ARACNe-AP v1.4 usage guide



Contents

[Overview 2](#_Toc447182473)

[Input files needed to run ARACNe 2](#_Toc447182474)

[Steps required to run ARACNe 2](#_Toc447182475)

[Optional ways to run ARACNe 2](#_Toc447182476)

[Input file format 3](#_Toc447182477)

[Gene lists 3](#_Toc447182478)

[Dataset 3](#_Toc447182479)

[Parameters 4](#_Toc447182480)

[Examples 5](#_Toc447182481)

[Example 1 – calculate threshold with a fixed seed 5](#_Toc447182482)

[Example 2 – run ARACNe on a single bootstrap 5](#_Toc447182483)

[Example 3 – run 100 reproducible bootstraps 5](#_Toc447182484)

[Example 4 – consolidate bootstraps in the output folder 5](#_Toc447182485)

[Example 5 – run a single ARACNE with no bootstrap and no DPI 5](#_Toc447182486)

[Example 6 – consolidate bootstraps without Bonferroni correction 5](#_Toc447182487)

# Overview

ARACNe-AP (Algorithm for the Reconstruction of Accurate Cellular Networks with Adaptive Partitioning) is a complete overhaul of the first ARACNe implementation, originally published by Margolin and colleagues in BMC Bioinformatics in 2006.

ARACNe-AP is a command-line tool written in JAVA®. It requires a JAVA Running Environment version 1.7 or higher.

## Input files needed to run ARACNe

See below for file format specification (or download the test files from our repository)

1. Gene expression matrix.
2. List of regulators (e.g. Transcription Factors)

## Steps required to run ARACNe

1. Calculate a threshold for Mutual Information
2. Run ARACNe on bootstraps of the input matrix
3. Consolidate, i.e. combine the bootstraps into a final network file

## Optional ways to run ARACNe

1. Removing DPI (Data Process Inequality) will preserve every edge that passes the Mutual Information threshold.
2. ARACNe by default operates on a bootstrapped version of the input matrix. It is possible to turn this feature off.
3. During the consolidation step, a Bonferroni correction is applied to the p-values obtained from the Poisson distribution used to determine the significance of each edge according to their appearance in different bootstraps. It is possible to turn this correction off, with the result of having a slightly bigger output network.

## Output of ARACNe-AP

Apart from the individual bootstraps, the consolidation step of ARACNe-AP will produce a file called **network.txt** in the output folder provided by the user. This file shows every significant interaction in four columns

1. The regulator.
2. The target.
3. The MI (Mutual Information) of the pair.
4. The pvalue of the pair (assessed during the consolidation step by integrating the bootstraps).

# Input file format

## Gene lists

A text file, containing one gene symbol per line, e.g.

g165

g196

g257

g367

g401

g1390

## Dataset

A text file, tab separated, with genes on rows and samples on columns

gene Sample1 Sample2 Sample3

g1 1.8 5.2 4.1

g2 5.7 8.3 2.0

g3 6.2 3.1 9.2

g4 7.2 9.1 0.6

# Parameters

**-e** is the expression file

**-d** is an optional expression file for the targets (meaning you can specify an expression file for tfs, and one for targets, with the same sample names. This is for aracne plus)

**-t** is the TF list

**-o** is the output folder

**--consolidate** is telling java to run aracne in consolidate mode (that is, you point it to a directory with bootstraps, and they will be consolidated)

**--calculateThreshold** is telling Java to run it in threshold mode

**-p** is the p-value threshold for the MI to be significant (1E-8 usually)

**--consolidatepvalue** is the p-value threshold for the Poisson test of edge significance in multi-bootstrap mode (if omitted, it is set to 0.05)

**-s** is the optional seed, to make the threshold mode and the bootstrap reproducible

**--threads** is the number of threads (it is used only in standard mode, i.e. bootstrap)

**--nodpi** tells ARACNE not to run DPI

**--nobootstrap** tells ARACNE not to do bootstrapping

**--nobonferroni** removes the Bonferroni correction

# Examples

Note: the examples have been written based on the provided test sets: **matrix.txt** (the gene expression matrix)and **tfs.txt** (the list of regulators). Also, example 3 (the running of 100 bootstraps) is written using a “for loop” as a useful method to run 100 bootstraps with a controlled seed.

## Example 1 – calculate threshold with a fixed seed

java -Xmx5G -jar Aracne.jar -e matrix.txt -o outputFolder --tfs tfs.txt --pvalue 1E-8 --seed 1 --calculateThreshold

## Example 2 – run ARACNe on a single bootstrap

java -Xmx5G -jar Aracne.jar -e matrix.txt -o outputFolder --tfs tfs.txt --pvalue 1E-8 --seed 1

## Example 3 – run 100 reproducible bootstraps

***UNIX loop***

for i in {1..100}

do

java -Xmx5G -jar Aracne.jar -e matrix.txt -o outputFolder --tfs tfs.txt --pvalue 1E-8 --seed $i

done

***Windows loop***

for /l %i in (1, 1, 100) do java -Xmx5G -jar Aracne.jar -e matrix.txt -o outputFolder --tfs tfs.txt --pvalue 1E-8 --seed %i

## Example 4 – consolidate bootstraps in the output folder

java -Xmx5G -jar Aracne.jar -o outputFolder --consolidate

## Example 5 – run a single ARACNE with no bootstrap and no DPI

java -Xmx5G -jar Aracne.jar -e matrix.txt -o outputFolder --tfs tfs.txt --pvalue 1E-8 --seed 1 --nobootstrap --noDPI

## Example 6 – consolidate bootstraps without Bonferroni correction

java -Xmx5G -jar Aracne.jar -o outputFolder --consolidate --nobonferroni