

CLANS Help  
Tancred Frickey 2006

First work on this program began in an attempt to gauge the quality of multiple sequence alignments. The basic idea was to use self-organizing maps to determine whether sequence groups the multiple alignments were based on could be recovered using other means. Shortly after a preliminary version of the program was completed, our group undertook the classification of a large family of proteins and this prompted a major rewrite. The reason was that the group we were trying to analyze contained so many members, that traditional multiple-sequence-alignment and subsequent phylogenetic tree inference could not be used. Instead we coupled the self-organizing map to BLAST results and very rapidly were able to determine the groups of interest from within the large set of sequences.

Since the development of the first version of CLANS a number of requests for additional features have been integrated into the program.

The old version was (almost) exclusively meant for analysis/filtering of sets of protein sequences and was fairly memory and computation intensive;

This version sports a number of rewrites and features that decrease memory use, increase speed and make clans usable for a wide variety of other data as well.

The most noticeable changes are:

- Use of the <att> </att> tags in the savefile allows users to enter attraction values instead having to simulate P-values by entering the -log(attraction\_value).

- Use of the <weight> </weight> tags in the savefile allows weights for the individual vertices/sequences to be specified.

- Extension of the "edit sequence groups" window. Now both color and shapes can defined for a sequence group.

- A new "Microarray data" window that permits visualization of expression values/foldchanges for the various sequences.

- A new "Functional mapping" window that overlays functional groups on to the current graph or overlays selected groups of sequences on to the functional categories (based on MAPMAN or KEGG-pathways).

### **Prerequisites for running CLANS**

- A Java1.5 (or better) runtime environment needs to be installed (allows loading of savefiles)

### **necessary for full functionality:**

- A working installation of the NCBI BLAST/PSIBLAST suite (if you want clans to do the BLAST searches for you)

### **possible extension:**

- the expr2clans java program. (permits conversion of expression data to a clans-savefile with pairwise correlation values)

- files containing the corresponding expression values (format below)

- a file containing a hierarchical grouping of sequences (i.e. mapman or KEGG functional classification)(see below for the exact format)

- a lookup file correlating the identifiers used in the expression files with those present in the mapping file

### **Input** (one of the following):

- A file with FASTA format sequences

- A CLANS savefile (format: see below)

- A CLANS matrix file (format: see below)

- A BioLayout input file (some data will not show (colors & defined groups))

### **NOTE: A TUTORIAL IS AVAILABLE SEPARATELY**

Clans is a program that takes multiple FASTA format sequences as input, performs an all against all BLAST search and displays the sequence similarities in a 3d-graph.

The BLAST P-values are used to calculate attractive forces between each sequence pair (the lower (better) the P-value, the higher the attractive force). In addition each sequence repulses every other with a certain force. Clustering is achieved by iteratively moving sequences according to the force vector resulting from all pairwise interactions (i.e. take sequence 1, sum all pairwise interactions & remember force vector; take sequence 2, do the same; repeat for all other sequences; once all sequences have their force vector calculated, move each according to their force vector; repeat all of the above for round 2, taking into account the new sequence positions; repeat until the map does not change any more).

Using the conversion program expr2clans, it is now also possible to cluster and visualize the pairwise correlation of probe-sets in microarray experiments.

---

### **Settings:**

#### **Command line options:**

all these options can be found in the file: clans.conf. Command line parameters supercede those specified in the conf file.

A standard command might look like: java -Xmx300m -jar clans.jar -infile input\_file -blastpath "blastall -p blastp" -formatdbpath "formatdb" -eval 1 -pval 0.1

(this will try to compare all sequences present in "input\_file" using the program called "blastall -p blastp" and return all HSP's up to E-values of 1 or P-values of 0.1 whichever is more stringent)

or, to load a saved data file: java -jar clans.jar

(the latter only starts the graphical user interface but loads no data. This will expect a savefile to be present somewhere on the disk. To load it use the 'Load Run' item in the 'File' menu of the graphical user interface and select the corresponding file.)

#### **java Options:**

java	run the following using the java interpreter
-Xmx300m	allow this program to use a maximum of 300 MB of memory (default is ~70MB on a windows 32 bit system)
-jar	program to run is a "jar" archive

---

#### **Program options:**

##### **basic:**

-?

prints basic help for the parameters to the command line.

-conf

name of the configuration file

Default:-conf clans.conf

-infile

name of the input file containing the FASTA format sequences to plot

-infile filename

-blastpath

path to the blast executable (NCBI blast) and some blast settings (NOTE: eval to be used in blast searches should NOT be entered here!)

-blastpath "blastpgp -j 3 -h 0.001"

-formatdbpath

path to the formatdb executable (NCBI) used to format the blast database

-formatdbpath formatdb

-eval

extract BLAST HSP's up to E-values of (SPECIFY the E-value HERE!)

-eval 10

-pval

extract BLAST HSP's up to P-values of

-pval 0.1

-verbose

verbose mode (0(no output)-10(flood the screen))

-verbose 1

**advanced:**

-cmd

a string to prepend to all commands (i.e. "nice 19" (unix) or "cmd" (windows))

default: ""

-referencedb

alternate reference databases you want to search (with blastpgp) to build a profile (uses this database for all but the last round of a PSIBLAST search. Last round is performed on the database of sequences that was specified in -infile

default: "" (perform all blast runs on the -infile database)

-cpu

number of CPU's to use on this machine (Threaded)

-cpu 1

-readblast

read results from a former run (saved to the file "tmpblasthsp"); (useful for crash recovery)

-readblast T

-savepos

save the positional information while clustering (slower, but clustering can be recovered after a crash)

-savepos F

-docalc

do calculations or just show the interface (use "F" if you want to load a saved file via the GUI)

-docalc T

-olddata

read old data from a savefile

default: "" (don't read any old data)

-newseqs

sequences present in newseqs are added to olddata and clustered

default: "" (don't add any new sequences)

-enrichseqs

take the sequences from newseqs and enrich them with related sequences (from referencedb)

default: F (don't enrich new sequences with additional representatives, use file as it is)

-gatherseqseval

get sequences (for enrichment) up to evalues of

-gatherseqseval 1e-10

-rmseqseval

do not add sequences (when enriching newseqs) with evalues better than X (sequences would be too similar and thus not very informative)

-rmseqseval 1e-20

-maxenrichseqsnum

enrich each newseq by at most X additional sequences

-maxenrichseqsnum 5000

-exhaustive

when adding new sequences to existing data; how should the pairwise blast hits be computed? 0=one way search, 1=re-search with the sequences that had hits with -newseqs and forget the rest, 2=redo all blast runs

-exhaustive 1

-nographics (t/F)

Setting this to true (T) will cause the program to perform all the blast runs and NOT start the clustering. (results are saved to "tmpblasthsp.txt") For clustering you can then re-start the program later (using -nographics F). It will automatically look for a "tmpblasthsp.txt" file and load those results.

---

Most of the options should be rather self explanatory, except for the -olddata and -newseqs related options. Here is a little more info on those:

If a saved run with many sequences has already been clustered, it would be wasteful to repeat an all against all blast search just because a few new sequences were added. The -olddata and -newseqs options allow extension of existing datasets by a few new sequences. With these flags, Blast runs will then only be performed for the new sequences. During clustering the old datapoints are held static and only the new sequences are allowed to move within the 3d-space. This makes comparison of old and new cluster maps easier.

The -enrichseqs option was added in case only a single sequence representative for a large group of sequences is provided. In this case the aim is to gather all sequences belonging to this group and then clustering a representative subset thereof, instead of using only the single sequence present in the input file. This requires a few more parameters:

-gatherseqseval: all sequences with HSP's better than this E-value are taken to be members of the group you want to enrich for.

-rmseqseval: Sequences with evalues better than this are taken to be too closely related to add significant information and only one sequence is retained (helps find a representative set and avoids overrepresentation of highly similar sequence groups)

-maxenrichseqsnum: Some large families of proteins might include so many sequences as to become unmanageable. This option sets the upper limit on the number of "enrichment" sequences to be gathered per sequence specified in -newseqs.

-exhaustive : once all sequences to add to the existing data are gathered you can set how pairwise blast scores are to be computed. 2= repeat the all against all blast search for all (old+new) sequences; 1= BLAST all new sequences and then re-BLAST only those old sequences which were found in these searches; 0= BLAST only the new sequences and assume BLAST new sequence X ->old sequence Y returns the same results as BLAST old sequence Y ->new sequence X.

---

## Graphical User Interface (GUI) Options

### MOUSE:

Press on gray outline & drag to rotate around the axis perpendicular to the screen.

Press & drag inside 3d graph window to rotate around a horizontal or vertical axis.

If BUTTON Select/Move is highlighted: Press & drag to select all sequences in a region.

Holding "shift" and dragging the mouse pans the view

Holding "alt" or "ctrl" or "meta" and dragging the mouse either deselects all sequences in selected region (if in selection mode) or moves the selected sequence according to the mouse movement (if in movement mode).

### BUTTONS:

-Initialize: randomize the sequence positions prior to clustering.

-Start run/Stop/Resume: Start, Stop or Resume a clustering run.

-Select/Move: toggle between selecting sequences & rotating the 3d-space using the mouse.

-Show selected sequences: open a window showing the names of the currently selected sequences.

-Zoom on selected: Focus on the selected sequences. NOTE: the other sequences are still displayed and part of the dataset. To actually remove unselected sequences from the dataset use: Menu->Misc->Use selected subset.

-Select all/Clear selection: Select all/none of the sequences

-Use P-values/attraction better than (text field): remove all connections between sequences with P-values/attractions worse than specified (text field next to it) from the graph and ignore them in the calculations. The reason the same functionality is present both as a textfield and a button is the following: when using x-forwarding on a solaris machine, the "return" key is sometimes not recognized and entering values in the text field will have no effect. Pressing the button causes the program to read the data from all text fields and updates the program accordingly.

#### CHECKBOXES:

- Show names:display the sequence names in the graph
- Show numbers: display the sequence numbers in the graph
- Show connections: Draw connections between all sequences with BLAST hits/attraction values better than the specified cutoff. Color of the line reflects the P-value for this connection.(and can be changed under Draw->Change color(connections))

#### TEXTFIELDS:

NOTE: The program disregards any changes made to the text fields until "return/enter" is pressed.

- Use P-values/attraction better than (text field): remove all connections between sequences with P-values/attractions worse than specified (text field next to it) from the graph and ignore them in the calculations. The reason the same functionality is present both as a textfield and a button is the following: when using x-forwarding on a solaris machine, the "return" key is sometimes not recognized and entering values in the text field will have no effect. Pressing the button causes the program to read the data from all text fields and updates the program accordingly.

#### MENU->

##### File->

- Load run: Load a CLANS savefile (or a BioLayout input file).
- Save run: Save current clustering to file.
- Add sequences: Do that from the command line via -olddata & -newseqs (see above).
- Save matrix values: Save only the matrix containing attraction values for the sequences.
- Save 2d graph data: Save the current position of all sequences (the 2d representation) & the sequence names to a file.
- Print: print the current view (scaled to fit on one page)
- Load matrix data: Load data from a file containing precomputed attraction values in a matrix.
- Append sequence groups from file: read predefined sequence groups from a file and append the groups to "sequence groups" (see savefile format and "menu->windows->edit sequence groups")

##### Misc->

- Extract selected sequences: write the full length sequence & name for the currently selected sequences (red dots) to a separate file.
- Hide singletons: remove all sequences with no BLAST hits to any other sequence from the graph.
- Use selected subset: Remove all but the selected sequences from the graph.
- Use parent group: Undo the last (or multiple) "use selected subset" or "hide singletons" steps.
- Set rotation values:Set the values for the current rotation matrix (9 values).
- Complex attraction: use only the best P-value to calculate attractive forces between two sequences (simple) or multiply the probabilities for all HSP's connecting these two sequences (complex).
- Optimize only selected sequences: cluster only the selected sequences; leave the rest static.(attraction & repulsion is still computed using ALL sequences).
- Cluster in 2D: cluster in 2 dimentions only
- Rescale attraction values: set the worst P-value as "no attraction"(0), the best as "best attraction" (1) and rescale all other values in between (can improve cluster detection)
- only draw every Nth round: can be used to speed up the clustering when displaying the data to screen takes an inordinate amount of time. Using this, only every Nth round is actually drawn to the screen.

##### Draw->

- Change Font: Change the font used in the graph display
- Set sequence dot size: sets the size of the dots representing a sequence in 3D-space.
- Set selected circle size: sets the size of the circle highlighting selected sequences.

- Change color (Connections): change the coloring scheme of the lines connecting vertices.
- Color dots by sequence length: Colors the sequences according to their length (yellow=shortest, blue=longest, gradient=in-between)
- Color by edge "frustration": color the edges in the graph according to whether they are longer (red) or shorter (blue) than they should be according to the attraction values in the matrix.
- Change color (Foreground): change the foreground color (default: black);
- Change color (Background): change the background color (default: white);
- Change color (Selected sequences): change the color of the ovals highlighting selected sequences.
- Change color (BLAST hit numbers): change the color of the sequence numbers for those with BLAST hits (only if menu->windows-> show BLAST hits for sequence: was used)
- Change color (BLAST hit circles): change the color of the circles highlighting sequences with BLAST hits (only if menu->windows-> show BLAST hits for sequence: was used)
- Show origin: Draw a red "X" at the origin (0,0,0)
- Show info: Show information about the current clustering (bad->good blast HSP coloring, BLAST command line used, maximum x&y coordinates, current rotation matrix)
- Show names while selecting: While selecting, pop up a window showing the names of the currently selected set of sequences.
- Show HSP sequence numbers: Show the sequence numbers for those sequences that were found via the current blast search (only if menu->windows->show blast hits for sequence)
- Zoom: set a zoom factor for the view (default: 100%; fits all vertices on to the screen)
- Center graph: set the current view on the center of the graph.
- antialiasing: enable antialiasing (nicer graphics, but slower)

#### Windows->

##### Show options window:

This is a window combining parameters influencing clustering.

- Cooling: A multiplier for maxmove (see below). Every round "current\_cooling" is multiplied with "cooling" and then multiplied with maxmove to define how far each dot may move in that round. If set to <1, maxmove slowly converges to 0 (no further movement); if >1, Maxmove slowly rises to infinity. Both cooling and current\_cooling can be set.
- Dampening: If dampening=0, the movement vector this sequence had in the last round has no influence on this round's movement. Dampening=1; last round's movement counts as much as this round's force vector (they are simply added); (0<x<1; any value in-between states how much the movement of last round influence this round's movement? (similar to assigning a mass to the dots))
- Maxmove: maximum distance a point is allowed to travel per round. The reason there also is a button is that when x-forwarding between machines sometimes the "return" key is not recognized. pressing the button tells the program to re-read all text-field values and update the variables.
- Attract value : Multiplier for the attractive forces between two sequences.
- Attract exponent : determines how the attractive force scales with distance (default=1, attraction increases linearly with distance). The higher the exponent is set, the less the actual attraction value of a pair matters. For large exponents it only matters whether or not a connection is present. (increasing the exponent for a few rounds can help resolve the problem of dots being caught in local minima)
- Repval value : Multiplier for the repulsive forces between two sequences.
- Repval exponent : determines how the repulsive force scales with distance (default=1, repulsio drops off linearly with distance.
- Min. attraction: A minimal force that attracts each sequence towards the origin of the graph (keeps unconnected clusters/dots from drifting apart into infinity). (scales linearly with distance)
- cluster for rounds: will perform exactly X rounds of clustering (-1=infinite; has to be stopped via the "stop" button)

NOTE: using the Attract value and exponent you can influence the compactness of the clusters and their relative distances.

- Sequences: Opens a window showing the sequence numbers and names (more, see below)
- P-value plot: Opens a window showing the cumulative distribution of P-values/attractions for this dataset.
- Show BLAST hits for sequence: Select a sequence and re-BLAST against the current dataset.



```

qname: sequence0          #query name: sequence0
qseq:PRKIPNPDFFEDLEPF    #query sequence present in HSP
hname: sequence2         #hit sequence name: sequence2
hseq:ICADRRIVDDWANDGW    #hit sequence present in HSP
qstart: 1                #HSP start in full length query sequence
qend: 301                #HSP end in full length query sequence
hstart: 1                #HSP start in full length hit sequence
hend: 301                #HSP end in full length hit sequence
pvalue: 1.821734248190753E-151 #P-value for this HSP

```

etc. for all HSP's for all sequences.  
 EOF

### CLANS\_savefile:

This is what a CLANS savefile looks like:

```

sequences=78             #total number of sequences in this file
<param>                  #a param block is optional
maxmove=0.05            #the maximum movement permitted in this clustering
pval=1.0E-70            #P-value cutoff used for this clustering
cluster2d=true/false    #was this clustering done in 2D or 3D?
</param>
<rotmtx>                 #the current rotation matrix for this clustering (up to </rotmtx>) (optional)
1.0;0.0;0.0;
0.0;1.0;0.0;
0.0;0.0;1.0;
</rotmtx>
<seq>                   #from here to </seq> come the sequences contained in this clustering in fasta
format (MUST-HAVE) (if you are not interested in the sequence data, just enter the names of whatever you are
interested one after the other; names MUST start with a ">" symbol.
>sequence0
PRKIPNPDFFEDLEPFPRKIPNPDFFEDLEPF...
>sequence2
...
...etc.
</seq>
<weight>                #the weight block is optional and makes it possible to assign a weights to each of the
vertices. This weight influences the attractive and repulsive force of the vertex. If unspecified, all vertices have
the weight 1.
>sequence0
0.7
>sequence1
1.3
etc.
</weight>
<seqgroups>             #optional; defined groups of sequences that should be colored/highlighted in a
specified manner
name=group_name         #the name of this group (any identifier)
type=integer            #default 0=circle (can be changed via the "edit sequence groups dialog")
size=integer            #default 5 (size of circle)
color=int;int;int       #the red;green;blue components of a color (0-255)
numbers=int;int;int...  #the sequence numbers to add to this group
name=group_name2       #group2
type=integer2
size=integer2
color=int2;int2;int2
numbers=int2;int2;int2...
name=group_name3       #group3
type=integer3

```

```

size=integer3
color=int3;int3;int3
numbers=int3;int3;int3...
</seqgroups>
<pos> #the positions for the sequences (up to </pos>) optional; if left out, all
sequences are assigned the position (0,0,0) (pressing the "initialize" button in the GUI will randomly place
them in 3D)
0 3.015268 -1.3726062 -0.82156813 #sequence 0 X=3.015268 Y=-1.3726062 Z=-0.82156813
1 -0.79675895 1.9573041 1.6266986
etc.
</pos>
<hsp> #the connections between sequences (MUST-HAVE) (format: seed_sequence
hit_sequence:value1,value2,etc.)(multiple hits possible)
0 2:4.209705224465372E-92 #HSP P-values from sequence 0 to sequence 2 is 4.209705224465372E-
92.
0 3:8.419410448930744E-42
0 4:4.2097052244653725E-55 0.022451761197148652 0.049113227618762675 #here multiple HSP's were
present for seq0=query, seq4=hit
0 5:4.2097052244653715E-20
0 6:4.209705224465372E-53
0 7:4.209705224465372E-61
0 8:0.0 #no blast hits found in this case
0 9:4.209705224465372E-61
0 10:2.806470149643581E-95
0 11:4.209705224465372E-93
0 12:5.612940299287162E-53
etc.
</hsp>
EOF

```

#### Attraction matrix savefile:

This file contains the pairwise BLAST -log P-values

Format:

sequences=number\_of\_sequences

<seqs>

sequence\_names

</seqs>

<mtx>

matrix of pairwise attraction values (in the order the sequence names are given) individual values are separated by spaces.

</mtx>

EOF

#### 2d-graph savefile:

This file is written in the format:

ID	NAME	X	Y
0	sequence0	0.5	0.376

#sequence id, sequence name, x-axis coordinate, yaxis

coordinate

etc. for all sequences

EOF

#### Extentions:

Since developing the first version of clans, I have come across a multitude of other projects to which clans could be applied. In most cases this involved moderate extentions of the old code (i.e. adding color and shape features to specified sequence groups, loading predefined groups from another file and overlaying data from

another clans file on to the current graph).

### Microarray data

(Menu->windows->microarray data)

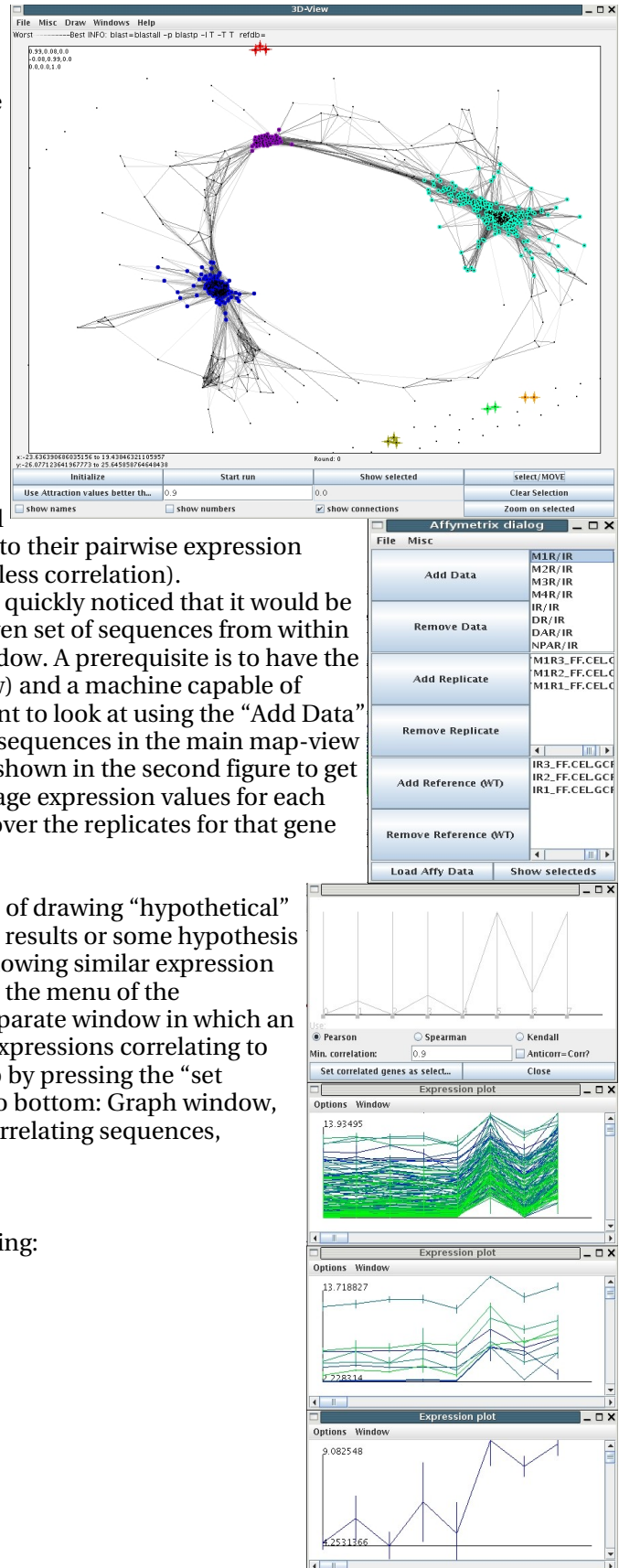
Finding sets of coexpressed genes is very similar to finding sets of similar proteins. In both cases you have lots of pairwise data (i.e. BLAST-similarities or correlation-coefficients for the expression values) and want to get the sets of sequences that are similar. Using the program expr2clans it is quite simple to convert Affymetrix or GCRMA-normalized expression data to a clans save-file that can then be analyzed. expr2clans lets you specify what datasets you want to compare, how you want to filter the data (i.e. exclude noisy and unreliable data, genes that don't change over the experiments, etc.) and what type of correlation you want to use to determine the groups (i.e. linear correlation or rank-order or ... (look at the program for all options)) Once the CLANS savefile is created you can simply load it in clans and start analyzing the groups. (top figure; The dots correspond to single genes, the lines connecting them correspond to their pairwise expression correlation. Dark lines=high correlation, lighter lines=less correlation). Once I actually started working with microarray data I quickly noticed that it would be helpful to be able to see the expression values for a given set of sequences from within clans. This is possible from within the microarray window. A prerequisite is to have the expression data in a clans-compatible form (see below) and a machine capable of handling the amount of data. (Include the data you want to look at using the "Add Data" and "Remove Data" buttons. (once you have selected sequences in the main map-view (top figure) simply press the "show selecteds" button shown in the second figure to get an expression plot (3<sup>rd</sup>. Figure); lines connect the average expression values for each condition, vertical lines show the standard deviation over the replicates for that gene for that condition).

A short while later I was also asked to include a means of drawing "hypothetical" expression plots (for example based on real-time-PCR results or some hypothesis of gene-interaction) and using that to find all genes showing similar expression patterns. This is accessible under the "Misc" option in the menu of the microarray window (second figure) and will open a separate window in which an expression graph can be drawn. The sequences with expressions correlating to that graph can be highlighted in the main CLANS map by pressing the "set correlated genes as selected" button (right; from top to bottom: Graph window, linear correlating sequences, Spearman rank-order correlating sequences, Kendall's tau rank-order correlating sequences)

### Format for microarray expression data:

for each replicate you will need a separate file containing:

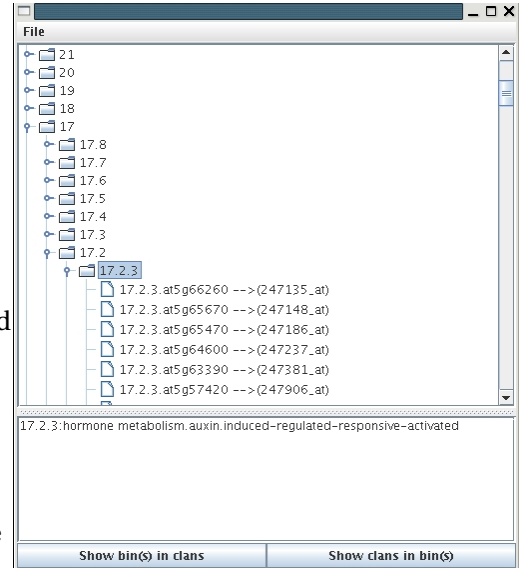
```
name_gene_1      expression_value
name_gene_2      expression_value
etc.
```



## Functional mapping

(Menu->windows->functional mapping)

Finding groups of co-expressed genes is very nice, but not very helpful when trying to determine what is going on at the molecular level. For that it is helpful to quickly reference back to annotation information or, even better, sth. that gives both functional information for each gene and also what pathways it may be involved in (i.e. MapMan or KEGG pathway information). As I was working on *Arabidopsis thaliana*, I included the very nice MapMan annotation and then extended it further, allowing for GeneBins annotation files as well (which are based on the KEGG pathways and available for more organisms). The figure on the right shows the functional annotation window. The MapMan classification tree is displayed. The tree consists of a number of bins. Each major pathway has a separate "bin" (1-35) and can be split into sub-bins, sub-sub-bins, etc. until reaching the actual genes (in the above example: bin 17 (hormone metabolism) has a sub-bin 17.2 (Hormone metabolism.auxin) which has a sub-bin 17.2.3 (hormone metabolism.auxin.induced-regulated-responsive-activated) which contains a number of genes (247135\_at and a few others). Selecting one or multiple bins or genes and pressing the "show bins in clans" button will look for all genes present in these bins and highlight them in the clans map. Pressing the "Show clans in bins" button will do the reverse, i.e. show where the currently selected sequences in the clans map fall in the classification tree (for example to find out if a group of co-expressed genes are involved in the same metabolic pathway or not.).



To be able to use this extension you have load a file containing the annotation information (for example a Gene-ontology, MapMan or GeneBins file). A Second file containing a lookup telling CLANS what clans identifiers (in this case the same as the affymetrix identifiers) correspond to what gene-names in the classification (here: At-numbers (i.e. 247135\_at corresponds to At5g66260)) may be necessary if the classification identifiers no not correspond to those used in CLANS.

### Format for the classification file:

Bin-number;Bin-name;;	for defining a metabolic bin
Bin-number;Bin-name;Gene_name;other_data	for defining a specific gene
Example:	
'3';'minor CHO metabolism';;	bin 3
'3.1';'minor CHO metabolism.raffinose family';;	bin3.1
'3.2';'minor CHO metabolism.trehalose';;	bin3.2
'3.3';'minor CHO metabolism.sugar alcohols';;	bin3.3
'3.3';'minor CHO metabolism.sugar alcohols';'At5g51970';'sorbitol dehydrogenase, putative / L-idoitol 2-dehydrogenase, putative  similar to NAD-dependent sorbitol dehydrogenase from Malus x domestica (gi:4519539)'	gene within bin 3.3
etc.	

### Format for the lookup file

mapman/genebins_name1	clans/affymetrix_name1
mapman/genebins_name2	clans/affymetrix_name2
etc.	

Example:

At1g01010	261585_at
At1g01010	261586_at (the same mapping identifier can be assigned to multiple clans identifiers)
At1g01030	261568_at
At1g01040	261584_at
At1g01050	261579_at
At1g01060	261569_at
At1g01070	261576_at

