Disclosed: An efficient depth-first, top-down algorithm for mining disjunctive closed itemsets in high-dimensional data Supplementary Material – Example of Disclosed in practice

Renato Vimieiro and Pablo Moscato
Centre for Bioinformatics, Biomarker Discovery and Information-Based Medicine
The University of Newcastle
Hunter Medical Research Institute
Lot 1, Kookaburra Circuit, New Lambton Heights, NSW, 2305, Australia.
Email: {renato.vimieiro, pablo.moscato}@newcastle.edu.au

We show in this supplementary material a step-by-step execution of Disclosed with the example data set presented in Table 1a — reproduction of the same table presented in Example 1 in the main text. Also to increase readability we replicate the pseudo-code of the algorithm below. In spite of this, we encourage the reader to first read Section Disclosed of the main text, as we thoroughly present the algorithm there, before she proceeds with this text.

Algorithm 1 The algorithm Disclosed

Input: $(S, F, R), minsupp$
Output: $FDCI$ the set of closed itemsets
{The whole set of features is a trivial disjunctive closed itemset.}
1: $FDCI \leftarrow \{F\}$
2: $Traverse(S, \infty, TT|_S)$

Algorithm 2 Traverse$(X, s, TT|_X)$ — Disclosed's search space traverse algorithm

1: **if** $prune1(X, s, TT|_X)$ **then**
2: **return**
3: **end**
4: **if** $X = \bigcup\{\alpha(i) \mid (i, \alpha(i)) \in TT|_X\}$ **then**
5: $insert(FDCI, \beta(X))$
6: **end**
7: **if** $reducible(X, s, TT|_X)$ **then**
8: $Traverse(reduce(X), s, TT|_X)$
9: **return**
10: **end**
11: **if** $prune2(X, TT|_X)$ **then**
12: **return**
13: **end**
14: **for** $s' \in X$ *such that* $s' \prec s$ **do**
15: $X' \leftarrow X - \{s'\}$
16: $TT|_{X'} \leftarrow \{(f, \alpha(f)) \in TT|_X \mid \alpha(f) \subseteq X'\}$
17: **if** $TT_{X'} \neq \emptyset \ \wedge \ |X'| \geq minsupp$ **then**
18: $Traverse(X', s', TT|_{X'})$
19: **end**
20: **end for**

Table 1a is a discretized subset of the yeast gene expression data set from Eisen et al. (1998). The data set was discretized using quartiles as thresholds; gene expressions under/over the

Table 1: An example of a data set. Samples and features are subsets of, respectively, the set of experiments (cell cycles) and genes from the yeast gene expressions data set Eisen et al. (1998)

(a) Original data set

ID	Name	List of features
1	spo5	CDC20↑ CLN1↓ CDC53↑ SIC1↓ SWI6↓
2	elu150	CLN1↑ CDC53↓ SIC1↓
3	cdc15-130	CDC20↑ MCM1↑ CLN1↑ SIC1↑
4	cdc15-190	MCM1↑
5	alpha70	MCM1↑ CLN1↑ SIC1↑ SWI6↓
6	diau-c	SWI6↑

(b) Transposed data set

ID	Name	List of samples
a	SWI6↓	spo5 alpha70
b	CLN1↓	spo5
c	CDC53↑	spo5
d	SIC1↑	cdc15-130
e	SIC1↓	spo5 elu150
f	CDC53↓	elu150
g	SWI6↑	diau-c
h	MCM1↑	cdc15-130 cdc15-190 alpha70
i	CDC20↑	spo5 cdc15-130
j	CLN1↑	cdc15-130 alpha70

first/third quartile were classified as down/up-regulated. In this data set, samples corresponds to the following cell cycles in the original data {spo5, elu150, cdc15-130, cdc15-190, alpha70, diau-c} and features corresponds to discretized gene expressions {SWI6↓, CLN1↓, CDC53↑, SIC1↑, SIC1↓, CDC53↓, SWI6↑, MCM1↑, CDC20↑, CLN1↑} — arrows indicate whether a gene is up-regulated or down-regulated. Instead of using the full name of samples and features, we use numbers from 1 to 6 to represent samples, and letters from a to j to represent features as shown in Tables 1a and 1b. We also drop braces and commas in set notation for conciseness. Then, 12 represents the set {spo5, elu150}, while cj represents the set {CDC53↑, CLN1↑}. We adopt this set notation throughout this work.

Example 1. *The algorithm begins inserting the set of all features in the result set, $FDCI$. So, the first set inserted is $abcdefghij$. It then calls the procedure* Traverse *with $S = 123456$, $s = \infty$ and the current conditional table, $TT|_{123456}$.*

The first step on Traverse *is to check if the current candidate can be pruned by Proposition 6 (Prune 1 – see main text). Remembering, a candidate is "prunable" according to Prune 1 if it does contain any sample greater than the last excluded element, such that the sample does not belong to any feature in the conditional table. Since this is still the first iteration, no sample has been excluded yet, thus the candidate* 123456 *is not prunable. Then, it 1) checks whether the candidate is closed, inserting it into $FDCI$ if so; 2) checks whether it is reducible or not; and, finally, 3) checks whether it is prunable by* Prune 2. *We skipped these steps in first iteration since they are trivial by the same reasons as the first one.*

The next step is to evaluate each child of X. For that, the algorithm excludes the elements that are less than the last excluded in X in order. As we said, we assume the traditional greater than *order over the samples of S. Then, the next candidate to be evaluated is $X' = 12345$.*

Traverse *makes a recursive call to itself to evaluate the candidate $X' = 12345$. The candidate is not prunable by any technique we presented, nor it is reducible; it is closed although. Then, the set of features from the conditional table is inserted into $FDCI$, which now contains the whole set of features and $abcdefhij$.*

Next, it evaluates the candidate's children. Since the last element removed was 6*, the next element to be removed is* 5*, and the next candidate is $X' = 1234$. The conditional table is computed (it is displayed on Table 2), and the new candidate is evaluated since it fulfills all constraints required on Algorithm 17.*

As we notice on Table 2, the candidate $X = 1234$ is reducible because its conditional table does not contain sample 4*. Then, the candidate set is reduced to* 123 *and the procedure flows normally.*

We cease the description here since we covered the major aspects of the algorithm's execu-

Table 2: The steps of the execution of Disclosed on the data set of Table 1a. Here *Parent* indicates the previous node in the enumeration tree — the node of the candidate set $X \cup \{s\}$. The sixth column is the union of the sets of samples of the features in the conditional table. Grey rows indicate candidates caught by a search space reduction technique: Step 3 is a reducible candidate; and Step 9 was pruned by *Prune 1*.

Step	*Parent*	X	s	$TT\|_X$	$\bigcup \alpha(f)$	*Closed*
1	—	123456	∞	$abcdefghij$	123456	*yes*
2	1	12345	6	$abcdefhij$	12345	*yes*
3	2	1234	5	$bcefi$	123	*no*
4	3	123	5	$bcefi$	123	*yes*
5	2	1235	4	$abcdefij$	1235	*yes*
6	5	125	3	$abcef$	125	*yes*
7	5	135	2	$abcdi$	135	*yes*
8	5	235	1	dfj	235	*yes*
9	2	1245	3	$abcef$	125	*no*
10	2	1345	2	$abcdhi$	1345	*yes*
11	10	345	1	dh	345	*yes*
12	2	2345	1	$dfhj$	2345	*yes*

tion. In spite of that, Table 2 contains the results of each step in the enumeration rooted with the candidate 12345*. We did not include all steps here because of the size, and also because we believe it would be tedious for the reader. Nonetheless, we also believe that the example was complete enough on illustrating the way that the algorithm operates, and helping the reader better understanding it.*

□

References

Eisen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences 95*(25), 14863–14868.