Ордена Ленина

ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ

им. М.В.Келдыша

Российской Академии Наук

В.Н.Емельянов, И. В. Плетнев, А.Л.Чугреев

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА НА ОСНОВЕ ТЕХНОЛОГИЙ INTERNET ДЛЯ ХИМИИ И СМЕЖНЫХ ОБЛАСТЕЙ

Москва

2003г.

УДК 519.68

В.Н.Емельянов, И. В. Плетнев, А.Л.Чугреев.

ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА НА ОСНОВЕ ТЕХНОЛОГИЙ INTERNET ДЛЯ ХИМИИ И СМЕЖНЫХ ОБЛАСТЕЙ

В работе рассматривается информационно-вычислительная систем NetLaboratory, предназначенная для проведения расчетов и ведения баз данных (предметные области – химия и смежные дисциплины). Описываются основные модули системы; интерфейс пользователя, предназначенный для облегчения подготовки и исполнения прикладных программ в среде Интернет; язык манипулирования молекулярными объектами COSMOS.

Ключевые слова: интегрированная информационно-вычислительная система, вычислительный эксперимент, базы данных, физико-химические и биологические свойства

V. N. Emelyanov, I. V. Pletnev, A. L. Tchougreeff. Internet based information computational system for for chemistry and related fields. The preprint of the Keldysh Institute of Applied Mathematics, Russian Academy of Sciences.

Computing and Information system NetLab oriented to chemistry and related fields is discussed. The consideration concerns principles of system design; description of base modules; user interface for submitting and performinf computing via Internet; the langiage of molecular object manipulation COSMOS.

Key words: integrated computing and information system; computational experiment; databases; physico-chemical and biological properties.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты 98-07-90155 и 01-07-90383) .

СОДЕРЖАНИЕ

1. Введение 4

2. Общее описание ИВС 6

3. Включенные в систему пакеты программ 7

4. Пользовательский интерфейс 9

5. Управление заданиями, мониторинг и аудитпользователей, заданий и ресурсов 14

6. Язык манипулирования молекулярными объектами 16

7. Литература……...31

1. Введение

Роль вычислительного эксперимента при проведении исследований в химии, молекулярной физике, биохимии, биофизике, материаловедении в последнее время непрерывно возрастает. В основе этой общемировой тенденции лежит резкий рост стоимости химических исследований, проводимых традиционными методами. Вычислительный эксперимент служит важным дополнением и альтернативой обычному, например, химическому эксперименту. Существуют области, где только вычислительный эксперимент может предоставить необходимую информацию.

Приведем один яркий пример. Современный подход к созданию новых лекарственных препаратов подразумевает скрининг (“просеивания”) миллионов молекул-кандидатов – в реальности еще не синтезированных, но доступных для компьютерного манипулирования благодаря т.н. виртуальным комбинаторным библиотекам – через “сито” расчсчитанных физико-химических и биологических свойств; лишь немногие потенциально наилучшие молекулы (обычно несколько десятков-сотен) впоследствии исследуются длительными и трудоемкими экспериментальными методами.

Таким образом, остро стоит задача проведения массированных расчетов электронной структуры, геометрии, разнообразных физических, химических и биологических свойств молекул. Для предоставления соответствующих услуг сообществу исследователей и практиков необходима стабильно работающая, развитая система, предоставляющая, что особенно важно - унифицированный доступ к важнейшим расчетным пакетам и базам данных.

Взрывной рост сети ИНТЕРНЕТ и массированные инвестиции (в России - прежде всего, по линии РФФИ) в ее развитие создают новые возможности и новые проблемы. Возможности связаны с появлением уникальной

глобальной среды распределенных вычислений и хранения информации, проблемы - с тем, что сеть "наполняется содержанием" существенно медленнее, чем развивается коммуникационная инфраструктура. Так, теоретически обоснована и получает аппаратную базу "сетецентрическая" модель вычислений, представляющая логическое развитие клиент-серверного подхода. Однако если развитие клиентской стороны уже привело к появлению сначала концепции, а недавно - и промышленных образцов "сетевых компьютеров" (NC, NetPC), развитие серверной, "содержательной" стороны практически во всех предметных областях заметно отстает.

Пожалуй, лишь в молекулярной биологии и генетике уже имеется ряд примеров успешно работающих систем такого рода, в том числе – коммерческих Веб-порталов. Так, расшифровка генома человека и продолжающиеся исследования протеомы (набора белков живых организмов) были бы невозможны без массированного использования подобных систем, “прозрачно” для пользователя интегрирующих десятки программ и баз данных, функционирующих на множестве серверов во всем мире (например, банки данных о генах, молекулярной экспрессии, трехмерной структуре белков и методы многомерного статистического анализа/визуализации многомерных данных, расчеты типов свертывания белков и т.п.).

В химии же и смежных областях практически отсутствуют специализированные центры, на регулярной основе предоставляющие научному сообществу вычислительные и информационные услуги; отсутствует программное обеспечение, интегрирующее разнообразные вычислительные средства и базы данных в рамках единого интерфейса и унифицированного представления (молекулярных) объектов и организующее работу пользователя в режиме удаленного доступа на основе технологий ИНТЕРНЕТ.

Задача создания такого программного обеспечения - не техническая, а фундаментальная. В химии и смежных областях известны сотни прикладных

программ и баз данных. Одних форматов представления молекулярных структур - не менее 30-40. Наличие большого числа разнородных программных продуктов создает значительные проблемы как при организации их взаимодействия, так и для работы пользователей. По этой причине представляется необходимым создание интегрированной среды (информационно-вычислительной системы), которая позволяла бы исследователю использовать единый пользовательский интерфейс при обращении к любым прикладным программам и базам данных, подготовке входных данных для программ, визуализации, анализе и архивировании результатов расчетов. Такая система неминуемо должна иметь свое собственное представление данных (молекулярных и родственных объектов) и средства манипулирования ими; только при этом возможна организация "конвейера", обеспечивающего анализ конкретных задач всеми доступными методами, реализованными в базах данных и пакетах прикладных программ.

В течение последних лет такая ИВС, получившая название NetLaboratory (“сетевая лаборатория”) разрабатывалась коллективом сотрудников ИПМ РАН им. М.В. Келдыша, НИФХИ им. Л. Я. Карпова и Химического факультета МГУ им. М. В. Ломоносова. В настоящей работе представлены дизайн и основные характеристики системы.

2. Общее описание ИВС

Система обеспечивает работу пользователей *на основе технологий Интернет*, предоставляя им доступ к ряду пакетов программ. Используется *технология "клиент-сервер"*, которая по праву считается одной из базовых технологий для современных компьютерных сетей, в ее современном варианте - технологии "всемирной паутины", WWW. Используя *Web-браузер*, пользователь осуществляет выбор необходимого ему вычислительного

ресурса, передает ИВС задания на проведение расчетов и получает их результаты. Пользователи имеют возможность получать справочную информацию о программных пакетах, адаптированных в системе.

Система включает центральный (управляющий) сервер и ряд вычислительных серверов. Пользователь взаимодействует с центральным сервером через Веб-интерфейс (см. ниже), вычислительные сервера ему напрямую не доступны. Такое проектное решение обеспечивает легкую расширяемость системы и повышает безопасность.

Вычислительные сервера физически удалены, однако доступны для связи с управляющим сервером по сети Интернет. Связь осуществляется по протоколу TCP/IP (транспортный и сетевой уровни модели OSI), собственно управление потоком заданий – по протоколам ssh и rsh (уровень приложения в модели OSI). требования к их аппаратно/программной конфигурации.

В разное время в состав системы входило от двух до пяти вычислительных серверов, физически расположенных в разных точках г. Москвы (в частности, суперкомпьютер Convex 1000, рабочая станция HP 710, высокопроизводительные персональные компьютеры). Отдельные программы установлены на разных машинах (иногда – с дублированием).

3. Включенные в систему пакеты программ

Существенным этапом работы стала адаптация пакетов расчетных программ на имеющихся вычислительных средствах. Заметим, что развитие вычислительной химии и смежных областей постоянно приводит к появлению новых версий существующих программ и баз данных, что требовало и требует соответствующего обновления ИВС.

Установленные и доступные программные пакеты включают:

- **GAMESS** - ab-initio расчеты электронной структуры и поверхностей потенциальной энергии молекулярных систем
- **MOPAC** - полуэмпирические расчеты органических молекул и их реакций ;
- **ECF** - полуэмпирические расчеты электронного строения и спектров комплексов переходных металлов.
- **TINKER** – расчеты структуры и свойств молекулярных систем методом молекулярной механики и молекулярной динамики.
- **MMPC** – расчеты структуры и свойств молекулярных систем методом молекулярной механики; программа общего назначения, особо ориентированная на координационные соединения.
- **MOLCRYST** - расчеты структуры, теплоты сублимации и фононных спектров молекулярных кристаллов методом атом-атомных потенциалов.
- **BF** - полуэмпирические расчеты электронного строения органических молекул оригинальным методом линейным по размеру системы, обеспечивающим учет электронных корреляций.
- **DatEx** - пакет многомерного анализа данных.

Данные пакеты предоставляются на некоммерческой основе, но авторские права на них защищены в соответствии с внутренним законодательством и международными обязательствами России. Они обеспечивают базовый набор возможностей вычислительной химии для практического использования при решении широкого круга задач расчета физических, химических и биологических свойств веществ и материалов.

Пакеты **ECF** [1]**, MOLCRYST, BF** [2] и **MMPC** [3]**, DatEx** представляют собой оригинальные разработки авторов (соответственно, А. Л. Чугреев с сотр. и И. В. Плетнев с сотр.).

Кроме того, адаптирован и включен в систему новый программный пакет **ECFMM** [4] (А. Л. Чугреев, И. В. Плетнев и сотр.). Он дополняет имеющиеся в составе ИВС программы квантовой химии и молекулярной механики, реализуя гибридный метод расчетов больших молекул, объединяющий - на основе соответствующей теории - подходы квантовой и молекулярной механики. Тем самым делаются доступными для строгого анализа ранее недостижимые задачи (например, биологически важные металлсодержащие молекулы большого размера). Этот программный пакет, как и другие, доступен через Web-интерфейс широкому кругу специалистов.

Для всех адаптированных пакетов прикладных программ пользователям системы доступны краткие инструкции по использованию с образцами входных и соответствующих выходных файлов.

4. Пользовательский интерфейс

Доступ к вычислительным и информационным ресурсам предоставляется на основе технологий Интернет. Используется технология "клиент-сервер" в ее современном варианте – технологии "всемирной паутины" или **WWW**. Web-технология идеально подходит для работы с гипертекстовыми документами, большими информационными архивами и базами данных. Программное обеспечение для хранения данных и вычислений располагается на серверах системы, интерфейс с пользователем - на стороне клиента, а обработка данных распределяется между клиентской и серверной частями.

Используя Web-браузер, обеспечивающий удобный и хорошо знакомый интерфейс для доступа к информации, пользователь подает на вход информационно-вычислительной системы задания на проведение расчетов и

получает результаты расчетов (в виде соответствующих входных/выходных файлов, которые загружаются с клиентской машины или на нее).

В качестве программного средства для разработки пользовательского интерфейса принят язык встроенных сценариев **PHP3** [5]. Данное средство предоставляется на некоммерческой основе, и обеспечивает возможность обращения к различным СУБД, включая MySQL. В качестве примера, демонстрирующего применение языка **PHP3** для организации доступа пользователей к информационно-вычислительной системе включая реализацию доступа к СУБД MySQL приведена головная Web страница с включенным в нее сценарием проверки прав доступа к ресурсам системы.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2
Final//EN">
<HTML>
 <HEAD>
  <TITLE></TITLE>
 </HEAD>
```

```
<!-- Background white, links blue (unvisited), navy
(visited), red (active) -->
 <BODY
  BGCOLOR="#FFFFFF"
  TEXT="#000000"
```

```
  LINK="#0000FF"
  VLINK="#000080"
  ALINK="#FF0000"
 >

<p align="center"><font size="5">Net Laboratory of
Computational Chemistry</font></p>
  <H1 ALIGN="CENTER"></H1>
  <P>Dear visitor, to proceed further you must be a
registered user. If you have not registered yet do it
<A
HREF="http:
//qcc.ioc.ac.ru/~netlab/u_interface/preregistration
.htm">now</A> and let
a couple of days to process your registration and
receive your secret
password.
```

```
  <HR WIDTH="50%" SIZE="8">

<?php
```

```
    cfunction authenticate() {


    echo "<html>";

    echo "<body>";

    echo "<p align=center>";

    echo "<h3> Enter your Login Name and Your Password
</h3>";

    echo                                    "<form
action=\"../cgi-bin/u_interface/authenticate_new.ph
tml\" method=\"POST\">";

    echo "<table border = \"0\">";

    echo  "<TR><TD>Login:</TD><TD><input  type=text
name=uid></TD></TR>";

    echo          "<TR><TD>Password:</TD><TD><input
type=password name=passwd></TD></TR>";

    echo "</table>";

    echo    "<p><input    type=submit    name=Submit
value=Send>";

        echo    "<input    type=reset    name=Reset
value=Clear></p>";

    echo "</form>";

    echo "</body>";

    echo "</html>";

        exit;
```

```
    }
    authenticate();
    echo "$uid";
    $authentication_failed=1;
    while($authentication_failed) {
        authenticate();
    } else {
        mysql_connect($host,$user,$pass)
            or die("Unable to connect to SQL server");
        mysql_select_db($dbase) or die("Unable to
select database");
        $id=strtolower($uid);
        $query = mysql_query("select * from
participant where name='$uid'".
                            " and password =
'$passwd'");
        if(!mysql_num_rows($query)) {
            authenticate();
        }
    }
?>
 </BODY>
```

```
</HTML>
```

Доступ к ресурсам системы обеспечен для зарегистрированных пользователей.

5. Управление заданиями, мониторинг и аудит пользователей, заданий и ресурсов

Подсистема управления (ПСУ) включает в себя собственную динамически модифицируемую БД, содержащую данные о задачах, пользователях и ресурсах; средства доступа к базам данных (для администраторов и пользователей отдельно) и программу, реализующую распределение и запуск заданий пользователей, а также выдачу результатов (последняя программа - “демон” в терминах ОС UNIX – выполняет дополнительные служебные функции, например, опрос вычислительных серверов системы с целью проверки их доступности).

Соответствующее программное обеспечение - пакет скриптов на языке PHP3 и программа на языке Python [6]. В качестве программного средства ведения БД используется СУБД общего назначения MySQL [7]. СУБД работает в среде операционных систем UNIX (Linux и др.) и Windows NT, причем на платформе UNIX предоставляется на некоммерческой основе. Она обеспечивает высокую надежность и универсальность, обеспечивает защиту данных от несанкционированного доступа. В настоящее время в мире используется более 100 тысяч копий данной СУБД; существуют и используются базы данных, поддерживающие более 50 млн. записей.

ИВС организует управление заданиями следующим образом. Пользователи ИВС пополняют очередь задач, модифицируя соответствующие таблицы управляющей БД и осуществляя загрузку входных данных на ЦС ИВС. При этом отслеживается соответствие между полномочиями конкретного пользователя и затребованными им ресурсами. ИВС периодически опрашивает очередь задач. При помощи основной программы на языке Python осуществляется передача файлов данных для каждой задачи с ЦС ИВС на вычислительные сервера (при этом используется команды rcp/scp), запуск скрипта прикладной программы с именами файлов данных в качестве параметров на вычислительном сервере (при этом используется команда rsh/ssh). Такой способ доступа (r/s-комманды) к вычислительным ресурсам (серверам или узлам многопроцессорных серверов) является общепринятым в известных системах организации сетевых очередей (DQS, NQS). Контроль выполнения прикладной программы производится по признаку появления в доступной ИВС директории на вычислительном сервере соответствующего файла-семафора. По окончании расчета ИВС осуществляет передачу файлов с результатами расчетов с вычислительных серверов на ЦС ИВС. Та же программа осуществляет модификацию таблиц управляющей БД, содержащих информацию о задачах, программах, пользователях.

6. Язык манипулирования молекулярными объектами

В рамках проекта создания ИВС **NetLab** был разработан и апробирован язык манипулирования молекулярными объектами **COSMOS** (И. В. Плетнев и сотр.). Дадим его краткое описание.

Короткое введение. Интегрированная информационно-вычислительная система требует специальных средств для *описания* и для *манипулирования* молекулярными объектами. Средства описания необходимы для единообразного представления молекулярных объектов и обеспечения межпрограммной коммуникации. Простейший пример: молекула “с точки зрения” программы для квантовохимического расчета и программы, реализующей метод молекулярной механики — разные объекты. Для квантовой механики атомы молекулы необходимо характеризовать *номером элемента* в Периодической системе, числом валентных электронов и координатами. Для молекулярной механики помимо координат атомов нужны их *типы*, отнюдь не определяемые однозначно положением элемента в Периодической системе. Они определяются с учетом валентного окружения (иногда не только ближайшего) атома; иными словами, необходима информация о *связности* молекулы. При анализе же, например, реакционной способности или влияния заместителей, в ряде же случаев стереохимические данные вообще не нужны, важна лишь связность, или структурная формула. Встречаются и ситуации, когда важен вообще только компонентный состав молекулы, брутто-формула (например, при поиске в некоторых базах данных по неорганической химии). Естественно, такое многообразие отражает многообразие моделей, принятых в химии (не говоря уже о смежных областях). Если представление молекулы ориентировано на работу только с одной программой или базой данных, проблем не возникает – используется соответствующий уникальный файловый формат. Однако в рамках интегрированной среды необходимо единое представление.

Существенно, что единообразное представление молекулярных объектов создает возможность *манипулирования* ими – вне пределов

конкретной расчетной программы или базы данных. Такое манипулирование кардинально меняет способы подготовки данных для расчетов, а иногда и технологию расчета вообще.

Один пример. Ряд программ молекулярной механики позволяет провести однократный расчет оптимальной конформации молекулы при заданной начальной геометрии. Во многих случаях, однако, требуется провести серию таких расчетов, последовательно изменяя начальную геометрию (например, изменяя два гликозидных торсионных угла в молекуле дисахарида). Очевидно, крайне заманчива возможность реализовать соответствующий сценарий вычислительного эксперимента при помощи короткой программы, манипулирующей геометрией молекулы и вызывающей внешнюю программу конформационного анализа.

Язык манипулирования молекулярными объектами должен включать:

- основу, обычного алгоритмического языка программирования (желательно с поддержкой объектов);
- структуры данных, соответствующие молекулярным объектам;
- функции, оперирующие этим объектами (методов).

Основой языка **COSMOS** послужил объектно-ориентированный язык программирования (язык сценариев) **Python** [6].

Это в высокой степени переносимый (портирован на большее число платформ, чем Java), легко расширяемый и надежный язык. Он полностью подерживает объектно-ориентированное программирование, что позволяет легко надстраивать над общей частью специфические конструкции, имеющие отношение к задачам вычислительной химии. При этом сохраняется богатая семантическая база, позволяющая оперировать не только с числами, но и с развитыми структурами данных; это, например, списки (lists), словари

(dictionaries), кортежи (tuples). **Python** - интерпретируемый язык, что резко облегчает и ускоряет создание сложных программ. Он широко используется, особенно в среде ОС Unix and Windows NT, для решения ряда задач системного администрирования, работы с Интернет (CGI), управления базами данных (имеется порт к MySQL).

Наш опыт показывает, что **Python** – удобное и простое средство программирования манипуляций со сложными объектами. Есть примеры применения **Python** к задачам численного анализа (NumPy, Numerical Python - Lawrence Livermore National Laboratory) и даже некоторых задач молекулярного моделирования [3].

Ниже приведено краткое описание молекулярных объектов языка **COSMOS**.

Очевидно, иерархия молекулярных объектов должна включать некоторые примитивы и надстроенные над ними более сложные конструкции. В качестве примитивов **COSMOS** включает следующие объекты:

Atom - основа любой химической конструкции;

Bond - объект, представляющий связь между парой (на данном этапе) атомов;

Point - объект, представляющий положение атома в пространстве.

Более сложные конструкции обычно представляют собой наборы примитивов. Технически соответствующие классы сходны со встроенными в **Python** списками (list), но должны допускать присутствие только уникальных объектов (запрещены дубликаты членов списка).

Это объекты:

Assembly - набор атомов (объектов **Atom**), являющийся базой для более сложных наборов;

Molecule - валентная схема, надстроенный над **Assembly** набор объектов **Bond**;

Conformation - конформация, надстроенный над **Assembly** набор объектов **Point**.

Кроме того, имеется объект **Structure**, представляющий собой комбинацию объектов **Molecule** и **Conformation**, надстроенных над одним объектом **Assembly**.

Данная иерархия уже позволяет описывать огромное разнообразие реальных химических объектов (“от метана до нуклеотидов”). Для манипулирования этими объктами предусмотрен развитый набор методов.

Ниже приведены, в качестве примера, только методы, предусмотренные для объекта Molecule:

```
Assembly() - возвращает базовый объект Assembly;

Molecules() - возвращает набор всех объектов типа Molecule, надстроенных над базовым объектом Assembly;
```

```
Conformations() - возвращает набор всех объектов типа Conformation, надстроенных над базовым объектом Assembly;

Structures()- возвращает набор всех объектов типа Structure, надстроенных над базовым объектом Assembly;

CopyTree(preserve_base_atoms=1) - вызывает соответствующий метод базового объекта Assembly;

GetCN(atom) - возвращает координационное число атома в молекуле;
```

Item(i) - возвращает i-ую связь молекулы (объект Bond) ;

Items() - возвращает копию списка всех связей молекулы;

Index(bond) - возвращает индекс (порядковый номер) связи в молекуле;

is_here(bond) - возвращает 1, если данная связь присутствует в данной молекуле;

Count() - возвращает число связей в молекуле;

Atoms() - возвращает список атомов базового объекта Assembly;

Pairs() - возвращает список уникальных пар связей;

AssocAtoms(atom) - возвращает список атомов, связанных с данным;

NonAssocAtoms(atom) - возвращает список атомов, образующих невалентные контакты с данным;

AssocBonds(atom) - возвращает список связей, примыкающих к данному атому;

Reachable(atom) - возвращает список атомов, достижимых (по химическим связям) из данного;

Div(bond) - разбивает молекулу по данной связи, возвращает кортеж из двух списков атомов, <левой< и <правой< половины;

IsConnected() - возвращает 1 для связной молекулы;

GetBond(atom1,atom2) - возвращает связь (объект Bond), соединяющую данные атомы или None;

Filter(atlist1,atlist2,bolist,is_ring=ANY) - возвращает список связей, удовлетворяющих заданному условию (например, для заданного порядка связи);

IsBonded(atom1,atom2) - возвращает 1 для связанных атомов;

Is13(atom1,atom2) - возвращает 1 для атомов, расположенных <через один> ;

IsRing(atom) - возвращает 1 для атомов внутри цикла;

MinChain(atom1,atom2) - возвращает минимальный путь между атомами;

IsRingBond(bond) - возвращает 1 для связей внутри цикла;

MinRing(atom) - возвращает минимальный цикл, в который входит атом;

VAngles() - возвращает список валентных углов (троек атомов) ;

GetBondTorsions(bond) - возвращает список торсионных углов (четверок атомов)молекулы;

Torsions() - возвращает список всех торсионных углов (четверок атомов) молекулы;

Remove(bond) - удаляет связь;

AtRemove(i) - удаляет связь по ее индексу;

RemoveList(list) - удаляет группу связей;

AddBond(atom1,atom2,bond_order) - создает связь и добавляет соответствующий объект в молекулу;

Add(bond) - добавляет существующую связь в молекулу;

AddList(blist) - добавляет группу связей в молекулу;

RemoveTerminal(tatom) - удаляет атом tatom и все связанные с ним концевые атомы (терминальную группу);

AtomID(atom) - вызывает соответствующий метод базового объекта Assembly;

UnlabelAtoms() - вызывает соответствующий метод базового объекта Assembly;

LabelAtoms() - вызывает соответствующий метод базового объекта Assembly;

UnlabelAtom(atom) - вызывает соответствующий метод базового объекта Assembly;

LabelAtom(atom) - вызывает соответствующий метод базового объекта Assembly;

Print(format=2) - печатает внутреннее представление данных объекта;

PrintStat() - печатает отладочное представление данных объекта.

В качестве примера ниже приведена короткая программа на языке COSMOS, решающая модельную задачу подготовки данных для расчета. Программа считывает данные о структуре молекулы (олигосахарида) из файла в формате MDL MOL; печатает значения всех длин связей O-C; печатает значения всех торсионных углов O-C вне циклов; печатает все валентные углы X-O-X и устанавливает их значение (для углов вне цикла) равным 113 град. Такая последовательность действий типична, например, при подготовке начального приближения для расчета методом молекулярной механики по рентгеноструктурным данным.

```
#!/usr/bin/python
""" SAMPLE COSMOS PROGRAM.
GET SACCHARIDE FROM MDL MOL FILE AND
PRINT ALL O-C RING BOND LENGTHS
PRINT ALL NON-RING O-C TORSIONS
PRINT ALL X-O-X BOND ANGLES AND
TRY TO SET EACH TO 113 DEG
(WILL BE OK FOR NON-RING ANGLES)
"""
from cosmos import *
def test():
    # get structure
    struc = ImportStructure('sacchar.mol')
    # print out ring O-C lengths
    print '\n* * * O-C RING BONDS * * *\n'
    bonds =
struc.Molecule().Filter(['O'],['C'],[],RING_YES)
    if bonds!=None:
        id = struc.AtomID
        bondval = struc.GetDist
        n, av = 0, 0.0
        # print out bonds
        for bond in bonds:
```

```
            a1, a2 = bond.Atoms()
            l = bondval(a1,a2)
            print '%-4s %-4s %-8.4f'% ( id(a1),id(a2),l
        )
            av = av + l
            n = n+1
        # print out statistics
        print '--------'
        print 'Av.', '%-8.4f'% (av/n), '(%-i bonds)'%n
    # print out non-ring O-C torsions
    print '\n* * * O-C NON-RING TORSIONS * * *\n'
    bonds =
struc.Molecule().Filter(['C'],['O'],[],RING_NO)
    for bond in bonds:
        torsions = struc.GetBondTorsions(bond)
        if torsions != None:
            for torsion in torsions:
                a1,a2,a3,a4 = torsion
                print '%-4s %-4s %-4s %-4s %-8.2f' % \
                ( id(a1),id(a2),id(a3),id(a4), \
                struc.GetTAngle(a1,a2,a3,a4) )
    # print out all C-O-C bond angles and try to set them
    print '\n* * * X-O-X BOND ANGLES * * *\n'
    angles = struc.VAngles()
```

```
    if angles!=None:
        for angle in angles:
            a1, a2, a3 = angle
            if a2.GetElSymbol() != 'O' : continue
            print '%-4s %-4s %-4s %-8.2f' % \
            ( id(a1),id(a2),id(a3), \
            struc.GetVAngle(a1,a2,a3)),
            # try to set to 113 degrees
            print 'TRY TO SET...',
            ok = struc.SetVAngle(a1,a2,a3,113)
            if not ok: print 'FAILED. ',
            else: print 'SUCCESS.',
            print 'NEW VALUE', '%-8.2f' %
            struc.GetVAngle(a1,a2,a3)

if __name__ == "__main__":
test()
```

Приведем более сложный, реальный пример.

Доступная через ИВС программа ММРС позволяет осуществлять достаточно сложные конформационные расчеты макромолекул - например, вычислять зависимости энергии от значений торсионных углов в олигосахаридах. Однако такие задачи требуют тщательно подготовленных исходных данных - начальные конформации отдельных моносахарилов должны быть достаточно близки к экспериментальным. Пользователь должен

провести значительную работу по поиску экспериментальных данных для каждого моносахарида и сборке их в макромолекулу. Аналогичные сложности возникают с обработкой результата расчетов.

На языке COSMOS (Python) составлена программа (листинг доступен по адресу http://analyt.chem.msu.ru/preconcentration/pletnev/cosmos/carbbuild.py), реализующая автоматическую сборку олигосахаридов из интегрированной базы структур моносахаридных остатков; пользователю достаточно ввести сокращенное название олигосахарида в соответствии с общепринятой химической номенклатурой IUPAC (например, Neu5Ac(b2-3)Gal(a1-4b)[Fuc(a1-3)]GlcNAc). Программа рассчитывает также конформационную карту (по гликозидным торсионным углам) для каждой пары моносахаридов олигосахарида и определяет стерически запрещенные/разрешенные области конформационного пространства.

Заметим, что в язык COSMOS включен базовый набор средств для работы с файлами химических баз данных (формат SDF). С его помощью можно получить доступ к численным/текстовым полям БД и содержащимся в БД молекулярным структурам. Поскольку формат пока SDF является фактическим стандартом для химических БД, это дает возможность широкой интеграции баз данных и расчетных программ. В дальнейшем предполагается дополнить библиотеку объектов/методов COSMOS средствами поддержки XML.

Приведем краткое описание.

SD файл представляется как коллекция (список, list, в терминах языка Python) записей с ассоциированным именем файла. Каждая запись представляется как словарь (dictionary, в терминах языка Python), т.е. последовательность пар {ключ:значение}. Один ключ предопределен, 'Structure_as_text'; ему соответствует значение record['Structure_as_text'], содержащее текстовое представление молекулярной структуры, как оно

записано в SD файле. Все другие ключи (и значения) извлекаются из присутствующих в SD файле полей.

Основные базовые операции над файлами в формате SDF таковы:

Open - открыть SD файл

Read - прочитать SD файл

Readnext - считать очередную запись

Exportcsv - записать содержимое заданных (по имени) полей в CSV-файл (CSV - comma-separated values, стандартный формат для файлов числовых данных)

Writecopy - записать копию SD файла

Write- записать SD файл в соответствии с заданной последовательностью и числом записей

Getstructureless - получить список записей, в которых отсутствует информация о молекулярной структуре

Getfieldduplicates - получить список записей, для которых в поле, идентифицировнном именем, содержится одна и та же информация

Delrecord - удалить запись

Get_cosmos_structure - преобразовать информацию о молекулярной структуре в объект языка COSMOS

Append_cosmos_structure - присоединить к записи информацию о молекулярной структуре, представленной объектом языка COSMOS

На основе этих базовых средств и объектов/методов языка COSMOS составлен пакет SDFTools. Он автоматизирует типичные рутинные операции с файлами химических данных; позволяет получить доступ к численным/текстовым полям БД и содержащимся в БД молекулярным структурам. Таким образом, расширены возможность межпрограммной интеграции и коммуникации.

Пакет SDFTools/COSMOS можно сравнить с разработанным в National Cancer Institute (NCI, США) пакетом сходного назначения SDF Toolkit, использующим язык Perl [8]. Последний позволяет, например, выбрать записи из SD файла по ключу (присутствию/отсутствию заданного поля); удалить записи при помощи заданного фильтра; удалить ненужную информацию; слить два файла с удалением дубликатов и т.п. Соответствующие функции (и ряд других) доступны и при использовании SDFTools/COSMOS. Кроме того, у последнего есть важное преимущество: интегрированность со средствами манипулирования молекулярными объектами. SDF Toolkit (NCI), по существу, предназначен - что естественно для программ на Perl - для разбора и манипулирования полями текстовых файлов специального вида, SDF. В его рамках нельзя - или очень нелегко - выполнять химически значимые операции над молекулярными структурами, содержащимися в файле (например, повернуть молекулу как целое; изменить заданные длины связей и т.п.). В среде же COSMOS возможность таких операций предоставляется автоматически.

В настоящее время пакет включает следующие утилиты:

sdfinfo.py – утилита предназначена для получения информации об содержимом SDF файла. Позволяет получить следующие данные: общее число записей, общее число полей, общее число ненулевых структурных записей, разброс молекулярной массы и числа всех и тяжелых атомов в структурах SDF файла. Информацию о всех химических элементах, встречающихся в структурах SDF файла: минимальное, максимальное число атомов в структурах, где этот элемент встречается, число структур, в которых встречается этот элемент.

Для неструктурных полей определяется число записей, в которых это поле присутствует, и тип всех значений этого поля (целое число, дробное или строка символов).

slicesdf.py – утилита предназначена для разрезания SDF файла на несколько SDF файлов меньшего размера с заданным числом записей. Новые SDF файлы именуются по порядку:

1.sdf, 2.sdf,…

extract.py – утилита предназначена для извлечения из SDF файла записей, в которых присутствует или не присутствует требуемое поле, при этом проверяется, не является ли его значение пустым.

delfield.py – утилита предназначена для удаления из всех записей SDF файла требуемого поля или всех полей, кроме требуемого. В качеств поля может выступать и структурная запись.

sdf2csv.py – утилита предназначена для извлечения информации из немолекулярных полей SDF файла и помещения ее в текстовый файл табличного вида, где значения полей разделены запятыми (формат CSV). Поля в создаваемом CSV файле следуют в порядке, в котором они были указаны в строке аргументов. Порядок записей CSV файла соответствует порядку записей исходного SDF файла.

sdf2mol.py – утилита предназначена для извлечения структур из SDF файла и записи их в отдельные файлы MOL формата. В качестве имени молекулярного файла можно задавать значение любого поля SDF файла (содержащее символы, не запрещенные в операционной системе для применения в именах файлов). При извлечении проверяется корректность каждой структурной записи.

mol2sdf.py – утилита предназначена для записи всех молекул, содержащихся в данной директории в MOL формате (с расширением *.mol), в один SDF файл. При работе проверяется корректность каждого MOL файла. Конечный SDF файл содержит для каждой структурной записи два дополнительных поля: порядковый номер записи и имя исходного MOL файла (без расширения).

workname.py – утилита для работы с именем структуры. Именем структуры в SDF файле является первая строка структурного поля. Утилита позволяет: удалять имена для всех структур SDF файла, заменять их на значение заданного поля, преобразовывать их в значения нового заданного поля, назначать в качестве имени порядковый номер структуры в SDF файле.

Литература

1. A.V. Soudackov,*A.L. Tchougreeff*, I.A. Misurkin. Electronic structure and optical spectra of transition metal complexes by the effective Hamiltonian method. **Theor. Chim. Acta** 83 (1992) 389 – 416.
2. A.M. Tokmachev, *A.L. Tchougreeff*. Fast NDDO Method for Molecular Structure Calculations Based on Strictly Localized Geminals. **J. Phys. Chem. A** 107 (2003) 358 – 365.
3. Pletnev, I. V. Can.J.Chem., 1994. V. 72. N5. Pp. 1404-1411.
4. M.B. Darkhovskii, M.G. Razumov, I.V. Pletnev, and *A.L. Tchougreeff*. Hybrid molecular mechanics-effective crystal field method for modeling potential energy surfaces of transition metal complexes. **Int. J. Quant. Chem.** 88 (2002) 588-605
5. Дж. Кастаньетто, Х. Рават, С. Шуман, к, Сколло, Д. Велиаф. *PHP программирование* СПб Символ 2001; *http://www.php.org*
6. Бизли Д. М. Язык программирования Python. Киев: Диасофт, 2000. 326 сс. *http://www.python.org*
7. Р.Дж. Яргер, Дж. Риз, Т. Кинг *MySQL и mSQL* СПб Символ 2000.
8. Дж. Остераут. Сценарии как высокоуровневое программирование для XXI века. Открытые системы, 1998, №3б сс. 12-18.
9. K. Hinsen, Molecular Modelling ToolKit. *http://starship.python.net/crew/hinsen/MMTK/*
10. SDF Toolkit (версия 1.11: Dec 31, 2002). http://cactus.nci.nih.gov/SDF_toolkit).