Vcharge v1.0

CONTENTS

OVERVIEW

Back to top

Vcharge is a standalone program which calculates accurate, conformation-independent, "*ab initio*-like" partial atomic charges for an SDfile of drug-like compounds in ~0.2 seconds per drug-like compound. It is useful for a wide range of modeling and QSAR applications. The separate **Vdisplay** program provides a convenient graphical way to view the output of Vcharge. The MS Windows version of Vcharge includes a simple graphical user interface which may be used if the user prefers not to use the command-line interface.

The main inputs are

- an SDfile containing one or more molecules
- command-line parameters for settings and output options.

The outputs are:

- an SDfile containing the input structures with hydrogens added to fill valences if they were missing, and with two additional datablocks; <Partial_charges>, which lists the number of atoms in the molecule followed by a list of partial charges and <Error> which contains any error message generated by Vcharge (see Error Messages below).
- a log file containing summary information and error messages.

COMMAND-LINE EXECUTION

Quick Start

```
Prompt% Vcharge.exe <molecules.sdf>
```

where <molecules.sdf> is in MDL SDfile or MOLfile format. The progress of the calculation can be monitored by viewing the log file <molecules>.log.

Stopping a Run in Progress

A run in progress can be cleanly stopped by placing an empty file with the name of the log file but with ".log" replaced by ".end" into the working directory. For example, if the log file is "diversity_vcharge.log", the run will be stopped by placing a file named "diversity_vcharge.end" in the same directory as the log file.

The Linux "touch" command provides an easy way to create this file; e.g., "touch diversity_vcharge.end".

The Files section of the log file includes a reminder of the name and location of the file required to stop the calculation; see the "Stop file:" line.

Command Line Options Back to top

Vcharge is invoked via the following command line:

```
Vcharge.exe <molecules.sdf> [-f firstMol] [-l lastMol] [-o logFile] [-
p chargedSDF] [-fc] [-tl timeLimit] [-v] [-help]
```

Here <molecules.sdf> is an MDL SDfile (www.mdli.com/downloads/literature/ctfile.pdf) containing multiple molecules, or a MOLfile that contains one molecule. The program processes molecules i=firstMol, ..., lastMol from the SDfile and generates a new SDfile named <molecules>_vcharge.sdf that contains the input structures annotated with the calculated properties. It also generates a log file named, by default, <molecules_vcharge.log>.

The command line parameters in square brackets are optional; their usage is as follows.

1. firstMol and lastMol: These integer parameters point respectively to the first and last molecules in the input SDfile that are to be processed.
   Default: 1 and the number of molecules in the SDfile, respectively.

2. logFile: Name of file which records the progress of the run, error messages, and warnings.
   Default: <molecules>_vcharge.log

3. chargedSDF: Name of file which output file containing the datablock with the computed partial charges.
   Default: <molecules>_vcharge.sdf.

4. –fc: By default, Vcharge assumes that formal charges are specified in the input file and that no hydrogen atoms are included, and it adds the required hydrogens accordingly. The -fc flag causes Vcharge to assume instead that all hydrogens are present and that formal charges are missing. Note that the -fc option can cause valence checking errors and/or abnormal formal charges if the SDfile does not include all required hydrogens.
   Default: assume formal charges are present and hydrogens need to be added.

5. timeLimit: Approximate cpu time limit for each molecule being processed. Occasionally, an SDfile may contain a few particularly complex molecules requiring extra processing time. Setting an appropriate value of timeLimit causes such molecules to be skipped after the the time limit is reached and thus ensures a predictable processing time for the SDfile as a whole. Note that the actual cpu time used for a molecule may be slightly longer than timeLimit because the time usage is only checked at the end of certain steps of the charging process.
   Default: 5 seconds.

6. –v: verbose option; causes some of the information normally found only in the log file to be printed to standard out. This is useful if one wants to monitor the progress of the calculation without accessing the log file.

Back To Top

7. –help: List all Vcharge options with brief explanations.

INPUT FILES

The main input file, <molecules.sdf>, is a standard SDfile or MOLfile containing a single molecule. **Note**: Aromatic bond types are not supported in Vcharge.

OUTPUT FILES

Vcharge generates an SDfile containing the input structures along with the output charges in standard SD format data blocks. If the input file includes formal charges and no hydrogens (see –fc option above), then Vcharge adds the missing hydrogens in order to complete the molecules and permit partial charges to be calculated. The output SD file is designed to be compatible with other programs that read SD files, and the data blocks are formatted for easy parsing.

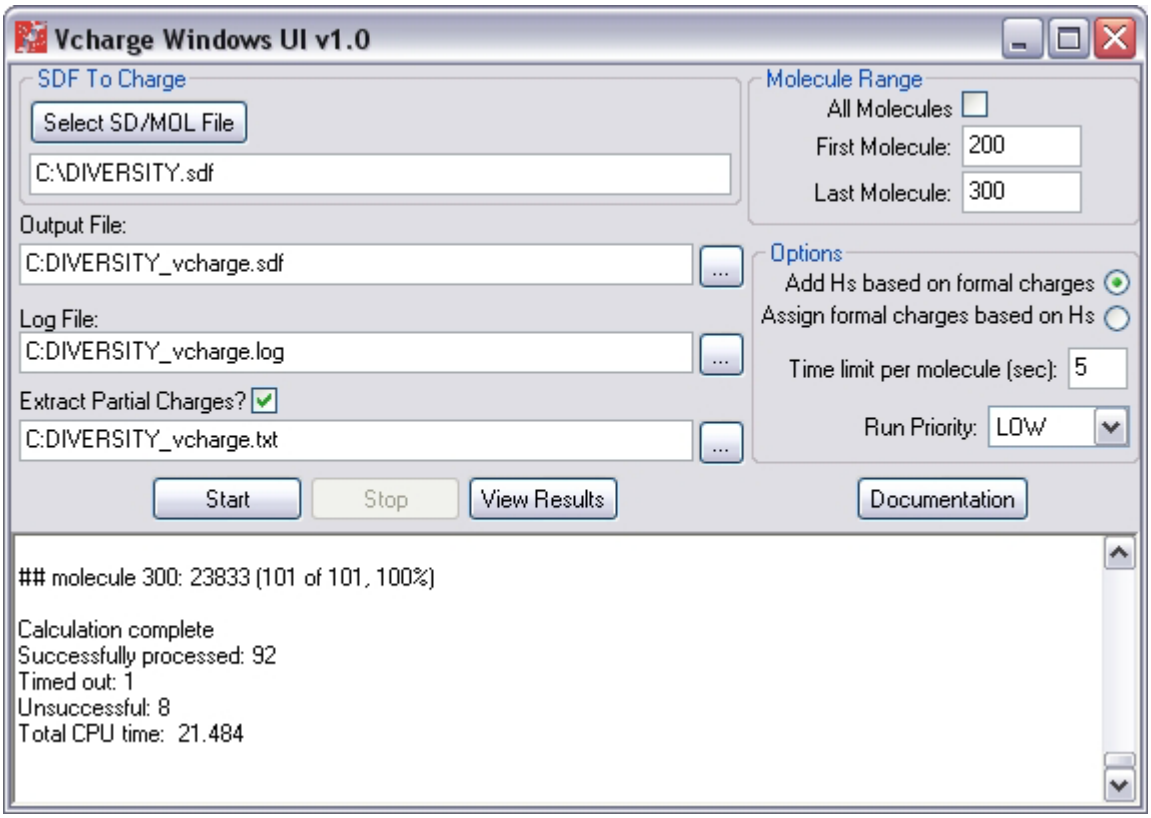
The log file records additional information about the run, including the values of parameters used in the run; the name and index of each molecule processed; warnings, errors and other informational messages; the identities of the core computational modules used; and the names of the output files. The log file is updated continually and thus provides current information on the progress of a calculation. If a file with this name already exists in the working directory, it will be overwritten, so it is advisable to specify a new log file name by using the [-o logFile] option on the command line.

Temporary files are written to /tmp; at least 40kB of space is required for smooth operation.

MS WINDOWS USER INTERFACE

Back to top

A simple graphical user interface is provided for MS Windows users, which brings together the Vcharge program, the molecular display program Vdisplay, and the Vcharge support utilities (see below).



To run Vcharge though the interface, first click on "Select SD/MOL File" in the upper left corner of the application. This will bring up a file browser so the user can select an input file. The Output File (see –p command line option) and Log File (see –o command line option) will automatically be filled in with default file names. To change these file names, type directly into the corresponding entry boxes, or use the "…" buttons to bring up a file browser. If desired, use the controls on the right-hand side of the window to change the molecule range, calculation options, and run priority from their default values. (See below.)

Click on the **Start** button to begin the calculation. The progress of the calculation will be shown in the text box on the bottom of the application.

To stop a calculation already in progress, click the **Stop** button.

To close the application, click on the X in the upper right corner of the application window.

Molecule Range Back to top

The default option is to process all molecules in the input file. To restrict the calculation to a specific range enter the molecule range in the First Molecule and Last Molecule entry boxes. (See the –f and –l command line options, above.)

Options

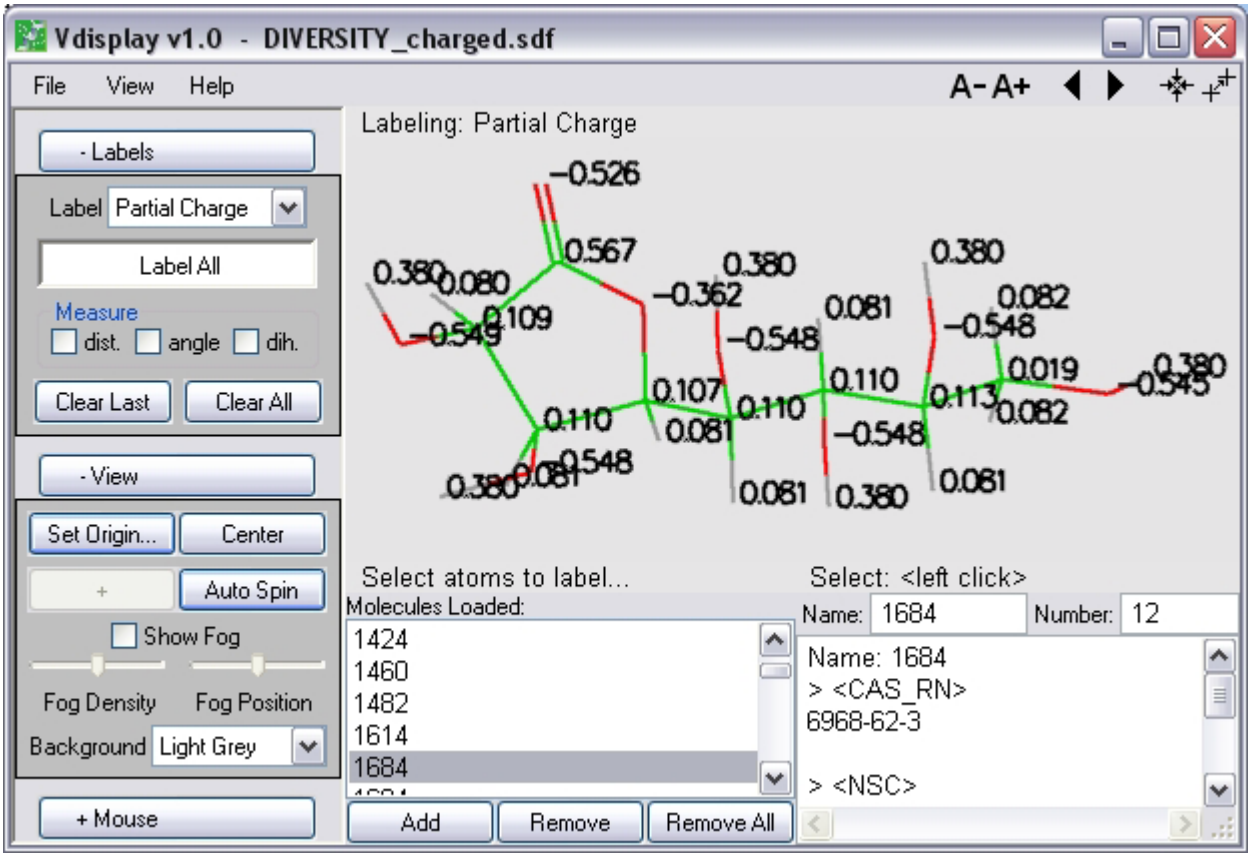
Choose "Add Hs based on formal charges" (default) if the molecules in the database are missing all or some of the hydrogens. Hydrogens will be added to satisfy the valences specified by the formal charges. If all hydrogens are present but formal charges are missing from the input choose "Assign formal charges based on Hs". (See –fc command line option, above.)

To change the maximum time Vcharge will spend on calculating the charge of any single molecule from the default value of 5 seconds, enter the desired value in the "Time limit per molecule (sec)" box. (See –tl command line option, above.)

The run priority options are LOW and NORMAL. Note that if the priority is changed to NORMAL, other programs running on the computer may slow down noticeably.

The **Documentation** button displays this file.

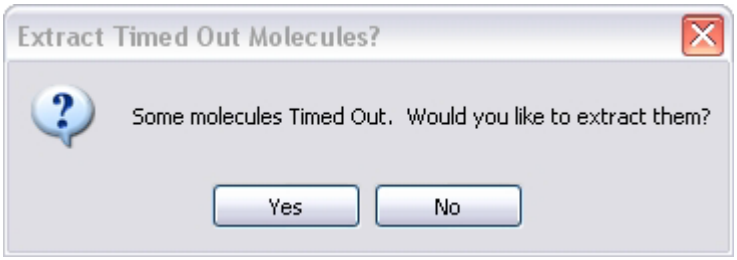
For a graphical view of the results, click **View Results** to bring up **Vdisplay** with the Vcharge Output File loaded and displaying the atomic partial charges. (See **Vdisplay** documentation for further information.)



Optional Post Processing Back to top

To save an additional file containing only the molecule name, number of atoms and partial charges for each molecule, check the "Extract Partial Charges?" box. A default filename is automatically entered, which can be changed as described above. After the Vcharge run is complete, the extractCharges utility will be run.

If Vcharge did not generate charges for any molecules in the input file, the interface will offer the option of extracting the problematic molecules from the output file and saving them separately so they can easily be reviewed. If molecules were not processed because of a cpu time-out (see extractMols –e time), the following dialog box will be shown:



Click **Yes** to extract the molecules that did not run because of a time-out into a separate SDfile; a file browser with a default file name will be provided. If Vcharge did not process any molecules for other reasons, such as a valence problem, a similar dialog box will be presented to allow the molecules to be saved to another SDfile. (See extractMols –e other.)

ERROR MESSAGES

Any errors encountered during the run will be written to the <Errors> block of the output SDfile and to the log file. The potential error messages are as follows:

1. "Unable to parse molecule" This message occurs when the software is unable to interpret the input file (SDfile or MOL) for a given molecule. This usually results from problems such as missing or corrupt lines.

2. "File format error" This message indicates that the input file for a given molecule appears to contain a violation of the MOL or SDf file format. This message usually results from data out of range. For example, the stereo parity column of the atom block should contain only 0,1,2, or 3; any other integer violates the file format.

3. "Molecule has no atoms" The input file appears to contain no atom records for the given molecule.

4. "Bond order not specified" This version of Vcharge only supports input in which all bond orders are given explicitly. Input files with bonds designated as "aromatic" or with an ambiguous order will generate this message.

Back to Top

5. “Unsupported element” A molecule contains an element for which the partial charge cannot be calculated. (See list of supported elements below).

6. “Unsupported atom type (valence error?)” An atom cannot be matched to an atom type supported by Vcharge due to what appears to be an incorrect valence.

7. “Unsupported atom type (radical): Radicals are not supported by this version of Vcharge.

8. “Unsupported atom type(s)” An atom is not of a type supported by charging algorithm due to lack of parameters. For example, there are no parameters for an oxygen atom with a positive formal charge.

9. "Time limit exceeded" This message is produced whenever the charge calculation for a molecule exceeds the time limit, 5 seconds by default or as specified with the –tl option. This can occur in molecules with unusually complex rings and many functional groups that have the potential to resonate through the ring system. Note that in some cases the molecule may run longer than the specified time limit if a given step in the process takes longer to complete than the time limit itself.

EXAMPLE

A sample input file, example.sdf, is included, along with output files generated with different command line options. The input file contains three molecules, two that run successfully with the default parameters, and one that requires the –fc option because formal charges are not specified although it contains all hydrogens (NoFormalChargesError). <I think we should change this molecule name; I initially thought this indicated an error message not specified in the error message list. How about MoleculeNeedingFC? (just a preliminary suggestion…>

- The command:

  ```
  Vcharge.exe examples.sdf
  ```

  generates the output files example_vcharge.log and example_vcharge.sdf.

- Processing the molecule called NoFormalChargesError with the –fc option allows it to be charged properly. Here the –f and –l flags are used to select just this molecule, the 3rd in the SDfile:

  ```
  % Vcharge examples.sdf –f 3 –l 3 –fc –o fcexample.log –p
     fcexample_vcharge.sdf
  ```

  This command generates the outputs files fcexample.log and fcexample_vcharge.sdf

Back to top

SUPPORTED ELEMENTS FOR ASSIGNMENT OF PARTIAL ATOMIC CHARGES

Hydrogen
Carbon
Oxygen
Nitrogen
Fluorine
Phosphorous
Sulfur
Chlorine
Bromine
Iodine

SUPPORT UTILITIES

extractMols analyzes the SDfile generated by **Vcharge** and writes an SDFile containing any molecules for which **Vcharge** did not calculate charges. This makes it easy to check and correct any molecules that may require special attention and then reprocess them with Vcharge.

The most common reasons that charges are not computed for a molecule are that it includes an element or atom type for which Vcharge does not have parameters, or that it is unusually complex therefore exceeds the Vcharge time limit.

extractMols is invoked via the following command line:

```
extractMols <Vcharge output log file> -s <Vcharge output sdf file>
-e [time, other, all] [-o outputfile] [-h, -help]
```

where <Vcharge output log file> is the log file generated by **Vcharge** and <Vcharge output sdf file> is the output SDfile generated by **Vcharge**.

Options:

time: extract only molecules for which Vcharge timed out. This is the default option

other: extract molecules that had errors, but not molecules that timed out.

all: extract both molecules that had errors and molecules that timed out

outputfile: Name of the SDfile that will contain the extracted molecules
default: <Vcharge output sdf file>_extracted.sdf

–h, -help: prints reminder for usage of extractMols

Back to top

extractCharges extracts the partial charges from the output SDfile generated by Vcharge and writes them to a text file which is formatted for easy parsing. It lists the name of each molecule followed by its atomic partial charges, one per line. This utility is provided to simplify viewing of calculated partial charges. (The free Vdisplay program also provides an easy graphical way to view Vcharge outputs.)

extractCharges is invoked via the following command line:

```
extractCharges <Vcharge output sdf file> [-o outputfile]
[-h, -help]
```

where <Vcharge output sdf file> is the output SDfile generated by **Vcharge**.

Options:

outputfile: name of the file to be produced by extractCharges
default: <Vcharge output sdf filename without .sdf extension>.txt
–h, -help: prints reminder for usage of extractCharges

The output file is formatted as follows, for each molecule:

line 1: name: <molecule name> <ncharge>
lines 2 through ncharge – the calculated partial charges

where <molecule name> is the molecule name extracted from the SDfile and <ncharge> is the number of partial charges calculated for the molecule. Note that ncharge equals the number of atoms in the molecule, except if **Vcharge** does not generate charges for the molecule, which case ncharge equals 0.

Sample output

```
name: Tyrosine 24
-0.113
-0.116
0.128
-0.116
-0.113
-0.008
-0.148
0.090
0.641
-0.627
-0.627
-0.515
-0.605
0.085
0.107
0.107
0.107
0.107
0.088
0.088
0.386
0.351
0.351
0.351
```

where “Tyrosine” is the molecule name and “24” is ncharge.

KNOWN BUGS

Back to top

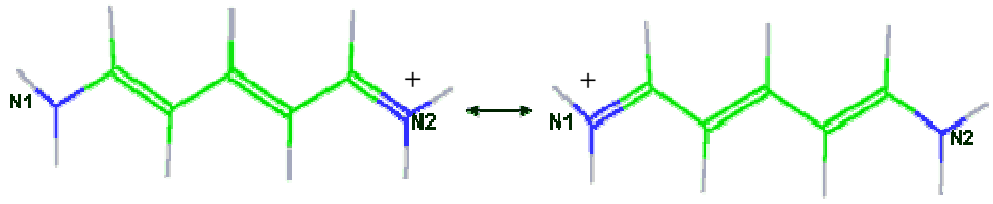
There are currently no known bugs with Vcharge.

COMPUTATIONAL METHODOLOGY AND VALIDATION

Method

Vcharge is an electronegativity equalization method in which the electronegativity of each atom depends upon its atomic number, hybridization, and bonding environment within the molecule. Electronegativities and hardnesses are averaged over alternate resonance forms, which are automatically detected. Appropriate charging of ionized groups is ensured by constraining formal charges to remain localized near formally charged atoms. The electronegativity equalization problem is solved efficiently by an adaptation of the method of Lagrangian multipliers. The adjustable parameters are tuned so the computed charges reproduce electrostatic potentials from Hartree-Fock 6-31G* calculations at CHELPG sampling points for a training set of approximately 300 molecules.

Vcharge automatically generates alternate resonance forms in order to ensure that equal charges are assigned to chemically equivalent atoms. For example, nitrogens N1 and N2 illustrated below would not be considered equivalent if alternate resonance forms were not considered.



The ionization and tautomer states specified in the input SDfile are used without modification.

Validation

The VC/2004 parameterization of Vcharge yields charges that accurately reproduce Hartree-Fock 6-31G* potentials at CHELPG sampling points. The potentials typically agree to within an RMSD of ~4 kcal/(mol-e). Tests on several molecules for which published data are available indicate that Vcharge is as accurate, relative to 6-31G* potentials, as those obtained by more time-consuming methods, such as RESP and AM1/BCC.

VC/2004 charges are strikingly similar to those in widely used force fields, such as AMBER and CHARMM22. Thus, for the 20 common amino acids, VC/2004 partial charges match CHARMM22 and AMBER94 with RMSDs of only 0.09 e and 0.11 e, respectively, and with correlation coefficients of 0.93 and 0.89. These differences are comparable to the differences between CHARMM22 and AMBER94 themselves, RMSD 0.10, correlation 0.90. Therefore, it is reasonable to use VC/2004 charges for candidate ligands interacting with proteins treated by the CHARMM22 or AMBER94 force fields.

REFERENCES

Back to top

Main Citation: Fast assignment of accurate partial atomic charges. An electronegativity equalization method that accounts for alternate resonance forms; Gilson,M.K., Gilson,H.S.R. & Potter,M.J.; *J. Chem. Inf. Comput. Sci*. **2003**, 43(6), 1982-1997.

Background Citations

Bayly, C. I.; Cieplak, P.; Cornell, W. D.; Kollman, P.A.; *J. Phys. Chem.* **1993**, 97, 10269-1028.

Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I;. *J. Comput. Chem.* **2000**, 21, 132-146.

Jakalian, A.; Jack, D. B.; Bayly, C. I.; *J. Comput. Chem.* **2002**, 23, 1623-1641.

Breneman, C. M.; Wiberg, K. B.; *J. Comput. Chem.* **1990**, 11, 361.

Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S. J.; Windus, T. L.; Dupuis, M.; Montgomery, J. A; *J. Comp. Chem.* **1993**, 14, 1347-1363.

Cornell, W.; Cieplak, P.; Bayly, C.; Gould, I.; Merz,Jr., K.; Ferguson, D.; Spellmeyer, D.; Caldwell, T. F. J.; P.A.,; Kollman; *J. Am. Chem. Soc.* **1995**, 117, 5179-5197.

Molecular Simulations Inc. Waltham, MA. CHARMm Version 22., **1992**.

MacKerell, Jr., A. D.; Wiokiewicz-Kuczera, J.; Karplus, M.; *J. Am. Chem. Soc.* **1995**, 117, 11946-11975