Kiwi 1.0 walkthrough

Created by: Obi L. Griffith
Last modified: April 3, 2007

The KiWi 1.0 software implements an algorithm first reported in:

Gao BJ, Griffith OL, Ester M, Jones SJ. 2006. Discovering significant OPSM subspace clusters in massive gene expression data. In Proceedings of the 12th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Philadelphia, PA, USA, August 20 - 23, 2006). KDD '06. ACM Press, New York, NY, 922-928.

The software was released together with a manuscript for submission to Bioinformatics:

Griffith OL, Gao BJ, Bilenky M, Prychyna Y, Ester M and Jones SJM. KiWi: A scalable subspace clustering algorithm for the identification of coregulated genes from massive gene expression datasets.

The source code, supplementary materials, and other documentation can be found at:

http://www.bcgsc.ca/platform/bioinfo/ge/kiwi

Or

http://www.cs.sfu.ca/~bgao/personal/

General tips for running KiWi

- KiWi makes possible the identification of subspace clusters (specifically, order preserving sub-matrices) from very large data matrices.
- The number and quality of clusters and runtime required for KiWi to produce results is directly related to several parameters that the user specifies as well as the size of the dataset being analyzed.
- In the optimizing parameter section we will discuss how to go about choosing these parameters.
- The user should experiment with different parameter settings to get an estimate of runtime that will maximize the result set while still completing in a reasonable amount of time.
- We recommend using a multi-core and/or hyper-threading capable system. KiWi is not coded for parallel processing but using a multi-core system will allow you to run other programs simultaneously without interference with KiWi.
- For larger datasets (>1000 rows and/or columns) and/or large values of k (>10,000) we also recommend having a large amount of RAM memory (2 to 4gb recommended).
- Some steps of the KiWi program may require several hours to complete (for larger datasets and/or parameter settings). During these steps, the program may be unresponsive and appear to be frozen or crashed but is in fact running normally. As long as the process is active (as reported in Windows task manager) leave it running and it will eventually complete. If this becomes too long of a time period, kill the application and restart with different parameter settings (e.g. smaller k).

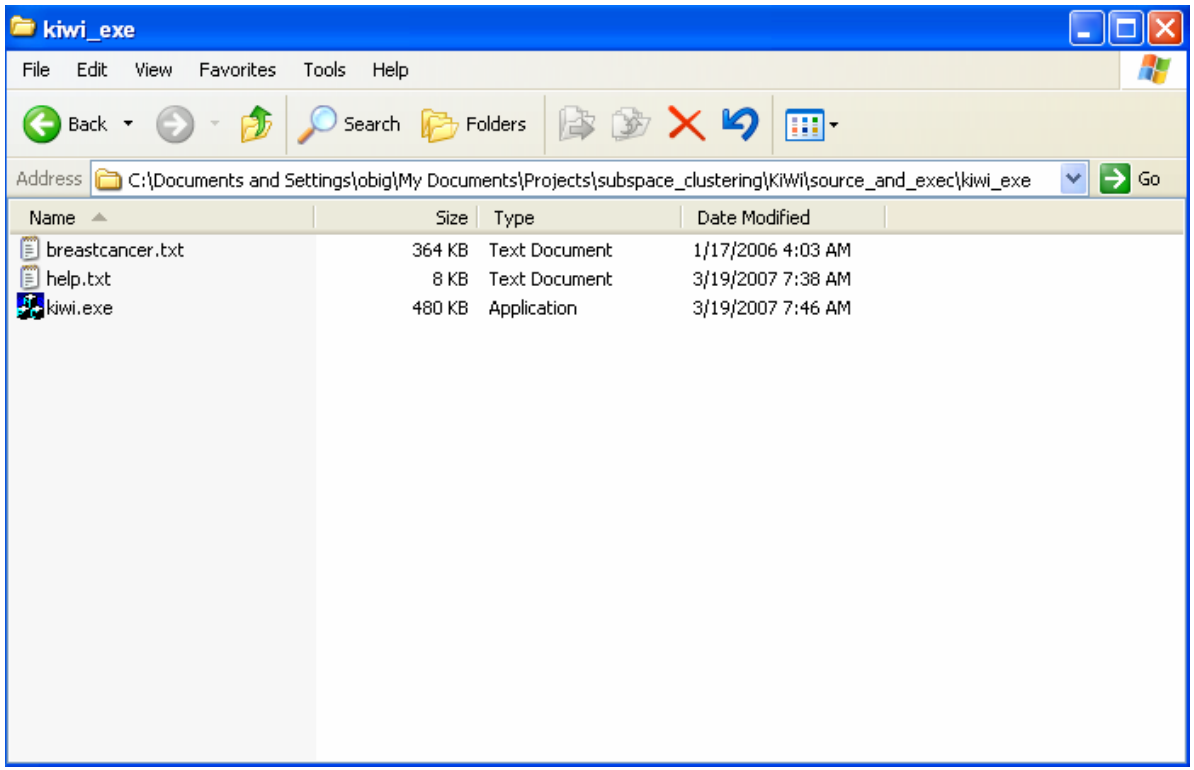
I) Data input formats

a) KiWi accepts any tab-delimited matrix of data.
b) Values should be continuous (i.e. not categorical, Boolean, etc)
c) Row and columns should be labeled (e.g. with experiment labels and gene ids)

Sample data: Matrix of 3 genes x 3 experiments

Mutation	BRCA1	BRCA1	BRCA1
HK1A1	0.15	0.22	0.3
HK1A2	1.54	1.27	0.76
HK1A4	1.72	1.57	2.13

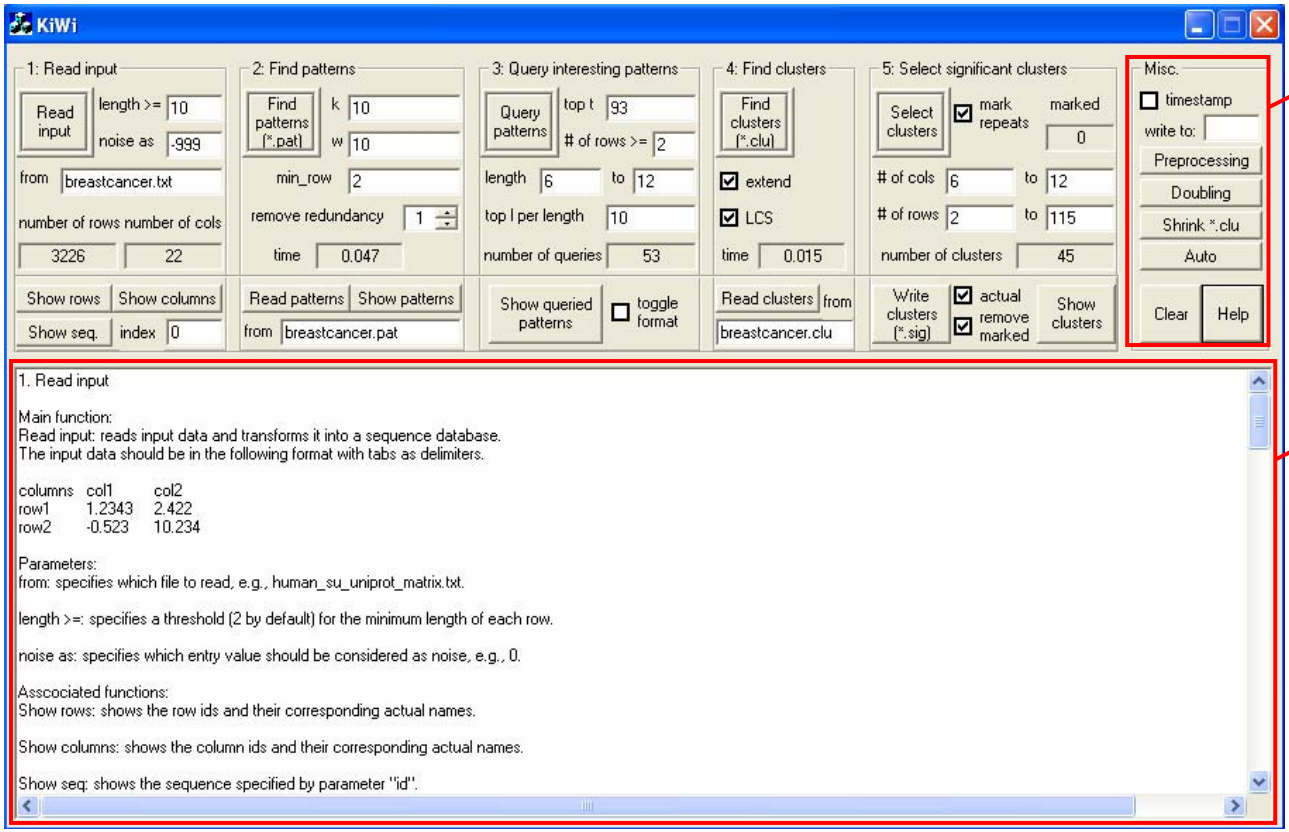
II) Starting KiWi



a) Create a new folder and put kiwi.exe, help.txt and your datafile (e.g. breastcancer.txt) in the folder.

b) Launch KiWi by double-clicking kiwi.exe

III) Miscellaneous options



Miscellaneous options

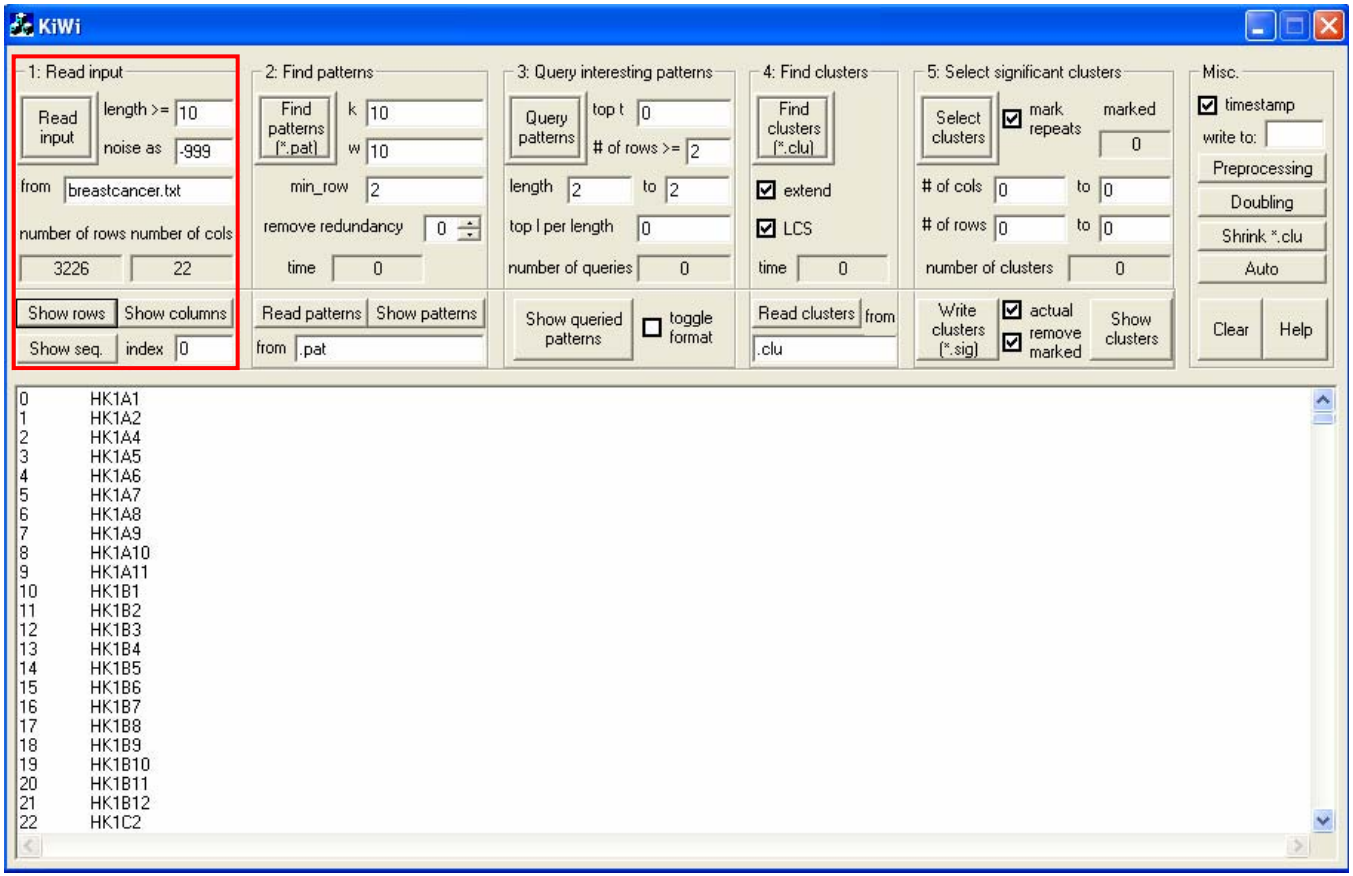
Display window

a) Uncheck ‘timestamp’ if you do not want timestamp in output file names

b) At any time use the ‘Help’ button for detailed documentation in the display window. Use ‘Clear’ at any time to clear the display window.

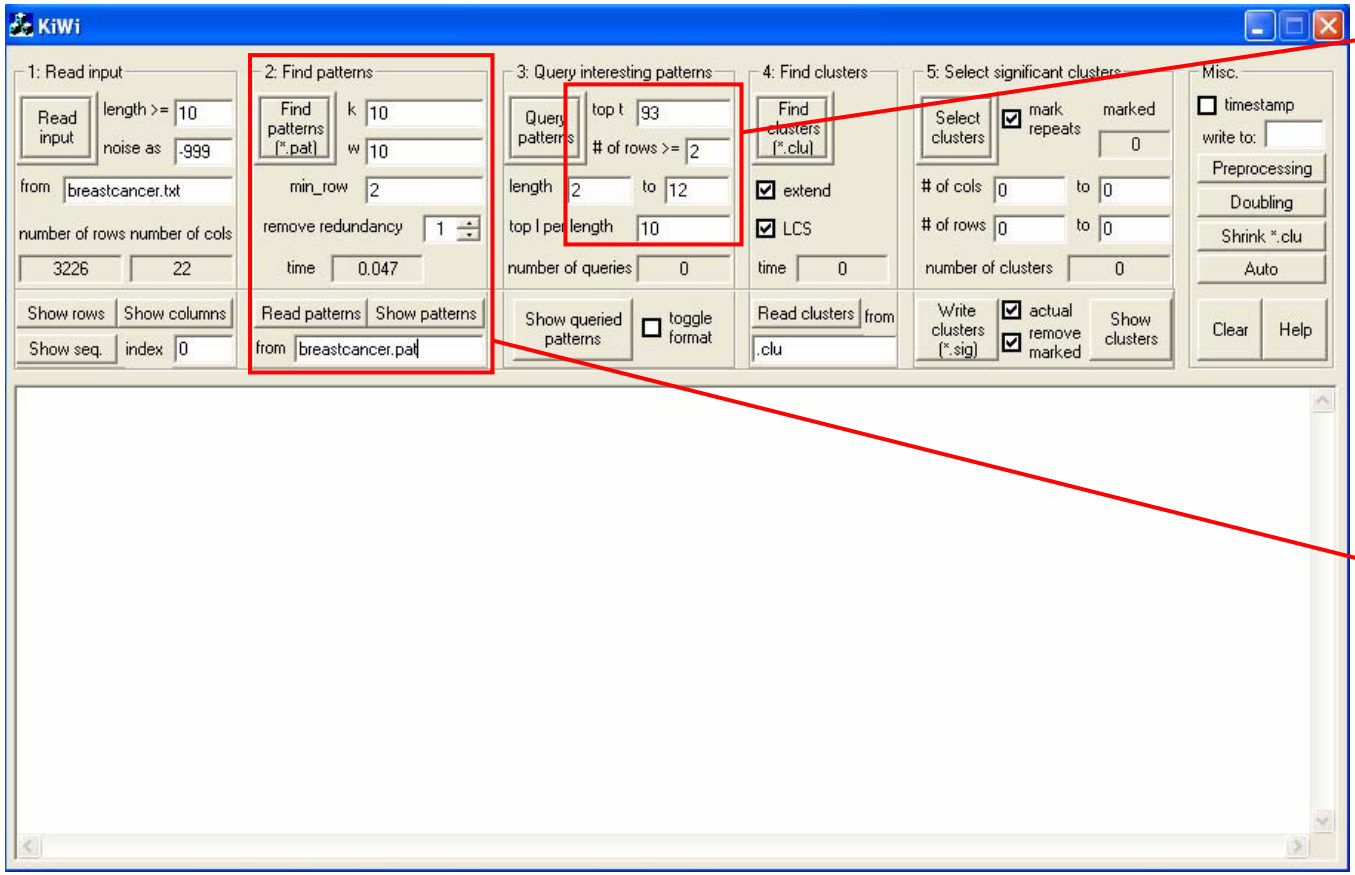
c) For explanations of ‘Preprocessing’, ‘Doubling’, ‘Shrink’ and ‘Auto’ functions see help document.

1) Reading data into KiWi



a) Specify the minimum length allowed for a row in the ‘length >=‘ field. If there are missing values in the data, this will eliminate rows with too few datapoints.
b) Specify the value to be considered ‘noise’ or missing value (0, NA, -999, etc)
c) Enter file name for data to be analyzed in the ‘from’ field.
d) Click the ‘Read input’ button.
e) Use the ‘Show rows’ and ‘Show columns’ to confirm that data were input correctly

2) Finding KiWi patterns



After finding patterns, results are summarized here. In particular, note the maximum length of pattern found. This will be important for optimizing w.

Also, note the time required for pattern discovery. This will be important for estimating a final k value to try.

a) Specify the desired values for k and w parameters (see next page for tips on optimizing k and w)

b) Set the 'min_row' parameter. This is the minimum number of rows required for a KiWi cluster (e.g. the minimum number of genes).

c) Set 'remove redundancy' to desired level. This option improves the quality of the k patterns on each level by removing some unpromising patterns. There are 6 choices, 0 ~ 5, the bigger the number, the slower. We suggest you choose 0 (not used) or 1 (better results).

d) Click the 'Find patterns'. Run time will be shown in the 'time' field.

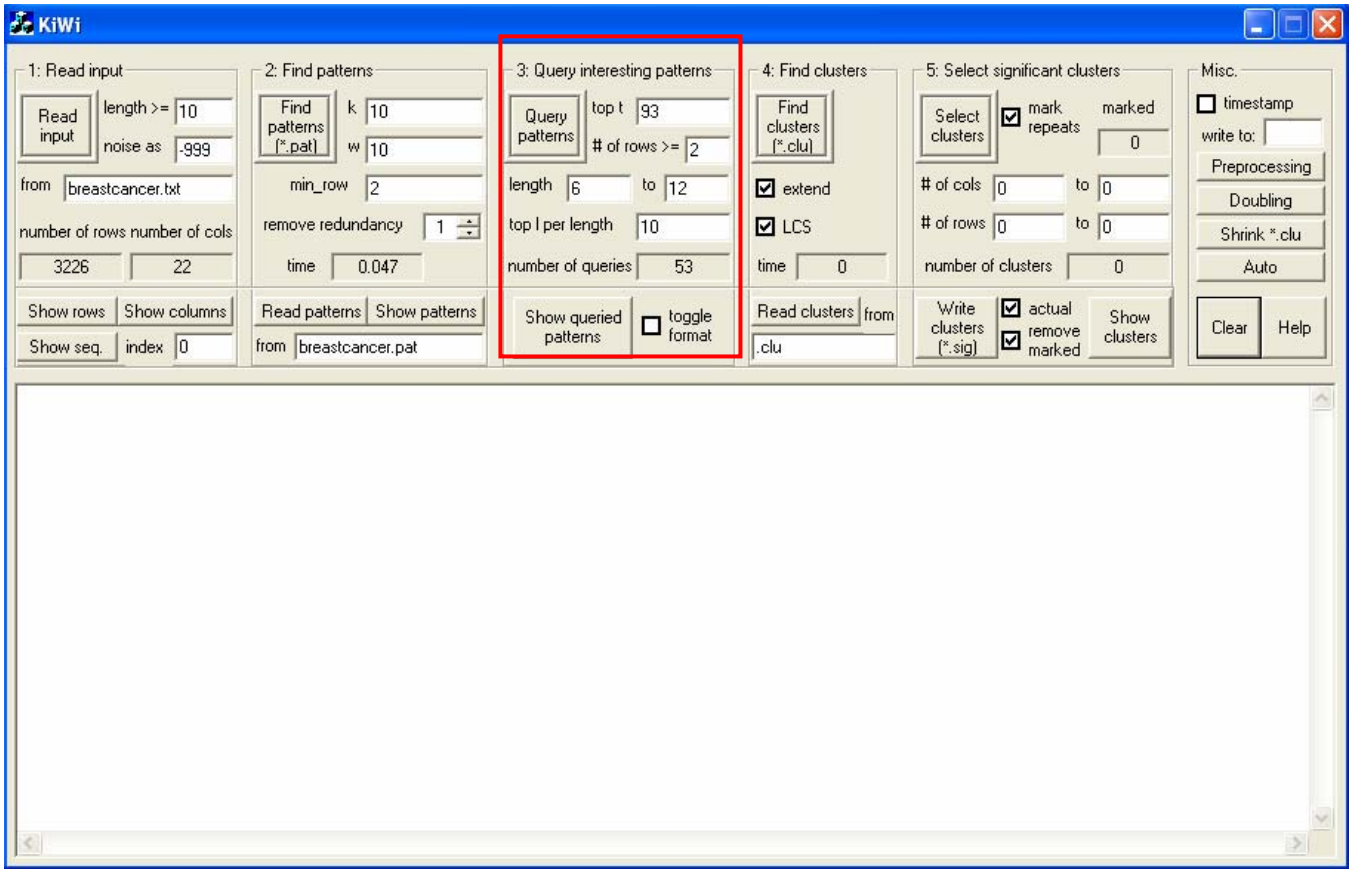
e) Use 'Show patterns' to visualize the patterns found in the display window.

f) Use 'Read patterns' to load a pattern file from a previous kiwi run.

Optimizing pattern discovery parameters

- As discussed above, it is often necessary to experiment with KiWi parameters in order to balance runtime against quality and quantity of discovery results.
- There are several parameters to be considered.
  - ‘w’ does not have a straightforward relationship to runtime but does affect the quality of results in terms of the length of patterns discovered. The smaller the w, the shorter the runtime for each round, where the number of rounds is the length of the discovered pattern. So in general, bigger w should lead to longer runtime. However, if w is too big (significantly bigger than the ‘best choice’), KiWi cannot generate long patterns. In such cases, the software terminates after only a couple of runs and the total runtime could actually be shorter than a smaller w. If you wish to find long patterns, experiment with w to maximize pattern length for a static value of k that does not take too long to complete (e.g. k=1000). Then use this value of w with a larger value of k.
  - ‘k’ ultimately determines the number of patterns discovered. Once other parameters are finalized, choose the largest k that you think will return results in a reasonable amount of time. For our datasets of ~1000 experiments and ~10,000 genes we often try k=100,000 and allow the program to run for a day or two.
  - ‘min_row’ affects both runtime and the results produced. This parameter should be set solely according to the user’s interest. In our case, we are interested in clusters as small as 2 genes (min_row=2). If only clusters of size 10 or greater are of interest then use min_row=10 to reduce runtime.
  - ‘remove redundancy’ improves quality at the expense of runtime. We normally recommend using a value of 1. But, if runtime is a major issue you might try 0.

3) Query patterns



a) If desired, you can again limit the minimum number of rows for a cluster

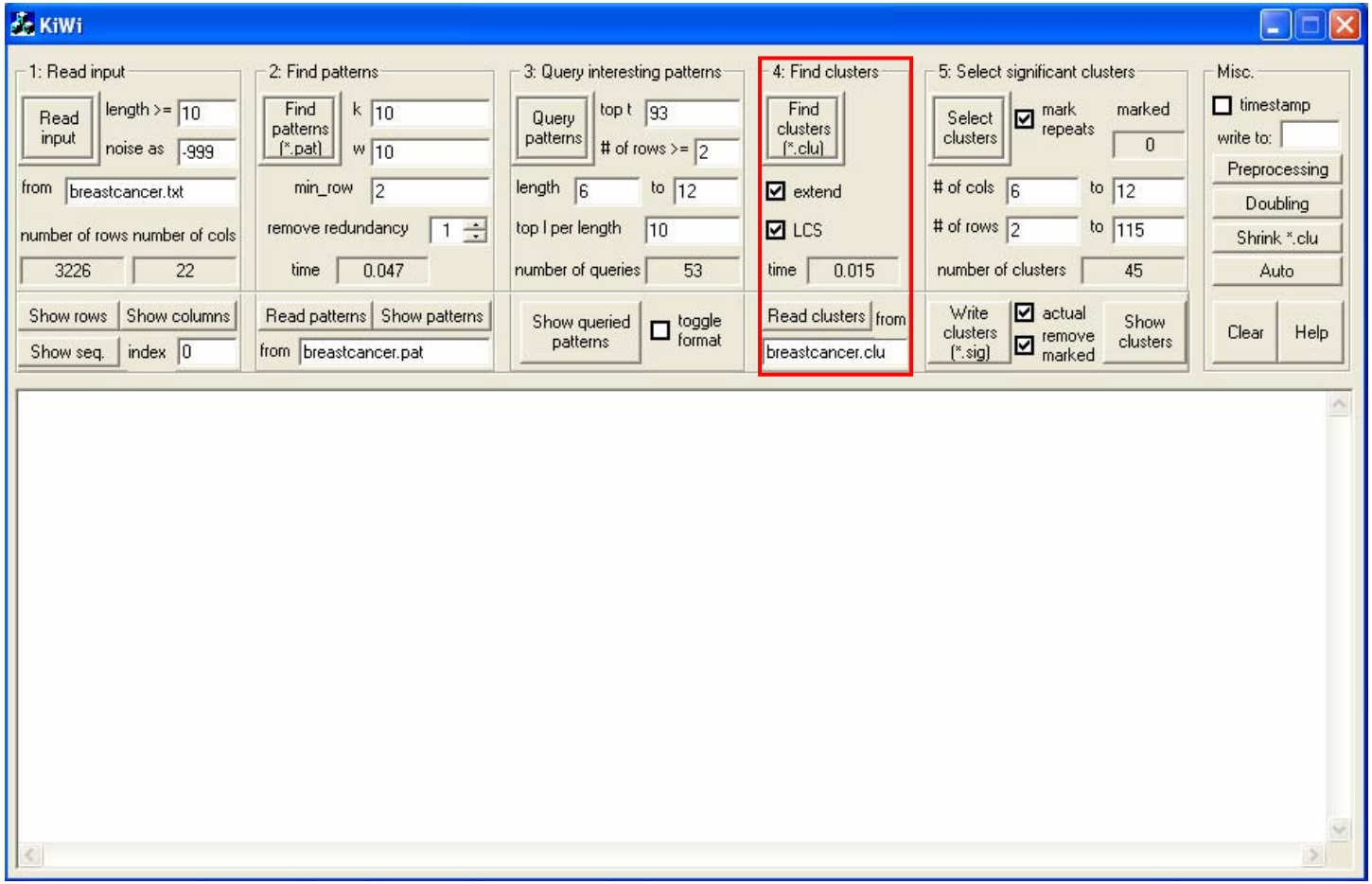
b) The minimum pattern length (minimum # columns) should be selected.
   - Due to pattern extension in the next step, this should be underestimated slightly. For example, if gene clusters with coexpression across at least 10 experiments are desired you would choose a minimum length of 8 or 9.
   - Normally long patterns are desirable. Therefore, leave the maximum length at its default value (determined by the longest pattern found)

c) Once parameters are chosen, click the ‘Query patterns’ to see how many patterns pass criteria.

d) Use ‘Show queried patterns’ to visualize patterns in display window.

e) For an explanation of ‘top t’ and ‘top I per length’ see help documentation. It is safe to leave these as defaults.

4) Finding clusters

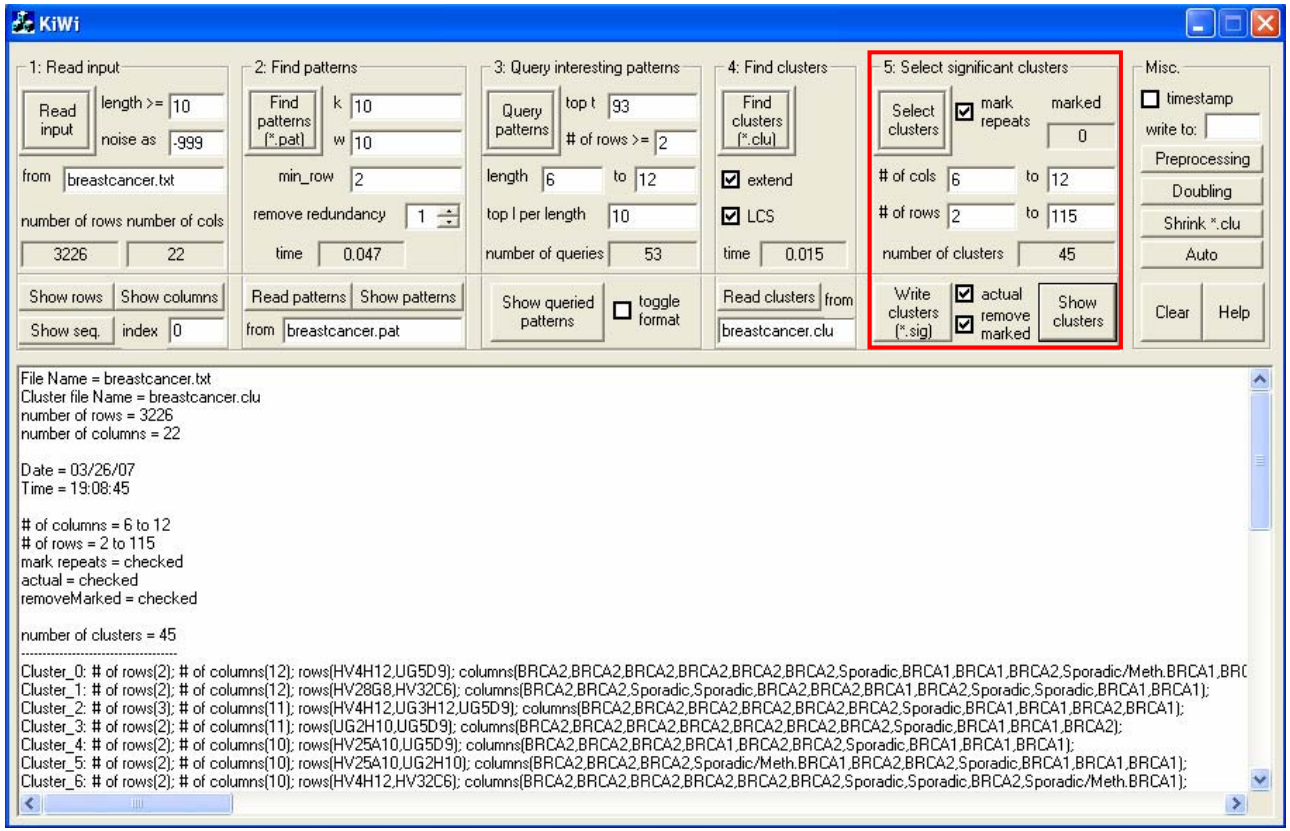


a) The ‘extend’ and ‘LCS’ options are responsible for extending patterns where possible. Normally these should be left checked (as default).

b) To initiate the cluster finding function, click the ‘Find clusters’ button.
   - Once complete, the runtime for finding clusters is displayed.

c) To load a cluster file from a previous KiWi run use the ‘Read clusters’ button and ‘from’ field.

5) Selecting significant clusters



a) Set the minimum/maximum numbers of rows and columns can be set for output.

b) Write the clusters to file with the ‘Write clusters’ button. This will create a .sig file in the working directory.

c) If the number of clusters is not too large (<1000) they can be visualized in the display window using the ‘Show clusters’ button.

d) Once clusters have been written to the .sig file, the clustering is complete and KiWi can be closed.

- Normally ‘mark repeats’ should be left checked. This removes redundant clusters.

Understanding KiWi output file

- Final KiWi clusters are output to a ".sig" file.
- The .sig file contains details about the data file analysed such as file name, number of row, number of columns, data/time
- Cluster results are also summarized with the range in rows and columns for all subspace clusters and the total number of clusters found.
- After the general summary information (indicated by a series of dashes), each cluster is reported on its own line.
- Clusters are reported simply as the list of rows and columns that make up that subspace cluster.
- The cluster format is as follows:
  cluster_id: # of rows; # of columns; rows(row1, row2, etc); columns(column1, column2, etc);
- This file can be parsed with a simple scripting program (e.g. perl) for any downstream analysis of clusters.

Example of output:

File Name = breastcancer.txt
Cluster file Name = breastcancer.clu
number of rows = 3226
number of columns = 22

Date = 03/26/07
Time = 19:11:52

# of columns = 6 to 12
# of rows = 2 to 115
mark repeats = checked
actual = checked
removeMarked = checked

number of clusters = 45

Cluster_0: # of rows(2); # of columns(6); rows(HV4H12, UG5D9); columns(BRCA2, Sporadic, BRCA1, BRCA1, BRCA2, BRCA1);
Cluster_1: # of rows(2); # of columns(6); rows(HV28G8, HV32C6); columns(BRCA2, Sporadic, BRCA1, BRCA2, Sporadic, BRCA2);
etc…

Summary

- KiWi 1.0 is a simple interface for highly scalable subspace cluster (OPSM) discovery from large data matrices (such as gene expression datasets).
- This tutorial covers the basic running operations and parameter selection from data input to cluster output.
- For additional parameters and options (such as anti-correlation discovery) please see the help documentation available through the interface.
- Details about the algorithm and extensive biological assessment can be found in the KiWi papers (see cover page for references).
- If you have suggestions or questions please contact Byron Gao (bgao@cs.sfu.ca) or Obi Griffith (obig@bcgsc.ca).