CGH-Plotter

User Manual

Contents

1 Introduction 2

2 Installation 2

2.1 Installation Instructions 2

3 Instructions 4

3.1 User Interface Pages 4

3.1.1 Main Page 4

3.1.2 Create Data Struct 6

3.1.3 Find Amplicons 10

3.1.4 Plot Data 12

4 Methods 22

4.1 Filtering 22

4.2 *k*-means Clustering 24

4.3 Dynamic Programming 25

5 Summary 26

1 Introduction

Copy number changes, such as deletions and amplifications, are common aberrations in cancer and are known to involve genes that play a crucial role in the development and progression of the malignant disease [5]. The copy number changes span usually large regions of the genome and therefore influence multiple genes at the same time. Comparative genomic hybridization (CGH) on DNA microarray allows simultaneous monitoring of copy numbers of thousands of genes throughout the genome [6], [7].

CGH-Plotter is a versatile software that allows the user to plot CGH copy number data as a function of the position of the genes along the human genome, and to rapidly determine the exact locations of copy number changes, such as amplicons and deletions.

In this user manual we explain in details:

1. How to install CGH-Plotter,
2. How to use CGH-Plotter,
3. How to store and analyze the results,
4. What are the assumptions behind the analysis.

We also provide several examples on the use of CGH-Plotter.

2 Installation

CGH-Plotter requires Matlab 6.1 or higher in order to operate. Accordingly, all data must be in Matlab (*.mat) format or in tab delimited text (*.txt) format.

2.1 Installation Instructions

Archive 'CGH-Plotter.zip' consists of five folders: *CGH-Plotter*, *gui*, *ampli_math*, *data_structs* and *ampli_data*.

- Main folder *CGH-Plotter* contains the following folders and files:
  - *gui*

  - *ampli_math*
  - 'CGH_Plotter.m' and 'CGH_Plotter.fig'.

- Folder *gui* (Graphical User Interface) includes functions and corresponding figures:

  - 'create_struct.m', 'create_struct.fig'
  - 'amplikoni.m', 'amplikoni.fig'
  - 'plot_data.m', 'plot_data.fig'
  - 'end_all.m', 'end_all.fig'

- Folder *ampli_math* includes all mathematical functions used in CGH-Plotter:

  - 'combined.m'
  - 'compute_kmean.m'
  - 'cumulative.m'
  - 'define_amplicons.m'
  - 'dynamic_prog.m'
  - 'filter_data.m'
  - 'handle_NaNs.m'
  - 'kmean.m'
  - 'transform_data.m'
  - 'writeresults.m'

- Folder *data_structs* can be located arbitrary. It is meant for storage of data structs of CGH-data and is initially empty.

- Folder *ampli_data* is intended for storing the analyzed data and it can be located arbitrary. Folder *ampli_data* is initially empty.

The diagram of folders in CGH-Plotter is illustrated in Figure 1.

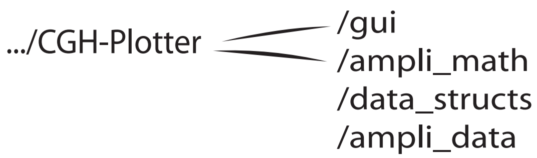


Figure 1: Folders of CGH-Plotter. Folders gui and ampli_math are subfolders of CGH-Plotter, data_structs and ampli_data can be located arbitrary.

3 Instructions

Basically CGH-Plotter functions as follows. First, CGH-Plotter filters the data using median or mean filter with window size that has been input. Secondly, the filtered data are clustered using the k-means clustering algorithm. The purpose of the k-means clustering is to find the maximum number of amplicons/deletions at each chromosome. This number is required by the last phase, dynamic programming, which actually estimates the amplicons and deletions. CGH-Plotter saves the result file, which consists of the original data, filtered data, probable amplicons and deletions, indices to the changes of amplicons and deletions of the CGH-data, names of the samples, cumulative basepairs and genomic indices.

To be more precise, CGH-Plotter consists of five phases:

1. CGH-Plotter creates a data struct of separate data files that the user has specified,

2. CGH-Plotter reads the data struct,

3. CGH-Plotter analyzes the data struct,

4. CGH-Plotter stores the analyzed data,

5. CGH-Plotter plots the data.

In this section a more detailed explanation is given for each of these phases.

3.1 User Interface Pages

3.1.1 Main Page

CGH-Plotter is started with command 'CGH_Plotter', providing that current directory in Matlab is CGH-Plotter. Main page is opened and CGH-Plotter is available for use as illustrated in Figure 2.

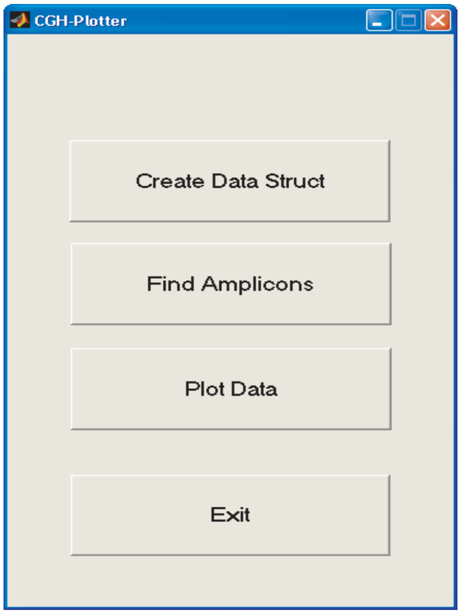


Figure 2: Main page of CGH-Plotter.

The main page contains four buttons: ’Create Data Struct’, ’Find Amplicons’, ’Plot Data’ and ’Exit’. First, the data struct should be constructed in the page ’Create Data Struct’, if it is not done already. After the data struct is created and stored, the analysis part is executed at the page ’Find Amplicons’. Finally, in the page ’Plot Data’ the analyzed data may be plotted and results of the analysis saved in ASCII file. Button ’Exit’ ends session and returns the user to the Matlab workspace. The idea of the blocks in CGH-Plotter is illustrated in Figure 3.

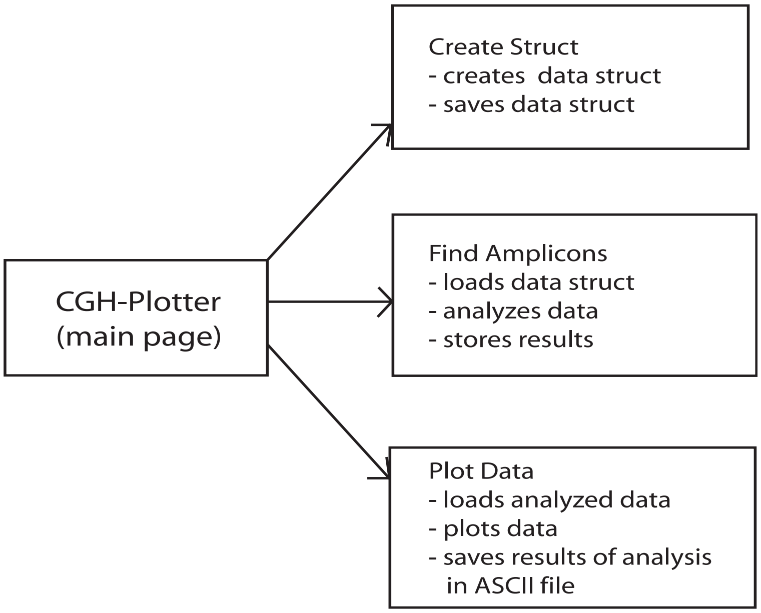


Figure 3: Main tasks of CGH-Plotter blocks.

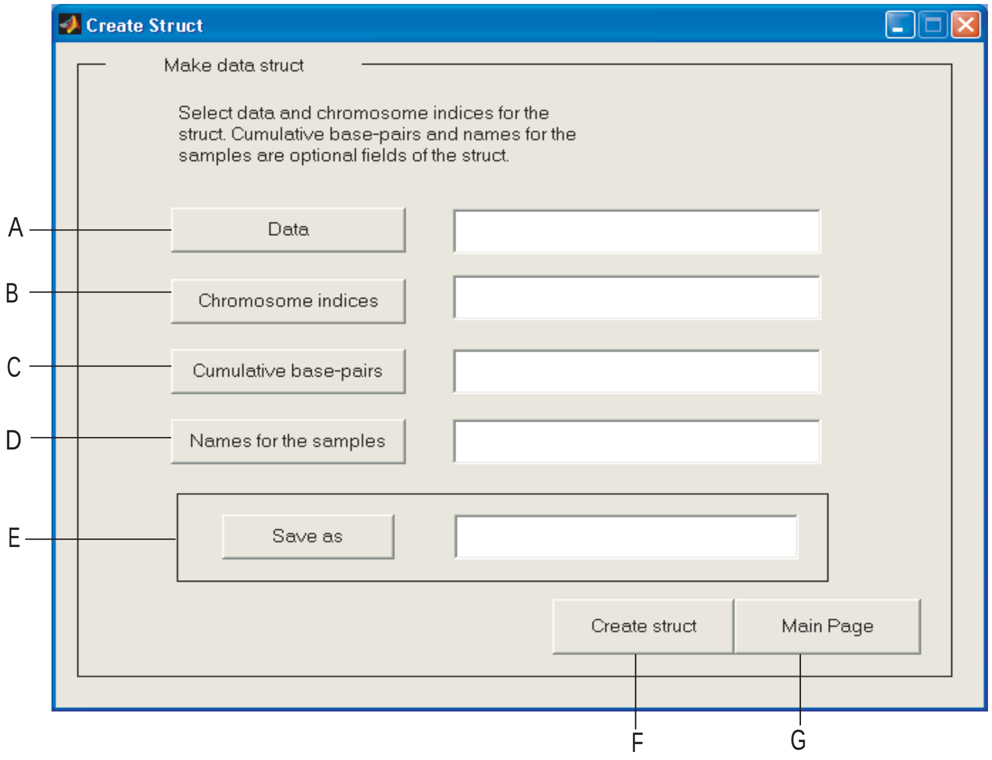
3.1.2 Create Data Struct

In the page 'Create Data Struct' one is able to create a data struct that consists of the CGH-data and essential indices. It is assumed throughout CGH-Plotter that the data contain fields given in this section. All the data has to be either

1. in Matlab (*.mat) format or
2. in tab delimited text (*.txt) format.

Examples of the files (both formats) are in folder $\backslash CGH{-}Plotter\backslash data\_structs$.

A) Data-button (obligatory)
This button enables loading of the data file. The data file is assumed to be $m \times n$ matrix, where m is the number of genes and n is the number of the samples. Furthermore, it is assumed that the genes are arranged according to their genomic order from p-telomere of chromosome 1 to q-telomere of the Y chromosome. This order of genes is referred to as genomic index. Missing



Select data and chromosome indices for the struct. Cumulative base-pairs and names for the samples are optional fields of the struct.

Figure 4: Create Data Struct -window.

values should be replaced with NaNs (Not-a-Numbers). If the input file format is *.txt, CGH-Plotter will automatically replace the missing values with NaNs. Finally, the data should not be transformed e.g. with log-transform prior to CGH-Plotter. After selecting a data matrix, the name of the selected data appears to the text box next to data button.

B) Chromosome indices -button (obligatory)
As it is essential to know where each chromosome begins, the starting points of the chromosomes as indices to the data matrix needs to be specified. Chromosome indices is a 24×1 matrix. First 22 indices are the starting points of chromosomes 1-22, 23:rd is the starting point of chromosome X and 24:th of the chromosome Y. Also the chromosome indices can be in *.txt or in *.mat format. An example of chromosome indices matrix in *.mat format is shown below: [3]

$$Chromosome\_indices = \begin{bmatrix} 1 \\ 1338 \\ 2121 \\ 2829 \\ 3292 \\ 3851 \\ 4548 \\ 5115 \\ 5480 \\ 5924 \\ 6408 \\ 7047 \\ 7729 \\ 7941 \\ 8353 \\ 8701 \\ 9193 \\ 9812 \\ 9994 \\ 10695 \\ 11047 \\ 11198 \\ 11529 \\ 11980 \end{bmatrix}$$

CGH-Plotter adds the last index of the chromosome 'Y' to chromosome indices matrix. Therefore the chromosome indices is a 25 × 1 matrix during the analysis.

C) Base-pairs -button (optional)
It is illustrative to plot the CGH ratios as a function of their actual location along the genome in base-pairs. Therefore we have included the possibility to define cumulative base-pairs for the data. Also the Base-pairs file can be in *.mat or *.txt format. Base-pairs file is an $m \times 1$ vector, where m is the number of genes. If base-pairs are not specified, CGH-Plotter will use only the order of the genes along the genome, i.e. the genomic indices.

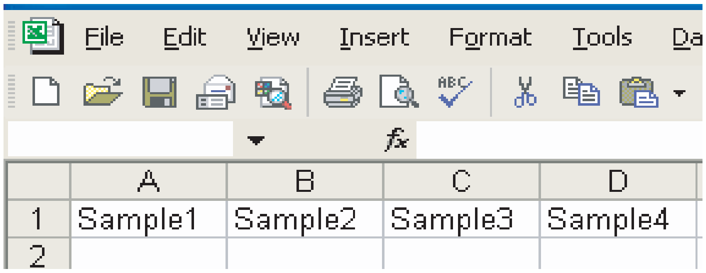
D) Names of the samples -button (optional)
The names of the samples can be specified. If names are given in *.mat format they should be given in $n \times 1$ string vector, where n is the number of samples. Names cannot include space characters or special characters that Matlab considers as mathematical symbols, like ’-’ or ’+’. For example, if the number of samples is three, the cell struct can be made and saved in Matlab as follows.

```
>>Names = [{’BT474’}; {’MCF7’}; {’ZR7530’}]

>>save Names Names
```

If names for the samples are not defined, CGH-Plotter refers to first sample as ’sample1’, second sample as ’sample2’ etc.

Furthermore if the names are defined in *.txt file, they must be given in one row and each in own column as shown in figure 5.



	A	B	C	D
1	Sample1	Sample2	Sample3	Sample4
2				

Figure 5: Names of the samples in *.txt file.

E) Save as -button
One must give a name for the data struct and select the folder where it will be saved. Folder *data_structs* is meant for this purpose, but it is not obligatory to save data structs there.

F) Create -button
CGH-Plotter creates a data struct. When the struct is created a message box with text ’Ready’ pops out.

G) Main page -button

Main page -button returns one to the main page.

Data struct can also be created manually. However, the struct must have the following fields:

- data_struct.data (CGH-data, size $m \times n$),
- data_struct.chromo (Indices to chromosomes, size 25×1),
- data_struct.basepair (Cumulative base-pairs, size $m \times 1$),
- data_struct.samples (Names of the samples, size $n \times 1$).

3.1.3 Find Amplicons

Phase 'Find Amplicons' involves several components. The aim of this phase is first to find amplicons or deletions and then create a result file for plotting.

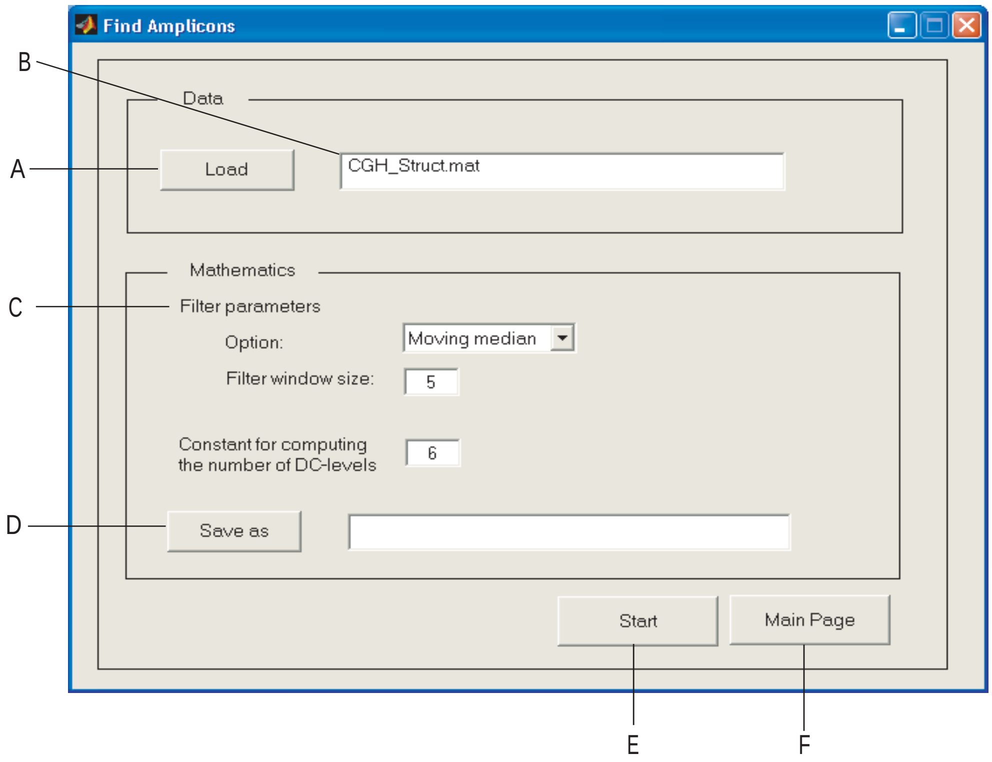
A) Load data -button
This button enables loading of the data struct made in phase 'Create Data Struct'.

B) Selected data -text box
When the data have been selected, the name of the data file can be seen in the text box next to 'Load data' button.

C) Filter parameters

- It is possible to specify the type of the filter, possible options are 'Move median' and 'Move average'. By default CGH-Plotter uses 'Move median' filter.
- Also the window size for filtering the data may be defined. Default window size is five. Window size is dependent on the amount of noise in the data. When the amount of the noise in the data is small, it is enough to have small window size (e.g. 1-3). However, if data are very noisy, window size should also be quite large (> 5).

D) Constant for computing the number of changes.
One may specify the constant that is used when the number of changes is



CGH_Struct.mat

Option: Moving median

Filter window size: 5

Constant for computing the number of DC-levels 6

Figure 6: Find Amplicons -window.

computed. Default constant is six. The procedure how to compute the number of the changes along with some guidelines is given in Section 4.2.

E) Save As -button
Before starting the analysis, one has to specify the name for the data struct to be analyzed. Save As -button opens a Save As -dialog and the name and the location for the result file may be selected. It is recommended that result files are stored in the folder *ampli_data*.

F) Start -button
After providing all required information the analysis may be started by selecting the 'Start' button. Analysis of the data takes few minutes. For example, analysis of CGH ratios of 11994 genes from 14 samples with Intel Pentium IIII/2.4 GHz took approximately 5 minutes. When CGH-Plotter is ready, a message box appears notifying that the data set has been successfully analyzed and results of the analysis are saved.

G) Main Page -button
By pushing 'Main Page' button one can return to the main page.

3.1.4 Plot Data

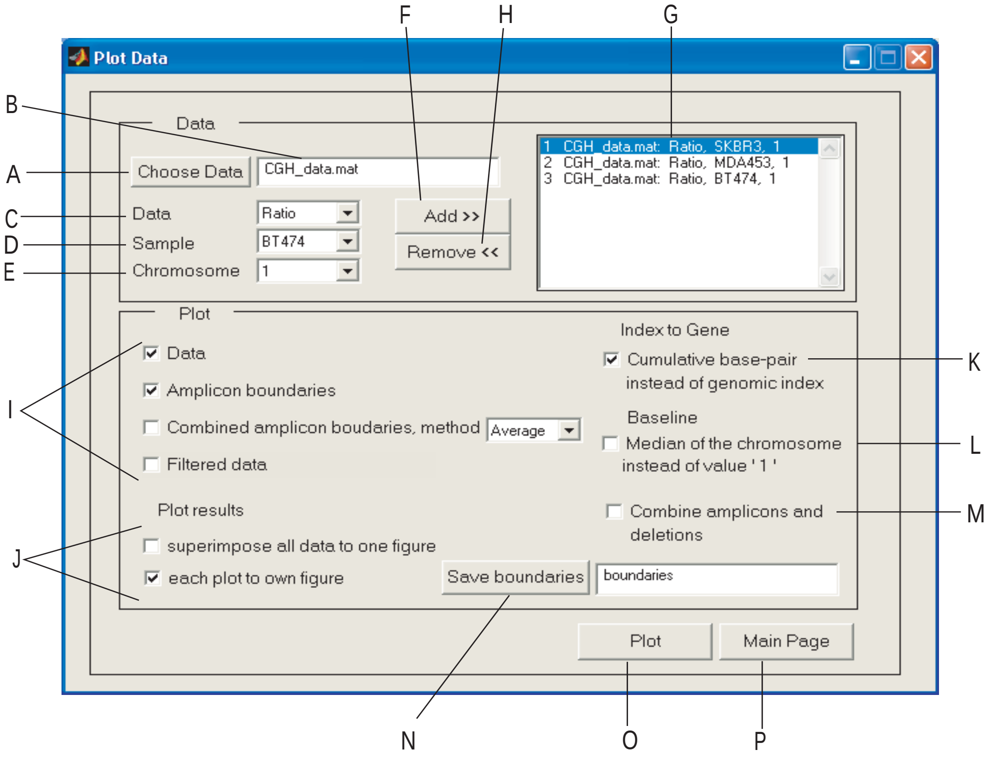
In 'Plot Data' phase it is possible to compare the data and results from dynamic programming. One may choose the properties of the created data set to be illustrated. It is possible to plot the CGH-data as ratios or log-transformed ratios, and to plot amplicon boundaries from an individual sample or combined amplicon boundaries from a group of samples.

One may plot the CGH-data, filtered data, or amplicon boundaries either from one chromosome or across all chromosomes. It is also possible to plot results from several samples at the same time. Thus one may choose whether the results are illustrated in one figure or in multiple figures. By default CGH-Plotter uses genomic indices to plot the data but one may also select to use cumulative basepairs.

A) Choose data -button
By pushing choose data -button one can select the data to be illustrated. Result file has to be constructed in the 'Find Amplicons' phase, and consist of seven fields:

- 'data': CGH-data,
- 'datafilt': Filtered data,
- 'dp': Amplicon boundaries computed with dynamic programming,
- 'tu': Indices to the changes of amplicon boundaries,
- 'chromo': Indices to chromosome starts,
- 'basepair': Cumulative base-pairs,
- 'samples': Names of the samples.

Only one data set can be illustrated at a time but it is possible to observe several properties of the data simultaneously.



Plot Data

Data

CGH_data.mat

Data: Ratio

Sample: BT474

Chromosome: 1

1 CGH_data.mat: Ratio, SKBR3, 1
2 CGH_data.mat: Ratio, MDA453, 1
3 CGH_data.mat: Ratio, BT474, 1

Plot

☑ Data

☑ Amplicon boundaries

□ Combined amplicon boudaries, method Average

□ Filtered data

Plot results

□ superimpose all data to one figure

☑ each plot to own figure

boundaries

Index to Gene

☑ Cumulative base-pair instead of genomic index

Baseline

□ Median of the chromosome instead of value ' 1 '

□ Combine amplicons and deletions

Figure 7: Plot Data -window.

B) Selected data -text box
The name of the selected data file is seen in textbox: 'Selected data'.

C) Data type
The CGH-data can be plotted either as log-transformed or as ratios. If the the data is plotted as log-transformed, CGH-Plotter adds '1' to the natural logarithm value in order to move the baseline to around ratio of one. In every case amplicon/deletion boundaries and filtered data are seen as ratios.

D) Samples
One may choose which CGH-data sample he wants to plot. If the last option 'All' is selected, CGH-Plotter adds the selected chromosome of each sample to the data listbox.

E) Chromosome
In CGH-Plotter one needs to select either the chromosome that he wants to illustrate or the option 'All' when the ratios of the sample will be plotted genome-wide.

F) Add -button
After above mentioned attributes are selected 'Add -button' will take the facts of the data to the listbox on the right. Data must always be exported to the data listbox, because CGH-Plotter handles only the data in listbox.

G) Data listbox
In the data listbox one can see the part or parts of the data that CGH-Plotter is about to plot. Parts of the data are written in the form: *'Data name: Data type, Sample, Chromosome'*. It is possible to select several parts of the data, but the number of genes must be same for every part.

H) Remove -button
'Remove' button removes selected data from the data listbox. First one has to select the data that is wanted to be removed.

I) Plot
One can select the properties to be plotted.

- If 'Data' is selected, CGH-Plotter plots original CGH-data.

- If 'Amplicon boundaries' is selected, CGH-Plotter will plot the amplicon/deletion boundaries that are computed by the dynamic programming algorithm.

- If 'Combined amplicon boundaries' is selected, CGH-Plotter will plot combined amplicon boundaries from selected samples.

- The method for computing the combined amplicon boundaries can be selected. Possible choices are average, median, maximum, and minimum. By default CGH-Plotter uses average.

- If 'Filtered Data' is selected, CGH-Plotter will plot filtered data that are computed by the filtering algorithm. The window size and the type of the filter were determined in the phase 'Find Amplicons'.

J) Show results
One can select how he wants CGH-Plotter to present the data.

- If 'superimpose all data to one figure' is selected CGH-Plotter will plot all selected data to the same figure. Each sample, filtered data of the sample and amplicon boundaries of the sample have the same color, samples are illustrated with points, filtered data and amplicon boundaries with lines. Combined amplicon boundaries are seen as thick black line (Figure 8).

- If one selects 'each plot to own figure', CGH-Plotter will illustrate every sample individually (Figures 9 and 11). CGH-Plotter plots CGH-data with blue line and amplicon boundaries with red. If 'Filtered data' is selected CGH-Plotter will plot filtered data of the sample with green line and if 'Combined amplicon boundaries' is selected CGH-Plotter will plot combined boundaries with black line.

K) Index to a gene
One may select whether he wants to see cumulative base-pairs in the x-axis instead of genomic indices.

L) Baseline
One may select whether he wants CGH-Plotter to use median of each chromosome as baseline of the chromosome. By default baseline is value '1'.

M) One may select to define adjoining amplicons (or deletions) as one amplicon (deletion) in the resulted boundary-file.

N) Save Boundaries -button
This button allows one to specify a name for the boundary file and select the folder where he wants to save it. CGH-Plotter creates a tabular separated ASCII file as illustrated in figure 13. If the name is not specified, the results are not saved. By default CGH-Plotter will save the amplicons with height over 1.2 and deletions with height smaller than 0.95. If needed it is really straightforward to change these limits in the beginning of the function DefineAmplicons.m. If the file where to one is about to save the results already exists CGH-Plotter will write the results after the existing text.

O) Plot -button
CGH-Plotter plots only the data that are seen in the data -list box and uses properties that have been specified. CGH-Plotter shows a message box that gives genomic indices to the amplicons. (Name of the samples: indices of the boundaries). Amplicon boundaries -message is modal and it will disappear permanently after pushing the OK-button.

P) Main Page -button
'Main Page' button takes one back to the main page.

A capture of the typical plotting figure is provided in Figure 8, which illustrates the ratios from chromosome 20 across five samples. It is also possible to explore only one of the samples by illustrating it separately as shown in Figure 9. Amplicon/deletion boundaries of the samples are listed in Figure 12, while Figure 13 illustrates the created ASCII file that reveals the properties of each amplicon and deletion.

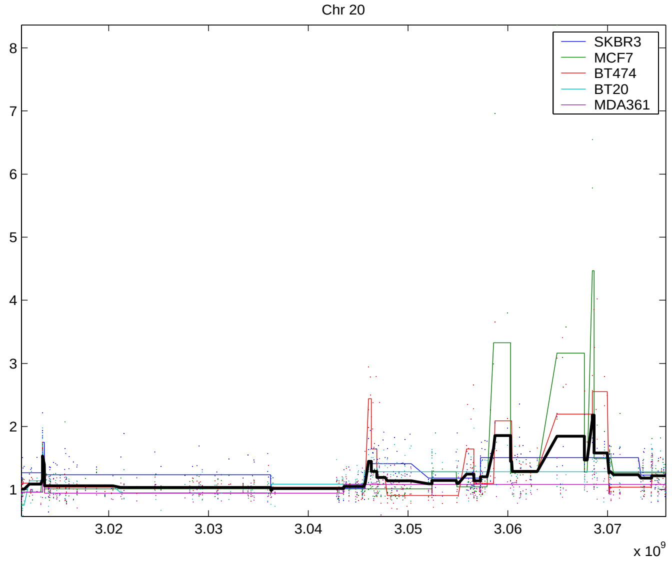


Figure 8: Ratios from five samples (chromosome 20) illustrated in one figure. Amplicon boundaries are seen with the same color as the corresponding sample. Combined amplicon boundaries are colored black. Cumulative basepairs are in the x-axis.

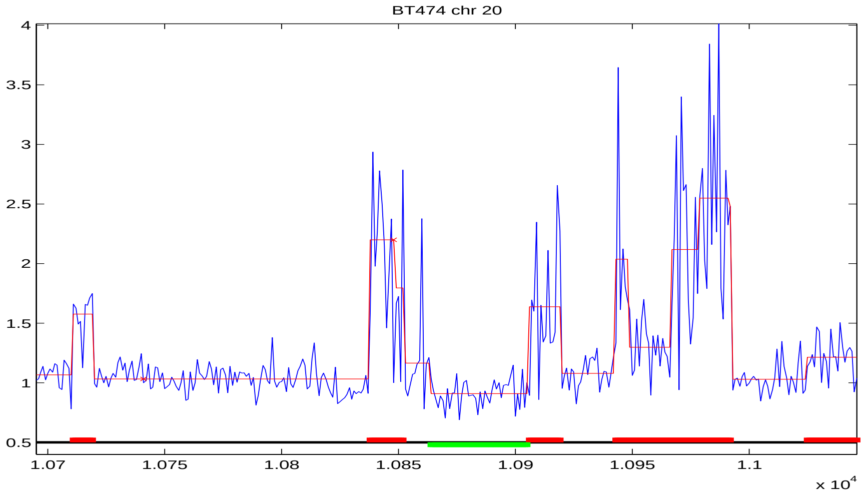


Figure 9: Chromosome 20 of the sample ’BT474’. CGH-Plotter has now plotted each of selected data into different figures using genomic index. CGH-data is blue line, amplicon boundaries red line. NaN values of original data are now marked with crosses. Underneath of the data is a bar where the amplicons and deletions of the data are marked with red and green bars.

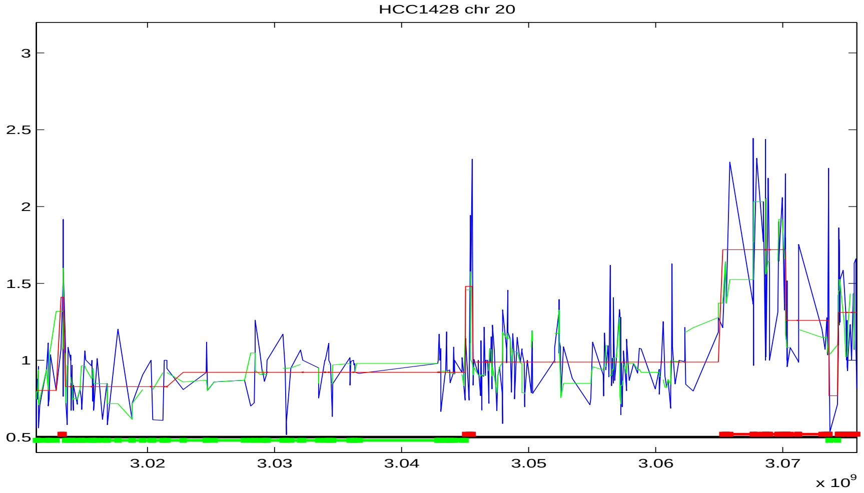


Figure 10: Chromosome 20 of sample HCC1428 plotted against cumulative basepairs. CGH-data are seen in blue, and filtered data as green line.

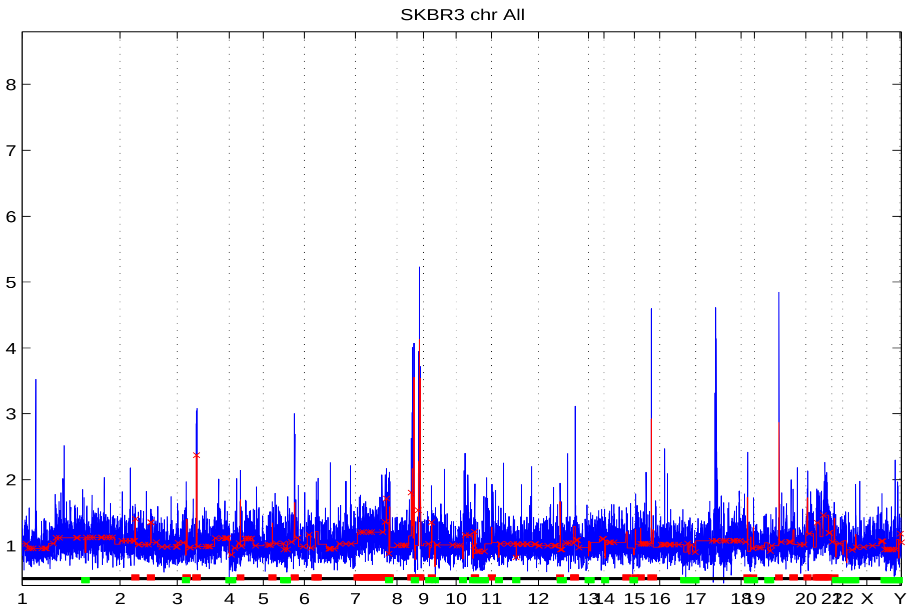
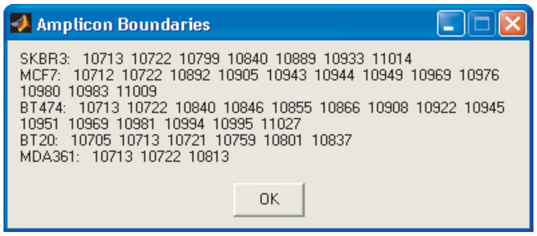
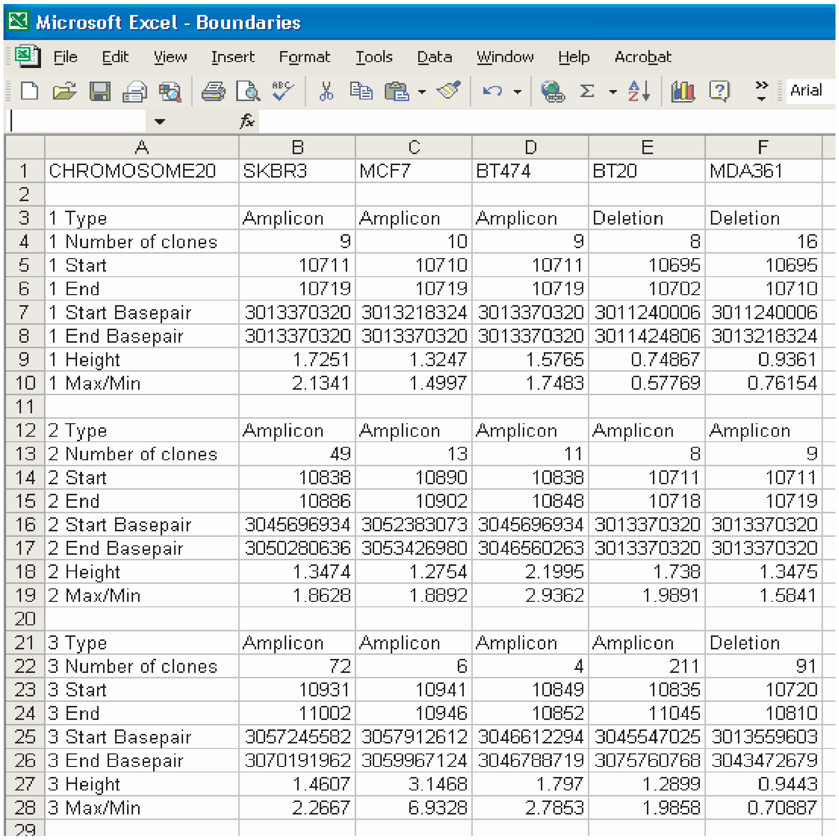


Figure 11: All chromosomes of sample SKBR3. CGH-data are seen in blue, and amplicon boundaries as red line. CGH-Plotter plots dividing lines between the chromosomes. The bar below the data is indicating the amplicons and deletions.



```
SKBR3:  10713 10722 10799 10840 10889 10933 11014
MCF7:  10712 10722 10892 10905 10943 10944 10949 10969 10976
10980 10983 11009
BT474:  10713 10722 10840 10846 10855 10866 10908 10922 10945
10951 10969 10981 10994 10995 11027
BT20:  10705 10713 10721 10759 10801 10837
MDA361:  10713 10722 10813
```

Figure 12: Amplicon Boundaries message box.



	A	B	C	D	E	F
1	CHROMOSOME20	SKBR3	MCF7	BT474	BT20	MDA361
2						
3	1 Type	Amplicon	Amplicon	Amplicon	Deletion	Deletion
4	1 Number of clones	9	10	9	8	16
5	1 Start	10711	10710	10711	10695	10695
6	1 End	10719	10719	10719	10702	10710
7	1 Start Basepair	3013370320	3013218324	3013370320	3011240006	3011240006
8	1 End Basepair	3013370320	3013370320	3013370320	3011424806	3013218324
9	1 Height	1.7251	1.3247	1.5765	0.74867	0.9361
10	1 Max/Min	2.1341	1.4997	1.7483	0.57769	0.76154
11						
12	2 Type	Amplicon	Amplicon	Amplicon	Amplicon	Amplicon
13	2 Number of clones	49	13	11	8	9
14	2 Start	10838	10890	10838	10711	10711
15	2 End	10886	10902	10848	10718	10719
16	2 Start Basepair	3045696934	3052383073	3045696934	3013370320	3013370320
17	2 End Basepair	3050280636	3053426980	3046560263	3013370320	3013370320
18	2 Height	1.3474	1.2754	2.1995	1.738	1.3475
19	2 Max/Min	1.8628	1.8892	2.9362	1.9891	1.5841
20						
21	3 Type	Amplicon	Amplicon	Amplicon	Amplicon	Deletion
22	3 Number of clones	72	6	4	211	91
23	3 Start	10931	10941	10849	10835	10720
24	3 End	11002	10946	10852	11045	10810
25	3 Start Basepair	3057245582	3057912612	3046612294	3045547025	3013559603
26	3 End Basepair	3070191962	3059967124	3046788719	3075760768	3043472679
27	3 Height	1.4607	3.1468	1.797	1.2899	0.9443
28	3 Max/Min	2.2667	6.9328	2.7853	1.9858	0.70887
29						

Figure 13: The title of the first column tells which chromosome is in question. Names of the samples are titles of the other columns. File presents the type, start, end and height of the amplicon or deletion. It also gives the maximum ratio value of the amplicon and the minimum ratio value of the deletion.

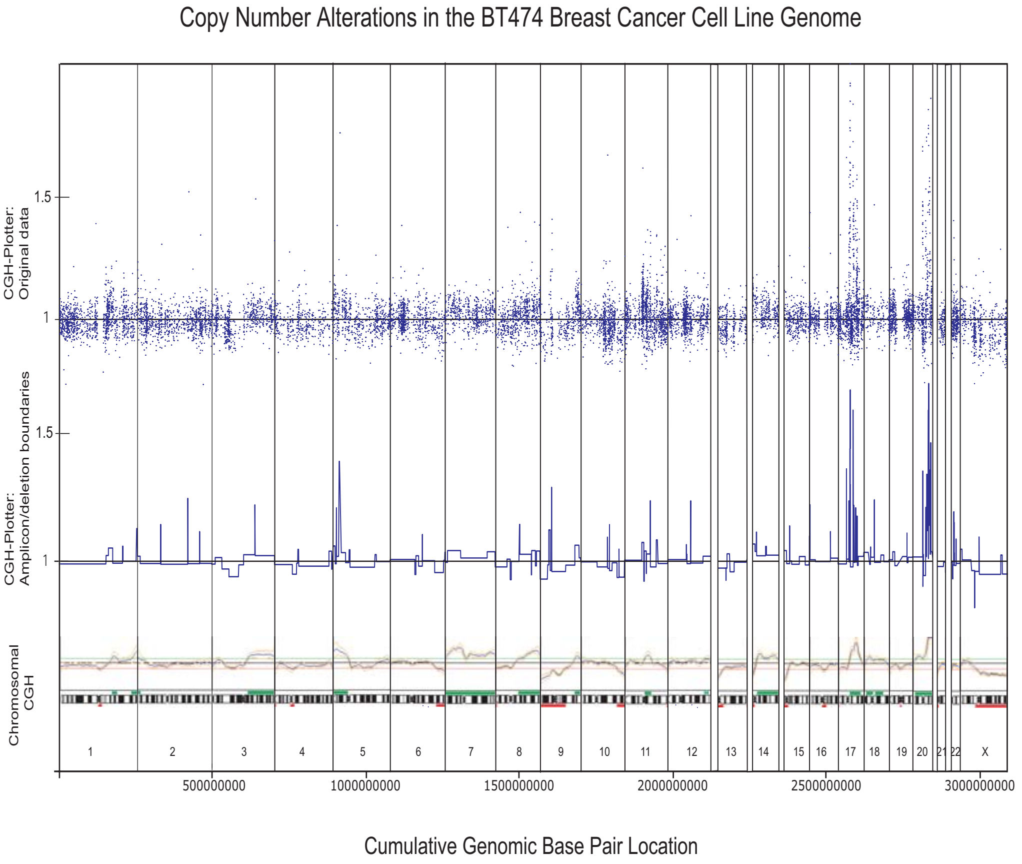


Figure 14: Chromosomal CGH and output of CGH-Plotter for breast cancer cell line 'BT474'. CGH-Plotter original data is shown on top, amplicon/deletion boundaries in the middle and chromosomal CGH-data on bottom. CGH-Plotter can clearly identify amplicons and deletions detected by chromosomal CGH and, as expected due to the higher resolution of array-CGH, also reveals additional aberrations.

In order to compare the performance of CGH-Plotter we have illustrated both the chromosomal CGH and the output of CGH-Plotter in Figure 14.

4 Methods

In this section we describe the methods used in CGH-Plotter in greater detail. The overall view for CGH-Plotter is given in Figure 15.

4.1 Filtering

Before applying the k-means clustering, CGH-ratios in each chromosome are filtered with the moving median or average filter. The user may input the type (i.e. mean or median) and the size of window for the filter. Suggested window sizes are between three and nine.

The filtering proceeds as follows. First CGH-Plotter computes the median/average of first w values, where w is the size of the window. For example, if w is five, the first value in the filtered data is median/average of the first five CGH-data points. Then CGH-Plotter takes again w values beginning from the second data point and computes the median/average depending on the user's choice. The filtering stops when the last data point is reached. Therefore, in standard filtering utilizing moving window, the filtered data are w-1 shorter than the original data. In order to keep data sets in the same size as the originals, CGH-Plotter inputs w-1 NaNs at the end of each chromosome. The chromosomes are filtered individually because, for example, otherwise values in the end of chromosome one would affect to the values of the chromosome two. [1] The filtered data are saved and so it is possible to plot filtered data in the phase 'Plot data'.

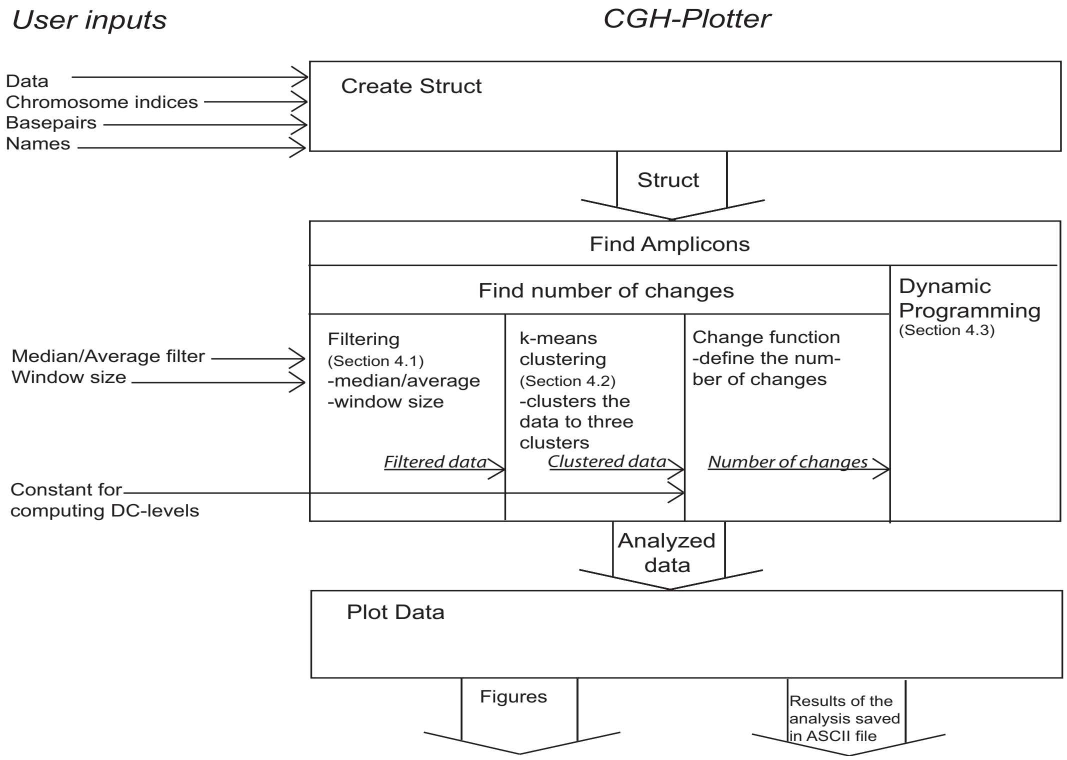


Figure 15: Overall view of CGH-Plotter. The user inputs CGH-data, chromosome indices, basepairs and names of the samples in 'Create Struct' phase. CGH-Plotter creates a struct that is used in the phase 'Find Amplicons'. Further, the user defines the type of the filter and size of the window, which are used in filtering phase. CGH-Plotter clusters filtered data into three clusters with k-means clustering algorithm. Clustered data are delivered to the function that computes the maximum number of the change points. The number of changes is needed when dynamic programming algorithm computes the amplicons and deletions. In 'Plot Data' phase the user may plot the results of the analysis and save the results in ASCII-file.

4.2 *k*-means Clustering

k-means clustering algorithm is used for finding the number of amplicons/ deletions for each chromosome. The idea behind the k-means clustering is to cluster the data to k clusters (k is assumed known). Here the number of the clusters is three denoting amplified genes, deleted genes and baseline genes.

In the k-mean clustering means μ_1, μ_2, μ_3 are first initialized to be the 5:th biggest, the median and the 5:th smallest values, respectively. Actual k-mean clustering proceeds as follows. First, a ratio from the sample is drawn and nearest mean μ_{winner} is found using Euclidian distance. Second, μ_{winner} is updated by moving it closer to the ratio. This procedure is repeated until all m ratios are used. Pseudo-code for the training phase: [2]

```
begin: initialize $\mu_1, \mu_2, \mu_3$
    do classify $m$ ratios to nearest $\mu_i$
        update $\mu_{winner}$
    until the last $m$
end: return $\mu_1, \mu_2, \mu_3$
```

After training phase every ratio is classified to the nearest cluster. The clusters are presented as -1, 0 and 1, denoting deleted, base line and amplified genes. The number of the changes is determined as follows. CGH-Plotter computes $\overline{x}_{max}$ that denotes the mean value of 2% of the highest values in the cluster 'amplified':

$$\overline{x}_{max} = mean(max_{2\%}(\text{cluster '1'}))$$

In a similar fashion $\overline{x}_{min}$ denotes the mean value of 2% of the smallest values in the cluster 'deleted':

$$\overline{x}_{min} = mean(min_{2\%}(\text{cluster '-1'}))$$

We have chosen 2% of the highest/smallest values since the data we used were not very noisy. However, this parameter can be changed in function 'Compute_kmean.m'.

The distance between $\overline{x}_{max}$ and $\overline{x}_{min}$ is computed and multiplied with the constant that the user has determined. The number of the changes (c) is the result of the multiplication rounded downwards:

$$c = constant \cdot (\overline{x}_{max} - \overline{x}_{min}).$$

The default *constant* is six. This number was determined empirically by adjusting it so that known amplicons are found from chromosome 17. The result was then validated by comparing the results to other chromosomes containing known amplicons and by chromosomal CGH (illustrated in Figure 14). In other data sets there may be a need to change this number. If there is known amplicons, we suggest similar way to assess the number of the changes as we have done. However, one should note that if the data are very noisy, the user should try smaller *constant* in order not to detect noise instead of amplicons and deletions. There are surely many other ways to determine the number of changes and in that case the user may want to modify the way the number of the changes is determined to the file 'Compute_kmean.m'.

4.3 Dynamic Programming

In this section dynamic programming is briefly explained. More detailed presentation on dynamic programming can be found, for instance, from [4].

In CGH-Plotter it is assumed that copy number ratios can be approximated with a constant and an error term. As a consequence, CGH-data can be understood as a signal having constant levels, and In essence, there exists three kinds of constant levels: base line, amplicon and deletion levels and these are to be identified by the dynamic programming algorithm. It is assumed that the number of the changes of constant levels (c) is known. We use k-means for this purpose as explained in previous section.

Assume that the CGH-signal

$$x[n] = \begin{cases} A_1 & n = 1, 2, \ldots, n_1 \\ A_2 & n = n_1 + 1, n_1 + 2, \ldots, n_2 \\ \vdots & \vdots \\ A_{c+1} & n = n_c + 1, n_c + 2, \ldots, N \end{cases}$$

is corrupted by noise. Dynamic programming identifies constant levels $A = [A_1, A_2, A_3, \ldots, A_{c+1}]$ and change points $n = [n_0, n_1, n_2, n_3, \ldots, n_c, n_{c+1}]$, where $n_0 = 1$ and $n_{c+1} = N$ by minimizing the function

$$J(A, n) = \Sigma_{i=0}^{c} \Sigma_{n=n_i+1}^{n_{i+1}} (x[n] - A_i)^2.$$

The idea of the dynamic programming is to find the shortest path from the value $x[1]$ to value $x[N]$. Dynamic programming utilizes the Markov property, which ensures that the distance between points $x[n_1]$ and $x[n_2]$ does not depend upon which path was used at arriving to the point $x[n_1]$. Therefore dynamic programming is capable for finding the minimum of $J(A, n)$ without checking every possible combinations of $n_1, n_2 \ldots, n_c$.

In practice, the procedure for identifying the constant levels proceeds as follows. First, constant levels are estimated. A_i is the mean of the interval $[n_{i-1} + 1, n_i]$ and

$$\Delta_i[n_{i-1} + 1, n_i] = \Sigma_{n=n_{i-1}+1}^{n_i} (x[n] - A_i)^2.$$

Second, function $J(A, n)$ is minimized over n using dynamic programming:

$$\begin{aligned} I_k[L] &= min \Sigma_{i=1}^{k} \Delta_i[n_{i-1} + 1, n_i] \\ &= min[(min \Sigma_{i=1}^{k-1} \Delta_i[n_{i-1} + 1, n_i]) + \Delta_k[n_{k-1} + 1, n_k]] \\ &= min(I_{k-1}[n_{k-1}] + \Delta_k[n_k + 1, L]). \end{aligned}$$

This shows that the minimum error for the interval $[1, L]$ can be computed by adding the minimum error of the last segment to the error of the previous segments.

CGH-Plotter stores constant levels A and indices to the change points of these levels.

5 Summary

CGH-Plotter is a Matlab toolbox that is aimed to CGH-data analysis. The main purpose of CGH-Plotter is to identify and visualize the amplicon and deletion regions of CGH-data. With a graphical user interface CGH-Plotter is straightforward to use. The user has many possibilities to illustrate the CGH-data. For example, the data can be illustrated as ratios or log-transformed ratios and plotted against basepairs (if available). CGH-Plotter enables the user to visualize each sample individually or all samples in parallel. It is also possible to plot the data of one chromosome or the data of the sample genomic wide. The results can be stored to tab delimited text file, in which the results can easily be examined.

The freely available CGH-Plotter is really easy to operate with. Further it is easy to modify and add functions to CGH-Plotter. CGH-Plotter toolbox is under continuous development and in the future it will include new analysis and illustration functions.

CGH-Plotter has shown to be capable of rapid high-throughput analysis of CGH-data. Moreover the results obtained from CGH-Plotter are consistent with chromosomal CGH and thereby the results given by CGH-Plotter are verified by biological knowledge.

References

[1] Astola, J., Kuosmanen, P. (1997). *Fundamentals of Nonlinear Digital Filtering,* CRC Press LLC, Florida.

[2] Duda, R.O., Hart, P.E., Stork, D.G. (2001) *Pattern Classification,* John Wiley & Sons, Inc, New York, 2nd edition.

[3] Hyman, E., Kauraniemi, P., Hautaniemi, S., Wolf, M., Mousses, S., Rozenblum, E., Ringnér, M., Sauter, G., Monni, O., Elkahloun, A., Kallioniemi, O-P. and Kallioniemi, A. (2002). Impact of DNA amplification on gene expression patterns in breast cancer. *Cancer Research* **Vol. 62**, pp. 6240-6245.

[4] Kay, S.M. (1998). *Fundamentals of Statistical Signal Processing, Volume II, Detection Theory,* Prentice-Hall, New Jersey.

[5] Gray, J. W., Collins, C. (2000). Genome changes and gene expression in human solid tumors. *Carcinogenesis,* **Vol. 21**, pp. 443–452.

[6] Monni, O., Bärlund, M., Mousses, S., Kononen, J., Sauter, G., Heiskanen, M., Chen, Y., Bittner, M., Kallioniemi, A. (2001). Comprehensive copy number and gene expression profiling of the 17q23 amplicon in human breast cancer. *Proceedings of the National Academy of Sciences, USA,* **Vol. 98**, pp. 5711–5716.

[7] Pollack, J., Perou, C., Alizadeh, A., Eisen, M., Pergamenschikov, A., Williams, C., Jeffrey, S., Botstein, D., Brown, P. (1999). Genome-wide analysis of DNA copy number changes using cDNA microarrays. *Nature Genetics,* **Vol. 23,** pp. 41–46.