User Manual of CREAM

Contents

1 Introduction

CREAM is an analytical framework for detecting co-regulated gene modules and performing subsequent analyses. This system contains three executable programs: ‘CREAM.jar’ is used to detect co-regulated gene modules from gene/protein expression data; and ‘CREAM_dcanalysis.jar’ and ‘CREAM_survival.jar’ are used to perform differential co-expression analysis and survival analysis on the resulting modules. The source code and the complied tool can be downloaded at https://github.com/free1234hm/CREAM.

2 Preliminaries

- To use CREAM a version of Java 1.5 or later must be installed. If Java 1.5 or later is not currently installed, then it can be downloaded from http://www.java.com.
- CREAM can be executed by double-clicking on ‘CREAM.jar’, or from a command line change to the CREAM directory and then type: java -mx1024M -jar CREAM.jar. If CREAM reports ‘Out of Memory Error’, users can increase the -mx parameter.

3 CREAM.jar

3.1 Main Interface

The main interface has three parts: ‘Load Data’, ‘Set Parameters’, and ‘Search Gene Modules’.

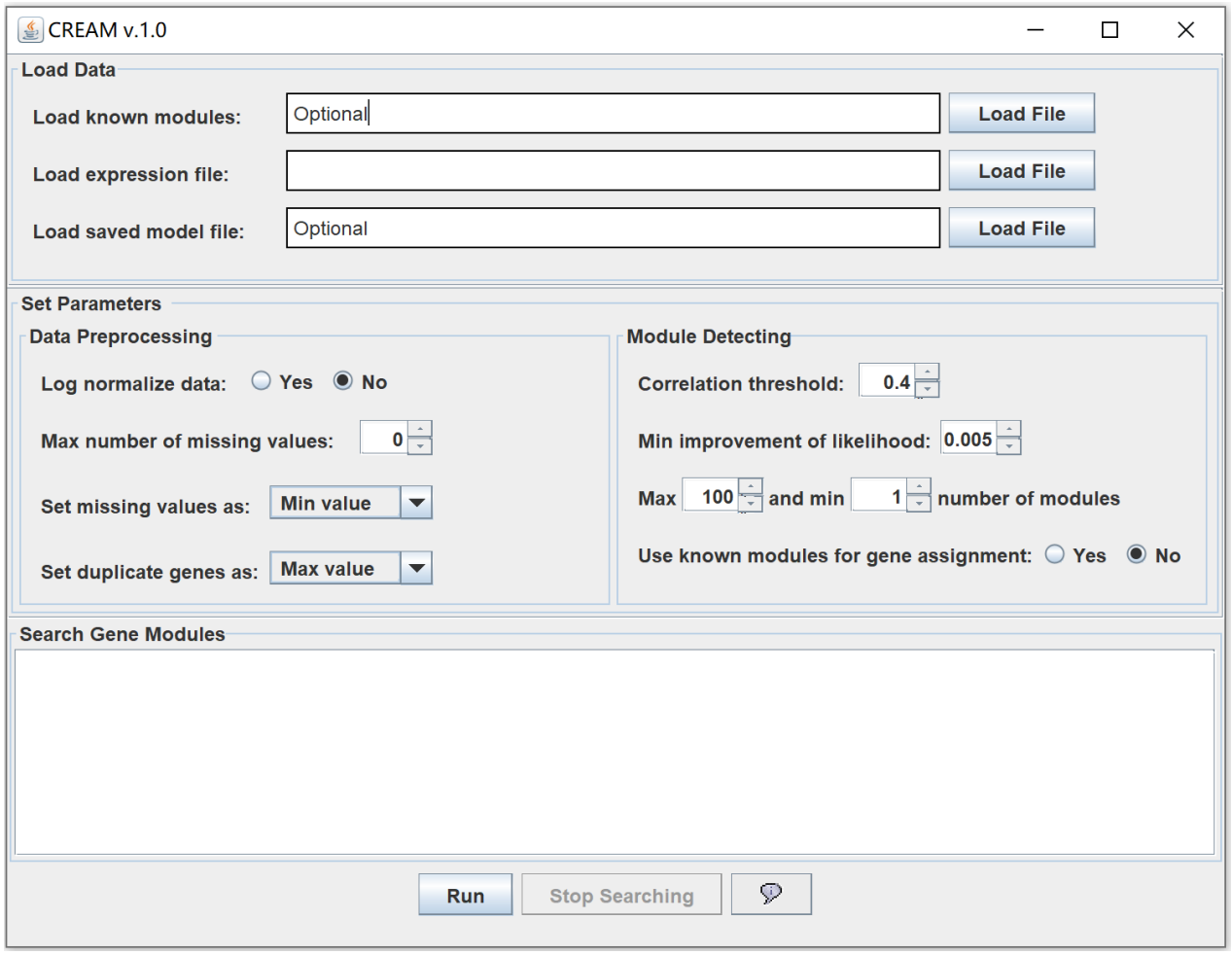


Figure 1. The main interface of CREAM

3.1.1 Load known modules

Import known module files, including GO annotation, KEGG pathway information, transcription factor (TF)– and microRNA (miRNA)–target interaction information. The source of these information can either be user provided or one of the files present in the ‘Known gene modules’ directory of the CREAM directory. The known module files can be in one of two formats, a grid format or a three-column format. In the grid format the columns correspond to module name such as TFs, miRNAs or pathways, and the rows correspond to genes. The first column contains gene symbols. Using TF as an example, an entry of 1 corresponds to the prediction that the TF does regulate the gene. Users can differentiate between positive and negative interactions by using a ‘1’ for positive correlation and ‘-1’ for negative interactions. (Fig. 2)

ID	Sin3a	Nr1i3	Ctnnb1	Cebpb	Cebpa	Ets2	Hif1a
Gamt	0	0	0	0	0	0	0
Adora2b	0	0	0	0	0	0	0
Erbb2	0	0	1	0	0	0	0
Psme3	0	0	0	0	0	0	0
Dqx1	0	1	0	0	0	0	0
Erbb3	0	1	0	0	0	0	0
6330439K1	0	0	0	0	0	0	0
Psme2	0	0	0	0	0	0	0
Luc7l	1	0	0	0	0	0	0
Smc6	0	0	0	0	0	0	1
Vps35	0	0	1	0	0	0	1
Smc3	0	0	0	0	0	0	0
Elk3	0	0	0	0	0	0	0
Smc2	1	0	0	0	0	0	0
Elk4	0	0	0	0	0	0	0
Ipw	0	0	0	0	0	0	0

Figure 2. A sample of known module file in grid format when viewed in Microsoft Excel.

In the three-column format the first column contains module name, the second column the regulated gene, and the third column input value. The first row is a header row where the header of the first column can be ‘TF’, ‘miRNA’ and ‘pathway’ et al, and the second column must have the header ‘Gene’. A value of ‘1’ represents that the TF-gene pair shows positive correlation interaction, while a value of ‘-1’ represents that the TF-gene pair shows negative correlation interaction. (Fig. 3).

TF	Gene	Input
Nr1i3	Otc	1
Creb3l3	Leap2	1
Nr1i3	Ugt2a3	1
Nr1i3	Leap2	1
Nr1i3	Rdh7	1
Creb3l3	Apoa4	1
Nr1i3	Cyp3a25	1
Nr1i3	Cps1	1
Nr1i3	Hsd17b6	1

Figure 3. A sample of TF-gene data file in three-column format when viewed in Microsoft Excel.

3.1.2 Load expression file

The ‘Load expression file’ field is used to Import a gene/protein expression data. The first column is gene names, and the remaining columns contain the expression values in each sample. If an expression value is missing, then the field should be left empty. The first row of the data contains column headers. A sample expression data file is shown in Figure 4.

Ensembl_II	TCGA-AA-	TCGA-AA-	TCGA-A6-	TCGA-A6-	TCGA-AA-	TCGA-CK-	TCGA-AA-	TCGA-AA-	TCGA-AU-	TCGA-QG
A1CF	0.92999	1.066218	2.564578	2.242078	1.208603	0.159212	2.189218	0.061351	0.889145	1.183308
A2M	4.529061	4.870139	5.687503	6.662382	5.405043	5.591557	3.445681	5.108467	4.960791	3.854576
A4GALT	2.662687	1.147233	1.104459	2.482948	1.853063	1.383715	0.828363	2.687003	2.098147	1.040627
AAAS	3.833031	3.628329	3.198295	3.443551	4.099529	3.6168	3.774873	3.734673	3.361572	3.774655
AACS	2.145268	2.235108	2.06301	1.917419	2.63819	1.72618	2.344681	1.980136	2.0168	2.221353
AADAT	0.901021	2.117209	2.33651	1.226698	1.584483	3.170597	2.184566	1.468223	2.000801	2.486626
AAGAB	3.384627	4.911665	4.380784	3.860188	3.745479	3.901575	4.193553	4.471359	3.947197	3.8725
AAK1	0.891756	0.938476	2.48266	1.607293	0.550033	1.147723	1.93236	1.247271	1.246245	1.121613
AAMDC	2.905587	2.386935	3.092975	2.742403	2.883851	2.447846	2.444698	3.620468	1.942847	2.423223
AAMP	6.108965	5.606741	5.733432	5.76106	5.797197	5.758501	5.858539	6.147721	5.74096	5.935313
AAR2	4.62933	3.917359	5.064283	4.874728	4.248512	3.939791	4.888974	4.012086	3.845083	3.927044
AARS2	2.320018	2.451048	3.10118	2.902352	3.031804	3.035382	2.926849	3.194291	2.81012	3.086282
AARSD1	0.78421	0.60167	0.864706	0.742812	0.932212	0.947192	1.105827	0.979237	1.248157	1.35347
AASDH	1.038219	1.847106	2.895586	1.67843	1.136062	1.625119	2.044143	1.792878	1.646203	1.880203
AASDHPPT	2.52936	2.220441	3.44982	2.819377	1.674377	3.527966	2.598485	2.700203	2.565926	2.825342
AATF	3.834741	3.709372	4.429679	4.393276	4.616406	4.195666	4.284369	4.555975	4.052856	4.076867
ABAT	1.619443	0.532441	2.368285	3.182818	2.357015	0.972122	2.26131	0.342083	0.986267	1.378474
ABCA1	1.230966	1.153374	2.143139	1.730441	0.891639	2.322669	1.32176	1.31	1.863681	1.358856
ABCA2	2.71114	4.202821	1.586837	3.293604	2.069393	3.293773	4.021893	3.098194	2.880384	3.48955

Figure 4. A sample of expression data file when viewed in Microsoft Excel.

3.1.3 Load saved model file

The 'Load saved model file' field allows the user to specify a file containing a set of saved expression patterns (Figure 5), thus saving time if the co-regulated gene modules of an expression data have already been detected.

Module1	0.03967	0.405222	-0.36234	1.190898	1.900636	-0.91111	0.672894	-1.0704	0.189161	1.686573
Module2	0.015905	0.148512	0.422478	-0.51482	-1.2356	-1.76675	-0.64225	-2.28238	0.891457	0.299482
Module3	0.16417	0.200187	-0.50583	0.835397	0.357377	-1.31913	0.328816	0.166863	0.135561	0.692104
Module4	-0.11234	-0.25724	1.566917	-0.32842	0.337924	0.938745	0.004372	-0.05839	0.066478	-0.29413
Module5	0.496984	0.537191	-0.27244	-0.16935	-0.90829	-1.64771	0.152171	-1.46547	1.079376	0.991317
Module6	0.370578	0.041303	-0.51396	-0.45961	-2.11782	0.113174	-0.38189	0.032023	0.433214	-0.56686
Module7	-0.36829	-0.33748	0.563487	-0.39259	1.455291	0.72515	0.397169	-0.55123	0.146438	0.235769
Module8	0.541632	1.605993	0.162757	0.77692	0.789027	0.208924	0.083275	-0.12239	-0.96627	0.982543
Module9	0.284635	0.149303	-0.41945	0.311363	0.592805	0.122872	0.269897	-1.0626	0.882116	1.319755
Module10	0.568635	0.012038	-0.59944	-0.18891	-1.31881	0.55664	-0.60431	0.281863	0.277769	-0.12116
Module11	0.27862	0.247257	-0.16462	0.028241	-0.77162	-4.16876	-0.08779	-1.07862	0.530419	0.584825
Module12	0.290281	-0.00483	-0.22937	-0.23632	-1.27911	-1.96178	0.019694	-0.23806	0.199837	0.143658
Module13	0.31105	0.031979	0.42246	0.294031	0.831223	-0.44424	0.557385	-0.77007	-0.62021	0.203346
Module14	-0.33104	0.585915	0.724734	0.672033	1.736791	0.149426	0.415488	-0.69827	0.51792	0.647048
Module15	0.448937	0.074394	-0.29808	-0.83848	-2.10951	0.178222	-0.45699	0.583738	-0.24477	-1.26191
Module16	0.807964	-0.3037	-0.36618	-0.687	-1.93105	0.88753	-1.06	1.01541	0.01819	-0.47825
Module17	-0.125	-1.00518	0.455515	-0.49126	-0.64816	0.307793	0.261693	0.789195	-0.63921	-1.49034
Module18	0.420021	0.43557	-0.25491	0.689612	1.369693	-0.88308	0.799794	-1.28331	0.499949	1.714974
Module19	1.396891	1.429818	-0.20104	-0.39985	-0.5255	0.476891	0.508937	-0.05945	-0.26513	0.655126
Module20	0.15136	0.729316	-0.88443	-0.10305	-0.12302	0.278122	-0.40627	-1.01897	0.489385	0.123519
Module21	-0.24293	-0.68236	0.263083	0.050511	0.041583	0.675133	-0.05787	0.489086	-0.05948	-0.88593

Figure 5. An example of the save module file

3.1.4 Set Parameters

Through the parameters on the 'Data Preprocessing' panel the user can adjust the criteria for filtering genes. If a gene is filtered, then it will be excluded from further analysis. Assuming that the expression vector of a gene is $\{v_1, v_2, \ldots, v_n\}$.

- Log normalize data—transforms the vector to $\{\log_2 v_1, \ldots, \log_2 v_n\}$.
- Maximum number of missing values—a gene will be filtered if the number of missing values in all samples exceeds this parameter.

- Set missing value as—the missing value of a gene can be set as ‘min value’ (default), ‘mean value’ or ‘zero’.
- Set duplicated genes as—combine the duplicate expression profiles of a gene based on the ‘mean value’ or ‘max value’ (default).
- Correlation threshold—among the genes assigned to an expression pattern (p_1), genes showing absolute correlation coefficients higher and lower than this threshold (0.4 as default) with p_1 are divided into two groups, which are further applied to construct two new expression patterns. This parameter influences the final module number.
- Min improvement of likelihood—after each iteration of adding a new expression pattern, if the highest improvement of likelihood is lower than this parameter (5‰ as default), stop the iteration process. This parameter influences the final module number.
- Max and min number of modules—the maximum (100 as default) and minimum (1 as default) number of final modules.
- Use known modules for gene assignment—if ‘Yes’ is selected, the known module information (e.g., transcriptional regulation and biological pathways) will influence the assignment of genes to different expression patterns.

3.1.5 Search Gene Modules

The text box here displays the running progress of CREAM after pressing the ‘Run’ button. And there is a ‘Stop Searching’ button controlling the model training. Pressing the ‘Stop Searching’ Button forces CREAM to stop adding expression patterns. CREAM then proceeds to the gene assignment phase.

3.2 Display Interface

After the module detection process executes, the main output window appears. An example of such a window is shown in Figure 6. Figure 6A displays the detailed module information. First, users can select an item in the module table to display two sets of genes positively and negatively correlated with the selected expression pattern (Figure6B). Alternatively, users can convert the gene tables to gene expression heatmaps by clicking the ‘HeatMap’ button (Figure 6C). To our knowledge, CREAM is the first tool capable of separately visualizing the positively and negatively correlated genes in the same module. Second, user can press the ‘Enrichment Analysis’ button to check the functional enrichment analysis results based on the user uploaded known module files (Figures 6D and E). Third, users can press the ‘Save Model’ button to save the final expression patterns. Figure 5 shows an example of the saved model file, which can be imported in CREAM through ‘Load saved model file’ for the next time we analyze this data to save time.

Figure 6. An example of the Display Interface.

4 CREAM_dcanalysis.jar

4.1 Main Interface

'CREAM_dcanalysis.jar' is used to evaluate whether genes in a module are significantly differentially correlated between conditions. Specifically, a set of genes are highly correlated across sample in one condition and significantly lowly correlated across sample in another condition. The main interface has three parts: 'Load Data', 'Data Preprocessing', and 'Analysis' (Figure 7).

4.1.1 Load expression data 1/2

The 'Load expression data 1/2' field is used to import the expression datasets of two comparable biological conditions. The data format is same as that of CREAM (see section 3.1.2). A sample expression data file is shown in Figure 4. Notably, genes in a differentially co-expressed module must be highly correlated in expression data 1 and lowly correlated in expression data 2. Therefore, the expression data 1 and 2 files cannot be exchanged.

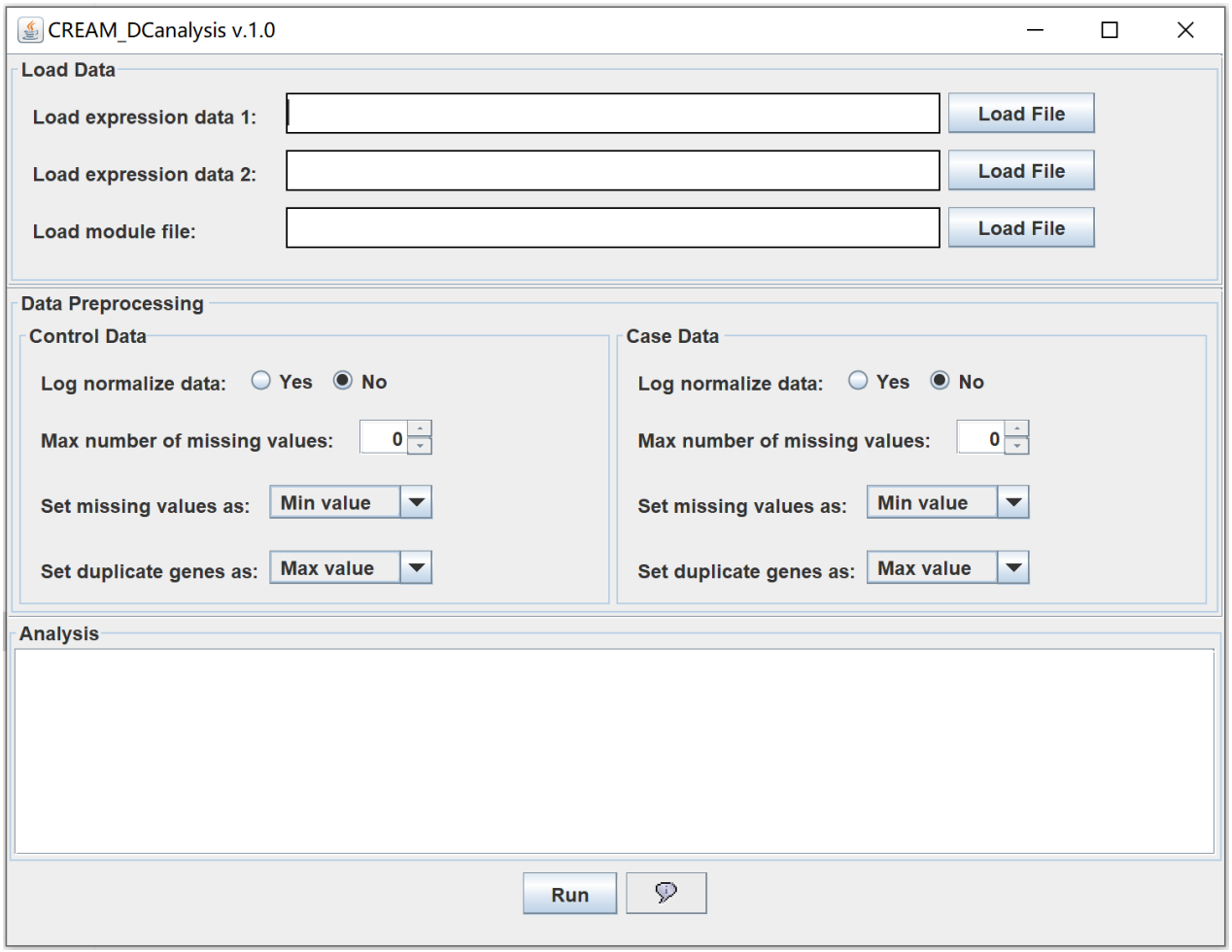


Figure 7. The main interface of CREAM_dcanalysis.jar

4.1.2 Load module file

The 'Load module file' field is used to import module information. The file should be in a three-column format (Figure 8): the first column contains module name, the second column the regulated gene, and the third column regulation value. Value of 1 and -1 represent that genes are positively and negatively regulated in the module.

Module name	Gene	Correlatior
Module1	UBE2D3	1
Module1	UBE2E1	1
Module1	TRIM37	1
Module1	UBE2E3	1
Module1	UBE2K	1
Module1	UBE2Q2	1
Module1	UBE2R2	1
Module2	MUTYH	1
Module2	NEIL3	1
Module2	PARP1	1
Module2	PARP2	1
Module2	POLD3	1
Module2	UNG	1
Module2	OGG1	1
Module3	ATG12	1
Module3	PTEN	1

Pathway	Gene	Correlatior
Glycolysis / Gluconeogenesis	PKM	1
Glycolysis / Gluconeogenesis	GPI	1
Glycolysis / Gluconeogenesis	ALDOA	1
Glycolysis / Gluconeogenesis	ALDOC	1
Glycolysis / Gluconeogenesis	PFKP	1
Glycolysis / Gluconeogenesis	PGAM1	1
Glycolysis / Gluconeogenesis	TPI1	1
Glycolysis / Gluconeogenesis	ENO1	1
Glycolysis / Gluconeogenesis	ENO2	1
Spliceosome	WBP11	1
Spliceosome	SNRPA1	1
Spliceosome	SNRPD1	1
Spliceosome	SNW1	1
Spliceosome	RBM17	1
Spliceosome	TRA2B	1
Spliceosome	PRPF4	1
Spliceosome	BCAS2	1
Spliceosome	THOC1	1
Spliceosome	SMNDC1	1
Spliceosome	SRSF1	1
Spliceosome	SRSF3	1

Figure 8. Two sample module files

4.1.3 Data Preprocessing

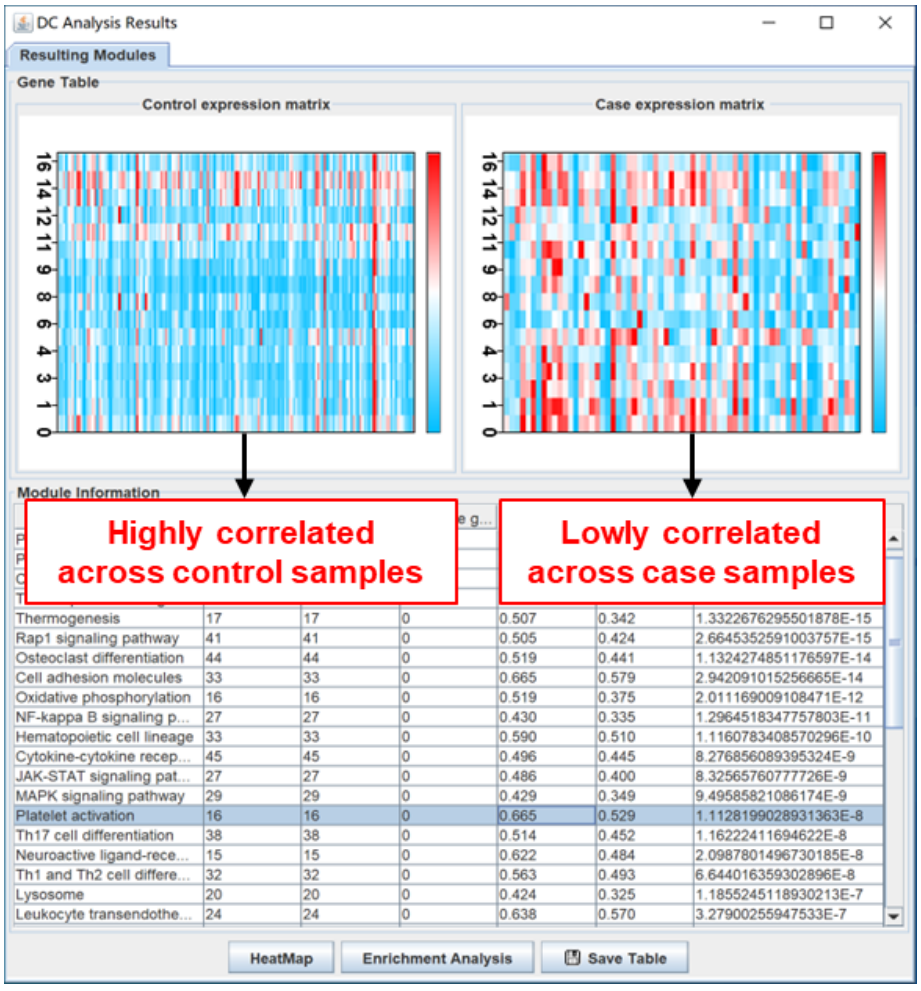
The parameters on the 'Data Preprocessing' panel are same as those of CREAM (see section 3.1.4). The user can adjust the criteria for filtering genes in control and case data, respectively.

4.1.4 Analysis

The text box here displays the running progress of differential co-expression analysis after pressing the 'Run' button.

4.2 Display Interface

Figure 9 shows an example of the interface that displays the differentially co-expressed modules. The table below shows the differential co-expression p-values of the tested modules. Users can select an item in the table to display the expression heatmaps of the selected module in expression data 1 and 2 (Figure 9).



Thermogenesis	17	17	0	0.507	0.342	1.3322676295501878E-15
Rap1 signaling pathway	41	41	0	0.505	0.424	2.6645352591003757E-15
Osteoclast differentiation	44	44	0	0.519	0.441	1.1324274851176597E-14
Cell adhesion molecules	33	33	0	0.665	0.579	2.942091015256665E-14
Oxidative phosphorylation	16	16	0	0.519	0.375	2.011169009108471E-12
NF-kappa B signaling p...	27	27	0	0.430	0.335	1.2964518347757803E-11
Hematopoietic cell lineage	33	33	0	0.590	0.510	1.1160783408570296E-10
Cytokine-cytokine recep...	45	45	0	0.496	0.445	8.276856089395324E-9
JAK-STAT signaling pat...	27	27	0	0.486	0.400	8.32565760777726E-9
MAPK signaling pathway	29	29	0	0.429	0.349	9.49585821086174E-9
Platelet activation	16	16	0	0.665	0.529	1.1128199028931363E-8
Th17 cell differentiation	38	38	0	0.514	0.452	1.16222411694622E-8
Neuroactive ligand-rece...	15	15	0	0.622	0.484	2.0987801496730185E-8
Th1 and Th2 cell differe...	32	32	0	0.563	0.493	6.644016359302896E-8
Lysosome	20	20	0	0.424	0.325	1.1855245118930213E-7
Leukocyte transendothe...	24	24	0	0.638	0.570	3.27900255947533E-7

Figure 9. An example of the Display Interface.

The user can also perform enrichment analysis for the selected module by pressing the 'Enrichment Analysis' button. An example enrichment analysis file can be found in the folder 'Enrichment_analysis_files'. Users can replace this file with their own gene annotation files. Just make sure the new file is in the same three-column format.

5 CREAM_survival.jar

5.1 Main Interface

'CREAM_survival.jar' is used to evaluate whether the correlations between genes in a module are significantly correlated with patient survival. The main interface has three parts: 'Load Data', 'Set parameters', and 'Analysis' (Figure 10).

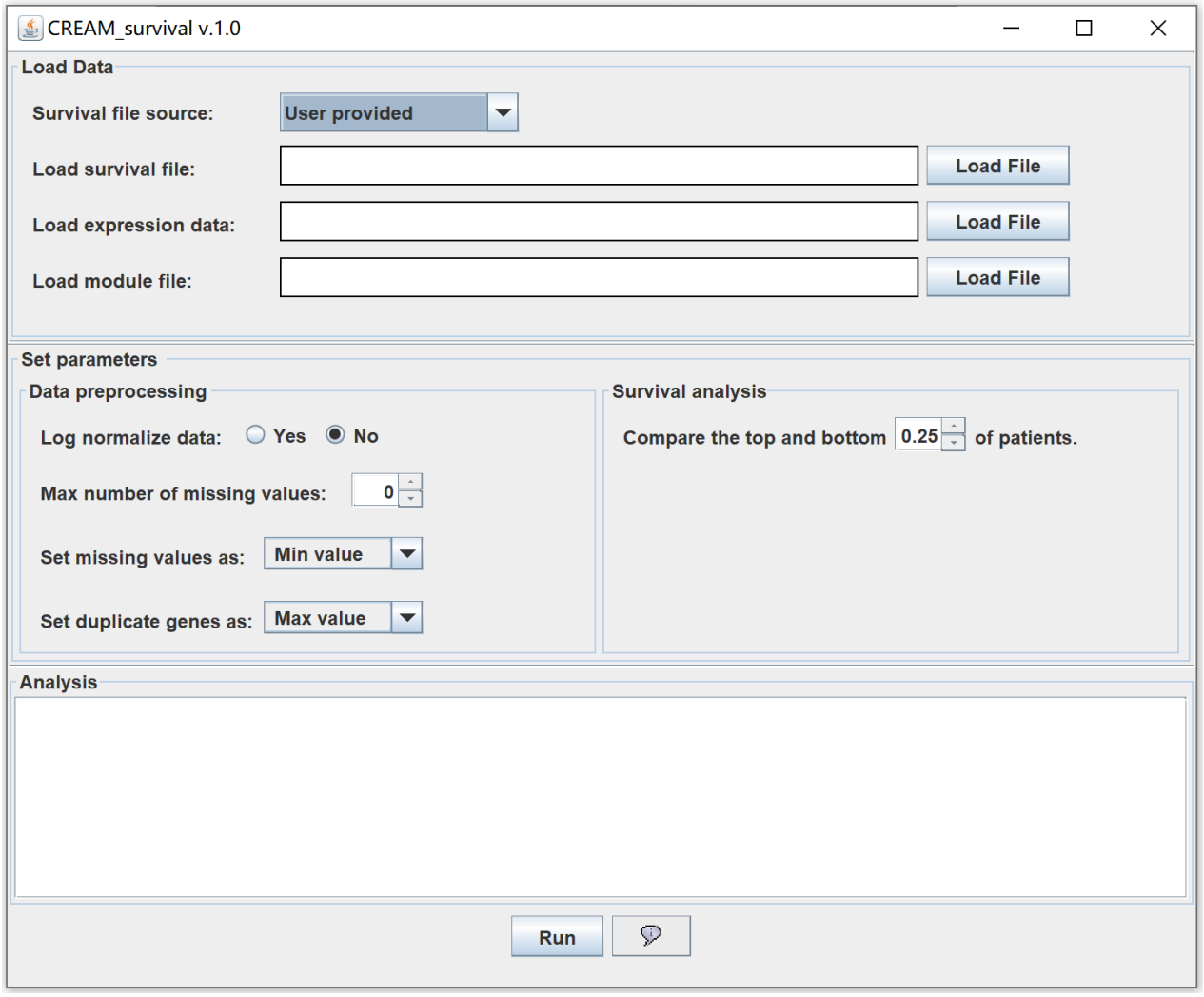


Figure 10. The main interface of CREAM_survival.jar

5.1.1 'Survival file source' and 'Load survival file'

The survival information of all patients is input. The source of these predictions can either be user provided or one of the files that currently is present in the 'TCGA survival data' directory of the CREAM directory. If User Provided' is selected, then the 'Load survival file' field is editable and a user can select the file directory. The survival file should be in a three-column format (Figure 11).

sample	OS	OS.time
TCGA-AA-3492-01A	1	1
TCGA-AA-3492-11A	1	1
TCGA-G4-6626-01A	1	1
TCGA-AA-3525-01A	0	1
TCGA-AA-3525-11A	0	1
TCGA-AY-6196-01A	0	6
TCGA-AM-5820-01A	0	14
TCGA-F4-6854-01A	0	16
TCGA-AA-A02O-11A	0	28
TCGA-AA-A02O-01A	0	28
TCGA-AM-5821-01A	0	28
TCGA-AY-4071-01A	1	29
TCGA-AA-A00R-01A	0	30
TCGA-AA-3543-01A	0	30
TCGA-AA-3818-01A	1	30

Figure 11. An example of the survival file

5.1.2 Load expression file

The 'Load expression file' field is used to import the expression data of all patients. The data format is same as that of CREAM (see section 3.1.2). A sample expression data file is shown in Figure 4. Notably, the patient IDs in the first row of the data should match those in the survival data.

5.1.3 Load module file

The 'Load module file' field is used to import module information. The data format is same as that of 'CREAM_dcanalysis.jar' (see section 4.1.2). Two sample module files are shown in Figure 8.

5.1.4 Data Preprocessing

The parameters on the 'Data Preprocessing' panel are same as those of CREAM (see section 3.1.4).

5.1.5 Survival analysis

In survival analysis, all patients are ranked based on the co-expression level of genes in a module, and then two groups of patients (A and B) are extracted. Groups A and B contains the top and bottom p (a user-defined percentage) patients, respectively. Finally, the survival time of these two groups are compared through log-rank test.

5.1.6 Analysis

The text box here displays the running progress of module-based survival analysis after pressing the 'Run' button.

5.2 Display Interface

Figure 12 shows an example of the interface that displays the survival analysis results. The table below shows the log-rank p-values of the tested modules. Users can select an item in the table to display the expression heatmaps of group A (genes show high co-expression level) and group B (genes show low co-expression level).

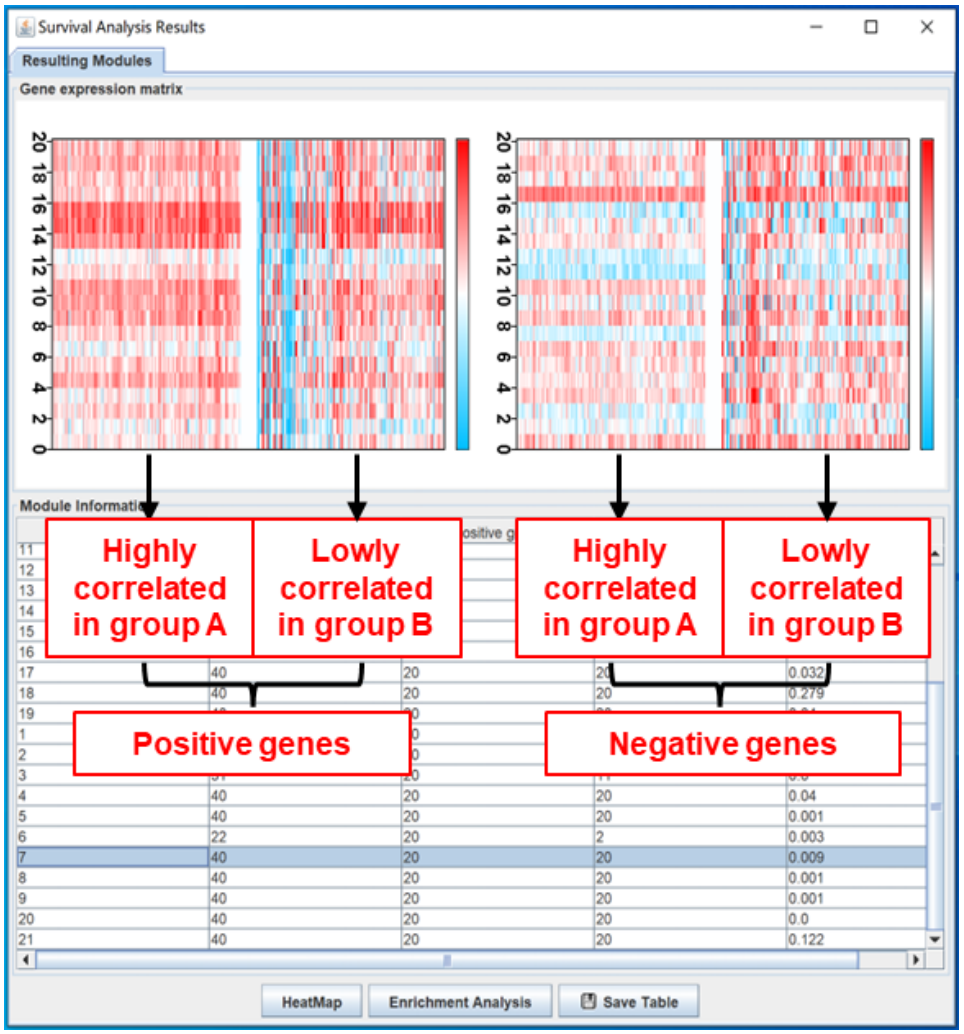


Figure 12. An example of the Display Interface.

The user can also perform enrichment analysis for the selected module by pressing the 'Enrichment Analysis' button. An example enrichment analysis file can be found in the folder 'Enrichment_analysis_files'. Users can replace this file with their own gene annotation files. Just make sure the new file is in the same three-column format.