User manual

(June 2008, v1.0)

CARPET

(Collection of Automated Routine Programs for Easy Tiling)

A web-based package for the analysis of ChIP-chip and expression tiling data

Matteo Cesaroni[1]*, Davide Cittaro[2], Alessandro Brozzi[1], Pier Giuseppe Pelicci[1] and Lucilla Luzi[3]*

[1] Department of Experimental Oncology, European Institute of Oncology, Via Ripamonti 435, 20141 Milano, ITALY
[2] Cogentech, Consortium for Genomic Technologies, Via Adamello 16, 20139 Milano, ITALY
[3] IFOM, FIRC Institute of Molecular Oncology Foundation, Via Adamello 16, 20139 Milano, ITALY

TABLE OF CONTENTS

1. Introduction

CARPET (Collection of Automated Routine Programs for Easy Tiling) is a set of Perl, Python and R scripts, integrated on the Galaxy2 web-based platform (Blankenberg et al., 2007), for the analysis of ChIP-chip and expression tiling data. CARPET allows rapid experimental data entry, simple quality control, easy identification and annotation of enriched ChIP-chip regions, detection of the absolute or relative transcriptional status of genes assessed by expression tiling experiments and, more importantly, it allows the integration of ChIP-chip and expression data. Results can be visualized instantly in a genomic context within the UCSC genome browser as graph-based custom tracks through Galaxy2. All generated and uploaded data can be stored within sessions and are easily shared with other users.

The suite program can be accessed through the Galaxy mirror site of IFOM-IEO-CAMPUS at http://bio.ifom-ieo-campus.it/galaxy.

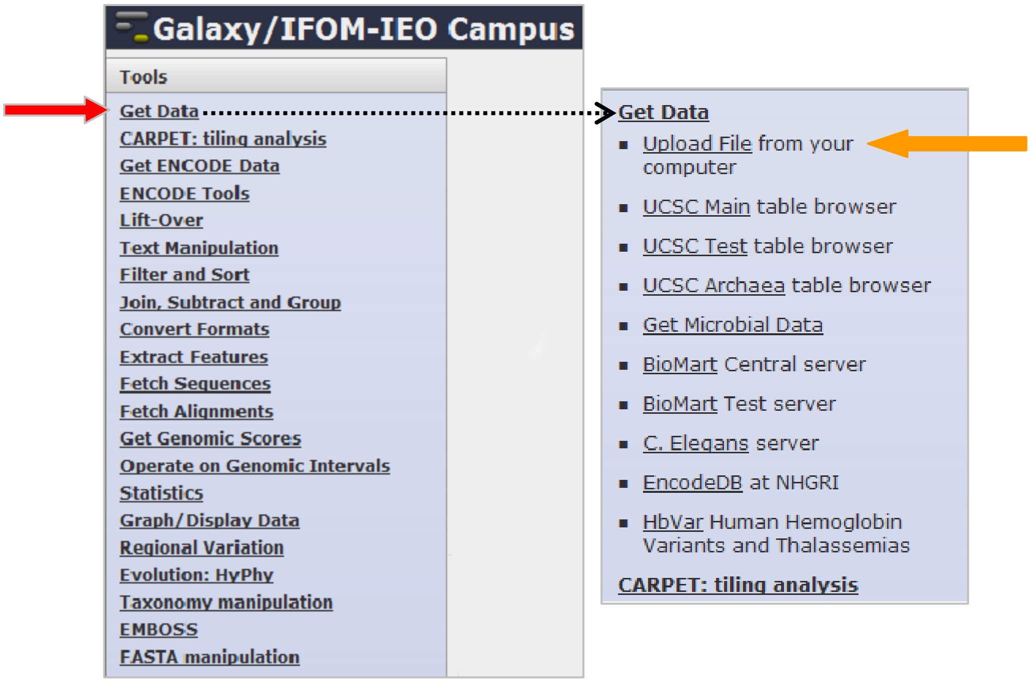
For questions and suggestions you may please contact
matteo.cesaroni@ifom-ieo-campus.it
lucilla.luzi@ifom-ieo-campus.it
davide.cittaro@ifom-ieo-campus.it

2. The Galaxy platform

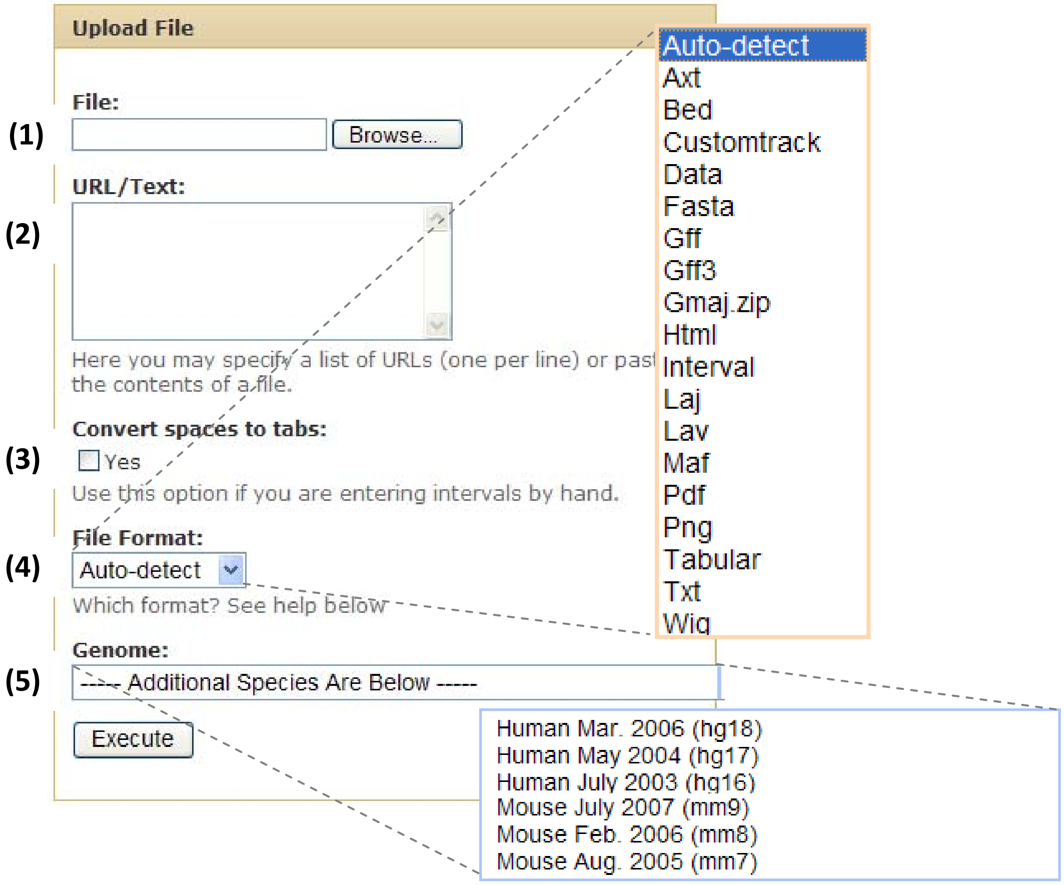
The Galaxy project was firstly developed on 2005 by Giardine and coworker (Giardine et al., 2005) who suggested it as a platform for interactive large-scale genome analysis; more recently it was furthermore proposed as framework for collaborative analysis of the outstanding ENCODE data (Blankenberg et al., 2007). Galaxy is not a browser, instead, it allows users to gather and manipulate data from existing resources in a variety of ways and, what is more, Galaxy provides a very user-friendly interface that permits interactions between experimental and computational biologists by providing a simple interface (important to the former) and a robust software integration environment (important for the latter). The simplicity of Galaxy2's tool integration protocol allows developers easily to integrate their programs and make them available to biologists.
You can find further information on Galaxy2 main site http://g2.trac.bx.psu.edu/.
Please, for general issues regarding the Galaxy usage, consult the detail tutorial session at the official "Galaxy Screencasts and Demos" webpage.

3. Uploading data

The first primarily phase to use all the CARPET programs is to upload your data files for the analysis.
CARPET, for this purpose, makes use of the primarily *Get Data/Upload File* Galaxy tool.



Data can be uploaded either browsing for a file in your hard disk (1 in the figure below) or making a cut&paste in the “URL/Text:” window space (2 in the figure below).



Here you may specify a list of URLs (one per line) or past the contents of a file.

Use this option if you are entering intervals by hand.

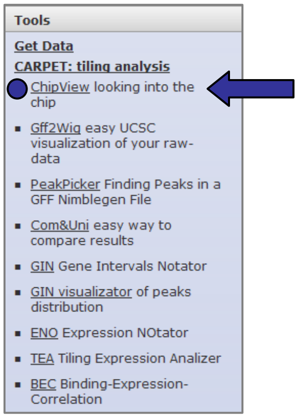
Which format? See help below

Galaxy accepts a lot of different file formats (the list is show on previous screenshot; for details refer to “Galaxy Screencasts and Demos” webpage, or to Appendix A of this manual for a detailed description of all CARPET handled file format).
The system is able to auto-detect the format of files you are uploading, otherwise you can choose the proper one from the “File Format” popup menu (4, in the figure above).
Basically, every “table-like” file that contains row and (tab delimited) columns are recognized. If you are not sure if your file is tab delimited, ask the system to convert spaces to tabs (3, in the figure above).

The most of the CARPET programs work with GFF format files as input, that are the standard format files provided by Nimblegen for array tiling data results; in all the other cases, a detail description of the required file format is depicted in the further dedicated tool sections or in the Appendix A of this manual. Corresponding example files are provided on server for downloading and linked in this manual through the digital text.

It is not mandatory, but if you are uploading files that can be visualized on the UCSC Genome Browser (GFF or BED files) we suggest to associate it to its corresponding Organism type and Genome Assembly version choosing the right one in the “Genome:” popup menu (5, in the figure above).

4. Quality assessment by chip image visualization: ChipView

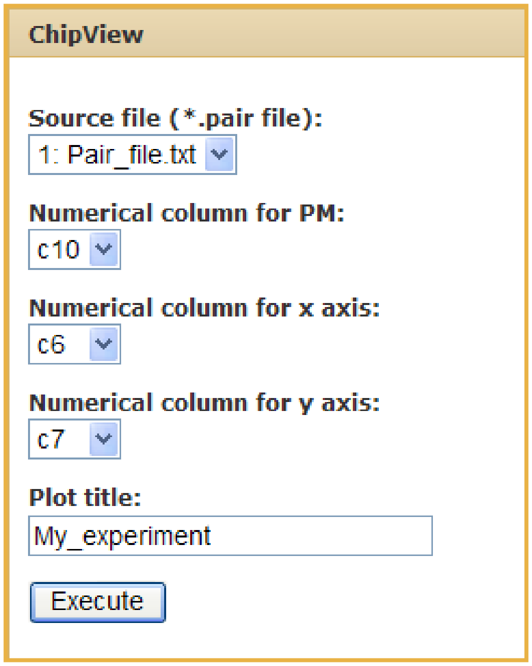

Tools
Get Data
CARPET: tiling analysis
ChipView looking into the chip
Gff2Wig easy UCSC visualization of your raw-data
PeakPicker Finding Peaks in a GFF Nimblegen File
Com&Uni easy way to compare results
GIN Gene Intervals Notator
GIN visualizator of peaks distribution
ENO Expression NOtator
TEA Tiling Expression Analizer
BEC Binding-Expression-Correlation

Usually Nimblegen does not provide, as results, an image of the hybridized chip. However, maybe you would be interested to inspect the distribution of the signal over the chip, because it would be important to determine the presence of artefacts or hybridization problems. ***ChipView,*** starting from the “Pair files” data provided by Nimblegen (look at Appendix A for details), or from your “custom raw signal-coordinates file”, allows you to create and visualize an image simulating the chip surface, as show in the exampleof output file image.
Fundamentally, the program considers only 3 kinds of informations to reproduce the image of the chip: chip raw signals and the corresponding coordinate positions on x and y chip’s sides. Therefore, even if you don’t have a standard Nimblegen “Pair File” to direct upload on the system, ***ChipView*** will use every type of raw data table, if they also supply corresponding coordinates positions on chip. Your duty will be just to specify to the system which are columns that contains the matching data.

For more detailed “Pair File” format description look to Appendix A of this manual.

ChipView usage:

- upload the Pair File (or your custom raw signal-coordinates file) in the *Get Data/Upload File* Galaxy tool as "Tabular" format (*See Section 3*);
- (for Nimblegen Pair File) once the file is uploaded, control that it contain 11 columns;
- click on ***ChipView*** link from the "CARPET: tiling analysis" tools list;
- (for Nimblegen Pair File) choose column C10 for "PM value", C6 for "X position" and C7 for "Y position", as show below; (for custom raw signal-coordinates file) choose column "PM value", "X position" and "Y position" accordantly with your custom file;
- click Execute.



ChipView

Source file (*.pair file): 1: Pair_file.txt

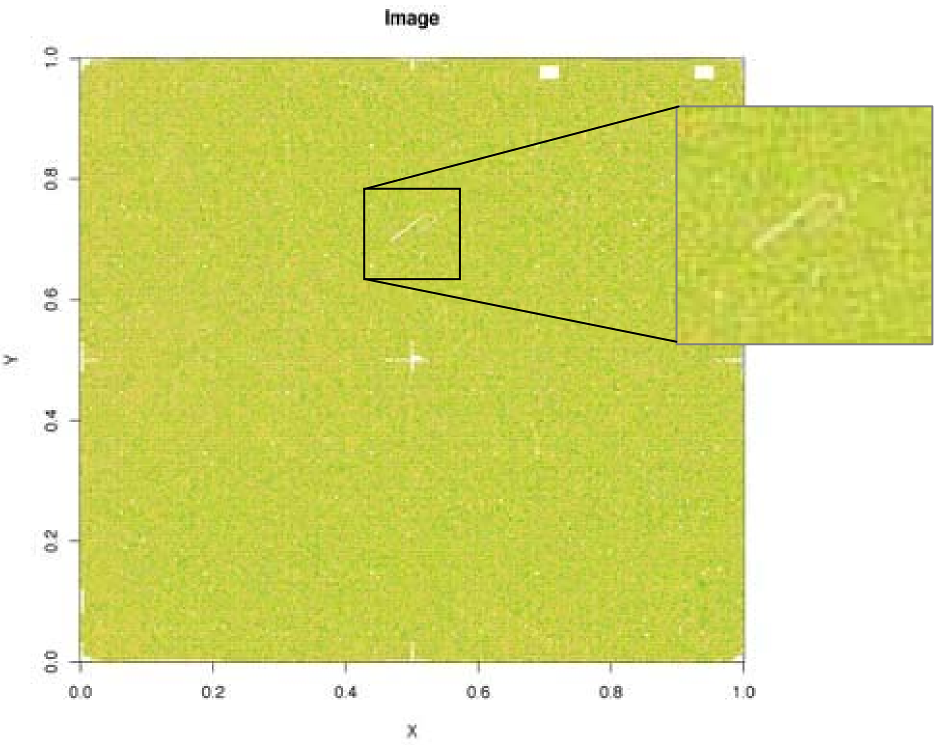
Numerical column for PM: c10

Numerical column for x axis: c6

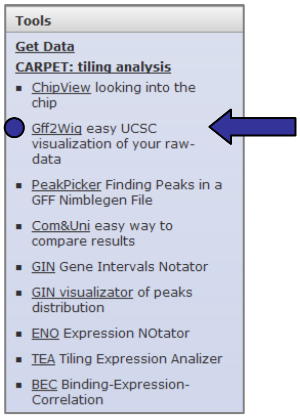
Numerical column for y axis: c7

Plot title: My_experiment

ChipView produces a PDF image file available for download and save on your hard disk. An example is reported in the next figure.



5. Visualize chip intensities data on UCSC Genome Browser: GFF2WIG

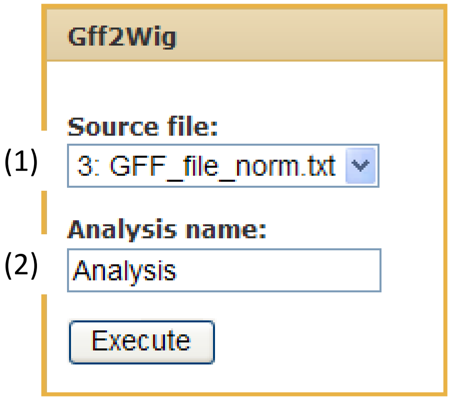

Tools

Get Data

CARPET: tiling analysis

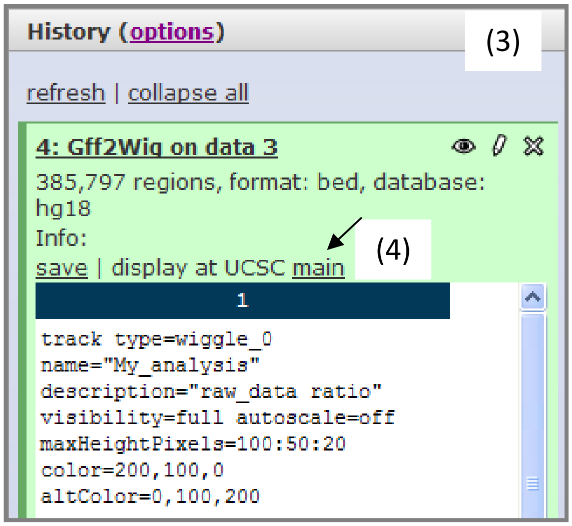
- ChipView looking into the chip
- Gff2Wig easy UCSC visualization of your raw-data
- PeakPicker Finding Peaks in a GFF Nimblegen File
- Com&Uni easy way to compare results
- GIN Gene Intervals Notator
- GIN visualizator of peaks distribution
- ENO Expression NOtator
- TEA Tiling Expression Analizer
- BEC Binding-Expression-Correlation

Another interesting opportunity that maybe you would take, before to analyse more deeply you tiling data, is to have a preliminary look to the profile of your log2(ratio) raw signal. To visualize data intensity of each probe (log2(ratio) of Cy5, Cy3 signals) on UCSC Genome Browser as a continuous histogram on top of the genome sequence, the GFF file of your tiling experiment need to be transformed in WIG_bed format (http://genome.ifom-ieo-campus.it/goldenPath/help/wiggle.html).


Gff2Wig

(1) Source file: 3: GFF_file_norm.txt

(2) Analysis name: Analysis


History (options) (3)

4: Gff2Wig on data 3

385,797 regions, format: bed, database: hg18

Info:

save | display at UCSC main (4)

1

```
track type=wiggle_0
name="My_analysis"
description="raw_data ratio"
visibility=full autoscale=off
maxHeightPixels=100:50:20
color=200,100,0
altColor=0,100,200
```

GFF file example *(you may click on the link to download a GFF file example - zipped compressed ~4.4Mb)*

GFF2WIG usage:

- upload your GFF File in the *Get Data/Upload File* Galaxy tool as "GFF" format (*See Section 3*);
- once the file is uploaded click the ***GFF2WIG*** link from the "CARPET: tiling analysis" tools list;
- the name of your GFF file should be appear in the "Source file:" popup menu (1, in the figure above);
- select it;
- if you like you can specify an "Analysis name" that will be shown in the UCSC Genome Browser as track name (2, in the figure above);
- click Execute;

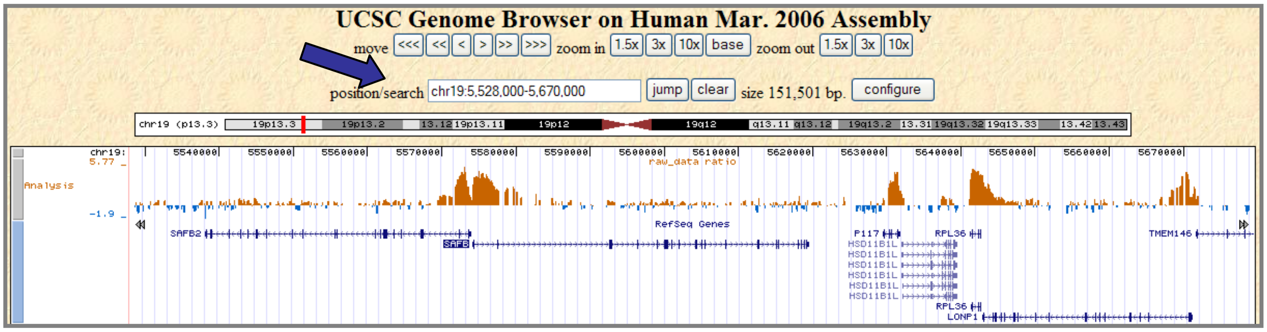
GFF2WIG transformed the GFF file to a WIG_bed file similar to that reported below (just the first 3 rows are shown).

```
track  type=wiggle_0  name="Analysis  name"  description="raw_data  ratio"  visibility=full
autoscale=off maxHeightPixels=100:50:20 color=200,100,0 altColor=0,100,200
chr19   1000000  1000050   -1.2
chr19   1000100  1000150    2.9
```

……

- to visualize your chip log2(ratio) intensity on Genome Browser, click the UCSC link (4, in the figure above) in the history frame (3, in the figure above).

Please note: *if you are using the GFF example file (that contains tiling array data of the human chr19) and the Genome Browser does not get automatically to the chr19, insert any chr19 coordinate (i.e. chr19:5,528,000-5,670,000) in the "position/search" window in the Genome Browser. You will obtain a similar result.*



UCSC Genome Browser on Human Mar. 2006 Assembly

position/search chr19:5,528,000-5,670,000

size 151,501 bp.

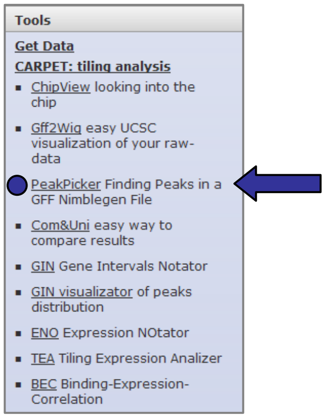
Once you had a look of the chip surface with ***ChipView*** and of the enrichment profile of your log2(ratio) data as WIG_bed in the Genome Browser by the ***GFF2WIG*** tool, it is now the time to start the real analysis of your tiling experiments. The next sections 6 and 7 will describe, separately, the analysis pipeline and the CARPET instruments we suggest for ChIP-chip (Section 6) and expression tiling data (Section 7).

6. Working with ChIP-chip data

6.1. Peaks identification: PeakPeaker

finding Peaks in a GFF Nimblegen File

PeakPeaker is a Perl script that is able to identify enriched regions (peaks) from a ChIP-chip experiment. ***PeakPeaker*** tool utilizes NimbleGen log2(ratio) files in GFF format (or a proper reformatted GFF file obtained from other platforms) as INPUT FILE and it identifies regions of enriched signals (peaks), providing as output, a table, in the same GFF format, that contains peaks genomic coordinates, and only scoring or scoring and statistical values.
Input GFF file example (*click on the link to download a GFF file example - zipped compressed ≈4.4Mb*); for more details look to the Appendix A of this manual.



Tools

Get Data

CARPET: tiling analysis

- ChipView looking into the chip
- Gff2Wig easy UCSC visualization of your raw-data
- PeakPicker Finding Peaks in a GFF Nimblegen File
- Com&Uni easy way to compare results
- GIN Gene Intervals Notator
- GIN visualizator of peaks distribution
- ENO Expression NOtator
- TEA Tiling Expression Analizer
- BEC Binding-Expression-Correlation

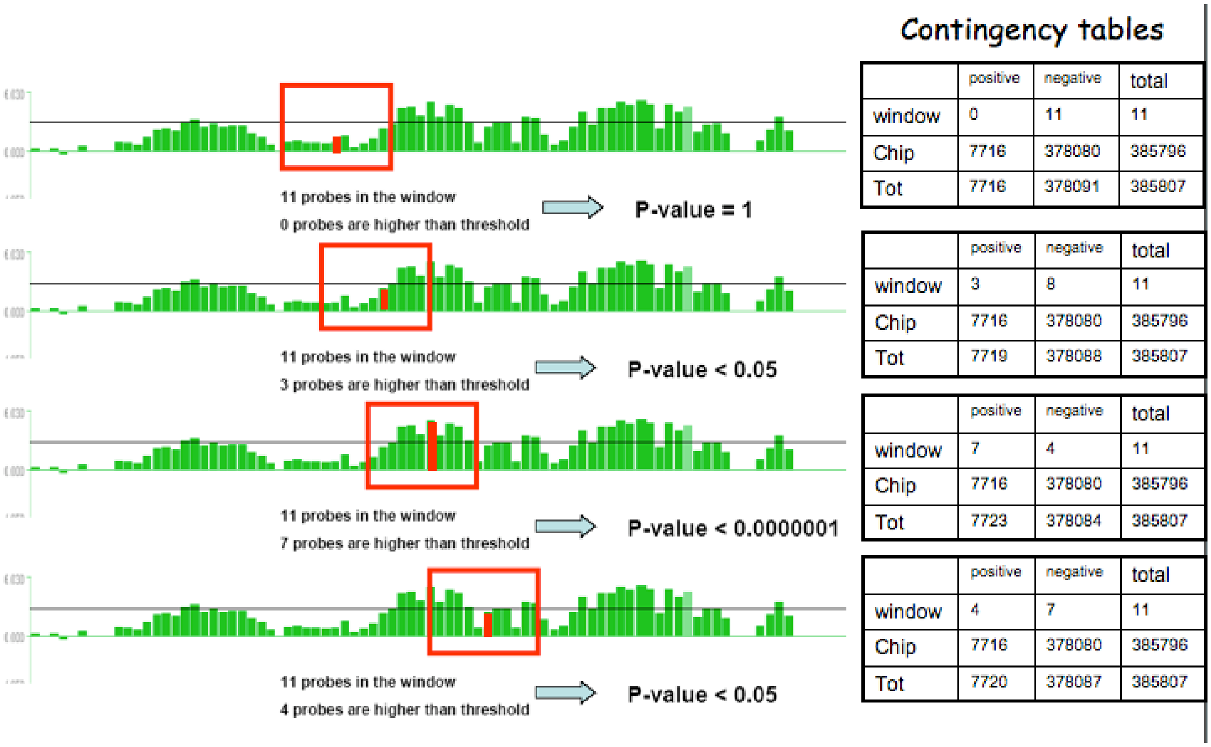
How does ***PeakPeaker*** work?
A "peak" is defined as the region where multiple probes, with a log2(ratio) greater than a user-defined threshold, are located genomically close to one an-other.

PeakPeaker makes two assumptions: i) data are enriched for signals in the positive direction ("one-tailed"); ii) a peak is represented by multiple probes genomically located close to each other.

It makes use of the sliding window statistical approach and it proceeds essentially as previously described by Scacheri et al (Scacheri et al., 2006). A window centered at each probe of the array moves probe by probe; in each window Chi squared is calculated

$$\chi^2=\sum\frac{(f(a)-f(e))^2}{f(e)}$$

by building a contingency table for each window/probe position, and a p-value is assigned.



Contingency tables

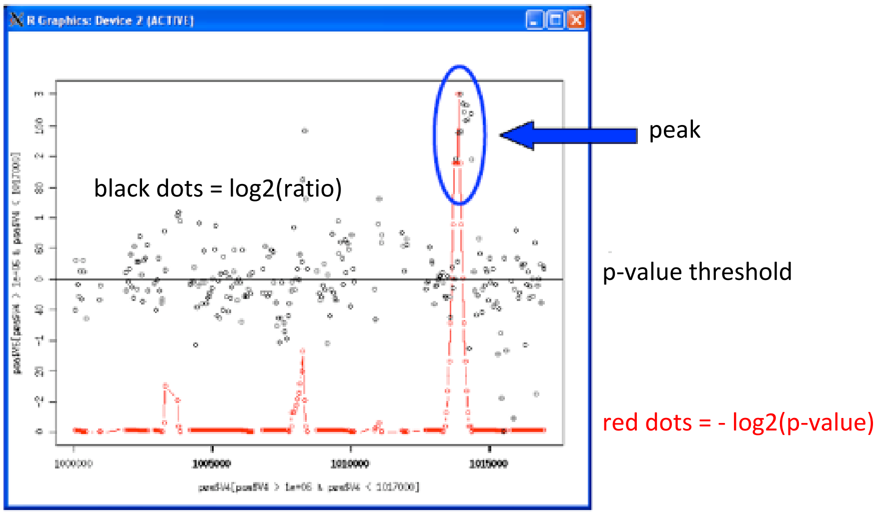
	positive	negative	total
window	0	11	11
Chip	7716	378080	385796
Tot	7716	378091	385807

	positive	negative	total
window	3	8	11
Chip	7716	378080	385796
Tot	7719	378088	385807

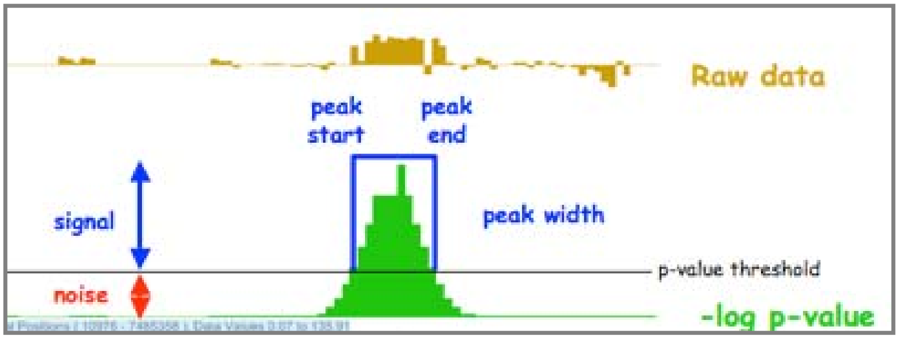
	positive	negative	total
window	7	4	11
Chip	7716	378080	385796
Tot	7723	378084	385807

	positive	negative	total
window	4	7	11
Chip	7716	378080	385796
Tot	7720	378087	385807

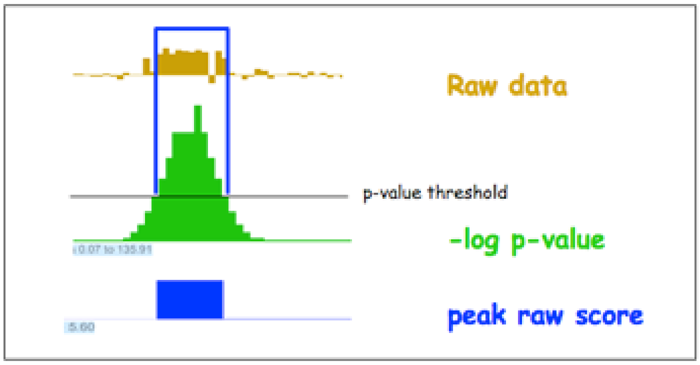
Therefore the program produce a new profile of your experiment (figure below), derived from the "-log2(p-values)" associated to each probe, and from that profile peak margins are finally delineated respect to a user-defined p-value threshold.


R Graphics: Device 2 (ACTIVE)
black dots = log2(ratio)
peak
p-value threshold
red dots = - log2(p-value)
1000000
1005000
1010000
1015000
posEV4[posEV4 > 1e+06 & posEV4 < 1017000]

"-log2(p-values)" are used, instead of plain p-values, for this procedure since they takes in account the so called “neighboring probes effect”, hence dramatically decrease the background signal influence.



Alternatively or beside the statistical p-value calculation for each peak, ***PeakPeaker*** estimates a peak score value that takes into account the length and the intensity of the raw log2(ratio) signals under the peak and defined from the formula reported below.



$$S = \boxed{\frac{\sum_1^n \log(\text{ratio})}{n}} + \boxed{\sqrt{n}}$$

Average of probe raw values belonging to a peak

Height contribution

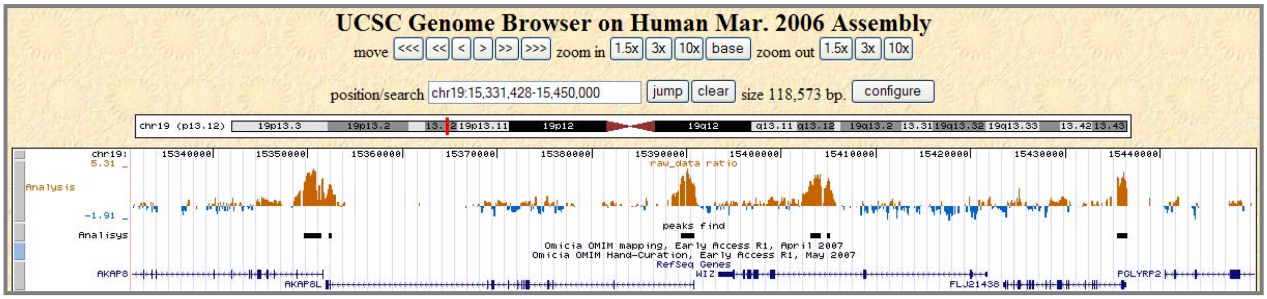
Square root of probe number belonging to a peak

Length contribution

S stands for Score value, $\sum_1^n$ log(ratio) is the sum of all log2(ratio) values of the probes under the peak, n the number of probes under the peak and $\sqrt{n}$ is an empirically defined correction factor for the peak length contribution.

The score value consist of to main contributions: the peak height, represented by the average of log2(ratio) values of the probes below the peak and the peak length, described by an empiric correction factor established by the square root of the number of probes.

PeakPeaker allows the user to set a lot of different parameters: user can decide the minimal number of probes that must exceed the defined threshold (fix this parameter in function of the peak length you expect), as well as the maximum distance allowed, in order to consider two probes as contiguous (fix this parameter in function of your chip tiling design). Thresholds on p-values can also be set if you would like to vary the stringency of your analysis, and neighbor-enriched regions can be joined together (fix this parameter in function of the peak spread you expect from your studied feature). The output of the analysis is again a GFF file that can be visualized simultaneously with your raw log2(ratio) data on the UCSC Genome Browser, as shown in the next screenshot.



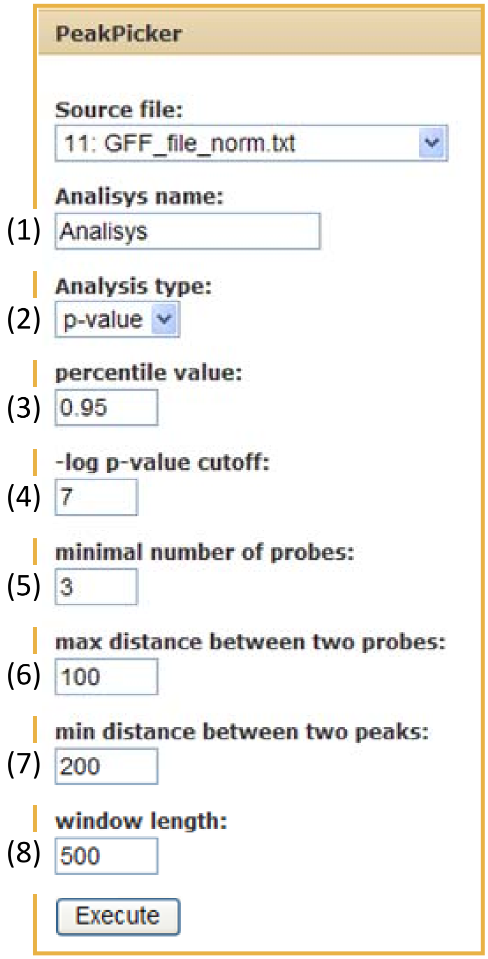
position/search chr19:15,331,428-15,450,000 size 118,573 bp.

PeakPeaker usage:

- upload your GFF File in the *Get Data/Upload File* Galaxy tool as "GFF" format (*See Section 3*);
- once the file is uploaded click the ***PeakPeaker*** link from the "CARPET: tiling analysis" tools list;
- if you like you can specify an "Analysis name" that will be shown in the UCSC Genome Browser as track name (1, in the figure below);
- set each requested parameters:
  - analysis type (2, in the figure below):
    - p-value analysis performs peaks determination based on p-value inference (statistical analysis);
    - score analysis performs peaks determination based only on the scoring system (scoring analysis);
  - percentile value (3, in the figure below):
    - it is used to calculate the threshold rate based on dataset distribution to filter out background;
  - -log p-value cutoff (4, in the figure below):
    - (required only for p-value based analysis) it is the cutoff integer used to identify a significant peak;
  - minimal # of probes (5, in the figure below):
    - minimal number of consecutive probes used to define a peak

- max distance 2 probes (6, in the figure below):
  - the greatest nucleotide distance (bp) between two probes that allow to consider two probes as adjacent;
- min distance 2 peaks (7, in the figure below):
  - minimum nucleotide distance (bp) required to consider two peaks as separate entities;
- window length (8, in the figure below):
  - length in bp of the window used for statistical analysis.

- click Execute;



Source file: 11: GFF_file_norm.txt
Analisys name: Analisys
Analysis type: p-value
percentile value: 0.95
-log p-value cutoff: 7
minimal number of probes: 3
max distance between two probes: 100
min distance between two peaks: 200
window length: 500

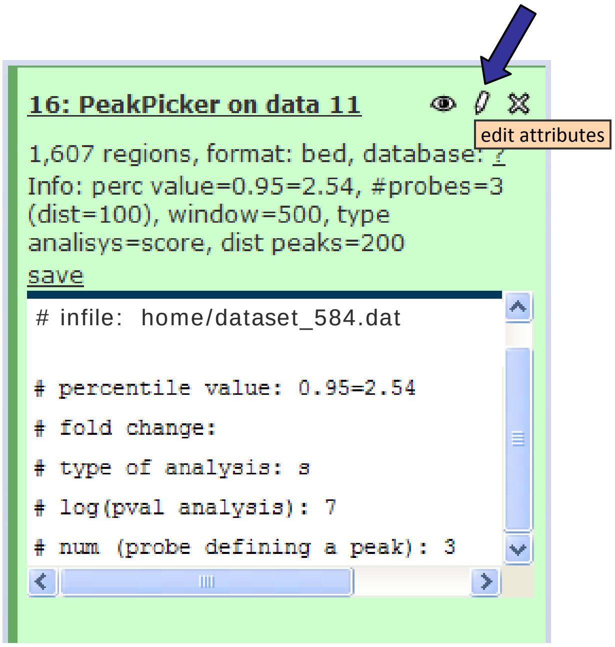
When you chose to proceed with the <u>statistical analysis</u> the program produce as output, a GFF file similar to that show up below (the first 4 lines are displayed), where in column 6, C6, and in column 9, C9, the maximum value of –log2(p-value) reached by the probes belonging to the peak and the peak score are reported.

C1	C2	C3	C4	C5	C6	C7	C8	C9
#chromosome	*Source*	*Feature*	*Start*	*End*	*max -log2(p-value)*			*Score*
track name=Statical Analysis description="PeakPeaker identified peaks" visibility=2								
chr19	NimbleScan	Analysis	20724	21581	25.38	.	.	7.27
chr19	NimbleScan	Analysis	22168	22463	13.95	.	.	5.79
chr19	NimbleScan	Analysis	293061	293367	13.27	.	.	4.79

When you chose to proceed with the scoring analysis the program produce as output, a GFF file similar to that reported below (the first 4 lines are reported), where in column 6, C6, and in column 9, C9, the computed peak score is reported.

C1	C2	C3	C4	C5	C6	C7	C8	C9
#chromosome	*Source*	*Feature*	*Start*	*End*	*Score*			*Score*
track name=Scoring Analysis description="PeakPeaker identified peaks" visibility=2								
chr19	NimbleScan	Analysis	22122	22313	5.95	.	.	5.95
chr19	NimbleScan	Analysis	293015	293157	4.81	.	.	4.81
chr19	NimbleScan	Analysis	294081	295180	8.10	.	.	8.10

To visualize your new results on the UCSC Genome Browser check that the GFF file is correctly interpreted by Galaxy, following the "edit attribute" hyperlink (show in the figure below) and verify that Database/Build (1), Start column (2), End column (3) are properly set; strand information is not needed in this case (4).



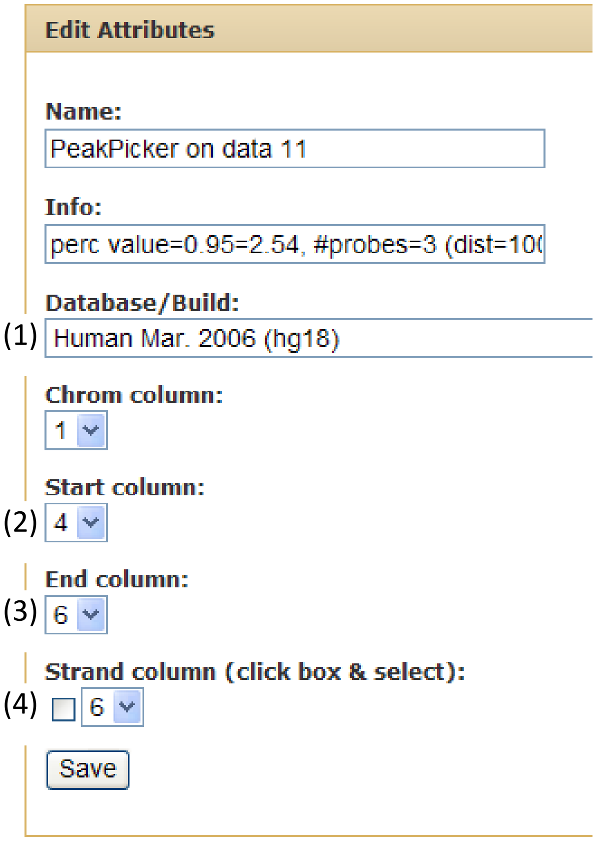
16: PeakPicker on data 11

1,607 regions, format: bed, database: ?
Info: perc value=0.95=2.54, #probes=3 (dist=100), window=500, type analisys=score, dist peaks=200

save

```
# infile:  home/dataset_584.dat

# percentile value: 0.95=2.54
# fold change:
# type of analysis: s
# log(pval analysis): 7
# num (probe defining a peak): 3
```



Edit Attributes

Name: PeakPicker on data 11

Info: perc value=0.95=2.54, #probes=3 (dist=10

Database/Build: Human Mar. 2006 (hg18)

Chrom column: 1

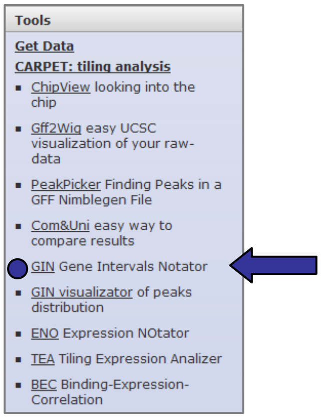
Start column: 4

End column: 6

Strand column (click box & select): ☐ 6

6.2. Peaks annotation: Genomic Interval Notator - GIN

Once you have identified and mapped enriched peaks of binding of your ChIP-chip experiment, one of the next point you may would estimate is which are the relationships of your data with gene loci. ***GIN*** (Genomic Interval Notator) tool helps you in this task, infact it annotates peak queries with a user-defined annotation tables (e.g. RefSeq, UCSC genes, Ensembl Genes) and it calculates the relative position of peaks with respect to the transcript associated features (e.g. promoter, exon, intron, intergenic).

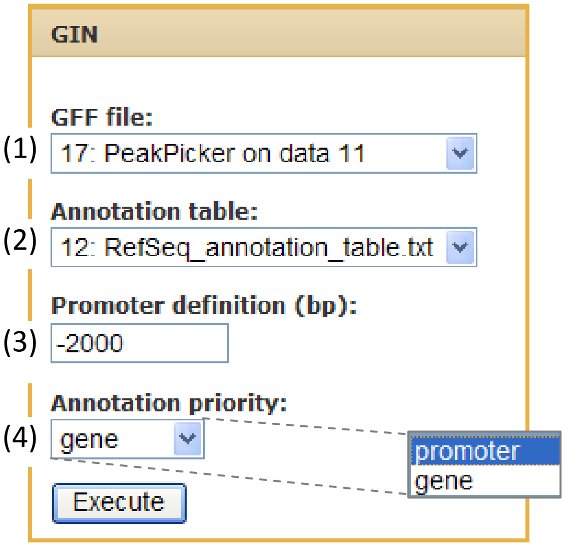

Tools

Get Data

CARPET: tiling analysis

- ChipView looking into the chip
- Gff2Wig easy UCSC visualization of your raw-data
- PeakPicker Finding Peaks in a GFF Nimblegen File
- Com&Uni easy way to compare results
- GIN Gene Intervals Notator
- GIN visualizator of peaks distribution
- ENO Expression NOtator
- TEA Tiling Expression Analizer
- BEC Binding-Expression-Correlation

How does ***GIN*** work? It uses two files: a GFF file with genomic intervals (i.e. the output file of ***PeakPeaker***) and any user-preferred transcript annotation tables (e.g. Ref-Seq, UCSC genes) that can be easily downloaded from the UCSC Genome Browser database (see Appendix A of this manual for more information).


GIN

(1) GFF file: 17: PeakPicker on data 11

(2) Annotation table: 12: RefSeq_annotation_table.txt

(3) Promoter definition (bp): -2000

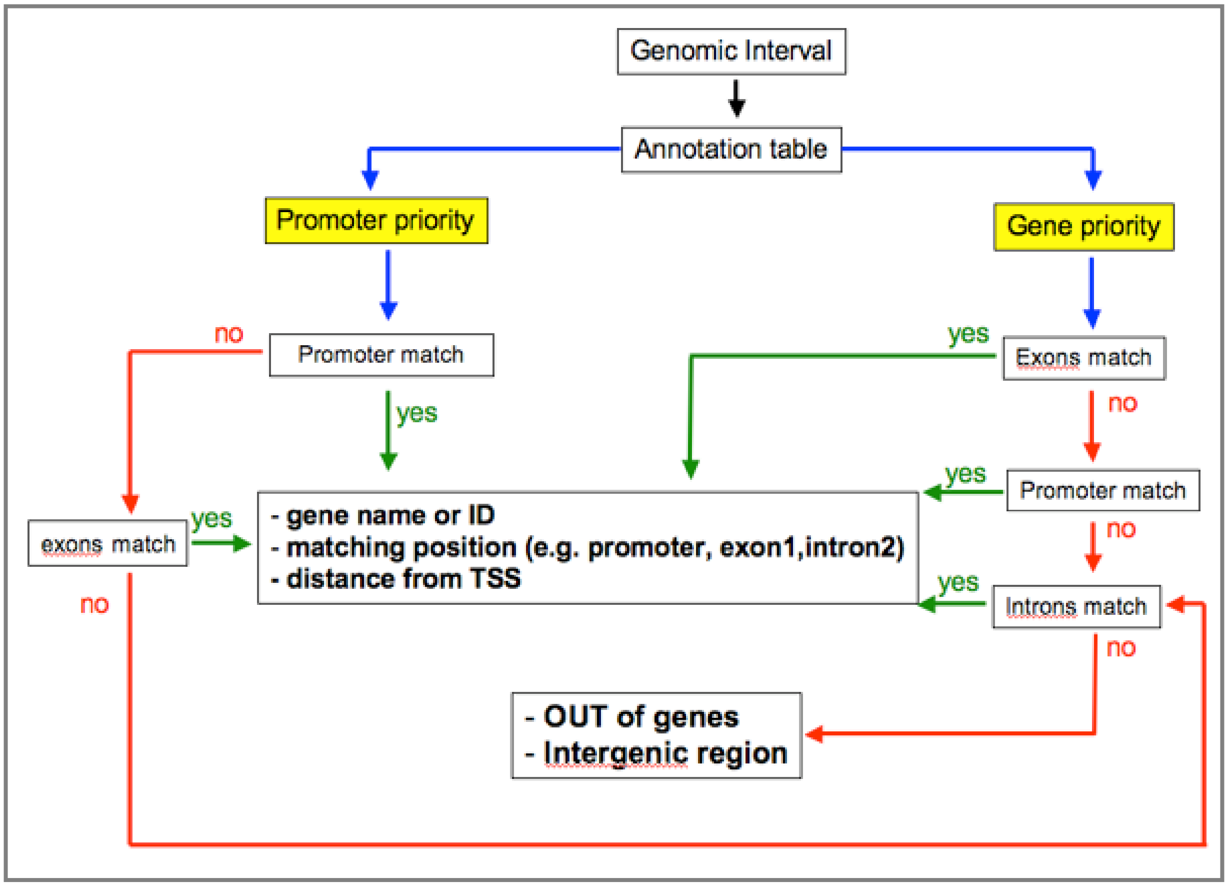
(4) Annotation priority: gene

GIN associates genomic interval queries (e.g your peaks) with the matching interrogated transcripts. The output, for each interval, includes the name and absolute chromosome coordinates of the assigned transcriptional units, as well as a call describing its relative position with respect to the transcribed unit (e.g. first exon, fourth intron, promoter) and the relative distance from the putative TSS (Transcription Start Site). Intervals that do not intersect any gene loci are annotated as “intergenic”.

The user can arbitrarily defines the length (in bps) of the putative promoter regions upstream to each TSS, setting the “Promoter definition” option (1, in the figure above).

GIN was thought to produces a not-redundant annotation table, meaning that each peak will have a unique annotation. Since quite often, overlapping of genes or gene features occur along

genome sequences (e.g. antisense transcripts, bidirectional putative promoters) prompting to an ambiguous annotation, the user have to give priority to the annotation of genes or of (putative) promoter regions. If promoter option is chosen, ***GIN*** tries to locate a peak in a promoter locus as first choice. If more than one promoter is found, the peak is associated to the closer transcriptional unit . If gene option is, instead, selected, ***GIN*** tries to locate a peak in an exon as first choice.
The program, in summary, tracks the flowchart depicted in the scheme reported below.



Together with the peak file, the second fundamental element that GIN needs is a proper “transcript annotation table” (see Appendix A of this manual for more information).

GIN usage:

- you can either upload your peak GFF File in the *Get Data/Upload File* Galaxy tool as “GFF” format (*See Section 3*) or just utilizing the output of ***PeakPeaker*** *(in this case, of course, you don’t need to upload anything, since the file is already in your History frame)*;
- upload the Transcript Annotation Table you want to use annotating your peaks (look at Appendix A for details);
- click the ***GIN*** link from the “CARPET: tiling analysis” tools list;
- select properly GFF file (1) and Annotation Table (2, in the second figure above) on the corresponding popup menu;
- set the requested parameters:
  - promoter definition (bp) (3, in the second figure above):
    - define the sequence length upstream the TSS you wish the program considered as putative promoter region;
  - annotation priority (4, in the second figure above):
    - promoter – the program tries to locate a peak in a promoter locus as first choice;
    - gene - ***GIN*** tries to locate a peak in an exon as first choice.
- click Execute;

Therefore the program produces, as output, an annotation table similar to that show up below.

*C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
chr	peakStart	peakEnd	peakScore	GeneID	GeneName	txStart	txEnd	Strand	# exons	relativeAnnotation	Distance TSS
chr19	22122	22313	5.95	AK311358	AK311358	21651	22626	-	2	exon 1	408
chr19	293015	293157	4.8120508	BC028203	BC028203	256574	295791	-	14	intron 2	2705
chr19	457488	457777	5.248068	BC009520	BC009520	458610	470653	+	2	promoter	-977
chr19	458174	458648	5.6043227	AK126170	AK126170	458496	471516	+	4	intronexon 1	-85
chr19	458947	459530	6.9007683	AK125401	AK125401	458499	461372	+	1	exon last	739
chr19	483767	484301	6.3392777	AK024373	AK024373	389420	2034745	-	19	intron 15	1550711

Each column respectively contain:

Chr: chromosome name (e.g. chr1, chrY);
peakStart: starting position of the peak defined by the first absolute genomic coordinate mapped on chromosome;
peakEnd: ending position of the peak defined by the last absolute genomic coordinate mapped on chromosome;
peakScore: the value of score or p-value derived from your peak query file;
GeneID: transcript ID;
GeneName: transcript name;
txStart: the starting position of the transcript expressed as the first genomic coordinate;
txEnd: the ending position of the transcript expressed as the last genomic coordinate;
Strand: strand direction of the transcript;

# exons: the number of exons of the transcript;
relativeAnnotation: annotation of relative matching position of peak on transcript;
Distance TSS: relative distance from the peak center to the transcript TSS.

Please NOTE: the output file doesn't contain a header line, therefore refer to this list to identity each column.

As a final point, you have associated your ChIP-chip enriched regions of binding to gene transcripts and their relative positions.

6.3. Peaks characterization: GIN visualizer

At this point

6.4. Peaks comparison: Common & Unique – Com&Uni

7. Working with expression tiling data

7.1. Expression chip annotation: Expression Notator – ENO

7.2. Analysis of Tiling expression data: Tiling Expression Analyzer – TEA

8. Comparing ChIP-chip and expression tiling data: Binding- Expression Correlation – BEC

9. References

Blankenberg, D., et al. (2007). A framework for collaborative analysis of ENCODE data: making large-scale analyses biologist-friendly. *Genome Res*, **17**, 960-4.

Giardine, B., et al. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Res*, **15**, 1451-5.

Scacheri, P.C., et al. (2006). Statistics for ChIP-chip and DNase hypersensitivity experiments on NimbleGen arrays. *Methods Enzymol*, **411**, 270-82.

10. Index

APPENDIX A: File Format and Tables

BED format

GFF format

This kind of file format is important in Galaxy and therefore in CARPET because represent one of the most common used one and because it is among the formats used from the Genome Browser to define its tracks.

GFF (General Feature Format) files have nine required fields that must be tab-separated. If the fields are separated by spaces instead of tabs, the track will not display correctly in the Genome Browser.

Here is a brief description of the standard GFF fields:

Seqname	The name of the sequence. Must be a chromosome or scaffold.
Source	The program that generated this feature.
Feature	The name of this type of feature. Some examples of standard feature types are "CDS", "start_codon", "stop_codon", and "exon".
Start	The starting position of the feature in the sequence. The first base is numbered 1.
End	The ending position of the feature (inclusive).
Score	A score between 0 and 1000. If the track line useScore attribute is set to 1 for this annotation data set, the score value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). If there is no score value, enter ".".
Strand	Valid entries include '+', '-', or '.' (for don't know/don't care).
Frame	If the feature is a coding exon, frame should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be '.'.
Group	All lines with the same group are linked together into a single item.

Example:

Here's an example of a GFF-based track.

C1*	C2	C3	C4	C5	C6	C7	C8	C9
#Seqname	*Source*	*Feature*	*Start*	*End*	*Score*	*Strand*	*Frame*	*Group*
chr19	my_chip	feature_1	10000000	10001000	500	+	.	TG1
chr19	my_chip	feature_2	10010000	10010100	900	+	.	TG1
chr19	my_chip	feature_2	10020000	10025000	800	-	.	TG2

* C1, C2, C3, ... C9 represent columns 1, 2, 3, … 9 and their relative position/number; a row beginning with "#" symbol is considered a comment. In the present example, the table is read as a matrix with 3 rows and 9 columns.

For further information on GFF format, have a look at http://www.sanger.ac.uk/Software/formats/GFF.

PAIR FILE Format

Nimblegen "Pair files" (http://www.nimblegen.com/products/methylation/data_guide.html) contain signal intensity data extracted from the scanned images of each array using NimbleScan™, the proprietary NimbleGen's data extraction software. Signal intensities for each probe are saved in Pair files (.txt). They contained eleven columns (C1, C2, C3, ... C11) but essentially the program considers only 3 of them: C10, that contains the chip raw signal, C6 and C7 that contain correspondent coordinate positions on x and y axes of the chip, respectively.

Example:

Here's an example of a PAIR FILE format.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
IMAGE_ID	GENE_EXPR_OPTION	SEQ_ID	PROBE_ID	POSITION	**X**	**Y**	MATCH_INDEX	SEQ_URL	**PM**	MM
1251702_635	FORWARD	CHR19	CHR1900P000011001	11001	565	381	64160375	<blank>	5459.89	0
1251702_635	FORWARD	CHR19	CHR1900P000011050	11050	610	656	64160376	<blank>	865.75	0

An exemplar of Nimblegen "Pair File" is here available for downloading *(zipped compressed, ~6.0Mb)*: it contains tiling data from an experiment of ChIP-chip on human chr19 (Human Mar.2006, hg18).

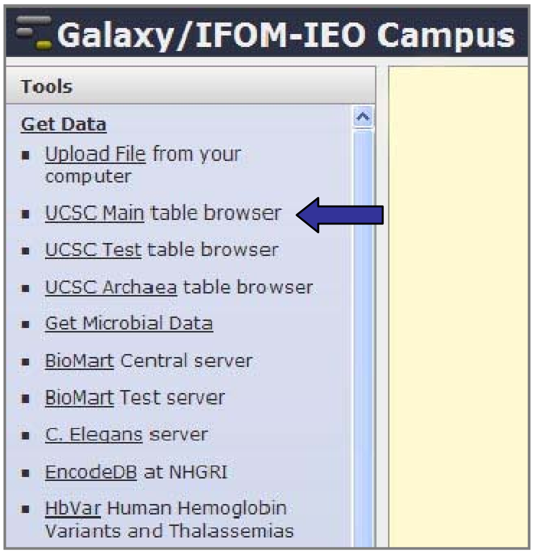
Transcript Annotation Tables

Transcript Annotation Tables used from the ***GIN*** tool to annotate user's GFF peak file can be:

- directly downloaded from the UCSC Genome Browser;
- derived from custom mapping information.

- From UCSC Genome Browser

Transcript Annotation Tables can be directly downloaded from UCSC Genome Browser by the *Get Data/UCSC Main table browser* of the Galaxy tool frame (see the figure below). For more information on how to manage with this task consult the detail



tutorial session at the official “Galaxy Screencasts and Demos” webpage: in particular have a look at the chapter “1. Interface, Screencast 1.1 - Introduction to Galaxy interface”.

When you finally will download the file table from the UCSC site, pay attention to choose the "all field from selected table" output format and to check the "send output to Galaxy" option.

It is possible to download many different annotation tables coming from different organisms and database (e.g. RefSeq, UCSC gene, FlyBase, EST).

An example of Transcript Annotation Table is here available for downloading *(.txt file, ~400.0Kb)*: it contains the Annotation Table of the human RefSeq of the chromosome 19, relative to the Human Mar.2006, hg18, genome assembly.

- From custom mapping information

If you, alternatively, would like to derived the Transcript Annotation Tables from custom mapping information, you have to strictly respect the following constrains.

Custom Annotation Table format must be formatted as follow and, **NOTE, must contain exactly the name fields listed**.

chrom - The name chromosome name (e.g. chr1, chrY).
chromStart - The starting position of the annotated feature in the chromosome. (The first base in a chromosome is numbered 0.)
chromEnd - The ending position the annotated feature in the chromosome, plus 1 (i.e. a half-open interval).
name - The name you want to give to your annotation in the BED line/track.
strand - Defines the strand, either + or - .
blockCount - The number of blocks (exons) in the BED line/track.
blockSizes - A comma-separated list of the block sizes. The number of items in this list should correspond to blockCount.
blockStarts - A comma-separated list of block starts (e.g. start of each exon of a transcript). All of the blockStart positions should be calculated relative to chromStart. The number of items in this list should correspond to blockCount.

The Annotation Table must always have headers. Look at the next example.

*C1	C2	C3	C4	C5	C6	C7	C8
chrom	chromStart	chromEnd	name	strand	blockCount	blockSizes	blockStarts
chr19	61678	62596	transcript_1	+	1	918	62596,
chr19	232043	242435	transcript_2	-	6	494,177,58,278,152,151,	232537,233310,233809,238751,239171,242435,
chr19	414359	425983	transcript_3	+	4	1005,114,108,363,	414359,418648,423393,425620,

* C1, C2, C3, ... C8 represent columns and their relative position/number, 1, 2, 3, … 8.