Thermo Manual

Simone Conti

September 20, 2019

Contents

1 Introduction to Thermo

Thermo calculates an approximation of thermodynamics quantities like entropy and free energy for molecular systems. The calculation is based on closed-form expressions derived from statistical mechanics, essentially obtained under the rigid rotor and harmonic oscillator (RRHO) approximations.

Thermo evaluates the internal energy, the entropy and the free energy of molecules from their molecular properties like the mass, the moments of inertia, or the vibrational frequencies.

In the following, it will be described how to download and install the code and how to use it with some examples. Next, a Theory chapter aims at describing the underlying statistical mechanics to understand how this works and under which approximations and limits the obtained results can be interpreted.

1.1 License and Copyright

This code and manual has been written by Simone Conti, partially while at work a the University of Strasbourg:

1.2 Citation References

If you use the Thermo software in scientific publications, please cite[1]

> S. Conti and M. Cecchini "Predicting molecular self-assembly at surfaces: a statistical thermodynamics and modeling approach", PCCP, 2016, 18, pp. 1480-31493

2 Install instructions

The last version of the code can be downloaded from git via:

`git clone https://github.com/SimoneCnt/thermo.git thermo`

which will copy the latest available version inside the `thermo` directory. If you do not have git installed and you want to install it, take a look at http://git-scm.com Otherwise you can get the full code as a zip archive directly from the git project page at https://github.com/SimoneCnt/thermo

Once downloaded the installation is straighforward. The code is written in standard C and the compilation is managed by a CMake script. So to compile the code the following list of commands will easily do the work:

```
cd thermo
mkdir build
cd build
cmake ..
make release
```

After these commands, you should find the `thermo` binary inside the `build` directory. To check if the compilation went file, you can issue the `make check` command (always from inside the build directory).

One option of Thermo is to give as input a hessian matrix to evaluate the normal mode frequences. To activate this option you need to link Thermo to a linear algebra (lapack) library. On MacOS the linking should be automatic (the developer tools in MacOS already contains an efficient lapack library). On Linux you need first to install an efficient lapack library (e.g. MKL from Intel, freeware), and second to specify the include directory and the link options to the Thermo build script. You have two options: hope it will work automatically (unlikely), or explicitly specifify the include and link directive through the `LAPACK_INC` and `LAPACK_LINK` environmental variables. For example, on my linux machine to link to MKL I have:

```
LAPACK_LINK=-L/opt/compilers_and_libraries_2017.0.098/linux/mkl/lib/intel64
    -Wl,--no-as-needed -lmkl_gf_lp64 -lmkl_gnu_thread -lmkl_core -lgomp
    -lpthread -lm -ldl
LAPACK_INCL=-m64 -I/opt/compilers_and_libraries_2017.0.098/linux/mkl/include
```

This is higly depenent on your system and which lapack library you use.

For any problem, feel free to refer to https://github.com/SimoneCnt/thermo/issues or directly to simonecnt@gmail.com.

3 Usage

To use Thermo you need to set some command line options and to write a thermo input file. For each molecule you wanto to study, you need to set up one thermo input file, which contains the molecular properties used to evaluate the entropy, the free energy and the internal energy. The thermo input is described first, the command line options follow.

3.1 Thermo input

The thermo input file is a simple text file which constains `key = value` pairs. Based on the "key" set, different contributions to the total free energy are evaluated. Generally speaking (see the Theory section for more details), the free energy (as the entropy and the internal energy) is split in an energetic, a translational, rotational and vibrational contributions. Usually, a molecule in solution has three rotational and three translational degrees of freedom and 3N-6 vibrations (N is the number of atoms). This is not valid for linear molecules or for systems with constrained degrees of freedom, like molecules adsorbed on a surface. The keys described below allow you to define the repartition of the 3N degrees of freedom and the molecular properties, like mass and moment of inertia, necessary to evaluate the free energy.

The keys you can set in the thermo input file are:

- **temperature**: The temperature in kelvin. Default 300 K.
- **nmoles**: The number of moles in mol. Default 1 mol.
- **volume**: Volume in liters. Together with the number of moles defines the concentration. Default 1 l.
- **energy**: The ground state energy in kcal/mol. Default 0 kcal/mol.
- **mass**: Molecular mass in g/mol. Default zero.
- **translations**: Number of translational degrees of freedom. Default 3.
- **rotations**: Number of rotational degrees of freedom. Default 3. This variable has to be followed by the moments of inertia one per line, in equal number of the specified number of rotational degrees of freedom, expressed in g/mol/A^2.
- **sigma**: Symmetry number. Default 1.
- **vibrations**: Number of vibrational degrees of freedom. Default zero. This variable has to be followed by the vibrational frequencies, one per line, in equal number of the specified umber of vibrational degrees of freedom, expressed in cm^{-1}.
- **hessian**: Name of the file where a hessian matrix can be read. The matrix will be internally diagonalized (if Thermo is compiled with lapack support), and the normal mode frequences calculated. The hessian and vibrations keywords are mutually exclusive. Default none.

For example, if you want to specify the temperature you can write in the input file:

```
temperature = 310
```

or for the moments of inertia:

```
rotations = 3
moment1
moment2
moment3
```

The keywords are identified based only on the first four charachters, so these three statements are actually identical:

```
temperature = 300
temp = 300
temperatura = 300
```

3.2 Command line options

The mode of working of Thermo id defined by few command line options. The basic is via the `--A` option followed by the name of the thermo input file. Thus running

```
thermo --A myinput.thermo
```

will calculate all internal energy, entropy and free energy based on the options set in the `myinput.thermo` imput file. The `--B` option is essetianlly the same as `--A`. You can use both

`--A` and `--B` in the same run. This is useful in particular with the `--stechio a:b` option. The parameters `a` and `b` are the stechiometric coefficient for the reaction $aA \rightleftharpoons bB$. Thus via this command line

```
thermo --A monomer.thermo --B dimer.thermo --stechio 2:1
```

thermo will evaluate the free energy of the monomer, the dimer, and the free energy difference for the dimerization reaction.

More classical command line options, `--out outfile.out` redirect the thermo output to the `outfile.out` file, `--help` print an hopefully useful help, and `--version` print the current version of the thermo code.

Still to document: `--cumul`, `--vdos`, `--dnu`. These essentially create and write to file the vibrational density of states (VDOS) and the cumulative vibrational free energy.

4 Examples

Here a small description on how to use the code is given. Inside the `examples` directory it is possible to find some ready to use input files for some more applications. A reference output is normally present with the `.ref` suffix.

4.1 Water

The easiest system we can take as example is water. Here a sample input script to evaluate the thermodynamic properties of an hypotetical box of 22.465 liters which contains 1 mol of gas water at 298.15 kelvin (1 atm pressure). Each water molecule will have three translational degrees of freedom, three rotational (with the associated moments of inertia), and $3N - 6 = 3$ vibrations (and associated frequencies). The simmetry number is also reported: since water symmetry pointgroup is C2v, the symmetry number is 2. The mass is also specified.

```
# Set temperature in Kelvin
temperature = 298.15

# Set number of mols and volume (1 atm pressure)
nmols = 1
volume = 22.465

# Mass in g/mol
mass = 18.01528

# Translational degrees of freedom
translational = 3

# Rotational degrees of freedom and moments of inertia in g/mol Ang^2
rotational = 3
1.7704
0.6169
1.1535
```

```
# Symmetry number
sigma = 2

# Number of vibrations and their frequencies in cm-1
vibrations = 3
1635.618
3849.420
3974.869
```

Save this input as e.g. `water.inp` and run it with `thermo -A water.inp` The output will print all parsed options and at the end all thermodynamic quantites, like energy, entropy, free energy, and chemical potential, are printed. For example, the evaluated translational molar entropy is evaluated to be 32.452 cal mol-1 K-1 and the zero point vibrational energy to be 13.524 kcal mol-1.

Input and reference output can be found in the `examples/water` subdirectory.

4.2 Dimerization of Insulin

A second example in based on the reference paper by B. Tidor and M. Karplus,[2] published in 1994:

> B. Tidor and M.Karplus, "The Contribution of Vibrational Entropy to Molecular Association: The Dimerization of Insulin", Journal of Molecular Biology, 238(3) 1994 pp. 405-414

In this work the effect of vibrations in the stabilization of the insulin dimer is discussed as example of how the vibrational entropy can partially counterbalance the net lost in translational and rotational degrees of freedom. Translational and rotational contribution are calculated in good agreement with the data reported in the paper, while the vibrational ones are incorrect due to the limited number of frequences reported in the paper itself.

The input files are inside the `examples/insulin` subdirectory. Since we want to study the dimerization reaction, it is possible to run at the same time both the monomer and the dimer, and, thanks to the '–stechio" command line option, the differences for the dimerization reaction are automatically printed. The Thermo command line can thus be:

```
thermo -A monomer.inp -B dimer.inp --stechio 2:1 > insulin.out
```

The reference output is `insulin.out.ref`. At the moment only reactions between two species can be studied (so only `-A` and `-B` command are available).

4.3 Free Energy Difference between Peptides and Proteins Conformers

Three examples are included to study the difference in free energy between peptides and proteins conformes:

- alanine dipeptide (`examples/diala` directory) in its c7 equatorial (c7eq) and c7 axial (c7ax) conformations;
- beta hairpin from protein G (`examples/bhp`) in the native beta-harpin (bhp1) and in a three-stranded beta sheet (bhp2) conformations;

- the converter of the biomolecular motor myosin VI (`examples/conv`) in the pre-powerstroke (pps) and rigor-like (rig) conformations.

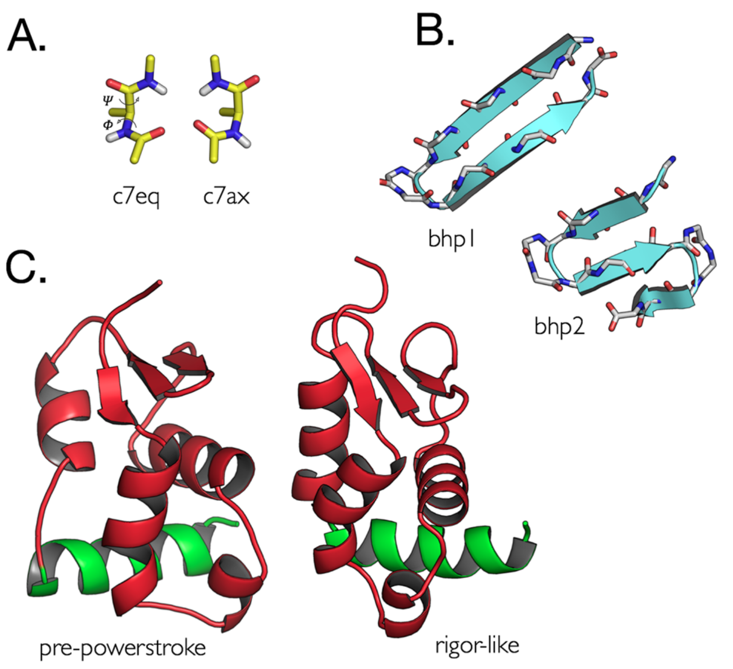

Figure 1: Representation of the three examples to study the quantum correction to the conformational free energy difference: the alanine dipeptide, the beta hairpin of protein G and the converter of myosin VI. Reproduced from ref. 3.

These examples, see Fig. 1, are taken form the work of M. Cecchini, JCTC 2015,[3] where they were used to develop a quantum correction to the classical conformational free energy difference.

The quantum correction is by default calculated by Thermo; for example, for the myosin converter

```
thermo -A rig.inp -B pps.inp --stechio 1:1
```

Thermo calculates a quantum correction of 1.09 kcal mol-1, in perfect agreement with the Cecchini paper.

4.4 Solvation Entropy

Thermo implements the "Solvation Entropy Made Simple" approach of Alejandro J. Garza to compute solvation entropies of small molecules in different solvents.[4] This approach include three different approximations of the solvation entropy, which are called in this code `Omega`, `Epsilon` and `Eps,Alpha`.

To compute solvation entropies you need two input files, one for the molecule in the gas state, one for the molecule in solution. For example, for the solvation entropy of methanol in water the input for the methanol in gas would look like:

```
> # Methanol gas
> temperature = 298.15
> pressure = 0.9869233
> translations = 3
> rotations = 3
> 3.98199904
> 20.50103433
> 21.27739477
> mass = 32.04
> sigma = 1
> energy = 0.0
> vibrations = 0
```

For the solution:

```
> # Methanol in water
> temperature = 298.15
> concentration = 1 unit M
> translations = 3
> rotations = 3
> 3.98199904
> 20.50103433
> 21.27739477
> mass = 32.04
> sigma = 1
> energy = 0.0
> vibrations = 0
> vvdw = 35.24161
> bbox = 119.5194
> rgyr = 1.1944
> solvent = water
```

The differences are in the concentration/pressure (1bar in gas phase, 1M in solution), and in the addition of 4 keywords for the solution: the van der Waals volume (vvdw), the surface of the bounding box (bbox), the radius of gyration (rgyr), and the name of the solvent. Running the first script you would get a total molar entropy of 53.373 cal/mol/K (Sgas). Running the second the total molar entropy becomes 46.994 cal/mol/K (Ssolv), but three more lines are present in the output, showing the solvation Omega, Epsilon and Eps,Alpha solvation entropies.

The columns `dS_trans` and `dS_rot` contain the corrections to the translational and rotational entropies, the column `dS_cav` instead contains the cavity entropy computed in the three different methods. Follow `dS_tot` with the total solvation entropy correction, and `S_totcl` and `S_totqm` which sums the solvation entropy corrections to the total entropy. Last the column `-TdS_tot`, `F_totcl`, `F_totqm` contain the free energies. For methanol in water, the solvation entropies corrections (dSsolv) are -21.066, -20.095 and -19.474 cal/mol/K for the three methods.

The final solvation entropy can be obtained as (Ssolv + dSsolv) - Sgas, obtaining -27.445, -26.474, and -25.853 cal/mol/K. The experimental value is -27.2 cal/mol/K.

4.4.1 Condensation entropy and vaporization enthalpy

In a very similar way it is possible to compute entropy of moving from the gas phase to the liquid (condensation) and the vaporization enthalpy. For the condensation entropy, the final state is a molecule in solution in a solvent composed by the same molecule, at a concentration that can be derived from the density of the liquid. For example, for the condensation entropy of gas methanol into liquid methanol, it is enough to change the concentration in the previous input file from `1M` to `0.796 unit g/ml`, and change the solvent line from water to methanol.

If the condensation – or better the evaporation – entropy is computed at the boiling temperature (Tb) of the liquid, we can take advantage of the fact that at Tb the vaporization free energy is equal to zero, which implies the vaporization enthalpy is equal to the vaporization entropy divided by the boiling temperature. To compute the vaporization entropy at Tb, change change the temperature field in the input files of both the gas phase and the liquid phase.

You can run these examples from the methanol directory:

```
# Solvation entropy of methanol from 1bar to 1M in water
thermo -A methanol-gas.thermo -B methanol-water.thermo \
    --stechio 1:1 --raw -o solution-water.out

# Vaporization entropy of methanol from the liquid at 25C
thermo -A methanol-gas.thermo -B methanol-liq.thermo \
    --stechio 1:1 --raw -o vaporization.out

# Vaporization entropy of methanol from the liquid at boiling temperature
thermo -A methanol-gas-tb.thermo -B methanol-liq-tb.thermo \
    --stechio 1:1 --raw -o vaporization-tb.out.ref
```

5 Theory

This part is still a draft. Please, see the McQuarrie "Statistical Mechanics" book[5] for more information.

5.1 Partition Function

$$Q = \frac{q^N}{N!} \tag{1}$$

$$\ln Q = N \ln q - N \ln N + N = N \ln \frac{qe}{N} \tag{2}$$

$$q = q_{tr}\, q_{rot}\, q_{vib}\, q_{elec} \tag{3}$$

Translational degrees of freedom:

$$q_{tr} = \left(\frac{2\pi m k_B T}{h^2}\right)^{t/2} V \tag{4}$$

$$\ln q_{tr} = \frac{t}{2}\ln\left(\frac{2\pi m k_B T}{h^2}\right) + \ln V \tag{5}$$

$$\frac{\partial}{\partial T}\ln q_{tr} = \frac{t}{2}\frac{1}{T} \tag{6}$$

Rotational:

$$q_{rot} = \frac{\sqrt{\pi}}{\sigma}\left(\frac{8\pi^2 k_B T}{h^2}\right)^{r/2}\left(\prod_{i=1}^{r} I_i\right)^{1/2} \tag{7}$$

$$\ln q_{rot} = \frac{1}{2}\ln\frac{\pi}{\sigma^2} + \frac{r}{2}\ln\frac{8\pi^2 k_B T}{h^2} + \frac{1}{2}\sum_{i=1}^{r}\ln I_i \tag{8}$$

$$\frac{\partial}{\partial T}\ln q_{rot} = \frac{r}{2}\frac{1}{T} \tag{9}$$

Vibrational (classical limit):

$$q_{vib,cl} = \prod_{i=1}^{f}\frac{k_B T}{h\nu_i} \tag{10}$$

$$\ln q_{vib,cl} = \sum_{i=1}^{f}\ln\frac{k_B T}{h\nu_i} \tag{11}$$

$$\frac{\partial}{\partial T}\ln q_{vib,cl} = f\frac{1}{T} \tag{12}$$

Vibrational (quantum):

$$q_{vib,qm} = \prod_{i=1}^{f}\frac{\exp\frac{h\nu}{2k_B T}}{1-\exp\frac{h\nu}{k_B T}} = \prod_{i=1}^{f}\left[2\sinh\left(\frac{h\nu}{2k_B T}\right)\right]^{-1} \tag{13}$$

$$\ln q_{vib,qm} = -\sum_{i=1}^{f}\ln\left[2\sinh\left(\frac{h\nu}{2k_B T}\right)\right] \tag{14}$$

$$\frac{\partial}{\partial T}\ln q_{vib,qm} = \frac{1}{T}\sum_{i=1}^{f}\frac{\frac{h\nu}{2k_B T}}{\tanh\left(\frac{h\nu}{2k_B T}\right)} \tag{15}$$

Electronic:

$$q_{elec} = \exp\left(-\frac{E_m}{k_B T}\right) \tag{16}$$

$$\ln q_{elec} = -\frac{E_m}{k_B T} \tag{17}$$

$$\frac{\partial}{\partial T}\ln q_{elec} = \frac{E_m}{k_B T}\frac{1}{T} \tag{18}$$

5.2 Helmholtz Free energy

$$F = U - TS \tag{19}$$

$$\begin{aligned} F &= -k_B T \ln Q = -Nk_B T \ln \frac{qe}{N} \quad (20)\\ &= -Nk_B T \ln \frac{q_{tr} e}{N} - Nk_B T \ln q_{rot} - Nk_B T \ln q_{vib} - Nk_B T \ln q_{elec} \quad (21)\\ &= F_{tr} + F_{rot} + F_{vib} + F_{elec} \quad (22) \end{aligned}$$

$$\begin{aligned} F_{tr} &= -Nk_B T \ln \frac{q_{tr} e}{N} = -Nk_B T \ln \left[\left(\frac{2\pi m k_B T}{h^2} \right)^{t/2} \frac{Ve}{N} \right] \quad (23)\\ &= -Nk_B T \left(\frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + 1 - \ln \frac{N}{V} \right) \quad (24) \end{aligned}$$

$$F_{rot} = -Nk_B T \ln q_{rot} = -Nk_B T \left[\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} \ln I_i \right] \tag{25}$$

$$F_{vib,cl} = -Nk_B T \ln q_{vib,cl} = -Nk_B T \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} \tag{26}$$

$$F_{vib,qm} = -Nk_B T \ln q_{vib,qm} = Nk_B T \sum_{i=1}^{f} \ln \left[2 \sinh \left(\frac{h\nu}{2k_B T} \right) \right] \tag{27}$$

$$F_{elec} = -Nk_B T \ln q_{elec} = NE_m \tag{28}$$

5.3 Molar Helmholtz free energy

$$\begin{aligned} \mu = F_m &= \frac{\partial F}{\partial N} \quad (29)\\ &= \frac{\partial F_{tr}}{\partial N} + \frac{\partial F_{rot}}{\partial N} + \frac{\partial F_{vib}}{\partial N} + \frac{\partial F_{elec}}{\partial N} \quad (30)\\ &= \mu_{tr} + \mu_{rot} + \mu_{vib} + \mu_{elec} \quad (31) \end{aligned}$$

$$\mu_{tr} = \frac{\partial F_{tr}}{\partial N} = -k_B T \frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + k_B T \ln \frac{N}{V} = \frac{F_{tr}}{N} + k_B T \tag{32}$$

$$\mu_{rot} = \frac{\partial F_{rot}}{\partial N} = -k_B T \left[\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} \ln I_i \right] = \frac{F_{rot}}{N} \tag{33}$$

$$\mu_{vib,cl} = \frac{\partial F_{vib,cl}}{\partial N} = -k_B T \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} = \frac{F_{vib,cl}}{N} \tag{34}$$

$$\mu_{vib,qm} = \frac{\partial F_{vib,qm}}{\partial N} = k_B T \sum_{i=1}^{f} \ln \left[2 \sinh \left(\frac{h\nu}{2k_B T} \right) \right] = \frac{F_{vib,qm}}{N} \tag{35}$$

$$\mu_{elec} = \frac{\partial F_{elec}}{\partial N} = E_m = \frac{F_{elec}}{N} \tag{36}$$

5.4 Internal energy

$$\begin{align} U &= k_B T^2 \frac{\partial}{\partial T} \ln Q = N k_B T^2 \frac{\partial}{\partial T} \ln q \tag{37} \\ &= N k_B T^2 \frac{\partial}{\partial T} \ln q_{tr} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{rot} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{elec} \tag{38} \\ &= U_{tr} + U_{rot} + U_{vib} + U_{elec} \tag{39} \end{align}$$

$$U_{tr} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{tr} = N k_B T \frac{t}{2} \tag{40}$$

$$U_{rot} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{rot} = N k_B T \frac{r}{2} \tag{41}$$

$$U_{vib,cl} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib,cl} = N k_B T f \tag{42}$$

$$U_{vib,qm} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib,qm} = N k_B T \sum_{i=1}^{f} \frac{\frac{h\nu}{2k_B T}}{\tanh\left(\frac{h\nu}{2k_B T}\right)} \tag{43}$$

$$U_{elec} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{elec} = N E_m \tag{44}$$

5.5 Molar internal energy

$$U_m = \frac{\partial U}{\partial N} = \frac{\partial U_{tr}}{\partial N} + \frac{\partial U_{rot}}{\partial N} + \frac{\partial U_{vib}}{\partial N} + \frac{\partial U_{elec}}{\partial N} = U_{m,tr} + U_{m,rot} + U_{m,vib} + U_{m,elec} \tag{45}$$

$$U_{m,tr} = \frac{\partial U_{tr}}{\partial N} = k_B T \frac{t}{2} = \frac{U_{tr}}{N} \tag{46}$$

$$U_{m,rot} = \frac{\partial U_{rot}}{\partial N} = k_B T \frac{r}{2} = \frac{U_{rot}}{N} \tag{47}$$

$$U_{m,vib,cl} = \frac{\partial U_{vib,cl}}{\partial N} = k_B T f = \frac{U_{vib,cl}}{N} \tag{48}$$

$$U_{m,vib,qm} = \frac{\partial U_{vib,qm}}{\partial N} = k_B T \sum_{i=1}^{f} \frac{\frac{h\nu}{2k_B T}}{\tanh\left(\frac{h\nu}{2k_B T}\right)} = \frac{U_{vib,qm}}{N} \tag{49}$$

$$U_{m,elec} = \frac{\partial U_{elec}}{\partial N} = E_m = \frac{U_{elec}}{N} \tag{50}$$

5.6 Entropy

$$S = \frac{\partial}{\partial T}\left(k_B T \ln Q\right) = k_B \ln Q + k_B T \frac{\partial}{\partial T} \ln Q = N k_B \ln \frac{qe}{N} + N k_B T \frac{\partial}{\partial T} \ln q \tag{51}$$

$$= \left(N k_B \ln \frac{q_{tr} e}{N} + N k_B T \frac{\partial}{\partial T} \ln q_{tr}\right) + \left(N k_B \ln q_{rot} + N k_B T \frac{\partial}{\partial T} \ln q_{rot}\right) + \left(N k_B \ln q_{vib} + N k_B T \frac{\partial}{\partial T} \ln q_{vib}\right) + \left(N k_B \ln q_{elec} + N k_B T \frac{\partial}{\partial T} \ln q_{elec}\right) \tag{52}$$

$$= S_{tr} + S_{rot} + S_{vib} + S_{elec} \tag{53}$$

$$S_{tr} = N k_B \ln \frac{q_{tr} e}{N} + N k_B T \frac{\partial}{\partial T} \ln q_{tr} = N k_B \left(\frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + \frac{t+2}{2} - \ln \frac{N}{V}\right) \tag{54}$$

$$S_{rot} = N k_B \ln q_{rot} + N k_B T \frac{\partial}{\partial T} \ln q_{rot} = N k_B \left(\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} \ln I_i + \frac{r}{2}\right) \tag{55}$$

$$S_{vib,cl} = N k_B \ln q_{vib,cl} + N k_B T \frac{\partial}{\partial T} \ln q_{vib,cl} = N k_B \left(f + \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i}\right) \tag{56}$$

$$S_{vib,qm} = N k_B \ln q_{vib,qm} + N k_B T \frac{\partial}{\partial T} \ln q_{vib,qm} = N k_B \sum_{i=1}^{f} \left[\frac{\frac{h\nu}{2k_B T}}{\tanh \frac{h\nu}{2k_B T}} - \ln\left(2 \sinh \frac{h\nu}{2k_B T}\right)\right] \tag{57}$$

$$S_{elec} = N k_B \ln q_{elec} + N k_B T \frac{\partial}{\partial T} \ln q_{elec} = 0 \tag{58}$$

5.7 Molar Entropy

$$S_m = \frac{\partial S}{\partial N} = \frac{\partial S_{tr}}{\partial N} + \frac{\partial S_{rot}}{\partial N} + \frac{\partial S_{vib}}{\partial N} + \frac{\partial S_{elec}}{\partial N} = S_{m,tr} + S_{m,rot} + S_{m,vib} + S_{m,elec} \tag{59}$$

$$S_{m,tr} = \frac{\partial S_{tr}}{\partial N} = k_B \left(\frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + \frac{t}{2} - \ln \frac{N}{V} \right) = \frac{S_{tr}}{N} - k_B \tag{60}$$

$$S_{m,rot} = \frac{\partial S_{rot}}{\partial N} = k_B \left(\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} \ln I_i + \frac{r}{2} \right) = \frac{S_{rot}}{N} \tag{61}$$

$$S_{m,vib,cl} = \frac{\partial S_{vib,cl}}{\partial N} = k_B \left(f + \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} \right) = \frac{S_{vib,cl}}{N} \tag{62}$$

$$S_{m,vib,qm} = \frac{\partial S_{vib,qm}}{\partial N} = k_B \sum_{i=1}^{f} \left[\frac{\frac{h\nu}{2k_B T}}{\tanh \frac{h\nu}{2k_B T}} - \ln \left(2 \sinh \frac{h\nu}{2k_B T} \right) \right] = \frac{S_{vib,qm}}{N} \tag{63}$$

$$S_{m,elec} = \frac{\partial S_{elec}}{\partial N} = 0 \tag{64}$$

References

(1) Conti, S.; Cecchini, M. Predicting Molececular Self-Assembly at Surfaces: A Statistical Thermodynamics and Modeling Approach. *Physical Chemistry Chemical Physics* **2016**, *18*, 1480–31493.

(2) Tidor, B.; Karplus, M. The Contribution of Vibrational Entropy to Molecular Association: The Dimerization of Insulin. *Journal of molecular biology* **1994**, *238* (3), 405–414.

(3) Cecchini, M. Quantum Corrections to the Free Energy Difference Between Peptides and Proteins Conformers. *Journal of Chemical Theory and Computation* **2015**, *11* (9), 4011–4022.

(4) Garza, A. J. Solvation Entropy Made Simple. *Journal of Chemical Theory and Computation* **2019**, *15*, 3204–3214.

(5) McQuarrie, D. A. *Statistical Mechanics*; University Science Books, 2000.