An Introduction to *AGDEX*

Stan Pounds, Cuilan Lani Gao

Mon Apr 27 20:37:14 2020

1 Introduction

A challenging problem in contemporary genomics research is how to integrate and compare gene expression data from studies that utilize different microarray platforms or even different species (e.g. a study of a human disease and a study of an animal model of that disease). We have developed the agreement of differential expression (AGDEX) procedure to integrate differential expression analysis results across two experiments that may utilize different platforms or even different species. AGDEX is able to combine transcriptome information across two experiments that compare expression across two biological conditions. AGDEX was initially used in the study of the pediatric brain tumor ependymoma (Johnson et al., Nature 2010) to characterize the transcriptional similarity of a mouse model to one subtype of human ependymoma.

The AGDEX procedure performs a rigorous differential expression analysis for each two-group comparison and formally evaluates the agreement of differential expression analysis results across the entire transcriptome. Optionally, users may use AGDEX to identify differentially expressed gene-sets for each comparison and evaluate agreement of differential expression analysis results within gene-sets.

In total, the AGDEX procedure performs the following statistical analyses:

1. identify genes that are differentially expressed in each experiment;
2. identify gene-sets that are differentially expressed in each experiment;
3. integrate results across experiments to identify differentially expressed genes;
4. integrate results across experiments to identify differentially expressed gene-sets;
5. characterize and determine the statistical significance of similarities of differential and expression profiles across the two experiments for the entire transcriptome and for specific gene-sets.

The AGDEX method is described in greater detail in the supplementary materials of Johnson et al. (2010) and Gibson et al. (2010).

2 Pre-requisite Packages

The AGDEX package depends on the *Biobase* and *GSEABase* packages. Users must know how to store the expression data as an *ExpressionSet* object defined by the *Biobase* package. To perform gene-set analyses, users must represent gene-set data as an *GeneSetCollection* object defined by the *GSEABase* package.

3 Data Requirements

Data must be prepared and stored in a specific format for AGDEX analysis. First, data from each experiment must be stored as an *ExpressionSet* object defined by the *Biobase* package. Secondly, the data from each experiment must be linked with a definition of the contrast (such as "tumor - control") for the differential expression analysis. Optionally, each experiment may have gene-set definitions represented as a *GeneSetCollection* object defined by the *GSEABase* package. These information provide all the details necessary to perform differential expression analysis of the data from each experiment. Finally, a data-set that matches the probe-set identifiers from the two experiments is necessary to integrate results across the two experiments and evaluate the agreement of differential expression results across the two experiments.

AGDEX requires that the information needed to perform differential expression analysis of one experiment be provided in the form of a *dex.set* list object. The expression and phenotype data are stored as an *ExpressionSet* in a component named *Eset.data*. Recall that an *ExpressionSet* stores the expression data as a samples-by-genes matrix in the component *exprs* and the phenotype data as a *data.frame* in the component *pData*. The expression data should be normalized log-intensity values. The phenotype data must include one column with group labels to be used for the two-group differential expression analysis comparison. The *comp.var* component of the *dex.set* list object gives the name or numeric index of the column of the phenotype data with those group labels. The *comp.def* component of the *dex.set* list object is a string that defines the contrast for the two-group comparison. For example, the *comp.def* component may contain the string "tumor-control" to indicate that the analysis will compare the expression of those samples with the label "tumor" to that of those samples with the label "control". Optionally, the *dex.set* object may include a *GeneSetCollection* object (as defined by the package *GSEABase*) in the *gset.collection* component. In this way, the *dex.set* object contains the data for and the definition of a two-group differential expression analysis. The data and definition for each differential expression analysis must be contained in a *dex.set* object.

To perform the cross-experiment integration and evaluate the cross-experiment agreement, AGDEX requires information to match the probe-set identifiers of the first differential expression analysis to those of the second differential expression analysis. This information is provided in the form of a *map.data* list object. The *probe.map* component of the *map.data* object is a *data.frame* that defines how probe-set identifiers are matched across experiments. As such,

map.data must include a column with probe-set identifiers from experiment "A" and a column with probe-set identifiers from experiment "B". The components *map.Aprobe.col* and *map.Bprobe.col* give the name or numeric index of the columns of the *probe.map* component with the probe-set identifiers from experiments "A" and "B", respectively.

Finally, the user must specify how many permutations must be performed. AGDEX allows users to utilize an adaptive permutation testing (APT) strategy to reduce computing time for gene-set analyses. APT performs permutations until obtaining *min.perms* permutation-statistics with absolute value greater than that of the observed test-statistic or until performing a maximum *max.nperms* permutations. Pounds et al. (2011) give a more detailed description of APT.

4 Example

This example illustrates how users may perform an AGDEX analysis.

4.1 Prepare the Expression Data as *ExpressionSet* Object

First, users must prepare the *ExpressionSet* for each experiment. The *human.data* and *mouse.data ExpressionSet* objects are included in the AGDEX package.

```
> library(AGDEX)
> data(human.data)              # Load the human.data ExpressionSet object
> head(exprs(human.data)[,1:5]) # Preview the human expression data

              ept002   ept011   ept064   ept065   ept066
200050_at   8.555086 7.804986 7.626863 8.303084 8.110728
200075_s_at 7.808323 7.835382 8.520249 7.604795 8.111148
200642_at   9.656154 9.445705 9.687636 9.889678 9.451709
200823_x_at 8.990840 9.568204 9.085321 8.903503 9.517546
200878_at   7.759230 8.606851 8.008266 8.625222 8.294400
200879_s_at 4.518522 5.392718 4.242765 4.732684 5.220356

> head(pData(human.data))       # Preview the human phenotype data

           id                grp
ept002 ept002 other.human.tumors
ept011 ept011 other.human.tumors
ept064 ept064 other.human.tumors
ept065 ept065 other.human.tumors
ept066 ept066 other.human.tumors
ept067 ept067 other.human.tumors

> table(pData(human.data)$grp)  # See number in each group
```

```
 human.tumor.typeD other.human.tumors
                 9                 74

> all(rownames(pData(human.data))==colnames(exprs(human.data)))  # Check that expression dat

[1] TRUE

> data(gset.data)  # A GeneSetCollection for human.data
> # Now the same for the mouse.data
> data(mouse.data)
> head(exprs(mouse.data)[,1:5])

               hmp002   hmp003   hmp004   hmp007   hmp008
1415674_a_at 7.674153 7.677817 7.762724 7.759486 7.709533
1415686_at   7.761532 7.425119 7.596593 7.761873 7.760978
1415691_at   7.710116 7.536471 7.477208 7.611941 7.682068
1415860_at   9.724080 9.771680 9.787628 9.810550 9.600272
1415888_at   8.169280 8.323366 8.153177 8.317424 8.021749
1415904_at   7.385975 6.666830 6.798164 7.253046 7.440029

> head(pData(mouse.data))

            id           grp
hmp002 hmp002 mouse.control
hmp003 hmp003 mouse.control
hmp004 hmp004 mouse.control
hmp007 hmp007 mouse.control
hmp008 hmp008 mouse.control
hmp009 hmp009 mouse.control

> table(pData(mouse.data)$grp)

mouse.control   mouse.tumor
          179            13

> all(colnames(exprs(mouse.data))==rownames(pData(mouse.data)))

[1] TRUE

>
```

4.2 Form a *dex.set* Object for Each Experiment

Second, for each experiment, information defining the differential expression analysis must be combined with the *ExpressionSet* data and stored in a *dex.set* object by using *make.dex.set.object*, as shown below.

```
> # Create dex.set for human.comparison
> dex.set.human <- make.dex.set.object(Eset.data= human.data,
+                                      comp.var=2,
+                                      comp.def="human.tumor.typeD-other.human.tumors",
+                                      gset.collection=gset.data)
> dex.set.mouse <- make.dex.set.object(mouse.data,
+                                      comp.var=2,
+                                      comp.def="mouse.tumor-mouse.control",
+                                      gset.collection=NULL)
>
```

In the first statement above, *Eset.data=human.data* indicates that the *ExpressionSet* object *human.data* contains the expression and phenotype data, *comp.var=2* indicates that the second column of the phenotype data (e.g. *pData(human.data)[,2]*) has the group labels for the differential expression analysis comparison, *comp.def="human.tumor.typeD-other.human.tumors"* indicates that the comparison will be computed as "human.tumor.typeD" minus "other.human.tumors", and *gset.collection=gset.data* indicates that the *GeneSetcollection* object *gset.data* defines gene-sets for the differential expression analysis. The second statement above performs an analogous operation for the mouse data, except that it does not provide gene-set definitions for gene-set analyses.

4.3 Prepare the *map.data* Object that Defines How Probe-Sets are Matched Across Experiments

The *map.data* list object includes a component *probe.map* with a *data.frame* that defines how probe-sets are matched across experiments and components *map.Aprobe.col* and *map.Bprobe.col* that give the name or numeric index of the columns with the probe-set identifiers from experiments "A" and "B", respectively. The code segment below illustrates the structure of the *map.data* object.

```
> data(map.data)
> names(map.data)

[1] "probe.map"      "map.Aprobe.col" "map.Bprobe.col"

> head(map.data$probe.map)

     human_probe  mouse_probe
7969   200050_at   1422135_at
8025 200075_s_at   1416395_at
8278   200642_at 1459976_s_at
8279   200642_at   1451124_at
8280   200642_at   1435304_at
8281   200642_at 1447761_x_at

> map.data$map.Aprobe.col
```

```
[1] 1
```

```
> map.data$map.Bprobe.col
```

```
[1] 2
```

4.4 Perform the AGDEX Analysis

Now that the *dex.set* objects for each experiment and the *map.data* object have been prepared, the AGDEX analysis may be performed by a simple call to the function *agdex*, as shown below.

```
> agdex.res<-agdex(dex.setA=dex.set.human,
+                  dex.setB=dex.set.mouse,
+                  map.data=map.data,
+                  min.nperms=5,
+                  max.nperms=10)
```

```
Preparing differential expression data sets (dex.setA and dex.setB): Mon Apr 27 20:37:19 202
Preparing data that maps probe set IDs across experiments: Mon Apr 27 20:37:19 2020
Computing statistics for observed data: Mon Apr 27 20:37:19 2020
Mapping gene-sets across experiments: Mon Apr 27 20:37:19 2020
Computing time for observed statistics (in seconds):
0.003 0 0.003 0 0
Permuting experiment A data 10 times: Mon Apr 27 20:37:19 2020
Computing time for permutation analysis of data set A (in seconds):
0.016 0 0.016 0 0
Permuting experiment B data 10 times: Mon Apr 27 20:37:19 2020
Computing time for permutation analysis of data set B (in seconds):
0.019 0 0.019 0 0
Packaging Result Object: Mon Apr 27 20:37:19 2020
Done: Mon Apr 27 20:37:19 2020
```

This statement performs the AGDEX analysis with the human data considered as experiment "A" and the mouse data considered as experiment "B". Note that it is important that the call to the function *agdex* and the *map.data* object label the experiments in the same way. Clearly, one usually will set larger values of *min.nperms* and *max.nperms* in most applications. The classical permutation test can be performed by setting *min.nperms* = *max.nperms*. See Pounds et al. (2011) for more details on how to set *min.nperms* and *max.nperms*. The AGDEX procedure will perform exact tests if the total number of permutations is less than the expected number of permutations under the null hypothesis of exchangeability.

4.5 Explore AGDEX Results

The results of the AGDEX analysis are stored in a list with multiple components. More details are available from *help(agdex.result)*. As shown below, several

components of the result object echo the input for the group labels and definition of the contrast for each differential expression analysis.

```
> names(agdex.res)

 [1] "dex.compA"          "dex.compB"
 [3] "gwide.agdex.result" "gset.result"
 [5] "meta.dex.res"       "dex.resA"
 [7] "dex.resB"           "dex.asgnA"
 [9] "dex.asgnB"          "gset.listA"
[11] "gset.listB"         "gset.list.agdex"

> agdex.res$dex.compA                # echoes comp.def of dex.setA

[1] "human.tumor.typeD-other.human.tumors"

> agdex.res$dex.compB                # echoes comp.def of dex.setB

[1] "mouse.tumor-mouse.control"

> head(agdex.res$dex.asgnA)          # echoes group-labels from dex.setA

      id            grp.lbl
1 ept002 other.human.tumors
2 ept011 other.human.tumors
3 ept064 other.human.tumors
4 ept065 other.human.tumors
5 ept066 other.human.tumors
6 ept067 other.human.tumors

> head(agdex.res$dex.asgnB)          # echoes group-labels from dex.setB

      id       grp.lbl
1 hmp002 mouse.control
2 hmp003 mouse.control
3 hmp004 mouse.control
4 hmp007 mouse.control
5 hmp008 mouse.control
6 hmp009 mouse.control
```

The result object also contains the probe-set level differential expression analysis results for each experiment. These components give the difference of means and p-values for each probe-set in their respective experiments.

```
> head(agdex.res$dex.resA) # Human results, difference of means and p-values

               probe.id       dstat dpval
200050_at     200050_at -0.28879244   0.0
200075_s_at 200075_s_at  0.03932749   0.8
```

```
200642_at     200642_at  0.14981534   0.3
200823_x_at 200823_x_at -0.39166373   0.0
200878_at     200878_at  0.33152395   0.2
200879_s_at 200879_s_at  0.21099839   0.2
```

```
> head(agdex.res$dex.resB) # Mouse Results, difference of means and p-values
```

```
                 probe.id       dstat dpval
1415674_a_at 1415674_a_at -0.45082351   0.0
1415686_at     1415686_at  0.02563087   0.4
1415691_at     1415691_at  0.90984699   0.0
1415860_at     1415860_at -1.34567368   0.0
1415888_at     1415888_at -0.89954516   0.0
1415904_at     1415904_at  1.07934126   0.0
```

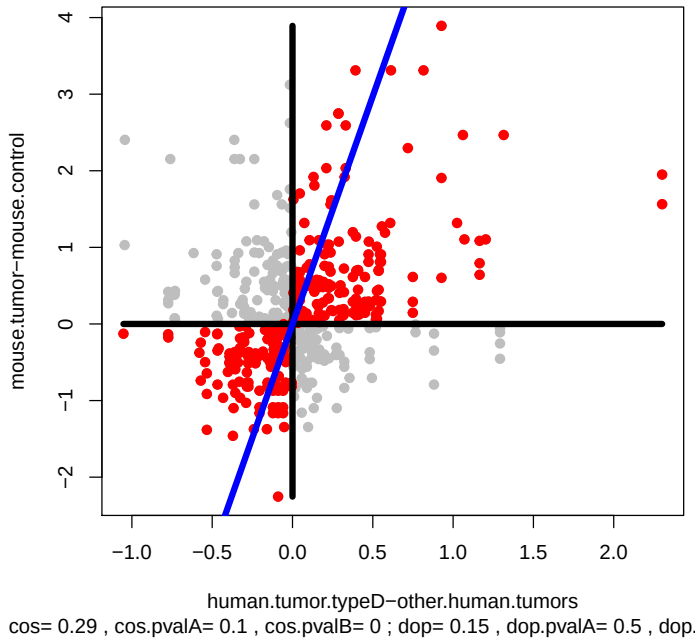
The *meta.dex.res* component contains these results and the meta-analysis z-statistic and p-value for the matched probe-set pairs.

```
> head(agdex.res$meta.dex.res)
```

```
        probeB      probeA A.index B.index      dstatA
1 1415674_a_at   217958_at     221       1  0.05025209
2 1415674_a_at 217959_s_at     222       1 -0.01934544
3   1415686_at 200927_s_at       8       2 -0.05587315
4   1415686_at 211503_s_at     162       2  0.04761622
5   1415686_at 200928_s_at       9       2  0.16640311
6   1415691_at   202514_at      37       3 -0.09494414
  dpvalA      dstatB dpvalB meta.zstat meta.pval
1    0.5 -0.45082351    0.0 -0.9498341 0.3421966
2    1.0 -0.45082351    0.0 -1.4006035 0.1613327
3    0.6  0.02563087    0.4  0.2081991 0.8350735
4    0.7  0.02563087    0.4  0.8186493 0.4129865
5    0.1  0.02563087    0.4  1.5958498 0.1105223
6    0.6  0.90984699    0.0  1.0490296 0.2941645
```

The function *agdex.scatterplot* produces a scatterplot of the difference-of-means statistics for probe-set pairs.

```
>  agdex.scatterplot(agdex.res, gset.id=NULL)
```



The results of the genome-wide AGDEX analysis are available in the *gwide.agdex.result* component.

```
> agdex.res$gwide.agdex.res
```

```
          stat.name stat.value A.pval B.pval A.nperms
cos.stat        cos  0.2926636    0.1      0       10
prop.stat       dop  0.1510204    0.5      0       10
          A.exact B.nperms B.exact
cos.stat    FALSE       10   FALSE
prop.stat   FALSE       10   FALSE
```

The *gwide.agdex.result* component is a *data.frame* with the cosine and difference-of-proportions statistics and their p-values by permutation of group labels from experiments "A" and "B". It also indicates the number of permutations performed for each experiment and whether or not the test is exact (i.e., based on all possible permutations).

The results of gene-set differential expression analysis for each experiment, cross-experiment meta-analysis, and cross-experiment agreement are available in the *gset.res* component.

```
> head(agdex.res$gset.res)
```

```
  gset.source
1           A
```

```
2           A
3           A
4           A
5           A
6           A
                                                                  gset.name
1                                                        PHOTOTRANSDUCTION
2                                    CDC42_PROTEIN_SIGNAL_TRANSDUCTION
3                                            GLYCOSAMINOGLYCAN_BINDING
4                                               DNA_CATABOLIC_PROCESS
5 RNA_POLYMERASE_II_TRANSCRIPTION_FACTOR_ACTIVITY__ENHANCER_BINDING
6                          NLS_BEARING_SUBSTRATE_IMPORT_INTO_NUCLEUS
  A.gset.dstat A.gset.dpval A.gset.cos.stat A.gset.cos.pval
1    0.2502446          0.3       0.3535225             0.3
2    0.4004950          0.0       0.5855269             0.2
3    0.3268436          0.0       0.1200461             0.9
4    0.2779532          0.0       0.4617482             0.3
5    0.3332079          0.0       0.3751870             0.3
6    0.2376916          0.0       0.2526408             0.5
  A.gset.dop.stat A.gset.dop.pval A.gset.nperms
1      0.41666667             0.3            10
2      0.09677419             1.0            10
3      0.25000000             0.5            10
4      0.20754717             0.3            10
5      0.22727273             0.6            10
6      0.04166667             1.0            10
  B.gset.dstat B.gset.dpval B.gset.cos.stat B.gset.cos.pval
1    0.8264484            0       0.3535225             0.6
2    0.7193008            0       0.5855269             0.0
3    0.6583517            0       0.1200461             0.4
4    0.5866383            0       0.4617482             0.1
5    0.6561492            0       0.3751870             0.1
6    0.6050002            0       0.2526408             0.1
  B.gset.dop.stat B.gset.dop.pval B.gset.nperms
1      0.41666667             0.1            10
2      0.09677419             0.2            10
3      0.25000000             0.4            10
4      0.20754717             0.4            10
5      0.22727273             0.5            10
6      0.04166667             0.9            10
  meta.enrich.zstat meta.enrich.pval
1          1.529762      0.063037774
2          2.390900      0.008403564
3          2.390900      0.008403564
4          2.390900      0.008403564
5          2.390900      0.008403564
```

```
6          2.390900       0.008403564
```

If there is interest in seeing probe-set level details for the gene-set analysis results, the function *get.gset.result.details* may be used. The function *get.gset.result.details* may be used to obtain details for a specific gene-set of particular interest or to obtain details for those with p-values less than a specific threshold.

```
> gset.res.stats<-get.gset.result.details(agdex.res, gset.ids = NULL, alpha=0.01)
> names(gset.res.stats)

[1] "enrichA.details" "enrichB.details" "agdex.details"

> head(gset.res.stats$enrichA.details)

     probe.id       dstat dpval
1 202844_s_at  0.31797091   0.0
2 202845_s_at  0.39569681   0.0
3 203381_s_at  1.06113090   0.0
4 203382_s_at  1.31501397   0.0
5 203504_s_at -0.06661015   0.9
6   203505_at  0.23706128   0.3
                         gset.name meta.enrich.zstat
1 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
2 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
3 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
4 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
5 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
6 CDC42_PROTEIN_SIGNAL_TRANSDUCTION            2.3909
  meta.enrich.pval
1      0.008403564
2      0.008403564
3      0.008403564
4      0.008403564
5      0.008403564
6      0.008403564

> head(gset.res.stats$agdex.details)

  gset.source                         gset.name
1           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
2           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
3           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
4           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
5           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
6           A CDC42_PROTEIN_SIGNAL_TRANSDUCTION
  A.gset.cos.stat A.gset.cos.pval A.gset.dop.stat
1       0.5855269             0.2      0.09677419
```

```
2       0.5855269             0.2      0.09677419
3       0.5855269             0.2      0.09677419
4       0.5855269             0.2      0.09677419
5       0.5855269             0.2      0.09677419
6       0.5855269             0.2      0.09677419
  A.gset.dop.pval B.gset.cos.stat B.gset.cos.pval
1               1       0.5855269               0
2               1       0.5855269               0
3               1       0.5855269               0
4               1       0.5855269               0
5               1       0.5855269               0
6               1       0.5855269               0
  B.gset.dop.stat B.gset.dop.pval       probeB      probeA
1      0.09677419             0.2   1416865_at   204819_at
2      0.09677419             0.2   1417248_at 202845_s_at
3      0.09677419             0.2   1417248_at 202844_s_at
4      0.09677419             0.2   1418278_at 205820_s_at
5      0.09677419             0.2 1419232_a_at 204450_x_at
6      0.09677419             0.2 1419232_a_at 217073_x_at
  A.index B.index      dstatA dpvalA      dstatB dpvalB
1      74      14  0.05772788    0.9 -1.15817398    0.0
2      44      17  0.39569681    0.0  0.19384416    0.0
3      43      17  0.31797091    0.0  0.19384416    0.0
4      80      29  0.39770711    0.1  0.69530585    0.0
5      70      45  0.51961004    0.1 -0.04381276    0.6
6     215      45 -0.10466286    0.8 -0.04381276    0.6
  meta.zstat   meta.pval
1 -1.3159995 0.188174224
2  2.8012069 0.005091187
3  2.8012069 0.005091187
4  2.4366802 0.014822782
5  0.6845029 0.493657687
6 -0.5220152 0.601659753

> dna.cat.process.gset.res<-get.gset.result.details(agdex.res, gset.ids="DNA_CATABOLIC_PROCE
> head(dna.cat.process.gset.res$agdex.details)

  gset.source             gset.name A.gset.cos.stat
1           A DNA_CATABOLIC_PROCESS       0.4617482
2           A DNA_CATABOLIC_PROCESS       0.4617482
3           A DNA_CATABOLIC_PROCESS       0.4617482
4           A DNA_CATABOLIC_PROCESS       0.4617482
5           A DNA_CATABOLIC_PROCESS       0.4617482
6           A DNA_CATABOLIC_PROCESS       0.4617482
  A.gset.cos.pval A.gset.dop.stat A.gset.dop.pval
1             0.3       0.2075472             0.3
```

```
2             0.3       0.2075472             0.3
3             0.3       0.2075472             0.3
4             0.3       0.2075472             0.3
5             0.3       0.2075472             0.3
6             0.3       0.2075472             0.3
  B.gset.cos.stat B.gset.cos.pval B.gset.dop.stat
1       0.4617482             0.1       0.2075472
2       0.4617482             0.1       0.2075472
3       0.4617482             0.1       0.2075472
4       0.4617482             0.1       0.2075472
5       0.4617482             0.1       0.2075472
6       0.4617482             0.1       0.2075472
  B.gset.dop.pval       probeB      probeA A.index B.index
1             0.4   1416837_at 208478_s_at     125      13
2             0.4   1416837_at 211833_s_at     171      13
3             0.4   1417328_at 203720_s_at      61      18
4             0.4   1417328_at   203719_at      60      18
5             0.4   1417956_at   221295_at     235      21
6             0.4 1417983_a_at   209096_at     134      24
       dstatA dpvalA     dstatB dpvalB meta.zstat
1 -0.15938070    0.4 -1.3733719      0 -1.9603765
2 -0.24029731    0.2 -1.3733719      0 -2.2348205
3 -0.12147246    0.3 -0.4491537      0 -2.0846738
4 -0.15520865    0.0 -0.4491537      0 -2.8012069
5  0.57610397    0.1  1.1911363      0  2.4366802
6  0.03219396    0.3 -0.3694369      0 -0.7165331
    meta.pval
1 0.049951803
2 0.025429133
3 0.037098920
4 0.005091187
5 0.014822782
6 0.473662236
```

4.6 Store and Report AGDEX Results

User may also use the *write.agdex.result* command to save their results in tab-delimited text format for viewing in Microsoft Excel. The command *read.agdex.result* may be used to read the output of *write.agdex.result* back into R.

Users may also wish to annotate the genes in each of the above result. Bioconductor annotation packages and annotation databases provide these capabilities for a wide range of gene expression microarrays.

5 References

1. Pounds, S. et al. A Procedure to statistically evaluate agreement of differential expression for cross-species genomics. *Bioinformatics*, doi: 10.1093/bioinformatics/btr362(2011).

2. Johnson, R. et al. Cross-species genomics matches driver mutations and cell compartments to model ependymoma. *Nature*, 466, 632-6 (2010).

3. Gibson, P. et al. Subtypes of medulloblastoma have distinct developmental origins. *Nature*, 468, 1095-99 (2010).

4. Pounds, S., et al. Integrated Analysis of Pharmacokinetic, Clinical, and SNP Microarray Data using Projection onto the Most Interesting Statistical Evidence with Adaptive Permutation Testing. *International Journal of Data Mining and Bioinformatics, 5:143-157 (2011).*