gWEGA: GPU-Accelerated WEGA for Molecular Superposition and Shape Comparison

Xin Yan,[a] Jiabo Li,*[b,c] Qiong Gu,[a] and Jun Xu*[a]

Virtual screening of a large chemical library for drug lead identification requires searching/superimposing a large number of three-dimensional (3D) chemical structures. This article reports a graphic processing unit (GPU)-accelerated weighted Gaussian algorithm (gWEGA) that expedites shape or shape-feature similarity score-based virtual screening. With 86 GPU nodes (each node has one GPU card), gWEGA can screen 110 million conformations derived from an entire ZINC drug-like database with diverse antidiabetic agents as query structures within 2 s (i.e., screening more than 55 million conformations per second). The rapid screening speed was accomplished through the massive parallelization on multiple GPU nodes and rapid prescreening of 3D structures (based on their shape descriptors and pharmacophore feature compositions). © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23603

Introduction

The three-dimensional (3D) molecular superposition algorithm is one of the key engines for high-throughput virtual screening and other applications in drug discovery.[1] A number of algorithms were developed in the past 2 decades[2–13]; these were used for virtual screening, scaffold hopping, and shape-feature-based molecular alignment.[8–17] Some algorithms, such as rapid overlay of chemical structures (ROCS), which is based on Gaussian functions,[2,3] are commercially available and well accepted in the drug discovery process. To speed up ROCS computations, the original Gaussian method of Grant et al.[2,3] was simplified and the first-order Gaussian approximation (FOGA) was adopted. A major issue of FOGA is that it introduces considerable noise in shape similarity calculations.[18] To improve the accuracy of molecular shape comparison, while maintaining FOGA's simplicity and efficiency, the weighted Gaussian algorithm (WEGA) was recently introduced by our group.[18] For the 3D virtual screening of very large databases, such as PubChem,[19] ZINC (which has roughly 21 million compounds),[20] and large virtual libraries (e.g., GDB-11/13/17[21–23]), there is a strong demand for even more efficient and accurate molecular shape comparison algorithms.

Graphic processing units (GPUs) have evolved into flexibly programmable, many-core processors with high chip-level parallelism and memory bandwidth. These GPUs have hundreds of computing cores on the same chip, and thus, provide a one or two orders of magnitude increase in computational capabilities (in comparison with conventional CPUs[24]). GPUs have been applied, with great success in computational chemistry,[25–27] bioinformatics,[28] and chemoinformatics.[29–32] For instance, the first GPU implementation of the ROCS algorithm was reported by Haque and Pande.[31] Later, a much more efficient GPU implementation of ROCS, the so-called fast ROCS,[32] was reported by OpenEye; it was claimed that roughly up to 2 million shape comparisons can performed per second on a machine with four Nvidia GPU cards.

To meet the demand of high throughput virtual screening in modern drug discovery for better accuracy in shape-similarity scoring, we have implemented a GPU-accelerated version of WEGA (gWEGA) and have increased the screening speed by roughly two orders of magnitude with one NVIDIA TESLA C2050 GPU card. Recently, Tianhe-2 (or Milkyway-2), a supercomputer with a performance of 33.86 peta-flops per second, has been under rapid construction in Guangzhou. The Tianhe-2 pilot system adopts the CPU–GPU architecture, and is available to our research group. This architecture allows us to take advantage of the massive parallelization capabilities of gWEGA on many GPU nodes. To further increase screening speed, we also developed an efficient filter to quickly rule out molecules scoring below a certain threshold; this filter can significantly increase the performance of virtual screenings. This article concerns, the GPU implementation of WEGA, its parallelization, and its performance tests.

[a] *X. Yan, Q. Gu, J. Xu*
Research Center for Drug Discovery, School of Pharmaceutical Sciences, Sun Yat-sen University, 132 East Circle at University City, Guangzhou, Guangdong 510006, China
[b] *J. Li*
SciNet Technologies, 9943 Fieldthron St., San Diego, California 92127
[c] *J. Li*
Schrödinger, Inc, 120 West Forty-Fifth Street, 17th Floor, Tower 45, New York, New York 10036
E-mail: jiabo.li@schrodinger.com or junxu@biochemomes.com
Contract/grant sponsor: National High Technology Research and Development Program of China (863 Program); Contract/grant number: 2012AA020307; Contract/grant sponsor: Guangdong Recruitment Program of Creative Research Groups; Contract/grant sponsor: National Natural Science Foundation of China; Contract/grant numbers: 81001372, 81173470; Contract/grant sponsor: National Supercomputer Center in Guangzhou; Contract/grant number: 2012Y2-00048/2013Y2-00045, 201200000037

Methods

WEGA for accurate shape comparison

A shape can be represented by a shape density function.[2,3] In the original articles by Grant et al.,[2,3] the shape density function $G(r)$ can be expressed in terms of the shape density functions of individual atoms and their overlaps:

$$G(r)=1-\prod_i[1-g_i(r)] \tag{1}$$

where,

$$g_i(r)=pe^{-\left(\frac{3p\pi^{1/2}}{4\sigma_i^3}\right)^{2/3}(r-r_i)^2} \tag{2}$$

r is the radius, and p is an adjustable parameter controlling the softness of the Gaussian spheres.

The overlap volume of two molecules is defined as the integral of the product of the Gaussian shape densities of two molecules:

$$V_{AB}^g=\int G_A(r)G_B(r)dr \tag{3}$$

The shape Tanimoto similarity can be defined as

$$S_{AB}=\frac{V_{AB}}{V_{AA}+V_{BB}-V_{AB}} \tag{4}$$

where V_{AA} is the self-overlap volume of molecule A, and V_{BB} is the self-overlap volume of molecule B. It can be proved that the similarity will range from 0 to 1; the similarity will be 1 only when the two shape densities are identical. For efficiency, the so-called FOGA has been adopted in some recent implementations, such as in ROCS and SHAFTS.[14] In FOGA, the shape density of a molecule is represented as simply the sum of the shape densities of all atoms in the molecule:

$$G(r)=\sum_i g_i(r) \tag{5}$$

and the overlap volume of two molecules is the summation of the contributions from all atom pairs, as shown in the following:

$$V_{AB}^g=\sum_{i\in A,j\in B} v_{ij}^g \tag{6}$$

As was established in our previous study, such an approximation leads to significant overestimation of the overlap volume of two molecules. Consequently, such approximations also lead to considerable noise in the shape similarity scores. A simple correction, which is called the WEGA, is introduced to solve this noise problem. In this method, the shape density of a molecule is represented in a linear combination of weighted atomic Gaussian functions as following

$$G(r)=\sum_i w_i pe^{-\left(\frac{3p\pi^{1/2}}{4\sigma_i^3}\right)^{2/3}(r-r_i)^2} \tag{7}$$

where w_i is a weighting factor that is determined by the following empirical formula:

$$w_i=\frac{v_i^g}{v_i^g+0.8665\sum_{j\neq i} v_{ij}^g} \tag{8}$$

The overlap volume of two molecules can be calculated via the following:

$$V_{AB}^g=\sum_{i\in A,j\in B} w_i w_j v_{ij}^g \tag{9}$$

The WEGA method not only significantly improves the accuracy of molecular overlap volume, but also the shape similarity scores. Consequently, the accuracy of shape-based virtual screening is improved (particularly when the shape-feature combo scoring function is used[18]).

GPU-accelerated WEGA

WEGA's accuracy and efficiency motivated us to develop an even more efficient algorithm for virtual screening by taking the advantage of GPUs. The hardware architecture, we used is the NVIDIA TESLA C2050 GPU, which has 3 GB DDR3 RAM (global memory) and 14 multiprocessors (MPs). Each MP contains 32 scalar processors. The corresponding software environment, known as CUDA C,[33] includes a C-like language that provides programmers with a convenient way of harnessing GPUs' parallel computing capabilities.

The implementation of gWEGA is designed for screening very large conformation databases with query structures. To superimpose two molecular conformations, we (1) freeze the coordinates for the query conformation, (2) apply translational and rotating operations on the other conformation, and (3) all torsion angles are fixed during the alignment. As in ROCS implementation, only heavy atoms are considered for molecular shape representation, and all heavy atoms have the same van der Waals carbon radius.

As each query structure will be compared with all database conformations, the conformation of a query structure is loaded onto the constant memory of a GPU card. To find the global optimal alignment of a query structure with a database structure, four initial alignments are considered. For this purpose, four copies of query structure conformations (which differ from each other only by their orientations) are stored in the GPU card's constant memory. Of course, the code can be easily modified for different numbers of initial alignments. Each thread handles a query-conformation pair in gWEGA, and the alignment is optimized using the Newton–Raphson method such that a shape similarity score is obtained. In the current implementation of gWEGA, each thread block consists of 128 threads, which can be divided into four computing warps; each warp consists of 32 computing threads that superimpose 32 conformations onto the query structure.

To speed up computation, the database conformations are preprocessed with the following procedures. For each conformation, the molecular volume and the Gaussian function weights

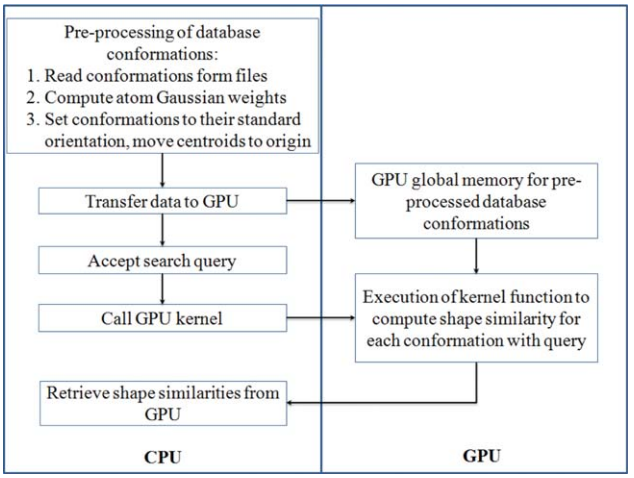

Figure 1. Flowchart of the GPU-implemented parallel WEGA.

of all atoms are precalculated. The conformations are also set to their standard location and orientation. For instance, the centroid of the 3D structure shape is translated to the origin, and the principle axes of the inertia of conformation shape are aligned with the *X*, *Y*, and *Z* coordinate axes, respectively. More details for setting standard orientations for database conformations can be found in Ref. [18]. All pharmacophore features of database conformations are also identified and stored. All the preprocessing for the database conformations is done with the CPU, and this process need only be done once. The preprocessed database conformations, together with the Gaussian weights of all atoms and pharmacophore features information, are then transferred to the GPU global memory. Then, the server is ready for multiple requests of GPU shape searching.

The flowchart of the gWEGA computation is illustrated in Figure 1. In our GPU code design, all database conformations are grouped, and the group size (usually a multiple of eight, such as 32 in the TESLA C2050) is based on GPU card parameters. Each group of 32 conformations is treated as a batch and is associated with a warp. Each conformation is associated with a thread. Thus, parallel superimposing calculations are created, managed, scheduled, and executed simultaneously in GPU cards. To avoid branch divergence within a warp, the database conformations are canonicalized in the above preprocessing so that all conformations within a group have the same number of atoms as the conformation with the largest number of atoms in the group. A simple trick for doing this is to add some dummy atoms with zero weights for conformations with fewer atoms. As each molecule has multiple conformations, the above canonicalization contributes very little overhead in terms of computation. The overhead can be even further reduced if the molecules in a database are sorted according to their numbers of heavy atoms.

The gWEGA kernel is designed in the following manner. Each thread block contains 128 threads, which compute the superposition of 32 database conformations onto the query structure (with its four initial orientations). Each thread within a block handles the superposition of a database conformation onto a query structure with one of the four initial orientations. The GPU kernel function's computational procedure can be shown in the following pseudocode:

```
Parallel execution of warp i (i=1~4) in each block:
{
  Parallel execution of thread j in warp i:
  {
    Load database conformation (C_j)
    according to thread index into local
    memory;
    Read data of initial orientation of
    query (Q_i) from constant memory;
    Iteration
    {
      Compute overlap volume (V_ij) of Q_i
      with C_j;
      Compute first and second derivatives
      (1st−2nd Dervs) of V_ij against
      Cartesian coordinates of all atoms
      in C_j;
      Transform the 1st−2nd Dervs into the
      variables for rigid rotation and
      translation (R-T Vars);
      Apply coordinates transformation to
      all atoms in C_j according toR-T Vars;
      If block convergences conditions are
      fulfilled or the maximum iteration is
      reached, exit iteration;
    }
    Copy V_ij to shared memory;
    Find the maximum volume V_max of V_1j,
    V_2j, V_3j and V_4j;
    Compute Tanimoto Similarity of C_j with
    query using V_max;
  }
}
```

The maximal overlay volume will be used for computing the Tanimoto shape similarity score according to eq. (4).

The parallel processing of the gWEGA is depicted in Figure 2.

Performance of gWEGA

World Drug Index (WDI; version 2010) data were used to test the performance of gWEGA. Empty records and counterions in salts in the WDI dataset were removed with Pipeline Pilot (version 8.5).[34] We got 70,555 chemical structures, from which 4.5 million 3D conformations were generated with the CAESAR algorithm[35] in discovery studio (version 3.5).[36] Six different sized molecules were prepared as individual queries (Fig. 3), and 3D conformations were generated.

The performance of gWEGA was measured by superimposing every query and all the molecules in the WDI database. Our implementation of Gaussian-based molecular shape comparison uses Newton–Raphson optimization method, which is numerically stable, converges fast in a few iterations, and single precision calculation is satisfactory. Our tests indicate that the GPU single

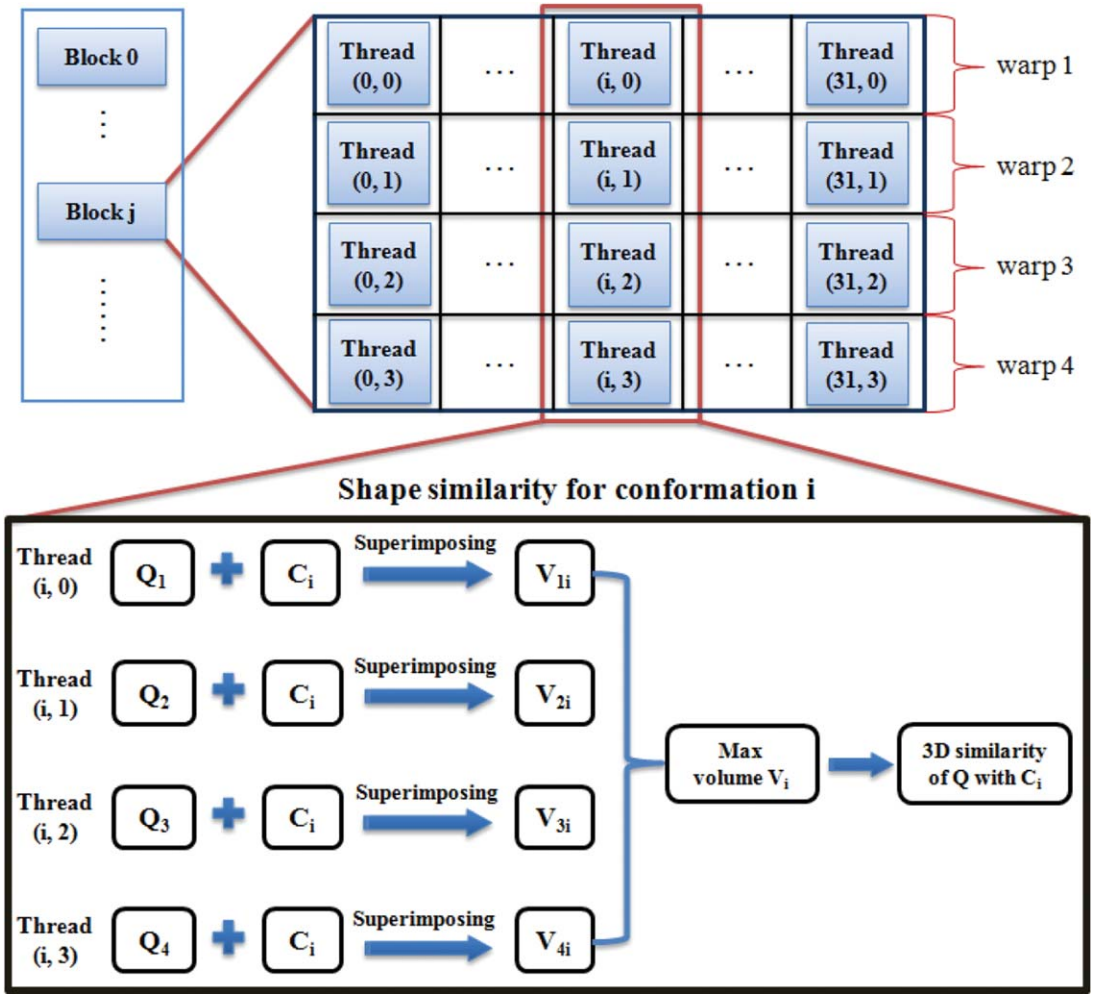

Figure 2. The gWEGA's kernel function scheme. Q_1–Q_4: the four initial orientations of the query structure. C_i: database conformation *i*. V_{1i}–V_{4i}: optimized overlay volumes for a conformation *i* with the query structure (with its four different orientations).

precision calculations are consistent with the CPU double precision calculation. The WEGA's performance was measured for comparison. gWEGA was executed on one NVIDIA Tesla C2050 card (the default configuration of CUDA was adopted; ECC support was enabled) and one quad-core CPU (Intel Xeon W3520, 2.67GHz) for WEGA. The testing results are shown in Figure 4.

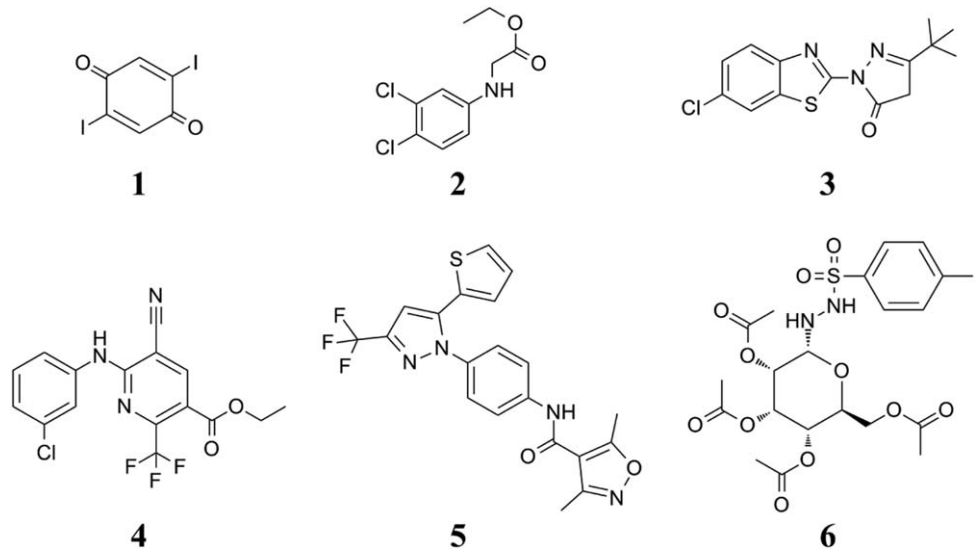

Figure 3. Six different sized query molecules.

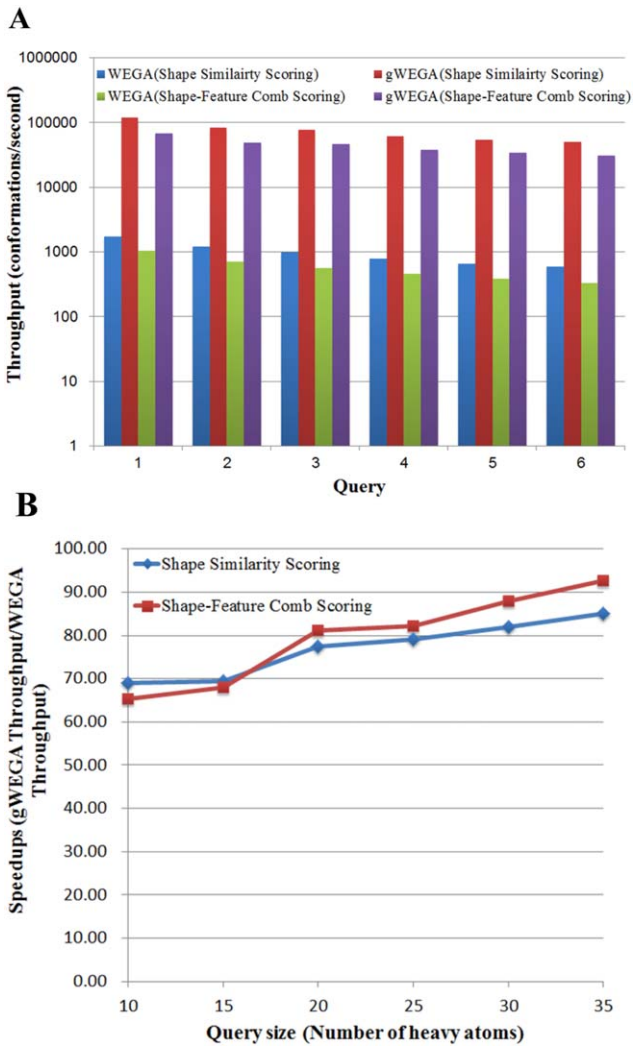

Figure 4. Testing the gWEGA on a computer with a single GPU card. (A): The performance of the gWEGA and WEGA in superimposing six queries and all molecules in the WDI database. (B): The relation between gWEGA speedup and query size. Detailed data can be found in Supporting Information Table S1. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Both gWEGA and WEGA throughputs depend on query size and the average molecular size of a compound library. In our testing data, the average molecular size was 28 heavy atoms. The throughputs of gWEGA and WEGA descend when the query size increases (Fig. 4A).

Superimposing larger conformations is more computation intensive. The strategy is more suitable for taking advantage of GPUs; the speedup (S = gWEGA throughput/WEGA throughput) increases with the query molecule size (Fig. 4B). For a drug-like query (average size ranges from 20 to 25 heavy atoms), gWEGA is about 80 times faster than WEGA with one GPU card.

High Throughput Virtual Screening

Client-server architecture of gWEGA

The above testing was executed on a single GPU card with database conformations preloaded onto GPU memory. For practical efficiency, the preloaded conformation database should be kept in memory for multiple database search queries so that the I/O bottleneck can be avoided. For this purpose, the Client-server architecture was adopted for the gWEGA program design. In our design, only about 500 MB of GPU memory are allocated to hold the database conformations for each GPU server so that the rest of the GPU memory is available for other GPU programs. When there are no search requests, the server will be idle and the computational resource can be used by other GPU programs. Our tests show that other GPU programs, such as the GPU-accelerated Amber, can run concurrently on the same GPU card with the gWEGA server and that gWEGA server performance is not affected by other GPU programs (in terms of the gWEGA server's GPU–CPU timing).

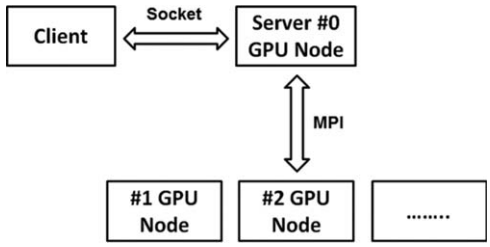

Figure 5. Schematic representation of the client-server architecture of gWEGA with MPI parallelization on the Tianhe-2 supercomputer.

Parallelization of gWEGA

Large conformation databases require significant amounts of memory. This memory can hardly fit onto the memory of a few GPU cards. Moreover, the massive amount of conformation data also requires even more powerful computational resources. A practical solution is to distribute the conformation data on many GPU cards to take advantage of the massive parallelization capabilities of the gWEGA on multiple GPU nodes using message passing interface (MPI). The design can be schematically represented in Figure 5.

Performance test for parallelization

gWEGA was tested with four different settings, which contain 2, 3, 4, and 5 computing nodes (each computing node contains an NVIDIA TESLA C2050 GPU card.), respectively. The performance of gWEGA with one computing node is chosen for comparison. As shown in Figures 6A and 6C, the performance of the parallel gWEGA increases almost linearly with the number of GPU cards. The cost for the communications between computing nodes is small (as shown in Figures 6B and 6D). Comparing Figures 6B and 6D, the speedup of the parallel, when shape-feature combo scoring is adopted, is slightly higher than what results when one adopts shape similarity scoring. This is because fewer virtual screening results are achieved when shape-feature combo scoring is adopted; fewer results lead to lower communications costs between computing nodes, and thus, produces a higher speedup of the parallel.

High throughput virtual screening for the ZINC database

The clean drug-like compound library (CDCL), which is derived from the ZINC database, is used to test gWEGA performance.

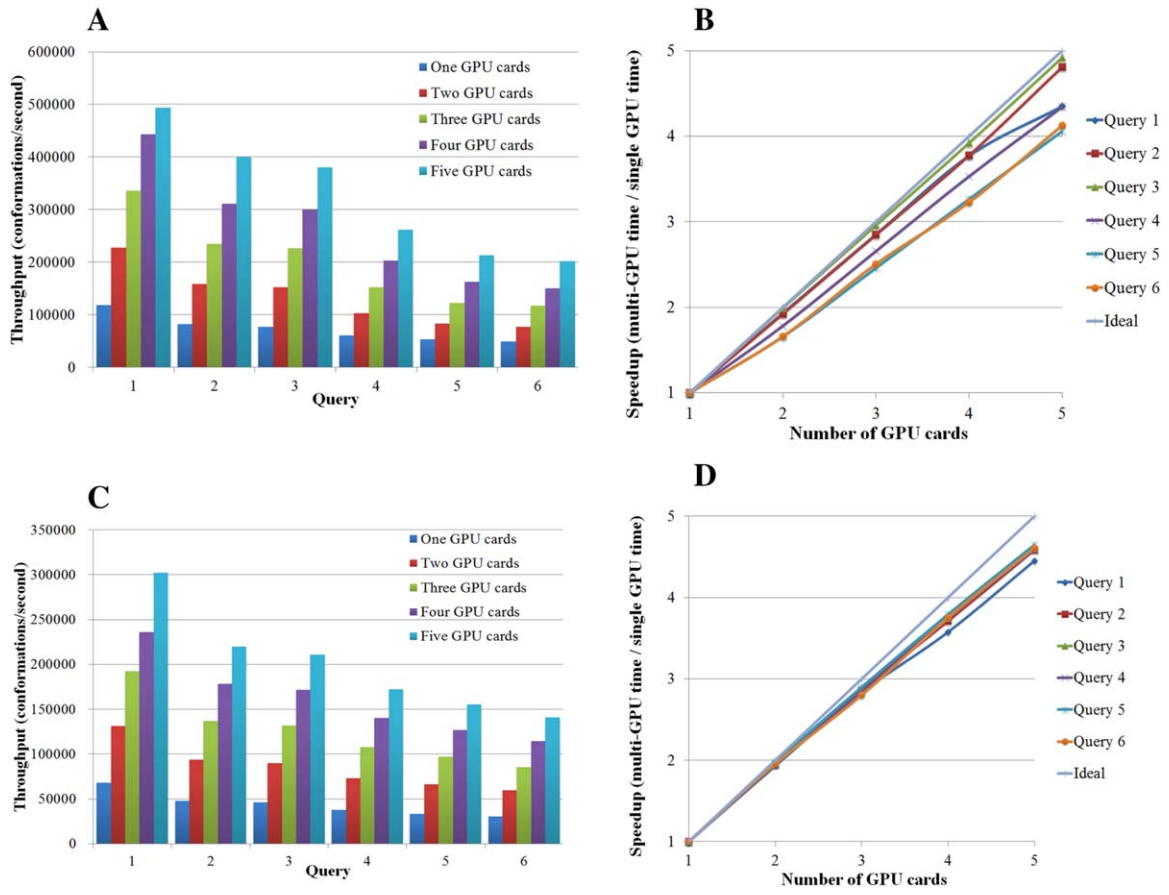

Figure 6. gWEGA performance and its parallel scaling with multiple GPU nodes. (A) The performance of gWEGA on multiple GPU cards for shape-based screening of the WDI database with six query structures. (B) Relation between gWEGA speedup and the number of GPU cards when shape similarity scoring is applied. (C) The performance of a gWEGA that uses shape-feature-based screening on multiple GPU cards. (D) The relation between gWEGA speedup and the number of GPU cards, when the shape-feature combo scoring is applied. Detailed data can be found in Supporting Information Table S2. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

CDCL contains nearly 11 million compounds, and a large conformation database of CDCL was created. As shown in a recent study by Kirchmair et al.[37] that the accuracy of shape-based virtual screening is insensitive to the number of conformations, when the number of conformation is greater than 10. In building our conformation database, a maximum of 10 conformations are generated for each compound using the CAESAR algorithm in discovery studio (version 3.5). This leads to about 110 million conformations for the entire ZINC database of drug-like compounds. The average number of heavy atoms of those conformations is 25. The six chemical structures in Figure 2 are the queries for the high throughput virtual screening. Eighty six computing nodes are used in this test and about 1.3 million conformations for each GPU node. The gWEGA server on each GPU node loads about 1.3 million database conformations into GPU memory during server initialization, and holds the data in GPU memory during the server's lifetime. When a master server receives a query from a client, the query molecule will be dispatched to all gWEGA servers, the GPU kernel will be executed on each gWEGA server, and the results will be collected and sent back to client.

In most virtual screening, we are usually only interested in the highest score hits. For instance, one may only need to retrieve molecules with a similarity score greater than a certain threshold (say 0.8). This observation provides for a shortcut to prefilter most of the database molecules below this threshold in a computationally much less expensive way. This can be achieved by an empirical formula that estimates the upper bound of the similarity score between two molecules using molecular shape descriptors, such as molecule volumes and shape moments. For instance, we can use molecular volumes to estimate the upper bound of shape similarity in the following way. Without losing generality, we assume the volumes of two molecules are V_1 and V_2 and that $V_1 < V_2$. In this case, the upper bound of the shape similarity of the two molecules is V_1/V_2. If this volume ratio is below a defined similarity threshold, then no explicit shape similarity calculation is needed. The similarity score estimation can be done extremely quickly. Therefore, most of those molecules with low similarity can be quickly filtered out without much computational cost. Thus, the overall throughput can be significantly increased. In these high throughput screening campaigns, 15 small molecules with antidiabetic activity are collected as query structures (Fig. 7). The runtimes (including the time for data exchanges between client and server) of the high throughput screening

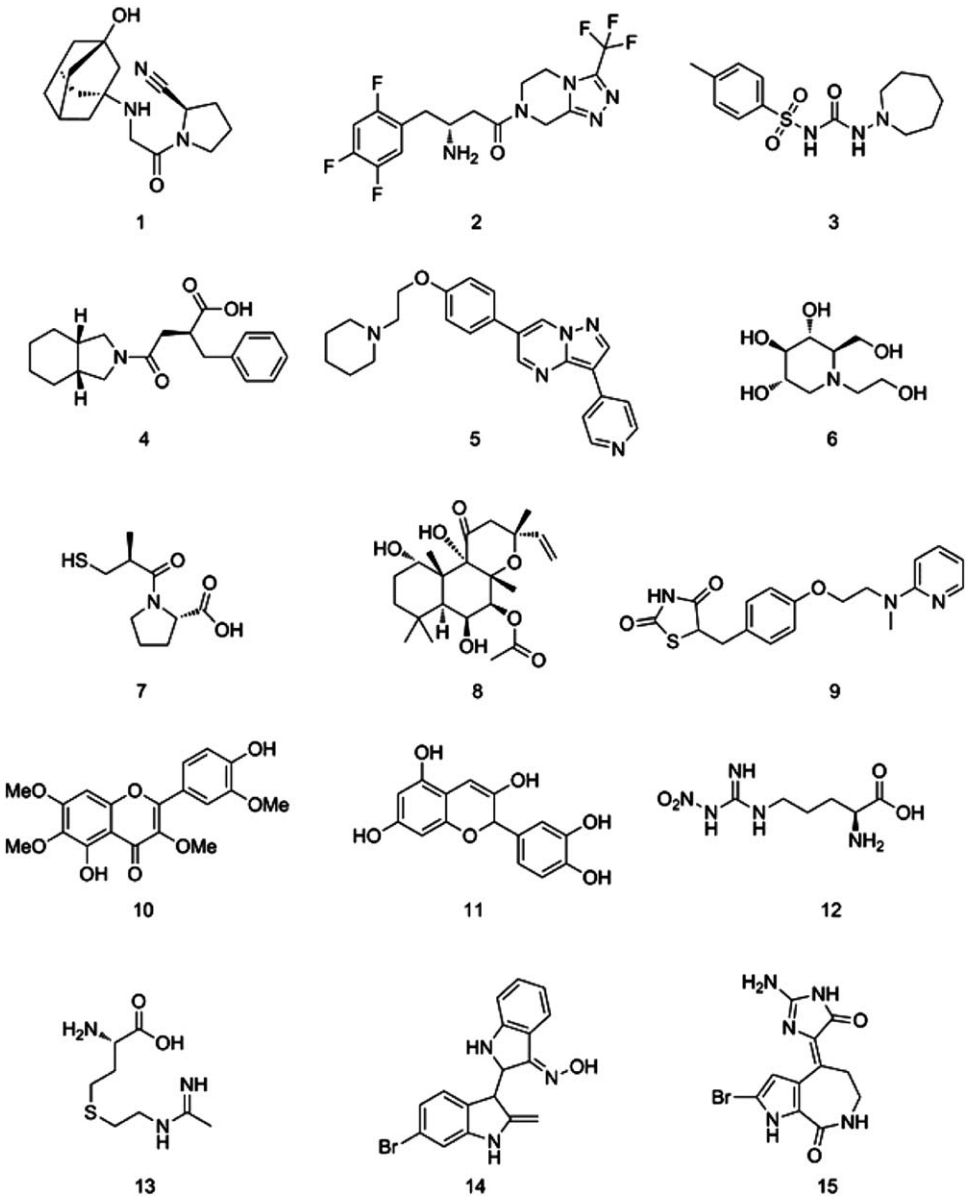

Figure 7. Fifteen antidiabetic queries.

campaigns, with or without the prefiltering method, are listed in Table 1.

As shown in Table 1, the runtime is dramatically reduced by prefiltering when the query size is small (such as the query molecule **6**). This is because in these cases, the query size is quite different from the average size of molecules in the ZINC dataset. The vast majority of molecules are skipped by prescreening and only a small amount of them are further examined by explicit shape comparison using gWEGA. The virtual screening of the entire ZINC drug-like compound database can be accomplished in less than 80 s without prescreening (or less than 40 s with prescreening for all tested query molecules). For a small query molecule with 10–15 heavy atoms, over 60 million conformations can be screened per second and the entire ZINC database can be screened in less than 2 s. Such a high performance enables the gWEGA to screen extremely large virtual libraries, such as GDB-13[22] or even GDB-17,[23] on the Tianhe-2 supercomputer.

Figure 8 shows three superimposing examples for gWEGA. Query is a known LXR agonist with $IC_{50} = 22$ nM; its 3D structure is derived from PDB databank (Access code: 3KFC). Three superimposed molecules 1–3 are taken from Ref. [38]. These molecules are well superimposed with the query; and their bioactivities are consistent with their similarities to the query crystal structure.

Our future work will make the client/server version of gWEGA available on the internet: the gWEGA server will preload large virtual libraries and can then wait for virtual screening requests, such that users or researchers can submit virtual screening requests using the website as a gWEGA client, and can thus obtain results promptly.

Table 1. gWEGA runtime for high throughput virtual screening.

Query#	*T*1 (s)[a]	*T*2 (s)[b]	*T*3 (s)[c]	*T*4 (s)[d]
1	29.4	12.61	56.9	21.48
2	43.75	20.99	65.61	30.71
3	38.68	25.08	62.72	38.81
4	36.34	25.04	54.3	34.53
5	49.03	15.08	77.53	25.59
6	25.79	1.8	48.09	1.92
7	24	1.67	50.81	10.37
8	47.75	16.97	65.41	13.1
9	42.09	19.98	69.86	34.56
10	37.92	16.49	59.31	23.12
11	29.33	11.23	53.22	26.05
12	21.87	1.89	49.93	2.55
13	23.7	2.81	46.68	4.27
14	40	17.16	55.9	33.37
15	28.13	5.78	60.06	27.11

[a] Shape similarity scoring with threshold 0.8 without prefiltering. [b] Shape similarity scoring with threshold 0.8 with prefiltering. [c] Shape-feature combo scoring with threshold 0.5 with out prefiltering. [d] Shape-feature combo scoring with threshold 0.5 with prefiltering.

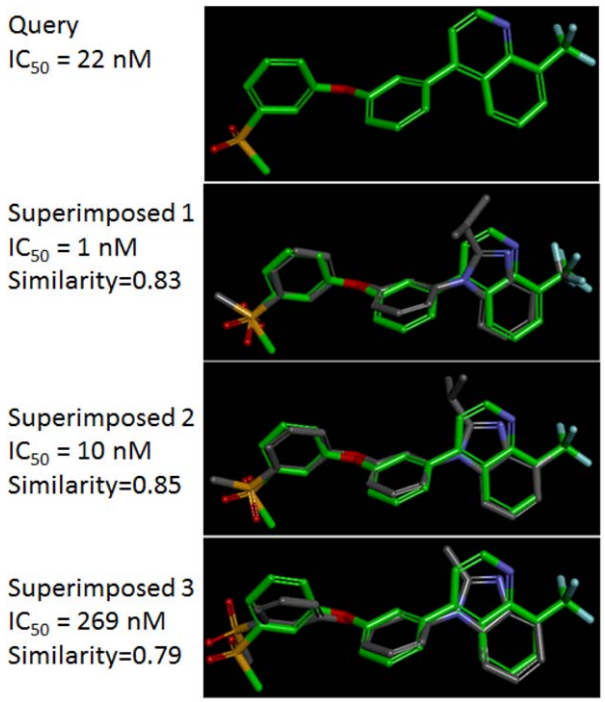

Figure 8. Examples demonstrating the accuracy of molecular superimposition of gWEGA. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Conclusions

In this work, we demonstrated that the GPU-gWEGA can speed up accurate molecular shape comparisons by nearly two orders of magnitude with a single GPU card. Using MPI, the significant scaling of the parallel on multiple GPU nodes has been achieved in the National Supercomputer Center in Guangzhou. In our tests with 86 GPU nodes, the entire ZINC drug-like database was screened in seconds; there are no limits for the number of GPU nodes that can be used for large-scale parallel screening. One important idea we introduced in the article is that by preloading the database conformations onto GPU memory, and by limiting the dedicated GPU memory for the gWEGA server to 500 MB on each GPU card, we eliminate the I/O bottleneck for database loading while allowing other GPU-based applications to be run on the same GPU cards when the gWEGA server is idle. Another idea presented in this article is that fast filtering can be done by estimating the upper bound of the similarity score (either via a pure shape or shape-feature combo score) between a query structure and database structures. The combination of the above technologies makes it possible to routinely screen even very large libraries.

Acknowledgment

The authors thank Mr. Heming Xu of Columbia University for his comments and corrections that help improve the manuscript.

Keywords: graphic processing unit · molecular superimposing · virtual screening · drug discovery · algorithm

How to cite this article: X. Yan, J. Li, J. Qiong Gu, J. Xu *J. Comput. Chem.* **2014**, *35*, 1122–1130. DOI: 10.1002/jcc.23603

Additional Supporting Information may be found in the online version of this article.

[1] A. Nicholls, G. B. McGaughey, R. P. Sheridan, A. C. Good, G. Warren, M. Mathieu, S. W. Muchmore, S. P. Brown, J. A. Grant, J. A. Haigh, N. Nevins, A. N. Jain, B. Kelly, *J. Med. Chem*. **2010**, *53*, 3862.
[2] J. A. Grant, B. T. Pickup, *J. Phys. Chem*. **1995**, *99*, 3503.
[3] J. A. Grant, M. A. Gallardo, B. T. Pickup, *J. Comput. Chem*. **1996**, *14*, 1653.
[4] A. Nicholls, N. E. MacCuish, J. D. MacCuish, *J. Comput. Aided Mol. Des*. **2004**, *18*, 451.
[5] F. J. Ballester, W. G. Richards, *J. Comput. Chem*. **2007**, *28*, 1711.
[6] R. J. Morris, R. J. Najmanovich, A. Kahraman, J. M. Thornton, *Bioinformatics* **2005**, *21*, 2347.
[7] L. Mavridis, B. D. Hudson, D. W. Ritchie, *J. Chem. Inf. Model*. **2007**, *47*, 1787.
[8] M. J. Vainio, J. S. Puranen, M. S. Johnson, *J. Chem. Inf. Model*. **2009**, *49*, 492.
[9] G. M. Sastry, S. L. Dixon, W. Sherman, *J. Chem. Inf. Model*. **2011**, *51*, 2455.
[10] X. Liu, H. Jiang, H. Li, *J. Chem. Inf. Model*. **2011**, *51*, 2372.
[11] A. C. Good, E. E. Hodgkin, W. G. Richards, *J. Chem. Inf. Comput. Sci*. **1992**, *32*, 188.
[12] A. C. Good, W. G. Richards, *J. Chem. Inf. Comput. Sci*. **1993**, *33*, 112.
[13] B. B. Masek, A. Merchant, J. B. Matthew, *J. Med. Chem*. **1993**, *36*, 1230.
[14] .W. Lu, X. Liu, X. Cao, M. Xue, K. Liu, Z. Zhao, X. Shen, H. Jiang, Y. Xu, J. Huang, H. Li, *J. Med. Chem*. **2011**, *54*, 3564.
[15] P. C. D. Hawkins, A. G. Skillman, A. Nicholls, *J. Med. Chem*. **2007**, *50*, 74.
[16] J. Venhorst, S. Nunez, J. W. Terpstra, C. G. Kruse, *J. Med. Chem*. **2008**, *51*, 3222.
[17] T. S. Rush, III, J. A. Grant, L. Mosyak, A. Nicholls, *J. Med. Chem*. **2005**, *48*,1489.
[18] X. Yan, J. Li, Z. Liu, M. Zheng, H. Ge, J. Xu, *J. Chem. Inf. Model*. **2013**, *53*, 1967.
[19] E. Bolton, Y. Wang, P. A. Thiessen, S. H. Bryant, Annual Reports in Computational Chemistry, Vol. *4*; American Chemical Society: Washington, DC, **2008**.
[20] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, R. G. Coleman, *J. Chem. Inf. Model*. **2012**, *52*, 1757.
[21] T. Fink, J. L. Reymond, *J. Chem. Inf. Model*. **2007**, *47*, 342.

[22] L. C. Blum, J.-L. Reymond, *J. Am. Chem. Soc*. **2009**, *131*, 8732.
[23] L. Ruddigkeit, R. van Deursen, L. C. Blum, J. L. Reymond, *J. Chem. Inf. Model*. **2012**, *52*, 2864.
[24] P. Liu, D. K. Agrafiotis, D. N. Rassokhin, E. Yang, *J. Chem. Inf. Model*. **2011**, *51*, 1807.
[25] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, V. S. Pande, *J. Comput. Chem*. **2009**, *30*, 864.
[26] J. E. Stone, J. S. Phillips, P. L. Freddolino, D. J. Hardy, L. G. Trabuco, K. Schulten, *J. Comput. Chem*. **2007**, *28*, 2618.
[27] A. W. Gotz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, R. C. Walker, *J. Chem. Theory Comput*. **2012**, *8*, 1542.
[28] A. D. Stivala, P. J. Stuckey, A. I. Wirth, *BMC Bioinformatics* **2010**, *11*, 446.
[29] Q. Liao, J. Wang, Y. Webster, I. A. Watson, *J. Chem. Inf. Model*. **2009**, *49*, 2718.
[30] I. S. Haque, V. S. Pande, W. P. Walters, *J. Chem. Inf. Model*. **2010**, *50*, 560.
[31] I. S. Haque, V. S. Pande, *J. Comput. Chem*. **2010**, *31*, 117.
[32] OpenEye Scientific Software Inc., FastROCS, Version 1.0; OpenEye Scientific Software Inc.: Santa Fe, NM, **2011**.
[33] NVidia, NVIDIA CUDA C Programming Guide 4.0; NVidia: Santa Clara, CA, **2011**.
[34] Accelrys Software Inc., Pipeline Pilot, Version 8.5; Accelrys Software Inc.: San Diego, CA, **2012**.
[35] J. Li, T. Ehlers, J. Sutter, S. Varma-O'Brien, J. Kirchmairs, *J. Chem. Inf. Model*. **2007**, *47*, 1923.
[36] Accelrys Software Inc., Discovery Studio, Version 3.5; Accelrys Software Inc.: San Diego, CA, **2012**.
[37] J. Kirchmair, S. Distinto, P. Markt, D. Schuster, G. M. Spitzer, K. R. Liedl, G. Wolber, *J. Chem. Inf. Model*. **2009**, *49*,678.
[38] W. Zhao, Q. Gu, L. Wang, H. Ge, J. Li, J. Xu,. *J. Chem. Inf. Model*. **2011**, *51*, 2147.

Received: 3 January 2014
Revised: 6 March 2014
Accepted: 14 March 2014
Published online on 12 April 2014