

Tutorial: aCGH Data Analysis With Chipster

Ilari Scheinin (firstname.lastname@gmail.com)

January 14, 2011

Abstract

This tutorial covers analysis of array comparative genomic hybridization (aCGH) data with [Chipster](#). It is divided into three sections. First one is about importing the data into Chipster, either from local files or from the CanGEM database. The second section contains a basic aCGH analysis workflow, which will also be made available directly from the Chipster user interface. The third section covers additional topics that are not possible to include into a standard, readymade workflow.

Contents

1	Importing, normalization and probe positions	2
1.1	Local files	2
1.2	CanGEM database	2
2	A basic aCGH data analysis workflow	3
2.1	Calling gains and losses	3
2.2	Identifying common regions	3
2.3	Clustering	3
2.4	Known copy number variations	5
2.5	From probes to genes	5
2.6	Enriched Gene Ontology categories	6
3	Additional analysis steps	7
3.1	Removing wavy artifacts	7
3.2	Comparisons between groups	8
3.3	Integration with expression	8
4	Workflow diagrams	10
4.1	Main aCGH tools	10
4.2	aCGH annotation tools	11
4.3	Tools for integrating aCGH and expression data	12

1 Importing, normalization and probe positions

1.1 Local files

The first step is to import the data into Chipster. Local files can be imported using the import tool as described in [this tutorial](#). For Agilent Feature Extraction files, choose ProbeName as the Identifier, and depending on the dyes used, either gMedianSignal/gBGMedianSignal for Sample/Sample BG and rMedianSignal/rBGMedianSignal for Control/Control BG, or *vice versa*. Depending on the settings of Feature Extraction, your files might contain columns for mean signals instead of medians (*e.g.* gMeanSignal), either in addition or instead of the median signals. They can also be used.

The next step is normalization, with *e.g.* the **Normalization / Agilent 2-color tool**. The default parameter values are recommended.

For all aCGH data analysis, it is crucial to know what locations in the genome the array probes hybridize to. These annotations can be downloaded using the **aCGH / Fetch probe positions from CanGEM** tool. Mappings are available for different builds of the human genome, and the list of available array platforms can be found [here](#). In Chipster these annotations are saved to columns named chromosome, start and end. For the rest of the tutorial, it is assumed that these columns are present in your data.

1.2 CanGEM database

If the starting data is stored in the [CanGEM database](#) [1], the previous three steps can be combined by using the **aCGH / Import from CanGEM** tool. Enter the accession number of the data in question, and change the normalization parameters if needed (the default values are recommended) and the genome build in case you do not want to use the latest one.

If your data is password-protected, there are two ways of accessing it. The first one is to enter your username and password into the corresponding parameters, but this will result in them being saved to any session and workflow files you create. A more secure approach is to log in on the CanGEM web site, locate the session ID on the bottom right corner of the page (the ID looks something like “ee8cbd9dcaa8284189f1582816531f46”), and copy&paste it into the session parameter in Chipster. This way Chipster can still download your data files. But after you log out (or the session times out after 24 minutes), saved sessions or workflows cannot access your private data anymore.

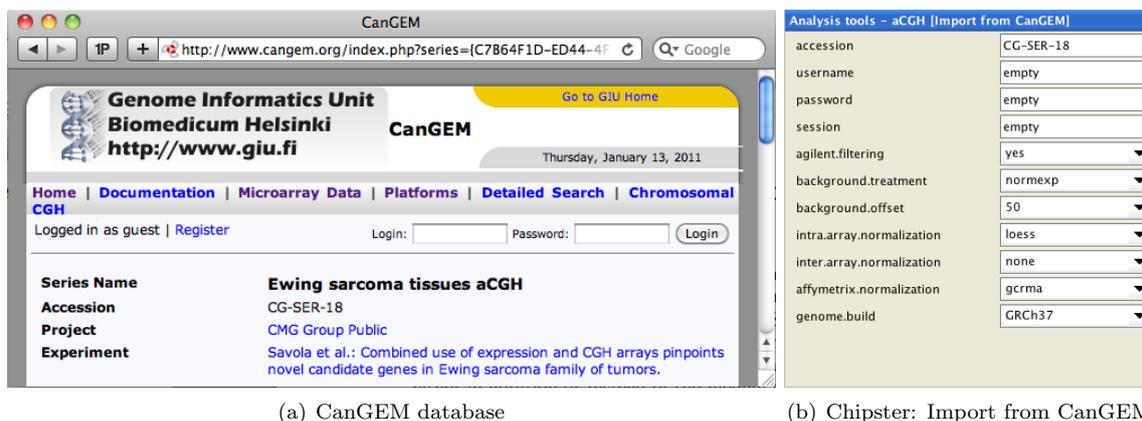


Figure 1: Importing data from the CanGEM database using the **aCGH / Import from CanGEM** tool. Note the accession number on the CanGEM web site, and enter it into Chipster.

2 A basic aCGH data analysis workflow

This section assumes a starting point of a normalized dataset that contains information for probe positions (columns chromosome, start and end). Similar quality control steps apply as with expression arrays, so they are not covered in this tutorial.

2.1 Calling gains and losses

This first step in aCGH data analysis is usually to detect copy number aberrations, *i.e.* gains and losses. Sometimes higher-level amplifications are also treated separately from gains. To do this in Chipster, use the tool **aCGH / Call copy number aberrations from aCGH data**. The parameters let you specify the number of chromosomes (usually 24 if you have sex-matched reference samples, 23 otherwise) and the number of copy number states (either 3 for loss/normal/gain, or 4 to also include amplifications).

To obtain the discrete copy number levels, the data is first segmented [2], which refers to dividing it into non-overlapping areas that are separated by breakpoints and most likely share a common copy number. Calling [3] then assigns each segment a copy number call of a loss (represented with -1), normal (0), gain (1), or, in case amplifications are separated from gains, an amplification (2). These are sometimes referred to as “hard calls”. As they are determined using a probabilistic model, each call also has an underlying probability, and these can be referred to as “soft calls”. For each probe on the microarray, there are therefore three (or four) call probabilities that add up to 100%. If the probability of (*e.g.* a loss) is over 50%, the probe is called as a loss (-1). Otherwise the call is normal (0).

The output from the tool is a huge table with large number of columns. Usually there is no need to deal with these manually, but for information’s sake they are as follows: columns labeled `chip.*` contain the original microarray log ratios, `segmented.*` contain segmented log ratios, `flag.*` contain copy number calls, and `probloss.*`, `probnorm.*`, `probgain.*` and `probamp.*` contain the probabilities for the specific calls. In addition, the frequencies of aberrations are shown in columns `loss.freq`, `gain.freq`, and if needed, `amp.freq`. In addition to the table, a summary plot is also produced and can be seen in Figure 2(a).

After the calling step, individual samples can be plotted with the **aCGH / Plot copy number profiles from called aCGH data** tool. Specify the number(s) of the sample(s), and chromosomes (0 means all chromosomes) to be plotted.

The implemented R packages for segmenting and calling are DNACopy [2] and CGHcall [3], respectively.

2.2 Identifying common regions

As aCGH data typically contains long stretches of DNA without breakpoints and a shared copy number, its dimensionality can be greatly reduced after the calling. This makes the data more manageable and also reduces the severity of multiple testing correction. In Chipster, this can be done with a tool called **aCGH / Identify common regions from called aCGH data**. These regions are what should be used for downstream analysis steps such as clustering and between-group comparisons.

The output is a condensed table containing the same columns as in the input file, and also an additional one containing the number of probes within each region. At this stage, the number of rows is usually also manageable, so that it is possible to order the table according to *e.g.* `loss.freq` or `gain.freq` to see where the most frequent aberrations are. To also include information about karyotype bands, run the tool **aCGH / Add cytogenetic bands**. In addition to the table, two plots are produced (see Figure 3).

The corresponding R package is called CGHregions [4].

2.3 Clustering

Using methods developed for expression data to cluster aCGH samples does not yield to optimal results. Therefore there is a separate tool for this purpose: **aCGH / Cluster called aCGH data**. It should be run after identifying the common regions. Otherwise meaningless, long stretches of DNA without breakpoints will have more weight on the clustering than small aberrations, as the long regions contain more probes on the array. Identifying the common regions compresses these regions into individual data points making the clustering more dependent on the actual differences between the samples.

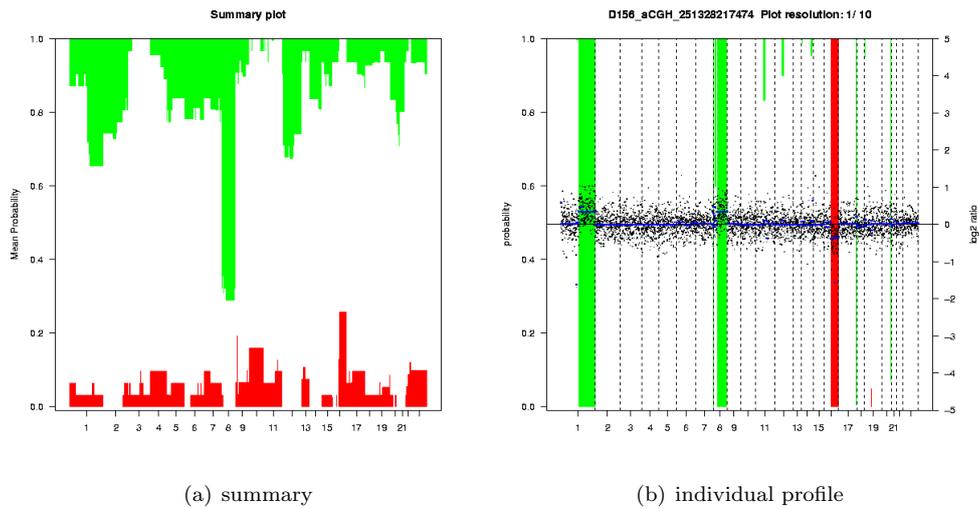


Figure 2: Plots of DNA copy number. For both plots, chromosomes are along the x-axis and call probabilities along the y-axis. The probabilities of losses are shown with red bars, and the values can be read directly from the y-axis. Probabilities of gains are shown in green, and the values can be read as '1 - the value on the y-axis'. Possible amplifications are shown with a blue tick mark on the top. a) A summary plot of all samples is generated with **aCGH / Call copy number aberrations from aCGH data** and shows the mean probabilities over all of the samples. b) A plot of on individual sample, produced with **aCGH / Plot copy number profiles from called aCGH data**. Original log ratios are shown in black and segmented log ratios in blue. This plot can also be drawn for a subset of chromosomes.

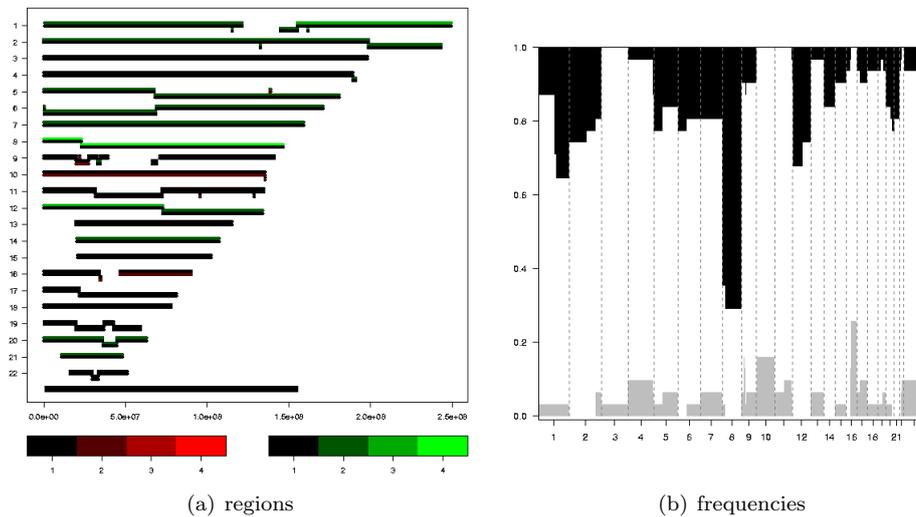


Figure 3: Plots of common regions, produced with **aCGH / Identify common regions from called aCGH data**. a) Each chromosome is shown with a horizontal bar, with bumps showing breakpoint locations. Frequencies of losses are shown in red, and gains in green. b) This plot is very similar to the summary plot in Figure 2(a), but contains aberration frequencies instead of mean call probabilities. Losses are shown in gray (values on the y-axis), and gains in black (values '1 - value on the y-axis').

Clustering can be performed both with hard or soft calls. Generally soft calls are recommended, as they not only include the hard calls, but also additional information about the reliability of these calls. The option to cluster using hard calls is provided only for situations when soft calls are not available. In case you have analyzed your data with the **aCGH / Call copy number aberrations from aCGH data** and **aCGH / Identify common regions from called aCGH data** tools, you will always have the soft calls available. Figure 4 shows a clustering with both types of calls.

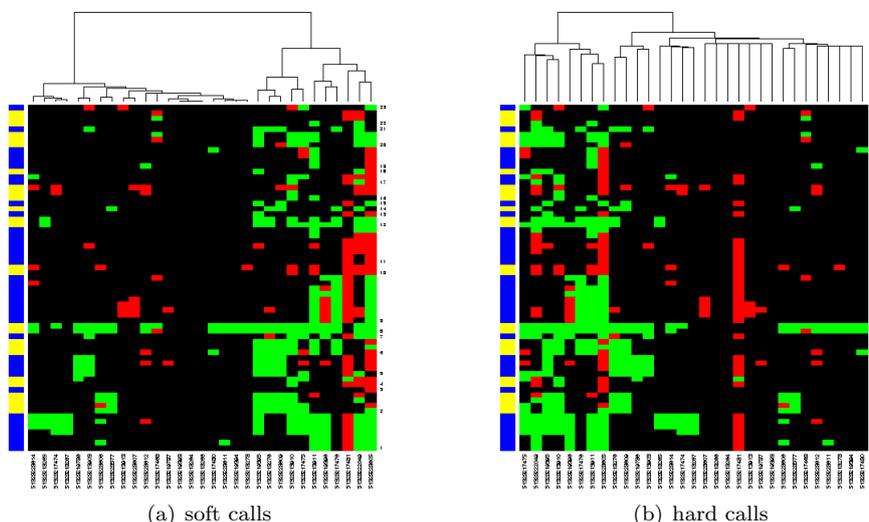


Figure 4: Clustering of samples, produced with **aCGH / Cluster called aCGH data**. Clustering using a) soft calls produces more reliable results and better shows the distances between samples than b) hard calls.

The implemented R package is WECCA [5].

2.4 Known copy number variations

The tool **aCGH / Count overlapping CNVs** downloads a list of known copy number variations (CNVs) from the Database of Genomic Variants [6] and appends two new columns to the data set: `cnv.count` and `cnv.per.Mb`. The first one is a raw count of how many entries there are in the database that overlap with the area of interest (can be *probes*, *regions* or *genes*). The latter one is calculated as follows: For each area of interest, the number of bases pairs that are within a known CNV is calculated and divided by the length of the area. Finally, the number is multiplied by 1,000,000 to yield the number of CNV base pairs per megabase of sequence.

To evaluate the distribution of the values across the entire genome, run the tool **Statistics / Calculate descriptive statistics** and specify “chips” for the parameter `calculate.descriptives.for`.

2.5 From probes to genes

In order to be able identify enriched Gene Ontology categories among gained/lost genes, we need to know the copy number of each gene. For this, we can use the **aCGH / Convert called aCGH data from probes to genes** tool, which works as follows. First, the list of human genes is downloaded from the Ensembl database [7]. Then for each gene, it is checked whether there are probes on the array that overlap with the position of the gene. If yes, these probe(s) are used to derive the copy number call for this particular gene. If no, the last probe preceding and first one tailing the gene are used. Tool parameters can be used to choose between two methods for deriving the copy number call: “majority” means that in order to call the gene *e.g.* gained, more than 50% of the probes in question have to show a gain. If “unambiguous” is chosen, the copy number of the gene is called as normal unless every one of the probes gives the same aberrant call.

2.6 Enriched Gene Ontology categories

After the aCGH data set has been converted from *probe* to *gene*-based, the tool **aCGH / GO enrichment for called aCGH genes** can be used to detect Gene Ontology categories enriched among frequently aberrated genes. The user can choose to pick only genes that are frequently lost, gained or amplified, or combine all aberrations together (default). The minimum frequency of aberrations can also be specified (default is 50%). Genes showing more frequent aberrations than the threshold are then picked as the test list, and a hypergeometric test performed to see if certain Gene Ontology categories are enriched. The entire gene list is used as the reference. It should therefore be an unfiltered list, *i.e.* the direct output from **aCGH / Convert called aCGH data from probes to genes**.

The rest of the parameters are the same as for the corresponding expression tools.

3 Additional analysis steps

3.1 Removing wavy artifacts

aCGH profiles typically contain a technical, wavy artifact [8]. When analyzing cancer samples, it is possible to remove the effect of these waves by using clinical genetics samples as calibration data, as they are not expected to contain large aberrations. Preferably the calibration data should be measured with the same array platform as the data to be analyzed. Smoothing the waves generally leads to more accurate calling and improved reliability. The effects can be seen in Figure 5.

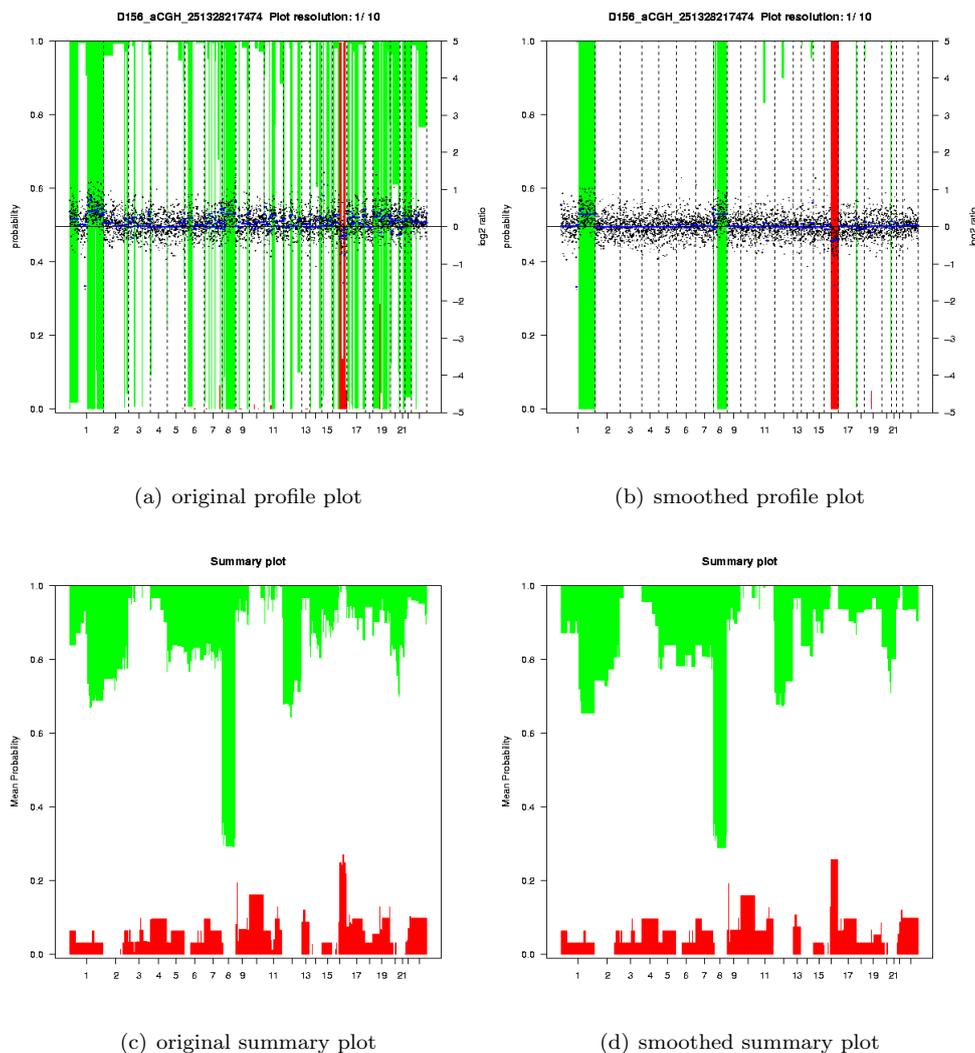


Figure 5: The effect of dewaving with aCGH / Smooth waves from normalized aCGH data. Profile plots of an individual sample are shown both for the a) original and b) smoothed data. Dewaving generally results in more confident calling (more probabilities close to 0% or 100%, instead of being around 50%). Summary plots are also shown for the c) original and d) smoothed data sets. The effect of dewaving can be observed as more clearly defined aberration boundaries.

One important note about using the tool is that while selecting the two normalized data sets, first click on the cancer data, then on the calibration set. Otherwise Chipster will try to do it the wrong way. The name of the implemented R package is NoWaves [9].

3.2 Comparisons between groups

If your data set contains two or more groups, statistical testing for between-group differences can be performed using the **aCGH / Group tests for called aCGH data** tool. It should normally be run on *regions* (*i.e.* results from the **aCGH / Identify common regions from called aCGH data** tool), but can also be run on *probe* or *gene*-based data as well, although running times are likely to be prohibitively long. A test statistic (either Chi-square, Wilcoxon or Kruskal-Wallis) is calculated for each region. As the distribution of the test statistic might be really skewed, significance is evaluated with a permutation-based approach instead of simple multiple testing correction. The group labels for individual arrays are randomly sampled, and the test statistics calculated for each repetition. Finally, a false discovery rate (FDR) is calculated for each region based on how frequently test statistics as extreme as the calculated one were observed during the permutations. The number of permutations to run can be set in the tool parameters. The larger the number, the longer the execution takes. For final analysis, at least 10,000 permutations are recommended.

The implemented R package is CGHtest, which is an updated version of CGHMultiArray [10].

3.3 Integration with expression

Integrating aCGH and expression data together is multi-step process involving four separate tools. The relationships between these tools are outlined in Figure 4.3. The first step is to run **aCGH / Match copy number and expression probes**, which takes two input files: the output of **aCGH / Call copy number aberrations from aCGH data** and a normalized and filtered expression data set. To be able to pair the samples of the two data sets, the accompanying phenodata tables must have columns that contain common identifiers unambiguously identifying the pairs. When importing data from CanGEM, this is usually a column called Sample. The output is a table of matched microarray probes, and a plot showing heatmaps of both data sets (Figure 6(a)). This file can also be used to plot profiles of individual samples with **aCGH / Plot profiles of matched copy number and expression**. Parameters allow the user to specify sample(s) and chromosome(s) to be plotted. The produced image (see Figure 6(b)) contains an aCGH profile plot similar to Figure 2(b) and another plot showing expression levels.

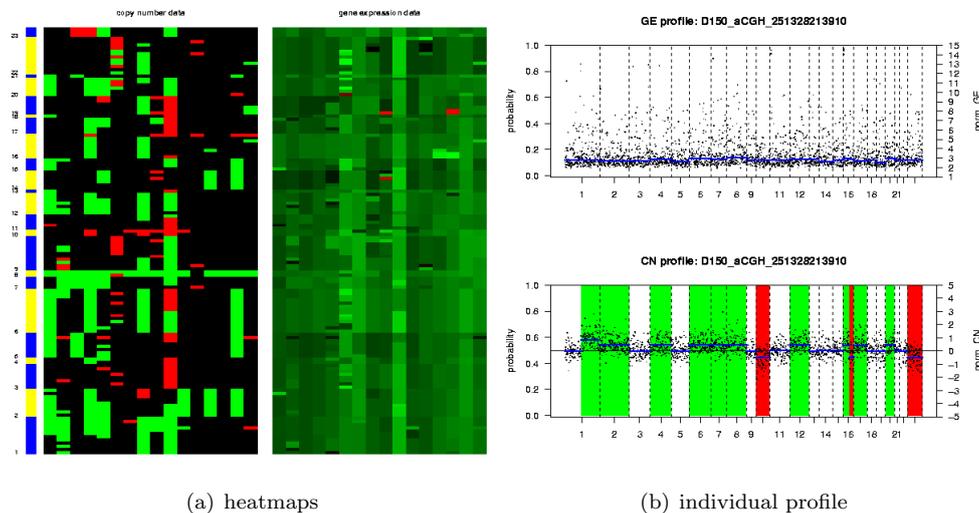


Figure 6: a) Heatmaps of matched aCGH and expression data, generated with **aCGH / Match copy number and expression probes**. The copy number data is on the left and samples are shown in the same order as in the expression heatmap on the right. Chromosomes are shown along the y-axis. b) aCGH and expression profiles of an individual sample, produced with **aCGH / Plot profiles of matched copy number and expression**. aCGH data is shown on the bottom (for interpretation see Figure 2(b)), and expression profile on top. Expression levels of individual genes are shown with black dots, and blue lines show the mean expression levels of genes within *regions* defined by the aCGH data.

To test the statistical significance of copy number changes on expression levels, run the **aCGH / Test for copy-number-induced expression changes** tool. It divides samples into two groups for each expression probe based on the aberration profile for that particular probe. The comparison is either between ‘loss *vs.* no-loss (normals, gains and amplifications)’ or between ‘no-gain (losses, normals) *vs.* gain (gains and amplifications)’. Statistical testing is performed using a permutation test, and the tool parameters let the user specify how many permutations to run. 10,000 are recommended for final analysis, but take a long time. The resulting p-values can be found in the adj.p column of the resulting output table. Also contained within this file is a column labeled as gene.id, which contains IDs that are needed to plot visual representations of individual genes with the **aCGH / Plot copy-number-induced gene expression** tool.

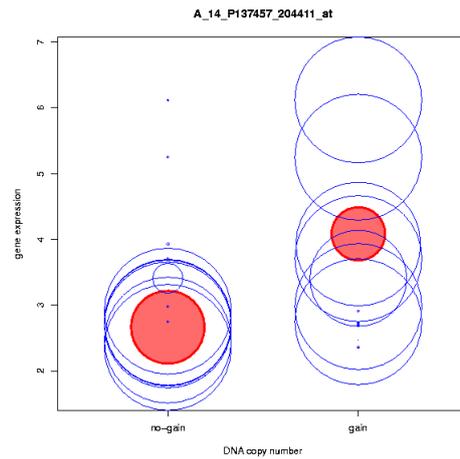


Figure 7: A plot with matched copy number and expression data. The title shows the names of the aCGH (A_14_P137457) and expression probes (204411_at). Based on the observed aberration frequencies, the test has been performed by comparing a “no-gain” group of samples (losses and normals) *vs.* a “gain” group of samples (gains and amplifications), as shown by the labels at the bottom. Expression levels of individual samples are shown with blue circles and the scale is along the y-axis. The radius of the circle represents the probability of the corresponding call. Each sample is therefore plotted on both columns, but using circles with different radii. Red circles represent mean values. This particular case had an adjusted p-value of 0.27.

The integration of copy number and expression data sets is implemented with the intCNGEan R package [11].

4 Workflow diagrams

4.1 Main aCGH tools

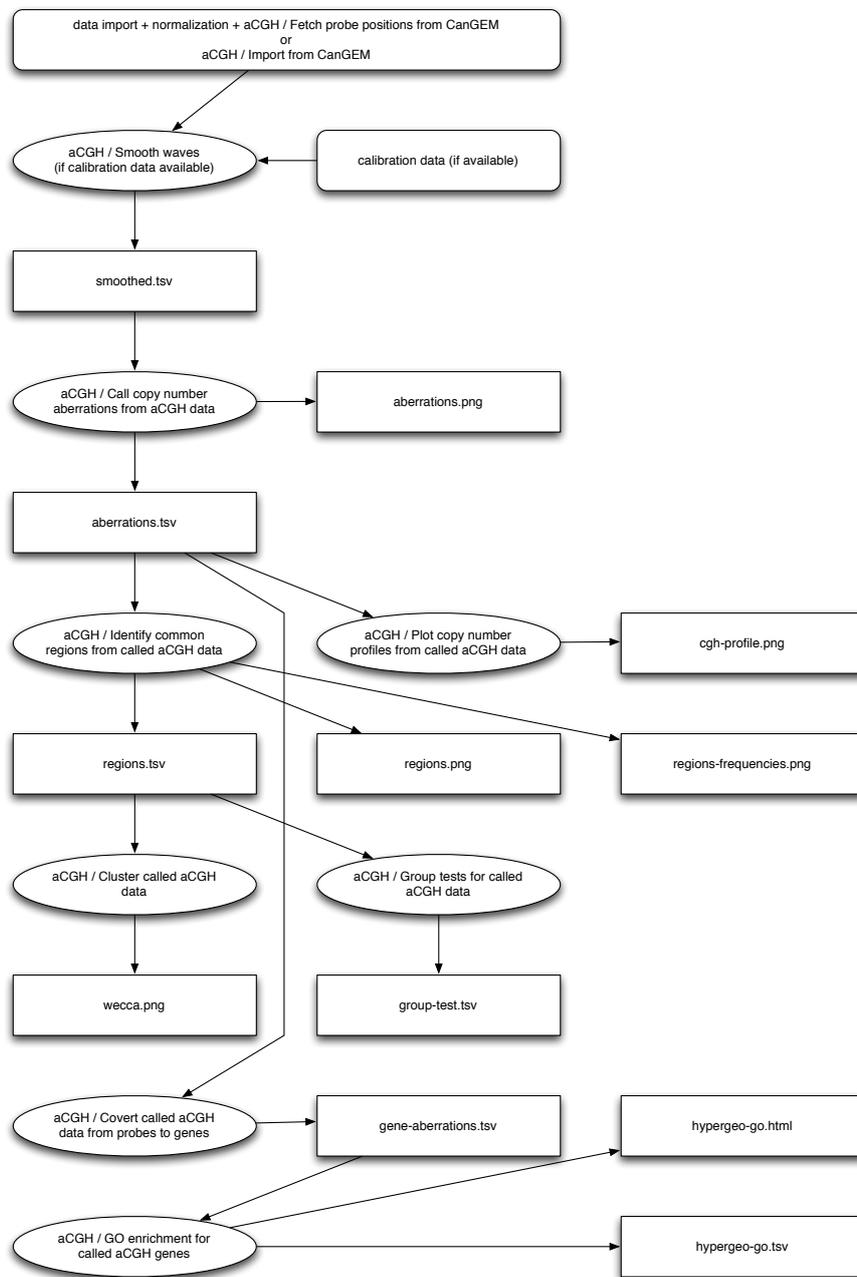


Figure 8: A diagram showing the order in which the aCGH tools should be executed.

4.2 aCGH annotation tools

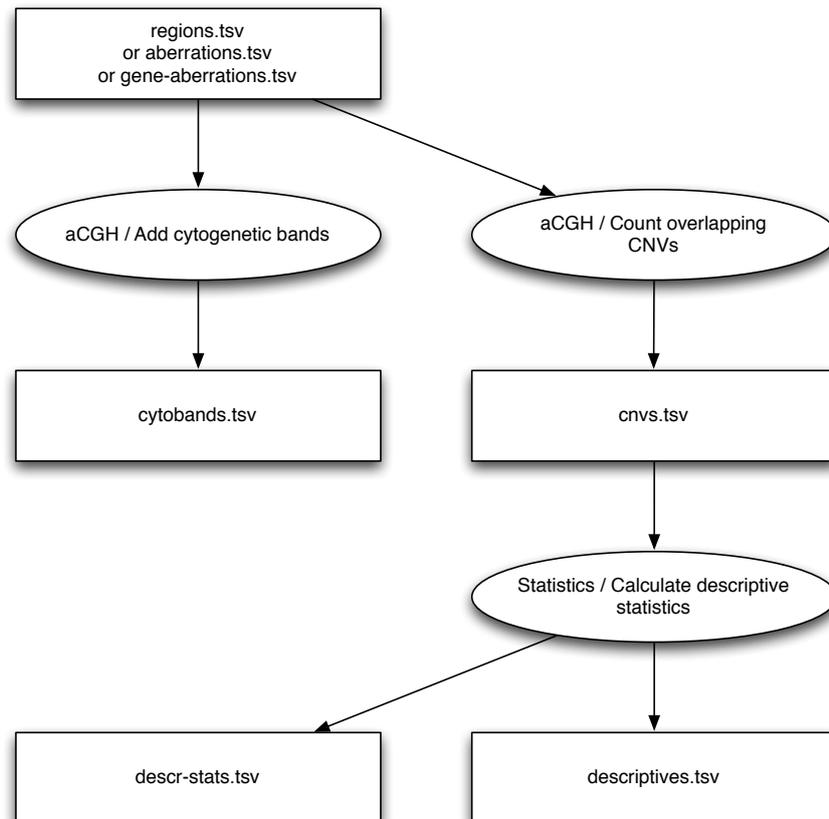


Figure 9: A diagram showing a typical use case of aCGH annotation tools.

4.3 Tools for integrating aCGH and expression data

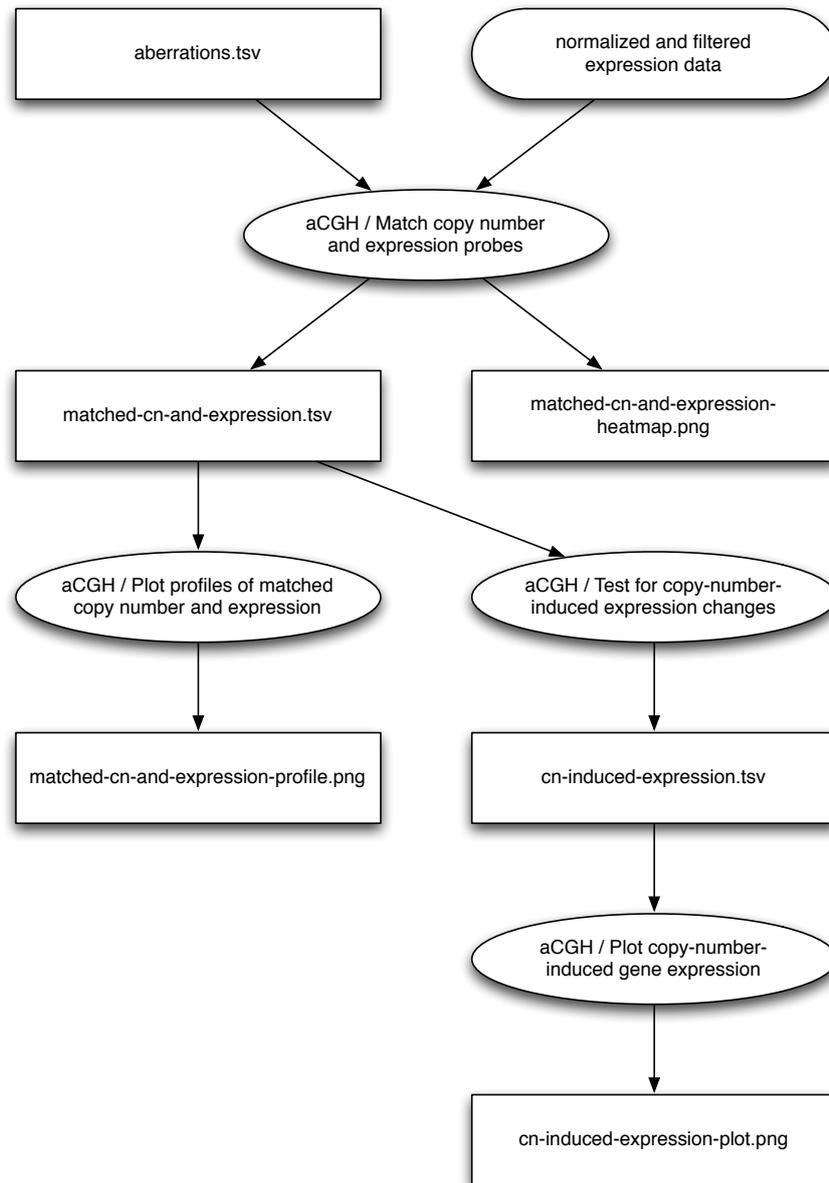


Figure 10: A diagram showing how the different tools involved in integrating aCGH and expression data are related to each other.

References

- [1] I. Scheinin, S. Myllykangas, I. Borze, T. Bohling, S. Knuutila, and J. Saharinen. CanGEM: mining gene copy number changes in cancer. *Nucleic Acids Res*, 36(Database issue):D830–D835, 2008.
- [2] E. S. Venkatraman and A. B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, 23(6):657–663, Mar 2007.
- [3] M. A. van de Wiel, K. I. Kim, S. J. Vosse, W. N. van Wieringen, S. M. Wilting, and B. Ylstra. CGHcall: calling aberrations for array CGH tumor profiles. *Bioinformatics*, 23(7):892–894, 2007.
- [4] M. A. van de Wiel and W. N. van Wieringen. CGHregions: dimension reduction for array CGH data with minimal information loss. *Cancer Informatics*, 3:55–63, 2007.
- [5] W. N. van Wieringen, M. A. van de Wiel, and B. Ylstra. Weighted clustering of called array CGH data. *Biostatistics*, 9(3):484–500, Jul 2008.
- [6] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee. Detection of large-scale variation in the human genome. *Nat Genet*, 36(9):949–951, 2004.
- [7] P. Flicek, B. L. Aken, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, G. Coates, S. Fairley, S. Fitzgerald, J. Fernandez-Banet, L. Gordon, S. Gräf, S. Haider, M. Hammond, K. Howe, A. Jenkinson, N. Johnson, A. Kähäri, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, G. Koscielny, E. Kulesha, D. Lawson, I. Longden, T. Massingham, W. McLaren, K. Megy, B. Overduin, B. Pritchard, D. Rios, M. Ruffier, M. Schuster, G. Slater, D. Smedley, G. Spudich, Y. A. Tang, S. Trevanion, A. Vilella, J. Vogel, S. White, S. P. Wilder, A. Zadissa, E. Birney, F. Cunningham, I. Dunham, R. Durbin, X. M. Fernández-Suarez, J. Herrero, T. J. P. Hubbard, A. Parker, G. Proctor, J. Smith, and S. M. J. Searle. Ensembl’s 10th year. *Nucleic Acids Res*, 38(Database issue):D557–62, Jan 2010.
- [8] J. C. Marioni, N. P. Thorne, A. Valsesia, T. Fitzgerald, R. Redon, H. Fiegler, T. D. Andrews, B. E. Stranger, A. G. Lynch, E. T. Dermitzakis, N. P. Carter, S. Tavaré, and M. E. Hurles. Breaking the waves: improved detection of copy number variation from microarray-based comparative genomic hybridization. *Genome Biol*, 8(10):R228, 2007.
- [9] M. A. van de Wiel, R. Brosens, P. H. C. Eilers, C. Kumps, G. A. Meijer, B. Menten, E. Sistermans, F. Speleman, M. E. Timmerman, and B. Ylstra. Smoothing waves in array CGH tumor profiles. *Bioinformatics*, 25(9):1099–1104, May 2009.
- [10] M. A. van de Wiel, S. J. Smeets, R. H. Brakenhoff, and B. Ylstra. CGHMultiArray: exact p-values for multi-array comparative genomic hybridization data. *Bioinformatics*, 21(14):3193–3194, 2005.
- [11] W. N. van Wieringen and M. A. van de Wiel. Nonparametric testing for DNA copy number induced differential mRNA gene expression. *Biometrics*, 65(1):19–29, Mar 2009.