**Manual to use Topological data Analysis for array CGH (TAaCGH)**

This text explains how to use TAaCGH scripts. Details in DeWoskin et al 2010, Arsuaga et al 2015, Ardanza-Trevijano et al. 2016, Gonzalez et al 2018.

TAaCGH currently consist of a set of scripts. Several of them read, create and save files that will be used later in the process. Therefore, input files need to be named in a specific way and everything need to be saved under a specific structure for them to work. Make sure to follow the instruction under SETTING UP.

There are two options to run TAaCGH, either by full arm or by splitting each arm in two sets of equally sized sections. Both set of sections cover the entire arm, but they are displaced by half the length of the sections (see ideogram at Arsuaga et al 2015).

**REQUIREMENTS**

* R
* Python 2 (to run 4\_hom\_stats\_parts.py)
* JDX (Java) needed to run jPlex in 4\_hom\_stats\_parts.py

Libraries

* aCGH library from R: used in 1\_impute\_aCGH.R (http://www.bioconductor.org/packages//2.11/bioc/html/aCGH.html)

**SETTING UP**

TAaCGH reads and writes files under a directory called “Research” within the home directory (~/Research). Within Research there will always be three other folders: “Data”, “TAaCGH” and “ Results”.

THE CODE FOLDER

The programs will be under Research/TAaCGH and must be run from there.

Here is what you need to have under the folder TAaCGH. Each script starts with a number flagging the order to run them. At the beginning of the script there is useful information about the arguments to use as well as the input and output.

|  |  |
| --- | --- |
| **Script** | **Description** |
| 1\_impute\_aCGH.R | It was used to impute our data sets (Needs to be updated) |
| 2\_cgh\_dictionary\_cytoband.R | Creates the “dictionaries” (map with positions for every arm and section), which are used later in almost every script. |
| 3\_Transposed\_aCGH.R | Creates a transposed version of the data (needed for 4\_hom\_stats\_parts.py) |
| 3b\_dist\_Q05.R | Computes the minimum and 5thpercentile of the distance between every pair of points and compute the average of them for the arm/section. The metrics are saved in the dictionary (set\_dict\_cyto.txt) to help you decide the appropriate epsilon increment |
| 4\_hom\_stats\_parts.py | Runs the homology (Betti-0 and Betti-1) |
| 5\_sig\_pcalc\_parts.R | Uses the output from 4\_hom\_stats\_parts.py and performs the permutations to generate the p-values for the difference between the Betti curves |
| 6\_FDR.R | Adjust the p-value with FDR for multiple comparisons |
| 7\_vis\_curves.R | Generates B0 curves for the significant sections resultant  from 6\_FDR.R |
| 8\_probesFDR.R | This program uses permutations to find significant probes (Ho: test=control) within a significant segment from 6\_FDR.R and 7\_vis\_curves.R |
| 9\_mean\_diff\_perm\_NoOut.R | Computing Centers of Mass for each arm and testing for difference (H0:mean(test)=mean(ctrl)). This program could be run just after 2\_cgh\_dictionary\_cytoband.R |
| 10\_class\_pat\_CM.R | Classifies patients for the center of mass of a particular arm (needs manual input from the output of 9\_mean\_diff\_perm\_NoOut.R) |
| 11\_class\_pat\_seg.R | Classifies patients for a significant section |

Other useful scripts (need manual adjustments):

|  |  |
| --- | --- |
| ind\_prof\_origpat\_local \_sect.R | Generates all profiles for a specific section (graphs) |
| ind\_prof\_origpat\_local.R | Generates all profiles for a specific arm (graphs) |
| vis\_avg\_betti\_curves.R | Generates the average B0/B1 curve for test and control and draw a selected patient in each of them (graphs) |

The following scripts need to be in the same directory but are not meant to be handled by the user as they contain functions used by different scripts:

* functions\_cgh.py
* functions\_data\_processing.R
* functions\_io.py
* functions\_sig.R
* plex.jar This is a java file and is a standalone program that runs the homology. Harlan Sexton and Mikael Vejdemo-Johansson. jPlex, December 2008. <http://comptop.stanford.edu/programs/jplex/>
* Readme.pdf

THE DATA FOLDER AND WHAT IS EXPECTED FROM THE DATASET

The input for TAaCGH consists of two files. One with the copy number data (aCGH file) and one with the clinical information (including the phenotype of each patient). The programs assume that input files are in a specific format, with a specific naming formula for each of them and specific variable names. Once the data is pre-processed and imputed, there are 3 main input data files for each data set. Let’s call the data set “SET” (before pre-processing) to follow an example:

Input files:

1. SET\_data\_full.txt
2. SET\_data.txt
3. SET\_phen.txt

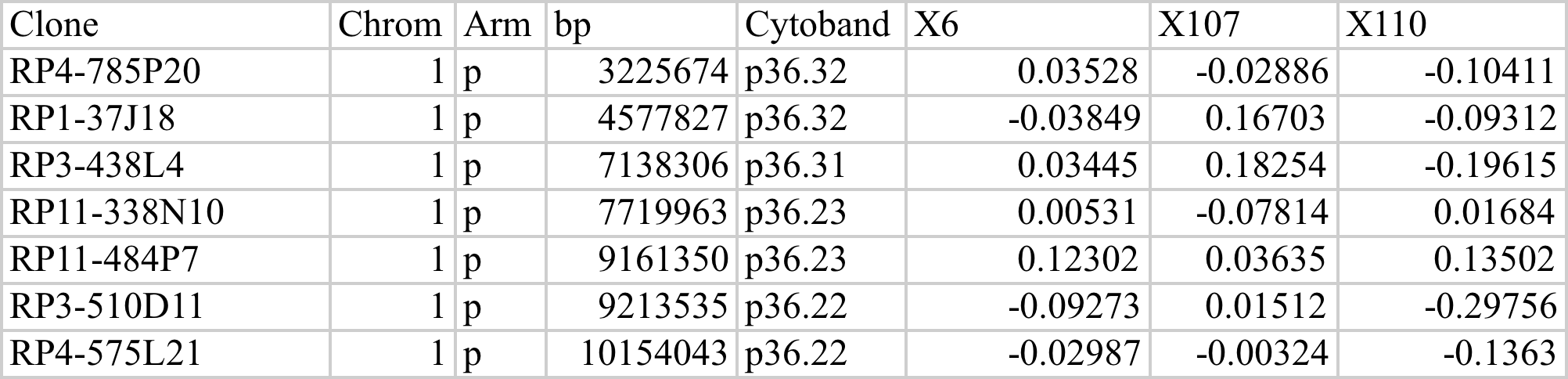
It is expected that the names of the files will always be like that because one of the most used input arguments for the programs is the name of the data set (SET). The program will complete the name of the input files needed from there (completing SET\_data\_full.txt etc) and the program will also create new files using that structure.

Copy number file (aCGH file):   
from SET.txt to SET\_data\_full.txt and SET\_data.txt

You will need to pre-process the aCGH file as there are some columns expected with a specific name and position. Also, the file is expected to be tab delimited. From here to the rest of the document we will call the file "SET.txt" to explain how “exactly” the scripts expect the names of the files in order to work.

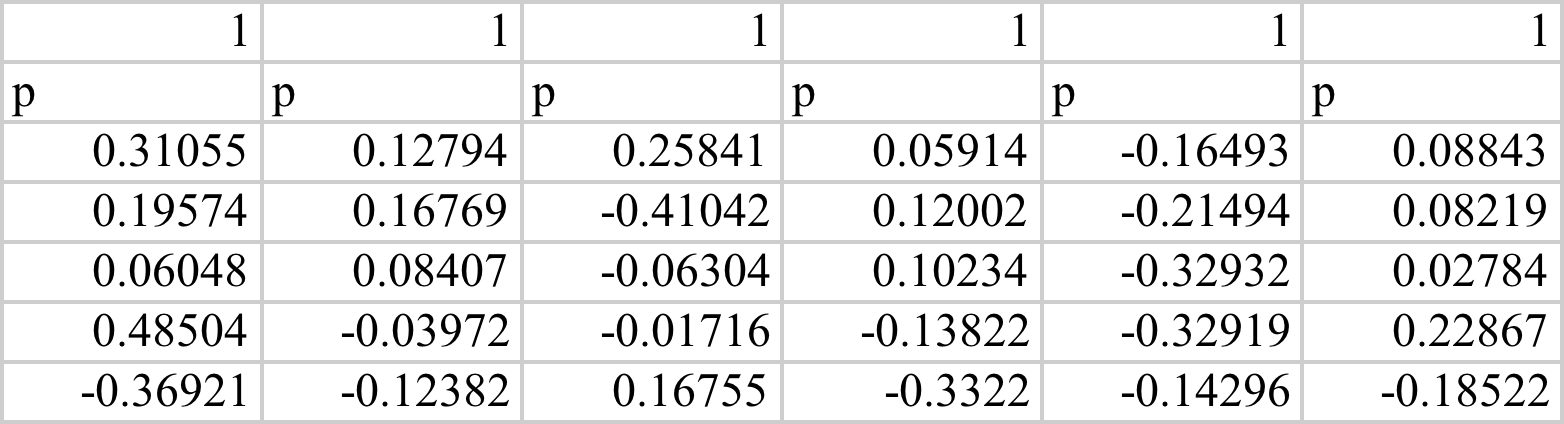
* Update the position (bp) for every clone using the last build from a genome browser
* **VERY IMPORTANT: The file should have the first 5 columns with the information about the clone and its position. The names of those columns must be "Clone" "Chrom" "Arm" "Cytoband" "bp". For X chromosome use 23 and for Y use 24 in humans (otherwise will produce weird ordering to dataSet)**
* **aCGH information for every patient(row) shall start in column number 6**
* Verify that every clone has a column with complete information for Cytoband, Chrom and Arm
* Verify that there are no duplicates in the file for every Chrom/Arm/bp combination. If there are any, they must be averaged
* Impute missing values using 1\_impute\_aCGH.R. This program uses lowess from the aCGH package from Bioconductor (<http://www.bioconductor.org/packages//2.11/bioc/html/aCGH.html)>. The script has not being updated so you might need to adjust it.
* **VERY IMPORTANT: Use a short name for the datasetfile, for instance "SET" and add at the end "data\_full.txt" so that the name of the file become something like SET\_data\_full.txt and save it under ~/Research/Data/SET**
* Inspect the data for outliers and find the chromosome and arm for those clones. If any, register them by chrom-arm combination creating a column in the phenotype file named "out\_chrArm" (e.g. "out\_17q") with 1’s for outliers and 0’s for the rest of the patients. There is no need for a column if the chrom-arm combination did not have any outliers. The patient will not be considered when computing the center of mass of that particular arm

Here is an example of how SET\_data\_full.txt should look



* SET\_data.txt will be created after running 3\_Transposed\_aCGH.R using SET\_data\_full.txt as an input. This is the transposed version of SET\_data\_full.txt. It is mainly the aCGH information for the patients (rows). Only Chrom and Arm are kept in this file (first and second row). This file is only used to get the B0 and B1 curves using 4\_hom\_stats.py. After that, only SET\_data\_full.txt is used. This file also MUST have the patients in the same order as the phenotype file to work properly.

(fragment from SET\_data.txt)

****

Phenotype file

The name of the phenotype file is expected to be SET\_phen.txt (assuming that the aCGH file is called SET.txt). Consists of two or more columns, one with an ID relating the patients to the aCGH file and one or more columns for the different phenotypes. Phenotypes must be recorded as binary variables coded with 1 for patients in the test group and 0 for those in control. Save the file under ~/Research/Data/SET

**VERY IMPORTANT**: **The program works using an index, therefore the patients in phenotype file need to be IN THE SAME ORDER as the patients in the aCGH file. And of course it should have the exact same number of patients, no more, no less.**

**RUNNING THE SCRIPTS**

All the programs (except for two of them. See below) are meant to run in the terminal by calling a command with the necessary arguments. Information about the purpose of the script, arguments and examples of the command lines are available at the beginning of the script. However, it is always possible to run the scripts within R by commenting the arguments line and providing the input locally. The purpose of running the script from the terminal is to facilitate parallel work and the use of a cluster as for some datasets it might take many hours to finish the job. I recommend starting with a small dataset to get familiar with the scripts and at some point considering using a cluster when you move to the full dataset.

The name of the scripts shows the order in which you need to run them. Here is a general description of the steps:

* 1\_impute\_aCGH.R (Need to be updated)  
  If the aCGH file has no missing values, you can skip script 1\_impute\_aCGH.R.
* Decide if you will run TAaCGH by complete arms or by sections. That will be an argument used in all of the following scripts
* 2\_cgh\_dictionary\_cytoband.R.   
  From SET\_data\_full.txt [see Settings, Copy number file (aCGH file)] create a "dictionary" specifying the row index positions (start and end) for each segment within the arms or for a full arm depending of the argument “action”. You need to decide a number (total number of parts) to run the the program in smaller parts. By running each part in parallel with only part of the data we reduce the time to run the homology and the permutations. It uses a minimum of clones (segLength) discarding those arms with not enough clones to run the study. This script will overwrite set\_data\_full.txt after reordering the dataset and will generate set\_data\_orig.txt with the original dataset.
* 3\_Transposed\_aCGH.R   
  Creates SET\_data.txt, the transposed version of SET\_data\_full.txt using 3\_Transposed\_aCGH.R. This file will be needed in script 4\_hom\_stats\_parts.py.
* 3b\_dist\_Q05.R  
  Computes summary statistics (Average minimum and average 5% percentile) for the distance between points using 3b\_dist\_Q05.R. Two columns ("Avg\_Min" and "Avg\_Q05") will be added to the file SET\_dict\_cyto.txt. This information is useful to select the epsilon increment. I suggest taking a number in between MIN(Avg\_Min) and MIN(Avg\_Q05).
* 4\_hom\_stats\_parts.py  
  Run the homology (B0 or B1) feeding SET\_data.txt into 4\_hom\_stats\_parts.py. You will need to run it as many times as the number of parts used to split the dictionary. The script will create the folder “Results” and a subfolder with the name of the dataset ( ~/Research/Results/SET) to save information about the homology for each section in the dictionary by patient. You will find two different kind of files:
  + Barcode file named Inter\_2D\_hom1\_SET\_ChromArm\_patientNo\_segNo.txt  
    under ~/Research/Results/SET/subdir/2D/Homology/Chromosome\_number
  + Jagged file (B0\_2D\_SET\_ChromArm\_segNo.txt) with the value for the homology for each patient (row) at each increment for epsilon (columns). The Epsilon increment is saved for future reference in a text file named Epsilon\_XXX.txt under the same directory as the jagged files.
* 5\_sig\_pcalc\_parts.R  
  Computes the un-adjusted p-values. You will need to run it as many times as the number of parts used to split the dictionary. The test statistic does not separate those pieces of areas between the curve when test is on top or control is on top. Up to now, the analyst must decide if the significance might be attributed to test or to control by either looking at the columns “AreaPos” and “AreaNeg” which separate them, or by inspecting the curves using 7\_vis\_curves.R
* 6\_FDR.R  
  Adjust the p-values using FDR. You will need to run this script only once even though it uses all parts with the un-adjusted p-values.
* 7\_vis\_curves.R  
  Generate B0 curves for significant sections in xxx\_FDRsig.txt (from 6\_FDR.R. Inspect manually the output and keep only as significant those sections with the test curve (blue) above the control curve (red). Generate a reviewed file with only the new significant sections and name it xxxx\_FDRsig\_rev.txt.
* 8\_probesFDR.R   
  Find the significant probes from significant sections by the file xxx\_FDRsig\_rev.txt into it. The output is two files, one with p-values from all significant sections and a second file with only those probes under the desired significant level. This second file will also show if the probe is a gain or a loss.
* 9\_mean\_diff.perm.R   
  This program computes the centers of mass for each chromosome-arm combination and performs the statistical test for the difference between test and control for a specific phenotype. The only input is the aCGH file SET\_data\_full.txt and the phenotype file. It could be run right after creating the dictionaries (script 2). There are two output files, one with the results for all centers of mass and a second one displaying only those arms with a p-value under the significance value and it also shows if it is a gain or a loss.
* 10\_class\_pat\_CM.R

This script will classify a patient as significant for the center of mass if the average for the arm (CM of the patient) lays outside de confidence interval from the control group of the phenotype. The confidence interval is from a t distribution:

avg +/- t\*sd/sqrt(n) t=alfa/2 percentile from a t with n-1 d.f.

This script is still under development. For now, it only works within R and input need to be provided manually by inspecting the output from 9\_mean\_diff.perm.R

* 11\_class\_pat\_seg.R  
  The purpose of the script is to classify each patient as aberrant (1) or non-aberrant (0) for a section that was found significant after 6\_FDR.R. To do so, it compares the homology curve from a patient against the average homology curve for test and for control using the leave-one-out procedure. The output is a new column generated in the phenotype file with binary data. This script is still under development. For now, it only works within R and input need to be provided manually.

TODO: need to create a warning when adding columns to phenotype file to avoid overwriting variables.

**REFERENCES**

* Sergio Ardanza-Trevijano, Georgina Gonzalez, Tyler Borrman, Juan Luis Garcia, and Javier Arsuaga. Topological analysis of amplicon structure in comparative genomic hybridization (cgh) data: an application to erbb2/her2/neu amplified tumors. In International Workshop on Computational Topology in Image Context, pages 113–129. Springer, 2016
* Javier Arsuaga, Tyler Borrman, Raymond Cavalcante, Georgina Gonzalez, and Catherine Park. Identification of copy number aberrations in breast cancer subtypes using persistence topology. Microarrays, 4(3):339–369, 2015.
* Daniel DeWoskin, Joan Climent, I Cruz-White, Mariel Vazquez, Catherine Park, and Javier Arsuaga. Applications of computational homology to the analysis of treatment response in breast cancer patients. Topology and its Applications, 157(1):157–164, 2010
* Gonzalez G., Ushakova A., Sazdanovic R., Arsuaga J., 2018. Prediction in cancer genomics using topological signatures illustrated with logistic regression [Submitted manuscript]
* Harlan Sexton and Mikael Vejdemo-Johansson. jPlex, December 2008. <http://comptop.stanford.edu/programs/jplex/>