features (edges) selected for each run may vary as the software uses a random number generator to perform the K-fold cross-validation. However, the key constituents of the predictive subnetworks should remain stable.

Running `iOmicsPASS` will yield three types of output files:

1) **SubnetworkDiscovery results** - scores and attributes of the selected features, classification probabilities and network summary of nodes on the network.
2) **Pathway enrichment** - statistical summary of the enrichment in pathways of the selected features and the list of all genes/proteins used in the enrichment.
3) **Datasets created** - standardized datasets and the network-level data.

7.1 Subnetwork Discovery results

Four files are generated to summarize the information in the overall network formed by the selected edges predictive of each sample group and report the classification probabilities on the samples based on the set of selected edges:

7.1.1 EdgesSelected_minThres.txt

Edges included in this output are the predictive subnetwork signatures in at least one of the groups. This file can be read into Cytoscape directly as a table of interactions for network visualization.

- **Edge** : Concatenated string of the two molecules involved in the interaction.
- **NodeA** : Molecule represented on the Node A, interacting with Node B.
- **NodeB** : Molecule represented on the Node B, interacting with Node A.
- **InteractionType** : Type of interaction: <tf> for the co-expression of the TF protein molecule in Node A and the target mRNA molecule in Node B in the TF network; or <ppi> for the co-expression of the protein molecule in Node A and the protein molecule in Node B in the PPI network.
- **XXX_dikNew** : The test statistics of the edge for the sample group as a measure of the predictiveness. A large absolute value of this score indicates that the edge is highly predictive of the sample group.
- **XXX_sigEdge** : Type of significance of the selected edge: <died> if the edge has a test statistics of 0; <ppi> if the test statistics is not zero and the type of interaction is PPI; <tf> if the test statistics is not zero and the type of interaction is TF.
- **XXX_dir** : The direction of the selected edge: <died> if the edge has a test statistics of 0; <up> if the test statistics is positive; <down> if the test statistics is negative.

- **XXX_absdiknew** : The absolute value of the test statistics of the edge for the sample group as a measure of the predictiveness.

7.1.2 AttributesTable.txt

This is the attribute table for the nodes in the set of selected edges. This file can be used as an attribute table in Cytoscape.

- **Node** : Identifiers used for nodes.
- **Gene** : Gene identifier provided by user.
- **Type** : Molecule type: <mrna> if the molecule is a mRNA; <prot> if the molecule is a protein.
- **XXX_surv** : "Survival" or selection status of the edge: <died> if the edge has a test statistics of 0; <mrna> if the test statistics is not zero and the molecule is a mRNA; <prot> if the test statistics is not zero and the molecule is a protein.

7.1.3 Node_Neighbors.txt

This file gives a summary of number of "direct neighbors" of each node in the integrated network constructed.

- **Node** : Identifiers used for nodes.
- **Gene** : The gene identifier provided by user.
- **Type** : Type of molecule: <mrna> if the molecule is a mRNA; <prot> if the molecule is a protein.
- **Neighbor_inData** : A concatenated string of molecules that have an edge with the current molecule (direct neighbors).
- **NumNeigh_inData** : Number of nodes connected with the current molecule (by an edge).

7.1.4 SampleClass_Probabilities.txt

This file gives the classification probabilities (calculated based on discriminant scores) of each sample to the different sample groups based on the set of selected features obtained from cross-validation.

- **Subject** : Sample ID
- **Prob_XXX** : Classification probability: Probability of the sample belonging to each of the sample group.
- **TrueClass** : The sample group that the sample belongs to.
- **PredictedClass** : The sample group that the sample would be assigned to, given the highest classification probability. <tied> is reported if there are more than one highest classification probability score.

7.2 Pathway Enrichment

For each sample group, pathway enrichment is tested on the set of up-regulated and down-regulated subnetworks, separately, and a statistical summary of the enrichment is reported. Here, the test of enrichment is performed accounting for the fact that the data are network edges. For each pathway, a smaller network is constructed based on the set of gene identifiers in the pathway and hypergeometric test is used to compute the statistical significance of enrichment for the selected edges (up and down separately).

For example, in the TCGA BRCA dataset, there were 4 cancer subtypes (Basal-like, Her2-enriched, Luminal A and Luminal B) specified, hence there is a total of 8 enrichment files generated based on the set of features selected within each of the subtype.

e.g. **Basal-like_Enrichment_up.txt**

- **Pathway** : The pathway identifier provided by user in pathway module file.
- **PathAnnotation** : The annotation of the pathway provided by user in pathway module file.
- **HypergeoPval** : Hypergeometric P-value for functions in the up- and down-regulated edges.
- **NumGene_inPathway** : The number of genes in the pathway.
- **NumGenes_inbg** : The number of genes that are in the pathway and present in the input data, used in the analyses (background list).
- **PropPathway** : The proportion of genes in the pathway that is represented in the background list.
- **NumEdgesformed_inbg** : The number of edges formed from the set of genes/proteins in the pathway.
- **Enriched_Edgesize** : The number of edges that are in the enrichment list that belong to the set of edges constructed in the pathway

7.3 Datasets Reported by iOmicsPASS

For each of the molecular-level, a dataset will be generated after data-filtering, log-transformation (if required) and standardization (i.e. **Ztransform_XXX.txt**). Also, if imputation was required, a separate dataset will also be generated (i.e. **imputedData_XXX.txt**).

8 Visualisation of Analysis Output

Along with the package, we provide a supplementary R-script in the folder `Rcodes`, to aid visualization for evaluating the performance of the cross-validation. Users can use this plot to pick a threshold with good trade-off between the mean misclassification error and number of features selected. Then, re-specify the threshold (i.e. by entering the value in `MIN_THRES`) in the input parameter and run `iOmicsPASS` again, without carrying out cross-validation (i.e. set `CROSS_VALIDATION` = false). This is useful especially when the optimal threshold picked out by the tool by default, results in too many features being selected.

After running `iOmicsPASS`, enter the following command in the same directory where the results are generated (i.e. example/Windows Binary folder):

```
> R CMD BATCH ../Rcodes/PerformancePlot.R
```

8.1 CVplot_Penalty.pdf

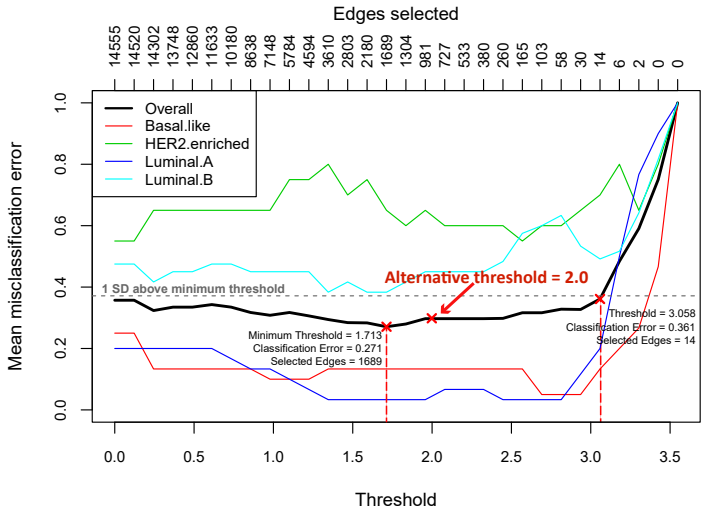

Figure 4: Performance Plot of K-fold cross-validations

A plot similar to the figure above will be produced. The black colored line shows the overall misclassification error rates and other colored lines show the group-specific misclassification error rates. We observed that, as the threshold increases, the overall misclassification error rate decreases slightly before plateauing and eventually rising to 1 as the number of features selected approaches 0. In this example, the threshold 1.713 is selected as the optimal threshold with the lowest error rate of 27.1%, which results in about 1,689 features selected.

Note: The number of features that are selected in the actual data (i.e. 1,981 edges) is slightly different from the number reported in the performance plot (i.e. 1,689 edges) as the latter is calculated using the mean of the features selected from each of the training datasets in K-fold cross-validations.

A dotted-line is also shown on the plot to illustrate the range of thresholds that keep the mis-classification error rates below one standard deviation (SD) from the minimum error. This enables the user to pick another threshold that provides good trade-off between number of edges selected and the error rate. For example in figure 4, thresholds ranging between 1.7 to 3.0 results in errors within one SD away from the minimum error of 27.1%. At the current threshold that minimizes the error, the number of edges selected may be still too large and to reduce this number, hence making the edges selected more specific, an alternatve threshold should be considered. One could pick a threshold of 2.0 instead and the number of edges slected would be reduced to about 900 and at the same time, not increasing the error by too much and keeping it controlled at 30%. The user can then re-run `iOmicsPASS` without carrying out cross-validation (saving computation time) and specifying a threshold in the input parameter file (Figure 5).

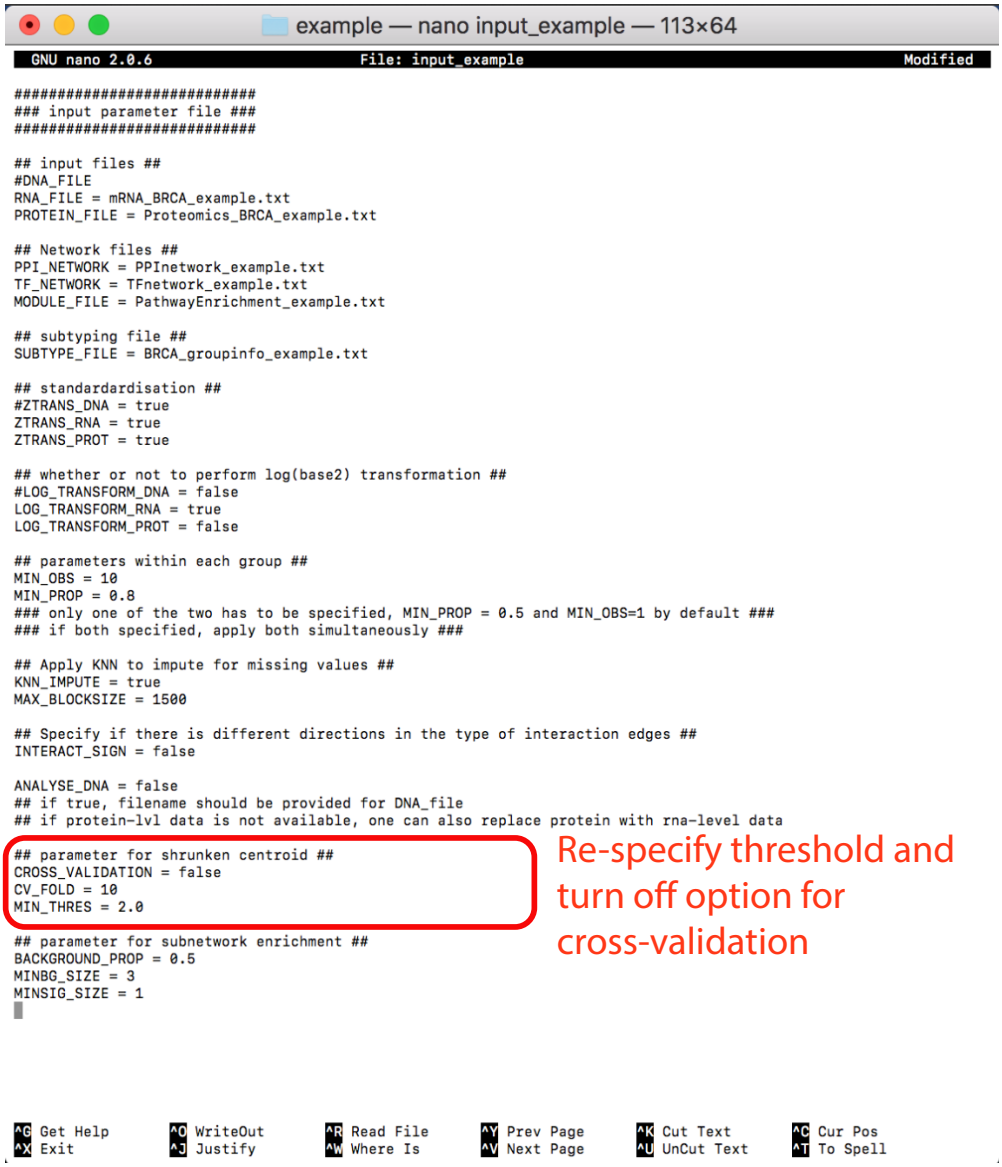

Figure 5: Screen shot of input parameter file after respecifying threshold of 2.0.

References

[1] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[2] J. Racine. The cygwin tools: A gnu toolkit for windows. *Journal of Applied Econometrics*, 15(3):331–341, 2000.

[3] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[4] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS; Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[5] Hiromi W L Koh, Damian Fermin, Kwok Pui Choi, Rob Ewing, and Hyungwon Choi. iOmicsPASS: Integration of multi-omics data over biological networks and discovery of predictive subnetworks. (in preparations).

[6] Holger Schwender. Imputing missing genotypes with weighted k nearest neighbors. *Journal of Toxicology and Environmental Health, Part A*, 75(8-10):438–446, 2012. PMID: 22686303.

[7] U Kass Stefan, Nicoletta Landsberger, and Alan P. Wolffe. Dna methylation directs a time-dependent repression of transcription initiation. *Current Biology*, 7(3):157–165, 1997.

[8] Peter A. Jones. The dna methylation paradox. *Trends in Genetics*, 15(1):34–37, 1999.

[9] Peter A. Jones. Functions of dna methylation: islands, start sites, gene bodies and beyond. *Nature Reviews Genetics*, 13(7):484–92, 05 2012.

[10] Haibo He He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[11] Rok Blagus and Lara Lusa. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 11(1):523, 2010.

[12] The Cancer Genome Atlas Research Network, John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–20, 2013.

[13] Nathan J. Edwards, Mauricio Oberti, Ratna R. Thangudu, Shuang Cai, Peter B. McGarvey, Shine Jacob, Subha Madhavan, and Karen A. Ketchum. The cptac data portal: A resource for cancer proteomics research. *Journal of Proteome Research*, 14(6):2707–2713, 2015.

[14] The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490:61–70, 2012.