

# DIGS (Differential Gene Signatures) User Guide

7 Oct, 2014

For any queries about this tool, please contact:

Prof Lazaros G. Papageorgiou at l.papageorgiou@ucl.ac.uk.

Dr Sophia Tsoka at sophia.tsoka@kcl.ac.uk.

If you use this tool, please cite: L. Yang, C. Ainali, S. Tsoka, L.G. Papageorgiou, Pathway activity inference for multi-class disease classification through mathematical programming optimisation framework, *BMC Bioinformatics*, under review, 2014.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Licensing and software requirements</b>	<b>2</b>
<b>3</b>	<b>Using DIGS</b>	<b>2</b>
3.1	Prepare input files . . . . .	2
3.2	Specify parameters . . . . .	3
3.3	Run DIGS . . . . .	3
3.4	Output files from DIGS . . . . .	3

## 1 Introduction

DIGS is a pathway-level computational tool for disease classification using microarray gene expression profile. For each pathway specific gene expression profile, DIGS derives a new composite feature, called pathway activity, which summarises the expression patterns of its constituent genes. Pathway activity is constructed as a weighted linear summation of the expression profiles of its constituent genes, with the gene weights being optimised by DIGS so that the resulting pathway activity can optimally distinguish samples from different phenotypes. In DIGS, the maximum number of genes having non-zero weights can be controlled explicitly by user. The resulting activity vectors from all pathways are then assembled to form a pathway activity profile, on where a classifier can be trained to

predict phenotype of new samples. DIGS is applicable to both two-phenotype and multi-phenotype disease classification problems. The following journal publication, entitled "Pathway Activity Inference for Multiclass Disease Classification through Mathematical Programming Optimisation Framework", by Lingjian Yang, Chrysanthi Ainali, Sophia Tsoka, Lazaros G. Papageorgiou, offers more details on the methodology of DIGS.

## 2 Licensing and software requirements

In order to run DIGS, both third party softwares GAMS ([www.gams.com](http://www.gams.com)) and TORQUE (<http://www.adaptivecomputing.com/products/open-source/torque/>), together with their licences are required. DIGS makes calls to GAMS, which solves the underlying optimisation model of DIGS to optimise the gene weights for each pathway. One of the mixed integer linear programming solvers, either CPLEX or GUROBI, must be available. TORQUE is used to schedule the batch jobs of solving multiple pathways. DIGS should be ran in Linux command line.

## 3 Using DIGS

We describe, with some example files, how to prepare input files and run DIGS.

### 3.1 Prepare input files

The user must prepare two kinds of input files in order to run DIGS:

#### Pathway expression data files

For each pathway, the user must supply the pathway-specific gene expression profile in the **comma-delimited csv** format. The first row contains the names of the constituent genes, which should be **single or double quoted** to avoid clash with reserved words of GAMS. Each following row corresponds to a sample and its expression data. Sample names should be **single or double quoted** to avoid clash with reserved words of GAMS. Each sample must begin on a new line. Missing values are **not** permitted. Please refer to `KEGG_ACUTE_MYELOID_LEUKEMIA.csv`, `KEGG_NOTCH_SIGNALING.csv` and `KEGG_VIRAL_MYOCARDITIS.csv` files as illustrative examples.

### Class label file

The class labels (phenotypes or disease outcomes) of samples should be stored in a file, named **class\_label.txt**. Each row contains the name of the sample, followed by space and then the class label of the sample, represented by an **integer number**. **Class labels take integer numbers between 1 and the number of classes in the dataset.** Please refer to class\_label.txt file provided in the package as an illustrative example.

## 3.2 Specify parameters

All the user-specific parameters can be set in the provided template **input\_parameters.txt**.

## 3.3 Run DIGS

After preparing the pathway-specific gene expression data, class\_label.txt and input\_parameters.txt, the user need to put them and the other two files DIGS.g00 and run\_DIGS.txt in the same working directory. Give the access permission to all the above files:

```
chmod +x *
```

And then run the run\_DIGS.txt file:

```
./run_DIGS.txt
```

## 3.4 Output files from DIGS

After executing the run\_DIGS.txt file, DIGS will create for each pathway a new sub-directory containing all the relative output files from DIGS, including:

### Pathway activity files

For each pathway, DIGS infers pathway activity vector for each training-testing run. The constructed pathway activities are stored in a number of comma-delimited files with the names of **num\*\_pa.csv**, where \* corresponds to the actual order of training-testing run.

Parameter	Description	Recommended values
work_path	Set the full path of the directory where the following files are placed: pathway expression data in csv format input_parameter.txt class_label.txt DIGS.g00 run_DIGS.txt.	
scenario_setting	Set a testing scheme: 1: repeated random sampling 2: leave-one-out cross validation 3: full training	
num_repetitions_scenario1	If testing scenario 1: repeated random sampling is chosen, the user need to set also how many runs this is done. For the other two scenarios, the values will be ignored and set to 1.	50
percentage_training_samples_scenario1	If testing scenario 1 is chosen, the user need to set also the percentage of samples randomly selected as training samples. For the other two scenarios, the values will be ignored.	0.7
num_active_genes	Set the maximum number of constituent genes having non-zero weights when deriving pathway activity for each pathway.	10
computational_time_limit	Set the maximum computational time in s allowed for each pathway activity inference using DIGS.	200
solver_name	Choose a mixed-integer optimisation solver to solve the underlying optimisation model of DIGS	Cplex or Gurobi
thread_number	Set the number of parallel threads allowed for the solver	
optimality_gap_value	Set the relative optimality criterion. This sets a relative termination tolerance when solving mixed-integer optimisation models.	0
call_GAMS	Specify here the command used in Linux system to call GAMS. Default command is: gams.	

In each of those files, each row starts with the sample name, following by its pathway activity score and the class label of this sample (as given by the user). The num\*\_pa.csv file is arranged in such a way that the samples on the top are training samples, while the samples at the bottom are testing samples. **Testing samples are strictly blind to the DIGS model when inferring pathway activity, and their pathway activity scores are computed with the pathway activity formula derived from training samples only.**

GAMS uses a deterministic algorithm for generating random numbers to ensure results are reproducible. In our case it means for the same order of training-testing run, the set

of training samples is identical for all pathways. The user can further merge the pathway activity files (num\*\_pa.csv) from all pathways to form a final pathway activity profile.

### **Gene weights file**

DIGS also outputs a file, named **gene\_weights.csv** storing the gene weights for each training-testing run. It is important to note that DIGS performs a Z-score transformation for each gene before deriving gene weights for different training-testing runs. To recover the pathway activity scores, the user would need to Z-transform the expression data before multiplying the gene weights.

### **Number of misclassified samples file**

Recall that, DIGS is a supervised method that aims to build pathway activity as a feature which optimally distinguishes samples of different classes. DIGS will generate a file, named **obj\_information.txt**, which tracks how many samples are misclassified in the training phase of each training-testing run. This can give a rough idea of how good the constructed pathway activities are, with the lower the number of misclassified samples, the better the discriminative power.