Storage Consumption Reduction using Improved Inverted Indexing for Similarity Search on LINGO Profiles

Muhammad Jaziem bin Mohamed Javeed[1], Nurul Hashimah Ahamed Hassain Malim[2]
School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia

***Abstract*—Millions of compounds which exist in huge datasets are represented using Simplified Molecular-Input Line- Entry System (SMILES) representation. Fragmenting SMILES strings into overlapping substrings of a defined size called LINGO Profiles avoids the otherwise time-consuming conversion process. One drawback of this process is the generation of numerous identical LINGO Profiles. Introduced by Kristensen et al, the inverted indexing approach represents a modification intended to deal with the large number of molecules residing in the database. Implementing this technique effectively reduced the storage space requirement of the dataset by half, while also achieving significant speedup and a favourable accuracy value when performing similarity searching. This report presents an in-depth analysis of results, with conclusions about the effectiveness of the working prototype for this study.**

***Keywords*—Simplified Molecular-Input Line-Entry System (SMILES); LINGO profiles; similarity searching; inverted indexing**

I. Introduction

Rapid advances in technology over the past few years have allowed for many virtual screening experiments to be conducted extensively [1]. In ligand-based screening, large chemical databases consisting of small molecules are effectively screened by a query molecule as to identify molecules with similar biological activity, applying the well-known similarity principle that “structurally similar molecules are likely to have similar properties” [2,3,4,5,6]. The query structure itself normally exhibits a potentially useful level of biological activity and might be, for example, a competitor’s compound or a structurally novel hit from an initial high-throughput screening (HTS) experiment [7]. Both the query and database molecules are characterized by descriptors.

Simplified Molecular-Input Line-Entry System (SMILES) is a type of 1D representation [8] which represents molecular structures in strings format [9,10]. The SMILES specialized algorithm known as LINGO [11] is introduced in the field as it delivers a required level of simplicity for retrieving the molecules from database. LINGO representation avoids the necessity for producing an explicit model of the chemical structure in the form of either a graph or a 3D structure because it generates the representation of a molecule directly from canonical SMILES [12].

The continuing rise in the number of compounds to be processed is one of the common challenges which have to be confronted in this field, in terms of the accompanying demand for higher processing power and storage costs [13]. Small libraries can take up to 10^5 compounds, while commercially available datasets have approximately (2 x 10^7 compounds) in their libraries [14]. Many research studies have been conducted to address this problem by developing a coherent technique to store the compounds, but this has been limited only to compounds represented in 2D fingerprints [15]. This situation has, consequently, led to the necessity of introducing a data structure efficient enough to store the compounds represented in LINGO Profiles. An inverted index is a type of index data structure which is commonly used to encode data in string format [16 ,17, 18]. It allows for term-based searches to be more effective [19,20]. This study seeks to ascertain whether the introduction of inverted indices actually achieves any reduction in storage and processing costs when performing similarity searching. Therefore, the rest of the paper is organised as follows: Section II presents several related studies pertaining to similarity searching methods. Next, Section III elaborates the research methodology in terms of implementation and experimental design, while Section IV discusses the analyses outcomes. Lastly, this paper ends with a conclusion depicted in Section V.

II. Background Review

The search for compounds similar to a given target ligand structure and compounds with defined biophysical profiles are two main important principles in modern drug discovery process [21]. Both tasks make use of molecular descriptors with different complexity (atomic, topographic, sub structural fingerprints, 3D, biophysical properties, etc.) leading to different representations of the same molecule [22]. In general, structural representation, also known as molecular descriptor is used in describing the characteristics of compounds [23].

Ozturk and co-workers [24] used a state-of-the-art algorithm; the Weighted Nearest Neighbor-Gaussian Interaction Profile (WNN-GIP) with which to evaluate the performance between 1D SMILES representation and 2D representation-based descriptors in the protein-drug interaction task. Their investigation successfully demonstrated that SMILES-based methods [25] of molecular similarity comparison perform as well as 2D-based methods. Moreover, SMILES-based kernels were found to be computationally faster and more flexible than their 2D competitors.

In a different experiment, comparisons were examined between 2D fingerprints such as Daylight, MOLPRINT 2D, MACCS, and Open Babel with 3D shape-based methods, typically SHAEP, PARAFIT and ROCS, in order to measure the efficiency of the similarity searching method across a range of virtual screening methods [26]. Results showed in the past [26][27] that 3D shape-based methods could not perform as well as a simple fingerprint similarity search, in spite of giving conformational information (i.e. shape information) and atomic coordinates of a compound.

Most previous drug-target interaction prediction tasks involving LINGO have utilized the Tanimoto coefficient. Vidal and colleagues [28] used a bioisostere dataset to compute intermolecular similarity between bioisosteric molecules and some randomly sampled pairs of molecules using an integral Tanimoto coefficient. The average similarities (LINGOsim) obtained effectively demonstrated that important information about a molecule is stored in LINGO Profiles. On the other hand, LINGO-DOSM, introduced by Hentabli et al [29], outperformed other descriptors such as EPFP4, GRFP, MACCS etc. LINGO-DOSM is the integral set derived from a given DOSM string. DOSM allows rigorous structure specification by implementing a small and natural grammar. The positive performance of LINGO-DOSM is not only limited to the top 5% for MDDR but it also gives best results for the top-1% for MDDR. This is mainly due to limiting the selection of LINGO length to just four characters. Finally, using the Briem and Lessel benchmark, Andrew and colleagues concluded that LINGO generated from isomeric SMILES can offer better retrieval rates, compared to non-isomeric SMILES. In addition, when LINGO was compared with more complex approaches (Daylight fingerprint) [25], it managed to identify active compounds better for two activity classes (ACE and TXA2).

The effectiveness of LINGO in predicting the property/activity of one molecule compared with another molecule similar to it, however, has a limitation [30]. This technique is associated with the length of the substrings obtained from the fragmentation of a canonical SMILES string, requiring the manipulation of the string and meaning that the processing cost will increase linearly along with the SMILES length [28]. Since search efficiency is progressively more vital with the ongoing expansion of these databases, scalability problems naturally arise when virtual compounds or recently synthesized compounds are added accordingly [31]. A variety of data structures and algorithms were consequently introduced throughout the years to accelerate this process by reducing the search, i.e. by rapidly eliminating the molecules that are not homogenous to the query, without computing their similarity to the query [32].

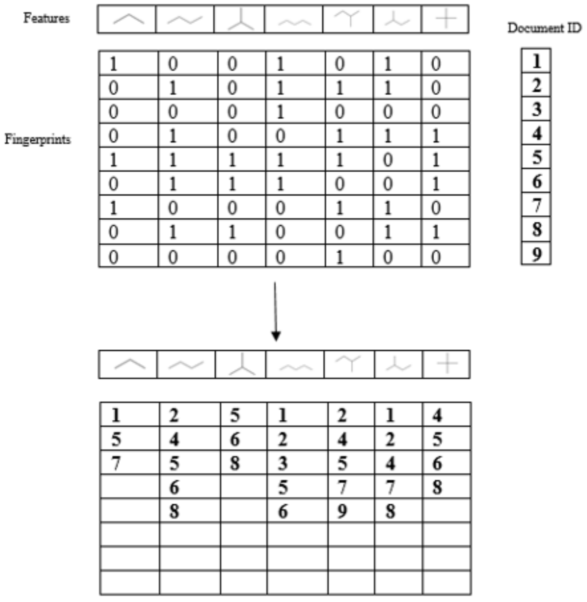
Imran and co-workers [33] presented a new algorithm known as the SIML ("Single-Instruction, Multiple-LINGO") to measure the similarity between molecules. Each multiset in a molecule is represented in 32-bit integers and it is stored in a sorted vector of 4-Lingos (represented as integers). A new algorithm, (1), was derived based on the vector representation of the multisets. This sparse vector algorithm speeds up the computation involved, as for every Tanimoto calculation only the intersection size ⟨A, B⟩ needs to be calculated.

$$T_{AB} = \frac{\langle A,B\rangle}{\langle A,A\rangle + \langle B,B\rangle - \langle A,B\rangle} \qquad (1)$$

Outside the field of cheminformatics, numerous information retrieval communities in general have been conducting experiments for decades on searching text in large datasets [34]. State-of-the-art algorithms from general information retrieval, known as inverted indices, are considered applicable for use in cheminformatics, as both domains arrived at the same similarity measure and representation [35] independently from one another. Features are associated with each respective list of documents contained in a given database, as shown in Fig. 1.

The features-documents association guarantees the reduction of the similarity computations between database molecules and the query as it removes database molecules which are irrelevant to the desired list. This approach can also be applied directly to SMILES string representations for molecules.

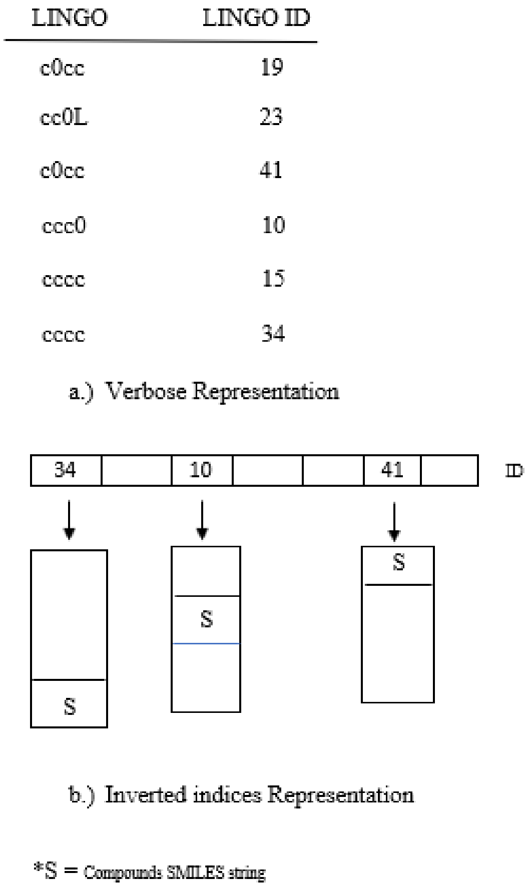
Kristensen et al. [36] proposed performing a similarity search between a target and database compounds represented using LINGO multisets by representing the database as inverted indices. The idea was to keep the LINGO multisets as a vector, where every cell in the vector is assigned to hold one of the LINGO identifiers (ID) from the verbose representation. Unlike SIML which uses two arrays to represent a LINGO multiset, verbose representation utilizes only an array to store the whole multisets including duplicate LINGO represented using multiple different IDs as shown in panel (a) of Fig. 2.



1	0	0	1	0	1	0	1
0	1	0	1	1	1	0	2
0	0	0	1	0	0	0	3
0	1	0	0	1	1	1	4
1	1	1	1	1	0	1	5
0	1	1	1	0	0	1	6
1	0	0	0	1	1	0	7
0	1	1	0	0	1	1	8
0	0	0	0	1	0	0	9

1	2	5	1	2	1	4
5	4	6	2	4	2	5
7	5	8	3	5	4	6
	6		5	7	7	8
	8		6	9	8	

Fig. 1. Molecules Represented in Fingerprint Format are Stored in Inverted Index Data Structure.



LINGO	LINGO ID
c0cc	19
cc0L	23
c0cc	41
ccc0	10
cccc	15
cccc	34

a.) Verbose Representation

b.) Inverted indices Representation

*S = Compounds SMILES string

Fig. 2. (a) Each LINGO is Associated with their Respective IDs. (b) LINGO and their Reference to their Original SMILES String in Inverted Indices Representation.

Input from panel (a) is used to create inverted indices (panel (b)) listing all the multisets associated with a given ID. Similarities are computed based on the value stored in the counting vector after the inverted indices are traversed. This strategy, however, led to a drawback as multiple occurrences of the similar LINGO in a compound will consume more storage space. It is certainly not feasible for huge datasets (e.g. ChEMBL). In addition, the construction of the inverted indices necessitates a search of the largest ID in the dataset. These situations will cause the increase in the processing time and consume high amount of resources, when performing similarity searching process. Besides, Kristensen work is only practical for chemical dataset such as Maybridge and ZINC.

Instead of finding a new method for indexing a database, a small modification of the inverted indexing scheme introduced by Kristensen et al. [36] is proposed in this study. Verbose representation is eliminated by the introduction of a pattern matching approach to resolve a query. This modification is made to increase the available storage space and to minimize the time taken to search a LINGO. A brief explanation of how the indexing method for this study was implemented is discussed in the following section.

III. Methodology

The work was conducted purely on the 102,540 MDDR dataset compounds, where searches were focused only on selected structures from eleven activity classes. The first experiment of this study aimed at measuring the recall values obtained by LINGO Profiles on MDDR dataset, comparing it with various other fingerprints. The second experiment of this study intended to perform similarity searching based on the proposed indexing method, which as discussed earlier in the literature. The time taken and the storage consumption for both experiments were to be computed along before presenting a full discussion of these results in the next section.

A. Performing Similarity Searching in Sequential Manner

A q-LINGO is a q-character string which may include letters, numbers, and symbols such as "(",")", "[", "]", "#", etc. and which is obtained by stepwise fragmentation of a canonical SMILES molecular representation [28]. Before the LINGOs are created from a compound, the compound ring numbers must be substituted for "0". If atoms such as "Cl" and "Br" are present, they will be replaced by "L" and "R", respectively. Raw MDDR Dataset (file A) stores all possible LINGOs for similarity searching after it is being fragmented and modified from the original MDDR dataset. It is attached together with its respective ID in sequential manner. Fig. 3 shows the whole process in generating LINGO Profiles.

Using the raw MDDR dataset (file *A*), to obtain LINGO for our query string (compound *A*) and a MDDR database compound (compound *B*), the ID of the compounds was compared with file *A*. Next, the LINGO*sim* function was used to calculate the similarities between the two compounds. Based on a comparison of the LINGOs of the two compounds, *A* (query compound) and *B* (MDDR database compound) any intermolecular similarities were computed using the integral Tanimoto coefficient. N_{Ai} represents the number of LINGOs of type (*i*) in compound *A*, while N_{Bi} represents the number of LINGOs of type (*i*) in compound *B*, and (*l*) is the number of LINGOs contained in either compound *A* or *B*.

B. Performing Similairty Searching using Proposed Indexing Scheme

Two columns existed in this inverted indexing scheme ("Word" and "Documents") allow the query to perform similarity searching via random access [37]. "Word" column in Table I can be referred to as the unique LINGO Profiles obtained from the MDDR dataset and the "Document" column signify the compound IDs which contains the respective LINGO [37].

From the list (file *A*) generated earlier, it is possible to map the LINGO and IDs into the indexing scheme. 409,752,8 LINGO Profiles contained in file *A* are compared with each other and if two or more identical LINGO Profiles is found, then their respective ID are appended together with the LINGO Profile in the list. In the end, the indexed database would only have 4054 unique LINGO Profiles. Fig. 4 summarizes the whole process.

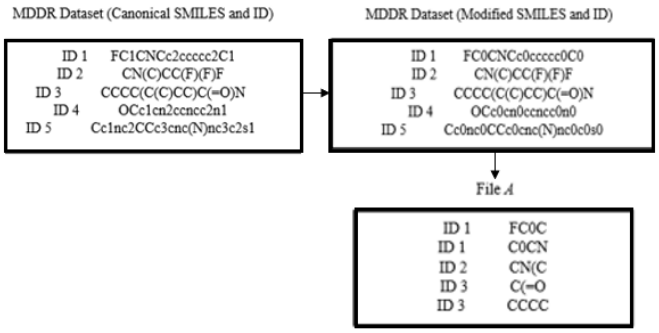


Fig. 3. Modifying and Fragmenting LINGO Profiles from MDDR Dataset.

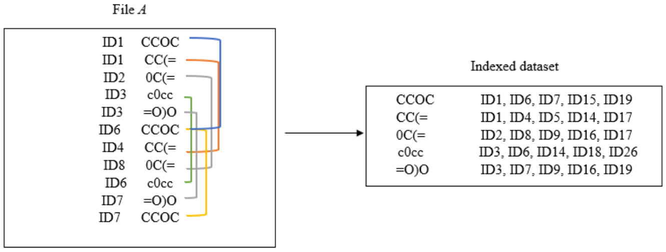

File *A*
ID1 CCOC
ID1 CC(=
ID2 0C(=
ID3 c0cc
ID3 =O)O
ID6 CCOC
ID4 CC(=
ID8 0C(=
ID6 c0cc
ID7 =O)O
ID7 CCOC
Indexed dataset
CCOC ID1, ID6, ID7, ID15, ID19
CC(= ID1, ID4, ID5, ID14, ID17
0C(= ID2, ID8, ID9, ID16, ID17
c0cc ID3, ID6, ID14, ID18, ID26
=O)O ID3, ID7, ID9, ID16, ID19

Fig. 4. Comparing LINGO Profiles and Eliminating Redundant LINGO Profiles on file a; as to Generate Indexed Dataset.

Calculating similarity values using our proposed method differed from the conventional method because it was based on a pattern-matching technique. Whenever the LINGOs in the query compound were found in the indexed database, the IDs in the "Document" column were retrieved and the frequency of occurrence is accumulated and calculated accordingly. The ranked list obtained were then sorted in a descending order to calculate the recall values. The whole process is illustrated in Fig. 5.

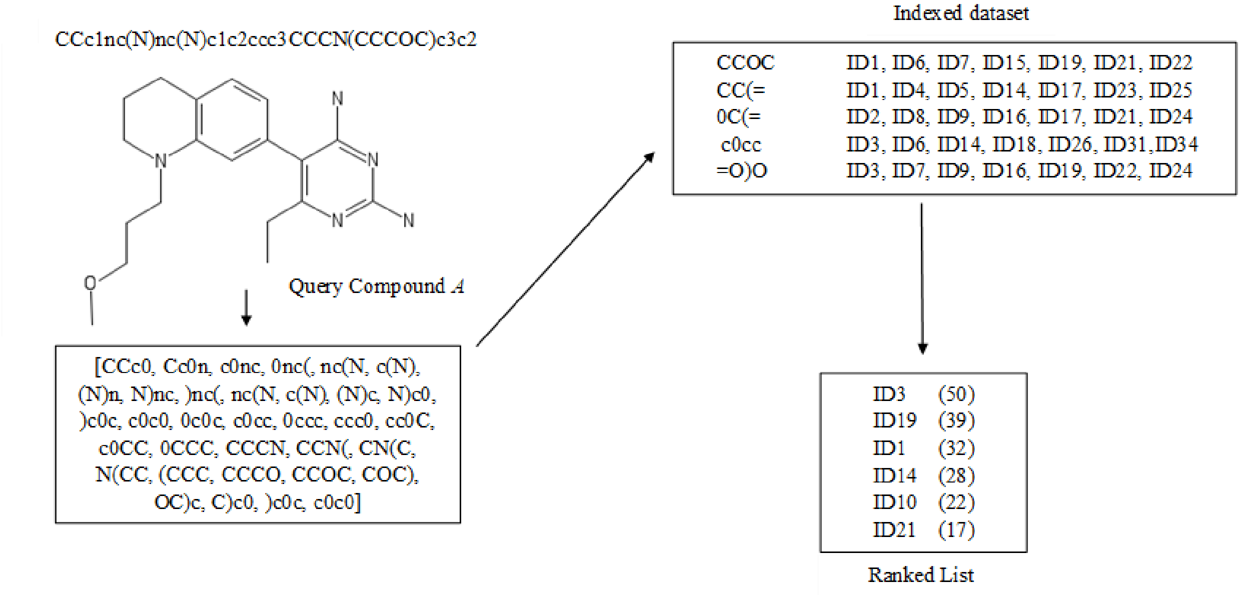
Table II shows the activity classes which were used in both experiments. Activity classes that were used in the experiments are slightly different in nature. The diversity was determined using the main pairwise Tanimoto similarity (MPS) and it is included in Table I. Structurally homogenous classes such as Renin and ATI has high MPS value as compared to COX and PKC which have low MPS value since they are structurally diverse.

TABLE I. STRUCTURE OF AN INVERTED INDEXING SCHEME

Word	**Documents**
Cow	Document 1, Document 4, Document 6, Document 9, Document 15
The	Document 2, Document 5, Document 8
Hello	Document 12
Cat	Document 7

TABLE II. ACCURACY COMPARISON BETWEEN LINGO AND OTHER DESCRIPTORS (TOP :ACTIVES RETRIEVED; BOTTOM : RECALL)

Activity Classes	**Number of Active Structures**	**Pairwise Similarity (Mean)**	
Renin inhibitors	1130	0.290	Most Homogenous
Angiotensin II ATI antagonists	943	0.229	
HIV Protease inhibitors	750	0.198	
Thrombin inhibitors	803	0.180	
Substance P inhibitors	1246	0.149	
5HT3 antagonists	752	0.140	
D2 antagonists	395	0.138	
5HT1A agonists	827	0.133	
5HT reuptake inhibitors	359	0.122	
Protein Kinase C inhibitors	453	0.120	Most Heterogenous
Cyclooxygenase Inhibitors	636	0.108	



[CCc0, Cc0n, c0nc, 0nc(, nc(N, c(N), (N)n, N)nc,)nc(, nc(N, c(N), (N)c, N)c0,)c0c, c0c0, 0c0c, c0cc, 0ccc, ccc0, cc0C, c0CC, 0CCC, CCCN, CCN(, CN(C, N(CC, (CCC, CCCO, CCOC, COC), OC)c, C)c0,)c0c, c0c0]

CCOC ID1, ID6, ID7, ID15, ID19, ID21, ID22
CC(= ID1, ID4, ID5, ID14, ID17, ID23, ID25
0C(= ID2, ID8, ID9, ID16, ID17, ID21, ID24
c0cc ID3, ID6, ID14, ID18, ID26, ID31,ID34
=O)O ID3, ID7, ID9, ID16, ID19, ID22, ID24

ID3 (50)
ID19 (39)
ID1 (32)
ID14 (28)
ID10 (22)
ID21 (17)

Fig. 5. Process Involved when Performing Similarity Searching using the Proposed Methodology.

IV. Results and Discussion

This section is divided into two sub-sections: A and B. Section A basically confirms Vidal's work via replication and compares performance to other fingerprints. Section B discusses the performance of the proposed method regarding time and storage consumption when benchmarked with the conventional method.

A. *Comparing Accuracy between LINGO Profiles and Various different Fingerprints*

The performance of the similarity searching process can be evaluated based on its effectiveness. Effectiveness includes the calculation of the recall value in every single search. The recall value, R is calculated by dividing the number of actives retrieved at the end of the process, n, by the number of compounds that available in the activity class, N, as shown in (2). In other words, recall can be defined as the percentage of the active molecules, which is gained from the cut-off point in the ranked list. Some of the cut-off points that have been widely used are at 1% and 5%. In this experiment, we only use 1% cut-off. The recall value gained indicated the probability of structures that are showing positive to the target. Thus, the higher the recall value gained, the higher the number of structures that react positively towards the target, which implies the accuracy of the method. Units

$$R = \frac{n}{N} \qquad (2)$$

The performance of similarity searches using LINGOs was compared with the performance of similarity searching using various fingerprints obtained from the work of Malim [23]. A total of 110 searches were performed using 10 queries from 11 activity classes. These searches were executed in accordance to Fig. 6. Table III presents the average results of the number of actives retrieved and recall values.

From Table III, the superior performance of ECFP4 is evident in comparison with other fingerprints and LINGO Profiles, except for two activity classes where LINGOs outperform ECFP4. However, it was observed that the performance of LINGO was comparable with other fingerprints such as MDL, Daylight, and Unity in general. A closer analysis of the difference in the accuracy between both descriptors (ECFP4 and LINGO) reveals that ECFP4 outperformed LINGO only by a small average difference of 2.975%. Renin recorded the highest difference between both methods at 9.13 %, while the lowest difference value was observed in the Thrombin activity class, which favors LINGO Profiles at 0.41%.

TABLE III. Accuracy Comparison between Lingo and other Descriptors (Top: Actives Retrieved; Bottom: Recall)

Activity Classes	**Descriptors**				
	Unity	*LINGO*	*Daylight*	*ECFP4*	*MDL*
5HT1A	56 6.79	64 7.77	59 7.15	**81** **9.79**	53 6.46
5HT3	59 7.83	68 9.10	63 8.30	**89** **11.89**	49 6.55
5HTReuptake	21 5.86	20 5.82	19 5.40	**24** **6.83**	20 5.58
AT1	90 9.49	154 16.36	99 10.54	**236** **25.02**	114 12.10
COX	15 2.41	14 2.34	21 3.22	**28** **4.45**	15 2.48
D2	19 4.74	22 5.70	22 5.63	**27** **6.86**	17 4.33
HIVP	46 6.19	51 6.88	37 4.90	**87** **11.57**	44 5.83
PKC	21 4.57	28 6.23	22 4.88	**35** **7.79**	17 3.75
Renin	167 14.76	316 28.02	133 11.76	**420** **37.15**	126 11.11
SubP	70 5.61	**120** **9.7**	57 4.53	**120** **9.7**	37 2.92
Thrombin	54 6.69	**60** **7.45**	33 4.07	57 7.04	**60** **7.45**

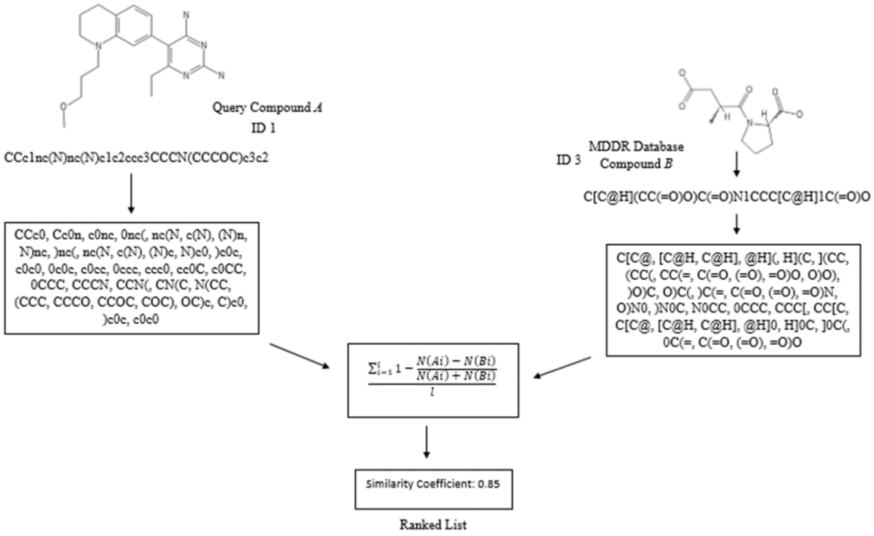

Query Compound *A*
ID 1
CCc1nc(N)nc(N)c1c2ccc3CCCN(CCCOC)c3c2
CCc0, Cc0n, c0nc, 0nc(, nc(N, c(N), (N)n, N)nc,)nc(, nc(N, c(N), (N)c, N)c0,)c0c, c0c0, 0c0c, c0cc, 0ccc, ccc0, cc0C, c0CC, 0CCC, CCCN, CCN(, CN(C, N(CC, (CCC, CCCO, CCOC, COC), OC)c, C)c0,)c0c, c0c0
ID 3
MDDR Database
Compound *B*
C[C@H](CC(=O)O)C(=O)N1CCC[C@H]1C(=O)O
C[C@, [C@H, C@H], @H](, H](C,](CC, (CC(, CC(=, C(=O, (=O), =O)O, O)O),)O)C, O)C(,)C(=, C(=O, (=O), =O)N, O)N0,)N0C, N0CC, 0CCC, CCC[, CC[C, C[C@, [C@H, C@H], @H]0, H]0C,]0C(, 0C(=, C(=O, (=O), =O)O
$\frac{\sum_{i=1}^{l} 1 - \frac{N(Ai) - N(Bi)}{N(Ai) + N(Bi)}}{l}$
Similarity Coefficient: 0.85
Ranked List

Fig. 6. Process Involved when Performing Similarity Searching using Tanimoto Coefficient.

This being the case, according to the work of Hert [38], the nature of the defined activity classes themselves may affect the performance of similarity searches, as more actives may be retrieved in homogenous activity classes, compared to heterogenous ones. Homogenous classes consist of compounds which are less diverse, as opposed to classes with fewer common fragments shared between their compounds, which are described as heterogeneous classes. It can, therefore, be concluded that LINGO works better in homogenous classes as compared to heterogeneous classes. A higher number of active compounds are retrieved in activity classes such as Renin and AT1, in contrast to heterogeneous classes such as COX and 5HT Reuptake. The outcome of this experiment is, then, in agreement with Hert's findings.

Based on the results of this study, it can be summarized that LINGOs may act as an effective alternative to ECFP4 and other fingerprints when performing similarity searching, since this method offers the capability of obtaining a high-accuracy value for a variety of activity classes. It should be noted, however, that the superiority of ECFP4 is widely-known, due to its ability to encode as much structural information as possible when representing the compounds. LINGO profiles, in contrast, only allow for the strings to be observed by shifting one position at a time.

B. *Analyzing the Performance of the Proposed Method in Terms of Time Taken and Storage Consumption*

1) Time complexity: Measuring the time taken for both methods is a very labour-intensive process, as it depends on the compiler and the type of computer or speed of the processor. For this research, the in-built time libraries in JAVA were used to determine the time taken. The timer was started before importing the input file and ended after the search was completed. The elapsed time was measured in milliseconds and for ease of reading it was then converted to hours.

Performing similarity searching using the proposed method is 782 times faster than the same using the conventional method. Achieving such an increase in speed was due to several reasons. Firstly, the indexed database which was created based on a raw MDDR dataset, contained fewer entries than the raw MDDR dataset itself. There was a total of 4053 unique LINGO Profiles in the indexed database as compared to a total of 4097258 LINGO Profiles which were generated in the raw MDDR dataset. With the reduction of the file size, time taken for a query compound to perform similarity searching using an indexed database would be reduced accordingly as now it only has smaller number of entries to browse through, in contrast to similarity searching performed on a raw MDDR dataset which requires a query compound to scan through the whole file to search for a LINGO Profile. The reduction in the file size, will be described in the next section.

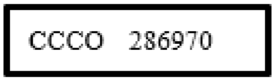
2) Storage complexity: Storage complexity is determined by considering the maximum amount of capacity needed by the secondary storage to store the raw MDDR dataset and the indexed database. The measurement unit used in this study was Megabytes (MB) (1,000,000 bytes in decimal notation). Specifically, there were no tools, libraries or applications used to measure the size of both files, as the sizes of the files were printed automatically by the operating system (OS) after the implementation process. The file size of the indexed database is smaller than the raw MDDR dataset. The reduction by almost half of the file size was achieved through the implementation of the inverted indexing technique, which yields a smaller number of entries in the file. The Raw MDDR dataset contained 4097528 entries, as each entry consisted of a LINGO Profile and its respective index number, as can be seen in Fig. 7.

Each entry here can be referred to as a string and each character within it has a size of a byte (8 bits), as the nature of JAVA language which encodes the strings in UTF-8 format. The '8' in UTF-8 means it uses 8-bit blocks to represent a character. The number of blocks needed to represent a character varies from 1 to 4. Theoretically, one string in a raw MDDR dataset might have a size which falls between 10-11 bytes. Multiplying the size of a string with the number of entries in the raw MDDR dataset and dividing it with the total number of bits in 1 Mb (8000000) will give an approximately similar result. Therefore, having a large number of entries will lead to a larger file size.

As reducing the number of entries is the only way to reduce the size of the file, a compact indexed database comprised of only 4054 entries was constructed for this study. In terms of the number of entries, it can clearly be seen that there is a massive reduction, compared with the raw MDDR dataset. In spite of using the raw MDDR dataset to create the indexed database, all the necessary information was addressed appropriately and the similarity searching process was fully accomplished on this one file.

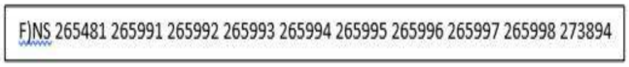
The underlying process involved in the reduction in the number of entries in the indexed database is explained by the removal of duplicate LINGO Profiles and the mapping of the same index number which belongs to a particular LINGO Profile. The structure of an entry in the indexed database is shown in Fig. 8.

It can be seen that one LINGO Profile "sits" together with its respective index number on a single line. In contrast to raw MDDR dataset, each entry may be duplicating a portion of the same information (the index number or LINGO Profile) from the previous or the next entry of the file. This situation can be observed in the Fig. 9.



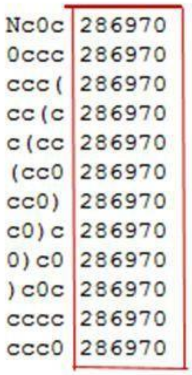
```
CCCO  286970
```

Fig. 7. Mapping of LINGO Profile with its Respective Index Number.



```
F)NS 265481 265991 265992 265993 265994 265995 265996 265997 265998 273894
```

Fig. 8. The Structure of an Entry in the Indexed Database.



Nc0c	286970
0ccc	286970
ccc(	286970
cc(c	286970
c(cc	286970
(cc0	286970
cc0)	286970
c0)c	286970
0)c0	286970
)c0c	286970
cccc	286970
ccc0	286970

Fig. 9. Structure of the Entries in the Raw MDDR Dataset.

V. Conclusions

The inverted indexing scheme has been highlighted in this study as there are several limitations when performing similarity searching using LINGO Profiles. The large raw MDDR dataset which is used in the conventional method to calculate the similarities requires a huge storage capacity, while at the same time increasing the time taken for one query compound to complete the whole process. The proposed method solves this problem by eliminating the redundant LINGO Profiles and multiple occurrences of the same index number. Despite this elimination, the important information associated with the compounds are preserved accordingly. In short, the proposed method makes it possible to process a huge dataset without the help of specialized hardware. In future, this scheme can be used to index a larger chemical database like ChEMBL which consist of more than 1 million compounds data.

References

[1] R. Dolezal, V. Sobeslav, O. Hornig, L. Balik, J. Korabecny and K. Kuca, "HPC Cloud Technologies for Virtual Screening in Drug Discovery", Intelligent Information and Database Systems, pp. 440-449, 2015.

[2] X. Yan, C. Liao, Z. Liu, A. T. Hagler, Q. Gu and J. Xu, "Chemical Structure Similarity Search for Ligand-based Virtual Screening: Methods and Computational Resources", Current Drug Targets, vol. 17, no. 14, pp. 1580-1585, 2016.

[3] A. Cereto-Massagué, M. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé and G. Pujadas, "Molecular fingerprint similarity search in virtual screening", Methods, vol. 71, pp. 58-63, 2015

[4] D. Clark and S. Pickett, "Computational methods for the prediction of ‘drug-likeness’", Drug Discovery Today, vol. 5, no. 2, pp. 49-58, 2000.

[5] P. Willett, "Similarity-based virtual screening using 2D fingerprints", Drug Discovery Today, vol. 11, no. 23-24, pp. 1046-1053, 2006.

[6] D. Stumpfe and J. Bajorath, "Similarity searching", Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 1, no. 2, pp. 260-282, 2011

[7] Y. Hanyf and H. Silkan, "A queries-based structure for similarity searching in static and dynamic metric spaces", Journal of King Saud University - Computer and Information Sciences, 2018.

[8] A. Ciancetta and S. Moro, "Protein–Ligand Docking: Virtual Screening and Applications to Drug Discovery", In Silico Drug Discovery and Design, pp. 189-213, 2015.

[9] "Daylight Theory: SMILES", Daylight.com, 2018. [Online]. Available: http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html. [Accessed: 01- Apr- 2018].

[10] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules", Journal of Chemical Information and Modeling, vol. 28, no. 1, pp. 31-36, 1988.

[11] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann and E. Willighagen, "The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics", Journal of Chemical Information and Computer Sciences, vol. 43, no. 2, pp. 493-500, 2003.

[12] D. Vidal, M. Thormann and M. Pons, "A Novel Search Engine for Virtual Screening of Very Large Databases", Journal of Chemical Information and Modeling, vol. 46, no. 2, pp. 836-843, 2006.

[13] P. Thiel, L. Sach-Peltason, C. Ottmann and O. Kohlbacher, "Blocked Inverted Indices for Exact Clustering of Large Chemical Spaces", Journal of Chemical Information and Modeling, vol. 54, no. 9, pp. 2395-2401, 2014.

[14] S. Dandapani, G. Rosse, N. Southall, J. Salvino and C. Thomas, "Selecting, Acquiring, and Using Small Molecule Libraries for High-Throughput Screening", Current Protocols in Chemical Biology, 2012.

[15] Z. Aung and S. Ng, "An Indexing Scheme for Fast and Accurate Chemical Fingerprint Database Searching", Lecture Notes in Computer Science, pp. 288-305, 2010.

[16] "Apache Lucene - Index File Formats", Lucene.apache.org, 2018. [Online]. Available:https://lucene.apache.org/core/3_0_3/fileformats.html#Inverted%20Indexing. [Accessed: 14- May- 2018].

[17] V. Anh and A. Moffat, "Inverted Index Compression Using Word-Aligned Binary Codes", Information Retrieval, vol. 8, no. 1, pp. 151-166, 2005.

[18] H. Yan, S. Ding and T. Suel, "Inverted index compression and query processing with optimized document ordering", Proceedings of the 18th international conference on World wide web - WWW '09, 2009.

[19] F. Hassen and G. Amel, "An efficient synchronous indexing technique for full-text retrieval in distributed databases", Procedia Computer Science, vol. 112, pp. 811-821, 2017.

[20] J. Zobel and A. Moffat, "Inverted files for text search engines", ACM Computing Surveys, vol. 38, no. 2, p. 6-es, 2006.

[21] P. Willett, J. Barnard and G. Downs, "Chemical Similarity Searching", Journal of Chemical Information and Computer Sciences, vol. 38, no. 6, pp. 983-996, 1998.

[22] D. Agrafiotis, J. Myslik and F. Salemme, "Advances in diversity profiling and combinatorial series design", Annual Reports in Combinatorial Chemistry and Molecular Diversity, pp. 71-92, 1999.

[23] N.H.A.H Malim, “Enhancing Similarity Searching,” Information School, University of Sheffield, Sheffield, 2011

[24] H. Öztürk, E. Ozkirimli and A. Özgür, "A comparative study of SMILES-based compound similarity functions for drug-target interaction prediction", BMC Bioinformatics, vol. 17, no. 1, 2016.

[25] J. Grant, J. Haigh, B. Pickup, A. Nicholls and R. Sayle, "Lingos, Finite State Machines, and Fast Similarity Searching", Journal of Chemical Information and Modeling, vol. 46, no. 5, pp. 1912-1918, 2006.

[26] G. Hu, G. Kuang, W. Xiao, W. Li, G. Liu and Y. Tang, "Performance Evaluation of 2D Fingerprint and 3D Shape Similarity Methods in Virtual Screening", Journal of Chemical Information and Modeling, vol. 52, no. 5, pp. 1103-1113, 2012.

[27] G. Jayashree and V. Perumal, "Enhancing similarity-based query searching performance using self-organized semantic overlay networks", Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14, 2014. Available: 10.1109/icccs.2014.7068168 [Accessed 6 February 2019].

[28] D. Vidal, M. Thorman,. & M.Pons, “LINGO, an Efficient Holographic Text Based Method to Calculate Biophysical Properties and Intermolecular Similarities”, Journal of Chemical Information and Computer Sciences, vol. 45, pp.386-393, 2014

[29] H. Hentabli, N. Salim, A. Abdo and F. Saeed, "LINGO-DOSM: LINGO for Descriptors of Outline Shape of Molecules", Intelligent Information and Database Systems, pp. 315-324, 2013.

[30] M. Skinnider, C. Dejong, B. Franczak, P. McNicholas and N. Magarvey, "Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm", Journal of Cheminformatics, vol. 9, no. 1, 2017.

[31] R. Guha, K. Gilbert, G. Fox, M. Pierce, D. Wild and H. Yuan, "Advances in Cheminformatics Methodologies and Infrastructure to Support the Data Mining of Large, Heterogeneous Chemical Datasets", Current Computer Aided-Drug Design, vol. 6, no. 1, pp. 50-67

[32] P. Sharma, S. Salapaka and C. Beck, "A Scalable Approach to Combinatorial Library Design for Drug Discovery", Journal of Chemical Information and Modeling, vol. 48, no. 1, pp. 27-41, 2008.

[33] I. Haque, V. Pande and W. Walters, "SIML: A Fast SIMD Algorithm for Calculating LINGO Chemical Similarities on GPUs and CPUs", Journal of Chemical Information and Modeling, vol. 50, no. 4, pp. 560-564, 2010.

[34] F. Rinaldi, "Text Mining Technologies for Database Curation", Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, 2014.

[35] R. Nasr, R. Vernica, C. Li and P. Baldi, "Speeding Up Chemical Searches Using the Inverted Index: The Convergence of Chemoinformatics and Text Search Methods", Journal of Chemical Information and Modeling, vol. 52, no. 4, pp. 891-900, 2012.

[36] T. Kristensen, J. Nielsen and C. Pedersen, "Using Inverted Indices for Accelerating LINGO Calculations", Journal of Chemical Information and Modeling, vol. 51, no. 3, pp. 597-600, 2011.

[37] E. S. D. Moura, “Text Indexing Techniques”, Encyclopedia of Database Systems, 4084–4088,2018.

[38] J. Hert, M. Keiser, J.J. Irwin, T.I. Oprea, and B.K. Shoichet, “Quantifying the Relationship among Drug Classes”, Journal of Chemical Information and Modelling, vol 48, pp. 755-765,2008.