GAIA: Genomic Analysis of Important Aberrations

Sandro Morganella Stefano Maria Pagnotta
Michele Ceccarelli

Contents

1 Overview

A current challenge in biology is the characterization of genetic mutations that occur as response to a particular disease. Development of array comparative genomic hybridization (aCGH) technology has been a very important step in genomic mutation analysis, indeed, it enables copy number measurement in hundreds of thousands of genomic points (called markers or probes). Despite the high resolution of aCGH, accurate analysis of these data is yet a challenge. In particular a major difficulty in mutation identification is the distinction between driver mutations (that play a fundamental role in cancer progression) and passenger mutations (which are random alterations with no selective advantages).

This document describes classes and functions of GAIA (Genomic Analysis of Important Aberrations) package. GAIA uses a statistical framework based

on a conservative permutation test allowing the estimation of the probability distribution of the contemporary mutations expected for non-driver markers. Afterwards, the computed probability distribution and the observed data are used to assess the statistical significance (in terms of q-value) of each marker. Finally an iterative "peel-off" procedure is used to identify the most significant independent regions which have a high probability to correspond to driver mutations.

GAIA algorithm is carefully described in [1].

2 Installation

Install GAIA on your computer by using the following command:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("gaia")
```

GAIA package can be loaded in R by using the following command:

```
> library(gaia)
```

3 Package Dependencies

In order to use GAIA you need of the R package `qvalue` available at the bioconductor repository.
Note that by using the installation command `BiocManager::install` the `qvalue` dependency is automatically fulfilled. In contrast if GAIA has been manually installed (e.g. using `R CMD INSTALL` command) you can install `qvalue` package by the following command:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("qvalue")
```

4 Vega Data Description

In this section the data object used by GAIA will be described. This description will be supported by the data provided in GAIA package.

4.1 Marker Descriptor Matrix

This matrix contains all needed informations about the observed markers (also called probes). This matrix has a row for each marker and each marker is described by a set of column reporting the name of the probe and its genomic position. In particular the matrix has the following columns:

Probe Name : The name of the observed probe;

Chromosome : The chromosome where the probe is located;

Start : The genomic position (in bp) where the probe starts;

End : The genomic position (in bp) where the probe ends;

The column specifying the **End** position is optional and when it is missed, start and end positions are considered to be coincident. This is the case of the data provided within the package.

In order to load the marker descriptor matrix provided in GAIA use the following command:

```
> data(synthMarkers_Matrix)
```

This marker descriptor matrix simulates the genomic position for the probes of 24 chromosomes (each chromosome has 1000 probes). Chromosomes 23 and 24 represents sex chromosomes X and Y respectively.

4.2 Aberrant Region Descriptor Matrix

This matrix contains all needed informations about the observed aberrant regions.This matrix has a row for each aberrant region and each of them is described by the following columns:

Sample Name : The name of the sample in which the aberrant region is observed;

Chromosome : The chromosome where the aberrant region is located;

Start : The genomic position (in bp) where the aberrant region starts;

End : The genomic position (in bp) where the aberrant region ends;

Num of Markers : The number of markers contained in the aberrant region;

Aberration : An integer indicating the kind of the mutation.

The column **Aberration** can assume only integer values in the range $0, \cdots, K-1$ where K is the number of considered aberrations. For example if we are considering loss, LOH and gain mutations than the only valid values for the column **Aberration** are 0, 1 and 2.

In order to load the aberrant region descriptor matrix use the following command:

```
> data(synthCNV_Matrix)
```

This aberrant region descriptor matrix simulates 10 samples for the chromosomes described by `synthMarkers_Matrix`. A summary of the aberrant regions contained in this matrix follows:

4.3 Real aCGH Datset

In GAIA a real aCGH dataset is also provided, in the next we provide a brief description of this dataset. Raw data were preprocessed by PennCNV tool [5] and segmented by using Vega R/Bioconductor package [4].

Chromosome	Frequency Percentage	Start	End	Aberration Kind
1	100%	301	700	0
2	80%	301	700	0
3	60%	301	700	0
4	40%	301	700	0
5	20%	301	700	0
10	100%	1	700	1
11	80%	1	700	1
12	60%	1	700	1
13	40%	1	700	1
14	20%	1	700	1
20	100%	801	1000	2
21	80%	801	1000	2
22	60%	801	1000	2
23	40%	801	1000	2
24	20%	801	1000	2

Table 1: `synthMarkers_Matrix`: Aberrant Region Detail. The column **Frequency Percentage** reports the percentage of samples containing the indicated aberration.

4.3.1 Colorectal Cancer (CRC) Dataset

CRC dataset is composed by 30 samples hybridized on SNP 250k Affymetrix GeneChip array [2]. Raw data are available in GEO with identifier GSE13429. Patients of this dataset were diagnosed with microsatellite-stable CRC without polyposis.

In order to run GAIA on this dataset both aberrant region descriptor matrix (`crc`) and marker descriptor matrix (`crc_markers`) are provided.

5 Function Description

In this section all exported functions of GAIA package are described. GAIA works on aCGH data in which hundreds of thousands probes are contemporary observed, so the data loading phase can be very time consuming. For this reason in GAIA two data loading functions are provided (see Sections 5.1 and 5.2). By using this functions the users can load just one time the data and save they into data objects that can be used for several GAIA executions.

5.1 `load_markers`

This function builds the marker descriptor object used in GAIA from a marker matrix as described in Section 4.1:

```
> markers_obj <- load_markers(synthMarkers_Matrix)
```

The `marker_obj` provides an easy access to the data informations contained in the marker matrix and and it is organized as a list:

the element `marker_obj[[i]]` contains the descriptions of all observed markers for the i-th chromosome and it is organized as a matrix of dimension $2 \times M$

(M is the number of observed probes for the i-th chromosome). First and second row of this matrix contains start and end position respectively.

5.2 load_cnv

This function builds the aberrant region descriptor object used in GAIA by using both the region matrix described in Section 4.2 and the marker descriptor object created by the function `load_markers`, in addition the number of the analyzed samples must be passed as argument:

```
> cnv_obj <- load_cnv(synthCNV_Matrix, markers_obj, 10)
```

The `cnv_obj` is organized as a double list:

the element `cnv_obj[[j]][[i]]` contains the informations about the j-th aberration of the i-th chromosome. In particular `cnv_obj[[j]][[i]]` is a matrix of dimension $N \times M$ where N is the number of samples (in the example 10) and M the the number of observed probes for the i-th chromosome. The element `cnv_obj[[j]][[i]][n,m]` is equal to 1 if in the marker m of the chromosome i a mutation of kind j is observed, it is equal to 0 otherwise.

5.3 runGAIA

This is the core function of the package, indeed it allows to execute GAIA algorithm. In particular `runGAIA` has the following header:

```
runGAIA(cnv_obj, markers_obj, output_file_name="", aberrations = -1,
chromosomes = -1, num_iterations = 10, threshold = 0.25)
```

`cnv_obj` : The aberrant region descriptor object created by using the function `load_cnv` (see Section 5.2);

`markers_obj` : The marker descriptor object created by the function `load_markers` (see Section 5.1);

`output_file_name` : (default "") The file name used to save the significant aberrant regions. If this argument is not specified the results will be not saved into a file;

`aberrations` : (default -1) The aberrations that will be analyzed. The default value -1 indicates that all aberration will be analyzed;

`chromosomes` : (default -1) The chromosomes that will be to analyzed. The default value -1 indicates that all chromosomes will be analyzed;

`num_iterations` : (default 10) if the number of permutation steps (if approximation is equal to -1) - the number of column of the approximation matrix (if approximation is different to -1);

`threshold` : (default 0.25) Markers having q-values lower than this threshold are labeled as significantly aberrant. This parameter must be a real number in the range $0 - 1$;

`hom_threshold` : (default 0.12) Threshold used in homogeneous peel-off (for more detail see Section 6). This parameter must be a real number in the range $0 - 1$ and by using values lower than -1 homogeneous peel-off is disabled and standard peel-off procedure is used;

`approximation` : (default=FALSE) if approximation is FALSE then GAIA explicitly performs the permutations, if it is TRUE then GAIA uses an approximated approach to compute the null distribution.

Suppose that we want to analyze all aberrations and all chromosomes and that we want to save the results within the file CompleteResults.txt, than we use the following command:

```
> results <- runGAIA(cnv_obj, markers_obj, "CompleteResults.txt")
```

Both in the file CompleteResults.txt and in the matrix variable `results` we can found all significant aberrant regions. In particular the following column are used to describe each significant aberrant region:

Chromosome : The chromosome where the aberrant region is located;

Aberration Kind : An integer indicating the kind of the mutation;

Region Start [bp]: The genomic position (in bp) where the aberrant region starts;

Region End [bp]: The genomic position (in bp) where the aberrant region ends;

Region Size [bp]: The size (in bp) of the aberrant region;

q-value : The estimated q-value for the aberrant region.

So with the following command we print the significant aberrant regions having the minimum q-values:

```
> results[which(results[,6]==min(results[,6])),]
```

Suppose now that we want to analyze only the aberration 1 on the chromosomes 10, 11 and 14 and that we want to save the results within the file Results.txt. In addition we increase the significance threshold value from 0.25 to 0.5, than we use the following command:

```
> results <- runGAIA(cnv_obj, markers_obj, "Results.txt", aberrations=1, chromosomes=c(10,
```

6 Homogeneous Peel-off

In GAIA a new peel-off procedure, called homogeneous, is available. By usinh homogeneous peel-off significant regions are detected by using both statistical significance and within-sample homogeneity, so that more accurate results can be obtained. Homogeneous peel-off is disabled for values lower than 0.

Homogeneous peel-off can be used only if two kinds of aberrations are observed. In the next we show as homogeneous peel-off can be used on CRC dataset. The first step is composed by the loading of the data and the creation of the respective markers and aberrant regions descriptor objects:

```
> data(crc_markers)
> data(crc)
> crc_markers_obj <- load_markers(crc_markers)
> crc_cnv_obj <- load_cnv(crc, crc_markers_obj, 30)
```

Now we can run GAIA with homogeneous pee-off on the chromosome 14 by using the suggested value for `hom_threshold` of 0.12:

```
> res <- runGAIA(crc_cnv_obj, crc_markers_obj, "crcResults.txt",  chromosomes=14, hom_thre
```

Results of the computation are saved in the file crcResults.txt

7 Run GAIA with Approximation

In GAIA a new approach to compute the null distribution is available. This approach performs an approximation of the permutations. In particular, after computed the frequency of alteration for each sample θ_j, GAIA simulates a matrix with dimension $N \times K$ (N is the number of samples and K is the number of performed approximation) where each column is a vector in which the element j has a probability for drawing 1 equal to θ_j. This asymptotically approximates the original simulation when $M \to \infty$ and since M is large enough, it is well approximated in practice. So we suggest to use this approach in high resolution scenario.

With the following command we run GAIA in its approximated version on all chromosomes by performing $K = 5000$ approximations.

```
> res <- runGAIA(crc_cnv_obj, crc_markers_obj, "crc_approx_Results.txt", hom_threshold=0.1
```

Results of the computation are saved in the file crcResults.txt

8 Integration of GAIA and Integrative Genomics Viewer

The Integrative Genomics Viewer (IGV) [6] is a high-performance visualization tool for interactive exploration of large, integrated datasets. It supports a wide variety of data types including sequence alignments, microarrays, and genomic annotations.

From version 1.5 GAIA produces in output a ".gistic"file that can be loaded in IGV so that computed q-values for all analyzed chromosomes can be plotted.

References

[1] Morganella,S. *et al.* (2010) Finding recurrent copy number alterations preserving within-sample homogeneity. *Bioinformatics*, DOI: 10.1093/bioinformatics/btr488.

[2] Venkatachalam,R. *et al.* (2010) Identification of candidate predisposing copy number variants in familial and early-onset colorectal cancer patients. *Int. J. Cancer.*

[3] Astolfi,A. *et al.* (2010) A molecular portrait of gastrointestinal stromal tumors: an integrative analysis of gene expression profiling and high-resolution genomic copy number. *Laboratory Investigation* **90**(9), 1285-1294.

[4] Morganella,S. *et al.* (2010) VEGA: variational segmentation for copy number detection. *Bioinformatics* **26**(25), 3020-3027.

[5] Wang,K. *et al.* (2007) PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research* **17**, 1665-1674.

[6] Integrative Genomics Viewer (IGV): `http://www.broadinstitute.org/software/igv/home`