Sample LaTeX document for mol2chemfig

This sample document illustrates the use of mol2chemfig in conjunction with the chemfig package. This document should compile as is on any system that has a working installation of the chemfig package. A local installation of the mol2chemfig program is not required to compile it, but it *is* required to run the example commands that were used to generate the code. As an alternative to local installation, you can use the web interface at chimpsky.uwaterloo.ca/mol2chemfig.

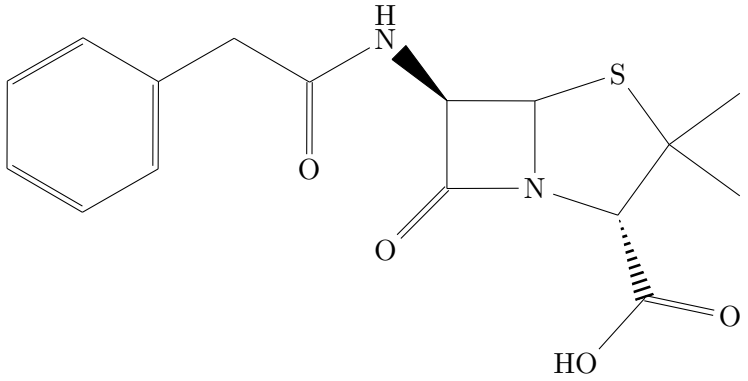
In any case, you will need the mol2chemfig LaTeX package in order to compile the generated LaTeX code. A copy of the package should be in the same directory as this source file.

The structure of penicillin G

The chemfig code contained in the file penicilling.tex was generated from the molfile with the command:

```
mol2chemfig -wf penicilling.mol > penicilling.tex
```

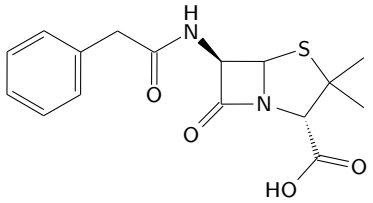
With chemfig's default settings, the structure comes out as follows:



This is a little bit out of proportion to the surrounding text. We can easily adjust the appearance by tweaking some of the settings provided by chemfig:

```
% bond styling
\setcrambond{2.5pt}{0.4pt}{1.0pt}
\setbondoffset{1pt}
\setdoublesep{2pt}
\setatomsep{16pt}
% print atoms with smaller font and in sans-serif
\renewcommand{\printatom}[1]%
{\fontsize{8pt}{10pt}\selectfont{\ensuremath{\mathsf{#1}}}}%
```

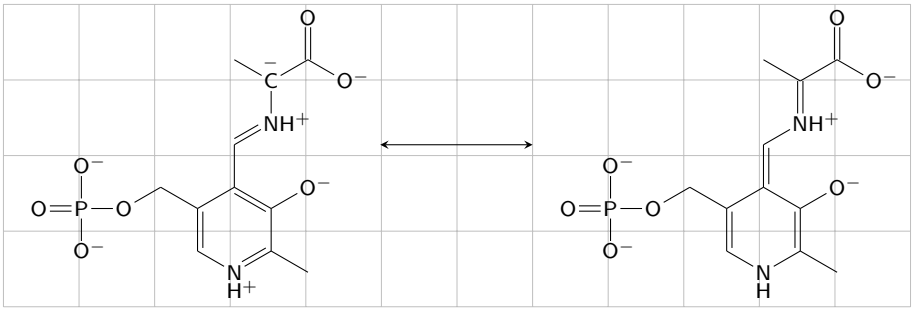
After these settings have been adjusted, the same structure now appears as follows:



We will leave these settings in effect for the remainder of this document.

Incorporating rendered structures into composite graphics

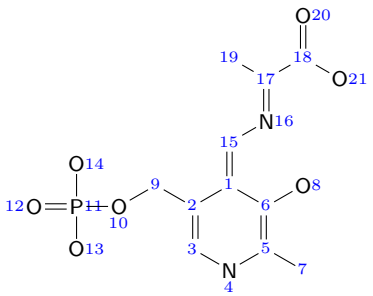
The `mol2chemfig` package loads `chemfig`, which in turn loads the general purpose graphics package `TikZ`. Through the latter package, we have access to the `tikzpicture` environment. Here is a `tikzpicture` that depicts two resonance structures of alanine bound to pyridoxal phosphate:



Note that the `\input` macro cannot be used inside a `\node` in the `tikzpicture` environment. As a workaround, the two structures were rendered as `\submol` definitions, which were `\input` outside the `tikzpicture` environment and then referenced from within the `\node` macros.

Of course, to be complete and valid, the scheme above should also include arrows that indicate the electron movements underlying the resonance effect. This is possible in `chemfig` but will in this case require manual annotation of the generated code. So that we can find our way through the code, we can first render the structure to be annotated with atom numbers and display it:

```
mol2chemfig -wn plp2.mol > plpn.tex
```



This tells us that we need to draw electron movement arrows from nitrogen 4 to the adjacent bond to carbon 3, from bond 2→3 to bond 1→2, and from bond 1→15 to bond 15→16.

The mechanism provided by `chemfig` for drawing push arrows is based on named handles for bonds and atoms. Since version 1.4, `mol2chemfig` allows you to automatically generate named handles for each atom and bond, using the `--markers` or `-g` option:

```
mol2chemfig -f -l plpa -g a plp.mol  > plpa.tex
mol2chemfig -f -l plpb -g b plp2.mol > plpb.tex
```

This option adds a unique identifier to each atom and to each bond. The option value (`a` and `b` in our example) is used as a prefix; this allows to unambiguously reference atoms in multiple molecules in the same drawing. In our first example, atom 3 will be given the marker `@{a3}` for atom 3. The bond between atoms 3 and 4 will be labeled with `@{a3-4}`; in bond markers, the smaller atom number always precedes the larger one. The generated code now gets a wee bit harsh on the eyeballs:

```
\definesubmol{plpa}{
                              @{a7}%
                -[@{a5-7}:150]@{a5}%
        -[@{a4-5}:210,,,,drhs]@{a4}\mcfbelowright{N}{H}{^{\mcfplus}}%
                -[@{a3-4}:150]@{a3}%
          -[@{a2-3}:90,,,,drh]@{a2}%

              ...
```

We can now reference the handles to attach the electron movement arrows. Note that, for this to work, the document has to be processed *twice* by `pdflatex`, since the commands internally use a PDF overlay mechanism; otherwise, the arrows may be misplaced.

The `\mcfpush` macro that is used to place the arrows is defined in the `mol2chemfig` package and is explained in the source code of this document; it is a convenience wrapper around the `\chemmove` command provided by `chemfig`.