KNIME Extensions 1.5

User Manual

Schrödinger Press

Revision A, September 2012

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, command input and output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: Document Conventions.

In descriptions of command syntax, the following UNIX conventions are used: braces { } enclose a choice of required items, square brackets [ ] enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the $ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

Keyboard references are given in the Windows convention by default, with Mac equivalents in parentheses, for example CTRL+H (⌘H). Where Mac equivalents are not given, COMMAND should be read in place of CTRL. The convention CTRL-H is not used.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction

1.1 About KNIME

KNIME, the Konstanz Information Miner, is a modular framework (or platform) for graphically building and executing workflows and data analysis pipelines from predefined components, called *nodes*. KNIME is developed by Prof. Michael Berthold's group at the University of Konstanz in Germany, and can be downloaded free of charge from www.knime.org. It is built on the Eclipse Interactive Development Environment (IDE). KNIME is implemented in Java and currently runs on Windows and Linux.

A substantial number of standard data analysis and manipulation tools are distributed with KNIME, which include the following:

- I/O nodes for reading and writing data from files and databases
- Data manipulation nodes for managing the internal data tables that are used to pass information between components (e.g. filtering rows and columns, partitioning and joining tables, and so on)
- Charting and plotting tools
- Statistics and data mining tools, such as clustering, neural networks, and decision trees.

Additional features and functionality can be provided as KNIME extensions. KNIME extensions are collections of nodes that provide additional capabilities not present in the core KNIME environment. They can very easily be added to an existing KNIME installation. KNIME extensions may be licensed differently from the core KNIME platform, in particular if they are provided by third parties or include third party packages. For example, the KNIME chemistry extensions provide basic chemistry-related features such as reading and writing of common data formats and rendering 2D structures via the open-source Chemistry Development Kit (CDK). Another set of extensions is an interface with R, which a software environment for statistical computing and graphics. These particular extensions can be downloaded from the KNIME web site. You can download extensions from within KNIME—see Section 3.10 of the *Installation Guide* (Linux) or Section 4.2 of the *Installation Guide* (Windows) for more information.

1.2 About KNIME Extensions

Schrödinger has selected KNIME as the foundation for its pipelining capabilities. The Schrödinger KNIME extensions provide a large collection of chemistry-related tools that interface with Schrödinger applications and utilities. With the KNIME Extensions you can make use of the full spectrum of Schrödinger applications from within KNIME workflows. These extensions are intended to be well designed and integrated, flexible yet stable and reliable, and thoroughly tested.

The version of KNIME that the Schrödinger extensions are built on is not a proprietary version, but a freely available core KNIME distribution. This means that any other extensions you develop should work in the absence of the KNIME Extensions.

You can of course develop your own extensions that make use of Schrödinger software. To develop custom nodes you need at least a basic understanding of Java and the KNIME API.

Some of the important features that are available through the KNIME Extensions are:

- Ability to assemble, edit and execute workflows using a graphical tool
- Access to most of Schrödinger's modeling and cheminformatics tools
- Ability to integrate existing command-line tools and scripts
- Interoperability with third party applications
- Web services integration
- Support for distributed and high-throughput computing and compute-intensive modeling tasks
- Ability to visualize and interact with data at every step of a workflow
- Ability to share workflows

The Schrödinger KNIME extensions can be downloaded or updated from the Schrödinger web site, through the KNIME interface—see Section 3.10 of the *Installation Guide* (Linux) or Section 4.2 of the *Installation Guide* (Windows). A collection of entire workflows is also available for download from the Schrödinger web site, at http://www.schrodinger.com/knimeworkflows.

KNIME Overview

This chapter provides an introduction to some of the basic concepts and tasks in KNIME. You can find more information on the KNIME web site, at www.knime.org. It also includes information on running KNIME and some helpful hints.

2.1 The KNIME Panel

The main KNIME panel, or *workbench*, contains the following components:

- Menu bar—provides access to a range of tasks.
- Toolbar—Provides shortcuts for common tasks.
- Editor window, or *workspace*—This is the area in the center where workflows can be constructed. Each workflow is in a separate tab.

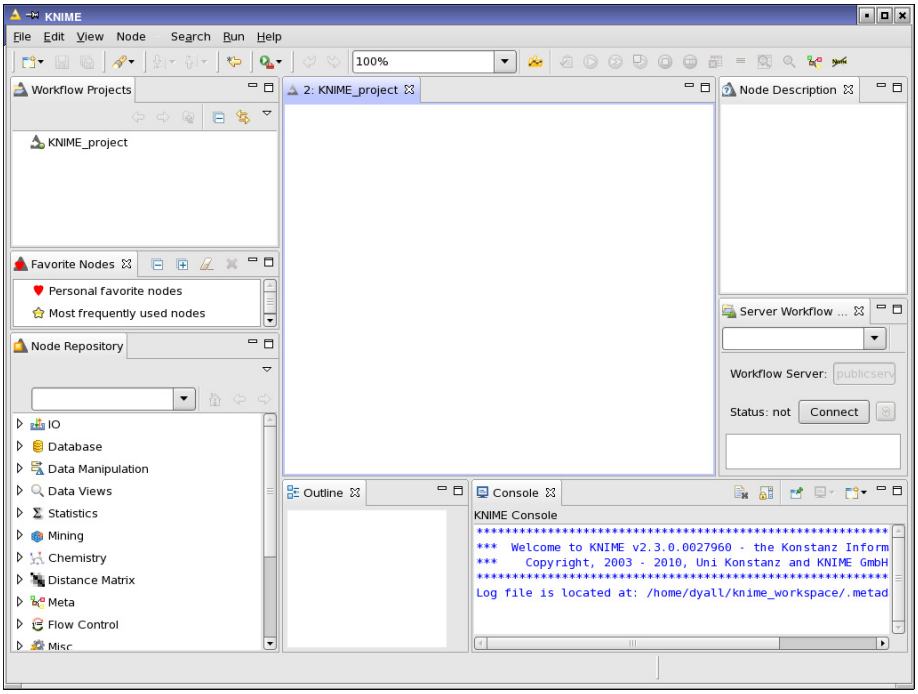

Figure 2.1. The KNIME workbench.

- Workflow Projects pane—shows all currently defined workflow projects. By default this pane is at the upper left. This pane has a shortcut (context, right-click) menu that allows you to perform various tasks on workflows, including creating new workflows, importing existing workflows, and exporting workflows.
- Favorite Nodes pane—shows nodes in order of last use and in order of frequency of use.
- Node Repository pane—shows all currently available nodes, in a tree view. The nodes are listed by name, and can be dragged into the workspace. By default this pane is at the lower left.
- Node Description pane—shows the description of a node.
- Outline pane—shows an outline of the current workflow. By default this pane is at the lower center.
- Console pane—displays warning and error messages. These messages are also written to the log file. By default this pane is at the lower right.

2.2 Nodes

The basic unit of a KNIME workflow is a *node*, also called a *module*. A node corresponds to a particular task, and is the basic processing unit of a workflow. In the workspace, each node has the following features:

- A title, at the top
- An icon, in the middle
- Ports for input and output. Each node must have at least one port, and can have multiple ports of the same type, for different kinds of input or output. The types of ports are:
  - Input ports, represented as triangles on the left of the icon, pointing in to the icon. These ports are for input of data.
  - Output ports, represented as triangles on the right of the icon, pointing out from the icon. These ports are for output of data.
  - Model ports, represented as blue squares, on either the right or the left of the icon. These ports are for input or output of data models.

  Each port has a tooltip that displays information about the kind of data that the port requires or generates.

Figure 2.2. Examples of nodes.

- A status display, below the icon. This display usually consists of a set of horizontal "traffic lights":
  - A red light means that the node is not ready to execute. It might not be fully connected; it might have incorrect or missing settings; or it might be connected to a node that is also not ready to execute.
  - An amber light means that the node is ready to be executed.
  - A green light means that the node has been executed and has sent any output to its output ports.

  When the node is executing, the status display changes to a progress indicator, with a blue bar.
- A sequence number, below the status display.
- A contextual (right-click) menu, which allows you to configure and execute the node, display the output views, edit the node, and display data for the ports.

When you select a node, either in the Node Repository or in the workspace, its description is displayed in the Node Description pane. The description should provide a summary of the function of the node, a description of its ports, and a description of the available views of the output.

Warnings and errors are indicated by an icon between the node icon and the status display. The warning or error message is displayed when you pause the pointer over the warning or error icon.

You can search for information on nodes by using Help → Search, or CTRL+F. A Search pane opens, in which you can enter an expression to search for and define the scope of the search.

2.3 Workflows

A workflow consists of a set of nodes, joined together so that all input and output is defined. To construct a workflow, drag the desired nodes into the workspace, and connect the ports. The ports are connected by dragging from the input port on a node to the output port on another node (or vice versa). Ports that are connected must have compatible data types, which you can check using the tooltips for the ports. Feedback loops are not permitted: you cannot connect the input of node A to the output of node B if the output of node A is already connected to the input of node B.

When you have connected all the nodes, you may need to configure some or all of the nodes. To do so, right-click on the node and choose Configure. The configuration is saved with the workflow. Re-configuring a node that has been executed resets it: all output is discarded.

To run all of a workflow, choose Execute All from the Node menu, or right-click on the last node and choose Execute. To execute a workflow up to and including a particular node, right-click on the desired node and choose Execute. Workflow execution starts with the first node that has not already been executed, and continues in sequence through the nodes until the node that you chose Execute for has been run.

If your KNIME installation or KNIME extensions is updated, executing any existing workflow loads and executes the updated nodes. A message is displayed in the Console to notify you of changes to Schrödinger nodes. You may have to reconfigure the changed nodes to run the workflow if the node settings have been altered.

A collection of workflows is available for download from the Schrödinger web site, at http://www.schrodinger.com/knimeworkflows.

2.4 Running KNIME from the Schrödinger Installation

When you install KNIME and the Schrödinger KNIME extensions from the Schrödinger distribution, they are installed into `$SCHRODINGER/knime-v`*version*.

Linux: Use the following command to run KNIME:

```
$SCHRODINGER/knime [options]
```

The options are described in Table 2.1.

Windows: Double-click the KNIME icon on your desktop. If you want to use any of the options, open a Schrödinger Command Prompt window from the Start menu, and use this command:

```
knime [options]
```

Table 2.1. Interactive options for the knime command.

Option	Description
`-data` *directory*	Use *directory* as the KNIME workspace. The default is Linux: `$HOME/knime_workspace` Windows: `%USERPROFILE%\knime_workspace`
`-defaultHost` *host*	Set the default host for execution of processes to the specified host. *host* is the name of an entry in the hosts file, `schrodinger.hosts`.
`-deleteTempFiles true|false`	Delete temporary files.
`-help`	Print information on command line options and exit.
`-knimeInstallDir` *directory*	Specify the path to the KNIME executable. This option should be used to run a version of KNIME other than the one in the Schrödinger software installation.
`-knimeTempDir` *directory*	Set the directory for KNIME temporary files to the specified location. If *directory* is `FromSchrodingerHost`, then the value of `tmpdir` for the `localhost` entry in the `schrodinger.hosts` file is used.
`-maxHeap` *value*	Set the maximum heap size to the specified value, e.g. 1024m.
`-maxThreads` *n*	Set the maximum number of working threads to the specified value.
`-ooCmd` *cmd*	Set the command to start a spreadsheet program from the View CSV node. The default is to open Open Office Spreadsheet with the command `oocalc`.
`-schrodingerTempDir` *directory*	Set the directory for Schrödinger temporary files to the specified location. If *directory* is `FromSchrodingerHost`, then the value of `tmpdir` for the `localhost` entry in `schrodinger.hosts` is used.
`-verbose`	Print more information on process and errors.
`-version`	Print version number of the Schrödinger extensions and exit.

Mac: Click the KNIME icon on the dock. If there is no KNIME icon on the dock, you can put it there by dragging it from the SchrodingerSuite2012 folder in your Applications folder. If you want to run KNIME from the command line, open a terminal window and use the same command as for Linux.

Some of the `knime` command options are stored as preferences, and can be changed by opening the Preferences panel from the File menu, and choosing KNIME → Chemistry → Schrödinger. Options are stored in a custom `knime.ini` file in your Schrödinger user resources directory (`$HOME/.schrodinger` on Linux and Mac, `%APPDATA%\Schrodinger` on Windows.)

2.5 Running KNIME from an External Installation

You can use the Schrödinger KNIME extensions with your own KNIME installation rather than use the installation provided by Schrödinger. First, you must install the extensions in your KNIME installation, as described in the Installation Guide, Section 3.10.1 (Linux) or Section 4.2.1 (Windows). When you have done so, you must set the `SCHRODINGER` environment variable to point to the Schrödinger installation, so that KNIME knows where to find the software. Instructions for setting environment variables are given in Appendix A of the *Installation Guide*. However, it is recommended that you do not set this environment variable globally because it is used by all applications, and you must remember to change it when you install a new Schrödinger software release.

Some alternatives for using the Schrödinger extensions with your own KNIME installation that do not set `SCHRODINGER` globally are given for each platform type in the following subsections. In the descriptions below, *knimepath* is the path to your KNIME installation, and you must substitute it with the actual path.

2.5.1 Linux

The options for Linux are listed below. For the first two, the commands should be run in a terminal window in which the SCHRODINGER environment variable is set.

- If your Schrödinger installation has an installation of KNIME, you can make it use the external KNIME installation by using this command:

  `$SCHRODINGER/knime -knimeInstallDir` *knimepath*

- You can use `$SCHRODINGER/run` with this command:

  `$SCHRODINGER/run` *knimepath*`/knime`

- You can create a script to run the external KNIME with the appropriate environment set, and put the script in your KNIME installation directory. An example script is given below:

```
#!/bin/bash
export SCHRODINGER=schrodinger-installation
if [ ! -d "$SCHRODINGER" ]; then
    echo "You must edit this script to set the SCHRODINGER environment variable to
the path to your Schrodinger software installation"
    exit 1
fi

mmshare=`$SCHRODINGER/hunt mmshare $SCHRODINGER | sed -e 's/bin/lib/'`
export LD_LIBRARY_PATH="$mmshare:$LD_LIBRARY_PATH"
if ! [ -f ./knime -a -x ./knime ]; then
```

```
    echo "You must place this script in the directory where KNIME is installed"
    exit 2
fi
./knime
```

2.5.2 Windows

- If your Schrödinger installation has an installation of KNIME, you can make it use the external KNIME installation by adding an option to the KNIME shortcut from your Schrödinger installation. Right-click on the KNIME shortcut and choose Properties, then add the option `-knimeInstallDir` *knimepath* to the end of the text in the Target text box (in the Shortcut tab). Make sure that you do not overwrite the existing text, and make sure that there is a space between the existing text and the new text.
- You can start KNIME from a Schrodinger Command Prompt window (Start → All Programs → Schrodinger2012 → Schrodinger Command Prompt) in one of two ways:

  `knime -knimeInstallDir` *knimepath*
  `run` *knimepath*`/knime`
- You can create a DOS (`.bat`) script and place it in your KNIME installation folder. An example script is given below.

```
@echo off

set SCHRODINGER=""
set status=0

if %SCHRODINGER%=="" (
     echo "You must edit this script to set the SCHRODINGER environment variable to
the path to your Schrodinger software installation"
     set status=1
     goto end
)

if not exist %SCHRODINGER%\ (
     echo "You must edit this script to set the SCHRODINGER environment variable to
the path to your Schrodinger software installation"
     set status=1
     goto end
)

set PATH=%WINDIR%\system32;.
set PATH=%SCHRODINGER%;%SCHRODINGER%\\unxutils;%SCHRODINGER%\\utilities;%PATH%
set PATH=%PATH:"=%

for /F %%I in ('%SCHRODINGER%\hunt.exe "mmshare"') do (
set MMSHARE_EXEC=%%I
)
```

```
set PATH=%MMSHARE_EXEC%; %PATH%

if not exist ./knime.exe (
    echo "You must place this script in the directory where KNIME is installed"
    set status=1
     goto end
)

knime.exe

:end
:: Hold this shell if any error occurs during startup
:: This will allow user to know the errors happened.
IF NOT "%status%"=="0" cmd /k
```

2.5.3 Mac

- You can use the commands given above for Linux (the first two options) in a terminal window.
- If your Schrödinger installation has an installation of KNIME, you can make it use the external KNIME installation by adding an option to the KNIME app. You cannot navigate to the KNIME app in Finder because it does not allow you to treat it as a directory or folder, but you can make the change by executing the following commands in a terminal window:

  ```
  cd /Applications/SchrodingerSuite2012/KNIME.app/Contents/MacOS
  sed '/exec/s/$/ -knimeInstallDir knimepath/' KNIME > KNIME.new
  mv KNIME.new KNIME
  ```

  where you must substitute *knimepath* with the path to the KNIME installation.

  Note that if you reinstall or replace the app, you will have to repeat this procedure. You can drag the modified app from the SchrodingerSuite2012 folder to the dock to use it from the dock.
- If KNIME is not installed in your Schrödinger installation, you can create a script to run the external KNIME with the appropriate environment set, and put the script in your KNIME installation directory. An example script is given below:

```
#!/bin/bash
export SCHRODINGER=schrodinger-installation
if [ ! -d "$SCHRODINGER" ]; then
    echo "You must edit this script to set the SCHRODINGER environment variable to
the path to your Schrodinger software installation"
    exit 1
fi
```

```
mmshare=`$SCHRODINGER/hunt mmshare $SCHRODINGER | sed -e 's/bin/lib/'`
export DYLD_LIBRARY_PATH="$mmshare:$DYLD_LIBRARY_PATH"
if ! [ -f ./knime -a -x ./knime ]; then
    echo "You must place this script in the directory where KNIME is installed"
    exit 2
fi
./knime
```

2.6 Common Tasks

To import an archived workflow (zip file):

1. Right click in the Workflow Projects pane, and choose Import KNIME workflow from the shortcut menu.
2. Select Select archive file, and click the corresponding Browse button.
3. Navigate to the desired zip file, and click OK.
4. Click Finish.

To add a bend in the connection between nodes:

1. Click on the connection to select it.
2. Pause the cursor over one of the ends of the connection until you see a hand (or on some systems a double-sided pair of crossed arrows)
3. Drag to create a bend.

To view the output of a node:

1. Right-click on the node.
2. Select Data Outport.

Helpful Hints

- Double click on a tab to enlarge that pane to full screen; double click again to return to the normal view.
- Drag tabs around to reposition panes. For example, drag the Workflow Project tab next to the Node Repository tab to have both in the same panel.
- Use the up/down and left/right arrow keys to navigate the nodes in a workflow.
- In tables, right-click on a header to display numbers as bars or in gray scale.

- In tables with 2D structures, drag the row height with the SHIFT key held down to adjust the height of all rows.
- If you would like to have more control over table width, use the Interactive Table node (View → Column Width).
- In the molecule Sketcher node, double click on a bond when in "draw bond" mode to change the bond order.
- Cut a node instead of deleting it if you don't want to see the "Do you really want to delete ..." warning.

KNIME Extensions Tutorial

This chapter provides a tutorial introduction to KNIME and the KNIME Extensions. In this tutorial, you will build a workflow that calculates molecular properties for a set of compounds provided as SMILES strings. The general outline of the workflow is as follows:

- Read a SMILES string from a file
- Carry out 1D to 3D conversion using LigPrep
- Calculate molecular properties using QikProp
- Extract a subset of properties for analysis
- View the molecular properties

3.1 Starting KNIME

1. Start KNIME as follows:

- On Linux, type the command `$SCHRODINGER/knime`.
- On Windows, double-click the KNIME icon on the desktop.

  While KNIME is starting, a message is displayed in the shell window from which KNIME was started. This message indicates that the workflows will be stored in the directory shown. On Linux, you can change the directory by using the `-data` *directory* option to the `knime` command.

  A splash screen is also displayed, as shown in Figure 3.1. The splash screen should include the Schrödinger logo under Installed Extensions. If this logo is missing, the version of KNIME you are running does not have the KNIME Extensions, and you must either add them, or run a version that does.

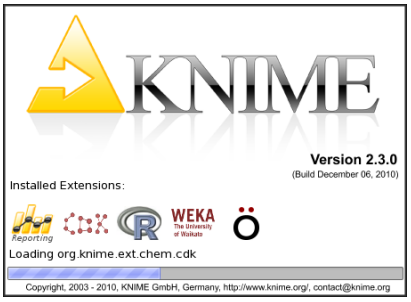

Figure 3.1. The KNIME splash screen.

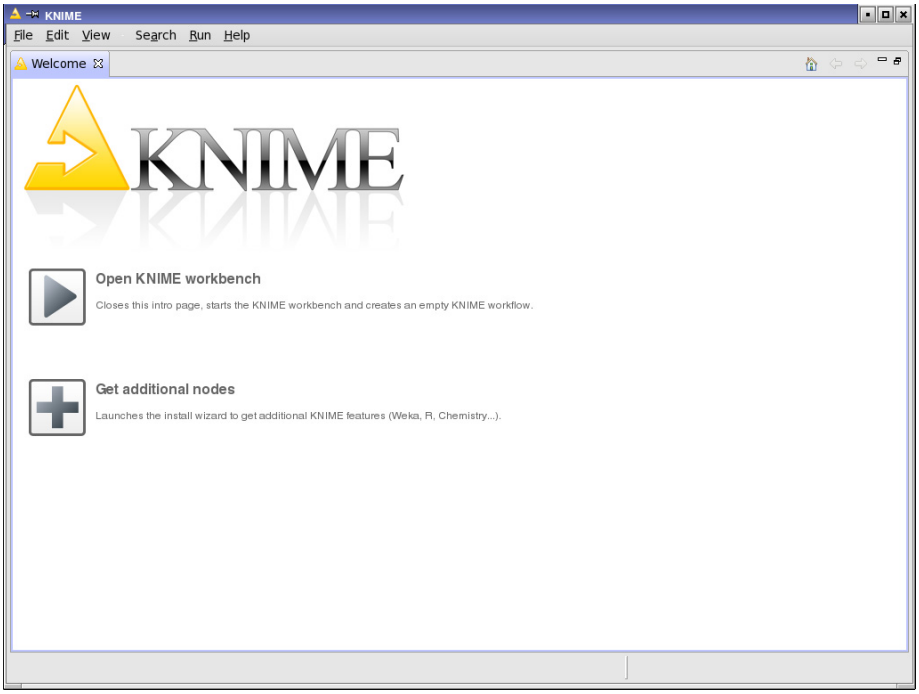

Figure 3.2. The initial KNIME window.

If this is the first time you have run KNIME, you will see a panel (Figure 3.2) that offers a choice of downloading additional features or launching KNIME. To continue with this tutorial, click Open KNIME workbench to display the KNIME workbench.

If you have run KNIME previously, the KNIME workbench opens directly.

Take some time to examine the layout of the KNIME workbench.

At the top there is a menu bar, with a toolbar below it. The View menu allows you to display various tabs. By default, all are displayed.

On the left side are two tabs, labeled Workflow Projects and Node Repository. The Workflow Projects tab lists the projects that are available in the current KNIME session. The Node Repository tab lists all the nodes that are available, in a tree structure. At the bottom are two tabs, Outline and Console. The Outline tab displays an outline of the current workflow. The Console tab shows error messages or log messages. On the right side is the Node Description tab, which displays the description of the selected node.

The remaining area is the workspace, where workflows can be constructed, edited, and executed. The current workflow is highlighted in the Workflow Projects tab.

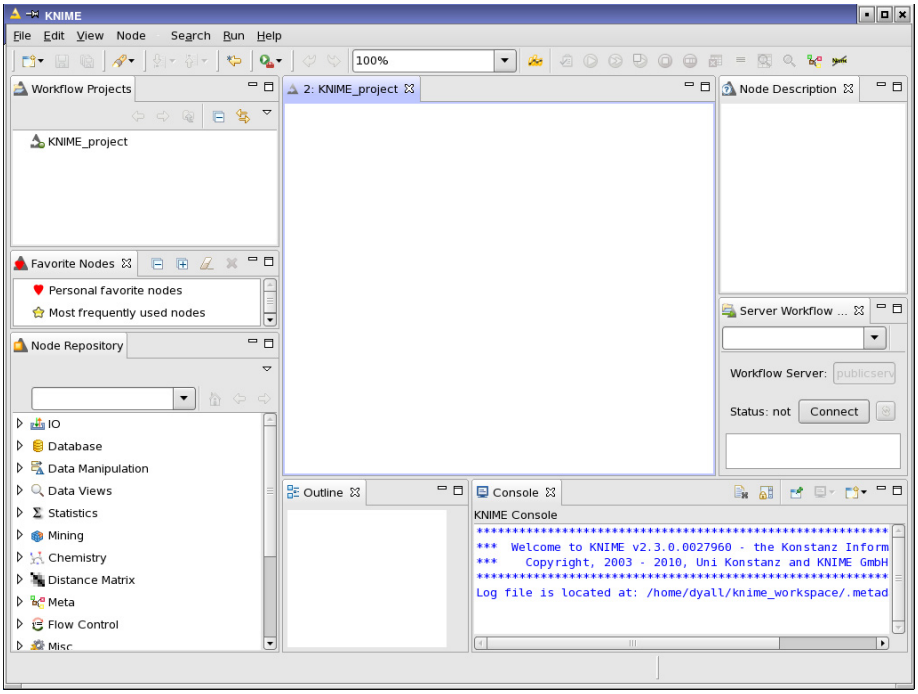

Figure 3.3. The initial KNIME workbench.

3.2 Creating a New KNIME Project

1. Choose File → New.

   The New panel opens. This panel allows you to create a new object using a wizard. In this case, we want to create a new KNIME project.

2. Select New KNIME Project in the Wizards list, and click Next.

   The next screen is labeled New KNIME Workflow Wizard, and allows you to name the project.

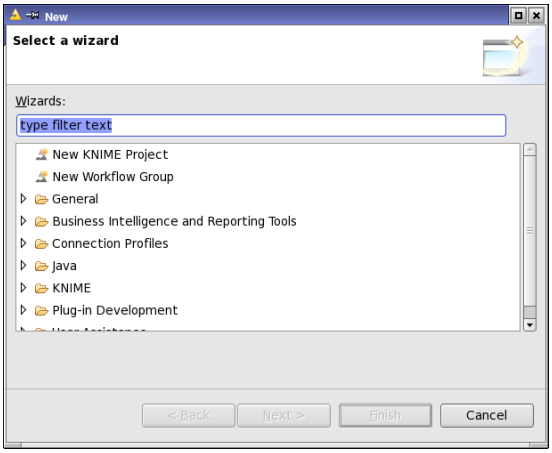

Figure 3.4. The New panel.

3. Enter `Molecular Properties` in the Name of the workflow to create text box, and click Finish.

You should now see a new entry in the Workflow Projects tab and a new tab in the main workspace, labeled Molecular Properties. To make sure the new project is the active workflow project, click it in the Workflow Projects pane.

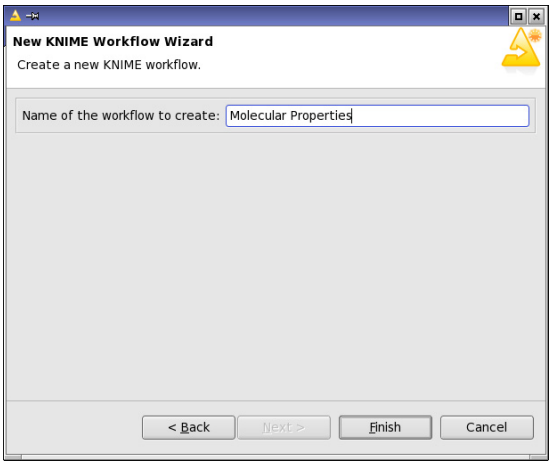

Figure 3.5. The New KNIME Project Wizard.

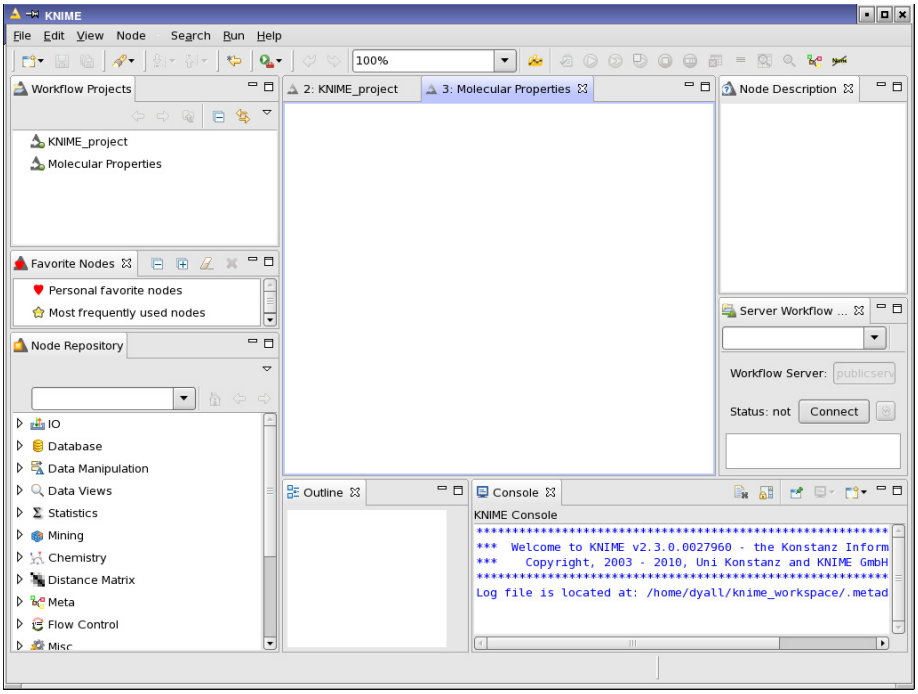

Figure 3.6. The KNIME workbench with the new project.

3.3 Adding a Smiles Reader

The first node (or component) to add to the workflow is a SMILES reader.

1. In the Node Repository, open up the Schrödinger category by clicking on the triangle on the left side.
2. Open up the Readers/Writers category.
3. Drag the Smiles Reader node into the workspace, and place it on the left.

   The node has a title, an icon that represents the node, a set of “traffic lights” that indicates the status of the node (status indicator), and a node number. The node description is displayed in the Node Description tab. The icon has a triangle on the right side. This is a point at which you can connect this node to another node, and represents the output of the node (the triangle is like an arrowhead pointing out from the node). If you pause the pointer over this triangle, it displays a brief description of the type of output, in a tooltip.

   The node is also added to the Outline tab. As you add nodes, they are added to this tab, which provides a view of the entire workflow.

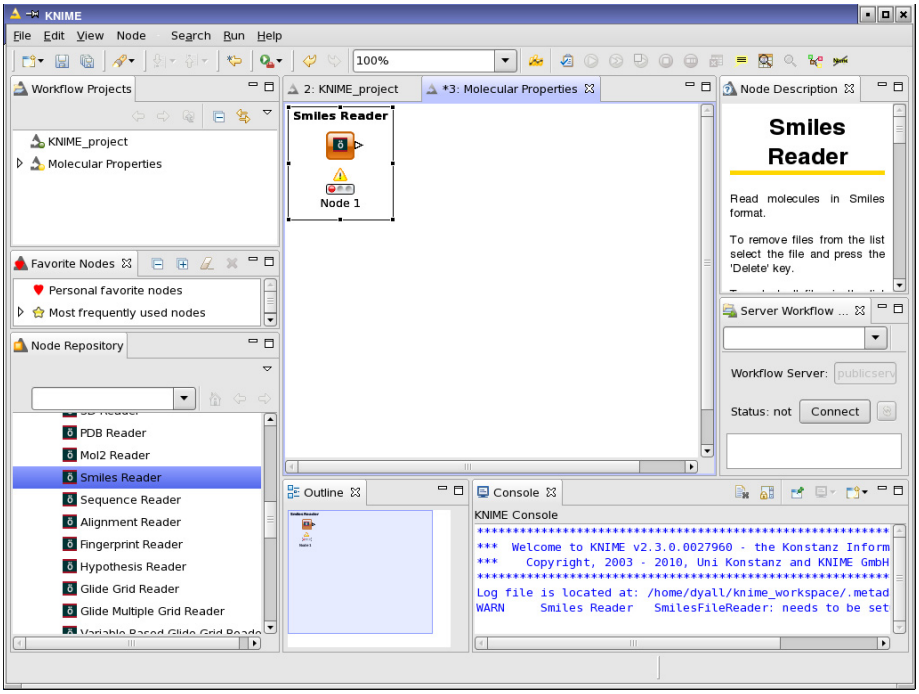

Figure 3.7. The KNIME workbench with the Smiles Reader node, before configuration.

The node initially shows a "red light" because it cannot be run as is, simply because it has not been configured yet. There is also a warning icon: an exclamation point in a yellow triangle above the traffic lights. If you pause the pointer over the exclamation point, you will see that this is a warning message, which tells you that the node needs to be set up. You might also see a warning message in the Console tab, and also in the terminal window from which you started KNIME. To configure this node, you need to specify the file it should read.

4. Right click on the Smiles Reader node (anywhere) and choose Configure from the shortcut menu.

   The configuration dialog box for the Smiles Reader node opens.

5. Click Add File(s).

   A file dialog box opens that allows you to select a file.

6. Navigate to and select the file `$SCHRODINGER/macromodel-v`*version*`/ligprep/samples/examples/1D_smiles.smi`, then click Open.

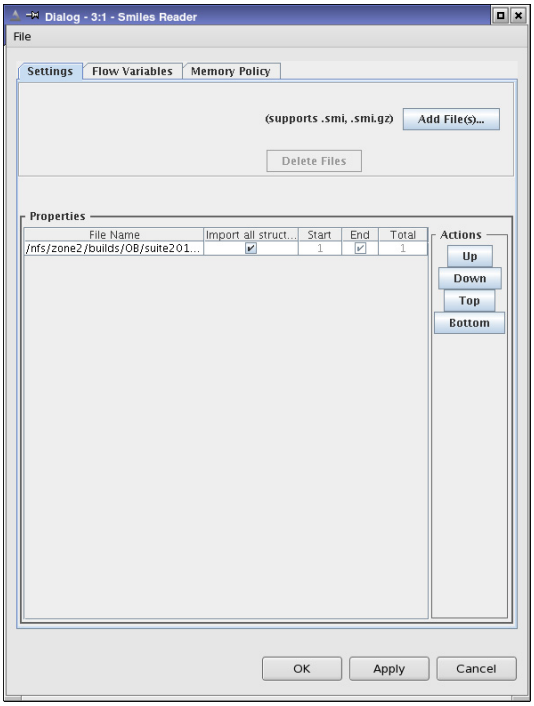

Figure 3.8. The configuration dialog box for the Smiles Reader.

The file is added to the Properties table in the configuration dialog box. This table has columns for the properties that define how the file is to be used. There is a check box in the Import all structures column. If you wanted to limit the range of structures imported, you could deselect the check box, and enter values in the Start and Total columns.

7. Click OK.

   The configuration dialog box closes. The warning symbol in the node has gone, and the "yellow light" is now showing.

3.4 Adding the LigPrep and QikProp Nodes

An alternative way of locating nodes is to use the search text box in the node repository. We will use this mechanism for subsequent nodes in this tutorial.

1. Type `ligprep` into the search text box and press ENTER.

   The node repository shows all the nodes that match the text entered in the search box. The search is case-insensitive.

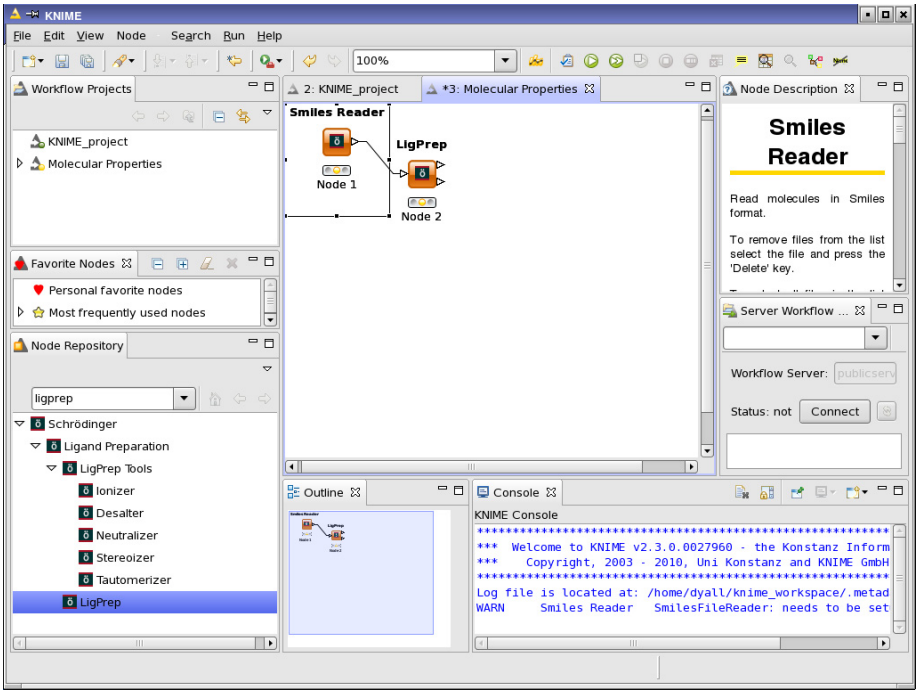

Figure 3.9. The KNIME workbench with the LigPrep node connected to the Smiles Reader node.

2. Select the LigPrep node and drag it into the workspace.
3. Connect the Smiles Reader node to the LigPrep node by clicking on the small triangle on the right side of the Smiles Reader node and dragging the pointer over to the small triangle on the left of the LigPrep node (see Figure 3.9).

   This process connects the output of the Smiles Reader node to the input of the LigPrep node. When you run the workflow, the structures read by the SMILES reader are passed on to LigPrep as input.

   The default LigPrep settings are appropriate for the calculations carried out in this tutorial so there is no need to configure the LigPrep node.

   If you are curious about what settings are being used, you can open the configuration panel for the LigPrep node (by right-clicking on the node and selecting Configure) and examine the settings.
4. Type `qikprop` into the search text box and press ENTER.
5. Select the QikProp node and drag it into the workspace.

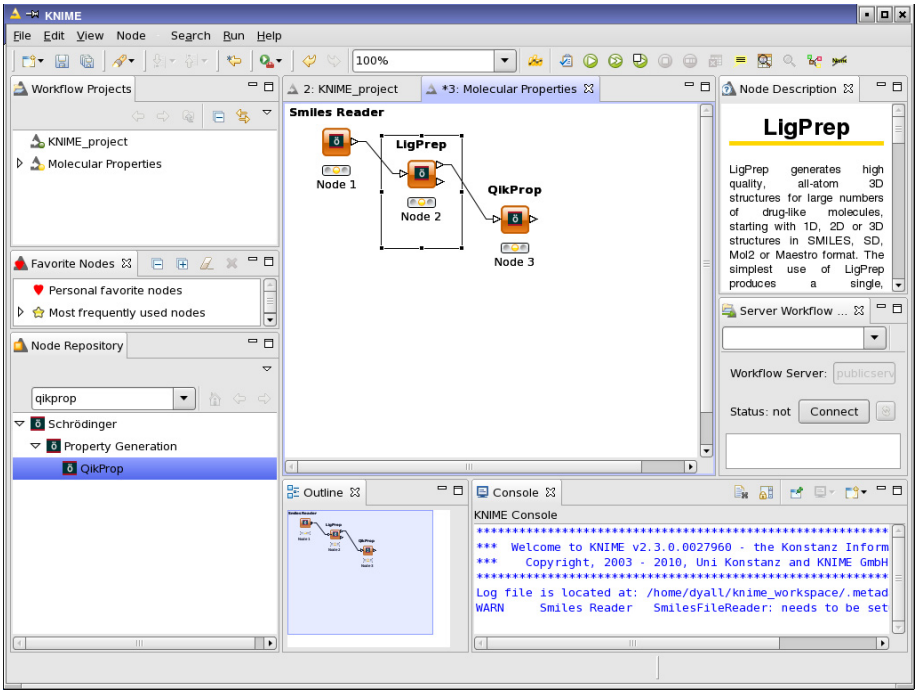

Figure 3.10. The KNIME workbench with the QikProp node connected to the LigPrep node.

6. Connect the LigPrep node to the QikProp node by clicking on the upper of the two small triangles on the right side of the LigPrep node and dragging the pointer over to the small triangle on the left of the QikProp node (see Figure 3.10).

   A description of each of the output ports (small triangles) is displayed when you pause the pointer over the port. Here the upper port is for the main output, and the second port is for "failed" molecules.

7. Right click on the QikProp node and choose Configure from the shortcut menu.

   The configuration dialog box for the QikProp node opens. This configuration dialog box has four tabs, one for QikProp settings, one for Job Control settings, one for flow variables, and one for memory policy. In the Job control tab, you can select the host on which to run the job, for example.

8. Select the Output only option, and click OK.

All three nodes should be showing yellow lights.

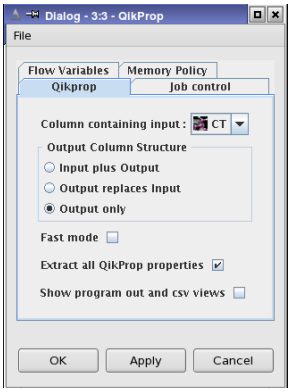

Figure 3.11. The configuration dialog box for QikProp.

3.5 Running the Workflow

At this point you have a complete workflow that is ready to run.

- Right-click on the QikProp node and choose Execute.

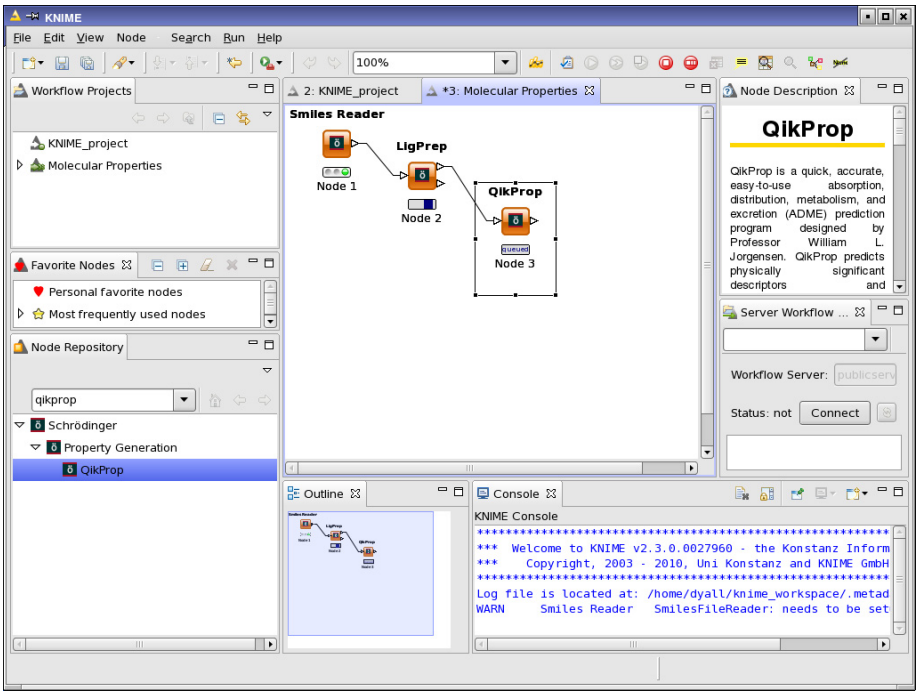

Figure 3.12. The KNIME workbench during execution of the workflow.

The individual nodes that make up the calculations are executed in sequence, up to the node that you chose to execute. Here, none of the nodes had already been executed. If, for example, the Smiles Reader had already been executed, the workflow would start at the next node in the sequence.

As each node finishes its task, its green traffic light shows. Running nodes have a dark blue box that moves left and right in the status indicator. For nodes that are waiting to run, the status indicator shows the word queued. When the workflow finishes, all nodes should show a green light.

3.6 Extracting Properties

QikProp adds the molecular properties that it calculates as Maestro properties to the molecular structures (or CTs) it creates. In this exercise, we will extract properties from these structures.

1. Type `extract` into the search text box and press ENTER.

   The node repository shows all the nodes that match the text entered in the search box. The search is case-insensitive.

2. Drag the Extract MAE Properties node into the workspace.

3. Connect the QikProp node to the Extract MAE Properties node.

   If you need to rearrange the nodes, simply drag them to where you want them. The connections remain intact when you do so.

4. Right click on the Extract MAE Properties node and choose Configure from the shortcut menu.

   The configuration dialog box for the Extract MAE Properties node opens. There are three sections in the Properties and target column tab: Exclude, which contains a list of properties to be excluded (not extracted); Select, which provides tools for selecting the properties; and Include, which contains a list of properties to be included (extracted). By default, all properties are included (selected for extraction). For this tutorial we want to analyze only small number of properties.

5. Click remove all.

   All the properties are moved from the Include list to the Exclude list.

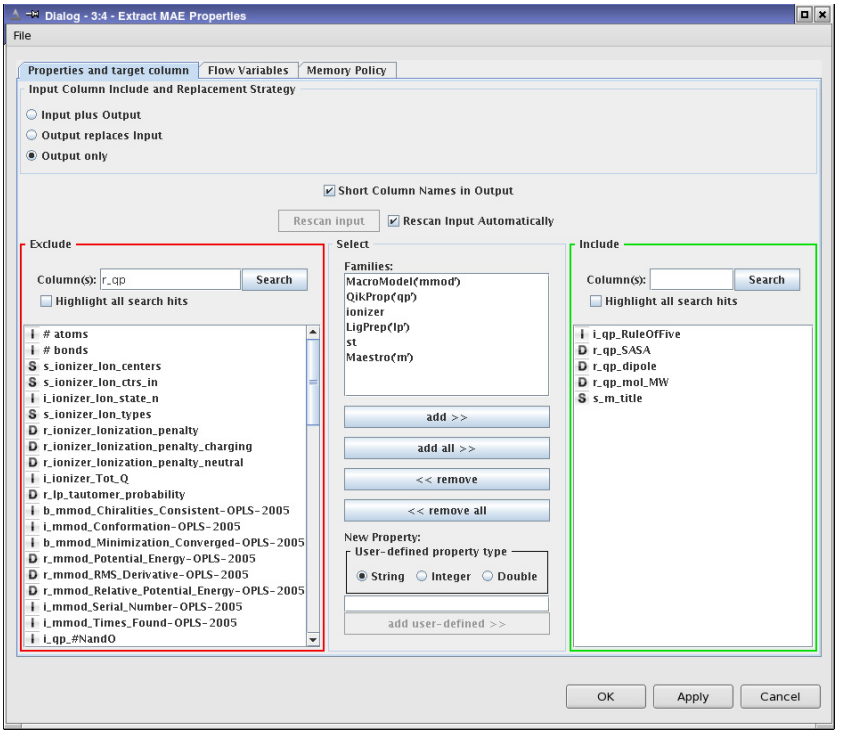

Figure 3.13. The configuration dialog box for the Extract MAE Properties node.

6. Add the following properties to the Include list, by selecting them in the Exclude list and clicking add.

   - s_m_title
   - r_qp_mol_MW
   - r_qp_SASA
   - r_qp_dipole
   - i_qp_RuleOfFive

   You can search for the property by typing the name into the Column(s) search box and pressing Enter. The matching properties are highlighted if Highlight all search hits is selected.

7. Select the Output only option, and click OK.

   The configuration dialog box closes.

8. Execute the Extract MAE properties node (right-click and choose Execute).

9. Right-click on the Extract MAE properties node and choose 0 Properties.

   A table is displayed with the extracted properties listed in it.

You can use the Interactive Table node to display the results automatically. Simply add it to the workflow using the procedure described above and choose Execute and Open View to run it.

3.7 Writing the Results to Disk in Excel Format

KNIME output data can be exported in common formats, so that you can use the data with other tools—for example, adding the data to a database, or analyzing the data. KNIME itself has a wide variety of data analysis and visualization tools, which are introduced in the next exercise. In this exercise, the table of molecular properties calculated in the workflow is exported in Excel format via the XLS Writer.

1. Type `XLS` into the search text box and press ENTER.
2. Drag the XLS Writer node into the workspace.

   Hereafter, we will simply say "Add a *nodename* node to the workspace" for these two steps.
3. Connect it to the output from the Extract MAE Properties node.

   The new node initially shows a 'red light' since it is not configured yet. In this case, configuration involves setting the name of the output file.
4. Right click on the XLS Writer node and select Configure.

   The configuration dialog box for the XLS Writer node opens.

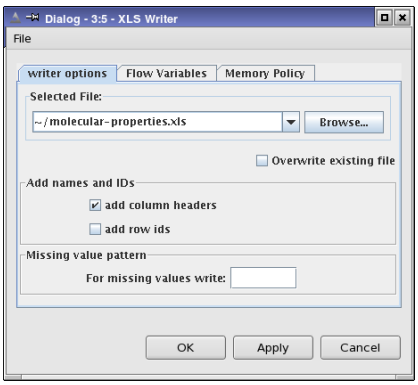

Figure 3.14. The configuration dialog box for the XLS Writer node.

5. Type the path to the output file into the Selected File text box.

   For example, to store it in your home directory, type `/home/`*username*`/molecular-properties.xls`.

6. In the Add names and IDs section, select add column headers.

   This option includes the original property names in the Excel output as column headers.

7. Click OK.

8. Execute the XLS Writer node.

The Excel file generated by the workflow above can now be opened with MS Excel or other applications that can work with Excel format (such as OpenOffice).

3.8 Visualization of the Results

KNIME includes a wide variety of data analysis and visualization tools. It can be very helpful to analyze the data set generated in a workflow using graphical tools to get a general sense of the data. Obviously, the details of the visual analysis very much depend on the questions you are trying to answer. In this exercise, we show two simple analyses to introduce some of these tools.

3.8.1 Analyzing the Distribution of Violations of Lipinski's Rules via a Histogram

To get a sense of how many compounds in the data set violate Lipinski's Rule of Five we can use a histogram. To carry out the analysis follow these steps:

1. Add a Column Filter node to the workspace.

2. Connect it to the output of the Extract MAE Properties node.

   You can have more than one node connected to the output of a given node. This allows you to make use of the output for different purposes.

3. Right click on the Column Filter node and choose Configure.

   The configuration dialog box for the Column Filter node opens. It is similar to the configuration dialog box for the Extract MAE Properties node (see Section 3.6 on page 23).

4. Click remove all.

   All the properties are moved from the Include list to the Exclude list.

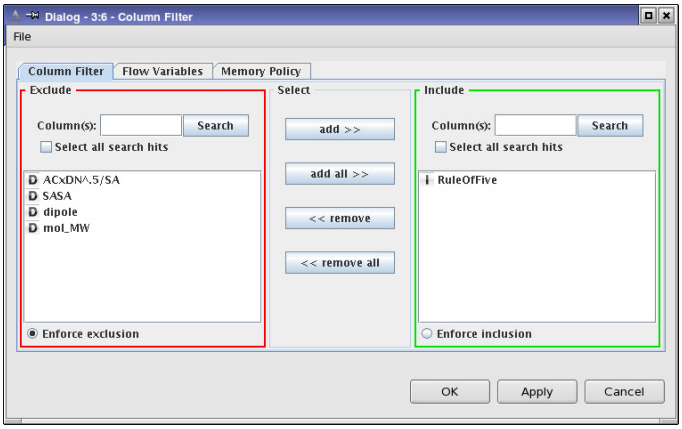

Figure 3.15. The configuration dialog box for the Column Filter node.

5. Select RuleOfFive in the Exclude list and click add.

   The property is added to the Include list.

6. Click OK.

   The configuration dialog box closes.

7. Add a Histogram node (the one labeled just Histogram) to the workflow and connect it to the output of the Column Filter node.

8. Right-click on the Histogram node and choose Configure.

   The configuration dialog box for the Histogram node opens.

9. In the Options tab, select RuleOfFive in the Available columns list (the only entry) and click add.

10. The property is moved to the Aggregation columns list.

11. Click OK.

    The configuration dialog box closes.

12. Right-click on the Histogram node and select Execute and open views.

    A window showing a histogram opens.

13. Click Fit to size in the Default Settings tab.

    Notice that the majority of the compounds do not violate Lipinski's Rule of Five but quite a few compounds violate at least one of the rules. You can find out how many are in each category using labels.

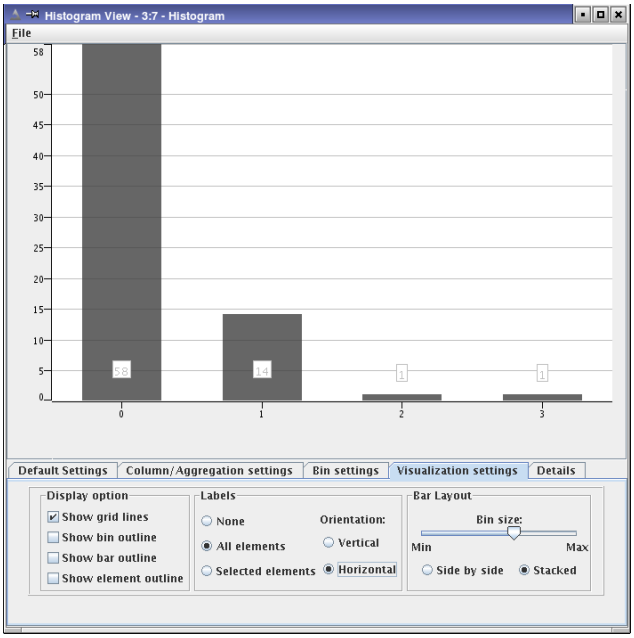

Figure 3.16. The histogram of Lipinski's rule violations.

14. In the Visualization settings tab, in the Labels section, click All elements.

    A box appears that displays the counts for each bar, but it is displayed in vertical orientation.

15. Click Horizontal under Orientation.

    The labels are now displayed in horizontal orientation.

3.8.2 Plotting the Solvent Accessible Surface Area (SASA) Against the Molecular Weight

In this exercise, you will create a scatter plot of the solvent accessible surface area (SASA) against the molecular weight.

1. Add another Column Filter node and connect it to the output from the Extract MAE Properties node.
2. Right click on the Column Filter node and select Configure.

   The configuration dialog box for the Column Filter node opens.

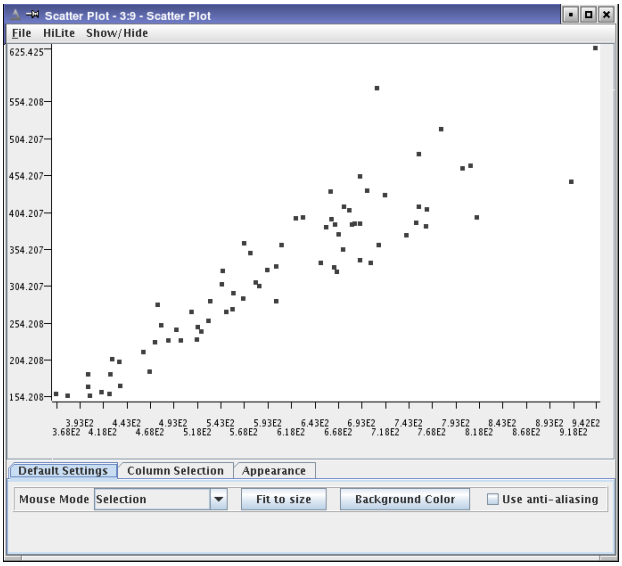

Figure 3.17. The scatter plot of SASA vs molecular weight.

3. Transfer title, dipole, and RuleOfFive to the Exclude list.

   You can use shift-click and control-click to select these properties, then click remove. The properties are moved from the Include list to the Exclude list, leaving the mol_MW and SASA properties in the Include list.

4. Click OK.

   The configuration dialog box closes.

5. Add a Scatter Plot node and connect it to the output of the Column Filter node.

   There is no need to configure this node for the analysis carried out here.

6. Right-click on the Scatter Plot node and select Execute and open view.

   A window showing a scatter plot opens.

You can carry out this analysis without including a Column Filter node, since the Scatter Plot node allows you to select the columns interactively. To do this, simply connect a Scatter Plot node to the output from the Extract MAE Properties node, choose Execute and open view from the shortcut menu, and select the X and Y columns in the Column Selection tab. Interactive plotting of the results is useful of course, but you may prefer to extract the relevant columns.

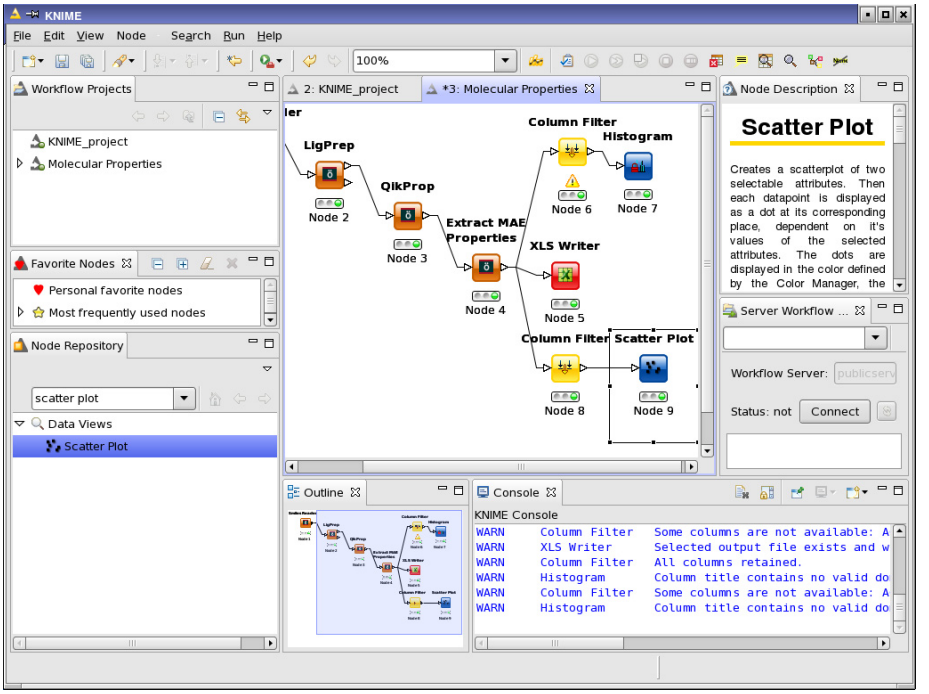

Figure 3.18. The final view of the KNIME workbench.

The final view of the KNIME workbench is shown in Figure 3.18. The nodes added in these two exercises did not fit in the default workspace. The workspace automatically scrolls and adds scroll bars as necessary, and the Outline view highlights the area covered by the visible part of the workspace in blue.

3.9 Workflow Samples

Samples of the workflow generated in this tutorial can be found at the following location:

`$SCHRODINGER/knime-v`*version*`/tutorial`

There are two copies of the workflow: one generated before the postprocessing steps, and one generated at the end.

For further workflow examples, go to the KNIME workflows download page at http://www.schrodinger.com/knimeworkflows. There is a short description for each workflow on this page and in the README file in the download. The README contains a list of the nodes used in each workflow, which can be used to find workflows illustrating how to use a specific node or perform a task of interest. Each of the important nodes in the workflow examples is commented to explain what it does.

Using KNIME with Maestro

Maestro can interact with KNIME in a number of ways:

- You can exchange structures with KNIME, by using a panel in Maestro and KNIME nodes designed for this purpose.
- You can run entire KNIME workflows from Maestro, setting up input data in a customizable panel.
- You can attach a Run Maestro node to your workflow, which starts a new Maestro session with the structure input imported as an entry group.

The first two features are described in the following sections.

4.1 Exchanging Structures with KNIME

You can transfer structures with their properties to the Maestro Project Table from a KNIME workflow, perform some tasks on the structures in Maestro, such as changing the structure or adding properties, and then transfer them back to KNIME. This is done via KNIME-Maestro Connector nodes in the KNIME workflow and a lightweight server that listens for connections from a KNIME workflow. A license is not required to use this feature.

To set up the connection:

1. In Maestro, choose Workflows → KNIME → Establish Structure Exchange with KNIME.
2. Click Start to start the KNIME-Maestro connection server.
3. Start KNIME and load the workflow you want to run.

To send structures to a node:

1. Choose the source of structures from the Structures to send option menu, and select or include the relevant structures.
2. Select the node that you want to send the structures to in the KNIME nodes connected list.
3. Click Send.

When a connector node is run, the input structures to that node are added to the Project Table in Maestro, and the node is added to the KNIME nodes connected list. Execution of the node

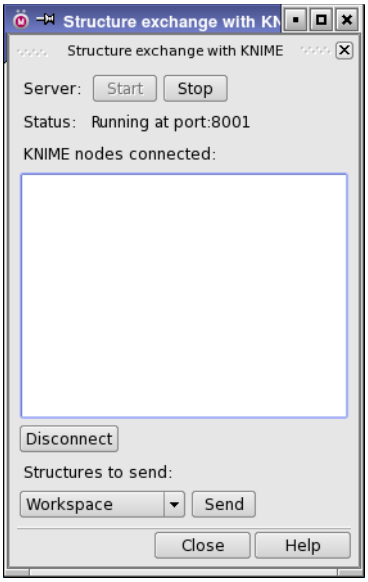

Figure 4.1. The Structure Exchange with KNIME panel.

halts until you send structures to it, by selecting the node in the list and clicking Send. If the node has no input, when it is executed it waits until you send structures from Maestro to it. Once the structure has been sent, the node is automatically disconnected. You can also disconnect the node manually if no structure is needed from Maestro, by selecting it in the list and clicking Disconnect.

4.2 Running KNIME Workflows from Maestro

As well as transferring structures between KNIME and Maestro, you can run entire workflows from Maestro. The workflows can be installed on the KNIME Workflows submenu of the Workflows menu, and can be set up and executed with panels that allow you to make settings that are passed to KNIME. The panels are created as part of the process of "wrapping" a workflow for use with Maestro. The wrapping involves creation of a Python script that contains both the Maestro panel for the workflow and the necessary code to execute the workflow.

4.2.1 Setting Up a Workflow

Setting up a workflow for use in Maestro must be done from Maestro. The workflow can be a new workflow, or it can be an existing workflow. Once a workflow is wrapped, with configuration of the nodes for communication with Maestro and a panel to run it, it can be tested, refined, and then installed on the KNIME Workflows submenu of the Workflows menu.

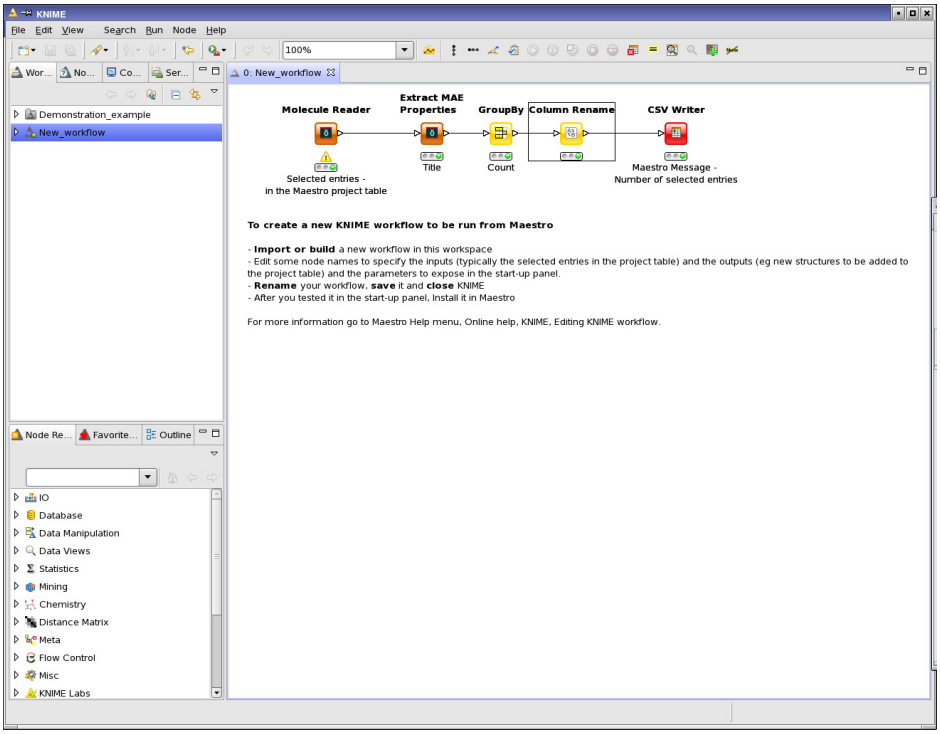

Figure 4.2. The KNIME workspace with the example workflow.

To set up a KNIME Workflow for use with Maestro:

1. Choose Workflows → KNIME Workflows → New.

   This action runs KNIME from a script that performs the necessary processing to create the Maestro interface after saving the workflow. You can set up a workflow in KNIME without using this menu item, but it will not be wrapped for use with Maestro.

   You may have to wait a minute or so while KNIME starts.

   An example workflow is displayed with brief instructions when KNIME opens (see Figure 4.2).

2. Import or build a new workflow in the KNIME workspace.

   If you have an existing workflow that you want to use in Maestro, you can import it. Use File → Import, select a source and select the workflow.

3. Edit the node name of the Reader, Writer and Quick Form nodes that you want to use to transfer information from or to Maestro.

The name must contain specific phrases, as described in Section 4.2.2 on page 35. For example, selected entries from the Maestro Project Table will be passed to a Molecule Reader node that has “Selected entries” in its name.

4. Configure any nodes that you need to configure.

   For example, if you always want to use a particular file as input to a node, you should configure the input port of the node at this stage. You will also need to configure Quick Form nodes for communication with Maestro, as described in Section 4.2.2 on page 35.

5. Add a description of the workflow to the Comments text box in the meta information.

   To do this, in the Workflow Projects panel, right-click on the workflow name and select Edit Meta Information. The description is displayed in the panel that is created for the workflow.

6. (Optional) Reset the workflow before saving it so as to keep its size to the minimum.

   To do this, right-click on the first node of the workflow and select Reset.

7. Close your workflow.

   Use the tab at the top of the workspace to close it. If you have unsaved changes, you will be prompted to save them.

8. Rename your workflow.

   Right click on the workflow name in the Workflow Project list and choose Rename to rename it. The workflow must not have spaces in the name, but these can be replaced with underbars (_). The name you use is the name listed in Maestro under Workflows → KNIME Workflows, with any underbars replaced by spaces.

9. Close KNIME.

10. Click Yes in the dialog box that is displayed.

    A startup panel is created for the workflow, and is opened so you can check the input, output, and parameters, test the workflow, and install it in Maestro. From this panel you can also return to KNIME and edit the workflow.

To test the new workflow:

1. Select entries in the Project Table or include them in the Workspace, according to what is listed under Input.
2. Make choices with the controls under Parameters.
3. If you want to store the results of testing the workflow (including input, output, and parameter settings) in the workflow, select Save after execution.

Figure 4.3. Workflow startup panel.

4. Click Test.

   The workflow runs.

If you encounter problems in your workflow, you can open the workflow in KNIME again by clicking Edit, and making changes. When you exit KNIME again, you are returned to the startup panel so you can test it again, or continue on to install the workflow. You can always edit the workflow again later, after installation.

To install the workflow:

- Click Install.

When the workflow is installed, a message is displayed in a dialog box. Dismissing this dialog box also closes the startup panel. The workflow is listed in Maestro under Workflows → KNIME Workflows.

4.2.2 Setting Up Nodes for Communication with Maestro

Communication between Maestro and KNIME is set up by adding special text to the names of Molecule Reader, Molecule Writer, CSV Writer, and Quick Form nodes. This text identifies the node as one that can receive input from Maestro or send output to Maestro. The text that you can use in these node names is given in Table 4.1. No license is needed to use these nodes.

Structural input and output is handled with Molecule Reader and Molecule Writer nodes. You can configure the Molecule Reader nodes with any file for testing the workflow. These file names will be replaced when running the workflow.

Table 4.1. Node name text for communication with Maestro.

Text	Node	Action
`Selected entries`	Molecule Reader	Read the entries that are selected in the Project Table.
`Included entries`	Molecule Reader	Read the entries that are included in the Workspace.
`New group`	Molecule Writer	Add the output structures as a new group to the Project Table.
`Maestro message`	CSV Writer	Display a message in an information box. The message is the output of the CSV Writer node.
`Maestro commands`	CSV Writer	Execute Maestro commands on completion of the workflow and incorporation of the results. The commands are in the first column of the CSV Writer output.
`GUIsetting`	Quick Form: Double Input, Integer Input, String Input	Add a text box control to the panel and assign the value that is set in the text box.
`GUIsettingRADIO`	Quick Form: String Radio Buttons	Add a set of radio buttons to the panel presenting alternatives to be selected.
`GUIsettingCOMBO`	Quick Form: String Radio Buttons	Add a combo box (option menu) to the panel presenting alternatives to be selected.

Setting of variables or choice of options is handled with Quick Form nodes. These nodes are used to add controls to the panel that is created for running the workflow. The types of Quick Form nodes you can use are:

- Double Input
- Integer Input
- String Input
- String Radio Buttons

The first three types create a text box in which you can enter a value of the specified type. The value is assigned to a flow variable that can be used as input to another node.

To configure an Input node:

- Enter the label to use on the text box in the Label text box.
- Enter the tool tip for the control in the Description text box.
- Enter the name of the flow variable in the Variable Name text box.
- Enter the default value of the flow variable in the Variable Value text box.
- Set a Weight value to determine the position of the control in the panel.

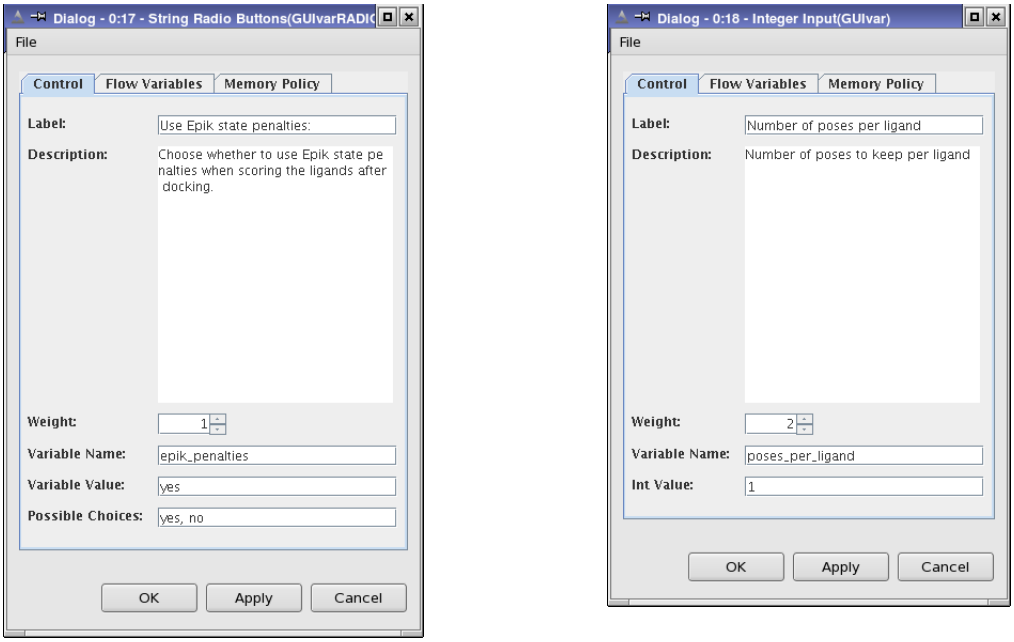

Figure 4.4. Configuration dialog boxes for String Radio Buttons node (left) and Integer Input node (right).

The String Radio Buttons node allows you to present choices between text values, which can be displayed in the panel as a set of radio buttons, if the node name contains the text `GUIsettingRADIO`, or an option menu (combo box) if the node name contains the text `GUIsettingCOMBO`. The text for the radio button or menu item that is selected in the panel is assigned to the flow variable as its value.

To configure a String Radio Buttons node:

- Enter the label of the radio set or combo box in the Label text box.
- Enter the tool tip for the control in the Description text box.
- Enter the name of the flow variable in the Variable Name text box.
- Enter the default value of the flow variable in the Variable Value text box.
- Enter the allowed values as a comma-separated list in the Possible Values text box.
- Set a Weight value to determine the position of the control in the panel.

The GUI controls are presented in the panel in order of the Weight value that you assign when you configure the node.

Using Quick Form nodes, you can set integer and real (double) variable values and string values. To set a Boolean variable, you must use a String Radio Buttons node with the values `true` and `false` (case insensitive).

If you want to use multiple Quick Form nodes for input to a particular node, you can string them together (connect the flow variable output from one to the flow variable input from the next).

After connecting the (last) Quick Form node to the flow variable input of the node whose variables you want to set, configure that node to associate the desired flow variables from the node with the ones you set in the Quick Form nodes.

If you want to execute Maestro commands after the structures are added to the Project Table, use a CSV Writer node with the text `Maestro commands` in the node name. The commands must be contained in the first column of the input table, one command per row. You must deselect Write column header when configuring the node. For Maestro command syntax, see the *Maestro Command Reference Manual*.

To display results you can also use a CSV Writer node, with the text `Maestro message` in the node name. The table of results is displayed in an information box, which opens when the node is executed.

4.2.3 Running and Modifying Workflows

Once a workflow is installed, you can run it by choosing Workflows → KNIME Workflows → *workflow-name*. The panel for the workflow opens, and you can make settings and then click Run. The workflow itself runs on the local host, but uses any host settings made in individual nodes.

If you want to save the workflow after running it, select Save after execution in the workflow panel. The workflow will then include all the results of running it, and any settings you changed under Parameters are also changed. You can open the workflow in KNIME to inspect Report nodes. Saving the workflow can also be useful for diagnosing execution problems. Saving the workflow when running on a large data set is not recommended, as all the data is stored with the workflow.

If you want to modify a workflow that is already installed, you can do so from the workflow panel. You can edit the workflow itself, or create a copy of the workflow.

- To edit the workflow, click Open in KNIME. KNIME is started with the workflow loaded, and you can make and save changes. You should not rename the workflow, as the workflow with the current name is associated with the panel.
- To copy the workflow, click Create a Copy. A copy of the workflow is placed on the KNIME Workflows submenu of the Workflows menu, with `copy` appended to the name. When you select this new workflow from the menu, a prompt to rename it is posted and the KNIME interface is started. You should make any changes to the workflow that you want to make, and close the workflow. You must then rename the workflow and close

KNIME before it will be available for execution from Maestro, under the new name.

4.2.4 Managing Workflows

The workflows that you have installed can be managed in the Manage Workflows panel, which you open by choosing Workflows → KNIME Workflows → Manage. In this panel you can do the following tasks:

- Rearrange the order of the workflows on the menu, by selecting them and using the arrow buttons.
- Duplicate workflows, by selecting them and clicking Duplicate. The duplicates are placed after the existing workflows, and you must then rename them before you use them.
- Delete workflows, by selecting them and clicking Delete.
- View the description of a workflow, by selecting it and clicking Description.

4.2.5 Installing and Updating Pregenerated Workflows

If you have wrapped workflows that are not installed in Maestro, you can install them by choosing Workflows → KNIME Workflows → Install. In the Install Workflows panel, you can browse to the desired location, and select workflows to install. The items that are listed are Python scripts. These can include ordinary Python scripts as well as KNIME workflow wrappers, so you must take care to choose the KNIME workflow wrappers for installation.

Schrödinger provides a set of pregenerated workflows on its web site. You can download and install these workflows by choosing Workflows → KNIME Workflows → Update. The Update Workflows panel allows you to view and select from the available workflows and download them. You must then install them into Maestro.

The installed workflows are kept in the `workflows` folder of your Maestro user resources directory, which is `$HOME/.schrodinger/maestro93` on Linux and Mac and `%APPDATA%\Schrodinger\maestro93` on Windows.

4.2.6 Summary

To set up and install a new workflow:

1. Choose Workflows → KNIME Workflows → New.
2. Import or build a new workflow in the KNIME workspace.
3. Edit the Reader, Writer, and Quick Form node names to configure them for input, output and parameter passing.

4. Rename your workflow without spaces.
5. Save it and close KNIME.
6. Test it in the startup panel.
7. Install it in Maestro.

To run a workflow:

1. Choose Workflows → KNIME Workflows → *workflow-name*.
2. Make settings in the panel for input and output.
3. Click Run.

Running Workflows from the Command Line

KNIME workflows can be run directly from the command line, rather than from the graphical interface. This feature can be particularly useful for time-consuming workflows or workflows that have to be run repeatedly for different input data sets. Furthermore, being able to run KNIME workflows from the command line is useful when trying to integrate KNIME workflows into other applications, such as command line scripts or web services.

Note: The KNIME Batch Executor is an experimental feature and may change significantly in future releases.

5.1 The knime Command

The command to use on Linux and Mac for batch execution of KNIME workflows has the following syntax:

```
$SCHRODINGER/knime -batch [-reset] {-workflowFile=file|-workflowdir=dir}
    [-destfile=file] [-option=option-setting] [-nosave] [options]
```

The batch options for this command are given in Table 5.1.

Table 5.1. Batch options for the knime command.

Option	Description
`-batch`	Run the batch engine rather than the graphical interface.
`-nosave`	Do not save the workflow after execution has finished. Results must be explicitly written to files, which will be preserved.
`-reset`	Reset the workflow prior to execution, so that it starts from the beginning. If not used, the workflow is executed from its current state.
`-workflowFile=`*file*	ZIP file with a ready-to-execute workflow in the root of the ZIP file.
`-workflowDir=`*dir*	Directory with a ready-to-execute workflow. If you use this option, the workflow must not be in use by another process, batch or interactive.
`-destFile=`*file*	ZIP file to which the executed workflow should be written. If omitted the workflow is saved in place, overwriting the input workflow.
`-option=`*nodeID*`,`*name*`,`*value*`,`*type*	Set the option named *name* of the node with ID *nodeID* to the given *value*, which has type *type*. There must be no spaces in the option setting. This option can be repeated as many times as settings are needed.

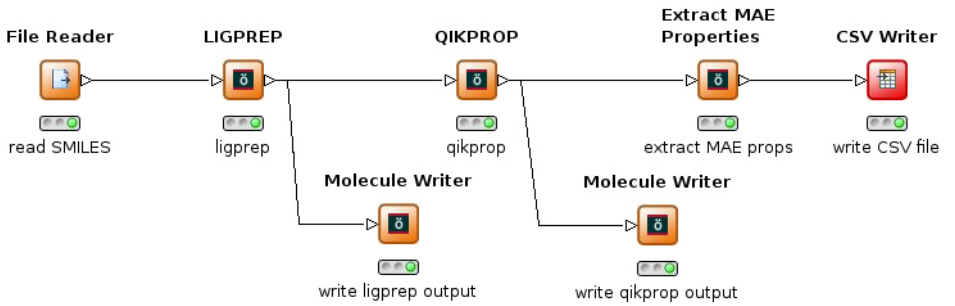

Figure 5.1. Example workflow for batch execution.

On Windows, you can run from the command line by first opening a Schrödinger Command Prompt or a Schrödinger Power Shell from the Start menu. Then use the following command (assuming that the workflow is on your C drive):

```
knime.exe -consoleLog -noexit -nosplash -reset
    -application org.knime.product.KNIME_BATCH_APPLICATION
    -workflowFile=C:\path-to-workflow\workflow.zip
```

No path to the executable is needed because the path is set up for you, and the `.exe` extension is optional. The Unix `-batch` argument is replaced on Windows with the first five arguments given in this command. Note that the path is specified with backslashes, because you are running in a DOS shell, not a Unix shell. This form of the command should be used in the examples below.

5.2 Batch Example

As an example of a batch workflow, consider the workflow shown in Figure 5.1. This sample workflow calculates molecular properties using QikProp from ligands provided as SMILES strings and writes those properties to a data file in CSV format. The SMILES strings are converted to 3D structures using LigPrep. In addition to the molecular properties, the intermediate structures produced by LigPrep and QikProp are written to disk in Maestro format. The input and output file names use generic input and output file names such as `/tmp/input.smi` and `/tmp/ligprep-out.mae`. For the workflow to run successfully the input SMILES data need to be saved in a disk file called `/tmp/input.smi`. The workflow can be run with the command

```
$SCHRODINGER/knime -batch -reset -workflowFile=wfpath/batch-example.zip
```

where *wfpath* is the directory in which you stored the zipped workflow. The output from the run simply gives the time taken for the run:

```
Finished in 41364ms
```

Upon successful completion the workflow generates the following output data files:

```
/tmp/ligprep-out.mae
/tmp/qikprop-out.mae
/tmp/molprops.csv
```

A copy of this workflow is available in a zip file, at `$SCHRODINGER/knime-v`*version*`/tutorial/batch-example.zip`.

5.3 Modifying Node Settings

The example workflow discussed above used hard-coded generic input and output file names. This approach allows you to run different input data sets by simply renaming or copying the actual input and output files to the respective names. While this is a simple approach, it lacks flexibility. Instead it would be desirable to be able to control the settings of the nodes directly. This becomes especially important when you want to change the settings of nodes, for example to turn certain options on or off, or to modify numerical settings such as cutoffs and convergence criteria.

The KNIME batch executor provides a facility for changing the settings of every node in a workflow. Conceptually this facility is simple and easy to use but the actual mechanics depend on the complexity of the workflow. The settings can be made with the `knime` command using `-option`. There are four pieces of information that need to be provided for each setting: the node ID, the setting name, the value, and the type of value.

The first step in controlling the settings of a workflow is finding out what the settings of the nodes actually are. To do this you have to analyze various files in the workflow itself. There are currently no tools available for doing this analysis, so it must be done manually.

For the purpose of describing the process, we will again use the sample workflow introduced above (starting with the ZIP file). If you want to use this as an exercise, create a temporary directory to hold the workflow, change to that directory and unzip the workflow archive:

```
mkdir tempdir
cd tempdir
unzip wfpath/batch-example.zip
```

The output from unzipping the workflow archive is as follows:

```
Archive: wfpath/batch-example.zip
  inflating: batch-example/.lock
  inflating: batch-example/.project
  inflating: batch-example/CSV Writer (#7)/settings.xml
  inflating: batch-example/Extract MAE Properties (#6)/data/data_0/data.xml
  inflating: batch-example/Extract MAE Properties (#6)/data/data_0/data.zip
  inflating: batch-example/Extract MAE Properties (#6)/data/data_0/spec.xml
  inflating: batch-example/Extract MAE Properties (#6)/settings.xml
  inflating: batch-example/File Reader (#1)/data/data_0/data.xml
  inflating: batch-example/File Reader (#1)/data/data_0/data.zip
  inflating: batch-example/File Reader (#1)/data/data_0/spec.xml
  inflating: batch-example/File Reader (#1)/settings.xml
  inflating: batch-example/LIGPREP (#2)/data/data_0/data.xml
  inflating: batch-example/LIGPREP (#2)/data/data_0/data.zip
  inflating: batch-example/LIGPREP (#2)/data/data_0/reference_0/data.xml
  inflating: batch-example/LIGPREP (#2)/data/data_0/reference_0/data.zip
  inflating: batch-example/LIGPREP (#2)/data/data_0/reference_0/spec.xml
  inflating: batch-example/LIGPREP (#2)/data/data_0/spec.xml
  inflating: batch-example/LIGPREP (#2)/internal/internalData.xml
  inflating: batch-example/LIGPREP (#2)/settings.xml
  inflating: batch-example/Molecule Writer (#3)/internal/internalData.xml
  inflating: batch-example/Molecule Writer (#3)/settings.xml
  inflating: batch-example/Molecule Writer (#5)/internal/internalData.xml
  inflating: batch-example/Molecule Writer (#5)/settings.xml
  inflating: batch-example/QIKPROP (#4)/data/data_0/data.xml
  inflating: batch-example/QIKPROP (#4)/data/data_0/data.zip
  inflating: batch-example/QIKPROP (#4)/data/data_0/reference_0/data.xml
  inflating: batch-example/QIKPROP (#4)/data/data_0/reference_0/data.zip
  inflating: batch-example/QIKPROP (#4)/data/data_0/reference_0/spec.xml
  inflating: batch-example/QIKPROP (#4)/data/data_0/spec.xml
  inflating: batch-example/QIKPROP (#4)/internal/internalData.xml
  inflating: batch-example/QIKPROP (#4)/settings.xml
  inflating: batch-example/workflow.knime
```

Note the layout of the workflow directory. Subdirectories correspond to nodes in the workflow and every node has a number (or ID) associated with it. The IDs are assigned when the workflow is created and do not change when you add or delete nodes.

To change the settings for the input file in this workflow you need to determine the ID for the File Reader node, of which there is only one. The relevant directory is "File Reader (#1)" so the node ID is 1. You can also determine the node ID in the GUI either from the default node name (which is Node *n*) or by opening the configuration dialog, which shows the node ID in the title bar, for example File Reader (#1) or Molecule Writer (#3).

Information on the setting that controls the name of the input data file is in the `settings.xml` within the "File Reader (#1)" subdirectory. To extract this information, you will normally have to open this file in a text editor. In this case, the easiest way to locate the relevant setting is to look for the hard-coded file name, `/tmp/input.smi`, which is on the following line:

```
<entry key="DataURL" type="xstring" value="file:/tmp/input.smi"/>
```

The relevant node setting is named `DataURL` and is of type `String`. The type is not exactly the same as in the XML file, which is `xstring`. This type maps to `String` for the purpose of input to the batch executor. The current value of the setting is `file:/tmp/input.smi`. Note that in this case the node represents the data location as a URL so the file name is prefixed with `file:`.

Simple scalar settings such as `DataURL` are easy to modify. To point the workflow to a different input file, such as `/tmp/new-input.smi`, you can use the following `-option` setting on the command line:

```
-option=1,DataURL,"file:/tmp/new-input.smi",String
```

To run the workflow with this new data file, you can use the following command:

```
$SCHRODINGER/knime -batch -reset -workflowFile=wfpath/batch-example.zip
    -option=1,DataURL,"file:/tmp/new-input.smi",String
```

Another example of a file type is the output file for the CSV Writer (node 7). Information on this file from the corresponding `settings.xml` file is found on the following line:

```
<entry key="filename" type="xstring" value="/tmp/molprops.csv"/>
```

Since the CSV writer can only write to actual files on disk (as opposed to generic URLs), the setting for the file name is a plain string without any prefix. To change the node setting to write the file `new-molprops.csv`, you can use the following `-option` setting:

```
-option=7,filename,"/tmp/new-molprops.csv",String
```

Thus to run the workflow with custom names for the input SMILES file and the output CSV file, you can use the following command:

```
$SCHRODINGER/knime -batch -reset -workflowFile=wfpath/batch-example.zip
    -option=1,DataURL,"file:/tmp/new-input.smi",String
    -option=7,filename,"/tmp/new-molprops.csv",String
```

The discussion so far has illustrated how to change input and output file settings. You can also change numerical settings. For example, the `settings.xml` file for the LIGPREP node contains the following lines (among others):

```
<entry key="hostname" type="xstring" value="localhost:2"/>
<entry key="force_field" type="xstring" value="OPLS_2005"/>
<entry key="force_field_arg" type="xstring" value="-bff 14"/>
<entry key="retain_state" type="xboolean" value="false"/>
<entry key="neutralize" type="xboolean" value="false"/>
<entry key="generate_possible" type="xboolean" value="true"/>
```

```
<entry key="ph" type="xstring" value="7.0"/>
<entry key="pht" type="xstring" value="2.0"/>
<entry key="ionizer" type="xboolean" value="true"/>
```

These lines contain settings for numeric values, which are treated as strings, and Booleans, for which the type to use in the `-option` setting is `Boolean`. The setting names have obvious interpretations (which should be true for all the Schrödinger nodes), so that it is not difficult to work out what settings to make.

These lines also contain a setting for the host name. This setting is passed as the `-HOST` argument when the Schrödinger program is executed. The host name can include the number of processors. For nodes that also specify the number of jobs (as LigPrep does), there is usually an `njobs` setting that allows you to set the number of jobs. You can run the workflow above with the following command to set the host name to `clus_queue`, the number of nodes to 2, and number of jobs to 2 for LigPrep:

```
$SCHRODINGER/knime -batch -reset -workflowFile=wfpath/batch-example.zip
    -option=2,hostname,"clus_queue:2",String -option=2,njobs,2,String
```

5.4 Running Workflows

This section contains information about running workflows in various circumstances.

When a KNIME workflow is executed, a lock is placed on the workflow, and it cannot be executed by any other process until the lock is released. The lock is actually a file in the workflow directory, so it only applies when you are running interactively or specify a directory for a workflow with the `–workflowDir` option. If you specify a zip file with the `-workflowFile` option, the zip file is first extracted into a temporary location, and then executed. The lock therefore exists in the temporary copy, and not in the zip file.

This means that if you want to run a particular workflow multiple times, the runs can be concurrent if you use a zipped workflow, but must be consecutive if you use a workflow directory. You might, for example, want to concurrently run several instances of the same workflow with different options or different input files.

The KNIME workflow that is executed starts execution at the current state. If you have already run the workflow up to a particular point, batch execution starts at that point, and continues to the end of the workflow. If you want to start from the beginning, you should use the `-reset` option.

If you want to be able to switch between interactive and batch execution, you can specify the workflow with the `–workflowDir` option. For example, you could execute early stages of the workflow in the GUI and complete later stages from the command line. When you do so, you should *not* include the `-reset` option, because that option clears the intermediate results.

If you are only interested in the final results of a workflow, and do not want to save any of the intermediate calculations, you can run with the `-nosave` option. The temporary copy of the workflow is discarded, but any files that are written are kept.

By default, the workflow is saved at the end of the run. If you used a zip file, it is replaced. If you want to write the results to a new location, you can use the `-destFile` option to specify the new zip file. This option is useful if you want to iterate over options for a particular node, for example. It also allows you to save the results of multiple concurrent runs in a unique location.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*. Check the following sources for other information:

- The knowledge base, at http://www.schrodinger.com/kb
- Known Issues pages, available on the Support Center.

The manuals are installed in the `docs` directory of the software installation, and are also available from the Schrödinger Support Center.

Contacting Technical Support

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail:	help@schrodinger.com
USPS:	Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone:	(503) 299-1150
Fax:	(503) 299-4532
WWW:	http://www.schrodinger.com
FTP:	ftp://ftp.schrodinger.com

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- KNIME Extensions purchaser (company, research institution, or individual)
- Primary KNIME Extensions user
- Installation, licensing, and machine information as described below.

Gathering Information for Technical Support

This section describes how to gather the required machine, licensing, and installation information, and any other job-related or failure-related information, to send to technical support.

1. Open the Diagnostics panel.
   - **Maestro:** Help → Diagnostics
   - **Windows:** Start → All Programs → Schrodinger-2012 → Diagnostics
   - **Mac:** Applications → Schrodinger2012 → Diagnostics
   - **Command line:** `$SCHRODINGER/diagnostics`
2. When the diagnostics have run, click Technical Support.

   A dialog box opens, with instructions. You can highlight and copy the name of the file.
3. Attach the file specified in the dialog box to your e-mail message.
4. If you are having problems with KNIME, send the console log as an attachment to the email message. You can find the log in the following locations:
   - **Linux:** the terminal window from which you started KNIME.
   - **Windows:** `%LOCALAPPDATA%\Schrodinger\appcrash\knime.log`.
   - **Mac:** Applications → Utilities → Console.

120 West 45th Street
17th Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

245 First Street
Riverview II, 18th Floor
Cambridge, MA 02142

8910 University Center Lane
Suite 270
San Diego, CA 92122

155 Gibbs St
Suite 430
Rockville, MD 20850-0353

Dynamostraße 13
D-68165 Mannheim
Germany

Zeppelinstraße 73
D-81669 München
Germany

Potsdamer Platz 11
D-10785 Berlin
Germany

Quatro House
Frimley Road
Camberley GU16 7ER
United Kingdom

8F Pacific Century Place
1-11-1 Marunouchi
Chiyoda-ku, Tokyo 100-6208
Japan

No. 102, 4th Block
3rd Main Road, 3rd Stage
Sharada Colony
Basaveshwaranagar
Bangalore 560079, India

SCHRÖDINGER®