J|A|C|S

COMPUTER SOFTWARE
REVIEWS

Pipeline Pilot 2.1. By Scitegic, 9665 Chesapeake Drive, Suite 401, San Diego, CA 92123-1365. www.scitegic.com. See Web Site for Pricing Information

Pipeline Pilot is a multifaceted software platform that can be viewed as a tool for enhancing productivity for automatic computational workflow, as a comprehensive cheminformatics package, and as a component of an integration platform. A convenient and intuitive user interface is provided for designing and executing automated computational protocols. These protocols are assembled using modules that are represented as icons by the graphical user interface. The modules, called components, include a variety of data readers, manipulators, calculators, data viewers, and writers. For example, there are convenient data reading modules for ISIS db files, SD-files, and SMILES, as well as delimited text and Excel spreadsheets. Data viewers and writers include increasingly standard applications, such as WebLabViewerPro and Spotfire, as well as Excel. An HTML molecular table viewer provides another convenient way to view tabular results with structures.

Pipeline Pilot is built on a three-tier client-server architecture. All protocols execute on a Microsoft server, and all database and file access is through the server. Communication between clients and the server is via Microsoft's DCOM. Scitegic recommends that the Pipeline Pilot server should have between two and four Intel Pentium 750 MHz CPUs. Their rule of thumb is one processor for every two or three simultaneous users. The remaining hardware recommendations for the server include 1 or 2 GB of RAM, 150 MB of disk space for installation, and 10 GB of disk space for the runtime environment. On the software side, both Windows NT 4.0 server and Windows 2000 are supported. For a Windows NT-based server, the installation should also include Microsoft Transaction Server (MTS) 2.0, Microsoft Management Console 1.0, Internet Information Server (IIS) 4.0, Windows NT Service Pack 6a, and Microsoft Data Access Components (MDAC) version 2.5 or later. For a Windows 2000-based server, the installation should also include Windows 2000 Service Pack 2.0 and Internet Information Server (IIS) 5.0. Regardless of the operating system, the server installation should also include Internet Explorer 5.0. Likewise, Oracle Client libraries should be installed if Oracle databases are to be accessed. Client PCs should be equipped with a minimum of a 500 MHz Pentium with 256 MB of RAM and 100 MB of disk space. Operating systems supported on the client include Windows NT 4.0 with service pack 6.a, Windows 2000 with service pack 2.0, and Windows XP. Client installations should also include Internet Explorer 5.0.

An advantage of the pipelining approach is the ability to capture and share protocols conveniently for reuse. The protocols are actually stored in an XML format and can be easily exchanged. Each component has an editable text description below it. Also, when a component is opened, text is exposed that gives a description and a detailed explanation of usage. There is also a comment box opened for each protocol. The GUI has a convenient explorer on the left-hand side for navigating through protocols and modules. Extensive help is provided through a Web browser; it contains a table of contents and is searchable and highly useful. A 250-page manual is also available.

Although the applicability of the pipelining provided by this software is generic, the numerous (>200) specific components provided by SciTegic are heavily geared toward a cheminformatics environment. Several classes of calculators for molecular property can be dropped into the computational pipeline (AlogP, molecular polar surface area, H-bond acceptors, etc.). In addition, there are Extended Connectivity Fingerprints that are based on atom-centered graphs. These properties and fingerprints are combinable in components for executing mathematical and logical computations to produce a variety of calculators and filters. A simple scripting language called PilotScript is used to code these operations. The code is exposed in dialogue boxes so that code from existing modules can be easily edited even by users who do not have much of an understanding of scripting. Parameters for components in a protocol can also be easily exposed for modification. A particular advantage of Pipeline Pilot is the speed of execution of protocols for very large data sets. For example, one million compounds as SMILES could be Lipinski-filtered and outputted to a file in less than 3 min.

PipelinePilot has a special set of learning modules for creating filters based on "Bayesian Categorization". The learning protocols take two sets of data that can be considered "good" and "bad" or "target-like" and "target-unlike". The protocol ranks the features according to the probability of occurrence in the two sets. These features can be molecular fingerprints and/or binned property values. Once these learning modules are created, they can be applied as a filter in other protocols. For example, "targetX-like" molecules can be filtered from a corporate or vendor database on the basis of knowledge of several examples of ligands for targetX. Some advantages of this approach are speed and the fact that multiple classes can be learned and given weight according to their occurrence. In addition, the learned fingerprint features can be displayed as spheres on the 3-D structures of the molecules in WebLabViewerPro. In this way, one can see which atoms contribute most to the learning by determining why the molecule is "targetX-like".

In addition to its built-in functionality, the architecture of Pipeline Pilot has been organized for integration and extensibility, and the program is designed to interoperate with external software objects and applications. This is accomplished through a number of mechanisms, the simplest of which is via the built-in "RunProgram" component. This component can execute an external program on either the client or the server. The "RunProgram" component can run in two modes, blocking or nonblocking. In blocking mode, Pipeline Pilot waits until the external application finishes before continuing its own processing. In nonblocking mode, Pipeline Pilot continues processing

10.1021/ja025304v CCC: $25.00 © 2003 American Chemical Society

immediately after scheduling the external program. Pipeline Pilot also supports tighter integration with external computing services. This is accomplished utilizing two Pipeline Pilot components, “SOAP Method” and “SOAP Method (Batched)”. These two components manage the communication issues involved with the transfer of parameters and results between Pipeline Pilot and the external system. This mechanism is useful for integrating proprietary filters and property calculators into Pipeline Pilot protocols. A third mechanism for integrating Pipeline Pilot with external applications is through a Web Services interface, through which Pipeline Pilot protocols can be accessed and executed. This mechanism allows a developer to provide the functionality of Pipeline Pilot protocols from within applications such as Excel, Spotfire, or custom software. Pipeline Pilot also provides components that support scripting with languages such as VBScript, Perl, and Python. Components built with these scripting languages have access to all of the data flowing through a protocol, while providing all of the functionality available to the particular scripting language.

For the nonexpert user, the Web Services allows protocols to be accessed through a Web browser. The user of the Web Services tool does not see the pipelines, only a list of protocols. When the protocols are executed, dialogue boxes expose modifiable parameters and requests for file locations as required. When the protocols are complete, a link is generated to launch the output. For example, medicinal chemists could have parameters calculated for a set of compounds in a file and the output returned for viewing. However, to fully leverage these capabilities in a large organization, there needs to be some modification of the architecture regarding security. One of the limitations of Pipeline Pilot has been file access. All protocols executing on the server run under a single user account. This creates problems in accessing files on client PCs in the world of Microsoft Windows. The simplest solution is to maintain relevant data files on the server. A second solution is to grant share privileges to the server for specific client folders. This can be problematic.

In summary, PipelinePilot can simplify several informatics-intensive tasks related to libraries of compounds, including purchase of compounds, design of combinatorial libraries, and analysis of data for lead discovery and optimization. It can be a powerful addition for increasing productivity in an established cheminformatics environment or as an enabling technology for small organizations that cannot dedicate the manpower to the development and specialization normally required to be effective in these tasks.

James M. Stevenson and Peter D. Mulready,
Boehringer Ingelheim Pharmaceuticals, Inc.

JA025304V