Supporting Information for:

pseudoQC: A Regression-Based Simulation Software for Correction and Normalization of Complex Metabolomics and Proteomics Datasets

Shisheng Wang, and Hao Yang*

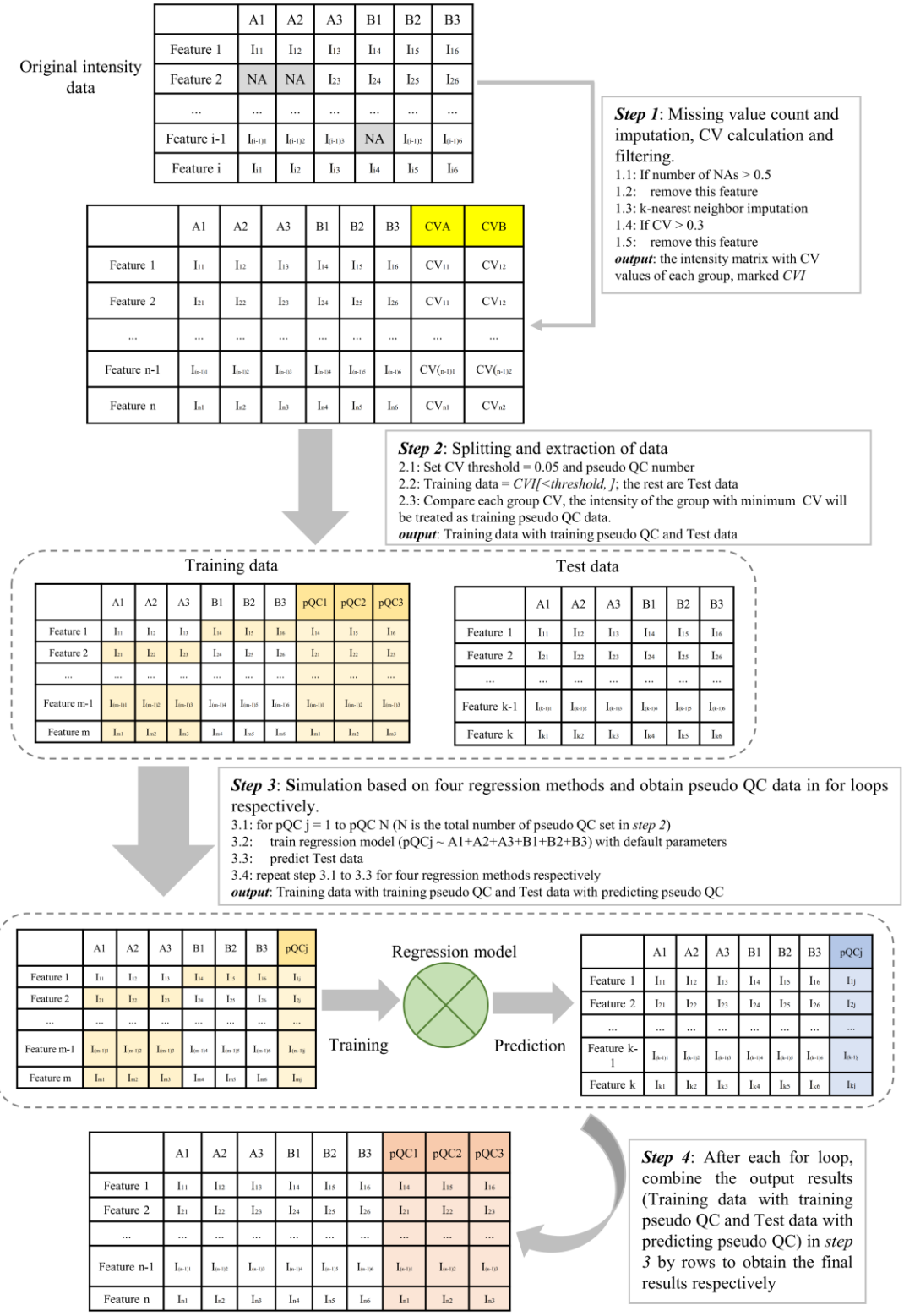
West China-Washington Mitochondria and Metabolism Research Center; Key Lab of Transplant Engineering and Immunology, MOH, West China Hospital, SCU. No. 88, Keyuan South Road, Hi-Tech Zone, Chengdu 610041, China

*Corresponding author.
E-mail: yanghao@scu.edu.cn

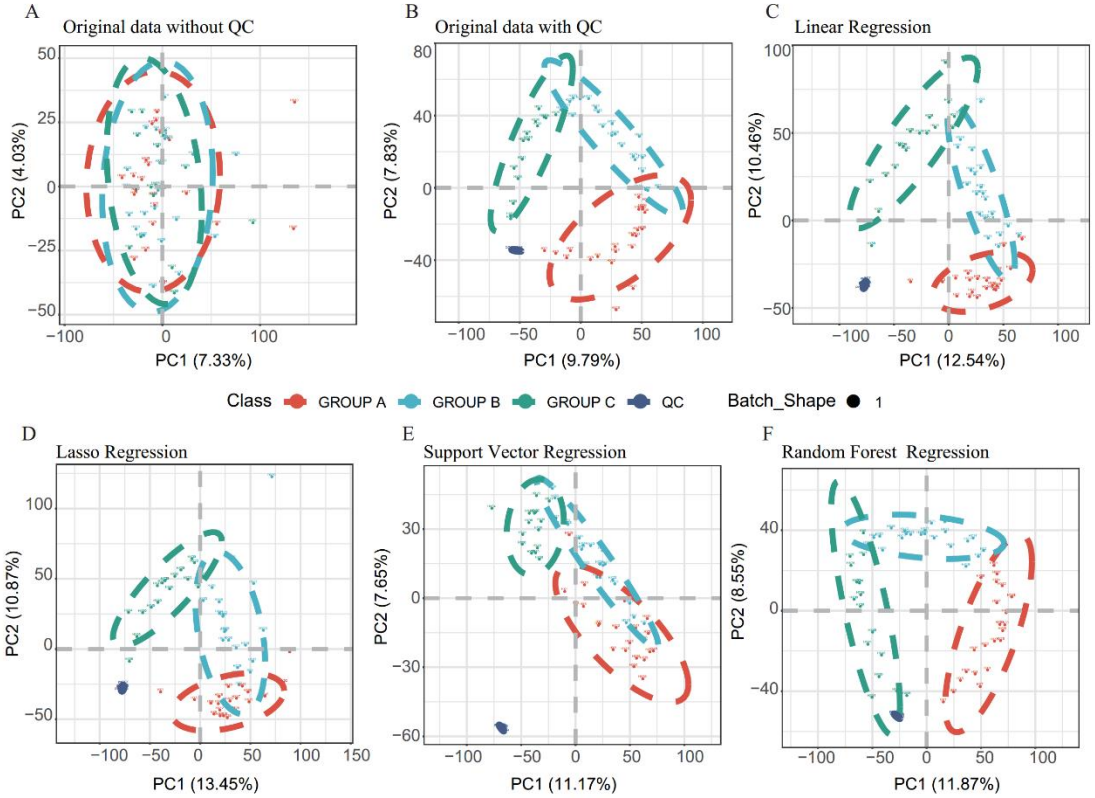
Table of Contents

I. Supplementary figure legends and Tables

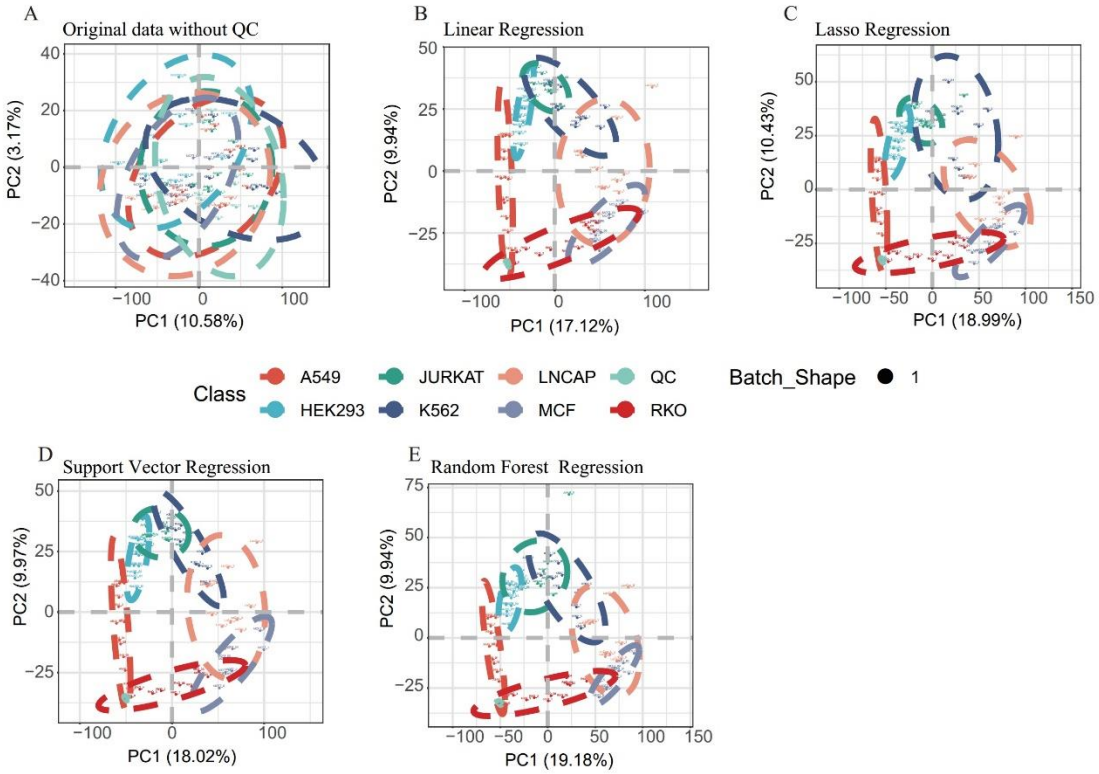
Supplementary Figure 1. Major steps of the pseudoQC simulation process. We take two groups of samples (three biological replicates in each group, labeled A1, A2, A3, B1, B2, B3 in the original intensity data) for example. Feature means the identified protein/metabolite.



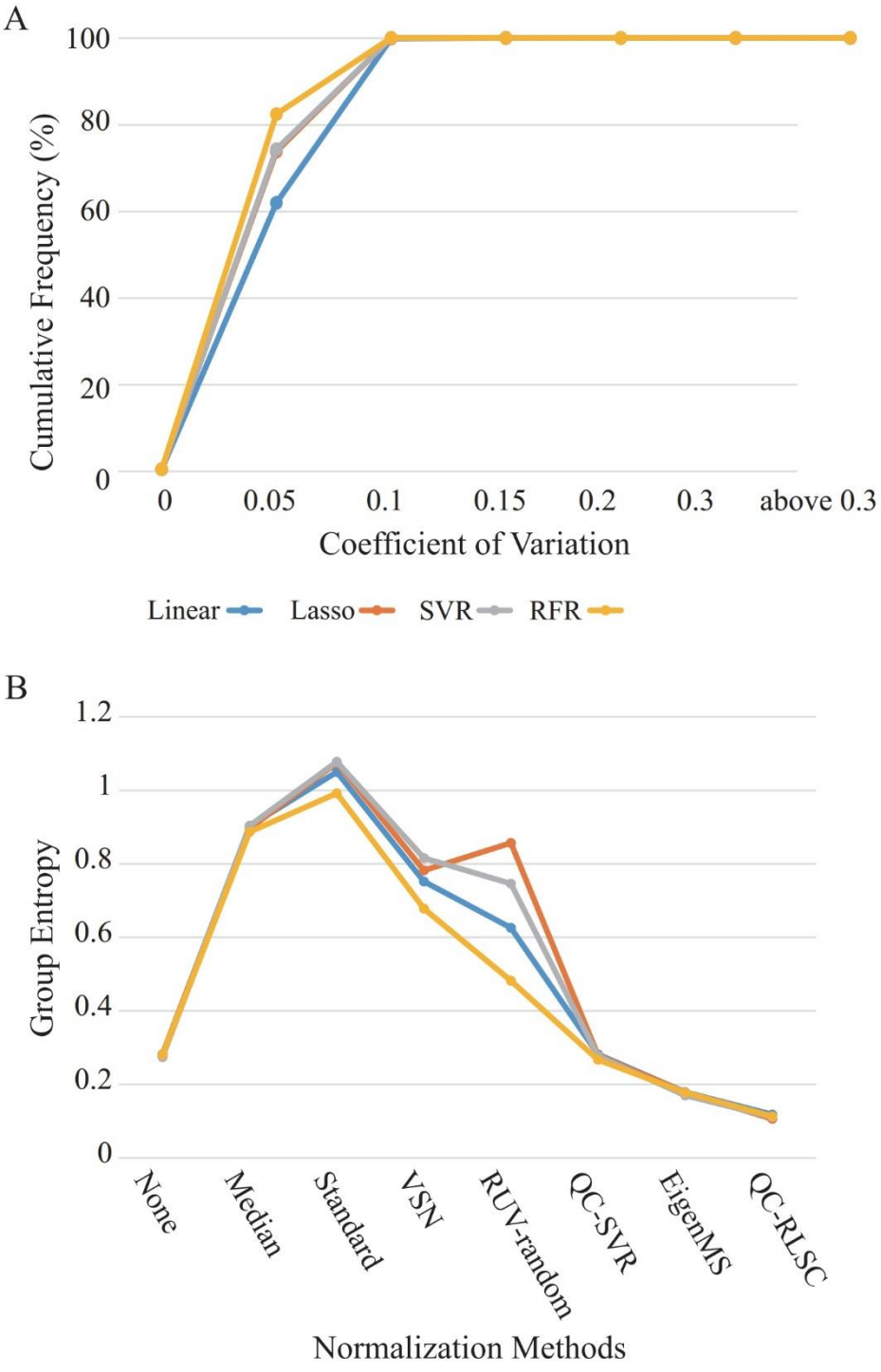
Supplementary Figure 2. Spatial distribution of the PCA scores of metabolomics data simulated by various regression methods after QC-RLSC normalization. (A) Original data without QC sample data. (B) Original data with QC sample data. QC sample data simulated by (C) linear regression, (D) lasso regression, (E) SVR, and (F) RFR.



Supplementary Figure 3. Spatial distribution of PCA score of proteomics data simulated by various regression methods after QC-RLSC normalization method. (A) Original data without QC samples data. Simulated QC sample data from (B) Linear regression. (C) Lasso regression. (D) Support vector regression. (E) Random forest regression.



Supplementary Figure 4. Performances of four different regression models for proteomics data. Cumulative frequency of CV% of all features for original QC data and those (pseudoQC data) obtained by simulation using four regression methods (A). Group entropy distribution of the original and simulated QC data across all seven normalization methods (B).



II. Supplementary Method

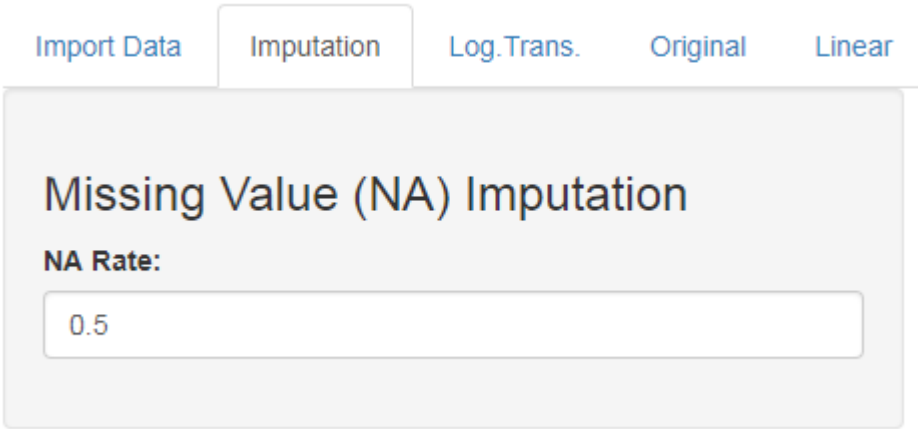
1. Software features description.

This is mainly to explain the contents in Table 1. There are total three aspects (applicability, compatibility and functions) to process the quantitative assessments.

Aspects	Features	Description
Applicability	Suit for metabolomics data	To evaluate whether the tool can analyze metabolomics data.
	Suit for proteomics data	To evaluate whether the tool can analyze proteomics data.
Compatibility	Platform independent	To evaluate whether the tool can be used in different platforms, such as Windows, Linux, and Max OS.
Functions	GUI	To evaluate whether the tool has a graphical user interface.
	Missing value processing	To evaluate whether the tool can process missing value imputation methods.
	Simulation QC data	To evaluate whether the tool can simulate QC sample data.
	Results visualization	To evaluate whether the tool supports to plot results.

2. Missing value imputation.

We set missing values (whose peak intensities are 0s) to not available values (NAs) and removed those features in which the NAs ratio was above 0.5 (50%, default), this parameter can be adjusted by users in the GUI (shown as below). After that, imputation is implemented with the k-Nearest Neighbor (KNN) algorithm, which can be implemented with MissingValues function in NormalizeMets package[1].



3. Coefficient of Variation.

The coefficient of variation measures the variability of a series of samples in each group and is defined as the ratio of the standard deviation (δ) to the mean (μ)[2]:

$$CV = \frac{\delta}{\mu}$$

4. Function Implementation including four regression methods.

There are four frequently used regression methods, namely linear regression[3], lasso regression[4], support vector regression (SVR)[5], and random forest regression (RFR)[6]. All functions were compiled with R (https://www.r-project.org/). The detailed implementation methods are summarized as below:

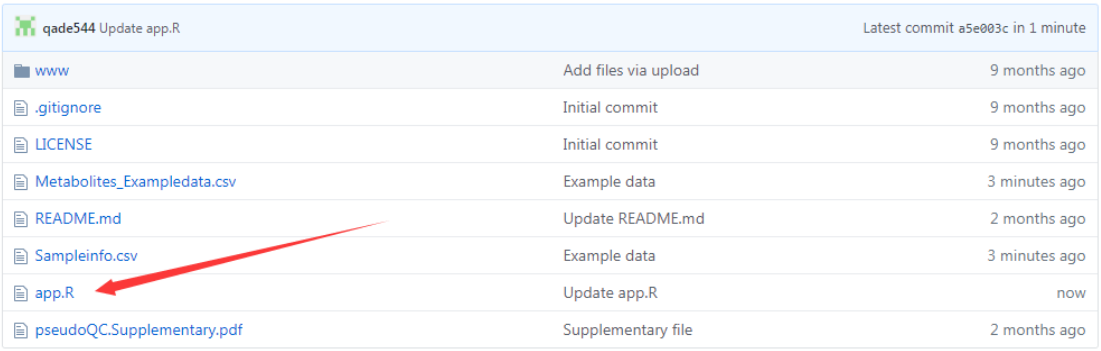
Regression Method	Function	Package
linear regression	lm	stats[7]
lasso regression	lars	lars[4]
support vector regression	svm	e1071[5]
random forest regression	randomForest	randomForest[6]

As described in Supplementary Figure 1, we analyze the training data and test data with a similar process for the four regression models, the main implementation R codes are shown as below:

```
train_pQC_data = CVI[CV<threshold, ]      # Training data with training pseudo QC
test_data = CVI[CV>=threshold, ]              # Test data without training pseudo QC
N=3       # N is the total number of pseudo QC, which can be set by users, but this value should be less
          #than or equal to the minimum replicate number. For instance, if one user has data (3 groups of
          #samples (A, B, C), A has 10 biological replicates, B has 6 biological replicates, C has 9
          #biological replicates), N should be less than or equal to 6.

#4.1. linear regression model:
pQC_predict = NULL
for(j in 1:N){
   yy = train_pQC_data [ ,ncol(test_data)+j]
   datai = cbind(yy, train_pQC_data[, 1: ncol(test_data)])
   lmi = lm(yy~., data = datai)          #training model
   lmipre = predict(lmi, test_data)       #prediction
   lmipredf = as.matrix(data.frame(xx=lmipre))
   pQC_predict = cbind(pQC_predict, lmipredf)
}
test_pQC_data = cbind(pQC_predict, test_data)      # Test data with training pseudo QC
lm_results = rbind(train_pQC_data, test_pQC_data)

#4.2. lasso regression model:
pQC_predict = NULL
for(j in 1:N){      # N is the total number of pseudo QC
   yy = train_pQC_data [ ,ncol(test_data)+j]
```

```
   datai = cbind(yy, train_pQC_data[, 1: ncol(test_data)])
   larsi = lars(x = datai [,-1],y= datai [,1],type = "lasso")          #training model
   larsipre = predict(larsi, test_data)        #prediction
   larsipredf = as.matrix(data.frame(xx=larsipre))
   pQC_predict = cbind(pQC_predict, larsipredf)
}
test_pQC_data = cbind(pQC_predict, test_data)      # Test data with training pseudo QC
lars_results = rbind(train_pQC_data, test_pQC_data)

#4.3 support vector regression model:
pQC_predict = NULL
for(j in 1:N){      # N is the total number of pseudo QC
   yy = train_pQC_data [ ,ncol(test_data)+j]
   datai = cbind(yy, train_pQC_data[, 1: ncol(test_data)])
   svmi = svm(x = datai [,-1],y= datai [,1])          #training model
   svmipre = predict(svmi, test_data)        #prediction
   svmipredf = as.matrix(data.frame(xx=svmipre))
   pQC_predict = cbind(pQC_predict, svmipredf)
}
test_pQC_data = cbind(pQC_predict, test_data)      # Test data with training pseudo QC
svm_results = rbind(train_pQC_data, test_pQC_data)

#4.4 random forest regression model:
pQC_predict = NULL
for(j in 1:N){      # N is the total number of pseudo QC
   yy = train_pQC_data [ ,ncol(test_data)+j]
   datai = cbind(yy, train_pQC_data[, 1: ncol(test_data)])
   rri = randomForest(x = datai [,-1],y= datai [,1])          #training model
   rripre = predict(rri, test_data)        #prediction
   rripredf = as.matrix(data.frame(xx=rripre))
   pQC_predict = cbind(pQC_predict, rripredf)
}
test_pQC_data = cbind(pQC_predict, test_data)      # Test data with training pseudo QC
randomForest_results = rbind(train_pQC_data, test_pQC_data)
#common symbol
```

If users are interested in the whole source codes, they can visit our github: https://github.com/qade544/pseudoQC. These codes are in app.R file:



5. Group Entropy.

The weighted group entropy is calculated based on PCA score distance matrix for each group sample with a James-Stein-type shrinkage estimator in MetaboGroupS software[8] and used to evaluate different normalization methods for users. It can be deduced as below:

$$\hat{H}_{ge}^{Shrink} = -\sum_{k=1}^{g} \hat{\theta}_k^{Shrink} \log(\hat{\theta}_k^{Shrink})$$

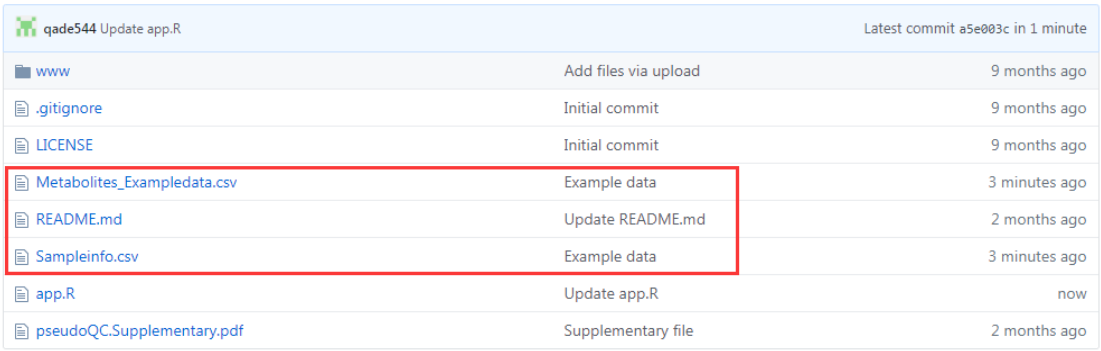
Where g is the replicate number of each group sample and θ_k is the bin frequencies of PCA score distance matrix of the *kth* group. The estimate of the shrink entropy can be calculated with entropy package in R[9], which can also be downloaded from http://www.strimmerlab.org/software/entropy/.

III. Case Study

The detailed implementation process can be found in supplementary method above. Herein we mainly analyze a published data as a real example for users to operate and understand this software better.

1. Data Preparation

This example data are belong to metabolomics data with real QC sample data, which are sourced from blood metabolite data of maintenance hemodialysis patients used in the MetaboGroup*S* platform[8] and can be download from out github (https://github.com/qade544/pseudoQC):



qade544 Update app.R		Latest commit a5e003c in 1 minute
www	Add files via upload	9 months ago
.gitignore	Initial commit	9 months ago
LICENSE	Initial commit	9 months ago
Metabolites_Exampledata.csv	Example data	3 minutes ago
README.md	Update README.md	2 months ago
Sampleinfo.csv	Example data	3 minutes ago
app.R	Update app.R	now
pseudoQC.Supplementary.pdf	Supplementary file	2 months ago

There are two kinds of data table, one is peak data and the other is sample information data. For example, CXL-pos-dingliang.csv file contains peak data, open it in Excel:

	A	L	M	N	O	P	Q	R	S	T
1	names	HX1	HX3	HX5	HX7	HX8	HX9	HX10	HX12	HX13
2	7.61_519.3323n_520.3395516	24147.68	43241.09	30406.31	71802.95	63405.6	27240.34	35193.99	21003.84	42275.68
3	9.32_538.4076n_561.3968443	673.7139	725.223	740.8863	802.5532	728.5324	671.0133	1415.216	797.9585	691.033
4	2.81_370.2204n_393.2095963	554.046	676.8951	729.7479	639.6943	738.2883	527.454	2127.968	941.4534	640.1021
5	6.22_408.2881n_431.2772898	4.580378	0	0	0	0	14.42876	7.545559	79.98482	0
6	2.54_310.1273n_333.1165333	1135.402	337.0426	1772.859	684.9352	1137.385	496.0765	502.8884	494.1148	322.7331
7	9.34_494.3817n_517.3708854	944.6639	998.5908	1050.224	925.818	1072.712	925.803	1932.03	1129.307	960.4034
8	4.62_236.1058n_259.095008	23.21685	31.76915	31.09069	30.04137	11.52148	4459.264	0.866247	27.05243	46.85369
9	8.97_528.3659n_551.3551076	2103.477	2359.567	2214.768	1951.39	2441.952	2392.231	4364.098	2454.576	2117.315
10	7.94_495.3326n_496.3398433	144777.6	127488	118231.6	205200.5	158300.1	143394.8	243274.6	111893.8	142363.5
11	8.88_704.4700n_727.4591956	980.6288	1109.764	1110.864	857.2376	1206.524	1286.299	2050.701	1244.133	979.9616
12	8.91_660.4440n_683.4332372	1250.892	1429.891	1389.049	1140.618	1482.737	1551.337	2538.441	1535.035	1280.3
13	1.73_327.0961n_366.0592746	140.0398	124.0002	111.3436	86.55886	46.36566	1207.938	0	166.9294	102.2056
14	2.95_414.2465n_437.2356828	1226.335	1226.315	1241.034	1259.203	1216.24	1155.202	2217.608	1283.826	1121.591
15	3.76_486.3039n_509.2928465	1738.907	1876.594	1925.416	991.1353	1924.902	1644.312	3866.76	2065.846	1861.184
16	8.93_616.4180n_639.4072157	1528.648	1807.749	1679.314	1371.43	1845.365	1867.644	3070.663	1923.494	1579.626
17	8.95_572.3916n_595.3807777	1896.959	2206.888	2172.971	1688.99	2321.392	2247.959	3896.499	2440.618	1999.557
18	8.39_426.2981n_449.2872957	549.3903	613.3497	604.8445	1224.764	691.8988	710.3057	1485.182	813.7781	825.6647
19	3.31_530.2586n_548.2924745	113.6082	27.7954	657.6478	55.35951	443.2037	627.0334	31.74018	125.5848	20.56846
20	3.08_458.2726n_476.3064233	1013.651	1004.549	946.1038	930.2538	951.6176	814.497	1790.795	994.8897	879.716
21	3.19_486.2297n_509.2188907	480.4696	343.264	1801.861	242.4814	1489.241	928.8548	108.4221	342.3964	394.7708
22	2.97_453.0416n_454.0488971	4.350401	0	0	0	0	0.014121	0	0	0
23	2.64_326.1944n_349.1836492	1272.05	1363.697	1292.658	1102.289	1333.388	1257.245	2184.248	1340.812	1222.025
24	7.94_577.2474n_595.2577156	435.3632	364.071	309.5823	399.7074	419.0606	353.8047	563.5003	293.2914	343.4645
25	2.91_398.1789n_421.1680938	1498.212	191.223	2934.522	586.4122	2747.624	1304.554	168.7988	302.3068	176.6568

The data can be obtained from some search softwares, such as Progenesis QI (Waters), Compound Discoverer (Thermo Fisher). There are three main parts: 1. The first column, can be the retention time, mass to charge (m/z) or their combination; 2. The first row, is usually Sample names; 3. Peaks intensities, can be obtained from the raw data with those softwares.

CXL_sampleinfo.csv file contains sample information data, a screenshot of this data in microsoft office excel is shown below.:

A	B	C	D
sample	batch	class	order
QC11	1	QC	1
QC12	1	QC	2
QC13	1	QC	3
QC14	1	QC	4
QC15	1	QC	5
QC16	1	QC	6
QC17	1	QC	7
QC18	1	QC	8
QC19	1	QC	9
QC20	1	QC	10
HX1	1	Group A	11
HX3	1	Group A	12
HX5	1	Group A	13
HX7	1	Group A	14
HX8	1	Group A	15
HX9	1	Group A	16
HX10	1	Group A	17
HX12	1	Group A	18
HX13	1	Group A	19

This data contain four columns (sample, batch, class, order). We highly recommend users to prepare their sample data like this and use same column names (case sensitive) and sequence. Here are what each column means:

sample: the sample names, same as the first row in peaks data.

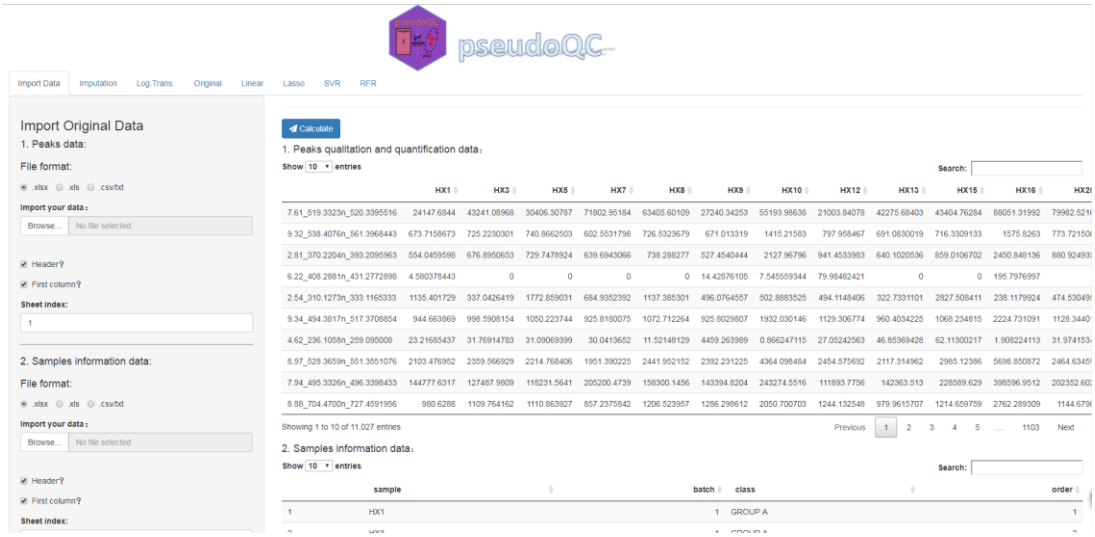
batch: whether the samples are processed at the same time, if so, they should be marked with same labels (e.g. label “1”), otherwise, marked with different labels (labels “1”, “2”, …). The number labels are recommendatory.

class: the group information of these samples, different group should be marked with different labels.

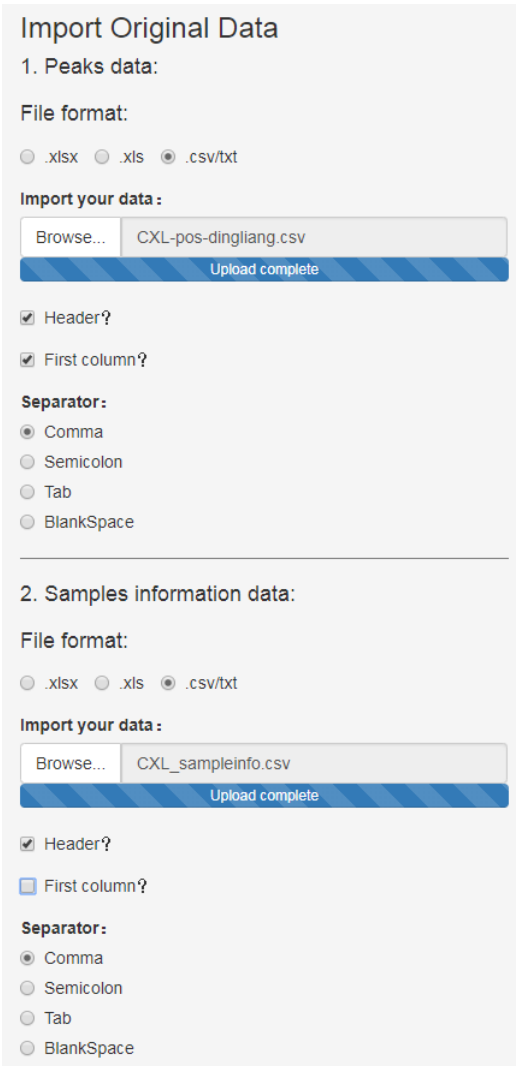
order: this just records the sequence of the samples, users can put the same order as what they upload into mass spectrometer or just serial numbers here.

2. Uploading Data

After preparing data, users can open pseudoQC software through https://www.omicsolution.org/wukong/pseudoQC/. Thus this requires that users should have access to the Internet.



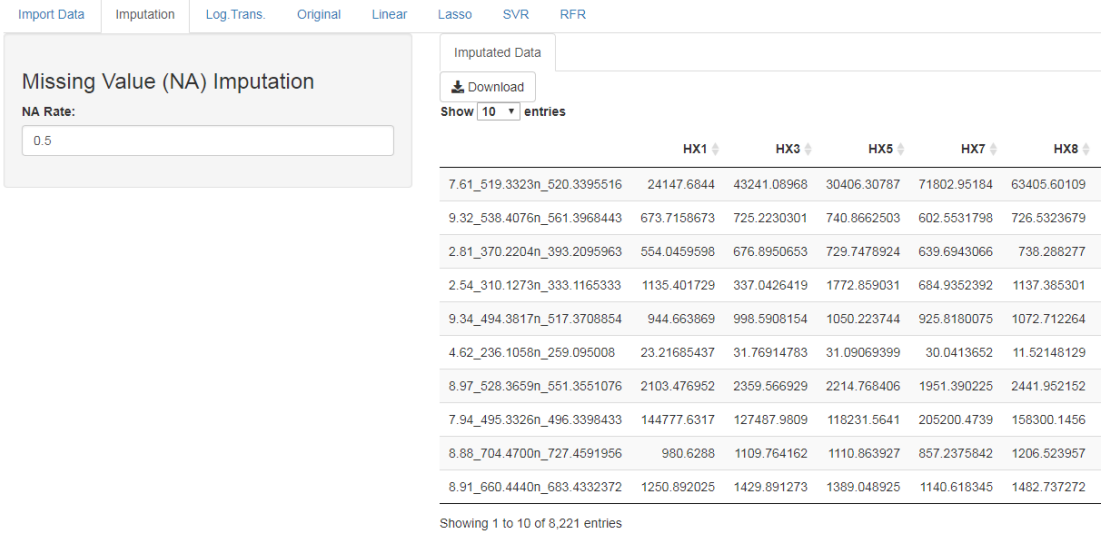
Then users can upload their own data in the parameter panel:



As our example data are saved in .csv files, we need choose the '.csv/txt' format here. Users should choose right file format based on their own data file. To date, several common formats including .xlsx, .xls, .csv and .txt are supported in pseudoQC.

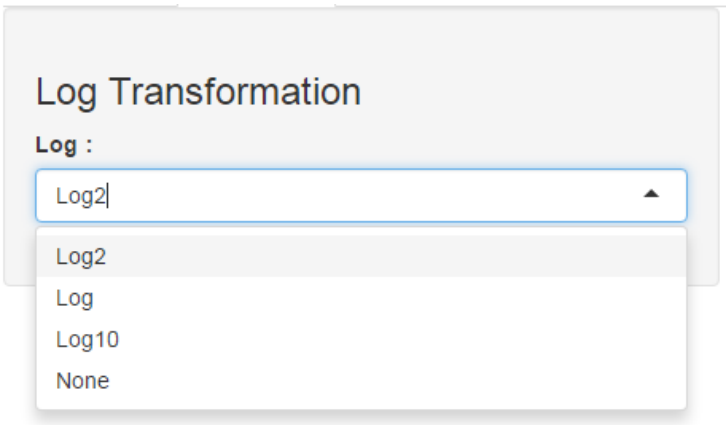
3. Missing Value Imputation

As described in supplementary method, those features whose NAs ratio is above 0.5 will be removed and the imputation is implemented with the k-Nearest Neighbor (KNN) algorithm. This ratio can also be adjusted by users, for example, if they want to control the ratio more rigorously, this parameter should be lower.

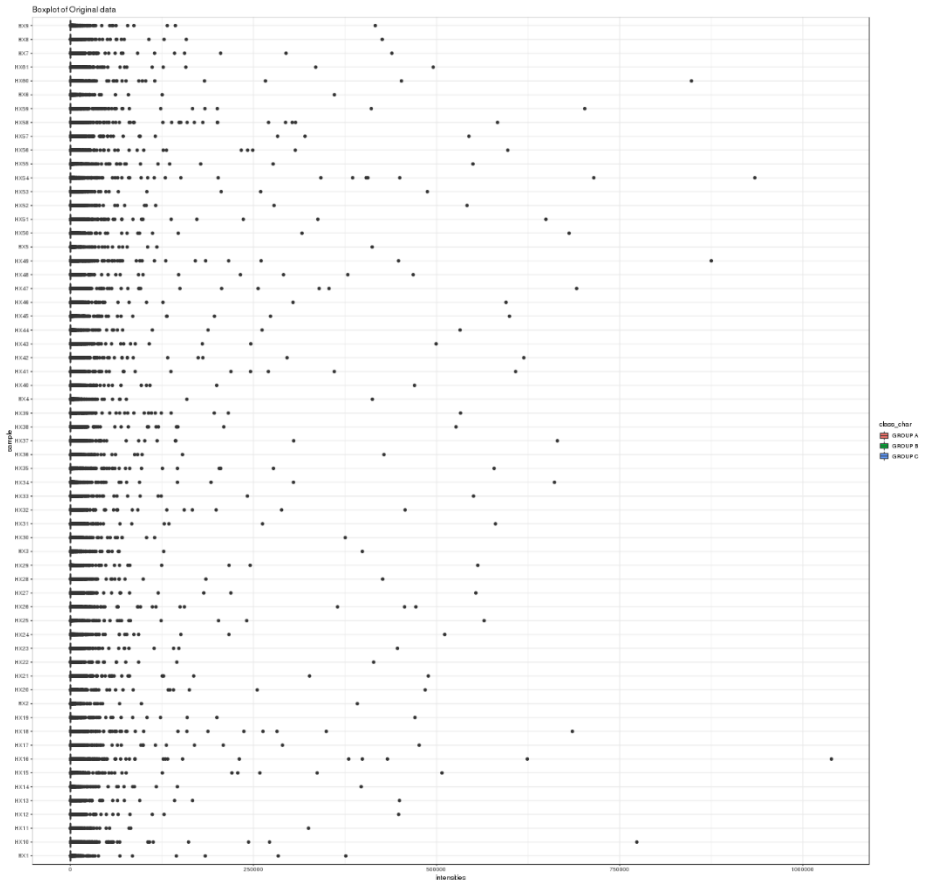


4. Log Transformation

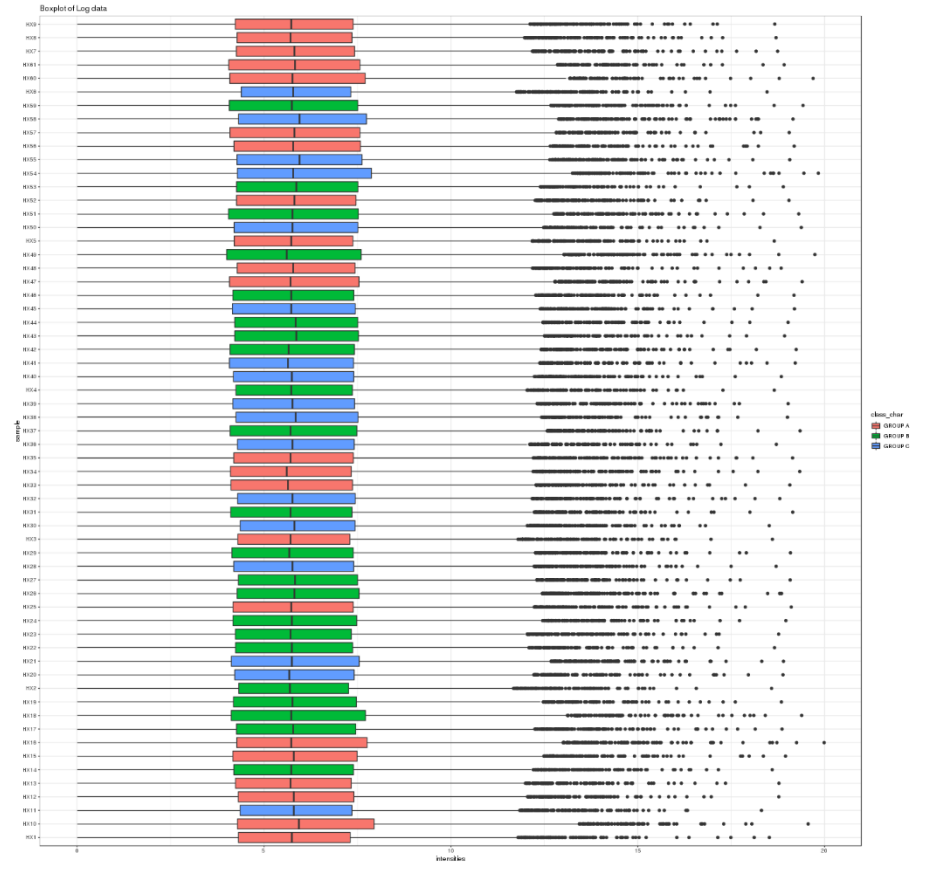
In pseudoQC, users can transform their raw intensities to log value here. Three log-transformation types are supported: Log2, Log, Log10, are log-base 2, e, and 10 respectively. "None" means no log transformation. As shown below:



From this part, users can obtain the boxplot of peak intensities in every sample before log transformation:



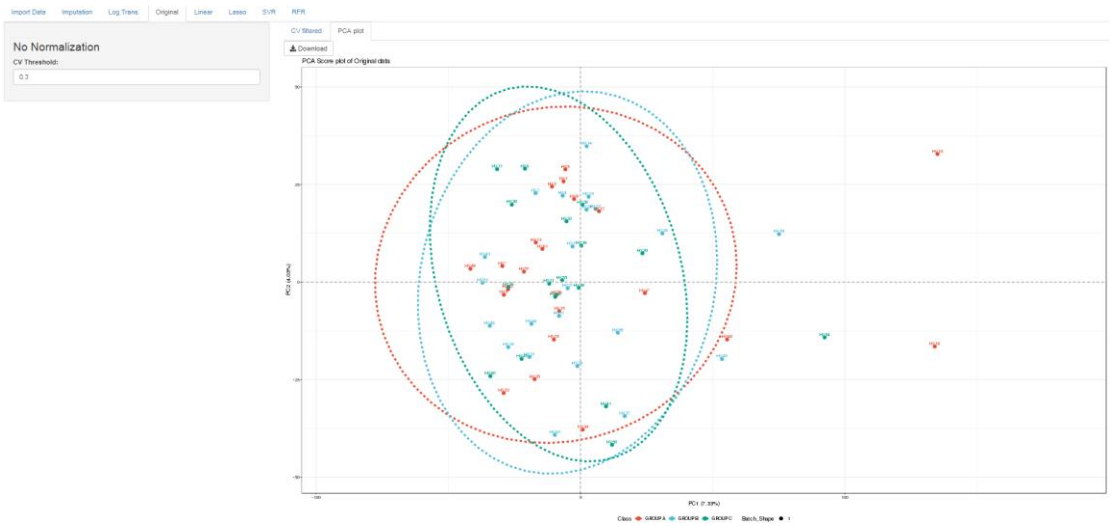
And after log transformation:



5. QC Data Simulation

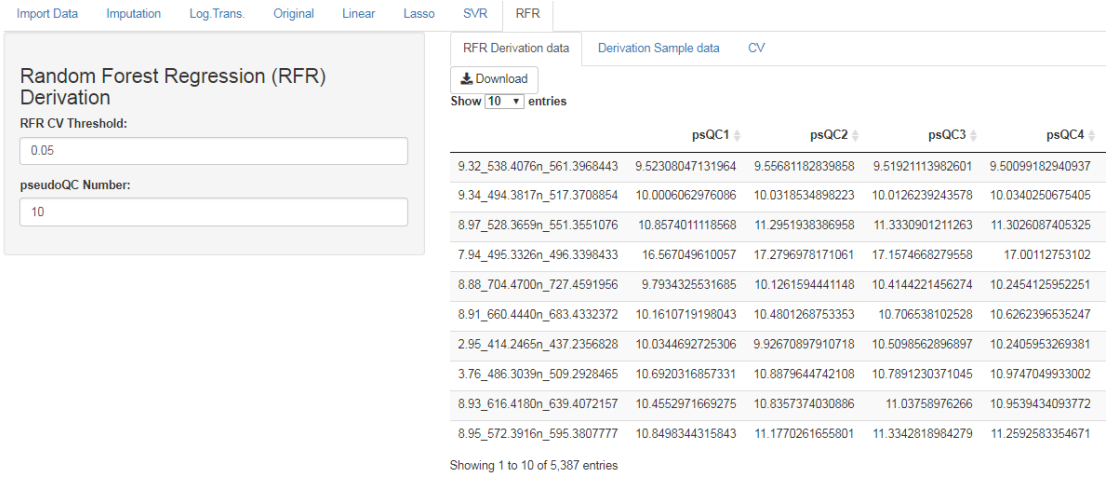
Four regression methods are implemented here for users to simulate the QC data based on the upload data.

Before simulation, pseudoQC also allows users to check the original data:



Coefficient of variation is calculated here for every peaks and those above 30% will be removed. Then the remaining data are used for PCA analysis to check these samples distribution.

Next users can click the regression buttons orderly:



In each regression method, users can adjust the CV threshold and pseudoQC Number. Then the QC simulation data can be derived based on original data, which will be labeled with “psQC”:

RFR Derivation data | Derivation Sample data | CV

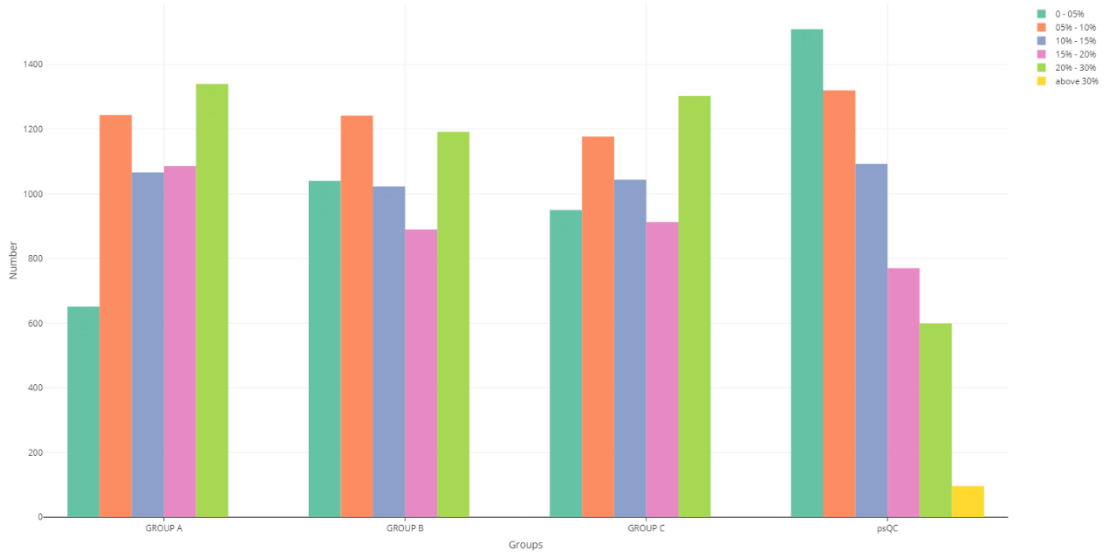
Download

Show 10 entries Search:

	psQC1	psQC2	psQC3	psQC4	psQC5	psQC6	psQC7	psQC8	psQC9	psQC10
9.32_538.4076n_561.3968443	9.52308047131964	9.55681182839858	9.51921113982601	9.50099182940937	10.1334988968141	9.42259058609819	9.5137258193247	9.53418635215534	9.80362949502027	9.56640580807767
9.34_494.3817n_517.3708854	10.0006062976086	10.0318534898223	10.0126239243578	10.0340250675405	10.6601027825428	9.88947001123376	10.025690208089	10.0744800040153	10.2905231881009	10.0498567101817
8.97_528.3659n_551.3551076	10.8574011118568	11.2951938386958	11.3330901211263	11.3026087405325	12.2253968737466	11.3773622380664	11.548663816343	11.5265167308521	11.8269706950346	11.8098643215154
7.94_495.3326n_496.3398433	16.567049610057	17.2796978171061	17.1574668279558	17.00112753102	18.1060007063325	17.2840926424552	17.1499075217107	17.1049995174952	17.7246920220431	16.7732572194997
8.88_704.4700n_727.4591956	9.7934325531685	10.1261594441148	10.4144221456274	10.2454125952251	11.2876192528873	10.4185394748951	10.5460536463978	10.6367949327833	10.9321386562889	11.0570489613995
8.91_660.4440n_683.4332372	10.1610719198043	10.4801268753353	10.706538102528	10.6262396535247	11.5855713877032	10.7270464074767	10.8945379548718	10.9622334053684	11.2296936138483	11.3363002831886
2.95_414.2465n_437.2356828	10.0344692725306	9.92670897910718	10.5098562896897	10.2405953269381	10.2646673194239	9.99775160075315	10.2098527955112	10.1083197089262	10.5041968632996	10.4703469294669
3.76_486.3039n_509.2928465	10.6920316857331	10.8879644742108	10.7891230371045	10.9747049933002	11.5131331211248	10.8100008025917	11.0090809502105	10.9683931202422	11.191083163811	10.8090450311266
8.93_616.4180n_639.4072157	10.4552971669275	10.8357374030886	11.03758976266	10.9539434093772	11.8957570747359	11.0071867480857	11.1931900717407	11.2507526715445	11.512739098118	11.6323595445125
8.95_572.3916n_595.3807777	10.8498344315843	11.1770261655801	11.3342818984279	11.2592583354671	12.1095905861712	11.2596849769277	11.4640608906612	11.495726889096	11.7527605551476	11.8233244685202

Showing 1 to 10 of 5,387 entries Previous 1 2 3 4 5 ... 539 Next

In addition, after simulation, pseudoQC also calculates its CV for users to check the simulation quality:



All main results (tables and figures) can be downloaded to local computer when users click 'Download' button for further analysis, i.e. uploading these simulation data into MetaboGroup*S* platform to calculate the group entropy.

IV. References

[1] A. M. De Livera, G. Olshansky, J. A. Simpson, D. J. Creek, Metabolomics : Official journal of the Metabolomic Society 2018, 14, 54.

[2] H. Abdi, Encyclopedia of research design 2010, 1, 169.

[3] G. Wilkinson, C. Rogers, Applied Statistics 1973, 392.

[4] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, The Annals of statistics 2004, 32, 407.

[5] C.-C. Chang, C.-J. Lin, ACM transactions on intelligent systems and technology (TIST) 2011, 2, 27.

[6] L. Breiman, Machine learning 2001, 45, 5.

[7] G. Wilkinson, C. Rogers, Journal of the Royal Statistical Society: Series C (Applied Statistics) 1973, 22, 392.

[8] S. Wang, X. Chen, D. Du, W. Zheng, L. Hu, H. Yang, J. Cheng, M. Gong, Anal Chem 2018.

[9] J. Hausser, K. Strimmer, Journal of Machine Learning Research 2009, 10, 1469.