Evaluating Parallel Ward Algorithm for Drug Discovery

Mohamed G. Malhat, Hamdy M. Mousa
Computer Science dept., Faculty of Computers and Information,
Menoufia University, Egypt
m.gmalhat@yahoo.com, hamdimmm@hotmail.com

***Abstract*— Millions of compounds are now available in chemical libraries and scientists have to test these compounds against biological targets in order to identify lead compounds. The identification of lead compounds is a key step in the drug discovery process. So, there are many hierarchical clustering algorithms are developed and modified for that purpose. Ward algorithm is one of the most popular hierarchical clustering algorithms that are used in many applications in the drug discovery process because of it is accuracy. But, it has limitation to handle large data sets within a reasonable time and memory resources. In this paper, we evaluate and compare two parallel approaches to run ward algorithm. The two approaches are parallel for loop and MapReduce framework. The results shows that parallel for loop failed to reduce computational time of ward algorithm due to overhead needed for data communications. But, MapReduce framework shows considerable reduction in computational time. The parallel ward algorithm saves 17% of time using three nodes and saves 58% of time using six nodes using MapReduce.**

Keywords—Drug Discovery; Hierarchical Clustering; Ward Clustering; Parallel for; MapReduce.

I. INTRODUCTION

The drug discovery is the process of making or identifying an organic molecule (lead compound) that would bind to a biological target with fewer side effects. It consists of seven steps: disease selection, target hypothesis, leads compound identification, lead optimization, pre-clinical trial, and clinical trial and pharmacogenomics optimization [1]. Chemoinformatics is a new discipline emerging from storing, manipulating, processing, design, creation, organization, management, retrieval, analysis, dissemination, visualization, and use of chemical information. Sometimes it's defined as the application of informatics methods to solve chemical problems [2]. Chemoinformatics are used in lead compound identification and optimization steps. The use of Chemoinformatics becomes a critical part of the drug discovery process as it accelerates the drug discovery process and reduces the overall cost [3].

Clustering is used increasingly in preliminary analyses of large data sets of medium and high dimensionality as a method of selection, diversity analysis and data reduction. Compared to the other costs of drug discovery, clustering can add significant value at minimal cost [4]. Many clustering algorithms are available for Chemoinformatics but the most popular clustering algorithm is ward clustering algorithm. This algorithm used in many applications in drug discovery process such as compound selection, compound acquisition, High-Throughput Screening (HTS), Quantitative Structure-Activity Relationship (QSAR) analysis and Absorption, Distribution, Metabolism, Elimination, Toxicity (ADMET) prediction [5-12].

In this age of data size and processing explosion, parallel processing is essential to process a massive volume of data in a timely manner. Different parallel processing techniques are available, but the most popular techniques are parallel for loops and MapReduce framework. Parallel for loops is a technique used by parallel computing toolbox in matlab. The iterations of for loop are divided across number of workers to speed up processing time. These iterations must be independent of each other's. MapReduce is a scalable and fault-tolerant data processing tool that enables to process a massive volume of data in parallel with many low-end computing nodes [13]. MapReduce framework used to solve the molecular docking for large-scale virtual screening that is needed to retrieve a large number of small molecules that meets a certain requirements from database [14]. The results show that parallel for loop takes more time than traditional sequential for loop. This is due to overhead time needed for data communication between different workers. On the other hand, MapReduce framework shows a good performance in term of time compared with traditional sequential clustering. The result shows that MapReduce saves half of time using six workers. The

organization of this paper is as follow. In section two, ward clustering algorithm is overviewed. In section three, Parallel for loop and MapReduce are overviewed. In section four, the proposed work is presented and the result is discussed. Finally in section five, conclusion and future work are given.

II. WARD CLUSTERING ALGORITHM

Clustering methodology has been developed and used in a variety of areas including archaeology, astronomy, biology, computer science, electronics, engineering, information science, and medicine. The current main uses of clustering algorithms for chemical data sets are to group homogeneous compounds into a cluster based on a model of similarity measures to determine its activity and to identify the chemical compounds (lead compounds) that display the desired behavior against specific bio-molecular target [3]. The overall process of clustering involves the following steps:

1. Generate appropriate descriptors for each compound in the data set.
2. Select an appropriate similarity measure.
3. Use an appropriate clustering method to cluster the data set.
4. Analyze the results.

In step 1, there are different ways to represent chemical compounds but the best way is descriptor-based representation. There are descriptors that reflect the structure of compounds and others that reflect the physiochemical properties of compounds [15]. In step 2, there are different distance measures such as hamming, Euclidean and soergel distance and Tanimoto, dice and cosine coefficients. Euclidean distance is the best for numerical descriptor and Tanimoto is the best for binary fingerprint [16]. In step 3, different clustering techniques can be applied in Chemoinformatics. In step 4, the result is analyzed to determine the accuracy of clustering output.

The clustering process is unsupervised, that is, there is no predefined grouping that the clustering seeks to reproduce. In contrast to supervised learning, where the task is to establish relationships between given inputs and outputs to enable prediction of the output from new inputs. In unsupervised learning only the inputs are available and the task is to reveal aspects of the underlying distribution of the input data. Clusters can be overlapping or non-overlapping. In overlapping clustering, compound can occur in more than one cluster. Each compound is a member of all clusters to a certain degree. In non-overlapping clustering, each compound is a member of exactly one cluster. The majority of clustering methods used on chemical data sets generate non-overlapping clusters [2].

Non-overlapping clustering can be hierarchical or non-hierarchical. In hierarchical clustering, data set is analyzed in an iterative way, such that at each step a pair of clusters is merged or a single cluster is divided. The successive levels can be visualized using a dendrogram, as shown in Fig.1. Each level of the hierarchy represents a partitioning of the data set into a set of clusters. In non-hierarchical clustering, the data set is analyzed to produce a single partition of the compounds resulting in a set of clusters.

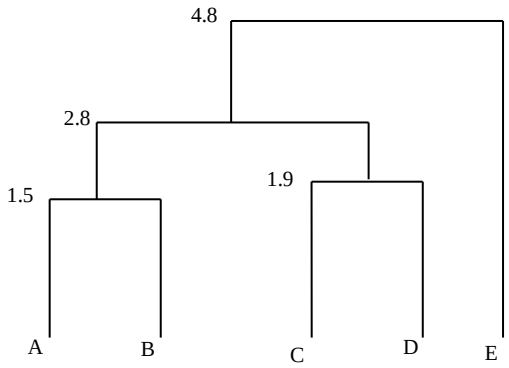

Fig. 1. Example of hierarchical dendrogram

Hierarchical can be agglomerative or divisive. In agglomerative clustering, the hierarchical method starts with all compounds as singletons and the clusters are merged iteratively until all compounds are in a single cluster. With respect to the dendrogram in

Fig.1, it is a bottom-up approach. In divisive clustering, the hierarchical method starts with all compounds as a single cluster and iteratively splits one cluster into two until all compounds are singletons. With respect to the dendrogram in Fig.1, it is a top-down approach.

The most popular agglomerative hierarchal clustering method that has a wide range of applications in drug discovery process is ward clustering algorithm [5-12]. Ward clustering methods implemented using stored-matrix algorithm, this matrix contains all pairwise proximities between compounds in the data set to be clustered. Each cluster initially corresponds to an individual compound. As clustering proceeds, each cluster may contain one or more items. The stored-matrix algorithm proceeds as follows:

1. Calculate the initial proximity matrix containing the pairwise proximities between all pairs of clusters in the data set.
2. Scan the matrix to find the most similar pair of clusters, and merge them into a new cluster.
3. Update the proximity matrix by inactivating one set of entries of the original pair and updating the other set with the proximities between the new cluster and all other clusters.
4. Repeat steps 2 and 3 until just one cluster remains.

The proximity calculation in step 3 typically uses the Lance-Williams matrix-update formula in Eq.1.

$$d[k,(i,j)] = \propto_i d[k,i] + \propto_j d[k,j] + \beta d[i,j] + \gamma |d[k,i] - d[k,j]| \quad (1)$$

$$\propto_i = \frac{N_i + N_k}{N_i + N_j + N_k} \quad (2)$$

$$\propto_j = \frac{N_j + N_k}{N_i + N_j + N_k} \quad (3)$$

$$\beta = \frac{-N_k}{N_i + N_j + N_k} \quad (4)$$

$$\gamma = 0 \quad (5)$$

Where $d[k,(i,j)]$ is the proximity between cluster k and cluster (i, j) formed from merging clusters i and j. Using this sequential implementation, ward algorithm is unable to handle large chemical data sets in reasonable time and memory resources. It requires O (n^2 *log* n) computation time where n is the number of clusters [17]. In recent years, researchers try to modify algorithms to run in parallel manner. In [18] a parallel version of SLINK algorithm is developed. In [19] a parallel version of hierarchical clustering is designed on an n-node hypercube and an n-node butterfly. In [17] hierarchical clustering is performed using several distance matrix based on PRAM. Graphics Processing Unit (GPU) and MapReduce become the most popular approaches for parallel processing [20, 21]. MapReduce will be used to run ward clustering algorithm in parallel manner. The computation of stored matrix will be divided across multiple machines to improve the performance of ward algorithm. In the next section, different parallel techniques are overviewed.

III. Parallel Techniques Overview

There are different techniques used to implement algorithms in parallel. But, the most popular techniques are Parallel Computing Toolbox and MapReduce framework.

Parallel Computing Toolbox is used to solve computationally and data-intensive problems using multicore processors, GPUs, and computer clusters. It uses the full processing power of multicore desktops by executing applications on workers that run locally. Without changing the code, the same applications can run on a computer cluster or a grid computing service. The parallel applications can run interactively or in batch. One of the most popular methods in parallel computing toolbox is parallel for loops. The parallel for loop invokes the traditional for loop but in parallel manner. The number of parallel for depends on the number of available workers. Typically, the only difference between iterations is defined by different input data. In these cases, the ability to

run separate iterations simultaneously can improve performance. The only restriction on parallel loops is that no iterations be allowed to depend on any other iteration.

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system [13].

Fig.2 shows MapReduce architecture consists of two phases: map and reduce. In the map phase, the mapper (the algorithm that specifies a map function) takes as input a key/value pair, and it outputs a sequence of key/value pairs. In the reduce phase, the reducer (the algorithm that specifies a reduce function) takes as input all the key/value pairs that have the same key, and it outputs a sequence of key/value pairs which have the same key as the input pairs; these pairs are either the final output, or they become the input of the next MapReduce round [22]. We consider Hadoop, the most popular open-source implementation of MapReduce. Hadoop uses block-level scheduling and a sort-merge technique to implement the functionality for parallel processing. The Hadoop Distributed File System (HDFS) handles fault tolerance and replication for reading job input data and writing job output data [23].

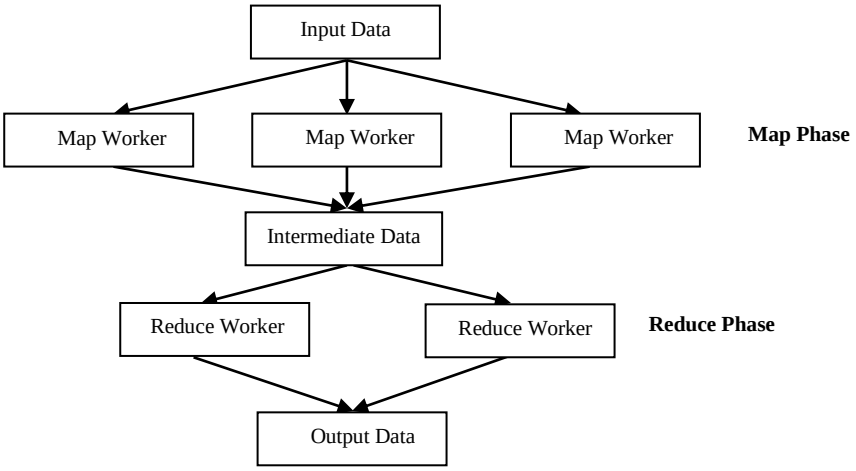

Fig. 2. Architecture of MapReduce

IV. Implementation and Experimental Results

Ward clustering algorithm calculations depends on distance matrix. This matrix is symmetric, with zeros in the main diagonal; only a triangular half of the distance matrix is stored and computed. The computation of distance matrix is divided across P workers. Each worker computes part of distance matrix as show in Fig.3. In parallel for, the workers are multicore processors. In MapReduce, the workers are independent machines. Then distance matrix is scanned to find minimum distance. The distance matrix values are divided across P workers. Each worker gets the minimum distance for its local values. Then another worker is initiated to get minimum distance of these minimum distances.

Parallel for-loop experiments are implemented in Matlab, under Windows-7 operating system, Intel core-i5, 2.5 GHz and Ram 4 GB. MapReduce experiments are done on Amazon Elastic MapReduce (EMR). EMR utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). EMR enable user to determine number and type of map and reduce instances. In our experiments, parallel ward clustering algorithm is run over four instances (three map instances and one reduce instance) and seven instances (six map instances and one reduce instance) with M1.large processing capabilities for each instance. For sequential ward clustering algorithm only one instance with M1.large processing capabilities is used. Table 1 shows the Processor Architecture, Virtual

Central Processing Unit (VCPU), Elastic Compute Unit (ECU), Memory, Instance Storage, Elastic Block Store (EBS) and Network Performance specification of each M1.large instance.

The NCI data set is used [24]. Two random subsets are taken from NCI data set. NCI-1 subset contains 500 compounds and NCI-2 subset contains 1,000 compounds. Over this subset, BCUT descriptor found in Chemical Development Kit is used [25].

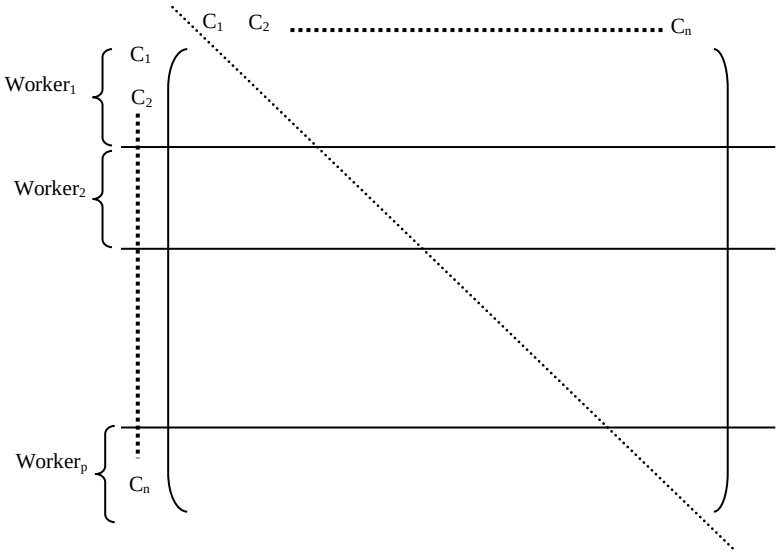

Fig. 3. Parallel distance matrix computation

TABLE.1. SPECIFICATION OF EC2 INSTANCE

M1.Large instance specification	Value
Processor Arch	64-bit
VCPU	2
ECU	4
Memory (GiB)	7.5
Instance Storage (GB)	2 x 240
EBS-optimized available	Yes
Network Performance	High

Table 2 shows the time required in minutes to run sequential ward using one worker and parallel ward using three and seven workers over two subsets of NCI data set using parallel for-loops.

TABLE.2. RESULTS IN MINUTES OF RUNNING WARD ALGORITHM USING PARALLEL FOR-LOOPS

Data Set Name	Sequential	Parallel	
		three workers	six workers
NCI-1	32.34	52.91	58.11
NCI-2	527.91	758.16	830.39

Table 3 shows the time required in minutes to run sequential ward using one node and parallel ward using three maps and one reduce nodes, and six maps and one reduce nodes over two subsets of NCI data set using MapReduce framework.

TABLE 3. RESULTS IN MINUTES OF RUNNING WARD ALGORITHM USING MAPREDUCE FRAMEWORK

Data Set Name	Sequential	Parallel	
		three nodes	six nodes
NCI-1	29.03	24.15	12.11
NCI-2	482.98	402.46	201.28

Fig.4 shows time required for sequential and parallel ward clustering using Parallel For; NCI-1 subset takes in 32.34 minutes in sequential, 52.91 minutes in parallel using three workers, 58.11minutes in parallel using six workers. NCI-2 subset takes in 527.91minutes in sequential, 758.16 minutes in parallel using three workers, 830.39 minutes in parallel using six workers.

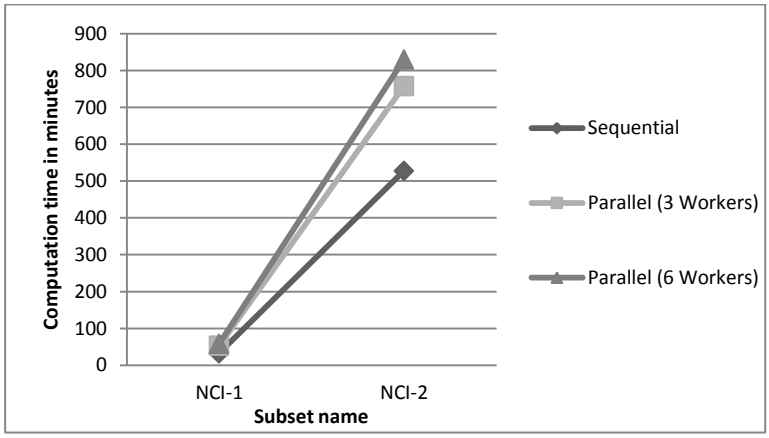

Fig. 4. Time required in minutes for sequential and parallel ward algorithm using Matlab Parallel For

Fig.5 shows time required for sequential and parallel ward clustering using MapReduce; NCI-1 subset takes in 29.03 minutes in sequential, 24.15 minutes in parallel using three nodes, 12.11 minutes in parallel using six nodes. NCI-2 subset takes in average 482.98 minutes in sequential, 402.46 minutes in parallel using three nodes, 201.28 minutes in parallel using six nodes.

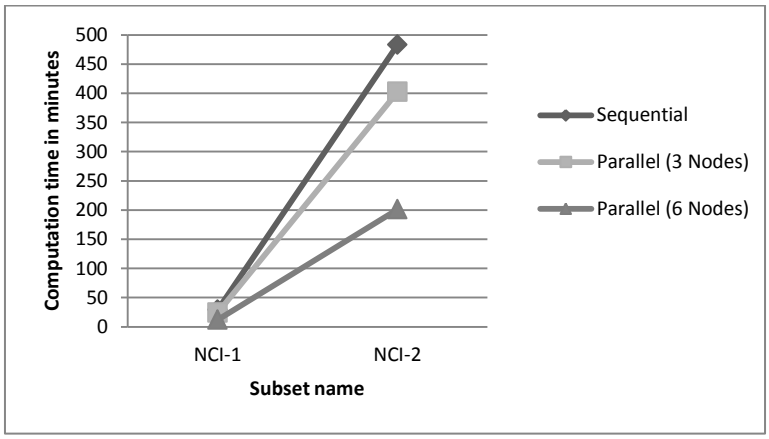

Fig. 5. Time required in minutes for sequential and parallel ward algorithm using MapReduce

The results of parallel for-loop show that sequential ward produce results better than parallel one. So, parallel-for is not a solution to reduce computational time because of data communication time between workers. On the other side, MapReduce saves 17% of time using three nodes and saves 58% of time using six nodes compared to sequential one. So, as the number of workers increase, the time required to cluster data set decrease.

V. Conclusion and Future Work

In the age of parallel processing and explosion of data size, MapReduce framework is preferable for processing large chemical data sets in timely manner fashion. In this paper, two different parallel approaches are used to implement ward in parallel manner. Experimental results show that MapReduce saves 17% of time using three nodes and saves 58% of time using six nodes. But, parallel for-loop failed to reduce computational time due to communication time needed between workers. In future work, other different parallel approaches will be used such as GPU and OpenCL to determine the best parallel approach for ward algorithm.

References

[1] T. Engel, "Basic Overview of Chemoinformatics", J. Chem. Inf. Model, Vol.64, pp.2267-2277, 2006.

[2] A. R. Leach and V. J. Gillet, "An Introduction to Chemoinformatics", Springer, 2003.

[3] C. Aggarwal and H. Wang, "Managing and Mining Graph Data", Springer, 2010.

[4] G. Keseru and G. Makra, "Hit discovery and hit-to-lead approaches", Drug discovery today, Vol.11, pp.741-748, 2006.

[5] R. D. Brown and Y. C. Martin, "Use of Structure–Activity Data to Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection", J. Chem. Inf. Comput. Sci., 36 (3), 572, 1996.

[6] P. R. Menard, R. A. Lewis, and J. S. Mason, "Rational Screening Set Design and Compound Selection: Cascaded Clustering", J. Chem. Inf. Comput. Sci., 38 (3), 379, 1998.

[7] T. N. Doman, J. M. Cibulskis, M. J. Cibulskis, P. D. McCray, and D. P. Spangler, "Algorithm5: A Technique for Fuzzy Clustering of Chemical Inventories", J. Chem. Inf. Comput. Sci., 36 (6), 1195, 1996.

[8] P. Willett, V. Winterman, and D. Bawden, "Implementation of Nonhierarchic Cluster-Analysis Methods in Chemical Information Systems; Selection of Compounds for Biological Testing and Clustering of Substructure Search Output", J. Chem. Inf. Comput. Sci., 26 (3), 109, 1986.

[9] J. L. Jenkins, A. Bender, and J. W. Davies, "In silico target fishing: Predicting biological targets from chemical structure", Drug Discovery Today,3(4): pp.413–421, 2006.

[10] M. Mishra, H. Fei and J. Huan, "Compuatational Prediction of Toxicity", IEEE International Conference on Bioinformatics and Biomedicine, pp.686-691, 2010.

[11] C. Korn and S. Balbach, "Compound selection for development – Is salt formation the ultimate answer? Experiences with an extended concept of the “100 mg approach” ", European Journal of Pharmaceutical Sciences, 2013.

[12] V. J.Gaikwad, "Application of chemoinforatics for innovative drug discovery", International Journal of Chemical and Apllication, Vol.1, Issue 1, pp.16-24, 2010.

[13] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters", Communications of the ACM, Vol.51(1), pp.107–113, 2008.

[14] J. Zhao, R. Zhang, Z. Zhao, D. Chen, and L. Hou, "Hadoop MapReduce Framework to Implement Molecular Docking of large-scale virtual screening", IEEE Asia-Pacific Services Computing Conference, pp.350-353, 2012.

[15] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification", Knowledge and Information Systems, Vol.14, pp.347–375, 2007.

[16] John M. Barnard and Geoffrey M. Downs, "Chemical Similarity Searching", J. Chem. Inf. Comput. Sci., Vol.38, pp.983-996, 1998.

[17] C.F. Olson, "Parallel Algorithms for Hierarchical clustering", Parallel Computing, Vol.21, pp.1313-1325, 1995.

[18] R. Sibson, "SLINK: an optimally efficient algorithm for the single link cluster methods", Computer Journal, Vol.16, pp.30-34, 1973.

[19] X. Li, Z. Fang, "Parallel clustering algorithms", Parallel Computing, Vol.11, pp.275-290, 1989.

[20] Q. Li, R. Salman, E. Test, R. Strack, and V. Kecman, "Parallel multitask cross validation for support vector machine using GPU", J.Parallel Distrib. Comput. Vol.73, pp.293-302, 2013.

[21] S. Upadhyaya, "Parallel approaches to machine learining-A comprehensive survey", J.Parallel Distrib. Comput, Vol.73, pp.284-292, 2013.

[22] H. Karloff, S. Suri, and S. Vassilvitskii, "A model of computation for mapreduce", In SODA, pp.938–948, 2010.

[23] T. White. Hadoop: The Definitive Guide. O’Reilly Media, Inc., 2009.

[24] NCI data set available on http://cactus.nci.nih.gov/download/nci/, last accessed on 6/7/2015.

[25] Chemical Development Kit project available on http://sourceforge.net/projects/cdk/.