Epik 3.2

User Manual

Schrödinger Press

While care has been taken in the preparation of this publication, Schrödinger assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

This publication may refer to other third party software not included in or with Schrödinger software ("such other third party software"), and provide links to third party Web sites ("linked sites"). References to such other third party software or linked sites do not constitute an endorsement by Schrödinger, LLC or its affiliates. Use of such other third party software and linked sites may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for such other third party software and linked sites, or for damage resulting from the use thereof. Any warranties that we make regarding Schrödinger products and services do not apply to such other third party software or linked sites, or to the interaction between, or interoperability of, Schrödinger products and services and such other third party software.

May 2015

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

Epik is a program for the prediction of the pK_a values of the ionizable groups in ligands, and for the generation of the probable ionized and tautomerized structures within a given pH range. Epik rapidly and consistently predicts pK_a values, employing the widely used and respected Hammett and Taft empirical equations. The values of the parameters in these equations are determined from fits to experimental data either from the literature or from Schrödinger's own development efforts. Epik employs extensions to Hammett and Taft technology for handling mesomers, including standardization of mesomeric forms and charge spreading. Tautomer probabilities are estimated using a database of tautomerizations derived from experiment and quantum chemical calculations. Both the ionization and tautomerization employ user-modifiable databases of parameters. Collections of probable structures are generated by iteratively tautomerizing and ionizing the base structure. In addition, Epik can be used to generate structures suitable for studies involving metalloproteins.

Since Epik is fast, and yet has a range of options, it is suitable for processing large collections of input structures for a variety of purposes. Epik may also be used by Schrödinger's ligand preparation product, LigPrep, instead of the `ionizer` and `tautomerizer` found in LigPrep. Epik may be run from the command line or more conveniently from Maestro's Epik panel.

Epik's structure generation mode estimates state penalties which reflect the predicted populations in solution for the structures it generates. These penalties can be used in the scoring of poses in Glide docking calculations to obtain better enrichment.

1.1 The Epik Manual

This manual explains how to use Epik, a program designed to predict pK_a values of the ionizable groups in ligands, and to generate probable ionized and tautomerized structures within a given pH range. The manual is organized as follows:

- Chapter 1 provides an overview of Epik and the processes it uses to predict pK_a values and generate probable structures.
- Chapter 2 explains how to run Epik from Maestro and from the command line.
- Chapter 3 provides detailed explanations of the science behind Epik.
- Chapter 4 details the process of structural adjustment used by Epik.
- Chapter 5 explains the process of generating sequential pK_a values.
- Appendix A describes the format of the pK_a parameter file.
- Appendix B describes the format of the tautomer database file.

- Appendix C lists the Maestro properties generated by an Epik run.
- Appendix D describes a script for comparing predicted sequential pK_a values with experiment.

The appendices are followed by a section on getting help.

1.2 Epik Technology

This section provides a brief overview of the technology behind Epik. More detailed information is available in Chapter 3. Epik runs in three general modes: pK_a prediction for the structures provided, protonation and tautomerization state adjustment consistent with a specified pH range, and sequential pK_a estimation by systematically adding and removing protons. pK_a values are estimated for the resulting structures for all of these methods.

Mesomers—different valid Lewis structures for the same molecule that can have the same net formal charge associated with different atoms—present challenges. For instance, multiple ways to represent the formal charges amongst the ionizable functional group and substituents can greatly expand the number of SMARTS patterns needed to identify them. To reduce this problem Epik uses an empirical approach to standardize mesomeric forms used in recognizing molecular fragments. Another complication is that mesomers typically represent extremes in the allocation of charge and thus in the values of the various contributions to the estimated pK_a values. In practice the molecule is better represented by a combination of the various mesomers. Epik uses a charge-spreading technology which partially cancels opposite formal charges and distributes the remaining formal charges. The spread formal charges are used to generate parameters that are interpolations between the charged and uncharged versions for the substituents and heteroaromatic groups.

1.2.1 pK_a Prediction

Epik uses empirical Hammett and Taft relations to predict pK_a values. The first step in this process involves recognizing functional groups that may be ionized by the addition or removal of a proton. Each functional group has a base-line pK_a value and ρ parameter which reflects the sensitivity of the functional group to perturbations from the rest of the molecule. For each such ionizable group the rest of the molecule is conceptually divided into fragments, each of which has a known tendency to perturb ionizable groups. From the tabulated base-line pK_a value, the sensitivity of the functional groups to perturbations, and the strength of the perturbing influences from the various fragments, a prediction of the pK_a for the functional group of this molecule is made.

The various parameters involved in Hammett and Taft equations are derived from fits to experimental data. There is a large collection of such data and pre-fit parameters in the literature.

When making pK_a predictions, Epik uses a mixture of literature parameters and parameters determined by Schrödinger. Since Hammett and Taft methodology is empirical it is extremely rapid; typically significantly less than 1 second is needed to estimate the pK_a values for all ionizable sites in a ligand-like molecule when using a 2GHz Pentium 4 processor; and the predicted pK_a values for molecules related to those used in the parametrization are fairly accurate. In addition, Epik attempts to make consistent estimates for the uncertainties of the pK_a values so that the user may elect to seek additional information for uncertain results. Sources of such additional information include experiments on suitable selected model compounds or theoretical estimates of the pK_a values from other programs such as Schrödinger's pK_a predictor, which is part of Jaguar.

Epik only designates the protonation state of the output structure from calculations as "conjugate acid" or "conjugate base" in the log file, indicating whether the proton is present or not. These designations adhere strictly to the IUPAC definitions of Brønsted acid and Brønsted base, but do not necessarily map in a simple manner to common names for functional groups that chemists sometimes use, that depend on whether the compound is protonated when uncharged (often called an "acid", as in "carboxylic acid") or deprotonated when uncharged (often called a "base", as in "Schiff base"). In general, mapping Epik's pK_a values with such acid/base designations based upon the contents of the log file will result in mismatches and inaccuracies and thus should not be done.

1.2.2 Structure adjustment

In structure adjustment, both the tautomerization and ionization state of the structure may be modified.

Tautomerization is carried out in the same manner, and using the same data, as the `tautomerizer` tool in LigPrep. SMARTS-like patterns from a tautomer database are used to identify and describe how to transform one tautomer into another. Each tautomeric form in the database is assigned a probability based upon experimental data or, more typically, ab initio quantum mechanical calculations.

In structure adjustment mode Epik adds or removes protons and adjusts the tautomeric state to generate structures that are most probable within a given pH range. We will somewhat loosely refer to this as an ensemble of protonic states consistent with the conditions specified. Since ionization and tautomerization are inter-related the construction of the ensemble is iterative. In each cycle of the iteration all structures accumulated so far are subjected to tautomerization, and then the resulting tautomers are ionized. Structure selection occurs at the end of tautomerization where only the more probable tautomers for each structure are retained; and at the end of the ionization stage where only those species whose overall probability within the currently accumulated ensemble are high enough are retained. When the collection undergoes no change during an iteration the process is judged to be complete. This iteration process can lead to superior results for molecules that can undergo both ionization and tautomerization, particularly if the tautomeric preference varies with the ionization state. At the end of the process the pK_a values are estimated for each functional group present. Each structure present at the end of the iteration process is assigned a probability and a number of penalties based upon: the tautomerizations and ionizations needed to generate it, the collection of molecules generated, and the desired pH. These properties should be useful in penalizing less-probable forms in downstream processing of the structures produced.

Since a number of tautomerizations and ionizations are attempted for a typical ligand-like molecule, and multiple output structures may be produced, the adjustment of the ionization state is considerably more computationally intensive than just estimating the pK_a values for the input structures. The typical time taken is, however, only a few seconds per molecule, but structures that contain many tautomerization and ionization sites can take of the order of a minute to adjust.

If the metal binding option is used, Epik adds to the normal structural adjustment procedure a stage that generates more structures suitable for use with metalloproteins. During this stage, each structure that is generated by normal processing is examined for the presence of heavy atoms that are known to interact with metals. Each of these heavy atoms is considered separately. Its pK_a value is shifted down by an amount that is specific to the functional group (typically 3.0 pK_a units), to arrive at an effective pK_a. If that atom was protonated and the new effective pK_a value falls below the upper limit of the pH range, the proton is removed and the new structure is added to the output structure file. Penalties using the normal and shifted pK_a values are recorded in the output structures.

1.2.3 Sequential pK_a estimation

Sequential pK_a estimation involves three stages:

1. Adjust the structure, as described in the previous section, to produce what Epik regards as the most probable form for the molecule at a specific pH (usually 7), referred to as the pH-adjusted-structure.
2. Sequentially remove the most acidic proton starting from the pH-adjusted-structure. After the removal of each proton the pK_a values are recalculated.
3. Sequentially add a proton to the most basic non-hydrogen atoms, starting form the pH-adjusted-structure. After each proton addition the pK_a values are recalculated.

In stages 2 and 3 the pK_a values of the protons removed or added are noted and recorded in both the output structure file and the log file. These are microscopic pK_a values. Some types of experiments, such as titrations, measure macroscopic pK_a values. When two or more microscopic pK_a values lie with 1 pK_a unit of each other, the macroscopic pK_a values can noticeably differ from the corresponding microscopic ones.

The sequential pK_a estimation mechanism is the most appropriate mechanism for comparing Epik's pK_a estimates with experimental values.

To facilitate the comparison of Epik's sequentially predicted pK_a values with experimental pK_a values, a script, `compare_epik_results.py`, has been provided. See Appendix D for more information.

1.3 Installation and Licensing

Instructions for installing Schrödinger software can be found in the *Installation Guide*.

Epik is licensed separately from other Schrödinger products. While Epik can be used as part of a LigPrep run, it still requires an Epik license when used in this manner.

1.4 Running Schrödinger Software

Schrödinger applications can be run from a graphical interface or from the command line. The software writes input and output files to a directory (folder) which is termed the *working directory*. If you run applications from the command line, the directory from which you run the application is the working directory for the job.

Linux:

To run any Schrödinger program on a Linux platform, or start a Schrödinger job on a remote host from a Linux platform, you must first set the `SCHRODINGER` environment variable to the installation directory for your Schrödinger software. To set this variable, enter the following command at a shell prompt:

csh/tcsh:	`setenv SCHRODINGER` *installation-directory*
bash/ksh:	`export SCHRODINGER=`*installation-directory*

Once you have set the `SCHRODINGER` environment variable, you can run programs and utilities with the following commands:

```
$SCHRODINGER/program &
$SCHRODINGER/utilities/utility &
```

You can start the Maestro interface with the following command:

```
$SCHRODINGER/maestro &
```

It is usually a good idea to change to the desired working directory before starting the Maestro interface. This directory then becomes the working directory.

Windows:

The primary way of running Schrödinger applications on a Windows platform is from a graphical interface. To start the Maestro interface, double-click on the Maestro icon, on a Maestro project, or on a structure file; or choose Start → All Programs → Schrodinger-2015-2 → Maestro. You do not need to make any settings before starting Maestro or running programs. The default working directory is the Schrodinger folder in your Documents folder.

If you want to run applications from the command line, you can do so in one of the shells that are provided with the installation and have the Schrödinger environment set up:

- Schrödinger Command Prompt—DOS shell.
- Schrödinger Power Shell—Windows Power Shell (if available).

You can open these shells from Start → All Programs → Schrodinger-2015-2. You do not need to include the path to a program or utility when you type the command to run it. If you want access to Unix-style utilities (such as `awk`, `grep`, and `sed`), preface the commands with `sh`, or type `sh` in either of these shells to start a Unix-style shell.

Mac:

The primary way of running Schrödinger software on a Mac is from a graphical interface. To start the Maestro interface, click its icon on the dock. If there is no Maestro icon on the dock, you can put one there by dragging it from the SchrodingerSuite2015-2 folder in your Applications folder. This folder contains icons for all the available interfaces. The default working directory is the Schrodinger folder in your Documents folder (`$HOME/Documents/Schrodinger`).

Running software from the command line is similar to Linux—open a terminal window and run the program. You can also start Maestro from the command line in the same way as on Linux. The default working directory is then the directory from which you start Maestro. You do not need to set the `SCHRODINGER` environment variable, as this is set in your default environment on installation. To set other variables, on OS X 10.7 use the command

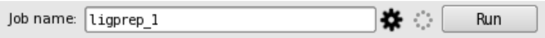
```
defaults write ~/.MacOSX/environment variable "value"
```

and on OS X 10.8, 10.9, and 10.10 use the command

```
launchctl setenv variable "value"
```

1.5 Starting Jobs from the Maestro Interface

To run a job from the Maestro interface, you open a panel from one of the menus (e.g. Tasks), make settings, and then submit the job to a host or a queueing system for execution. The panel settings are described in the help topics and in the user manuals. When you have finished making settings, you can use the Job toolbar to start the job.



Job name: ligprep_1

You can start a job immediately by clicking Run. The job is run on the currently selected host with the current job settings and the job name in the Job name text box. If you want to change the job name, you can edit it in the text box before starting the job. Details of the job settings are reported in the status bar, which is below the Job toolbar.

If you want to change the job settings, such as the host on which to run the job and the number of processors to use, click the Settings button. (You can also click the arrow next to the button and choose Job Settings from the menu that is displayed.)

⚙

You can then make the settings in the Job Settings dialog box, and choose to just save the settings by clicking OK, or save the settings and start the job by clicking Run. These settings apply only to jobs that are started from the current panel.

If you want to save the input files for the job but not run it, click the Settings button and choose Write. A dialog box opens in which you can provide the job name, which is used to name the files. The files are written to the current working directory.

The Settings button also allows you to change the panel settings. You can choose Read, to read settings from an input file for the job and apply them to the panel, or you can choose Reset Panel to reset all the panel settings to their default values.

You can also set preferences for all jobs and how the interface interacts with the job at various stages. This is done in the Preferences panel, which you can open at the Jobs section by choosing Preferences from the Settings button menu.

Note: The items present on the Settings menu can vary with the application. The descriptions above cover all of the items.

The icon on the Job Status button shows the status of jobs for the application that belong to the current project. It starts spinning when the first job is successfully launched, and stops spinning when the last job finishes. It changes to an exclamation point if a job is not launched successfully.

◌

Clicking the button shows a small job status window that lists the job name and status for all active jobs submitted for the application from the current project, and a summary message at the bottom. The rows are colored according to the status: yellow for submitted, green for launched, running, or finished, red for incorporated, died, or killed. You can double-click on a row to open the Monitor panel and monitor the job, or click the Monitor button to open the Monitor panel and close the job status window. The job status is updated while the window is open. If a job finishes while the window is open, the job remains displayed but with the new status. Click anywhere outside the window to close it.

Jobs are run under the Job Control facility, which manages the details of starting the job, transferring files, checking on status, and so on. For more information about this facility and how it operates, as well as details of the Job Settings dialog box, see the *Job Control Guide*.

1.6 Other Utilities

A number of utilities that are provided with Epik, but are not part of Epik itself, might be useful in conjunction with Epik. These utilities are available in `$SCHRODINGER/utilities`, and include:

- `proplister`—extracts requested properties from Maestro-formatted files
- `ligfilter`—selects structures from Maestro-formatted files using the property values stored in the files (supersedes `propfilter`)
- `pdbconvert`—converts files between Maestro and PDB formats
- `maesubset`—selects a subset of the structures in a Maestro-formatted file based on structure order
- `sdconvert`—converts files between Maestro and SD formats
- `sdsubset`—selects a subset of the structures in an SD-formatted file
- `structconvert`—converts structure files between various formats

More information on these utilities is available in the *General Utilities* manual.

1.7 Citing Epik in Publications

The use of this product should be acknowledged in publications as:

Epik, version 3.2, Schrödinger, LLC, New York, NY, 2015.

A reference to the following papers should also be used.

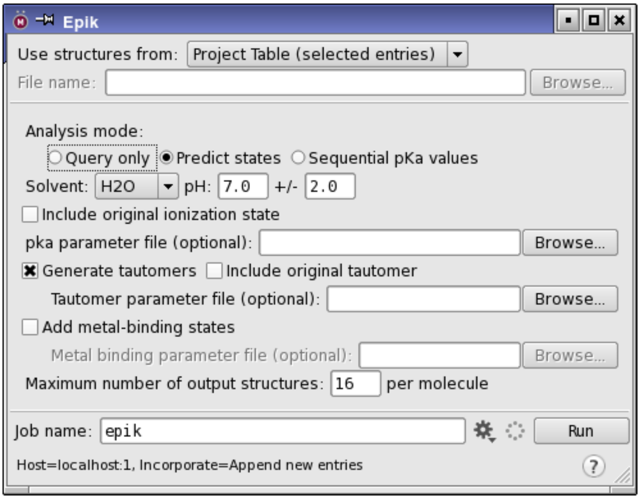
- Shelley, J. C.; Cholleti, A.; Frye, L. L.; Greenwood, J. R.; Timlin M. R.; Uchiyama, M. Epik: a software program for pKa prediction and protonation state generation for drug-like molecules *J. Comput. Aided Mol. Des.*, **2007**, *21*, 681–691.
- Greenwood, J. R.; Calkins, D.; Sullivan, A. P.; Shelley, J. C. Towards the comprehensive, rapid, and accurate prediction of the favorable tautomeric states of drug-like molecules in aqueous solution. *J. Comput. Aided Mol. Des.*, **2010**, *24*, 591–604.

Please note that the pK_a and tautomeric databases provided with Epik are copyrighted material, and should not be extracted, reproduced, or used outside of the context of Epik or LigPrep licensed calculations.

Running Epik

2.1 Running Epik from Maestro

Epik jobs can be submitted from the Epik panel in Maestro. To open the Epik panel, choose Epik from the Applications menu in the main window.



Epik

Use structures from: Project Table (selected entries)

File name: Browse...

Analysis mode:

○ Query only ● Predict states ○ Sequential pKa values

Solvent: H2O pH: 7.0 +/- 2.0

□ Include original ionization state

pka parameter file (optional): Browse...

☒ Generate tautomers □ Include original tautomer

Tautomer parameter file (optional): Browse...

□ Add metal-binding states

Metal binding parameter file (optional): Browse...

Maximum number of output structures: 16 per molecule

Job name: epik Run

Host=localhost:1, Incorporate=Append new entries

Figure 2.1. The Epik panel.

The tasks available from the Epik panel are:

- Prediction of the pK_a values of the ionizable atoms in a set of structures, either for independent or for successive protonation or deprotonation
- Generation of the probable ionized (and tautomerized) states of a set of structures within a given pH range.
- Sequential pK_a estimation.

The pK_a predictions are rule-based, and so can be generated very rapidly. The pK_a values and their uncertainties are stored as atomic properties with the structure.

When the Epik job finishes, the pK_a values are automatically displayed on the structures in the Workspace as atom labels. Previously calculated pK_a values can be viewed using the Atom Labels panel. The labels can also be cleared in the Atom Labels panel.

To predict the pKa of existing structures:

1. Select the source of the structures using the tools at the top of the panel.
2. Select Query only for the Analysis mode.
3. Choose the solvent from the Solvent option menu.
4. Click the Settings button.
5. Set job parameters in the Job Settings dialog box, and click Run.

To generate the ionization states of structures within a given pH range:

1. Select the source of the structures using the tools at the top of the panel.
2. Select Predict states for the Analysis mode.
3. Choose the solvent from the Solvent option menu.
4. Enter the target pH and range in the pH text boxes.

   The range is converted into a probability of 10^{-range}. Ionized (and tautomerized) structures whose probability exceeds this value, when all likely structures are considered, are kept. If the structure does not tautomerize, this is equivalent to keeping structures whose groups are ionized if their pK_a lies within the specified range of the target pH value.
5. If you want to keep the original ionization state, regardless of its probability in the given pH range, select Include original ionization state.
6. Select Generate tautomers to tautomerize structures during structural adjustment.
7. If you selected Generate tautomers and want to keep the original tautomer, regardless of its probability in the given pH range, select Include original tautomer.
8. Select Add metal-binding states if you want to generate states that are appropriate for binding to metals in a protein binding site.
9. Enter the maximum number of structures to generate for each input structure in the Maximum number of output structures text box.
10. Click the Settings button.
11. Set job parameters in the Job Settings dialog box, and click Run.

To generate sequential pKa values:

1. Select the source of the structures using the tools at the top of the panel.
2. Select Sequential pKa values for the Analysis mode.
3. Choose the solvent from the Solvent option menu.
4. Enter the target pH in the pH text box.

   This value is used to generate the most probable structure at the target pH as a reference for the pK_a calculations. The range text box is not available.
5. Select Generate tautomers to tautomerize structures during structural adjustment.
6. Click the Settings button.
7. Set job parameters in the Job Settings dialog box, and click Run.

For large numbers of structures, you can distribute the Epik calculation over multiple processors. This choice can be made in the Job Settings dialog box.

Epik settings can be stored in a command file, which has the extension `.in`. You can write out the command file by choosing Write from the Settings button menu, which opens a file selector so you can navigate to a location and name the file. You can read settings back in to the panel by choosing Read from the Settings button menu, and navigating to a command file in the file selector that opens. The panel is set up according to the settings in the command file.

When you write a command file, the structure file is also written, with the same stem as the command file, but with a `.maegz` extension. This makes it easy to run jobs from the command line, as described in the next section.

2.2 Running Epik from the Command Line

Epik can be run using the `epik` command. The syntax for this command is shown below.

```
epik [options] -imae input-file -omae output-file
```

The input and output files are required and must be specified using the `-imae` and `-omae` options respectively. Only Maestro-formatted files are supported. Input and output files can be in uncompressed (`.mae`) or compressed (`.maegz` or `.mae.gz`) form. The input file can contain more than one structure. The log file (output) contains information on the processing of the structures, including notes on any problems encountered.

For information on the `epik` command options, run the command with the `-h` option. The `epik` command supports the standard and optional job level and diagnostic options.

Epik jobs run in the background under Schrödinger's Job Control facility unless the `-NO_JOBCONTROL` or `-NOJOBID` options are given. Job Control runs the `epik` command in a temporary directory where up-to-date files are maintained while the job is running. When the job finishes, the output files are copied back to the job submission directory. While the job is running the `.log` file is monitored. That is, an up-to-date copy of this file is maintained in the job submission directory. Specifying the `-LOCAL` option overrides the use of a temporary directory and an attempt is made to run out of the job submission directory. For more information on the Job Control facility, see the *Job Control Guide*.

The `-ph` Option

- Unless `-ph` is specified, pK_a values are estimated using the input structure.
- If `-ph` is specified, variations on the input structure are generated, and the pK_a values for each of them is calculated. If no structure meets the criteria for output, the most probable structure is kept so that at least one structure is always produced for each input structure.

Notes on Retention Behavior

- If you forcibly retain structures, they are selected for inclusion in the output ahead of any more probable structures.
- The option `-retain_i` forces the retention of the exact combination of the input tautomerization and ionization state.
- The option `-retain_t` forces retention of the various ionized forms of the input tautomerization state based upon their ionization probabilities. At a minimum the most probable ionization state of the input tautomeric form is retained.
- If you specify both `-retain_i` and `-retain_t`, a combination of the behaviors occurs. A copy of the exact combination of the input tautomerization and ionization state plus probable ionization states of the input tautomer are retained. As well, variations on the ionization state for the input tautomer will be prioritized based upon the ionization penalties only.

2.2.1 Epik Command Line Examples

The Epik panel in Maestro is a convenient and general tool for running Epik, particularly for small batches of molecules. Epik can also be run from the command line, and thus run from user-written scripts.

- The default behavior of Epik is to find pK_a values for the input structures. To run this process from the command line, only the input and output file names need to be specified:

  ```
  epik -imae ligands.mae -omae ligands_with_pKas.mae
  ```

The output structure file will contain the same structures that were present in the input file, with atom level properties for the pK_a values. For acids, the pK_a values are associated with the acid hydrogens; while for bases, the pK_a values are associated with the atoms to which the proton would bond. Note that the pK_a values are for the input structure as given, regardless of how suitable the input structures actually are.

- To adjust the ionization and tautomerization state of the input molecules and predict probable forms at a specific pH, use the following command (replacing the pH value with your own choice):

  ```
  epik -ph 7.0 -imae epik_input.mae -omae epik_prob_forms.mae
  ```

  The output structure file will contain predicted tautomers and ionized forms of the input molecules with a population greater than 0.01 at pH 7.0, with one or more structures corresponding to each input structure. Each output structure includes atom level properties for the pK_a value of that structure and four structure level properties that describe aspects of the overall likelihood for that structure existing. For more information on these properties, and how structural adjustment is performed in general, see Chapter 4.

- To produce forms of the ligand appropriate for binding to metalloproteins in addition to those generated in the previous example, use the following command:

  ```
  epik -ph 7.0 -metal_binding -imae epik_input.mae
      -omae epik_prob_plus_metal_forms.mae
  ```

  In addition to the structures produced in the previous example, the output structure file will contain structures that are generated by deprotonating specific heavy atoms that are known to interact with metals in metalloproteins. In addition to the normal structural adjustment properties, properties that indicate whether the structure is a normal structure or was generated for metal binding, and a second set of properties for describing the likelihood of the metal-bound forms are added to each output structure. For more information on these properties, see Chapter 4.

- Epik produces at least one output structure for each input structure, but occasionally the specific form in the input file may be deemed improbable and thus not appear in the output structure file. If you want to create alternate versions of the input structures, yet still retain the original forms, the following command can be used:

  ```
  epik -retain_i -ph 7.0 -imae epik_input.mae
      -omae epik_retain_prob.mae
  ```

  The original ionization and tautomerization form for each input structure are kept, and additional forms that Epik predicts with a population greater than 0.01 are generated.

- To reduce the number of output structures produced, there are a number of options to choose from. For instance, the command:

  ```
  epik -ph 7.0 -nt -p 0.1 -ms 2 -imae epik_input.mae
      -omae epik_2_ions.mae
  ```

  instructs Epik to skip tautomerization (`-nt`) and produce at most two (`-ms 2`) ionized forms for each input structure that have a probability greater than 0.1 (`-p`) at pH 7.0. If no ionized form has this high a probability, then the most probable ionized form is saved to the output file.

  To predict only the most probable form at pH 6.0, the following command can be used:

  ```
  epik -ph 6.0 -ms 1 -imae epik_input.mae
      -omae epik_one_form_ph_6.mae
  ```

- Epik can be used to estimate the microscopic pK_a values obtained by sequentially adding and removing protons using the following command:

  ```
  epik -ph 7.0 -scan -imae epik_input.mae -omae epik_scan.mae
  ```

  For each structure in the input file, the log file lists the estimated microscopic pK_a values in order, starting from the lowest pK_a value. The output Maestro file contains the structure that Epik considers the most probable at pH 7.0 for input structures. The sequential pK_a values and the base atoms involved are included as properties for each output structure.

2.2.2 Distributing Epik Jobs

If you are running `epik` on many ligands, you can reduce the turnaround time by automatically distributing the calculation over multiple subjobs The `epik` command divides the collection of input structures into a specified number of sets, each of which is run in a single subjob. You can use either `-NJOBS` *subjobs* or `-JOBCTS` *maxperjob* to divide up the structures, but if the files for each subjob are larger than the *maxperjob* limit, the number of subjobs is increased. The number of processors used is set with the `-HOST` option (if you want to run the driver job remotely) or the `-SUBHOST` option.

2.2.3 Epik Command File

You can run Epik with a command file containing commands for the settings, rather than using command-line options. Values given on the command line override values in the command file. This file is in keyword-value format; the keywords are described in Table 2.1.

Table 2.1. Keywords for the Epik command file

Keyword	Description
`add_metal_binding_states`	Generate additional states appropriate for binding to metal ions in protein binding sites.
`analysis_mode` *mode*	Specify the kind of job to run. Allowed values: `predict` Predict ionization states as well as pKa values. This is the default. `query` Query only job to predict pKa values `scan` Sequential pKa prediction (scan)
`cg [`*rdiff rms*`]`	Use geometry as well as connectivity when identifying unique structures. Normally only connectivity is used. Structures are distinct if: *rdiff* > 0.0 and any position differs by more than *rdiff* *rms* > 0.0 and the rms difference in positions is larger than *rms*
`estimate_acids_only`	Estimate pK_a values for acids only.
`estimate_bases_only`	Estimate pK_a values for bases only.
`generate_tautomers`	Tautomerize structures.
`highest_pka` *number*	Set the highest pK_a to estimate when sequentially ionizing to *number*.
`include_orig_ionization_state`	Retain the initial ionization state. Also retains the initial tautomerization state.
`include_orig_tautomer_state`	Retain the initial tautomerization state.
`input_file_name` *input-file*	Name of Maestro-format input file.
`lowest_pka` *number*	Set the lowest pK_a to estimate when sequentially ionizing to *number*.
`max_atom` *number*	Structures containing more than *number* atoms will not be adjusted. Default: 150.
`max_output_str` *number*	Maximum number of generated structures per input structure. Default: 32.
`max_tautomers` *value*	Maximum number of tautomers. Default is 8.
`metal_binding_param_file` *filename*	Use the specified file for describing metal binding adjustments to pK_a values.
`min_probability` *value*	The minimum probability at the target pH for generated states to be kept. Probabilities are evaluated on the basis of an ensemble of likely states, as determined from ionization and tautomerization equilibria. Default: 0.01.

Table 2.1. Keywords for the Epik command file (Continued)

Keyword	Description
`min_tautomer_probability` *value*	Minimum probability for tautomers.
`njobs` *njobs*	Divide the overall job into *njobs* subjobs.
`output_file_name` *output-file*	Name of Maestro-format output file.
`ph` *value*	Target pH for generated states. Default: 7.
`ph_tolerance` *value*	pH tolerance for generated structures. The minimum probability for generated states is determined from $-\log_{10}(p) = value$. When tautomerization is disallowed, this is equivalent to keeping structures whose pK_a value lies within *value* units of the target pH value. No default.
`pka_atom`	Report pK_a of specific atoms as a Maestro property.
`pka_param_file` *filename*	Use the specified custom pK_a parameter file. Default: `pKa_water_HT_data`.
`retain_i_lab`	Similar to `-retain_i` except that only labeled input structures are retained.
`retain_t_lab`	Similar to `-retain_t` except that only labeled input structures are retained.
`solvent` *solvent*	The name of the solvent to use. Allowed values are `water` and `DMSO`. Default: `water`.
`tautomer_param_file` *filename*	Use the specified custom tautomer database file.
`verbose` *number*	controls the level of reporting. 0 `MMERR_FATAL` - only report errors 1 `MMERR_WARNING` - only report errors and warnings 2 `MMERR_INFO` - report errors, warnings, and processing information. 3 `MMERR_DEBUG` - report errors, warnings, processing information, and debugging information

2.3 Epik Limitations

Epik predicts pK_a values and generates ionization and tautomerization states for typical ligand-like organic molecules in a rapid and practical manner. Thus, some constraints are imposed on certain classes of systems, which Epik may not process as well as others.

- **Molecular representation:** Epik requires all-atom input structures to have the hydrogen atoms explicitly specified. While most of Epik's calculation facilities are coordinate-independent, some can depend on input geometry. Epik is expected to function acceptably with well-chosen 2D or 3D coordinates.
- **Molecule size limitation:** Epik's default molecular size setting does not adjust the structures of molecules larger than 150 atoms, which are atypically large for ligands. Molecules larger than 150 atoms may cause delays in processing times, due to the increase in the number of possible variations generated. The `-ma` option allows the adjustment of this setting to accommodate larger molecules.
- **Target molecule classes:** Epik is intended to function well and rapidly for typical organic ligand-like molecules. Many of the calculations performed in Epik use precalculated information that has been tabulated for fast lookup during the calculation. While the tabulations are extensive, they are not exhaustive, and some chemical functionalities may not have sufficient information for accurate treatment.
- **pK_a range:** Epik is intended to reliably predict the pK_a values for ionizable groups whose ionization state may change under the range of pH values most relevant for medicinal chemistry. For water this range is 4 to 10. For DMSO the range is less well defined but is approximately 4 to 30. To prevent protonation of atoms not normally regarded as basic due to simple, general, yet inappropriate matches for these atoms, more specific matches are used to assign very low, often negative pK_a values.

Epik Methodology

Empirical prediction of acid and base pK_a values for organic molecules has a long and largely successful history. The combination of two closely related linear free-energy approaches based upon the Hammett equation for aromatic molecules and the Taft equation for aliphatic molecules has been adopted for use in Epik to predict the pK_a values of organic acids and bases. The implementation used largely follows that described in *pK_a prediction for Organic Acids and Bases* (Perrin, D.D.; Dempsey, B.; and Serjeant, E.P.; Chapman and Hall, London (**1981**)). This approach is briefly and functionally described in the next section. Schrödinger's tautomerization methodology is described in Section 3.2. General information on Epik's standard parameter set is available in Section 3.3, while information on customizing the parametrization is available in Section 3.4.

3.1 pK_a Prediction

3.1.1 Overview of Hammett and Taft Prediction of pK_a Values

Hammett and Taft equations are intended to predict microscopic pK_a values. That is, given a molecular protonation state, what are the pK_a values for the first addition or removal of a proton from the various ionizable functional groups? The same pK_a prediction for a functional group applies to both the acidic and conjugate base forms. Both the Hammett and the Taft equations have the same general form:

$$\mathrm{p}K_a = \mathrm{p}K_a^{\ 0} + \mathrm{CF} - \sum_i \rho_i \sum_i \sigma_{i,j} \tag{1}$$

Ionizable function groups are recognized using SMARTS patterns. Each SMARTS pattern has a p$K_a^{\ 0}$ and ρ value associated with it. p$K_a^{\ 0}$ values describe the unperturbed pK_a value for the ionizable group and ρ values describe the sensitivity of the ionizable group to substituents attached to the ionizable group at particular locations. Often multiple SMARTS patterns for ionizable groups, each with different p$K_a^{\ 0}$ and ρ values, will match the same atoms within a molecule. The most appropriate or primary one is selected using priorities explicitly encoded in Epik's database. The ρ value for the primary match is used unless it cannot describe a particular substitution location, in which case the ρ value for another of the matching SMARTS patterns for that functional group is used.

The perturbing influences of most structural features, most commonly substituents, are described by σ values. A description of the method of selection and estimation of the various σ values is found in Section 3.1.2. The remaining adjustments are described by a general correction factor, CF.

$$\mathrm{CF} = -\log_{10}(n_{HR}/n_{HA}) + \mathrm{RA} \qquad (2)$$

The first term in Equation (2) accounts for the number of ways to remove equivalent H atoms from an acidic molecule (n_{HR}) versus the number of equivalent ways to add an H atom to the conjugate base (n_{HA}). The second term, RA, is an empirical ring adjustment term for aliphatic rings. For instance, a value for RA of roughly 0.2 is often used for ionizable amine atoms in a single aliphatic ring, such as morpholine.

Each pair of pK_a^0 and ρ values has an uncertainty associated with it that reflects the standard deviation versus experiment in the predictions given by that match. If insufficient data is available to determine an uncertainty, a default value of 2.0 pK_a units is used. Functional groups that lie outside Epik's parametrization coverage may be matched with unsuitable Hammett or Taft parameters, resulting in predictions whose accuracy lies well outside the uncertainties given.

3.1.2 ρ Values

Each `acid_base` block's ρ value is intended to describe the perturbations introduced by adding substituents to specific atoms, enumerated within the block using the `subs_atoms` designator. The ρ value from the primary match is first used for substituents attached to atoms explicitly listed in the `subs_atoms` lists for the primary match or any atoms in aromatic ring systems that are at least partially included in the `acid_base` SMARTS pattern.

In addition, the primary ρ value is used for non-substituent corrections such as:

- heteroaromatic atoms (non-carbon aromatic atoms) that are one of:
  i. non-carbon atoms that match general aromatic types in the SMARTS patterns ([a]) for the `acid_base` and are listed as `hetero_aromatic` in the primary match `acid_base` block.
  ii. atoms in portions of aromatic rings that are not listed in the SMARTS pattern that are part of a larger aromatic ring system that is at least partially listed in the SMARTS pattern for the primary match `acid_base` (e.g. a fused ring system, but not another ring singly-bonded to the primary aromatic ring).
- aromatic topological corrections:

  Corrections for known topological patterns in polycyclic aromatic ring systems for specific functional groups (e.g., those listed in Table 7.1 of Perrin et al.).

If the primary match has an aromatic ring that is fused to additional aromatic rings that are not contained within the primary match, the ρ from the primary match is used for attachments to the extended aromatic ring system.

If the atoms in the primary match have substituents that have not been covered by the primary match directly or by extending aromatic ring systems, other `acid_base` matches for this same functional group are examined to see if they have suitable explicit substituent locations. If so, the ρ values for the other patterns are used in Equation (1) for these contributions.

If there are still substituents that do not have a ρ value associated with them then the `acid_base` patterns are re-examined to see if they have substitution locations within the primary match that lie between the first atom in the substituent and the base atom. If more than one eligible `acid_base` pattern is found, the one with a substitution location closest to the first substituent atom is selected. If such locations are identified their ρ values are used to describe the contributions for these substituents and transmission effects are taken into account (see Section 3.1.3).

If all of the above criteria fail, then a general methodology for selecting a ρ value is used. The path in general will not involve aromatic atoms so the formula given by Perrin for aliphatic ionizable groups is used:

$$\rho = 0.8 \times 2^{2-h} \tag{3}$$

where h is the number of atoms between the substituent and the base atom.

3.1.3 σ Values for Substituents

In addition to the σ values for heteroaromatic atoms and topological patterns for rings mentioned earlier, σ terms come from substituents. The perturbing influence of a substituent depends on whether it is connected to an aliphatic portion or an aromatic portion of the `acid_base` pattern in use. Table 3.1 lists the σ types that may be given for a substituent. These σ types, along with a SMARTS pattern describing the chemical nature of the substituent group, are given in a substituent structure in the solvent database (see Appendix A).

We defer to the viewpoint expressed by Perrin that `special_sigma_para` constants can be used to describe how `sigma_para` values can vary depending on the ionizing group and do not support the `sigma_para`$^-$ formalism sometimes used in Hammett equations.

In general, parameterized `sigma_ortho` values are uncommon and often depend on the ionizable group in question, so `special_sigma_ortho` group values are not unusual.

Table 3.1. σ types for substituents

Type	Used for
sigma_star	most aliphatic attachment locations
sigma_ortho	on aromatic rings where the substituent is ortho to the ionizable location
sigma_meta	on aromatic rings where the substituent is meta to the ionizable location
sigma_para	on aromatic rings where the substituent is para to the ionizable location
special_sigma_ortho	similar to sigma_ortho but only for a particular ionizable group
special_sigma_meta	similar to sigma_meta but only for a particular ionizable group
special_sigma_para	similar to sigma_para but only for a particular ionizable group

For some unsaturated aliphatic systems, `sigma_ortho`, `sigma_meta`, or `sigma_para` (or combinations thereof) are used if the underlying Taft parameterization was performed using them rather than σ^* exclusively. If so, the non-zero weights for these various σ types are indicated in the `acid_base` block using explicit `sigma_star_wt`, `sigma_ortho_wt`, `sigma_meta_wt` or `sigma_para_wt` identifiers (see Appendix A).

In addition to sometimes having their own σ types, non-carbon atoms in five-membered aromatic rings are conceptually replaced by two aromatic carbon atoms creating a virtual six-membered ring. Determination of the ortho, meta, or para designations for substituents and hetero atoms within the ring relative to the ionizable group is done in this virtual ring.

In the absence of explicit `sigma_star_wt`, `sigma_ortho_wt`, `sigma_meta_wt`, or `sigma_para_wt` designations for the `acid_base` match, Epik uses the methods presented by Perrin et al. in section 7.2 to provide ways to generate σ values for aromatic ring systems if at least two types of aromatic σ values are given. For instance, in six-membered rings, if any two of the three σ values are provided, the other may be very roughly estimated (if needed) using those methods. For polyaromatic ring systems, this methodology relies on charges and geometries from template ring systems. Epik has a few of the most important ring systems encoded. The molecule’s aromatic ring system is converted into a virtual one by expanding real five-membered rings to six-membered rings when hetero atoms are present. The largest template ring system that matches the virtual one and contains the attachment location for the ionizable group is used to provide ring charges and geometry.

For substituents that are one aromatic bond or two aromatic bonds away from the ionizable group, a pure σ_o or σ_m (or their special version) is used if available. Otherwise a mixture of σ types are used (if possible) to estimate the influence of the substituent depending on the distance between the ionizable group and the substituent locations in the virtual ring system as well as the charges in the template ring system.

Epik has an extensive set of σ types for many organic substituents. When examining a substituent group at a particular location there may be more than one type of substituent description in the solvent database that matches it. When this occurs, Epik chooses the substituent description with the most atoms in its SMARTS pattern. If two such substituent descriptions have the same number of atoms, the first one in the solvent database file is used.

In some cases, even the largest matching substituent description does not cover all of the substituent atoms in that branch of the molecule. For aliphatic substitution locations, Epik can correct for the influence of these additional atoms. The methodology used for these cases involves treating these additional atoms as substituents whose influence needs to be propagated back to the substitution location for the `acid_base` group being used. The σ used in Equation (1) is given by:

$$\sigma_{\text{use}} = \sigma^{*}\tau_{\text{net}} \qquad (4)$$

σ^* is given for the additional substituent and τ_{net} is the net transmission coefficient of the atoms (k) between the substituent and the substitution location used in the `acid_base` match used for this branch of the molecule.

$$\tau_{\text{net}} = \prod_{k} \tau_k \qquad (5)$$

where τ_k is the transmission coefficient listed in the solvent parameter file for atoms like atom k.

If τ_{net} drops below 0.0025 the contribution from the substituent groups is neglected.

This same approach for transmitting the influence of a substituent across intervening atoms is also used to generate effective σ values for auxiliary `acid_base` groups whose substituent locations lie within the primary match.

The use of `sigma_ortho_wt`, `sigma_meta_wt`, or `sigma_para_wt` to force the use of ortho, meta or para parameters when they would otherwise not be used (e.g. aliphatic portions of a molecule) can create problems and should be avoided.

3.1.4 Charge Spreading

In cases where multiple mesomeric representations can reasonably be used, the molecule is typically best represented as a combination of the various mesomeric forms. An individual mesomeric form frequently provides an inaccurate description. In addition, the effective weighting of the different mesomeric forms in these combinations can also change when functional groups with atoms that are assigned formal charges in mesomeric forms are added. As such, making predictions based upon a single mesomeric form limits the range of accurate applicability of Hammett and Taft σ values.

Epik uses an empirical approach to handle mesomeric systems that involves partial charge cancellation and distribution of formal charges to produce spread formal charges. The spread formal charges are used to generate σ parameters for substituents and heteroaromatic atoms that are interpolations between the charged and uncharged versions of such functionalities.

More specifically, the net formal charge on each mesomeric group m is given by F_m. If the ionizing group contains mesomerizable sites, the formal charge and charge bias factor w_m for that site are adjusted so that they lie halfway between the acidic and basic forms for the ionizing group, ensuring that the pK_a calculations will give the same answer when starting from either form. This involves adding 0.5 to or subtracting 0.5 from F_m for this site for the acidic or basic forms of the ionizing group, respectively, and using the average of the w_m values for the acidic and basic forms of the ionizing group. In the following definitions, the sums run over all mesomeric sites for each ionizing site:

$$Q_+ = \sum_m F_m H(F_m); \qquad Q_- = \sum_m F_m H(-F_m) \tag{6}$$

$$W_+ = \sum_m w_m H(w_m); \qquad W_- = \sum_m w_m H(-w_m) \tag{7}$$

$$n_+ = \sum_m H(w_m); \qquad n_- = \sum_m H(-w_m) \tag{8}$$

where

$$H(x) = \begin{matrix} 1 & \text{if} & x > 0 \\ 0 & \text{if} & x \le 0 \end{matrix} \tag{9}$$

and the total charge is given by

$$Q_T = Q_+ + Q_- . \tag{10}$$

If the totals for both of the positive and negative formal charges, Q_+ and Q_-, are nonzero, charge cancellation is used to reduce their magnitudes. The cancellation charge C is calculated using the following scheme:

$$C = \begin{cases} -Q_- & \text{if } Q_T \geq 0 \text{ and } n_- = 0 \\ -c_f Q_- & \text{if } Q_T \geq 0 \text{ and } n_- > 0 \\ Q_+ & \text{if } Q_T < 0 \text{ and } n_+ = 0 \\ c_f Q_+ & \text{if } Q_T < 0 \text{ and } n_+ > 0 \end{cases} \qquad (11)$$

where the charge cancellation factor c_f is assigned a value of 0.84, based on empirical experience. The cancellation charge is then used to adjust the totals of the formal charges:

$$Q'_+ = Q_+ - C; \qquad Q'_- = Q_- - C \qquad (12)$$

which are in turn used to calculate the spread formal charges f_m on each of the mesomerization sites:

$$f_m = \begin{cases} Q'_+ w_m / W_+ & \text{if } w_m \geq 0 \\ Q'_- w_m / W_- & \text{if } w_m < 0 \end{cases} \qquad (13)$$

The σ values used in the Hammett equation or Taft equation for each mesomeric site, excluding those within the ionizable group itself, is given by

$$\sigma_m = (1 - r_m)\sigma_{m\alpha} + r_m \sigma_{m\beta} \qquad (14)$$

where α is the largest integer less than f_m, $\beta = \alpha + 1$, and $r_m = f_m - \alpha$. $\sigma_{m\alpha}$ and $\sigma_{m\beta}$ are the σ values for the functional group when it possesses formal charges of α and β, respectively.

3.2 Tautomerization

Tautomers are an important class of isomers that can interconvert under physiological conditions. Tautomeric forms have different chemical properties and interact differently. For instance, one tautomeric form may interact with the active site of a protein more strongly than the other forms. Therefore, for some types of calculations, such as docking with Glide, considering the appropriate tautomeric forms of ligands can be important.

There does not seem to be a universal definition for tautomers. In Epik, tautomers are defined as isomers that meet the following conditions:

- In aqueous solution tautomers interconvert rapidly enough to be present as mixtures.

- One or more hydrogen atoms are bound to different atoms and the orders of one or more of the bonds between non-hydrogen atoms differs between tautomers.
- The non-hydrogen atom topology of the structure does not change during these interconversions.

Epik's definition of tautomers therefore excludes the aldose-hemiacetal ring opening and closing equilibrium in sugars that are sometimes regarded as tautomerizations. An example of the well-known keto-enol tautomerization is shown in Figure 3.1.

The tautomerization facility of Epik is intended to generate tautomers for input structures in a practical and flexible manner. It relies on a database of tautomeric templates to guide it in generating tautomers. Flexibility is provided in the command-line options and by permitting users to modify the database. For more information on this database, see Appendix B.

The tautomerization facility is not intended to generate all possible tautomers. The collection of groups of tautomers in the database is not exhaustive. As well, each set of tautomeric forms in the database is limited to those tautomers that are expected to have significant populations in aqueous solution, rather than being a comprehensive list. Tautomers in the database are assigned probabilities to assist in focusing on the most highly populated tautomeric forms.

The tautomerization of each structure can be divided into two stages: tautomer pattern matching and structure generation. These stages are described in the following sections.

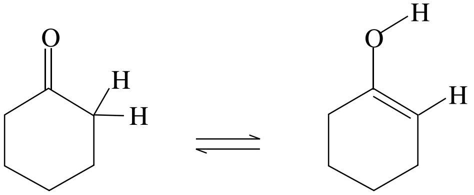


Figure 3.1. Keto-enol tautomerization.

3.2.1 Tautomer Pattern Matching

The current structure is examined for all matches of all tautomeric patterns in the tautomer database. All matches are cross-examined to see if the atoms are completely contained within another match. If so, the match containing the smaller number of atoms is eliminated. If they are the same size, then the match pattern found first is retained. This is the only case where the order of tautomers within a tautomer set or the order of the tautomer sets within the tautomer database affects processing.

3.2.2 Structure Generation

For each match, all tautomers in the tautomer set should be considered. Structures in which more than one match has been found have multiple locations in the structure that can be tautomerized, and all combinations of tautomers that are compatible should be considered. Double bonds that can shift to single bonds in any tautomer within a set can shift between E and Z forms, provided that other topological constraints do not prevent this shift.

The `tautomerizer` tries to generate all compatible combinations of all tautomers for each match. For each structure having a high enough probability, all combinations of 180 degree rotations about certain double bonds within each tautomeric pattern are generated. Double bonds are rotated if they involve C or N atoms in which both ends are not attached to two H atoms and in which the two atoms in the bond do not reside in the same ring. No adjustment is made to the probability for the tautomeric form for these double bond rotations. The probability for each structure resulting from this process is estimated as the product of the probabilities for all the tautomer templates used to generate it. The probabilities are normalized amongst all the structures generated for a given input structure.

Tautomers are generated in order of decreasing probability. By default, the maximum number of tautomers generated internally is 128. Of this collection of tautomers, up to eight tautomers are recorded by default in the output file for each input structure. This limit can be adjusted with the `-tn` option of the `epik` command. The most probable tautomer is always recorded. Except for this tautomer, only tautomers with a probability higher than 0.01 are recorded in the output structure file. This threshold value can be adjusted using the `-tp` option of the `epik` command. If any of the `-retain` options is used to keep a tautomeric form of a molecule that would normally have a probability lower than the threshold probability value, the probability for this form is set to the threshold value.

Note: The probability for forms of a tautomerizable group is not adjusted for other functional groups inside the molecule, but outside the pattern itself. This means that these probabilities, and thus the overall probabilities for molecular forms (particularly when multiple tautomeric sites are being adjusted), are approximate and intended as a guide.

A chiral atom in a tautomer can switch its chirality if one of its tautomeric forms involves a double bond to this atom. Epik does not vary the chiralities of such atoms and does not add chirality properties for such atoms. Chiralities can be varied with the `stereoizer` utility in LigPrep—see Section 4.10 of the *LigPrep User Manual* for more information.

3.3 Standard Parameters

Epik comes with an extensive set of pK_a parameters built into the program. In addition, the parameter files `pKa_water_HT_data` and `pKa_DMSO_HT_data` in `$SCHRODINGER/mmshare-v`*version*`/data/mmpKa` may be used to update the parameter sets without needing to update the executable itself.

The tautomer database is also built into Epik. There is no standard location for tautomer definition updates. The tautomer database for DMSO is just a copy of that for water.

3.4 Custom Parameters

You can use custom parameter files to add to or override the standard parameters in Epik. The process is somewhat involved and the interaction between various data structures can be complex. You should be aware that, due mainly to the complex hierarchy involved in matching suitable patterns, changes to the standard parameters are very difficult to make without causing unwanted side effects. This process may be changed in future releases.

Custom pK_a parameters and tautomer parameters are handled separately and differently. The format for pK_a parameter specification is described in Appendix A. The format for the tautomer database is described in Appendix B.

For pK_a parameters, information can be added to the files `pKa_water_HT_data` and `pKa_DMSO_HT_data` in `$SCHRODINGER/mmshare-v`*version*`/data/mmpKa`, or to a file specified using the command line option `-es`. If the latter method is used, it is advisable to create the file starting from a copy of the appropriate file from `$SCHRODINGER/mmshare-v`*version*`/data/mmpKa`, as specifying a file on the command line causes Epik to skip reading the file located in `mmshare`. When a custom parameter file is provided using the `-es` option, the standard parameters are *not* used unless you include the `-s` option (i.e. `-s water` or `-s DMSO`).

Including the `clear_standard` data item completely eliminates pre-existing pK_a parameters from the calculations. Using the `turn_off` data block overrides individual data structures. See Appendix A for more information on the use of these data items.

Custom tautomer parameters are provided to Epik through the `-ts` command line option. The default action of the `-ts` option adds the provided parameters to Epik’s standard tautomer parameters. To clear out the standard parameters, use the `clear_standard` data item in the custom tautomer file.

Structural Adjustment in Epik

In addition to estimating the pK_a values for a given structure, Epik can adjust structures to attempt to generate a collection of structures consistent with the pH, while eliminating minimally contributing structural variations. The adjustment process involves both tautomerization and ionization.

In this chapter, we loosely use the phrase "population of a molecular structure" to mean a rough estimate for the fraction of a collection of related structures that would adopt this particular structure at equilibrium. We also use it at intermediate stages of adjusting structures for the equivalent quantity within some set of candidate structures.

Tautomerization is dependent on the ionization state of the molecule. As well, ionization depends on estimates for the pK_a of the functional group being considered for ionization, which in turn depends on the ionization states of other groups in the molecule, and of course, on the tautomeric state. Therefore, to obtain representative structures for a particular pH, these interdependencies must be taken into account. In Epik, structural adjustment is carried out iteratively. The first iteration starts with just the input structure, and subsequent iterations are applied to the collection of structures derived from the input structure in the preceding iterations. Each iteration consists of an attempt to tautomerize all structures in the current collection of molecules followed by an attempt to ionize all ionizable functional groups in each tautomer. Up to 6 iterations are attempted. This means, however, that at most 6 functional groups in the input molecule will have their ionization state adjusted. During each iteration, Epik tries to generate most tautomeric and ionization variations on the current collection of structures whose populations exceed a population threshold (*minpop*), up to a specified maximum number of structures (*ms*). If the number of structures exceeds *ms*, then the `ms` structures with the highest populations are produced. By default, *minpop* is 0.01 and is adjustable using Epik's `-p` option (or indirectly by using the `-pht` option). The default value for *ms* is 16, and is adjustable using the `-ms` option. Section 4.1 describes how the populations of the various tautomeric and ionic forms are estimated. Section 4.2 provides a more detailed description of the structural adjustment process.

4.1 Penalties and Populations for Ionization State and Tautomeric Forms

Tautomerization methodology and the estimation of tautomer populations are described in Chapter 6 of the *LigPrep User Manual*. pK_a values are estimated using the methods described in Chapter 2 of this manual. pK_a penalties for each structure are estimated using the methods described in Chapter 5 of the *LigPrep User Manual*, using Epik's predicted pK_a values. These penalties, which are expressed in kcal/mol, are recorded with each structure as shown in Table 4.1.

Table 4.1. pKa penalties for generated structure

Property	Description
`r_epik_Ionization_Penalty`	overall penalty for the structure
`r_epik_Ionization_Penalty_Charging`	penalty for having ionizable groups charged
`r_epik_Ionization_Penalty_Neutral`	penalty for having ionizable groups neutral
`i_epik_Tot_Q`	total charge of the structure

The `r_epik_Ionization_Penalty` property can be used to provide a rough estimate for the weight to assign to a particular ionization state, *k*, within the collection of the various ions generated from the same tautomeric form, *t*, using the following equation:

$$\mathrm{Wi}_k = e^{(-(\mathtt{r\_epik\_Ionization\_Penalty}_k)/k_B\mathrm{T})} \qquad (1)$$

The calculation of populations for a specific combination of tautomeric and ionic forms of a molecule is a more complicated process since both tautomeric and ionic probabilities need to be accounted for at the same time. This process needs to be considered in the context of the interactive generation of a collection of structures, involving the calculation of `r_epik_State_Penalty` which is a more rigorous measure of the importance of a particular state than `r_epik_Ionization_Penalty`. See the following section for more information on the state generation process and `r_epik_State_Penalty`.

4.2 Creating Structural Variations and Estimating Populations

The process for generating collections of tautomeric and ionic structural variations on the input structure is iterative, with each iteration involving a tautomeric stage and an ionization stage. In describing the procedure for adjusting the structures generated from a single input structure, it is helpful to use the following phrases:

- “current collection of structures” describes the structures generated in the previous stage of the process
- “new collection of structures” describes the set of structures being generated in the current stage of the process.

At the start of the first iteration, the current collection of structures is the input structure which is initially assigned a population of 1.0.

During the tautomerization stage of each iteration, each structure, *j*, in the current collection is tautomerized separately. The resulting tautomers are assigned weights, $\mathrm{Wt}_{\mathrm{j,t}}$, given by the following equation:

$$\mathrm{Wt}_{j,\,t} = \mathrm{Pc}_j p_t \qquad (2)$$

where:

- Pc_{j} is the population of the structure *j* in the current collection
- p_{t} is the population of the tautomer divided by the population of the original tautomer

When the tautomers have been generated for all the current structures, the $\mathrm{Wt}_{\mathrm{j,t}}$ are normalized to give populations, $\mathrm{pt}_{\mathrm{j,t}}$ as shown in the following equation:

$$\mathrm{Pt}_{j,\,t} = \mathrm{Wt}_{j,\,t} \Big/ \left(\sum_{j,\,t} \mathrm{Wt}_{j,\,t} \right) \qquad (3)$$

All tautomeric forms with population weights lower than *minpop* are eliminated and the resulting structures are now considered the current collection of structures. If more than *ms* structures are present and *ms* Š 3, then only the *ms* most probable structures are kept. If *ms* < 3, three structures are kept.

All structures in the current collection of structures are now subjected to ionization. The pH range, *del_pH*, is given by:

$$del\_pH = -\log_{10}(minpop) \qquad (4)$$

The following steps are carried out separately for each of the current structures, *j*:

1. Estimate the pK_a values for all ionizable groups
2. Accumulate structures into the new collection by examining each ionizable group in turn:
    i. If it is in a basic form and pK_a < pH + del_pH, the basic form is added to the new collection of structures.
    ii. If it is in a basic form and pK_a > pH – *del_pH*, a proton is added to generate a new structure which is added to the new collection.
    iii. If it is an acidic form and pK_a > pH – *del_pH*, the acid form is added to the new collection of structures
    iv. If it is in an acidic form and pK_a < pH + *del_pH*, the proton is removed to generate a new structure which is added to the new collection.

If a new structure is the same as one already present in the new collection, it is not added a second time.

When all of the structures in the current collection of structures have been ionized, the weights for the new states are estimated:

$$\mathrm{Wn}_{j,k} = \mathrm{Pt}_j \mathrm{Wi}_k \tag{5}$$

which are normalized to give estimates for the populations:

$$\mathrm{Pc}_{j,k} = \mathrm{Wn}_{j,k} \Big/ \left(\sum_{j,k} \mathrm{Wn}_{j,k}\right) \tag{6}$$

Those with a population lower than the *minpop* threshold are eliminated. If there are more than *ms* structures, only the *ms* structures with the highest population are retained, with a minimum of 3 structures. The resulting set of structures is now considered the current collection of structures. The populations of individual structures in this collection, *j*, are now referred to as Pc_j.

For each iteration, the process of tautomerizing and ionizing is repeated until either the collection of structures does not change from one iteration to the next, or the maximum number of iterations (6) is reached. At the end of this process, using the population for each structure, Pc_j, the overall state penalty for each structure is calculated using:

$$\texttt{r\_epik\_State\_Penalty}_j = -k_B T \ln(\mathrm{Pc}_j) \tag{7}$$

The process used to determine r_epik_State_Penalty involves a number of significant approximations and thus, this quantity should only be used as a rough guide for gauging the importance of the output structures.

4.3 Structural Adjustment for Metal Binding

In protein binding sites that contain metal atoms, a drug-like molecule will often bind such that the metal replaces one of its protons that would normally have a high pK_a value. The high pK_a values mean that such deprotonated states have low populations in bulk solution and thus Epik's procedure to generate states would not normally produce them.

Epik's metal binding procedure examines the states produced by the state generation procedure (described in the previous section) to see if certain types of hydrogen atoms can be removed to generate additional structures suitable for binding to metals, by individually adjusting their pK_a values down. All of the structures produced by the normal state generation procedure are retained along with the metal binding states. In benchmark studies the use of this procedure significantly increases the fraction of the input molecules that are processed to give the appropriate protonation state based upon comparison with experiment.

When Epik is run with the metal binding option, all of the structures produced have the boolean property b_epik_Metal_Only, which indicates whether the protonation state was specifically produced by the metal binding calculation (1 = True) or would normally be produced by an Epik state generation calculation (0 = False). All of the structures, including the additional ones, have Epik's usual state penalty for bulk solution as a structure-level property (r_epik_State_Penalty). For the protonation states generated for metal binding, this penalty lies below the usual threshold for a state-generation calculation. The bulk penalties are not appropriate when considering the states that are important when the drug-like molecule binds to a metal. A second penalty, r_epik_Metal_State_Penalty, is calculated using the shifted metal-binding pK_a values. Since the penalty depends on which eligible atom binds to the metal this value is recorded as an atom-level property. The lowest such penalty of the atoms for a particular protonation state is also recorded as a structure-level property of the same name.

Sequential pK_a Estimation

Epik's sequential pK_a estimation capability allows you to calculate the microscopic pK_a values at which protons may be successively added or removed from a molecule. This can be useful for predicting or comparing with experimental pK_a values. Some types of experiments such as titrations measure macroscopic pK_a values. When two or more microscopic pK_a values lie with 1 pK_a unit of each other, the macroscopic pK_a values can noticeably differ from the corresponding microscopic values.

Sequential pK_a estimation is turned on by the `-scan` option of the `epik` command. It can be set up in the Epik panel by selecting Sequential pKa values.

Each input structure is processed in three stages:

1. Structure adjustment (see Chapter 4 for more information) is used to produce what Epik regards as the most probable form for the molecule at pH 7.0 or the value specified by the `-ph` option if present. This structure will be referred to as the pH-adjusted structure.
2. The pK_a of the most acidic proton is noted. This proton is removed and the pK_a values are recalculated. This process is repeated until no more acidic hydrogens remain or the pK_a for removing the next proton rises above the `highest_pKa` value specified in the solvation file, unless a different value is specified using the `-highest_pka` option on the command line.
3. The pH-adjusted structure is reinstated. The pK_a of the most basic non-hydrogen atom is noted. A proton is attached to this atom and the pK_a values are recalculated. This process is repeated until no basic atoms remain or the pK_a for adding the next proton drops below the `lowest_pKa` value specified in the solvation file, unless a different value is specified using the `-lowest_pka` option on the command line.

For each structure, the pK_a values for removing or adding protons are recorded in the `.log` file in ascending order along with the type of ionization relative to the input structure (acid or base), the heavy atom bonded to the proton and the type of Hammett or Taft pattern used. An example is given below:

```
Processing Input Structure #761
        Title: 285:Bases
. . . . . . . . . . . . . . . . . . . . . . .
    pKa  type            base   Description
                        atom #
```

```
   8.28  conjugate base    1   Quinolinium ions, 8-hydroxy
   9.06  conjugate acid   41   Phenol, c-OH
   9.62  conjugate acid   15   Tertiary Aminium Ions
```

The pH-adjusted structure is saved to the output structure file. The sequential pK_a values and base atoms are recorded in the properties of the structure. The Maestro property names are `r_epik_calc_pKa_#` and `i_epik_calc_pKa_atom_#`, where # starts at 1 for the lowest pK_a value. These properties are listed in the Project Table as epik_calc_pKa_# and epik_calc_pKa_atom_#.

A script, `compare_epik_results.py`, is available to facilitate comparing the results of an Epik sequential ionization calculation with a reference set of pK_a values. For more information see Appendix D.

pK_a Data File Format

This appendix describes the encoding of pK_a information in a solvent parameter file for Epik. The default parameter files for water and DMSO are not user-accessible. You may use your own parameter files to add to, or replace, the default parameter sets for these solvents or to describe new solvents.

A.1 Basic Elements of the Parameter File

The solvent parameter file has three basic elements:

- comment line
- information lines
- data blocks

Any line beginning with a # symbol is regarded as a comment. Information is provided by lines containing *information_name* : *information_value* pairings.

```
name: Hammett_Taft_water
short_name: H2O
```

Only one *information_name* : *information_value* pairing can appear on any line.

Data blocks associate information lines into distinct groups. For instance, the following data block specifies the information needed to describe the Taft relation for a tertiary alcohol:

```
acid_base{
        Sch_ID: 13
        name: Alcohols, tertiary
        pattern: [#1]-[OX2]-[CX4H0]
        identifier: alcohol_tertiary
        sub_atoms: 3
        pKa0: 15.9
        rho: 1.42
}
```

while this block contains Hammett information for protonating aniline:

```
acid_base{
        Sch_ID: 1029
        name: Anilinium ions
        pattern: [#1]-[NX4+](-[#1])(-[#1])-c1[a][a][a][a][a]1
        identifier: aniliniumIon_phenyl
        sub_atoms: 6 7 8 9 10
        hetero_aromatic: 6 7 8 9 10
        pKa0: 4.58
        rho: 2.88
}
```

Blocks must begin with `block_type_name{` and end with a corresponding }. Only one occurrence of any *information_name* can appear within a given block.

Information lines that lie outside a data block apply to the whole solvent description, while information lines that lie inside blocks apply only to instances where the block is used. Aside from block-level grouping of information, data can be provided in any order in the solvent parameter file.

Table A.1. Data types that occur outside a block

Information_name[a]	Type	Description
`name`	string	Full name for entire set of pK_a data
`short_name`	string	Shorter name for entire set of pK_a data
`temperature`	real	Temperature for the parametrization in degrees C.
`clear_standard`	none	Clears out pK_a parameter information present before reading this file.

a. All of these descriptors are required in the parameterization file.

There are a number of types of top-level data blocks:

- `acid_base` — Describes a distinct parameterization for an acid/conjugate base pair.
- `transmission` — definition of the atom in a transmission group, and how strongly it propagates the perturbations .
- `substituent` — Description of substituent fragments and their perturbing effects.
- `ring_adjustment` — Describes how to adjust pK_a values for base atoms (the second atom in acid_base patterns) in aliphatic rings.
- `hetero_atom_group` — Describes hetero atom based groups in aromatic rings for which automatic adjustment of the pK_a should be attempted.

- `charge_spreading` — Describes the chemical functionalities that are involved in charge spreading
- `hydrogen_penalty` — Describes which hydrogen σ types to use when removing hydrogens from non-explicit substitution locations. Only one `hydrogen_penalty` should be in the file.
- `general_crowding` — Defines shifts to calculated pK_a values for matching `acid_base` groups in specific local environments.
- `turn_off` — Lists previously defined data blocks to disable.

The following information_name values describe special σ types, and can only appear within substituent blocks:

- `special_sigma_ortho`
- `special_sigma_meta`
- `special_sigma_para`

Table A.2 provides a description of the types of data blocks (with the exception of the charge spreading block).

Table A.2. Data block component definitions

Information_name	Required	Type	Purpose
`acid_base`			
`name`	Yes	String	Name for ionizable functional group.
`Sch_ID`	No	Integer	For identification.
`pattern`	Yes	String	SMARTS pattern for acidic form.
`net_charge`	No	Integer	A valid match for the SMARTS pattern must also have this net total formal charge on the matching substructure. If absent the total formal charge is not checked.
`identifier`	No	String	Used to systematically describe the nature of the group.
`subs_atoms`	Yes	Integers	A list of atom numbers in the SMARTS pattern for which the ρ value given for the attachment of substituents is used.
`subs_atoms_rhos`	No	Real	A list of ρ values for each substitution location listed under `subs_atoms`. The overall ρ is still required as it applies to other terms in the Hammett/Taft calculation.

Table A.2. Data block component definitions

Information_name	Required	Type	Purpose
hetero_aromatic	No	Integers	A list of atom numbers within the SMARTS pattern for which automatic corrections for hetero atoms defined in the hetero_atom_group blocks may be used.
pKa0	Yes	Real	Specifies the Hammett or Taft pK_a^0 value.
rho	Yes	Real	Specifies the Hammett or Taft ρ value.
pKa_unc	No	Real	The uncertainty in the pK_a values estimated for this group. Default is 2.0.
use_hydrogen_sigma	No	Integer	Indicates whether hydrogen σ values are used. Default is 0 (no); yes is 1.
priority	No	Real	Assigns a priority to the group. Default is 0. Larger values denote a high precedence.
sigma_ortho_wt	No	Real	Normally, which σ value to use is determined by the nature of the molecule. However, some parametrizations for particular aliphatic functional groups explicitly state that certain non-default σ values must be used.
sigma_meta_wt	No	Real	
sigma_para_wt	No	Real	
sigma_star_wt	No	Real	
crowding	No	Structure	See crowding structure. Used to provide very specific pK_a adjustments depending on the environment for the acid_base group.
transmission			
name	Yes	String	Name for the transmission group.
Sch_ID	No	Integer	For identification
pattern	Yes	String	SMARTS pattern for the transmission group. Multi-atom transmission groups are not supported.
tcoef	Yes	Real	Transmission coefficient. τ typically 0.4 (range = 0.3 to 0.7)
substituent			
name	Yes	String	Name for substituent.
Sch_ID	No	Integer	For identification.

Table A.2. Data block component definitions

Information_name	Required	Type	Purpose
pattern	Yes	String	SMARTS pattern for substituent group.
sigma_ortho	At least one σ must be given	Real	Influence of substituent (> 0 acid strengthening)
sigma_meta		Real	
sigma_para		Real	
sigma_star		Real	
special_sigma_ortho	No	Structure	See special_sigma structure. Used to give special σ values for combinations of certain substituents of aromatics with certain acid/base groups.
special_sigma_meta	No	Structure	
special_sigma_para	No	Structure	
special_sigma_ortho special_sigma_meta special_sigma_para			
name	Yes	String	Name of acid for special σ value.
Sch_ID	No	Integer	For identification.
pattern	Yes	String	Pattern for acidic version of special acid/base substituent combination.
sigma	Yes	Real	σ value to use.
ring_adjustment			
name	Yes	String	Name of ring adjustment.
Sch_ID	No	Integer	For identification.
pattern	Yes	String	SMARTS pattern, usually just one atom (e.g., N).
pKa_shift1	Yes	Real	Shift if in 1 ring.
pKa_shift2	No	Real	Shift if in 2 rings.
hetero_atom_group			
name	Yes	String	Name of hetero atom group.
Sch_ID	No	Integer	For identification.
pattern	Yes	String	SMARTS pattern to match. First atom should be in the aromatic ring.

Table A.2. Data block component definitions

Information_name	Required	Type	Purpose
sigma_ortho	One of these three must be included	Real	σ adjustments for different ring locations.
sigma_meta		Real	
sigma_para		Real	
ring_size	Yes	Integer	Apply only to rings of this size.
hydrogen_penalty			
sigma_ortho	One of these four must be included	Real	σ value to use.
sigma_meta		Real	
sigma_para		Real	
sigma_star		Real	
crowding			
name	Yes	String	Name of crowding group.
pattern	Yes	String	SMARTS pattern for the crowding group. The first atom in this pattern must match the first atom in the SMARTS pattern for the acid_base group.
pKa_shift	Yes	Real	The numeric shift that should be applied to the calculated pK_a if the crowding group matches.
atom_map	No	Integers	A list of integers that determines which atoms in the crowding group may or may not match atoms in the acid_base group. There must be 1 integer for each atom in the SMARTS pattern for the crowding group, with one of these values: >0 : this atom in the crowding pattern specifies the corresponding atom in the acid_base SMARTS pattern 0 : this atom in the crowding pattern should not match any atom in the SMARTS pattern –1 : this atom in the crowding pattern can match any atom in the SMARTS pattern –2 : no restriction regarding the presence or absence of this atom in the crowding pattern.

Table A.2. Data block component definitions

Information_name	Required	Type	Purpose
general_crowding			
name	Yes	String	Name of crowding group.
Sch_ID	Yes	Integer	For identification.
pattern	Yes	String	SMARTS pattern for the crowding group. The first atom in this pattern must match the first atom in the SMARTS pattern for the acid_base group.
pKa_shift	Yes	Real	The numeric shift that should be applied to the calculated pK_a if the crowding group matches.
atom_map	No	Integers	A list of integers that determines which atoms in the crowding group may or may not match atoms in the acid_base group. There must be 1 integer for each atom in the SMARTS pattern for the crowding group, with one of these values: >0 : this atom in the crowding pattern specifies the corresponding atom in the acid_base SMARTS pattern 0 : this atom in the crowding pattern should not match any atom in the SMARTS pattern –1 : this atom in the crowding pattern can match any atom in the SMARTS pattern –2 : no restriction regarding the presence or absence of this atom in the crowding pattern.
use_rho	No	Integer	0 : apply pKa_shift unmodified (default). 1 : multiply pKa_shift by the rho value for the matching acid_base group.
turn_off			
acid_base	No	Integer	Sch_ID for structure to turn off.
transmission	No	Integer	Sch_ID for structure to turn off.
substituent	No	Integer	Sch_ID for structure to turn off.
ring_adjustment	No	Integer	Sch_ID for structure to turn off.
hetero_atom_group	No	Integer	Sch_ID for structure to turn off.

A.2 Notes about SMARTS patterns for acid_base and special_sigma blocks

The SMARTS patterns in these blocks have some restrictions placed on them. Patterns must show the acidic form. The conjugate base form is generated automatically from the acidic SMARTS pattern. The first atom must be the acidic hydrogen and be represented as: [#1]. The second atom in the pattern must be the base atom (the atom to which the acidic hydrogen is bonded). This atom may have the X, H, v and +/- (charge) qualifiers; if more than one of them is present, they must appear in this order (e.g., NX3H2v4+).

The base atom may also be assigned an alternate charge by including a comma right after the specification for the first atom and repeating that specification with the new charge (e.g. [nX3+1,nX3-0] would permit the nitrogen to be positive or neutral). Additional conditions may be applied but only using a low priority and ";". Those conditions are identical for the acid and automatically generated conjugate base forms for the functional group (e.g. [nX3+1,nX3-0;r6]).

A.3 Notes about acid_base groups

The parameter file must contain at least one `acid_base` group. Multiple groups may match for a particular ionizable location. One is selected based upon its priority, size, proximity of substitution locations to base atom, and extent of unsaturation between substitution locations and the base atom. The pK_a^0 from this pattern is used. ρ values from multiple patterns may be used. See Chapter 3 for more information.

Tautomer Database Format

In Epik 3.2 the default tautomer database is not accessible to users. However, you can provide your own file to either completely override or add patterns to the default tautomer collection.

At the top level of the tautomer database file the following four items can be present: `name`, `clear_standard`, `group_def`, and `tautomer_set`. These items are described in the following sections. Lines beginning with a # are comment lines and are ignored when interpreting the contents of the tautomer database file. Blank lines are also ignored.

name Data Item

`name` specifies the name of the solvent. For example:

```
name: water
```

`water` and `DMSO` are standard names for which the tautomerizer already has information. Currently, the DMSO tautomer information is just a duplicate of that for water.

clear_standard Directive

By default, information in a custom tautomer database file is added to any existing information already available for the solvent specified. Including `clear_standard:` in a tautomer database file clears any values for this solvent accumulated before the current file was read.

group_def Data Structure

The tautomerization facility does not support recursive SMARTS. However, a mechanism that supports some of the functionality of recursive SMARTS is provided by the `group_def` data structure. This data structure permits you to define variables that correspond to SMARTS patterns. The variables may be reused in groups and `tautomer_sets` that appear later in the tautomer database file.

Each group contains two items:

`name:` an arbitrary name for the group which is used to reference the group.

`pattern:` The SMARTS pattern for the group. This pattern may refer to previously defined groups using `$`*groupname*.

Below are some examples of group_def data structures:

```
group_def{
        name: Halogens
        pattern: [F,Cl,Br,I]
}

group_def{
        name: Amides
        pattern: [CX3](=[OX1])-[NX3]
}

group_def{
        name: Carbonyls
        pattern: [CX3](=[OX1])
}

group_def{
        name: Carbonyls_only
        pattern: [$Carbonyls;!$Amides]
}
```

tautomer_set Data Structure

tautomer_set data structures define sets of interconvertible tautomers. There are more than 150 tautomer sets available by default for water.

Some examples of tautomer_set data structures are given below, and the syntax for the data structures is described following these examples.

Note: The entry for pattern: values must be a single line. In the examples below, some of the pattern: text wraps to the next line due to formatting constraints within this manual. When creating tautomer data structure files in a text editor, ensure that text-wrapping is turned off, or that margins are set wide enough to accommodate single-line entry for this value.

```
tautomer_set{
    name: single-sided_ket-enol
# From: Handbook of organic chemistry

    tautomer{
      name: enol
      pattern: [CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC])=[CX3](-
[#1,$Sub_carbonyl_C])-[OX2]-[#1]
      probability: 0.00005
    }

    tautomer{
      name: ket
      pattern: [CX4](-[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC])-[CX3](-
[#1,$Sub_carbonyl_C])=[OX1]
      probability: 0.99995
    }
}

tautomer_set{
    name: double-sided_ket-enol
# From: Handbook of organic chemistry

    tautomer{
      name: 1enol
      pattern: [CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC])=[CX3](-[CX4](-
[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC]))-[OX2]-[#1]
      probability: 0.00000001
    }

    tautomer{
      name: ket
      pattern: [CX4](-[#1,$Sub_aC])(-[#1,$Sub_aC])(-[#1])-[CX3](-
[CX4](-[#1])(-[#1,$Sub_aC])(-[#1,$Sub_aC]))=[OX1]
      probability: .99999998
    }

    tautomer{
      name: 2enol
      pattern: [CX4](-[#1,$Sub_aC])(-[#1,$Sub_aC])(-[#1])-
[CX3](=[CX3](-[#1,$Sub_aC])(-[#1,$Sub_aC]))-[OX2]-[#1]
      probability: 0.00000001
    }
}
tautomer_set{
    name: imidazole

    tautomer{
      name: form1
      pattern: c1(~[#1,$Sub_c])n(-[#1,$Sub_n])-c(-
[#1,$Sub_c])=[nX2]c1(~[#1,$Sub_c])
      probability: 0.50
    }

    tautomer{
      name: form2
      pattern: c1(~[#1,$Sub_c])[nX2]=c(-[#1,$Sub_c])-n(-
[#1,$Sub_n])c1(~[#1,$Sub_c])
      probability: 0.50
    }
}
```

Each tautomer set contains a `name:` designator and a number of tautomer structures. The `name:` designator is followed by a space and a contiguous non-blank label to identify the class of tautomers described by the set. The label provided does not affect processing. In the examples below, there are three tautomeric sets: `single-sided_enol-ket`, `double-sided_enol-ket`, and `imidazole`.

The `tautomer` structure describes the properties of one tautomeric form. There are three designators that may be used within a `tautomer` structure: `name:`, `probability:`, and `pattern:`.

The `name:` designator provides a label for the tautomer but does not otherwise affect processing.

The `probability:` designator is used to assign a probability or fractional population of this tautomer within this tautomeric set. In many cases, reliable information on the probability of various tautomeric forms is not available and the values entered in the database are simply educated guesses.

The `pattern:` designator is followed by a contiguous SMARTS-like pattern. A difference between this pattern and a normal SMARTS pattern is that explicit single “–” and double “=” bond designators are used to make the corresponding Lewis structures clear. In addition, these patterns may include references to previously defined groups via the $*group_name* mechanism. Information on SMARTS patterns is provided on the web page: http://www.daylight.com/learn[1]. The SMARTS-like pattern is used to detect the corresponding groups of molecules in the input structures and to permit the tautomerization facility to understand how the bonding patterns (Lewis structures) differ between tautomers so that they may be interconverted. For heavy atoms that are expected to carry a formal charge it is advisable to include the charge in the SMARTS pattern. To ensure that the SMARTS patterns are properly interpreted by Epik, the following restrictions must be applied:

- The SMARTS patterns for all tautomers within a tautomer set include the same list of non-hydrogen atoms in the same order.
- All SMARTS patterns must explicitly designate the hydrogens that shift positions in any tautomer within a tautomer set with a `-[#1]` pattern.
- All SMARTS patterns within a tautomer set must contain the same number of explicitly designated mobile hydrogen atoms.
- In both non-aromatic and aromatic portions of the SMARTS pattern, bond orders that change between single and double in any tautomer must be explicitly specified in the SMARTS patterns for all tautomers in a tautomer set.
- In portions of molecules that must be represented by aromatic atom types (e.g., `c` and `n`), only changes in the bond orders of bonds involving `n` atoms in the corresponding Lewis structures are supported. If such a bond changes order in any tautomer in a tautomer set, it must be represented as '`:`' in all the tautomers. See the guanosine tautomer set in the example above.
- Recursive SMARTS patterns are not supported.
- SMARTS patterns within the same tautomer set must all specify the same overall formal charge.

The database provided with this release contains templates for keto-enol tautomers and their sulfur analogues, imine-enamine tautomers, histidine-like tautomers, tautomers of DNA and RNA bases, and a large number of common heteroaromatic rings containing C, S, O, and N.

[1] Please see the notice regarding third party programs and third party Web sites on the copyright page at the front of this manual.

Epik Properties

This appendix lists most of the properties generated by Epik. Epik can generate a number of structure-level and atom-level properties, depending on the options that are used, particularly the mode. The mode can be one of the following:

- Query (Q)—predict pK_a values for the input structure (run without `-ph` or `-scan` options)
- Predict states (PS)—predict structures consistent with a specified pH and pH tolerance (run with `-ph`), and optionally generate metal-binding states (PS+mb mode, run with `-ph` and `-metal_binding`)
- Sequential pKa values (SPV)—predict structures consistent with a specified pH and pH tolerance, (run with `-scan`)

The properties are described in the tables below. They are given by their internal name. The name presented in Maestro is generated by removing everything up to and including the second underscore character, and then replacing underscores with spaces. For example, `i_epik Tot_Q` is converted to Tot Q in Maestro.

Table C.1. Structure-level properties produced by Epik.

Property	Mode	Description
`r_epik_State_penalty`	PS, SPV	Overall state penalty in kcal/mol
`r_epik_Ionization_Penalty`	PS, SPV	Overall penalty for the structure
`r_epik_Ionization_Penalty_Charging`	PS, SPV	Penalty for having ionizable groups charged
`r_epik_Ionization_Penalty_Neutral`	PS, SPV	Penalty for having ionizable groups neutral
`r_epik_tautomer_probabilty`	PS, SPV	Fraction of the population of states at the same charge level due to the state
`i_epik_Tot_Q`	PS, SPV	Net charge on the molecule
`r_epik_pKa_`*N*	SPV	Value of the *N*th pK_a
`r_epik_pKa_atom_`*N*	SPV	Heavy atom for the *N*th pK_a value

Table C.1. Structure-level properties produced by Epik.

Property	Mode	Description
`r_epik_pKa_indentifier_`*N*	SPV	Unique and brief description of the type of functional group for the *N*th pK_a value
`r_epik_Metal_State_Penalty`	PS+mb	Overall metal binding state penalty in kcal/mol
`b_epik_Metal_Only`	PS+mb	Indicator of metal binding state: 0 state normally generated in PS mode 1 state generated in PS mode for metal binding
`s_epik_Chemistry_Notes`	Any	Notes on structures with problematic chemistry.
`s_epik_pKa_notes`	Any	A concatenation of messages produced by Epik. See below for the possible messages.

The messages that are used in the notes property are listed below with their meanings:

- Molecule size exceeds maximum supported— The molecule is larger than the threshold for normal processing by Epik.
- Reset negative state penalty to 0 from Metal Processing—Rarely, metal binding calculations can incorrectly predict a negative state penalty for new structures. This message indicates that such values have been reset to 0.

Table C.2. Atom-level properties produced by Epik.

Property	Mode	Description
`r_epik_`*solvent*`_pKa`	Any	The calculated pK_a value for an atom in a particular (H2O or DMSO).
`r_epik_H2O_pKa_uncertainty`	Any	The uncertainty in the calculated pK_a value for an atom in a particular <solvent> (H2O or DMSO).
`r_epik_Metal_State_Penalty`	PS+mb	The atom-specific state penalty calculated in the metal binding mode. This penalty is appropriate if this atom lies close to a metal atom in a metalloprotein.

The compare_epik_results.py Script

The `compare_epik_results.py` script can be used to compare the results of an Epik sequential pK_a calculation with a standard set of pK_a values. The sequential pK_a calculation can be run using the command:

```
$SCHRODINGER/epik -ph 7.0 -scan -imae input.mae -omae epik-output.mae
```

where *input*`.mae` is a Maestro file containing the ligands for which pK_a estimates are to be made. The file *epik-output*`.log` documents the results from the Epik calculation. Once this calculation is complete, the `compare_epik_results.py` script may be used to compare Epik's predictions against the reference set of results. This script can be run using the command:

```
$SCHRODINGER/run -FROM epik compare_epik_results.py
    reference-csv epik-output.log summary-file [-inorder] [-skip_first]
```

where:

- *ver* is the Epik version number
- *platform* is the appropriate platform type for a version of Epik that you have installed
- *reference-csv* is a comma-separated file containing the reference pK_a values with one line for each ligand of the form:

  title, *pKa1*, *pKa2*, ...
- *summary-file* is the file that contains the results of the comparison
- `-inorder` matches the results in the *reference-csv* and *epik_output*`.log` files based on their order in these files.
- `-skip_first` means skip the first line in *reference-csv*. This is useful if the first line contains column titles.

By default the result sets are compared by matching the titles for each ligand. The titles listed for each ligand in *reference-csv* and *epik_output*`.log` must exactly match unless `-inorder` is specified. In which case the titles are set to the count of the ligands in the *reference-csv* file.

There may be multiple ways to match up the reference and calculated pK_a values. Epik typically estimates more pK_a values than the reference values, usually numerically higher and lower. As well, Epik may not have an appropriate pattern for estimating a pK_a value. To avoid forcing matches when a pattern is missing Epik considers not matching one or more of the reference pK_a values. For each number of matches this script selects the alignment that gives the smallest sum of differences between the reference and predicted pK_a values. The alignment with the highest number of matches is selected, unless dropping a match reduces the sum of the differences by more than 4 pK_a units.

Typical output should look like the following:

```
--------------------------------------------------------------------------------
Processing Structure: Acids_test_472
Matched pKas
    Exp     Predict    Difference
     5.38     4.93     0.45
     8.42     7.71     0.71
Unmatched Predictions
             -0.28
              0.69
```

It is not unusual to have more predictions than reference values. If it was not possible to match a predicted pK_a to a reference pK_a the values in the Predict and Difference columns are listed as N/A.

At the bottom of the *summary-file*, a summary of all of the results is given and should look something like:

```
 Summary of results

Number of matches:   15
Average difference:  0.38
standard deviation:  0.98
median err:          0.50
```

Almost all pK_a prediction programs, including Epik, involve some sort of a parametrization based upon pattern recognition. Usually it is difficult, if not impossible, to include patterns for all types of proton addition or removal. As a result, when the appropriate pattern is missing either no prediction is made or another pattern intended for a different chemistry may be used. This can be problematic when comparing with reference results. A normal distribution is unlikely to result from such comparisons. As well, standard deviations tend to heavily weight poor matches between predicted and reference results. Consequently, based primarily on customer input, we typically use the median error to judge the level of accuracy of the results.

Getting Help

Information about Schrödinger software is available in two main places:

- The `docs` folder (directory) of your software installation, which contains HTML and PDF documentation. Index pages are available in this folder.
- The Schrödinger web site, http://www.schrodinger.com/, In particular, you can use the Knowledge Base, http://www.schrodinger.com/kb, to find current information on a range of topics, and the Known Issues page, http://www.schrodinger.com/knownissues, to find information on software issues.

Finding Information in Maestro

Maestro provides access to nearly all the information available on Schrödinger software.

To get information:

- Pause the pointer over a GUI feature (button, menu item, menu, ...). In the main window, information is displayed in the Auto-Help text box, which is located at the foot of the main window, or in a tooltip. In other panels, information is displayed in a tooltip.

  If the tooltip does not appear within a second, check that Show tooltips is selected under General → Appearance in the Preferences panel, which you can open with CTRL+, (⌘,). Not all features have tooltips.

- Click the Help button in the lower right corner of a panel or press F1, for information about a panel or the tab that is displayed in a panel. The help topic is displayed in the Help panel. The button may have text or an icon:

  ?

- Choose Help → Online Help or press CTRL+H (⌘H) to open the default help topic.
- When help is displayed in the Help panel, use the navigation links in the help topic or search the help.
- Choose Help → Documentation Index, to open a page that has links to all the documents. Click a link to open the document.

- Choose Help → Search Manuals to search the manuals. The search tab in Adobe Reader opens, and you can search across all the PDF documents. You must have Adobe Reader installed to use this feature.

For information on:

- Problems and solutions: choose Help → Knowledge Base or Help → Known Issues → *product*.
- New software features: choose Help → New Features.
- Python scripting: choose Help → Python Module Overview.
- Utility programs: choose Help → About Utilities.
- Keyboard shortcuts: choose Help → Keyboard Shortcuts.
- Installation and licensing: see the *Installation Guide*.
- Running and managing jobs: see the *Job Control Guide*.
- Using Maestro: see the *Maestro User Manual*.
- Maestro commands: see the *Maestro Command Reference Manual*.

Contacting Technical Support

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

Web:	http://www.schrodinger.com/supportcenter
E-mail:	help@schrodinger.com
Mail:	Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone:	+1 888 891-4701 (USA, 8am – 8pm Eastern Time)
	+49 621 438-55173 (Europe, 9am – 5pm Central European Time)
Fax:	+1 503 299-4532 (USA, Portland office)
FTP:	ftp://ftp.schrodinger.com

Generally, using the web form is best because you can add machine output and upload files, if necessary. You will need to include the following information:

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

Gathering Information for Technical Support

The instructions below describe how to gather the required machine, licensing, and installation information, and any other job-related or failure-related information, to send to technical support. Where the instructions depend on the profile used for Maestro, the profile is indicated.

For general enquiries or problems:

1. Open the Diagnostics panel.
   - **Maestro:** Help → Diagnostics
   - **Windows:** Start → All Programs → Schrodinger-2015-2 → Diagnostics
   - **Mac:** Applications → Schrodinger2015-2 → 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. Upload the file specified in the dialog box to the support web form.

   If you have already submitted a support request, use the upload link in the email response from Schrödinger to upload the file. If you need to submit a new request, you can upload the file when you fill in the form.

If your job failed:

1. Open the Monitor panel, using the instructions for your profile as given below:
   - **Maestro/Jaguar/Elements:** Tasks → Monitor Jobs
   - **BioLuminate/MaterialsScience:** Tasks → Job Monitor
2. Select the failed job in the table, and click Postmortem.

   The Postmortem panel opens.
3. If your data is not sensitive and you can send it, select Include structures and deselect Automatically obfuscate path names.
4. Click Create.

   An archive file is created, and an information dialog box with the name and location of the file opens. You can highlight and copy the name of the file.
5. Upload the file specified in the dialog box to the support web form.

   If you have already submitted a support request, use the upload link in the email response from Schrödinger to upload the file. If you need to submit a new request, you can upload the file when you fill in the form.

6. Copy and paste any log messages from the window used to start the interface or the job into the web form (or an e-mail message), or attach them as a file.

   - **Windows:** Right-click in the window and choose Select All, then press ENTER to copy the text.
   - **Mac:** Start the Console application (Applications → Utilities), filter on the application that you used to start the job (Maestro, BioLuminate, Elements), copy the text.

If Maestro failed:

1. Open the Diagnostics panel.

   - **Windows:** Start → All Programs → Schrodinger-2015-2 → Diagnostics
   - **Mac:** Applications → SchrodingerSuite2015-2 → Diagnostics
   - **Linux/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. Upload the file specified in the dialog box to the support web form.

   If you have already submitted a support request, use the upload link in the email response from Schrödinger to upload the file. If you need to submit a new request, you can upload the file when you fill in the form.

4. Upload the error files to the support web form.

   The files should be in the following location:

- **Windows:** `%LOCALAPPDATA%\Schrodinger\appcrash`
  (Choose Start → Run and paste this location into the Open text box.)
  Attach `maestro_error`*pid*`.txt` and `maestro.exe`*pid_timestamp*`.dmp`.

- **Mac:** `$HOME/Library/Logs/CrashReporter`
  (Go → Home → Library → Logs → CrashReporter)
  Attach `maestro_error`*pid*`.txt` and `maestro_`*timestamp_machinename*`.crash`.

- **Linux:** `$HOME/.schrodinger/appcrash`
  Attach `maestro_error`*pid*`.txt` and `crash_report_`*timestamp_pid*`.txt`.

If a Maestro panel failed to open:

1. Copy the text in the dialog box that opens.

2. Paste the text into the support web form.

Index

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

Dynamostraβe 13
D-68165 Mannheim
Germany

Zeppelinstraβe 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®