Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Molecular Graphics and Modelling 22 (2003) 71–82
Journal of Molecular Graphics and Modelling
www.elsevier.com/locate/JMGM

A web-based platform for virtual screening

Paul Watson*, Marcel Verdonk, Michael J. Hartshorn

Astex Technology Ltd., 436 Cambridge Science Park, Milton Road, Cambridge CB4 0QA, UK

Received 5 November 2002; received in revised form 11 February 2003; accepted 14 March 2003

Abstract

A fully integrated, web-based, virtual screening platform has been developed to allow rapid virtual screening of large numbers of compounds. ORACLE is used to store information at all stages of the process. The system includes a large database of historical compounds from high throughput screenings (HTS) chemical suppliers, ATLAS, containing over 3.1 million unique compounds with their associated physiochemical properties (ClogP, MW, etc.). The database can be screened using a web-based interface to produce compound subsets for virtual screening or virtual library (VL) enumeration. In order to carry out the latter task within ORACLE a reaction data cartridge has been developed. Virtual libraries can be enumerated rapidly using the web-based interface to the cartridge. The compound subsets can be seamlessly submitted for virtual screening experiments, and the results can be viewed via another web-based interface allowing ad hoc querying of the virtual screening data stored in ORACLE.

Keywords: Virtual screening; Virtual library enumeration; Docking; GOLD; ORACLE

1. Introduction

Virtual screening using protein–ligand docking (known hereafter as VS) has, in recent years, become recognised as a viable source of lead compounds in structure-based drug design [1–5]. With advances in hardware and the rapidly decreasing price of Linux clusters, large scale VS of hundreds of thousands of compounds in less than 1 day is now a reality [6]. However, whilst the computing power is readily available to achieve this, typically the task is fraught with difficulties.

In the first stage of the process, compounds have to be chosen for VS. These will usually be historic compounds, that is, from high throughput screening (HTS) chemical suppliers or available chemical directory [7] (ACD), or from a virtual library (VL) of compounds, enumerated using a known chemical reaction. In the case of historic compounds, the preparation of the set of compounds is highly time consuming as the file formats for the compounds may vary. As such, specialist skills and/or software will be required to amalgamate the files and/or produce the file format(s) of choice for VS. Additionally, producing subsets of these compounds that satisfy a molecular profile (such as undesirable/desirable substructures, Lipinski criteria [8], etc.) is also a technically skilled task. The creation of VLs for VS is arguably more difficult, again requiring specialist skills and software.

Once these files are prepared, the VS experiment can be run, however the analysis of the results of a virtual screening run is by no means a trivial task. It is widely known that docking scores are not always reliable in their rankings of docked compounds [3], and as such, it is difficult to apply a 'docking score cut-off' based on a single docking score (as opposed to consensus scoring) to remove the undesirable compounds. In addition to this, viewing VS results usually requires modelling software to be installed on specialist machines (e.g. SGIs), which greatly reduces the availability of the VS results.

Automated systems for compound selection and VL enumeration are reported in the literature. These include ADEPT [9], the cyclops system at Novartis [10] and the system developed at Vertex [11]. These are geared towards compound selection for HTS and the design of combinatorial libraries and, as such, are not fully integrated with the running and analysis of VS experiments. In addition the systems are all based on flat files, making relational access to the data more complex.

This paper describes a fully integrated web-based VS platform allowing selection of compounds from chemical suppliers, enumeration of VLs and the ability to run, and view the results of VS experiments. In order to accomplish this we have employed the ORACLE relational database

* Corresponding author. Tel.: +44-1223-226-285; fax: +44-1223-226-201.
E-mail address: p.watson@astex-technology.com (P. Watson).

1093-3263/03/$ – see front matter © 2003 Elsevier Science Inc. All rights reserved.
doi:10.1016/S1093-3263(03)00137-2

and a client-server model. The advantages of this system are numerous. ORACLE is the industry standard relational database offering highly efficient queries via a simple structure query language (SQL). ORACLE is highly scalable allowing the storage of large amounts of data with a minimal decrease in performance. Using a client-server model it is possible to centralise all the software pertaining to the VS system (i.e. docking code, etc.). The interfaces themselves are all perl/Javascript/JAVA CGI programs communicating with ORACLE and so there is no client side installation of software (IE or Netscape only).

2. Data, software and hardware

The VS set up is described in Fig. 1. The first stage in the process is pre-screening the ATLAS database in order to produce a compound subset (stored in ORACLE) for VS (COMPOUND SUBSET in Fig. 1) or for VL enumeration (REACTANT SET 1 and REACTANT SET 2 in Fig. 1). If VL enumeration is required, this can be carried out using the reaction data cartridge, the results of which are again stored in ORACLE (VIRTUAL LIBRARY in Fig. 1). The compound subsets are combined with protein target data (TARGET DATA in Fig. 1). The VS jobs are then farmed out to a Linux cluster (using the VS SCHEDULER) and docked using GOLD [12–14]. The results of the VS are then stored in ORACLE (VS DATABASES in Fig. 1) and can be queried using the VS results interface (VS INTERFACE in Fig. 1).

2.1. ATLAS database

The ATLAS database (Astex technology library of available substances) is a database of 'historical compounds' from HTS chemical suppliers. It contains 3,179,278 unique compounds from 714 suppliers. The database itself is stored in ORACLE. ATLAS comprises four tables, ATLAS COMPOUNDS, ATLAS SUPPLIERS, ATLAS PACK and SUPPLIERS; these are shown in Fig. 2 in an entity relationship diagram. The ATLAS COMPOUNDS table contains the desalted neutral SMILES [15,16] (SMILES (desalted)) and its associated 1D chemical information as well as a functional group fingerprint (FG FP) indicating the presence or absence of 27 carefully chosen functional groups within the molecule. The ATLAS SUPPLIERS table contains the compound including, if present, the salt (SMILES (salt)), the name of the supplier (which is found via a join with the SUPPLIERS table), and the catalogue number as well as a unique identifier (UNIQUE ID). Finally any pricing, pack sizes, and quantity information can be found in the ATLAS PACK table.

2.2. Pre-screening

ATLAS can be pre-screened to produce a compound subset for VS or a set of reactants to be used in VL enumeration. It can be pre-screened in terms of two types of information. The first of these is 1D chemical information, e.g. a molecular weight range or a maximum ClogP. The second of these is 2D chemical information (e.g. substructure searching). The interface supports the logical combination of up to 12 drawn substructures that are input using JME [17]. In addition to the drawn substructures there is a collection of 27 functional groups the user can insist are present or not present in the retrieved molecules.

All of these queries are carried out by simply issuing SQL statements from the web-based interface to ORACLE. For the 1D chemical information, these are numeric-based queries and are easily handled using SQL. The Daylight data cartridge (DayCart [18]) is used to carry out the substructure searching allowing the input of any SMILES or SMARTS pattern. In the last case of the 27 functional groups, custom-made fingerprints (FG FP in the ATLAS COMPOUNDS table) are used, which are created for each molecule as they are loaded into the database. These are

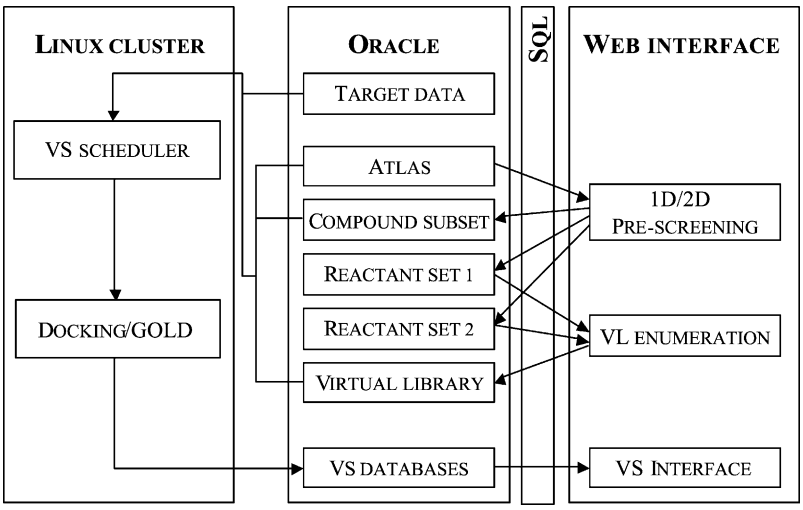

Fig. 1. VS workflow.

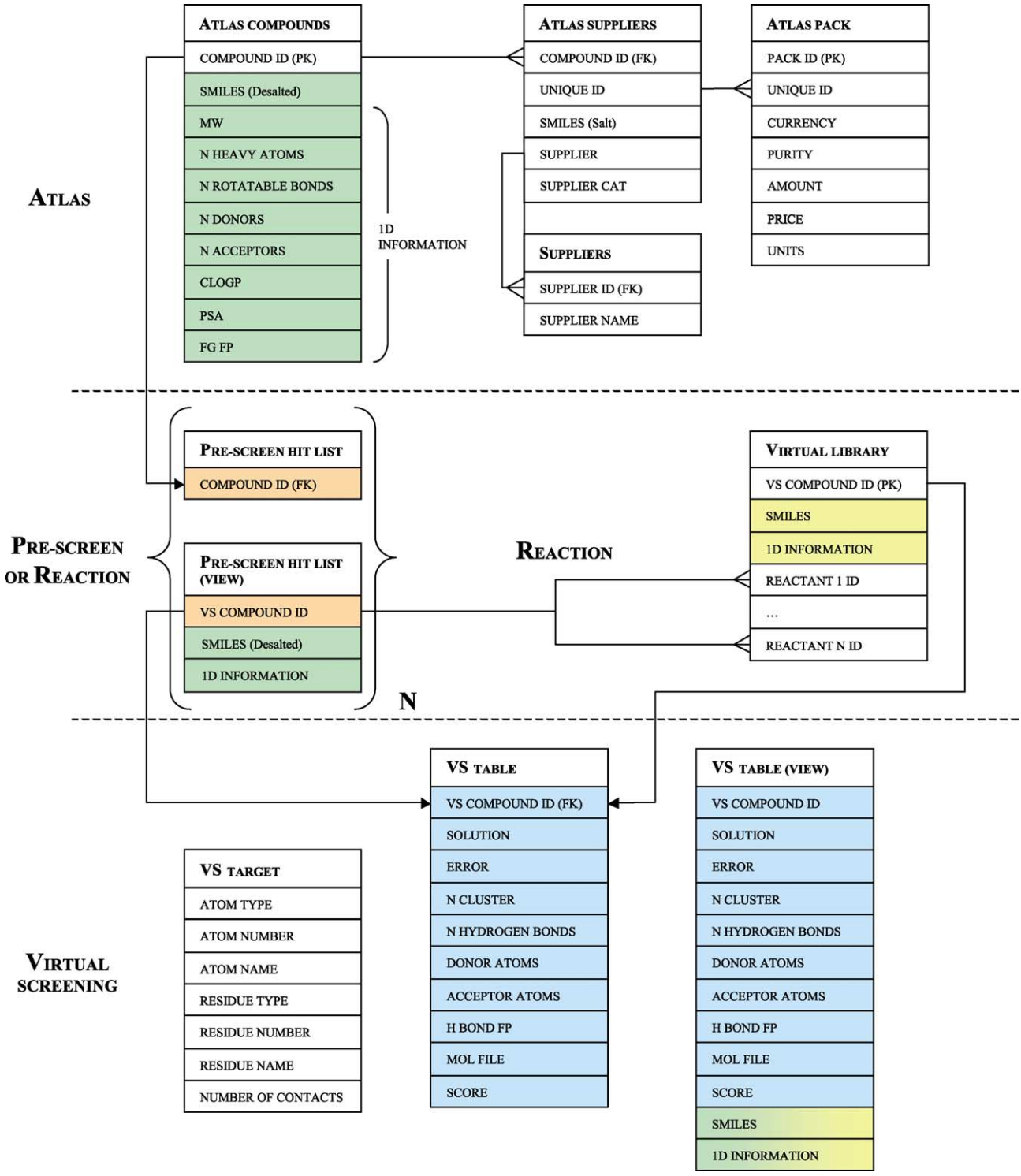

Fig. 2. Entity relationship diagram showing the relationships between the various tables used in the VS process. The colouring of the cells, representing table columns, illustrates from which table data has been joined to form the ORACLE views. For example, in the case of the VS TABLE (VIEW) the view has been created by fusing data from the VS TABLE and the PRE-SCREEN HIT LIST (VIEW) or VIRTUAL LIBRARY table by joining on the VS COMPOUND ID column.

```
SELECT
      compound_id
FROM
      atlas_compounds
WHERE
      matches ('[#8;D1]c1cccc2ccccc12', smiles) =
1 and not
      matches ('[#8;D1]c1cccc2cc(Cl)ccc12',
smiles) = 1 and
      mw between 250 and 400;
```

Fig. 3. Example pre-screening SQL statement. In this case the query is selecting the COMPOUND ID number from the ATLAS COMPOUNDS table where it matches the first SMARTS pattern, but not the second with a molecular weight (mw) in the range indicated. The 'matches' functions are part of the functionality provided by the DayCart data cartridge. This is an example of the sort of statement prepared and executed by the pre-screening user interface.

```
SELECT
react (
      '([#6,#7:1][N:2]([H])[H:3]).
       ([#6,#7:5][C:4](=[O:6])O)
       >>
      [#6,#7:5][C:4](=[O:6])[N:2]([#6,#7:1])[H:3]',
      prescreen1.smiles,
      prescreen2.smiles)
  FROM
      prescreen1, prescreen2
```

Fig. 4. Example SQL statement using the reaction cartridge. In this example pre-screens have already been run resulting in the PRE-SCREEN HIT LIST (VIEW) views; prescreen1 and prescreen2 comprising columns containing the reactant molecule SMILES strings. The reaction cartridge is simply a function ('react') taking the SMIRKS string (in this case a peptide coupling) and up to four sets of reactants contained in ORACLE views resulting from pre-screening (in this case prescreen1 and prescreen2).

created to increase the speed of the pre-screening. The fingerprints are stored as VARCHAR2 data types in ORACLE and are indexed. It is then possible to use simple comparative SQL statements to find molecules in the database satisfying the query fingerprint.

Any combination of the 1D and 2D information can be used to produce a hit list. A typical example of an SQL statement from a pre-screen is shown in Fig. 3. After the hit list is produced an ORACLE view is created from the hit list table and the ATLAS COMPOUNDS table, joining on the COMPOUND ID column. This is illustrated in the entity diagram shown in Fig. 2. The purpose of this is to create a subset of the ATLAS COMPOUNDS table, whilst saving space in the database (views are evaluated dynamically and as such do not use any disk space).

The compounds produced from the query can be visually inspected. To this end CACTVS [19] is used to dynamically convert the SMILES strings into pictures. Supplier information can be viewed by clicking on the pictures of the structures. An example of the use of the interface is provided below.

2.3. Virtual library (VL) enumeration

Compound subsets produced from ATLAS can be used to enumerate VLs for VS. These libraries are created and stored in ORACLE. We have created an ORACLE reaction data cartridge, based around the Daylight reaction toolkit [20], allowing reactions to be carried out within ORACLE using simple SQL statements. The speed of the reaction toolkit is such that around 100 molecules per second can be enumerated. The first stage in library enumeration is to create a reaction and store it in ORACLE. JME is used for this, allowing the user to input a reaction that is then stored as a SMIRKS. Next, hit lists from the pre-screening stage (PRE-SCREEN HIT LIST (VIEW) views in Fig. 2) can be combined with a reaction SMIRKS to produce the enumerated library (VIRTUAL LIBRARY table in Fig. 2). It is also possible to limit the products of the reaction to a given molecular weight or ClogP profile. The reaction cartridge supports up to four sets of reactants. An example SQL statement showing the way in which the reaction cartridge works is shown in Fig. 4.

2.4. Virtual screening (VS)

Tables or views of molecules produced from pre-screening or VL enumeration can be docked against a target of choice. This first step is to create the empty VS tables in ORACLE (VS TARGET and VS TABLE). The VS TABLE table contains information for each molecule from the docking run (such as the docking scores, the number of hydrogen bonds the molecule makes with the receptor and the atoms involved, the donor and acceptor atoms, a fingerprint used to represent the hydrogen bonding of the molecule to the receptor (used by the VS results interface for filtering), and the paths to the files containing the docked compounds). The VS TARGET table is used to store information about the active site of the target protein (such as the nature of the polar atoms (i.e. donor or acceptor), the number and name of the atoms as stored in the protein file, the residues and residue numbers that the atoms are from and the number of hydrogen bonds formed for each of these atoms over the entire VS run). These tables are then used to store the information from the individual dockings and are updated dynamically as each of the compounds is docked.

The SMILES from either the PRE-SCREEN HIT LIST (VIEW) or the VIRTUAL LIBRARY tables are written to a file

on the server and charged using in-house charging rules at the appropriate pH. This is done using a number of SMARTS patterns denoting functional groups that need to be ionised/protonated. The charged SMILES are then written to a file. In the next stage the VS scheduler is invoked to farm the VS jobs out to the Linux cluster. This initially starts docking jobs in batches of five, enabling initial results of the VS run to be viewed very quickly after it has been submitted to detect if anything has been set up incorrectly. If this is the case, the VS run can be stopped and the parameters altered. After this 'grace period' the jobs are then farmed out to each processor in batches of fifty molecules. For each molecule in each batch the protonated SMILES is converted to a low energy 3D conformer using CORINA [21], and this molecule is saved on the server in the SDF file format. If any stereochemistry is defined in the SMILES string, this is carried over into the 3D representation. If the stereochemistry is not defined an arbitrary stereoisomer is chosen. In future versions of the VS platform this will be amended so that all possible stereoisomers of these 'ambiguous' SMILES are docked. At present molecules are docked in the tautomeric forms stored in ATLAS. This will be amended using a reaction-based approach where selected tautomers will be produced for molecules containing given functional groups. All these tautomers will then be docked against the given target.

The docking itself is carried out using an in-house version of GOLD [12–14] developed in collaboration with the CCDC [22]. As the VS jobs finish, the information from the docking is post-processed and inserted into the VS tables in ORACLE. The VS TABLE (VIEW) view (shown in Fig. 2) is created at the start of the VS run by joining the SMILES and 1D property information from either the PRE-SCREEN HIT LIST (VIEW) view or the VIRTUAL LIBRARY table (depending on the source of the compounds) and the VS TABLE (shown in Fig. 2). It is then possible to view the VS results as each job finishes (ORACLE views are updated dynamically as new information is inserted into the parent tables). All the VS tables can then be queried by the VS results interface in order to mine the results. After potential compounds have been identified they can be stored in a hit list.

2.5. Database manager

Pre-screening hit lists, VLs and VS runs/hit lists can be managed using a web-based database manager interface. This allows the user to delete pre-screen searches, VLs and VS runs/hit lists. VS runs can also be rescored using different scoring functions. The output of the rescore is stored in ORACLE and can be used to consensus score a VS (i.e. combining with the initial docking score), which is considered more reliable than using one scoring function [3].

2.6. Hardware

ORACLE is housed on a Sun V880, with dual 750 MHz SunSparc III processors sharing 4 Gb of RAM, running the Solaris 8.0 operating system. The VS is carried out on a Linux cluster comprising 84 1 GHz Pentium III processors, each pair having 512 Mb of RAM, running Red Hat Linux version 7.1.

2.7. ORACLE integration

There were two main issues in the production of this VS platform regarding the use of the ORACLE relational database. The first of these is the creation and curation of ATLAS. Here three problems needed to be addressed: the conversion of the HTS supplier SD files to SMILES strings, the identification of the unique parent compounds and supplier records and efficiently loading the resulting flat files into ORACLE.

The SD files are converted to the SMILES representations using a modified version of the Daylight mol2smi contributed program [23]. The program has been modified to remove any salts, and calculate the molecular properties (shown in Fig. 2) of the parent molecule. This outputs a set of flat files containing the parent SMILES, original SMILES (i.e. including the salts if present), the molecular properties and the supplier information (i.e. name, catalogue number, pack sizes, etc.). At this stage of the process, the output flat files contained well over 7 million compound records.

These files are then post processed to remove any duplicate parent compounds, but retain the additional supplier information if the parent is available from a different supplier or from the same supplier in a different salt form.

In order to identify the unique parent compounds the canonical nature of SMILES strings was exploited. Here a perl program was written using the parent SMILES strings as a key in a hash to determine the unique compounds. This idea was extended to determine the unique supplier records where a combination of the original SMILES (including the salt if present), the catalogue number and the supplier name was used as the hash key. By doing this it was possible to assign unique identification numbers to each unique compound in the ATLAS COMPOUNDS, ATLAS SUPPLIERS and ATLAS PACK tables. This process produced three text files, each containing the data for the aforementioned tables. At this stage the file for the ATLAS COMPOUNDS table contained the 3,179,278 rows of data mentioned above, the file for the ATLAS SUPPLIERS table contained 7,702,239 rows of data and the ATLAS PACK file contained 2,451,381 rows of data.

Loading such a large quantity of data into ORACLE is not a trivial task and cannot be done efficiently using normal SQL commands. Instead the SQL* Loader program was used. This allows data from fixed format or character delimited flat files to be loaded into ORACLE quickly and efficiently. As the compounds already have unique identifiers assigned, the data can be loaded using the extremely efficient 'direct-mode'. Using this approach it was possible to load ~1 million rows of data per minute on the ORACLE server described above. After the compounds are loaded the necessary indexes can be created.

Fig. 5. Schematic illustrating communication between ORACLE and an external application via a perl wrapper using a bi-directional pipe. Data is fed from ORACLE via the Daylight pipetalk protocol to the perl wrapper program. This is then piped to the target application which returns the line buffered output via the bi-directional pipe. This is then fed back to ORACLE using the pipetalk protocol once again.

The second major issue concerning ORACLE is communication between the database and external programs. An example of this situation can be found in the calculation of the functional group fingerprints. Here a 27-bit fingerprint is calculated via an SQL statement that implements an external program (FG FP in Fig. 2).

One of the ways to do this is via a data cartridge where the data contained within ORACLE can be sent to external programs and the response is sent back to ORACLE. DayCart, developed by Daylight, and the reaction cartridge, developed by ourselves, are good examples of these systems. However, the implementation of a data cartridge is a non-trivial task and as such we decided to make use of the program object model provided by DayCart.

The program object model allows two way communication between ORACLE and external procedures via the pipetalk protocol [24], e.g. the ClogP calculation program in DayCart. If other programs are to be implemented in the same fashion, the Daylight program object library needs to be implemented in the code for your program of choice. An

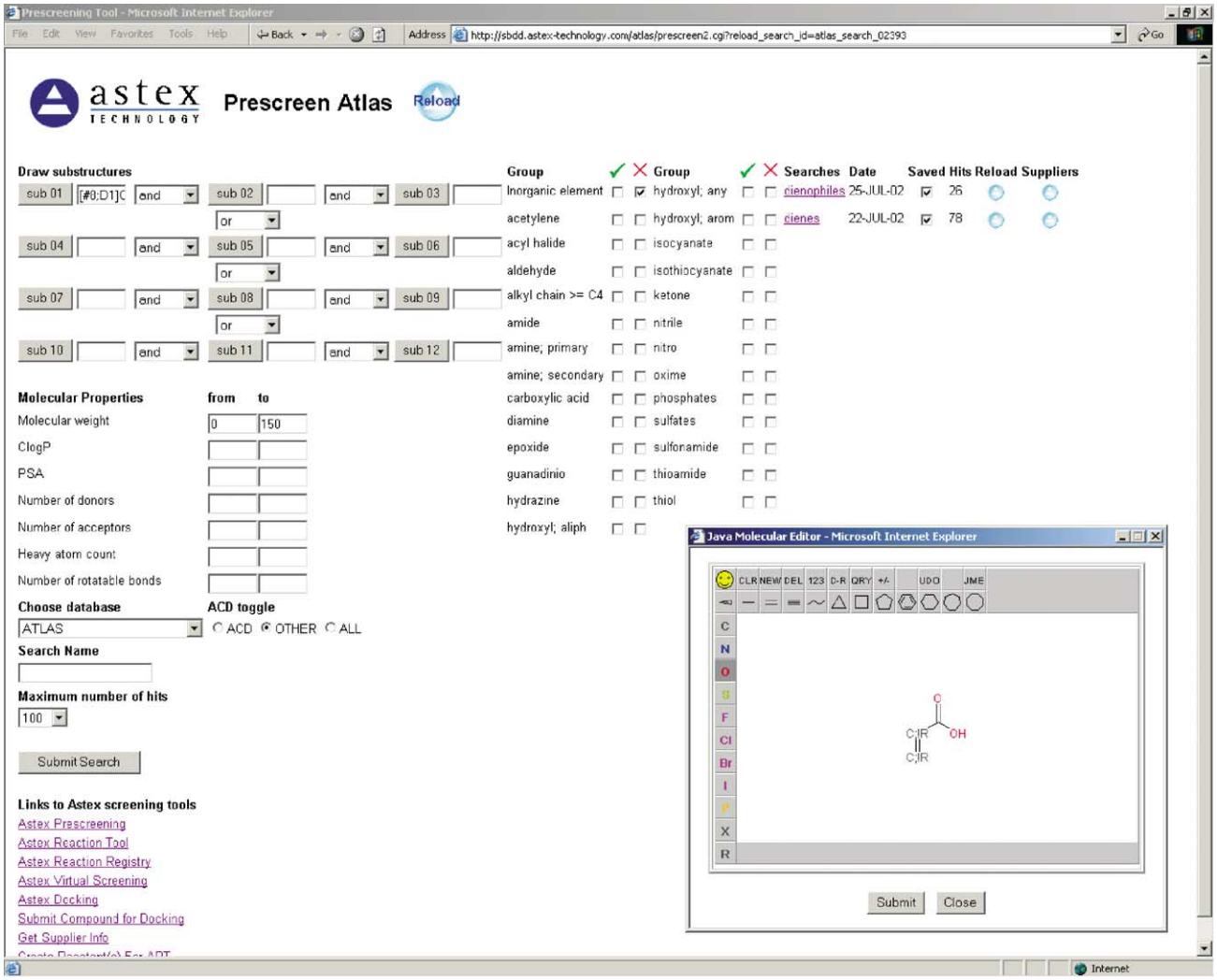

Fig. 6. Pre-screening interface screenshot. The JME pop-up appears after clicking one of the substructure buttons. In this example the interface is being used to search for reactants for VL enumeration using the Diels–Alder reaction. Here the search is for dienophiles containing a carboxylic acid group with a maximum molecular weight of 150.

alternative to this is to execute the program using a simple perl wrapper using a bi-directional pipe [25]. This has the advantage of only using standard perl libraries and it only starts the target program once per database session. The only drawback is that the target program must have line buffered output. A schematic describing the implementation of this protocol is shown in Fig. 5. This protocol is ideal for passing a SMILES string to an external process in order to calculate simple molecular properties, or the functional group fingerprints described above.

3. Examples

3.1. Pre-screening interface

Fig. 6 shows a screen shot of the pre-screening interface. The top-left hand corner contains the area where drawn substructures can be entered. By clicking one of the subXX buttons a pop-up window appears containing JME, allowing the user to enter a substructure. The substructures can be combined using Boolean logic and are grouped such that they can be considered bracketed on each row. Below this area is the 1D chemical property area in which any range, maximum or minimum of any of the molecular properties can be added to the query. To the right of the drawn substructures are the functional group fingerprints. Below the 'molecular properties' dialogue boxes the user can select which database is to be searched (i.e. all of ATLAS or supplier subsets). After this is selected, a name for the hit list is given and the search can be submitted. In Fig. 6 the substructure searched for is a dienophile for use in a Diels–Alder reaction. In this case the substructure must also contain a carboxylic acid group and have a molecular weight less than 150.

The hit lists are all stored in ORACLE. Previous searches are displayed on the top right hand side of the page and can be browsed. The suppliers for each of the compounds in the hit list can be displayed and the searches can be reloaded in order to further refine the query if desired. Fig. 7 shows a screenshot of the hit list browser and the supplier

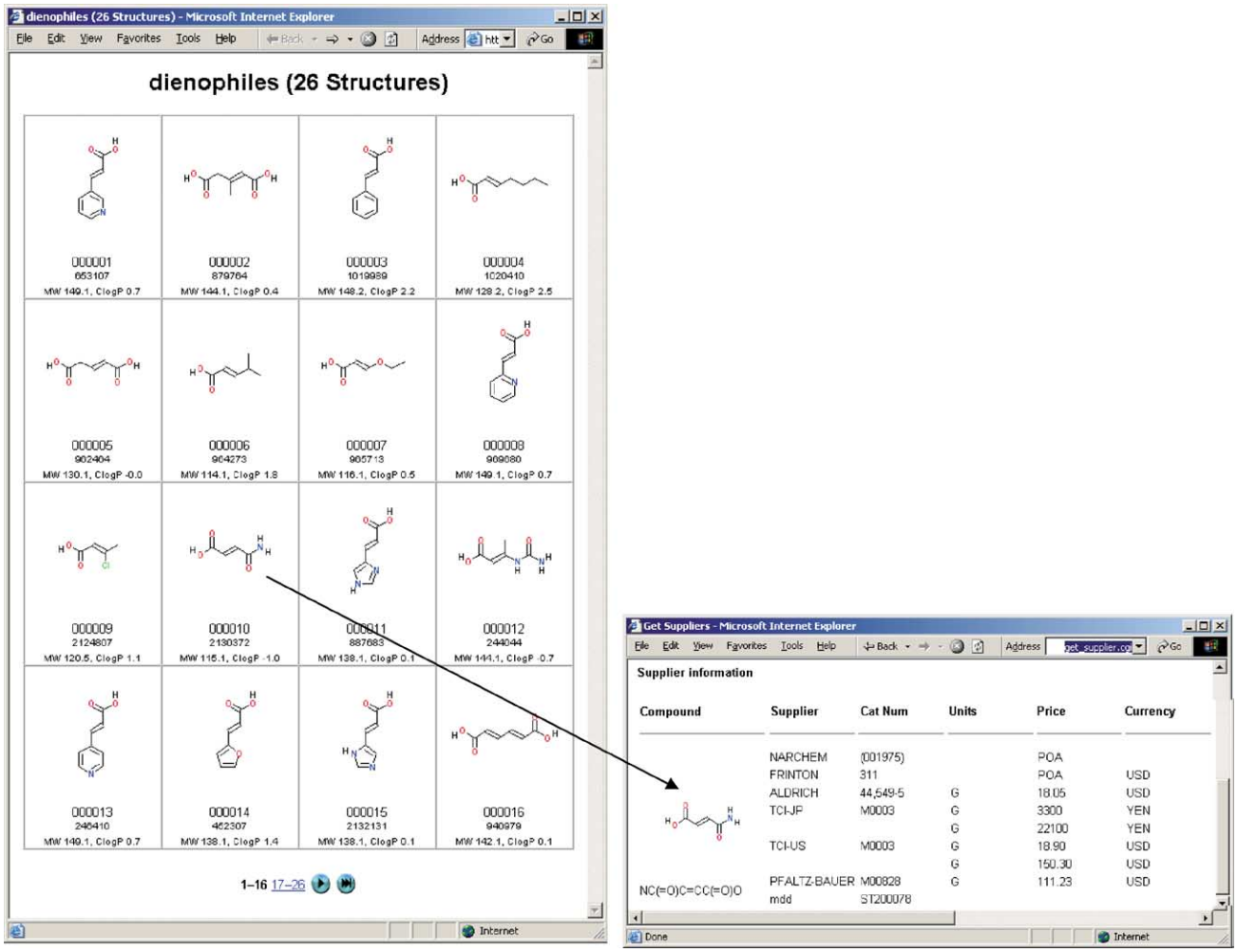

Fig. 7. Hit list browser. In this example the substructure searched for was indicated in Fig. 5. By clicking on one of the structures the supplier information for the compound can be accessed.

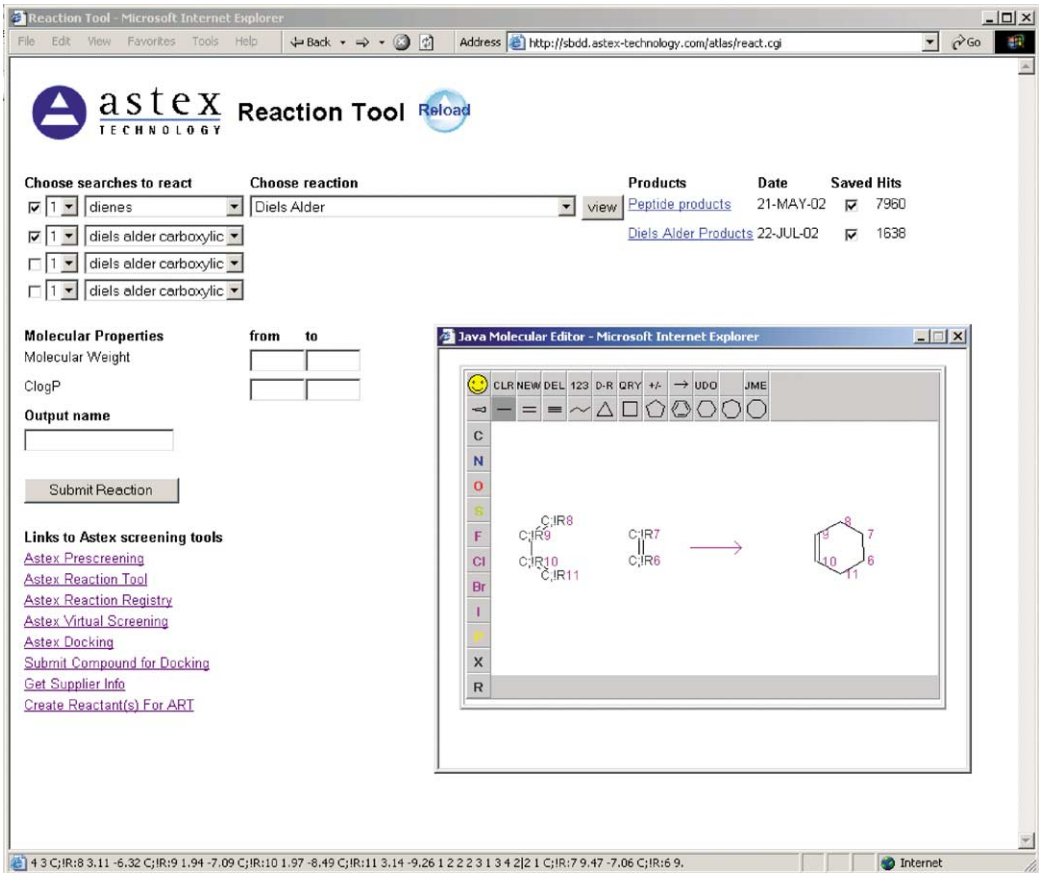

Fig. 8. VL enumeration interface screenshot. In this example a simple Diels–Alders reaction is carried out on two hit lists from a pre-screening search. The first of these is a set of dienes with a maximum molecular weight of 150. The second set is a set of molecules containing both a carboxylic acid and an ethane linker and a maximum molecular weight of 150.

information. The hit lists are browsed 16 compounds at a time and show some basic 1D chemical information. The pictures of the compounds are produced dynamically using CACTVS [19] and are then cached on the server.

3.2. Reaction interface

Fig. 8 shows a screenshot of the VL enumeration interface. On the top left are the pre-screen searches or reaction products from a previous reaction that can be used as reactants in the VL enumeration. The reactant tables can be selected by checking the boxes to the left of the drop down menus; the stoichiometry of the reactants can also be entered. To the right of this area are the reactions that are stored in ORACLE. If the required reaction is not stored a new one can be entered using the JME window. The example shown in Fig. 8 is a simple Diels–Alder reaction. Below this area are some simple molecular property dialogue boxes, where the output of the library enumeration can be limited to molecules satisfying the criteria entered.

Once the desired information has been entered, the VL is given a name and can then be enumerated. The enumerated products can be browsed in the same fashion as in the pre-screening. The supplier information for the reactants can be viewed by clicking on the pictures of the product molecules.

3.3. VS interface

VS jobs can be started, and the results viewed, via two separate interfaces. The VS start interface is shown in Fig. 9. The top left of the interface is the area where the user can choose the pre-screen hit list or the VL produced using the interfaces described earlier. The number of compounds in the selected set is shown below in red. Options are also available for choosing whether to charge the molecules using in-house charging rules (described earlier) and whether to store multiple binding modes as opposed to single solutions. The latter option is useful depending on the number of compounds being screened. If the number of compounds being screened is very large (i.e. 100,000) then usually only the top ranked binding mode is stored; however, if the number of compounds is smaller (e.g. a focused set of compounds) storing multiple binding modes may be useful. Below this area is the binding site information. Here the target of interest can be selected along with a specific structure, either

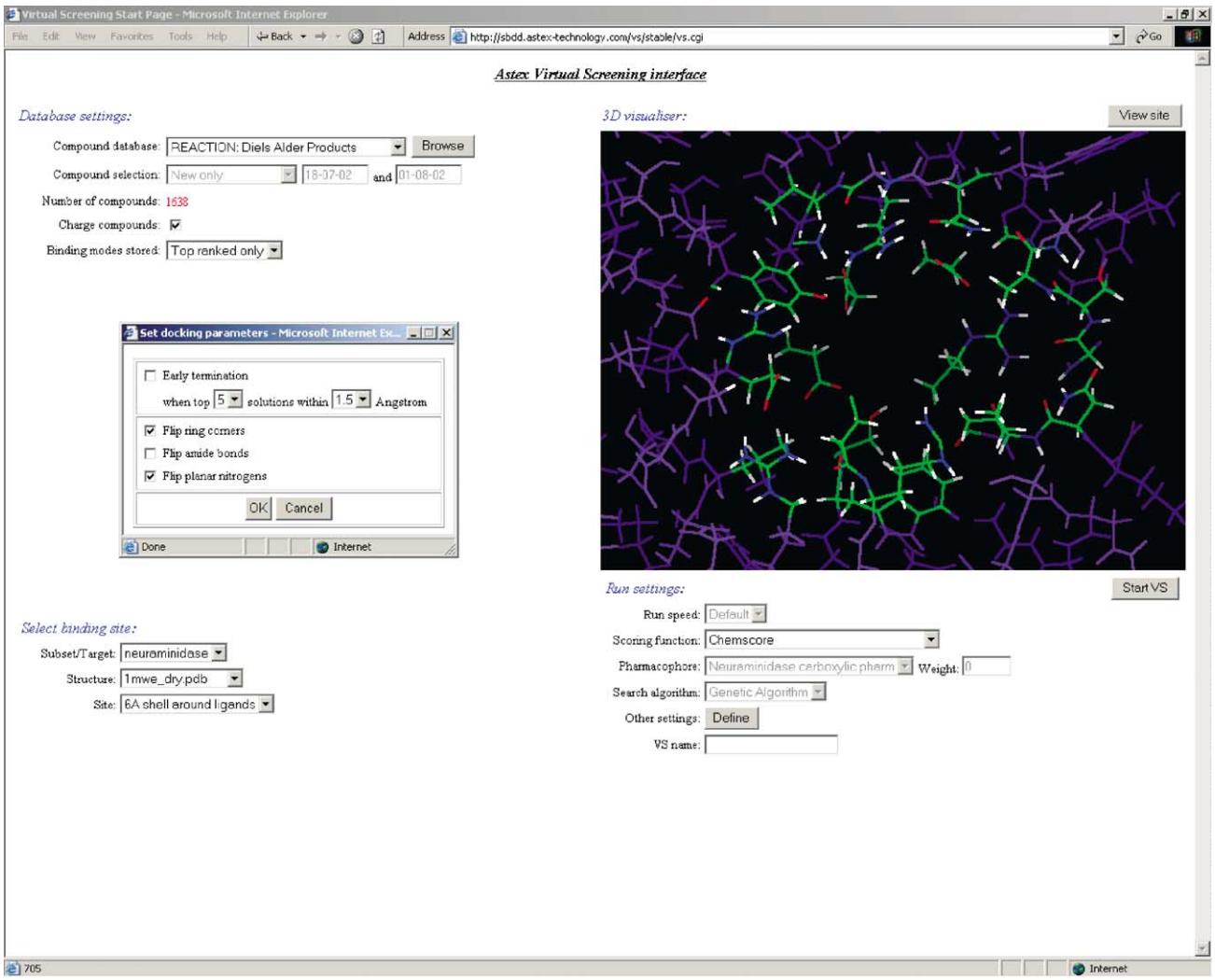

Fig. 9. VS start interface. In this example the VL produced using the reaction tool is to be docked against the active site of neuraminidase (PDB code 1MWE [29]).

a PDB file or an in-house protein structure, as well as the definition of the binding site, usually a pre-prepared cavity or a 4–6 Å shell around a known ligand. The bottom right area contains the run settings for the VS. Here the scoring function used for docking can be chosen (at present either Chemscore [2,26,27] or Goldscore [12–14]). Additionally, the scoring function can be combined with a pharmacophore score. In this case pharmacophores have been constructed from known ligands in their bound position in the parent protein. These pharmacophores are then stored in ORACLE. The ratio of the weights of the pharmacophore score to the docking score can be set to bias the objective function toward satisfying the pharmacophore or satisfying the docking score. Beneath this section the search algorithm can be chosen: at present either the GA [12–14] or the Tabu search [27]. The 'other settings' dialogue button activates a pop-up window (shown in Fig. 9) allowing activation of other docking settings such as ring-flipping and early termination criteria.

The final area of the VS start interface is the AstexViewer™ [28] window. This is a Java-based molecular viewer allowing protein–ligand complexes to be easily displayed in web pages. In Fig. 9 the active site of the neuraminidase protein from the 1MWE structure [29] is shown: the active site residues (defined using the earlier using the dialogue box) are coloured in green, the rest of the protein is coloured purple. In the example the products from the VL enumeration using the Diels–Alder reaction have been selected and can be docked against the active site of the protein selected. As with the pre-screening and VL enumeration the VS run is given a name and the screen can be submitted.

Once the docking run is complete the results can be viewed using the VS results interface. A snapshot of the interface is shown in Fig. 10. As can be seen, AstexViewer™ is used to view the results of the VS experiment. Below that, there are select boxes allowing other examples of the protein to be loaded (i.e. different PDB files) as well as their

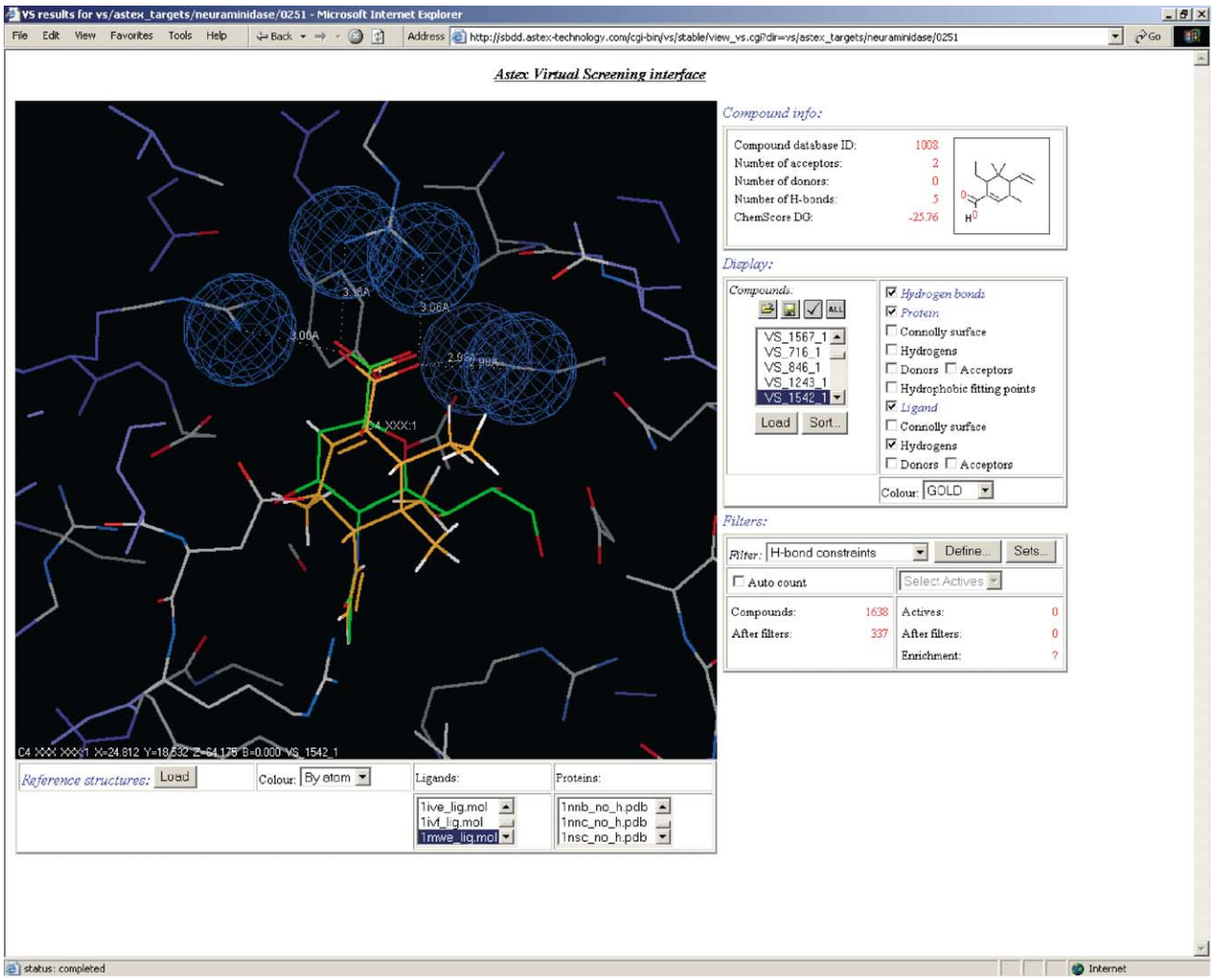

Fig. 10. VS results interface. In this example 1638 compounds from the VL produced from the Diels–Alder reaction have been docked against the active site of neuraminidase (PDB code 1MWE [29]). The native ligand is shown in green and one of the docked compounds from the VL is shown in orange. The arginine residues are highlighted because a filter has been applied insisting that the compounds loaded into the interface are forming hydrogen bonds with the selected atoms in these residues.

associated ligands. The top right of the diagram shows compound information such as the molecular properties of the compound (number of acceptors, donors, rotatable bonds, etc.), various docking scores (dependent on the scoring function used), as well as the number of interactions formed with the active site (number of hydrogen bonds, etc.). The information displayed here is configurable by the user: Table 1 shows the full list of information that can be viewed here. Below the compound information area are the display tools. This is where compounds can be loaded into the interface and viewed interactively. The compounds can be sorted using the criteria shown in Table 1 and hit lists of favoured compounds can be saved in ORACLE for future use. Additionally, there are a number of checkboxes for molecular display including Connolly surfaces, molecular colouring and the display of hydrogen bonds. Below this area is the filters area. This enables the docked compounds to be filtered on

Table 1
VS data types for sorting/filtering compounds

VS sorting/filtering/display data types
Compound ID
1D chemical information
Docking rank
Docking score
Components of docking score
Cluster size
Error flag
Number of hydrogen bonds
Hydrogen bonding motif
Include/exclude substructure

The entries shown in italics denote filtering only. In all the other cases it is possible to rank and/or filter the compounds within a VS results set based on user-defined criteria.

any number of the characteristics defined in Table 1. This feature has been employed in the example shown in Fig. 10. Here a hydrogen bond filter has been applied, insisting that the compounds loaded into the interface all form hydrogen bonds with highlighted atoms from the arginine residues in the active site. This greatly reduces the number of compounds loaded into the interface (337 from 1638). These can then be ranked in terms of any of the appropriate information contained in Table 1. This combination of filtering and sorting using a number of criteria allows the rapid identification of 'interesting' compounds.

After desirable compounds have been selected they can be saved in ORACLE in a VS hit list. These hit lists, as well as databases from other sources (e.g. pre-screening hit list, VLs, etc.), can be merged using Boolean logic. This is particularly useful to create a VS hit list of compounds exhibiting different desirable hydrogen bonding motifs. For example, a VS hit list could be produced by filtering on docking score and then a particular hydrogen bonding motif. A second VS hit list could be produced by filtering on docking score and a different hydrogen bonding motif. These two hit lists can then be merged together using Boolean logic to produce the final hit list.

4. Conclusions

In this paper, we have presented a web-based platform for VS. The overriding aims of such a system were to simplify the process of running VS experiments on large numbers of compounds. Using a client-sever model and Java applets for editing, viewing and manipulating structures removes the need for any client side software. This is key to the approach we have adopted as any reliance on specialist software and/or computers is removed and enables access to the results of VS to all relevant scientists, not only molecular modellers.

Using ORACLE to store large databases of compounds (either historical or those produced from VL enumeration) provides a highly scalable and efficient means of producing compound subsets for VS. Extending the usual capabilities of ORACLE with data cartridges means that all the substructure searching and VL enumeration can be carried out using SQL statements. This allows simple web-based interfaces to be easily implemented on top of this layer.

Storing the VS data in ORACLE is particularly advantageous because the data can be queried in an ad hoc fashion and allows very flexible manipulation of libraries of compounds (and their associated VS data) that have been docked against a target.

Acknowledgements

The authors would like to thank Dr. Alex Padova and Maria Carr for their assistance in preparing the ATLAS database and Dr. Ryan Smith for managing the ORACLE installation. We would also like to thank Dr. Peter Ertl and Novartis Pharma AV for providing JME. Lastly we would like to thank the referees for their constructive comments.

References

[1] W.P. Walters, M.T. Stahl, M.A. Murcko, Virtual screening—an overview, Drug Discov. Today 3 (1998) 160–178.
[2] C.A. Baxter, C.W. Murray, B. Waszkowycz, J. Li, R.A. Sykes, R.G.A. Bone, T.D.J. Perkins, W. Wylie, New approach to molecular docking and its application to virtual screening of chemical databases, J. Chem. Inf. Comput. Sci. 40 (2000) 254–262.
[3] M. Stahl, M. Rarey, Detailed analysis of scoring functions for virtual screening, J. Med. Chem. 44 (2001) 1035–1042.
[4] C. Bissantz, G. Folkers, D. Rognan, Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations, J. Med. Chem. 43 (2000) 4759–4767.
[5] E. Perola, K. Xu, T.M. Kollmeyer, S.H. Kaufmann, F.G. Prendergast, Y.P. Pang, Successful virtual screening of a chemical database for farnesyltransferase inhibitor leads, J. Med. Chem. 43 (2000) 401–408.
[6] B. Waszkowycz, T.D.J. Perkins, R.A. Sykes, J. Li, Large-scale virtual screening for discovering leads in the postgenomic era, IBM Sys. J. 40 (2001) 360–376.
[7] Available Chemicals Directory, MDL Information Systems Inc. (MDL), San Leandro, CA, USA.
[8] C.A. Lipinski, F. Lombardo, B.W. Dominy, P.J. Feeny, Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings, Adv. Drug Delivery Rev. 23 (1997) 3–25.
[9] R.L. Leach, J. Bradshaw, D.V.S. Green, M.H. Hann, Implementation of a system for reagent selection and library enumeration, profiling, and design, J. Chem. Inf. Comput. Sci. 39 (1999) 1161–1172.
[10] A. Gobbi, D. Poppinger, B. Rohde, Developing an inhouse system to support combinatorial chemistry, Perspect. Drug Discov. Des. 7–8 (1997) 131–158.
[11] P. Walters, Duct tape and superglue—creating a kinder, gentler software environment, Daylight User Group Meeting (MUG99), 1999, http://www.daylight.com/meetings/mug99/walters/index.html.
[12] G. Jones, P. Willett, R.C. Glen, Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation, J. Mol. Biol. 245 (1995) 43–53.
[13] G. Jones, P. Willett, R. C, A.R. Leach, R. Taylor, Development and validation of a genetic algorithm for flexible docking, J. Mol. Biol. 267 (1997) 727–748.
[14] M.L. Verdonk, J.C. Cole, M.J. Hartshorn, C.W. Murray, R.D. Taylor, Improved protein–ligand docking using GOLD, Proteins Struct. Funct. Genet., 2003, in press.
[15] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1998) 31–36.
[16] D. Weininger, A. Weininger, J.L. Weininger, SMILES. 2. Algorithm for generation of unique SMILES notation, J. Chem. Inf. Comput. Sci. 29 (1989) 97–101.
[17] P. Ertl (Ed.), Java Molecular, Novartis Pharma AV, Basel, Switzerland, http://www.molinspiration.com/jme/index.html.
[18] DayCart, Daylight Chemical Information Systems, Mission Viejo, CA, USA.
[19] W.D. Ihlenfeldt, Y. Takahashi, S. Abe, S. Sasaki, Computation and management of chemical properties in CACTVS: an extensible networked approach towards modularity and flexibility, J. Chem. Inf. Comput. Sci. 36 (1994) 109–116.
[20] Daylight Reaction Toolkit, Daylight Chemical Information Systems, Mission Viejo, CA, USA.

[21] J. Gasteiger, C. Rudolph, J. Sadowski, Automatic generation of 3D-atomic coordinates for organic molecules, Tetrahedron Comput. Methodol. 3 (1990) 537–547.
[22] GOLD, Cambridge Crystallographic Data Centre, 12 Union Road, Cambridge CB2 1EZ, UK.
[23] J. Wang, J. Delany, http://www.daylight.com/support/cont-rib/mol2smi.
[24] J. Bradshaw, Program objects, Daylight User Group Meeting (EuroMUG02), 2002, http://www.daylight.com/meetings/emug-02/bradshaw/progobs/index.html.
[25] L. Wall, T. Christiansen, R.I. Schwartz, Programming Perl, O'Reilly, Sebastapol, CA, 1996, pp. 455–457.
[26] M.D. Eldridge, C.W. Murray, T.R. Auton, G.V. Paolini, R.P. Mee, Empirical scoring functions. I. The development of a fast empirical scoring function to estimate the biding affinity of ligands in receptor complexes, J. Comput. Aid. Mol. Des. 11 (1997) 425–446.
[27] C.A. Baxter, C.A. Murray, D.E. Clark, D.R. Westhead, M.D. Eldridge, Flexible docking using the Tabu search and an empirical estimate of binding affinity, Proteins Struct. Func. Genet. 33 (1998) 367–382.
[28] M.J. Hartshorn, AstexViewer™: an aid for structure-based drug design, J. Comput. Aid. Mol. Des., 2003, in press, http://www.astex-technology.com/astexviewer.
[29] J.N. Varghese, P.M. Colman, A. van Donkelaar, T.J. Blick, A. Sahasrabudhe, J.L. Mckimm-Breschkin, Structural evidence for a second sialic acid binding site in avian influenza virus neuraminidases, Proc. Nat. Acad. Sci. U.S.A. 94 (1997) 11808–11812.