

---

# **ThermoDiff Documentation**

***Release 0.0.1***

**Milosz Wieczor**

**Oct 09, 2020**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	Theory . . . . .	3
1.2	Mods input file . . . . .	6
1.3	Targets input file . . . . .	7
1.4	Trajectories input file . . . . .	9
1.5	Force field refinement . . . . .	10
1.6	Coarse-grained model optimization . . . . .	10
1.7	Drug design . . . . .	10



ThermoDiff is a versatile Python tool that allows for efficient combination of data from multiple Molecular Dynamics trajectories in order to

- improve existing force fields,
- reparametrize coarse grained (CG) models for consistency with all-atom models,
- identify regions in molecules (e.g. ligands) that contribute to binding affinity or selectivity.



ThermoDiff calculates (a) free energy derivatives or (b) observable derivatives with respect to individual force field parameters, such as atomic charges  $q$ , Lennard-Jones  $\sigma$  or  $\varepsilon$  (also with custom NBFIX combination rules), as well as bonded terms: reference bond lengths, planar and dihedral angles, or force constants.

The general idea is to let the user provide as much data as possible - ideally multiple trajectories for a number of molecular setups - then select a number of force field parameters to investigate, and a subset of quantities of interest that are e.g. experimentally known constraints. The analysis should yield a ‘sensitivity matrix’, an estimate of sensitivity of individual quantities to individual force field parameters.

As a result, the user can make informed decisions about which parameters should be changed in order to efficiently improve the poorly performing properties of the model, while retaining the properties that currently agree with existing data. In principle, the three areas mentioned above - force field development, coarse-graining of atomistic models and lead optimization in drug design - could benefit the most from this approach.

## 1.1 Theory

ThermoDiff is based on simple numerical differentiation and reweighting schemes drawing from statistical thermodynamics, including free energy perturbation (FEP).

The main idea is that one can measure the “local sensitivity” of free energies to any individual force field parameter (here generically called  $\sigma$ ) by simply using the difference quotient:

$$\frac{\partial F}{\partial \sigma} = \lim_{\Delta \sigma \rightarrow 0} \frac{F(\sigma + \Delta \sigma) - F(\sigma)}{\Delta \sigma}$$

also expressed in terms of the classical thermodynamical integration formula:

$$\frac{\partial F}{\partial \sigma} = \left\langle \frac{\partial H}{\partial \sigma} \right\rangle_{\sigma=\sigma_0} = \lim_{\Delta \sigma \rightarrow 0} \frac{\langle H(\mathbf{x}; \sigma + \Delta \sigma) - H(\mathbf{x}; \sigma) \rangle}{\Delta \sigma}$$

Here,  $\frac{\partial F}{\partial \sigma}$  can be calculated using alchemical approaches that yield  $\langle \frac{\partial H}{\partial \sigma} \rangle_{\sigma}$  directly, or using the well-known Zwanzig equation:

$$\Delta F(A \rightarrow B) = -\beta^{-1} \log \left\langle e^{-\beta(H_B - H_A)} \right\rangle_A$$

where  $H_B = H(\mathbf{x}; \sigma + \Delta\sigma)$  and  $H_A = H(\mathbf{x}; \sigma)$ . Since the two ensembles only differ by an arbitrarily small change in a single force field parameter,  $\Delta\sigma$ , we avoid the common problem of phase space overlap that restricts the direct application of the Zwanzig equation to all but simplest model systems.

### 1.1.1 Practical implementation

#### Analytical derivatives

With the free energy module of Gromacs, it is possible to set up an alchemical topology where state A ( $\lambda = 0$ ) is the original ensemble, and state B ( $\lambda = 1$ ) corresponds to  $\sigma$  incremented by 1 unit. In this case, values of  $\frac{\partial H}{\partial \lambda}$  reported in a rerun correspond exactly to  $\frac{\partial H}{\partial \sigma}$ , allowing for straightforward calculation of  $\frac{\partial F(\xi)}{\partial \sigma}$  as  $\langle \frac{\partial H}{\partial \sigma} \rangle_{\sigma, \xi}$ .

To calculate the respective derivative for structure-based observables, one can apply the quotient rule to the definition of ensemble average:

$$\langle Q \rangle = \frac{\int Q(\mathbf{x}) e^{-\beta U(\mathbf{x}, \sigma)} d\mathbf{x}}{\int e^{-\beta U(\mathbf{x}, \sigma)} d\mathbf{x}} = \frac{P}{Z}$$

$$\frac{\partial}{\partial \sigma} \langle Q \rangle = \frac{P'Z - PZ'}{Z^2} = \frac{P'}{Z} - \frac{P}{Z} \frac{Z'}{Z} = \frac{P'}{Z} + \langle Q \rangle \beta F'$$

knowing that  $\frac{Z'}{Z} = \log(Z)' = -\beta F'$ . The  $P'$  term turns out to be

$$\frac{\partial}{\partial \sigma} \int Q(\mathbf{x}) e^{-\beta U(\mathbf{x}, \sigma)} d\mathbf{x} = -\beta \int Q(\mathbf{x}) \frac{\partial U}{\partial \sigma} e^{-\beta U(\mathbf{x}, \sigma)} d\mathbf{x}$$

and hence the  $\frac{P'}{Z}$  term can be written as  $-\beta \langle Q \frac{\partial U}{\partial \sigma} \rangle$ . Putting it all together, the term in question is

$$\frac{\partial}{\partial \sigma} \langle Q \rangle = \beta \left\langle Q \left( \frac{\partial F}{\partial \sigma} - \frac{\partial U}{\partial \sigma} \right) \right\rangle = \beta \left( \langle Q \rangle \left\langle \frac{\partial U}{\partial \sigma} \right\rangle - \left\langle Q \frac{\partial U}{\partial \sigma} \right\rangle \right),$$

showing it is, in fact, the (negative) covariance between the observable itself and the internal energy's sensitivity to  $\sigma$ ,  $\frac{\partial U}{\partial \sigma}$ .

One can trivially extend this formula to treat the ensemble average as a function of the reaction coordinate  $\xi$ , and use the corresponding derivative of the free energy profile,  $\frac{\partial F(\xi)}{\partial \sigma}$ .

#### Discrete states

In practice, we are often interested in the free energy difference between two arbitrarily defined states, A (e.g. bound) and B (e.g. unbound). If we have access to the free energy profile, and states A and B are defined with boundaries  $A_0, A_1$  and  $B_0, B_1$ , we can rewrite the standard formula,  $\Delta F_{AB} = -\beta^{-1} \log \left( \frac{p_A}{p_B} \right)$ , as:

$$\Delta F_{AB} = -\beta^{-1} \left( -\log \int_{A_0}^{A_1} e^{-\beta F(\xi, \sigma)} d\sigma + \log \int_{B_0}^{B_1} e^{-\beta F(\xi, \sigma)} d\sigma \right)$$

Differentiating with respect to  $\sigma$  then yields:

$$\frac{\partial}{\partial \sigma} \Delta F_{AB} = -\beta^{-1} \left( \frac{\beta}{p_A} \int_{A_0}^{A_1} \frac{\partial F}{\partial \sigma} e^{-\beta F(\xi, \sigma)} d\sigma - \frac{\beta}{p_B} \int_{B_0}^{B_1} \frac{\partial F}{\partial \sigma} e^{-\beta F(\xi, \sigma)} d\sigma \right) = \left\langle \frac{\partial F}{\partial \sigma} \right\rangle_A - \left\langle \frac{\partial F}{\partial \sigma} \right\rangle_B$$

so that the derivative of the free energy difference between states A and B is the difference between the respective averages of the derivative over ensembles corresponding to states A and B.



## Numerical approximation

Alternatively, one can start with the original ensemble generated by the unaltered Hamiltonian  $H_A$  and assigns a weight  $\frac{1}{Z'} e^{-\beta(H_B-H_A)}$  to each frame  $i$ , corresponding to the probability with which the given frame would show up in an ensemble generated by the slightly perturbed Hamiltonian  $H_B$ . This is evident by noting that reweighting original data with such weights yields the desired free energy values:

$$\left\langle e^{-\beta(H_B-H_A)} \right\rangle_A = \frac{\int e^{-\beta(H_B-H_A)} e^{-\beta H_A} d\mathbf{x}}{\int e^{-\beta H_A} d\mathbf{x}} = \frac{\int e^{-\beta H_B} d\mathbf{x}}{\int e^{-\beta H_A} d\mathbf{x}} = e^{-\beta \Delta F}$$

The normalization constant  $Z'$  can be straightforwardly calculated from the data as  $\sum_i e^{-\beta(H_B-H_A)}$ , or just by making sure the reweighting factors add up to 1.

Once weights are calculated, not only the derivatives of free energy  $\frac{\partial F}{\partial \sigma}$ , but also of any observed quantity  $\frac{\partial Q}{\partial \sigma}$  can be computed in a simple manner, provided that all the relevant geometries are sampled properly in the input trajectories:

$$\begin{aligned} \frac{\partial F}{\partial \sigma} &= \lim_{\Delta \sigma \rightarrow 0} \frac{-\log \langle e^{-\beta(H_B-H_A)} \rangle}{\beta \Delta \sigma} \\ Q[\Delta \sigma] &= \frac{\int Q(\mathbf{x}) e^{-\beta(H_B-H_A)} w_0(\mathbf{x}) d\mathbf{x}}{\int w_0(\mathbf{x}) d\mathbf{x}} \\ \frac{\partial Q}{\partial \sigma} &= \lim_{\Delta \sigma \rightarrow 0} \frac{Q[\Delta \sigma] - Q[0]}{\Delta \sigma} \end{aligned}$$

In brief, the procedure can be thought of as a search for a subset of frames in which the small perturbation in  $\sigma$  would significantly affect the energetics of the system. Such frames will be assigned large or small weights in the reweighting procedure, and if they cluster significantly in a specific region of the reaction coordinate, this region will be respectively favored or disfavored by a finite change in  $\sigma$ .

Assuming a locally linear behavior of  $F(\sigma)$  and  $Q(\sigma)$  (which can be a reasonable assumption sufficiently far from bifurcation points), one can also try to predict the response of the equilibrium ensemble to a finite change in  $\sigma$  as  $\Delta Q = \frac{\partial Q}{\partial \sigma} \Delta \sigma$ .

### 1.1.2 Application to multiple targets and multiple trajectories

Since this is just a post-processing procedure, it is in principle very cheap to calculate  $\frac{\partial Q_j}{\partial \sigma_i}$  for a large set of attempted modifications  $\sigma_i$ , using a large set of target observables  $q_i$  for which we have reliable reference values  $T_j$ . We can also straightforwardly include data from multiple trajectories, also using free energy methods, provided that Boltzmann weights are available. (Note that free energies can also be used as such observables.) Subsequently, different approaches can be used to identify a small set of parameters  $\sigma_{opt}$  that should simultaneously improve the description of multiple target observables, or that (in a more conservative approach) can be selected for further optimization.

## Sensitivity matrix

One straightforward (but not necessarily optimal) way to do this is to (again) assume linear behavior of  $Q(\sigma)$ , and define the sensitivity  $\mu_{ij}$  as

$$\mu_{ij} = \frac{\frac{\partial Q_j}{\partial \sigma_i}}{T_j - \bar{Q}_j}$$

where the denominator is the difference between the current (prior) and desired (experimental) estimated values of  $Q$ , and the numerator estimates the first-order change in  $\bar{Q}_j$  in response to a unit change in  $\sigma_i$ . (Loosely speaking, we divide “how much  $Q$  we get from a change in sigma” by “how much  $Q$  we want” to get “how much we need to change sigma”.)

## Least-squares approximation

Now,  $\Delta\sigma_i^{opt} = \mu_{ij}^{-1}$  estimates the change in  $\sigma_i$  required to match the target  $T_j$ , and the simultaneous optimization of many target quantities can be thought of solving the equation

$$\mu\Delta\sigma = 1$$

for a set of optimal values,  $\Delta\sigma_i^{opt}$ . Assuming that columns  $\mu_j$  are linearly independent and the number of parameters to optimize is less than the number of experimental targets, this is equivalent to solving the least-squares fit problem. In practice, it would be desirable to first compute the sensitivity matrix for a large number of tunable parameters and subsequently only calculate least-squares for a small subset of them (3-4) that shall produce best agreement with experimental targets, in order to reduce overfitting and minimize the effect of (so far neglected) couplings between individual parameters.

## 1.2 Mods input file

The mods input specifies which atomistic properties should be modified during a ThermoDiff run. Each modification requires a separate set of reruns, and can encompass any set of atoms (see below), from a single atom in a single residue to a specific group of atoms in all given residues to all atoms bearing a specified type.

ThermoDiff then calculates how the particular quantity (free energies, ensemble averages, profiles etc.) respond to a change in the given atomistic property.

Note that many lines can be contained in a single input file, each input letter ([cseadnm]) corresponding to a new set of reruns:

- c corresponds to a change in charge
- s corresponds to a change in  $\sigma$
- e corresponds to a change in  $\epsilon$
- a corresponds to a change in angle force constant
- d corresponds to a change in dihedral force constant
- n corresponds to a change in pairwise  $\sigma$  (NBFIX)
- m corresponds to a change in pairwise  $\epsilon$  (NBFIX)

### 1.2.1 Sample input lines

#### Modify charge/ $\sigma$ / $\epsilon$

- chosen atoms of a single type, all within a single residue (note: need to be of the same type):

```
cse AMFB-1 O11 O12 O13 O6 O7 O9 O10 O1 O2 O16
```

- chosen atoms of a single type, possibly different residues (note: need to be of the same type):

```
cse AMFB-1-O11 AMFB-1-O12 AMFB-1-O13 AMFB-1-O6 AMFB-1-O7 AMFB-1-O9
```

- all atoms of a specified type:

```
cse OG311
```

## Modify angle/dihedral parameters

- chosen sets of atoms, all within a single residue (possible multiples of 3 (angles) or 4 (dihedrals) in a single line):

```
d AMFB-1 C1 C2 C3 C4
a AMFB-1 C1 C2 C3 C1 C2 C4
```

- chosen sets of atoms, possibly different residues (note: need to have the same type signature):

```
d AMFB-1-C1 AMFB-1-C2 AMFB-1-C3 AMFB-1-C4
a AMFB-1-C1 AMFB-1-C2 AMFB-1-C3 AMFB-1-C1 AMFB-1-C2 AMFB-1-C4
```

- chosen set of types:

```
a CG311 CG311 OG311
d OG311 CG311 CG311 OG311
```

## Modify nbfix $\sigma/\epsilon$

- chosen sets of atoms, all within a single residue (“;”-delimited; here O11, O12 and O13 need to have consistent types, as well as H1, H2 and H3)

```
nm AMFB-1 O11 O12 O13; H1 H2 H3
```

- chosen sets of atoms, possibly different residues (“;”-delimited; need to have consistent types, as above)

```
nm AMFB-1-O11 AMFB-1-O12 AMFB-1-O13; AMFB-1-H1 AMFB-1-H2 AMFB-1-H3
```

- chosen pair of types:

```
nm HGA2 OG311
```

Note that ‘n’ refers to NBFIX- $\sigma$  and ‘m’ refers to NBFIX- $\epsilon$ ; other cases should be intuitive (‘c’ for charge, ‘s’ for sigma etc.)

## 1.3 Targets input file

In its force-field related applications, ThermoDiff attempts to find modifications that will alter the dynamics of the system so that target quantities (e.g. experimentally measured radii of gyration, binding affinities, force-extension curves, all-atom based free energy profiles) are achieved.

Again, many lines can be contained in a single input file, starting with three letters [fo][ew][pd] followed by a user-specified keyword-value pairs (see below for examples).

### 1.3.1 Run/target settings

- concerning the type of target (mutually exclusive):
  - f sets free energy as a target
  - o sets any other observable (e.g. dihedral, distance, radius of gyration etc.) as a target
- concerning how the target quantity is set (mutually exclusive):

- `p` indicates that the reference quantity is a profile (`ref=...` should contain path to a similarly formatted file with the reference profile)
- `d` indicates that the reference quantity is a discrete set of state-averaged quantities (`ref=...` should contain a comma-separated list of values)

### 1.3.2 Keyworded arguments

- `label=...` sets an alias that can be used to reference the quantity later on
- `ref_profile=...` sets a reference profile (path to a 2-column file with the reference profile)
- `ref_discrete=...` sets a target for discrete states analysis (comma-separated floating-point values, one value for each state)
- `data=...` refers to a file containing paths to files with user-defined observables (other than the reaction coordinate), one file per trajectory file (in the respective order)
- `meta=...` points to a WHAM metafile (see below for formatting options) with reaction coordinate data
- `threshold=...` should contain floating-point values defining boundaries between states;  $2 \times n_{\text{states}}$  are needed (beginning and end of each state) so that non-adjacent states can be defined
- `trajset=...` should pass a comma-separated list of trajectory groups (see *Trajectories input file* for reference)
- `relative=...` should specify True if only differences between values are relevant (i.e. the quantity is defined up to an additive constant); True is default for free energy calculations, and False is default for other observables

### 1.3.3 Sample input lines

#### Free energy as a reference

- a free energy difference (-5 kcal/mol) between a bound and unbound state, calculated from US with bound state defined at up to 2 nm:

```
fd label=binding-affinity meta=/path/to/file/metafile.dat ref=-5.0,0.0 threshold=1.0,  
↪2.0,2.0,2.5
```

- a free energy profile for the extension of a molecule, calculated from an equilibrium distribution

```
fp label=ext-curve meta=/path/to/file/metafile.dat ref_profile=/path/to/file/target_  
↪profile.dat
```

#### user-defined quantity as a reference

- a difference in dipole moment (2.5 D) between a bound and unbound state, calculated from an equilibrium distribution (bound state defined at up to 1.0 nm and unbound from 1.5 to 2.5 nm):

```
od label=dipole-moment meta=/path/to/file/metafile_distance.dat data=/path/to/file/  
↪dipole_moment_datafiles.dat ref=2.5,0.0 threshold=0.5,1.0,1.5,2.5 relative=True
```

- a profile of a pseudo-dihedral in DNA along an A-to-B conformational transition, calculated from two sets of US runs (constituting only a specific subset of all trajectories):

```
op label=dihedral-profile meta=/path/to/file/metafile_conformational.dat data= ref_
↪profile=/path/to/file/target_profile.dat trajset=us1,us2
```

### 1.3.4 metafile format

The metafile format for WHAM is designed to be similar to the popular Alan Grossfield's implementation, i.e. each line should correspond to a single Umbrella Sampling window, and:

1. 1st column should contain the path to the datafile holding the original reaction coordinate (two-column, first column (times) is actually omitted, second column contains the reaction coordinate)
2.
  - a) if no external potential was used (equilibrium runs only), no additional columns should be present
  - b) if harmonic potentials were used, 2nd col should contain the center of potential in the given window
  - c) if any other free energy method was used, 2nd col should contain the paths to files with calculated weights
3. if harmonic potentials were used, 3rd col should contain the force constant (in kcal/mol, corresponding to  $k$  in the formula  $V = \frac{1}{2}k(x - x_0)^2$ ), and 4th col should contain F values (obtained from running WHAM on original data).

## 1.4 Trajectories input file

It can be convenient to restrict the parametrization of certain quantities to a subset of data, e.g. if enhanced sampling techniques such as Umbrella Sampling were performed on different coordinates, resulting in several sets of trajectories sampling different regions of configurational space. For this reason, trajectories can be clustered into user-defined groups and passed along with target quantities so that only the selected subset(s) will be used for target optimization.

If the `trajset=...` argument is not passed in the targets input (see [Targets input file](#)), by default all trajectories are used to optimize the given target.

### 1.4.1 Input format

Each group should be named first (with a colon at the beginning of a line), then a set of paths to trajectory files belonging to that group should follow.

Trajectories can be followed by temperature (default: 300) and a 0-based index of topology, if multiple topologies are passed to the main script (if different trajectories correspond to different systems).

### 1.4.2 Sample input

Multiple groups:

```
:group1
/home/user/simulations/mysystem/traj1_us.xtc
/home/user/simulations/mysystem/traj2_us.xtc
/home/user/simulations/mysystem/traj3_us.xtc
:group2
/home/user/simulations/mysystem/traj1_eq.xtc
/home/user/simulations/mysystem/traj2_eq.xtc
/home/user/simulations/mysystem/traj3_eq.xtc
```

Multiple topologies:

```
/home/user/simulations/mysystem1/traj1_eq.xtc 300 0
/home/user/simulations/mysystem1/traj2_eq.xtc 300 0
/home/user/simulations/mysystem2/traj1_eq.xtc 300 1
/home/user/simulations/mysystem2/traj2_eq.xtc 300 1
```

when running the script with:

```
-p "/path/to/mysystem1/topol.top /path/to/mysystem2/topol.top "
```

## 1.5 Force field refinement

One common problem in force field refinement is the optimal choice of parameters to improve, given the vast amount of them required to accurately represent the behavior of complex macromolecules. While highly accurate quantum chemical data can be used as reference, the inclusion of ensemble-averaged solvation and competition effects is usually beyond the capabilities of quantum chemistry.

Hence, such bottom-up approaches might benefit from being complemented by a top-down one. Indeed, ThermoDiff attempts to manipulate force field parameters so as to reproduce known properties of the systems under study, given the states in question are properly sampled in simulations, and per-frame statistical weights are available.

The kind of questions ThermoDiff was designed to answer is: “Which parameters should I touch if I want to simultaneously make process X more favorable, increase average Y and retain the fraction of Z from the current force field?”. After that, the parameters need to be changed and tested empirically anyway, but there is a good chance one won’t be lost navigating the vast sea of overfitting and tradeoffs.

## 1.6 Coarse-grained model optimization

With ThermoDiff, CG models can be improved to match either experimental data (in a similar way to AA force field reparametrization) or AA-derived quantities, such as ensemble averages or profiles of any observables, as well as free energies.

A typical procedure aimed at improving the parameters of a particular CG model would be to perform a set of all-atom enhanced sampling simulations (e.g. umbrella sampling, or replica-exchange metadynamics coupled to an equilibrium ensemble) and then ask ThermoDiff to reproduce these quantities on the CG level; it is particularly reasonable to try to optimize the NBFIX (i.e. custom combination rules) components, as these allow to fine-tune the interactions between specific beads without affecting global properties of the molecular model. If one however aims to optimize global properties, the optimization of  $\sigma$ ,  $\epsilon$  and atomic charges can also be accomplished with ThermoDiff.

## 1.7 Drug design

ThermoDiff can be applied to lead optimization in a straightforward way, providing the location of “hot spots” - sites at which substituents should have a strong impact on the desired thermodynamic properties.

### 1.7.1 Affinity modulation

To search for sites that increase affinity, one needs to provide trajectories that sample two states: bound and unbound. Selected “query” sites have to be selected manually as independent lines in the *Mods input file*. To define states A and B, datafiles can be provided for each trajectory in which frames corresponding to the bound state have value 0,

and ones sampled from the unbound state have value 1. A list of these two-column data files, corresponding to the *Trajectories input file*, has to be stored in a separate file.

Now, the following line in the *Targets input file* requests the desired calculation (assume equilibrium simulations of both states):

```
fd label=binding-affinity meta=/path/to/file/meta.dat ref=B threshold=-0.5,0.5,0.5,1.5
```

The above line indicates that:

- we use free energy as target quantity ( $f$ ) using the difference between two states as target ( $d$ )
- the quantity will be labeled as `binding-affinity`
- `/path/to/file/meta.dat` will be used to locate datafiles with reaction coordinate data
- we seek to increase the free energy of state B (unbound; or, equivalently, lower the free energy of state A, bound; we follow the natural convention that the lower the free energy, the more favorable the respective state)
- state A is defined from -0.5 to 0.5, and state B from 0.5 to 1.5 (artificial bounds chosen so that 0 falls into A and 1 into B)

### 1.7.2 Selectivity modulation

To increase selectivity, an identical procedure can be used as above, as instead of increasing the free energy of the unbound state we wish to increase the free energy of the deprecated bound state with respect to the preferred bound state.

Note that we can attempt to simultaneously increase *both* affinity and selectivity simply by defining two target quantities and labeling them as follows:

- 0 for the unbound state,
- 1 for the preferred bound state,
- 2 for the deprecated bound state

so that the input looks like this:

```
fd label=binding-affinity data=/path/to/file/meta.dat ref=A threshold=-0.5,0.5,0.5,1.5
fd label=selective-affinity data=/path/to/file/meta.dat ref=B threshold=0.5,1.5,1.5,2.
↪5
```

Now we first attempt to increase the free energy of state A (unbound) with respect to B (preferred bound), and then to increase the free energy of state B (deprecated bound) with respect to A (preferred bound).