12/21/09

ATHENA Manual

Table of Contents

Introduction

ATHENA applies grammatical evolution to optimize neural networks for detection and modeling of gene-gene interactions. It replicates the features of GENN and incorporates new features. It will be extended in the future to allow for additional search algorithms and different model representations.

Example

ATHENA takes one command line argument, a configuration file specifying all the parameters for the run.

```
athena example.config
```

Input Files

Configuration file

ATHENA takes the name of a configuration file as its single command-line argument. The configuration file should list all the parameters for controlling the analysis. It should be in the format of keyword <whitespace> value. Each keyword should be on its own line. Comments can begin with a '#' and will be ignored by the program. Any parameter without a default value must be specified in the configuration file.

General parameters

General parameters affect the program as a whole and usually specify parameters such as input files and the start of algorithm specific parameters.

Parameter	**Default**	**Description**
DATASET	None	Dataset to analyze
CV	5	Number of cross-validation intervals to split data
RANDSEED	1	Random seed used in dividing data for cross-validation intervals
IDINCLUDED	False	If 'True', then the first column in the dataset file will be an ID number for the set.
CONTINFILE	None	Contains covariate data. Must be in same order as the main dataset file.
OUT	athena	Value will be the base name for all

		the output files generated by ATHENA. Different extensions will be applied to the basename.
ALGORITHM	None	Specifies name of algorithm to use in program. All parameters for a specified algorithm should follow below this keyword with the last one being followed by the END keyword.
END		Indicates that the parameters for an algorithm have been completed. An END always needs to follow an ALOGORITHM keyword.
MISSINGVALUE	-1	Missing value in genotype data file.
CONTINMISS	-9999	Missing value in continuous data file.
INPUT	TEXT	Specifies type of format of genotype data file. Default is TEXT, which corresponds to the format used by MDR.
MAPFILE	None	File contains locus names in same order as the genotype data file. If not specified, the genotypes will be numbered from 1 and reported that way.
DUMMYENCODE	False	If True, all genotype data will be dummy encoded using method specified by Jurg Ott.
NUMSTEPS	2	Number of times the best models will be exchanged among multiple populations in parallel version of ATHENA
WRITECV	False	If True, the individuals and genotypes used for training in each cross-validation are output to files named cv.1.txt, cv.2.txt, etc.
STATUSADJUST	None	If used with data that has a continuous value as status, will specify the type of transformation to do on the values. NORMMAX scales all the status values from 0 to 1 by dividing all values by the maximum status value.
INDOUTPUT	False	When set to True, outputs the scores for every individual evaluated by the

		best model in each cross-validation.
ALLNODESBEST	False	When set to True, outputs best model for each node at end of each cross-validation run.
TRAINFILE	None	Training genotype data file. . Can be used instead of using DATASET and CV if user already has split data.
TESTFILE	None	Testing genotype data file. Can be used instead of using DATASET and CV if user already has split data.
BIOFILTERFILE	None	Lists models that can be used to alter behavior of algorithms in program.
SUMMARYONLY	False	When set to true, ATHENA will not produce the .dot or .best output files.
LOG	NONE	Three options for controlling log output: NONE – no log files generated SUMMARY – only summary file DETAILED – summary file and files showing every fitness (.fitness.log) and number of snps in every model (.snpsize.log)
STATUSMISSINGVALUE	-1	Specifies value in file that identifies individuals whose status is missing or unknown. These individuals will be left out of the analysis.
BIOGENEFILE	None	File produced by biofilter software. Lists genes and SNPs that occur within the gene.
BIOARCHIVEFILE	None	File produced by biofilter software. Used with a biogenefile. Lists gene-gene combinations for testing.

GENN parameters

GENN algorithm parameters only affect the parameters of the algorithm specified when running the program.

Parameter	**Default**	**Description**
GRAMMARFILE	None	Grammar file for use with grammatical evolution
POPSIZE	100	Number of models in each population
PROBCROSS	0.9	Probability of a crossover for each

		mating in a generation
BIOFILTERFRACT	0.0	Fraction of initial population that will be initialized using models provided by a bio filter file. If there aren't enough models in the file, the extra models will be initialized with either sensible initialization or random initialization based on that parameter.
MINSIZE	50	Minimum size when random initialization
MAXSIZE	200	Maximum size when random initialization
TAILRATIO	0.0	Specifies size of tail percentage for initialization of solutions in population
GROWRATE	0.5	Specifies fraction of the population that will be initialized using the Grow method instead of the full method
SENSIBLEINIT	False	If 'True', the solutions will be initialized using sensible initialization. Otherwise, random initialization is used.
PROBMUT	0.01	Mutation rate per codon in solution genome
GENSPERSTEP	100	Number of generations performed before each exchange of best solutions is done
CALCTYPE	BALANCEDACC	Type of fitness calculation performed. BALANCEDACC is for simple binary status data. RSQUARED is for continuous status.
EFFECTIVEXO	False	When 'True', crossovers occur only within the effective coding region of the genome.
INCLUDEALLSNPS	False	When 'True', all variables in the dataset will be used regardless of the grammar file.
REQUIREALLVARS	False	Only solutions that include all variables in the dataset are evaluated for fitness.
REQUIREALLONCE	False	Only solutions that include each variable once are evaluated for fitness.
TAILSIZE	0	Size of tail added on to end of codons

		in initialization.
MAXDEPTH	10	Maximum depth of solution in tree form.
NUMGENSRESTRICTVARS	0	Number of generations that the grammar used by the algorithm is restricted to only variables (genotypes and covariates) that are part of the initialized networks.
RESETVARSMIGRATION	False	This parameter is used in conjunction with having NUMGENSRESTRICTVARS set. After a migration, the population will use a new grammar. If this parameter is true, the new grammar will only include variables that are in the population after the migration. When set to false, any new variables that migrated in will be added but all older variables will be maintained whether or not they are in the current population.
BACKPROPSTART	-1	First generation to run back propagation on. If set to 0, will run backprop after initialization of population. If set to < 0, no backprop will occur (default).
BACKPROPFREQ	0	Specifies frequency of backprop during run. If set to zero, backprop will not repeat during the run after the generation specified by BACKPROPSTART.
BLOCKCROSSGENS	0	Number of generations that crossover will use the block crossover which matches compatible regions of the genomes and insures the crossover will not be destructive.
BIOMODELSELECTION	ROULETTE	Method for selecting the models from the bio filter file. Options are ORDERED (where the models are taken in order of implication index) and ROULETTE (where the models are weighted based on implication index and selected randomly.

Data file format

ATHENA accepts data in a simple format. Each line is a separate individual. The first column is the ID (if that option is on) or it is the status. After that information, each additional column contains the value at a locus for the genotype data. The continuous data file is similar except there is no status column.

Map file format

The Map file identifies the SNPs present in the genotype data file. Each line corresponds to a column in the genotype file. First column in the map file is chromosome number. Second is SNP ID (rs number) and third is the position in base pairs on the chromosome.

Algorithms

ATHENA currently implements the same algorithm as GENN. Grammatical evolution (GE) is an evolutionary algorithm that uses linear genomes and grammars to define the populations. In GE, each individual consists of a binary genome divided into codons. Mutation takes place on individual bits but crossover only takes place between the codons. Translating codons using the grammar produces an individual or phenotype. The resulting individual can then be tested for fitness in the population and the usual evolutionary operators can be carried out. By using a grammar to define the phenotype, GE separates the genotype from the phenotype and allows greater genetic diversity within the population than other evolutionary algorithms. In GENN, the grammar creates a neural network that accepts variables from the dataset.

The type of status in the dataset determines the fitness used in the algorithm. When the status is binary (affected or unaffected), the fitness of a network is determined by the balanced accuracy, [(sensitivity+specificity)/2]. When the status is a continuous variable, fitness for the network is the R-squared (coefficient of determination).

ATHENA can be run using a cross-validation framework. In that case the data are divided into a training set and a testing set for each cross-validation interval. For example, in 10-fold cross-validation the training set will be 9/10 of the data and the testing set will be 1/10. The training set is utilized to set the fitness of each solution in the population during the running of the algorithm. After the best neural network is produced, its predictive ability is evaluated by determining the score of the testing set.

Sample files

Input files

ATHENA utilizes a number of input files. Simple examples are displayed below.

Configuration file

```
# sample configuration for use with ATHENA
ALGORITHM GENN
MINSIZE 20
MAXSIZE 300
MAXDEPTH 8
SENSIBLEINIT TRUE
POPSIZE 100
PROBCROSS 0.9
PROBMUT 0.01
GRAMMARFILE add.gram
CALCTYPE BALANCEDACC
EFFECTIVEXO TRUE
GENSPERSTEP 10
INCLUDEALLSNPS TRUE
END GENN
# specify general parameters for run
DATASET 27.dat
IDINCLUDED FALSE
MISSINGVALUE -1
DUMMYENCODE TRUE
RANDSEED 7
OUT 27.40gen
CV 10
NUMSTEPS 2
WRITECV FALSE
```

Genotype data file (no ID)

```
0 2 1 2 2 1 1 2 2 2 2 1 1
0 1 2 0 1 0 2 1 1 2 1 2 1
0 0 1 2 1 0 1 2 0 1 2 2 1
0 1 1 2 0 1 1 2 1 2 0 1 1
0 2 2 2 1 1 0 2 1 1 2 0 1
1 2 1 2 1 2 1 1 2 1 2 0 1
1 2 1 0 1 1 1 1 2 2 1 0 0
1 1 2 1 2 1 2 1 2 0 0 2 1
1 1 2 1 2 0 2 1 1 1 2 1 2
1 2 2 0 0 1 1 1 0 0 2 1 1
```

Genotype data file (with ID)

```
1 0 1 1 1 1 1 1 1 2 0 2 0 1
2 0 2 2 2 1 2 1 1 0 1 1 1 1
3 0 2 1 0 2 1 1 1 2 2 1 1 1
4 0 1 1 1 2 2 1 1 2 1 1 1 2
5 0 2 1 1 2 1 1 1 1 1 2 0 2
6 1 2 2 1 1 1 0 2 1 1 2 1 1
7 1 1 1 2 2 1 2 2 1 1 2 1 2
8 1 1 1 1 1 1 1 1 1 2 2 1 2
9 1 1 2 1 0 2 1 1 2 0 1 1 0
10 1 2 1 2 1 2 1 1 1 2 2 1 1
```

Map file

```
1       rs1     10502
1       rs2     220020
1       rs3     303034
2       rs4     10201
3       rs5     3303049
```

Continuous data file (with ID)

```
1  22.5 114.8 0.5
2  11.8 122   0.7
3  17.3 119.5 0.56
4  15.8 120.3 0.72
5  19.2 118.2 0.88
6  9.5  98.8  0.77
7  14.8 112.4 0.35
8  11.8 119.9 0.78
9  14.8 125   0.25
10 15.5 104   0.33
```

Bio Filter Model File

ATHENA accepts a list of models that can be incorporated into the initialization of networks. The first two columns list the marker IDs. The last column is the implication index which is a score designating how many sources specify the model.

```
50 40 3
96 24 3
82 51 3
44 14 2
89 5 2
85 76 2
37 10 2
```

Output files

Summary file

ATHENA produces a summary file listing the variables from the best model and its scores for each cross validation interval in the analysis. The file has the extension .athena.sum

```
CV   Variables       Training  Testing
1    G3 G12 G3 G6    0.5675    0.5175
2    G1 G12 G12 G1   0.5575    0.4175
```

Best model file

The best model files display the actual network produced by ATHENA. It has the extension cv<#>.<# rank in CV>.best. For example, the best model from cross validation one has the extension .cv1.1.best.

```
CV: 1
Model Rank: 1
Training result: 0.5675
Testing result: 0.5175
Model:
PS( W(5.78,G6), W((3+72.97),G24), W(6.55,G6), W((9.2-
25.76),G12),4)

Grammar-compatible version:
PS ( W ( Concat ( 5 . 7 8 4 ) , G6 ) , W ( ( Concat ( 3 1 ) +
Concat ( 7 2 . 9 7 5 ) ) , G24 ) , W ( Concat ( 6 . 5 5 4 ) , G6
) , W ( ( Concat ( 9 . 2 3 ) - Concat ( 2 5 . 7 6 5 ) ) , G12 ) ,
4 )
```

Dot file

ATHENA produces dot-compatible files that can be converted into image files using the dot program from the Graphviz visualization project (http://www.graphviz.org/). The files have the extension .cv<#>.<# rank in CV>.dot. For example, the best model from cross validation one has the extension .cv1.1.dot.

```
digraph G{
        size="7.5,11.0";
        dir="none";
        rankdir="LR";
        orientation="landscape";
        PSUB1 [shape="doublecircle" style="bold" label="PSUB"];
        W1->PSUB1;
        W1 [shape="circle" style="bold" label="W"];
        const1->W1;
        const1 [shape="box" style="bold" label="5.78"];
        G61->W1;
        G61 [shape="box" style="filled" label="G6"];
        W2->PSUB1;
        W2 [shape="circle" style="bold" label="W"];
        Add1->W2;
        Add1 [shape="diamond" style="bold" label="+"];
        const2->Add1;
        const2 [shape="box" style="bold" label="3"];
        const3->Add1;
        const3 [shape="box" style="bold" label="72.97"];
        G241->W2;
        G241 [shape="box" style="filled" label="G24"];
        W3->PSUB1;
        W3 [shape="circle" style="bold" label="W"];
        const4->W3;
        const4 [shape="box" style="bold" label="6.55"];
        G62->W3;
        G62 [shape="box" style="filled" label="G6"];
        W4->PSUB1;
        W4 [shape="circle" style="bold" label="W"];
        Sub1->W4;
        Sub1 [shape="diamond" style="bold" label="-"];
        const5->Sub1;
        const5 [shape="box" style="bold" label="9.2"];
        const6->Sub1;
        const6 [shape="box" style="bold" label="25.76"];
        G121->W4;
        G121 [shape="box" style="filled" label="G12"];
}
```

Individual score files

ATHENA can produce optional files displaying the score that each individual receives when being processed by the evolved networks. The files have the extension <cv#>.<rank#>.ind_results.txt. For example, the individual evaluations for the bet model from the first cross validation will be in a file with the extension .1.1.ind_results.txt. When ID numbers are present in the data files they are identified by those numbers. Otherwise, the file identifies each individual by the line number in the original data file.

```
Individual 1796 score = 0.397232
Individual 88 score = 0.602768
Individual 202 score = 0.397232
Individual 174 score = 0.382958
Individual 1583 score = 0.485004
Individual 1641 score = 0.382958
Individual 1375 score = 0.485004
Individual 1323 score = 0.368883
Individual 1514 score = 0.382958
Individual 532 score = 0.5
Individual 194 score = 0.368883
Individual 297 score = 0.368883
```

Cross-validation files

ATHENA can produce optional files listing the individuals in each cross validation's training set.   The files are named cv.1.txt, cv.2.txt, etc. They contain the values of the genotypes that are used, so they are shown with dummy encoding if that option was set in the configuration file.

```
0 0 2 0 2 0 2 1 -1 0 2 0 2 0 2 0 2 1 -1 0 2 0 2 0 2
0 1 -1 1 -1 0 2 -1 -1 0 2 1 -1 0 2 -1 -1 1 -1 -1 -1 -1 -1 -1 -1
0 0 2 0 2 0 2 1 -1 1 -1 0 2 1 -1 0 2 0 2 -1 -1 -1 -1 0 2
0 0 2 -1 -1 0 2 1 -1 1 -1 0 2 0 2 1 -1 0 2 0 2 0 2 0 2
0 1 -1 1 -1 1 -1 0 2 1 -1 -1 -1 1 -1 0 2 0 2 1 -1 0 2 1 -1
1 1 -1 1 -1 0 2 0 2 0 2 1 -1 1 -1 0 2 0 2 0 2 -1 -1 0 2
1 1 -1 0 2 0 2 1 -1 1 -1 0 2 0 2 0 2 -1 -1 0 2 0 2 0 2
1 1 -1 0 2 1 -1 0 2 1 -1 1 -1 1 -1 0 2 -1 -1 1 -1 1 -1 0 2
1 0 2 1 -1 1 -1 0 2 1 -1 0 2 1 -1 1 -1 1 -1 0 2 1 -1 1 -1
1 0 2 1 -1 0 2 0 2 0 2 1 -1 1 -1 0 2 0 2 1 -1 0 2 1 -1
```