GACK

Genomotyping Analysis by Charlie Kim

User's Manual

cckim@stanford.edu

Introduction

GACK addresses the need for better analytical methods for microarray-based genome-composition analyses. Currently, the most common approach to such analyses is to choose an empirically determined signal ratio value as a cutoff for assignment of genes into either present or divergent categories. Some of the problems with this approach are described in the paper describing the GACK algorithm. This manual includes practical points for use of the software, as well as a detailed treatment of the algorithm.

Requirements

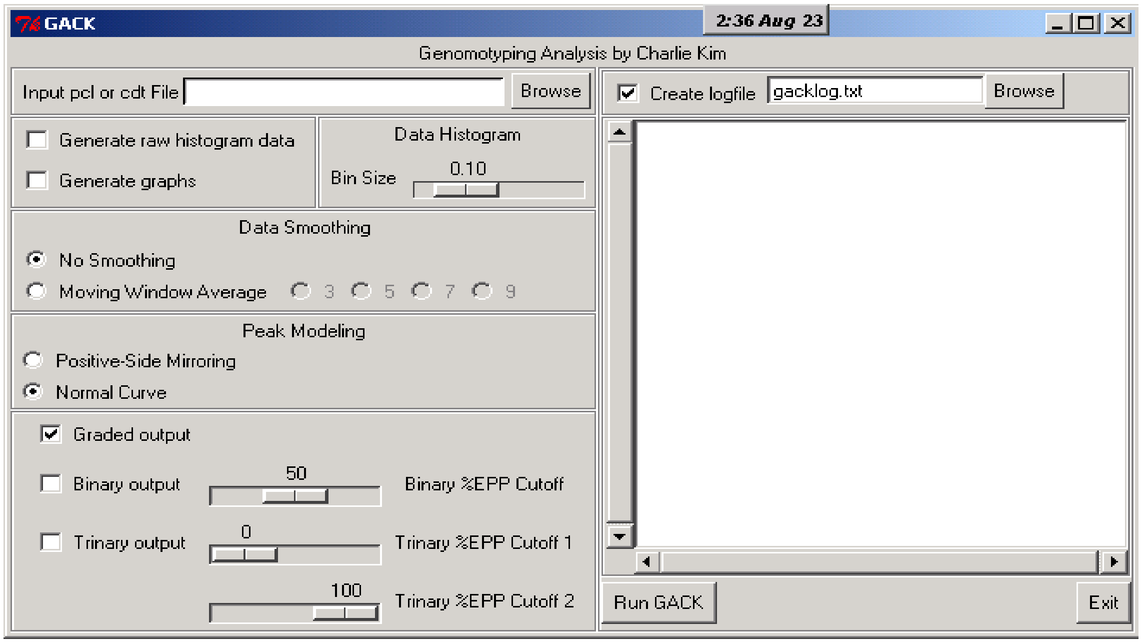
Most users will want to use the Windows executable version of GACK. This can be downloaded at http://falkow.stanford.edu/whatwedo/software. This version should run on any computer running any version of Windows. There is no installation program; the program runs without any setup.

For those interested in the more technical aspects, GACK is written in Perl with a Tk interface. The source is freely available, and has been tested primarily on Windows systems. It works under Windows 98, NT, and 2000 for sure; I believe it will works on others as well. I have also successfully tested some of the older versions on Linux. I believe it will run on most unices quite easily, and if not, porting should be trivial. The Windows executable was generated using Activestate's Perlapp.

Using the Software

Getting Started

To run the Windows executable, download it anywhere on your computer and simply double-click to run. You should see the interface appear, as in Fig 1. If you downloaded the source, I'm going to assume you know how to get things running on your own. You will need Perl plus a few modules listed in the beginning of the source code (Tk, GD, GD::Graph, Math::Round).



Input pcl or cdt File

☑ Create logfile gacklog.txt

□ Generate raw histogram data

□ Generate graphs

Data Histogram

Bin Size 0.10

Data Smoothing

● No Smoothing

○ Moving Window Average ○ 3 ○ 5 ○ 7 ○ 9

Peak Modeling

○ Positive-Side Mirroring

● Normal Curve

☑ Graded output

□ Binary output 50 Binary %EPP Cutoff

□ Trinary output 0 Trinary %EPP Cutoff 1

100 Trinary %EPP Cutoff 2

Fig 1 The GACK interface (version 3.630)

<u>Input File</u>

I originally started out trying to support a variety of different input file formats. I ultimately decided to focus on using the .pcl (pre-clustering) format, as it is already in common use in the Stanford microarray community. A description of this and other formats is available at http://falkow.stanford.edu/whatwedo/software. GACK also supports .cdt files, so that you can run GACK on a dataset which has already been clustered if desired. There are differences between clustering followed by GACKing versus GACKing followed by clustering. Unfortunately, I don't know which is better.

I would suggest isolating your input file in a directory of its own, as most of the time GACK will generate multiple files. This will keep your files more organized in the long run.

IMPORTANT NOTE: The data must be in log scale in order for the assumption of normality to be valid. The data is not normal in linear scale, although upon quick glance it may appear to be so. We use $\log_2$ transformed data, and I suggest you do as well. If you use a different log base, you will have to fiddle to get the bin size right because the distribution will have a different width. If you use $\log_2$ transformed data, a bin size of 0.1 should be fine.

Generate Raw Histogram Data

The GACK algorithm requires a histogram of the ratios to map the normal curve. It is sometimes useful to generate the raw data used in the histogram, especially for learning purposes and confirming that the algorithm is working properly on your dataset. Selecting this checkbox will generate a file of histogram values for each dataset in your input file. In addition, the output will contain values for the mapped curve (normal or positive-side mirrored) and EPP distributions (see below, and manuscript).

Generate Graphs

I recently added a feature for automatic generation of a .jpg file containing a histogram of the frequency, normal, and EPP distributions. Check this box to turn on this feature. Each histogram is output in a separate file.

Data Histogram

The only option currently available is bin size. A bin is the numerical increment by which a histogram's categories increase. For example, if a bin size is 0.1, the categories might be 0.1, 0.2, and 0.3. If you data is in $\log_2$, a bin of 0.1 should work fine. If you are in a different scale, you may need to change this to improve the normal approximation.

Data Smoothing

Depending on the quality of the hybridization and the bin size chosen, the histogram may or may not be noisy. In the case of noisy data, it may be appropriate to apply a smoothing algorithm. I HAVE NOT TESTED THIS EXTENSIVELY. If you data is noisy, I recommend playing with the bin sizes first. Nevertheless, I have implemented a moving window average, with sizes from 3 to 9 available. My experience was that with good data, averaging was not necessary; you should check the histograms yourself by generating the raw histogram data and look for smoothness in the distribution. You can do this in Excel or more conveniently using a program such as HIMACK (see "Additional Info" below). Note: automatic histogram generation described above circumvents the need to produce the results manually.

Peak Modeling

Two algorithms are currently available: positive-side mirroring and normal curve. In positive-side mirroring, the main peak is located, and the data to the right of that peak are mirrored over the peak point to create a pseudo-bell-shaped curve. This distribution is used to estimate the distribution of the "present" genes. The normal curve selection uses three points along the data distribution to map a normal curve to the data, also estimating the distribution of the "present" genes. While positive-side mirroring works with very high quality datasets, I found that the normal curve was much more robust with datasets from hybridizations of lower quality (in addition to working well with high quality hybridizations). I therefore recommend that you NEVER use the right-side modeling. I have only left it in the code for future flexibility of analysis options.

Output Formats

Three different visualization outputs are available: graded, binary, and trinary. The simplest is binary output, which results in present genes being designated "1" and divergent genes being designated "0." Trinary output designates present genes as "1," divergent genes as "-1," and uncertain genes as "0." Graded output generates a range of values from -0.5 to 0.5 in increments of 0.5, with -0.5 corresponding to high likelihood of divergence and 0.5 corresponding to high likelihood of presence.

Because binary and trinary outputs do not preserve any information regarding the probability that a given gene is present (whereas graded output does), the user must select cutoff values for assignment. Every gene has an estimated probability of being present (EPP), which can be used to assign genes into divergent, present, and/or uncertain categories. For example, in a binary analysis, one may wish to assign genes with greater than 50% chance of being present to the present category, and genes with less than 50% chance of being present to the divergent category. This is achieved using the EPP slider bars next to the binary and trinary output checkboxes.

In the case of a trinary analysis, one may wish to assign certain genes to an uncertain category. For example, you may wish to be 95% certain of your assignments to the present and divergent categories. In this case, you should set the cutoffs at 5% and 95%. Spots that fall into the 5-95% range will be classified with a "0" instead of a "-1" or "1."

Further Analysis

The purpose of GACK is to categorize the data into a more conceptually manageable format. Raw or normalized signal values from a microarray are meaningless without the context of the distribution; GACK takes the context into account and reduces the complexity of the dataset by representing divergent, present, and uncertain genes on a number scale which is consistent from dataset to dataset.

The advantage of this is simplicity in analysis. The outputs from GACK can be opened directly in Michael Eisen's Treeview (http://rana.lbl.gov/), which is already widely used in microarray analysis. If this is done, you will need to adjust the contrast (divide it in half) to get more attractive graphics (this is because of the contrast algorithm used by Treeview).

Many will also be interested in doing phylogenic analysis on the microarray data. While Cluster can generate a dendrogram which appears to reflect phylogenic trees fairly well, the validity of using Cluster to determine phylogenic relationships has not yet been established or accepted. More traditional tools in packages such as Phylip or PAUP* should work with the GACKed data (try the quantitative or discrete characters methods), although I have not tested this myself.

Recommendations for Analysis

For binary analysis, I would consider the type of analysis you would like to do. For an analysis of divergent genes, the most stringent analysis (close to zero false positives) would use a 0% EPP cutoff. For a stringent analysis of present genes, one would use a 100% EPP cutoff. For exploratory analysis, I would simply use the default 50% EPP cutoff, or better yet, use graded or trinary outputs.

For trinary analysis, 0% and 100% should be used as cutoffs in most cases to minimize any uncertainty in the divergent/present predictions. If a small degree of false assignment is tolerable, the uncertain range can be narrows (e.g. 10-90%, or 20-80%).

Graded analysis has no options because it preserves all of the predictive EPP information (although translated to a scale of -0.5 to 0.5, instead of 0% to 100%). I recommend this as the output of choice in most situations, as it preserves the most information about relative position of the spot's signal intensity along the signal distribution.

Further discussion of EPP is available in the manuscript.

Log file

All information can be logged into the designated text file. The log will contain the same information that is displayed in the message window during the program's execution.

Additional Info

A critical file which is generated in every run of GACK is the "EPP.val" file. This file contains the signal which corresponds to every EPP value. For example, a certain signal intensity will correspond to 0% EPP, 5% EPP, 10% EPP, … 100% EPP. Each array will have a specific set of signal values which correspond to the EPP values. This can provide useful information in understanding how the assignments are being made.

I highly recommend looking at a histogram of your data in order to assess the quality. You can generate the histograms automatically by selecting the appropriate check box. Alternatively, this can be done in Excel, or more conveniently using HIMACK (Histogram Maker, http://falkow.stanford.edu/software). In general, good genomotyping data tends to have a smooth, approximately normal peak with a left-side tail. If your data does not look like this, GACK may fail to accurately make gene predictions.

Additional Treatment of the Algorithm

To date, most genomotyping analysis (genome composition using microarrays) has been performed by normalizing the distribution, then using a constant as a cutoff point for determination of presence or absence of a gene. Unfortunately, this sort of static analysis method does not account for variation of the distributions between strains, as well as experimental variations which result in variable classification of genes.

Consider the following example, in which *Salmonella enterica* serovar Typhimurium is compared to *Salmonella bongori* on a serovar Typhimurium microarray. The data are normalized to log(rat2N) of zero, to reflect the case in which the genes are present in both strains.

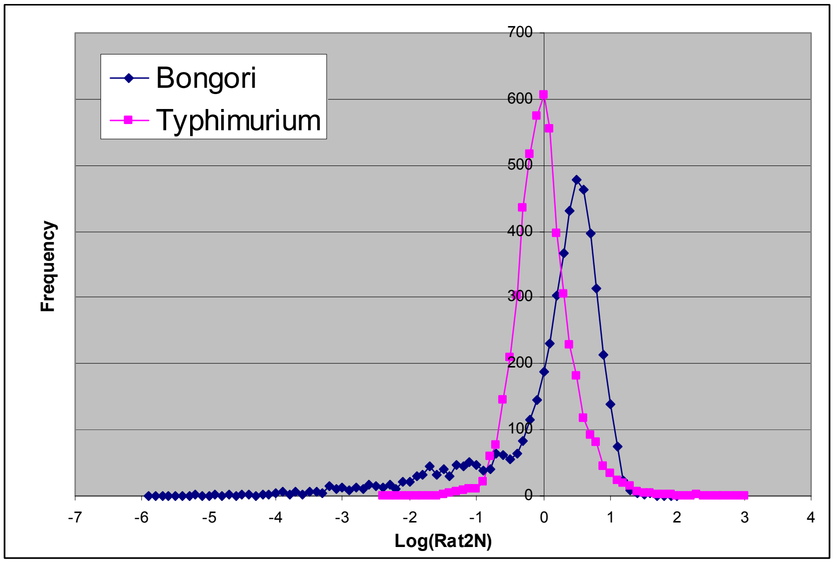


Figure 2

Many genes that are present in Typhimurium are divergent in Bongori, as manifested in the long left-side tail on the Bongori histogram (Figure 2). During normalization, the main peak of present genes must be shifted to the right in order to compensate for this long tail of divergent.

In determining a cutoff, one assumes that genes with log(rat2N) values lower than a certain constant will be absent. For example, since Typhimurium seems to have very few genes with ratios lower than -1, one might expect that Bongori genes with log(rat2N) of lower than -1 are absent from Bongori, but present in Typhimurium. However, observe the region from -0.5 to -1, which contains a large number of genes also likely to be absent from Bongori. These genes would be falsely categorized as present in this constant cutoff analysis.

The analysis because even more complex when more strains are added to the dataset. The strains have varying numbers of missing genes; thus, the peaks are shifted to varying degrees (where more missing genes shifts the peak further to the right), resulting in the constant cutoff value intersecting the distributions at varying points. In general, this constant cutoff method can be considered conservative, as it underestimates the number of missing genes. Conversely, many genes are then falsely classified as present. The proportion of genes that are falsely considered present will change from strain to strain depending on the true number of divergent genes.

Another potential problem of the constant cutoff analysis is observed when the hybridizations vary in their signal distribution widths. If the distribution of the log(rat2N) is wide, a portion of the main peak will spill over across the constant cutoff, resulting in many present genes being falsely called absent. This sort of error is impossible to detect without manual plotting of the log(rat2N) data for each strain.

For these reasons, a dynamic method for determining cutoff points was desired. We begin with dataset that may or may not be normalized. The algorithm is not sensitive to normalization, so this point is irrelevant. We will use the Bongori dataset described above.

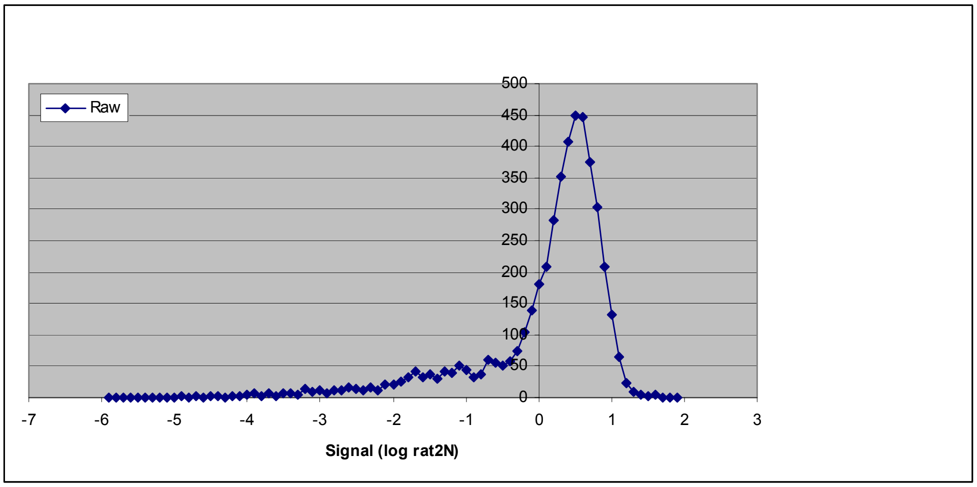


Figure 3

This particular dataset is normalized, but the peak fails to match zero due to the long left-side tail, as noted above. Different strains will have divergent numbers of divergent genes, and thus the main peak will be shifted to different degrees. It is evident that this is not desirable, as a constant cutoff (typically -1.0) would intersect the different distributions at different points. The first step of the GACK algorithm is to generate a histogram based on a user-defined bin size (0.1 in the case of the Bongori data shown above).

A smoothing algorithm may or may not be applied to the histogram to smooth over fluctuations in the data. Good datasets, such as this Bongori data, generally do not require smoothing. A moving window smoothing algorithm is available. Binomial smoothing was in the works, but I have found that smoothing is generally not necessary, and so have no plans to develop binomial smoothing or any other smoothing algorithms.

The next step in the algorithm is to estimate the distribution of the present genes. That is, we wish to mathematically model the main peak. Two algorithms are available: positive-side mirroring and normal curve mapping. In positive-side mirroring, the peak's maximum value is determined, and the log(rat2N) values to the right of the peak are mirrored over the peak to create a symmetrical distribution which estimates the present genes. In the normal curve mapping algorithm, the two side values of the peak at 50% of the peak's height are used to calculate a mean and standard deviation, which are used to generate a normal curve. We have found that the normal curve algorithm is much more robust than the positive-side mirroring algorithm, and recommend that the positive-side mirroring algorithm not be used.

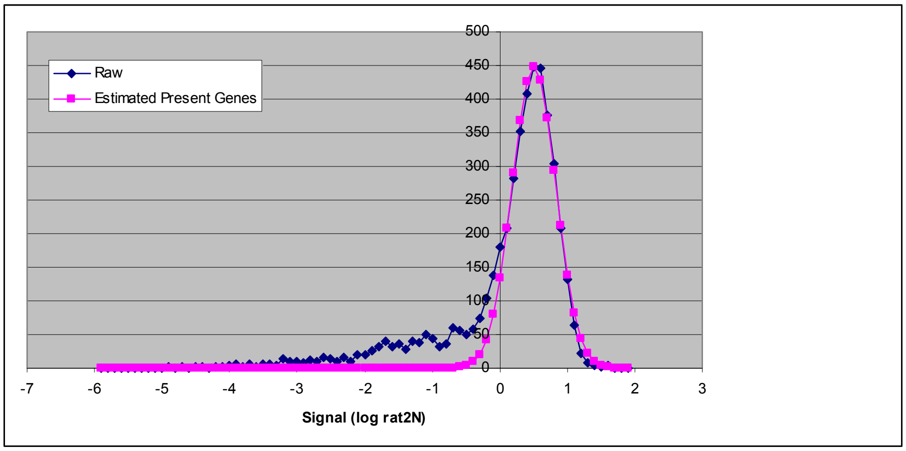


Figure 4

The estimated distribution of present genes in Figure 4 approximates the expected distribution of genes which are present in the strain. It is evident that the observed data deviates substantially from the expected distribution shape ("fake bell curve") if the strain is divergent. This left-tail is the mathematical manifestation of divergent genes.

We now have an estimate of the number of present genes for any given signal value. In addition, we have the actual observed number of genes with any given signal value from the raw dataset. We can therefore calculate the estimated proportion of present genes for a given signal intensity. This is the estimated probability of presence (EPP).

% EPP = 100% * (estimated / observed)

We calculate the %EPP for each signal bin in the histogram (Figure 5).

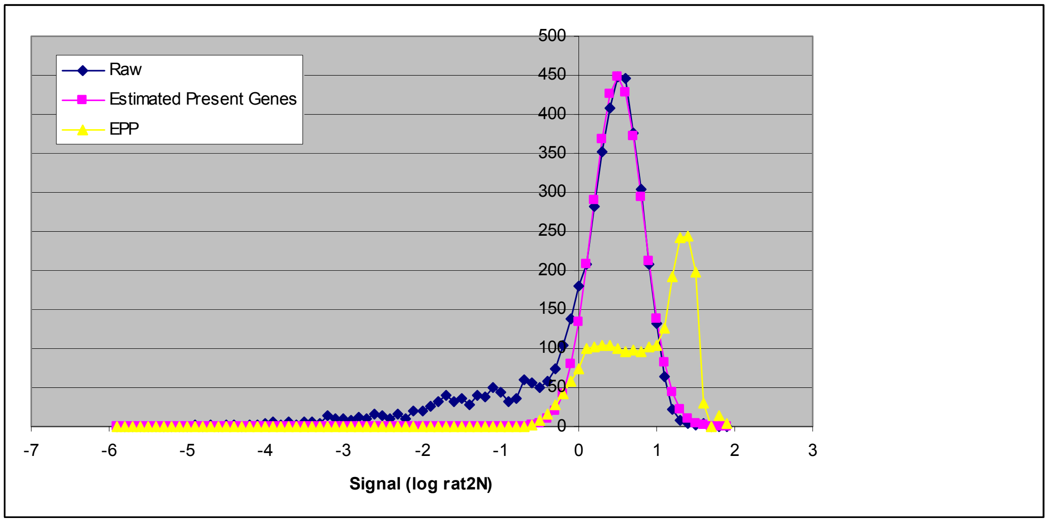


Figure 5

On the far left side of the distributions, we expect no present genes (based on the estimated present genes distribution). However, we observe some actual data values. These spots therefore have a 0% EPP, or very low chance of being present. As we approach the main peak, the chance that a gene is present rapidly increases to 100%. The region where the EPP distribution increases rapidly from 0% to 100% is the region where the cutoffs are selected. This region is referred to as the *transition region*. Thus, the difference in choosing a 30% EPP versus 60% EPP cutoff is relatively small, but will result in differential assignment of a portion of your genes (mostly dependent on the degree of divergence).

We can zoom in on the transition region to observe in closer detail the signal values to which a given %EPP corresponds. In this case, a 50% EPP corresponds to about -0.14, while 0% corresponds to -0.8 and 100% corresponds to 0.1 (Figure 6).

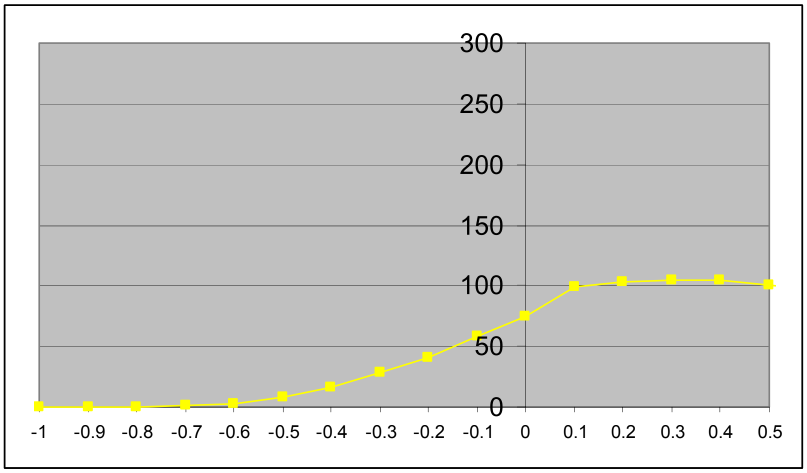


Figure 6

Development

While I do not currently have plans to improve GACK, I will entertain the possibility if demand is there. Possible areas of improvement I have considered include:

- Better approximation algorithms – t distribution, multinormal or exponential-normal mixed distributions
- Inclusion of some sort of confidence algorithm for replicated arrays
- Better systems for representation of the assignments