Thermo Manual
v1.0.x

Simone Conti

Chapter 1

Thermo

ToDo

Chapter 2

License

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

> S. Haar, A. Ciesielski, J. Clough, H. Yang, R. Mazzaro, F. Richard, S. Conti, N. Merstorf, M. Cecchini, V. Morandi, C. Casiraghi, and P. Samorì. "A supramolecular strategy to leverage the liquid-phase exfoliation of graphene in presence of surfactants: unraveling the role of the length of fatty acids.", Small, 2015, 11(14), pp. 1736–1736

If you use the quantum correction of the free energy difference, please cite [2]

> M. Cecchini, "Quantum Corrections to the Free Energy Difference between Peptides and Proteins Conformers", Journal of Chemical Theory and Computation, 2015 ASAP

Chapter 3

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
- you do not want to install it, you can get the 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:

```
mkdir build
cd build
cmake ..
make
```

If you are missing CMake, you can get if from http://www.cmake.org.

Once the compilation finishes, you can find the `thermo` binary inside the `build` directory.

This procedure has been tested on Linux (Ubuntu) and MacOsX machines. On Ubuntu, and probably also in other Linux distributions, you can find both CMake and git on the standard packaging tools. So in Ubuntu this line should smoothly install both applications:

```
sudo apt-get install git cmake
```

No test has been performed on Windows machines.

Chapter 4

Theory

This part is still a draft. Please, see the McQuarrie "Statistical Mechanics" book [3] for more information.

Partition Function

$$Q = \frac{q^N}{N!} \tag{4.1}$$

$$\ln Q = N \ln q - N \ln N + N = N \ln \frac{qe}{N} \tag{4.2}$$

$$q = q_{tr}\ q_{rot}\ q_{vib}\ q_{elec} \tag{4.3}$$

Translational degrees of freedom:

$$q_{tr} = \left(\frac{2\pi m k_B T}{h^2}\right)^{t/2} V \tag{4.4}$$

$$\ln q_{tr} = \frac{t}{2} \ln\left(\frac{2\pi m k_B T}{h^2}\right) + \ln V \tag{4.5}$$

$$\frac{\partial}{\partial T} \ln q_{tr} = \frac{t}{2}\frac{1}{T} \tag{4.6}$$

Rotational:

$$q_{rot} = \frac{\sqrt{\pi}}{\sigma}\left(\frac{8\pi^2 k_B T}{h^2}\right)^{r/2}\left(\prod_{i=1}^{r} I_i\right)^{1/2} \tag{4.7}$$

$$\ln q_{rot} = \frac{1}{2}\ln\frac{\pi}{\sigma^2} + \frac{r}{2}\ln\frac{8\pi^2 k_B T}{h^2} + \frac{1}{2}\sum_{i=1}^{r}\ln I_i \tag{4.8}$$

$$\frac{\partial}{\partial T} \ln q_{rot} = \frac{r}{2}\frac{1}{T} \tag{4.9}$$

Vibrational (classical limit):

$$q_{vib,cl} = \prod_{i=1}^{f} \frac{k_B T}{h\nu_i} \tag{4.10}$$

$$\ln q_{vib,cl} = \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} \tag{4.11}$$

$$\frac{\partial}{\partial T} \ln q_{vib,cl} = f\frac{1}{T} \tag{4.12}$$

Vibrational (quantum):

$$q_{vib,qm} = \prod_{i=1}^{f} \frac{\exp \frac{h\nu}{2k_BT}}{1 - \exp \frac{h\nu}{k_BT}} = \prod_{i=1}^{f} \left[2\sinh\left(\frac{h\nu}{2k_BT}\right)\right]^{-1} \tag{4.13}$$

$$\ln q_{vib,qm} = -\sum_{i=1}^{f} \ln \left[2\sinh\left(\frac{h\nu}{2k_BT}\right)\right] \tag{4.14}$$

$$\frac{\partial}{\partial T} \ln q_{vib,qm} = \frac{1}{T} \sum_{i=1}^{f} \frac{\frac{h\nu}{2k_BT}}{\tanh\left(\frac{h\nu}{2k_BT}\right)} \tag{4.15}$$

Electronic:

$$q_{elec} = \exp\left(-\frac{E_m}{k_BT}\right) \tag{4.16}$$

$$\ln q_{elec} = -\frac{E_m}{k_BT} \tag{4.17}$$

$$\frac{\partial}{\partial T} \ln q_{elec} = \frac{E_m}{k_BT}\frac{1}{T} \tag{4.18}$$

Helmholtz Free energy

$$F = U - TS \tag{4.19}$$

$$F = -k_BT \ln Q = -Nk_BT \ln \frac{qe}{N} \tag{4.20}$$

$$= -Nk_BT \ln \frac{q_{tr}e}{N} - Nk_BT \ln q_{rot} - Nk_BT \ln q_{vib} - Nk_BT \ln q_{elec} \tag{4.21}$$

$$= F_{tr} + F_{rot} + F_{vib} + F_{elec} \tag{4.22}$$

$$F_{tr} = -Nk_BT \ln \frac{q_{tr}e}{N} = -Nk_BT \ln \left[\left(\frac{2\pi m k_BT}{h^2}\right)^{t/2} \frac{Ve}{N}\right] \tag{4.23}$$

$$= -Nk_BT \left(\frac{t}{2} \ln \frac{2\pi m k_BT}{h^2} + 1 - \ln \frac{N}{V}\right) \tag{4.24}$$

$$F_{rot} = -Nk_BT \ln q_{rot} = -Nk_BT \left[\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_BT}{h^2} + \frac{1}{2} \sum_{i=1}^{r} I_i \right] \tag{4.25}$$

$$F_{vib,cl} = -Nk_BT \ln q_{vib,cl} = -Nk_BT \sum_{i=1}^{f} \ln \frac{k_BT}{h\nu_i} \tag{4.26}$$

$$F_{vib,qm} = -Nk_BT \ln q_{vib,qm} = Nk_BT \sum_{i=1}^{f} \ln \left[2\sinh\left(\frac{h\nu}{2k_BT}\right)\right] \tag{4.27}$$

$$F_{elec} = -Nk_BT \ln q_{elec} = NE_m \tag{4.28}$$

Molar Helmholtz free energy

$$\mu = F_m = \frac{\partial F}{\partial N} \tag{4.29}$$

$$= \frac{\partial F_{tr}}{\partial N} + \frac{\partial F_{rot}}{\partial N} + \frac{\partial F_{vib}}{\partial N} + \frac{\partial F_{elec}}{\partial N} \tag{4.30}$$

$$= \mu_{tr} + \mu_{rot} + \mu_{vib} + \mu_{elec} \tag{4.31}$$

$$\mu_{tr} = \frac{\partial F_{tr}}{\partial N} = -k_B T \frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + k_B T \ln \frac{N}{V} = \frac{F_{tr}}{N} + k_B T \tag{4.32}$$

$$\mu_{rot} = \frac{\partial F_{rot}}{\partial N} = -k_B T \left[\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} I_i \right] = \frac{F_{rot}}{N} \tag{4.33}$$

$$\mu_{vib,cl} = \frac{\partial F_{vib,cl}}{\partial N} = -k_B T \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} = \frac{F_{vib,cl}}{N} \tag{4.34}$$

$$\mu_{vib,qm} = \frac{\partial F_{vib,qm}}{\partial N} = k_B T \sum_{i=1}^{f} \ln \left[2 \sinh \left(\frac{h\nu}{2 k_B T} \right) \right] = \frac{F_{vib,qm}}{N} \tag{4.35}$$

$$\mu_{elec} = \frac{\partial F_{elec}}{\partial N} = E_m = \frac{F_{elec}}{N} \tag{4.36}$$

Internal energy

$$U = k_B T^2 \frac{\partial}{\partial T} \ln Q = N k_B T^2 \frac{\partial}{\partial T} \ln q \tag{4.37}$$

$$= N k_B T^2 \frac{\partial}{\partial T} \ln q_{tr} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{rot} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib} + N k_B T^2 \frac{\partial}{\partial T} \ln q_{elec} \tag{4.38}$$

$$= U_{tr} + U_{rot} + U_{vib} + U_{elec} \tag{4.39}$$

$$U_{tr} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{tr} = N k_B T \frac{t}{2} \tag{4.40}$$

$$U_{rot} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{rot} = N k_B T \frac{r}{2} \tag{4.41}$$

$$U_{vib,cl} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib,cl} = N k_B T f \tag{4.42}$$

$$U_{vib,qm} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{vib,qm} = N k_B T \sum_{i=1}^{f} \frac{\frac{h\nu}{2 k_B T}}{\tanh \left(\frac{h\nu}{2 k_B T} \right)} \tag{4.43}$$

$$U_{elec} = N k_B T^2 \frac{\partial}{\partial T} \ln q_{elec} = N E_m \tag{4.44}$$

Molar internal energy

$$U_m = \frac{\partial U}{\partial N} = \frac{\partial U_{tr}}{\partial N} + \frac{\partial U_{rot}}{\partial N} + \frac{\partial U_{vib}}{\partial N} + \frac{\partial U_{elec}}{\partial N} = U_{m,tr} + U_{m,rot} + U_{m,vib} + U_{m,elec} \tag{4.45}$$

$$U_{m,tr} = \frac{\partial U_{tr}}{\partial N} = k_B T \frac{t}{2} = \frac{U_{tr}}{N} \tag{4.46}$$

$$U_{m,rot} = \frac{\partial U_{rot}}{\partial N} = k_B T \frac{r}{2} = \frac{U_{rot}}{N} \tag{4.47}$$

$$U_{m,vib,cl} = \frac{\partial U_{vib,cl}}{\partial N} = k_B T f = \frac{U_{vib,cl}}{N} \tag{4.48}$$

$$U_{m,vib,qm} = \frac{\partial U_{vib,qm}}{\partial N} = k_B T \sum_{i=1}^{f} \frac{\frac{h\nu}{2k_BT}}{\tanh\left(\frac{h\nu}{2k_BT}\right)} = \frac{U_{vib,qm}}{N} \tag{4.49}$$

$$U_{m,elec} = \frac{\partial U_{elec}}{\partial N} = E_m = \frac{U_{elec}}{N} \tag{4.50}$$

Entropy

$$S = \frac{\partial}{\partial T}\left(k_B T \ln Q\right) = k_B \ln Q + k_B T \frac{\partial}{\partial T} \ln Q = N k_B \ln \frac{qe}{N} + N k_B T \frac{\partial}{\partial T} \ln q \tag{4.51}$$

$$\begin{aligned} &= \left(N k_B \ln \frac{q_{tr} e}{N} + N k_B T \frac{\partial}{\partial T} \ln q_{tr}\right) + \left(N k_B \ln q_{rot} + N k_B T \frac{\partial}{\partial T} \ln q_{rot}\right) \\ &\qquad + \left(N k_B \ln q_{vib} + N k_B T \frac{\partial}{\partial T} \ln q_{vib}\right) + \left(N k_B \ln q_{elec} + N k_B T \frac{\partial}{\partial T} \ln q_{elec}\right) \end{aligned} \tag{4.52}$$

$$= S_{tr} + S_{rot} + S_{vib} + S_{elec} \tag{4.53}$$

$$S_{tr} = N k_B \ln \frac{q_{tr} e}{N} + N k_B T \frac{\partial}{\partial T} \ln q_{tr} = N k_B \left(\frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + \frac{t+2}{2} - \ln \frac{N}{V}\right) \tag{4.54}$$

$$S_{rot} = N k_B \ln q_{rot} + N k_B T \frac{\partial}{\partial T} \ln q_{rot} = N k_B \left(\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} I_i + \frac{r}{2}\right) \tag{4.55}$$

$$S_{vib,cl} = N k_B \ln q_{vib,cl} + N k_B T \frac{\partial}{\partial T} \ln q_{vib,cl} = N k_B \left(f + \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i}\right) \tag{4.56}$$

$$S_{vib,qm} = N k_B \ln q_{vib,qm} + N k_B T \frac{\partial}{\partial T} \ln q_{vib,qm} = N k_B \sum_{i=1}^{f} \left[\frac{\frac{h\nu}{2k_BT}}{\tanh \frac{h\nu}{2k_BT}} - \ln\left(2 \sinh \frac{h\nu}{2k_BT}\right)\right] \tag{4.57}$$

$$S_{elec} = N k_B \ln q_{elec} + N k_B T \frac{\partial}{\partial T} \ln q_{elec} = 0 \tag{4.58}$$

Molar Entropy

$$S_m = \frac{\partial S}{\partial N} = \frac{\partial S_{tr}}{\partial N} + \frac{\partial S_{rot}}{\partial N} + \frac{\partial S_{vib}}{\partial N} + \frac{\partial S_{elec}}{\partial N} = S_{m,tr} + S_{m,rot} + S_{m,vib} + S_{m,elec} \tag{4.59}$$

$$S_{m,tr} = \frac{\partial S_{tr}}{\partial N} = k_B \left(\frac{t}{2} \ln \frac{2\pi m k_B T}{h^2} + \frac{t}{2} - \ln \frac{N}{V} \right) = \frac{S_{tr}}{N} - k_B \tag{4.60}$$

$$S_{m,rot} = \frac{\partial S_{rot}}{\partial N} = k_B \left(\frac{1}{2} \ln \frac{\pi}{\sigma^2} + \frac{r}{2} \ln \frac{8\pi^2 k_B T}{h^2} + \frac{1}{2} \sum_{i=1}^{r} I_i + \frac{r}{2} \right) = \frac{S_{rot}}{N} \tag{4.61}$$

$$S_{m,vib,cl} = \frac{\partial S_{vib,cl}}{\partial N} = k_B \left(f + \sum_{i=1}^{f} \ln \frac{k_B T}{h\nu_i} \right) = \frac{S_{vib,cl}}{N} \tag{4.62}$$

$$S_{m,vib,qm} = \frac{\partial S_{vib,qm}}{\partial N} = k_B \sum_{i=1}^{f} \left[\frac{\frac{h\nu}{2k_B T}}{\tanh \frac{h\nu}{2k_B T}} - \ln \left(2 \sinh \frac{h\nu}{2k_B T} \right) \right] = \frac{S_{vib,qm}}{N} \tag{4.63}$$

$$S_{m,elec} = \frac{\partial S_{elec}}{\partial N} = 0 \tag{4.64}$$

Chapter 5

Examples

Inside the `examples` directory it is possible to find some ready to use input files for some published applications. A reference output is present.

Dimerization of Insulin

This first example in based on the reference paper by B. Tidor and M. Karplus [4], 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 main text.

The input files are inside the `insulin` subdirectory. They can be run with:

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

The reference output is `insulin.out.ref`

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 (`diala` directory) in its c7 equatorial (c7eq) and c7 axial (c7ax) conformations;
- beta hairpin from protein G (`bhp`) in the native beta-harpin (bhp1) and in a three-stranded beta sheet (bhp2) conformations;
- the converter of the biomolecular motor myosin VI (`conv`) in the pre-powerstroke (pps) and rigor-like (rig) conformations.

These examples, see Fig. 5.1, are taken form the work of M. Cecchini, JCTC 2015 [2] , where they were used to develop a quantum correction to the classical conformational free energy difference.

> M. Cecchini, "Quantum Corrections to the Free Energy Difference between Peptides and Proteins Conformers", Journal of Chemical Theory and Computation, 2015 ASAP

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

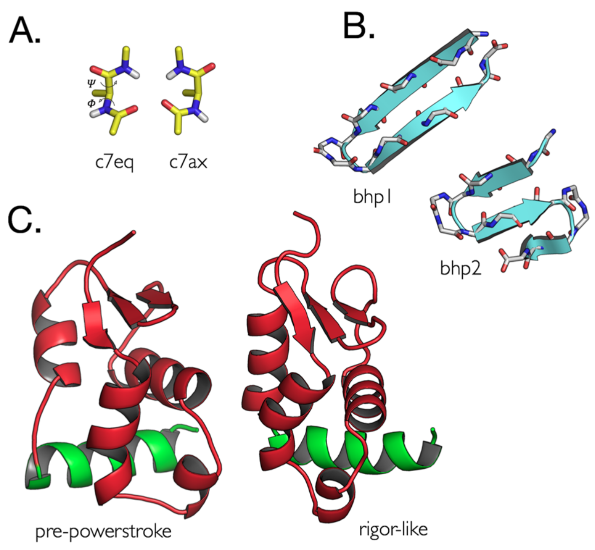

Figure 5.1: Representation of the three examples to study the quantum correction to the conformational free energy difference: the alanine dipeptide, the beta hairpin of protein G and the converter of myosin VI. Reproduced from [2].

```
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.

Chapter 6

Module Documentation

6.1 Thermo Module

Calculate themodynamic quantities based on the canonical partition function.

Functions

- void thermo_calcthermo (Thermo ∗A)

  Calculate all thermodynamic quantities based on partition function.
- void thermo_cumulvib (const Thermo ∗A, const char ∗fname)

  Print the vibrational free energy in a cumulative way as a function of the vibrational frequencies.
- void thermo_delete (Thermo ∗A)

  Delete all allocated memory inside a Thermo structure.
- void thermo_diffthermo (const Thermo ∗A, const Thermo ∗B, int nA, int nB, Thermo ∗D)

  Calculate the thermodynamic quantities for a chemical reaction.
- void thermo_init (Thermo ∗A)

  Initialize to defualt values a Thermo structure.
- void thermo_printconfig (const Thermo ∗A)

  Print to stdout the parsed quantities from a Thermo input file.
- void thermo_printthermo (const Thermo ∗A, int onlyInt)

  Print to stdout the evaluated internal energy, entropy and free energy.
- int thermo_readthermo (Thermo ∗A, const char ∗fname)

  Read a thermo file and save all quantities inside a Thermo structure.
- void thermo_vdos (Thermo ∗A, const char ∗fname)

  Evaluate the vibrational density of states (VDOS) starting from the vibrational frequencies.

6.1.1 Detailed Description

Calculate themodynamic quantities based on the canonical partition function.

6.1.2 Function Documentation

6.1.2.1 void thermo_calcthermo (Thermo ∗ *A*)

Calculate all thermodynamic quantities based on partition function.

Taken an initialized `Thermo` structure, evaluates the translational, rotational, vibrational and electronic contributions to the partition function. From that, the internal energy, entropy and free energy are evaluated.

Parameters

`in,out`	*A*	Pointer to an initialized `Thermo` structure

6.1.2.2 void thermo_cumulvib (const Thermo $*$ *A*, const char $*$ *fname*)

Print the vibrational free energy in a cumulative way as a function of the vibrational frequencies.

This function generates two file called `fname.k.dat` and `fname.f.dat` . Both contain the cumulative vibrational free energy: the first as a function of the number of modes, the second as a function of the frequency.

Parameters

`in`	*A*	Pointer to an initialized `Thermo` structure
`in`	*fname*	Base filename to save the cumulative vibrational free energy

6.1.2.3 void thermo_delete (Thermo $*$ *A*)

Delete all allocated memory inside a Thermo structure.

Parameters

`in,out`	*A*	Pointer to an initialized `Thermo` structure

6.1.2.4 void thermo_diffthermo (const Thermo $*$ *A*, const Thermo $*$ *B*, int *nA*, int *nB*, Thermo $*$ *D*)

Calculate the thermodynamic quantities for a chemical reaction.

Given two systems and the stechiometric coefficents calculates the thermodynamic quantities for the reaction $aA \rightleftharpoons bB$. Evaluates $D = nB \cdot B - nA \cdot A$. For all energy, entropy and free energy.

Parameters

`in`	*A*	Pointer to an initialized `Thermo` structure
`in`	*B*	Pointer to an initialized `Thermo` structure
`in`	*nA*	Stechiometric coefficient for structure `A`
`in`	*nB*	Stechiometric coefficient for structure `B`
`out`	*D*	Pointer to an initialized `Thermo` structure

6.1.2.5 void thermo_init (Thermo $*$ *A*)

Initialize to defualt values a Thermo structure.

In particular: temperature sets to 300K, number of moles to 1, volume to 1 liter, and accuracy in vibrational to 1cm^{-1}.

Parameters

`in,out`	*A*	Pointer to an initialized `Thermo` structure

6.1.2.6 void thermo_printconfig (const Thermo $*$ *A*)

Print to stdout the parsed quantities from a Thermo input file.

Parameters

`in`	*A*	Pointer to an initialized `Thermo` structure

6.1.2.7 void thermo_printthermo (const Thermo ∗ *A*, int *onlyInt*)

Print to stdout the evaluated internal energy, entropy and free energy.

Parameters

`in`	*A*	Pointer to an initialized `Thermo` structure
`in`	*onlyInt*	If set to one, only the intensive quantities are printed

6.1.2.8 int thermo_readthermo (Thermo ∗ *A*, const char ∗ *fname*)

Read a thermo file and save all quantities inside a Thermo structure.

Parameters

`in,out`	*A*	Pointer to an initialized `Thermo` structure
`in`	*fname*	Input thermo filename

Return values

EXIT_SUCCESS	if everithing is ok, EXIT_FAILURE otherwise

6.1.2.9 void thermo_vdos (Thermo ∗ *A*, const char ∗ *fname*)

Evaluate the vibrational density of states (VDOS) starting from the vibrational frequencies.

The free energy is vibrational free energy is also evaluated as integral over the VDOS (see Theory). In the output file, the VDOS, the free energy at each point, and the cumulative free energy is printes as a function of the frequency. To change the resolutin of the calculated VDOS, use the -n, –dnu command line option.

Parameters

`in`	*A*	Pointer to an initialized `Thermo` structure
`in`	*fname*	Filename where to print the VDOS and the free energy.

Bibliography

[1] Sébastien Haar, Artur Ciesielski, Joseph Clough, Huafeng Yang, Raffaello Mazzaro, Fanny Richard, Simone Conti, Nicolas Merstorf, Marco Cecchini, Vittorio Morandi, Cinzia Casiraghi, and Paolo Samorì. A supramolecular strategy to leverage the liquid-phase exfoliation of graphene in presence of surfactants: unraveling the role of the length of fatty acids. *Small*, 11(14):1736–1736, 2015. 2

[2] Marco Cecchini. Quantum corrections to the free energy difference between peptides and proteins conformers. *Journal of Chemical Theory and Computation*, ASAP, 2015. 2, 9, 10

[3] Donald A. McQuarrie. *Statistical Mechanics*. University Science Books, 2000. 4

[4] Bruce Tidor and Martin Karplus. The contribution of vibrational entropy to molecular association: the dimerization of insulin. *Journal of molecular biology*, 238(3):405–414, 1994. 9