30 August 2026

# Atomic Decomposition and Mapping of Solute Thermodynamics

Joe Cruz, Samrat Lohar, Helmut Carter, Steven Ramsey[1], Daniel R. Roe[2], Michael K. Gilson[3], Tom Kurtzman[1]

1. Department of Chemistry, Lehman College4, City University of New York

2. National Heart, Lung, and Blood Institute, National Institutes of Health

3. Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego

## Abstract

Here we present theory, methods, and software that decomposes the free energy of a solvated solute into its energetic and entropic contributions and maps these quantities to solute atoms. Because the mapping is atomic, contributions can be summed over any set of atoms of interest, giving thermodynamic values for ligand functional groups, protein backbone and side chains, individual residues, or an entire binding site. Software implementing the energy decomposition is integrated into cpptraj in AmberTools and software introduced and described here. We illustrate the approach by atomistically mapping the energetic and entropic contributions of ligand 735 and of the binding-site residues of PPAR-α.

# Atomic Decomposition and Mapping of Solute Thermodynamics

Joe Cruz[1], Samrat Lohar[1], Helmut Carter[2], Steven Ramsey[4], Daniel R. Roe[5], Michael K. Gilson[*,6], Tom Kurtzman[*,1−4]

August 2026

1-3. Ph.D. Programs in Chemistry[1], Biology[2], Biochemistry[3], The Graduate Center of the City University of New York, New York, 10016, USA;

4. Department of Chemistry, Lehman College[4], The City University of New York, Bronx, New York, 10468, USA.

5. Laboratory of Computational Biology, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, Maryland, USA

6. Department of Pharmaceutical Sciences, Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, La Jolla, CA 92093, USA.

* To whom correspondence should be addressed. Tom Kurtzman: simpleliquid@gmail.com Michael Gilson: mgilson@ucsd.edu

**Abstract**

Here we present theory, methods, and software that decomposes the free energy of a solvated solute into its energetic and entropic contributions and maps these quantities to solute atoms. Because the mapping is atomic, contributions can be summed over any set of atoms of interest, giving thermodynamic values for ligand functional groups, protein backbone and side chains, individual residues, or an entire binding site. Software implementing the energy decomposition is integrated into `cpptraj` in AmberTools and software introduced and described here. We illustrate the approach by atomistically mapping the energetic and entropic contributions of ligand 735 and of the binding-site residues of PPAR-$\alpha$.

# 1 Introduction

Molecular recognition is routinely discussed in terms of a small set of thermodynamic intuitions. When a ligand binds a protein, it is generally held that the ligand pays an entropic price, because it is confined to the subset of conformations that fit the binding site and that place its polar groups so as to donate and accept hydrogen bonds and its apolar

groups so as to make favorable contacts. The binding site is expected to pay a similar price, as backbone and side chain fluctuations are damped by the presence of the ligand. Against these losses are set the energetic gains of the new protein-ligand interactions and the release of unfavorable water from the site. This picture is useful, and it underlies a great deal of medicinal chemistry practice, but it is a statement about whole molecules, and it has largely been assumed rather than measured.

The true situation is more differentiated. Entropy and energy are not distributed uniformly over a molecule, and binding does not restrict every part of a ligand or a binding site to the same degree. A rigid core may be tightly held in place while a peripheral substituent remains free to rotate, and a side chain adjacent to the ligand may become more mobile on binding rather than less, if binding relieves a packing constraint or displaces an ordered water network. Terms of this kind are invisible to methods that return a single number. Alchemical free energy methods such as free energy perturbation and thermodynamic integration[1, 2] can predict binding affinities to useful accuracy, but they do not indicate which parts of a ligand or protein contribute to the predicted values, and so give little direct guidance about how to modify a lead compound.

Spatially resolved analyses of the solvent have shown how valuable the alternative can be. Inhomogeneous solvation theory[3, 4] and its numerical implementations, including WaterMap[5], GIST[6], and related tools, assign energetic and entropic quantities to positions in space, and have given designers a concrete way to reason about which waters to displace and which to preserve. No comparable treatment has been available for the solute itself, that is, for the ligand and the protein.

Here we present theory, methods, and software that decompose the thermodynamics of a solvated solute and map the energetic and entropic contributions onto its individual atoms. Because the mapping is atomic, contributions can be summed over any set of atoms of interest, giving thermodynamic values for ligand functional groups, for protein side chains and backbone, or for a whole residue or binding site. The approach applies to any single thermodynamic end state, such as a free ligand in solution, an apo protein, or a bound complex, so that maps computed for two end states can be subtracted to give an atomically resolved picture of what changes on binding. We anticipate that these maps will provide insight into molecular recognition much as solvation thermodynamic mapping methods have, by indicating which parts of a ligand and of a binding site favor binding, which oppose it, and which are essentially passive, and by offering a direct way to examine the received wisdom described above. A formal generalization of the present treatment, which places the solute and solvent within a single mapping framework, is developed in generalized thermodynamic mapping (GTM) theory[7].

# 2 Theory and Methods

Here, we present theory and methods to calculate the free energetic contributions of a solvated solute and to decompose the free energy so as to atomistically map the entropic and energetic contributions to the free energy of a solute in solution. Section 2.1 develops the multibody expansion of the potential energy, the assignment of energy to individual atoms, and the definitions of the solute system and solute internal energies. Section 2.2

describes the evaluation of these quantities for a standard molecular mechanics force field. Section 2.3 separates the solute and solvent contributions to the entropy exactly, and then develops the mutual information expansion and the atomic assignment used to estimate and map the solute configurational entropy. Sections 2.4 and 2.5 describe the bond angle torsion coordinate set and the nearest neighbor estimator used to evaluate the entropy terms, and Section 2.6 gives simulation details.

In separate work, we offer an extended formal treatment of the theory presented here generalized to include both solute and solvent thermodynamic mapping can be found in generalized thermodynamic mapping (GTM) theory[7], which places the atomic decomposition of energy and entropy on a common statistical mechanical footing and extends it to both solute and solvent, and to spatial as well as atomic mappings. GTM Theory is complementary to the work presented here: the present paper develops and applies a practical route to atomic solute free energy maps, while GTM theory[7] provides the more formal and general framework in which those maps sit.

## 2.1 Energy Decomposition and Atomic Mapping

The potential energy of a classical system in a given configuration can be written, without approximation, as a multibody expansion over the atoms:

$$U(\mathbf{z}^M) = \sum_{\alpha=1}^{M} u_\alpha(\mathbf{z}_\alpha) + \sum_{\alpha=1}^{M}\sum_{\beta>\alpha}^{M} u_{\alpha\beta}(\mathbf{z}_\alpha, \mathbf{z}_\beta) + \sum_{\alpha=1}^{M}\sum_{\beta>\alpha}^{M}\sum_{\gamma>\beta}^{M} u_{\alpha\beta\gamma}(\mathbf{z}_\alpha, \mathbf{z}_\beta, \mathbf{z}_\gamma) + O(4+) \quad (1)$$

where $\alpha$, $\beta$, and $\gamma$ index the $M$ atoms of the system, and $u_\alpha$, $u_{\alpha\beta}$, and $u_{\alpha\beta\gamma}$ are the one body, two body, and three body energy contributions associated with atom $\alpha$, atom pair $\alpha, \beta$, and atomic triplet $\alpha, \beta, \gamma$, respectively. Each term is assigned to the atoms that define it and shared equally among them, so that atom $\alpha$ receives its full one body energy, one half of each two body term it participates in, one third of each three body term. Thus, the mean energy of atom $\alpha$ is given by

$$U_\alpha = \langle u_\alpha(\mathbf{z}_\alpha)\rangle + \frac{1}{2}\sum_{\beta\neq\alpha}^{M} \langle u_{\alpha\beta}(\mathbf{z}_\alpha, \mathbf{z}_\beta)\rangle + \frac{1}{3}\sum_{\beta\neq\alpha}^{M}\sum_{\substack{\gamma\neq\alpha \\ \gamma>\beta}}^{M} \langle u_{\alpha\beta\gamma}(\mathbf{z}_\alpha, \mathbf{z}_\beta, \mathbf{z}_\gamma)\rangle + O(4+) \quad (2)$$

where the angle brackets denote a thermodynamic average, and the higher order contributions to each atom follow the same pattern. Note that the sums in Equation 1 run over all atoms of the solutes and the solvent, so that summing over all atomic quantities yields the full energy, $U$, of the system:

$$U = \sum_{\alpha=1}^{M} U_\alpha \quad (3)$$

Summing the mapped atomic energies over only the solute atoms gives the solute system

energy:

$$U_{\mathrm{sol}}^{\mathrm{sys}} = \sum_{\alpha\in\mathcal{S}} U_\alpha = \sum_{\alpha\in\mathcal{S}} \left[ \langle u_\alpha(z_\alpha)\rangle + \frac{1}{2}\sum_{\beta\neq\alpha}^{M} \langle u_{\alpha\beta}(z_\alpha, z_\beta)\rangle + \frac{1}{3}\sum_{\beta\neq\alpha}^{M}\sum_{\substack{\gamma\neq\alpha \\ \gamma>\beta}}^{M} \langle u_{\alpha\beta\gamma}(z_\alpha, z_\beta, z_\gamma)\rangle + O(4+) \right] \tag{4}$$

Here, $\mathcal{S}$ denotes the $N$ atoms of the solute. The inner sums run over all $M$ atoms, so this quantity carries the internal terms of the solute together with the solute's share of its interactions with the solvent, one half of each pair term and one third of each triplet term.

Similarly restricting the outer sums to the solute set as well gives the solute internal energy, which we define here as the entire set of bonded and non-bonded interactions that a solute has with itself. More generally, the atomic assignment is robust to how the sums are restricted. Because every energetic and entropic quantity is first assigned to individual atoms, the outer sums may be limited to any subset of atoms of interest, so that energies and entropies can be reported for ligand functional groups, protein backbone and side chains, individual residues, or an entire binding site, while the inner sums independently control which interaction partners are included.

$$U_{\mathrm{sol}}^{\mathrm{int}} = \sum_{\alpha\in\mathcal{S}} \left[ \langle u_\alpha(z_\alpha)\rangle + \frac{1}{2}\sum_{\substack{\beta\in\mathcal{S} \\ \beta\neq\alpha}} \langle u_{\alpha\beta}(z_\alpha, z_\beta)\rangle + \frac{1}{3}\sum_{\substack{\beta\in\mathcal{S} \\ \beta\neq\alpha}}\sum_{\substack{\gamma\in\mathcal{S} \\ \gamma\neq\alpha,\ \gamma>\beta}} \langle u_{\alpha\beta\gamma}(z_\alpha, z_\beta, z_\gamma)\rangle + O(4+) \right] \tag{5}$$

The internal energy and the system energy for each solute atom, $\alpha$, is simply:

$$U_\alpha^{\mathrm{sys}} = \langle u_\alpha(z_\alpha)\rangle + \frac{1}{2}\sum_{\beta\neq\alpha}^{M} \langle u_{\alpha\beta}(z_\alpha, z_\beta)\rangle + \frac{1}{3}\sum_{\beta\neq\alpha}^{M}\sum_{\substack{\gamma\neq\alpha \\ \gamma>\beta}}^{M} \langle u_{\alpha\beta\gamma}(z_\alpha, z_\beta, z_\gamma)\rangle + O(4+) \tag{6}$$

$$U_\alpha^{\mathrm{int}} = \langle u_\alpha(z_\alpha)\rangle + \frac{1}{2}\sum_{\substack{\beta\in S \\ \beta\neq\alpha}} \langle u_{\alpha\beta}(z_\alpha, z_\beta)\rangle + \frac{1}{3}\sum_{\substack{\beta\in S \\ \beta\neq\alpha}}\sum_{\substack{\gamma\in S \\ \gamma\neq\alpha,\ \gamma>\beta}} \langle u_{\alpha\beta\gamma}(z_\alpha, z_\beta, z_\gamma)\rangle + O(4+) \tag{7}$$

And the total energies for each quantity are simply the sums over all solute atoms:

$$U_{\mathrm{sol}}^{\mathrm{sys}} = \sum_{\alpha\in S} U_\alpha^{\mathrm{sys}}, \qquad U_{\mathrm{sol}}^{\mathrm{int}} = \sum_{\alpha\in S} U_\alpha^{\mathrm{int}} \tag{8}$$

## 2.2 Evaluation of the solute energy

Classical molecular mechanics force fields are typically constructed following the form used in the AMBER force field. We have used this forcefield in the results section:

$$E_{system} = \sum_{bonds} k_b(x_i - x_i^\circ)^2 + \sum_{angles} k_a(\theta_i - \theta_i^\circ)^2 + \sum_{torsions} \sum_{n} [1 + cos(n\omega_i - \gamma_i)] + \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} \left\{ \varepsilon_{ij} \left[ \left( \frac{r_{ij}^\circ}{r_{ij}} \right)^{12} - 2 \left( \frac{r_{ij}^\circ}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\varepsilon_\circ r_{ij}} \right\} \tag{9}$$

Here $E_{system}$ is the total energy of the system for a given configuration, $\mathbf{z}$, and follows the naming convention in Amber. The first three terms are the usual harmonic bond, harmonic angle, and periodic torsion terms, and the last two are the Lennard-Jones and Coulomb interactions between atom pairs $i$ and $j$ at separation $r_{ij}$. Every term can be evaluated at each frame of the simulation, so its contribution can be tabulated and mapped onto individual atoms, bonds, angles, or torsions.

Unlike the mutual information expansion used for the entropy below, Equation 1 need not be truncated for Amber or other commonly used force fields. Bonded terms extend at most to fourth order, since a torsion involves four atoms, and non-bonded terms terminate at second order, so the multibody expansion is finite. Every term is therefore retained in the evaluation, and the energetic mapping carries no truncation error.

The internal solute energy for the Amber force field retains the bond, angle, and torsion energies of the solute together with the intramolecular nonbonded energies, including the scaled 1,4 pairs, so that the internal energy is the bond angle torsion (BAT) plus 1,4 energy of the solute. Because bonded terms never span solute and solvent and non-bonded terms terminate at second order, the two quantities differ by exactly the solute's half share of the solute-solvent interaction energy:

$$U^{\text{sys}}_{\text{solute}} - U^{\text{int}}_{\text{sol}} = \frac{1}{2} \sum_{\alpha \in \mathcal{S}} \sum_{\beta \notin \mathcal{S}} \langle u_{\alpha\beta}(z_\alpha, z_\beta) \rangle \tag{10}$$

We assign the energy of each term to the atoms involved such that every atom participating in a given interaction shares that energy equally: the two atoms of a bond equally share the bond energy, the three atoms of an angle share the angle energy, and the four atoms of a torsion share the torsional energy. This atom-wise partitioning for solute atoms has been implemented in the `enedecomp` package in Amber Tools[8]. In addition to the total energy assigned to each atom, `enedecomp` resolves that energy into the individual force field components of Equation 9, so that the bond, angle, torsion, Lennard-Jones, and electrostatic contributions are each mapped separately. Combined with the freedom to restrict the sums to arbitrary sets of atoms, this allows specific energetic interactions to be isolated and mapped. For example, one can extract the van der Waals and electrostatic interaction energies between the ligand and the protein as a whole, between a single ligand functional group and the entire protein, or between that functional group and a particular side chain lining the binding site.

The overall requirement for per atom energy decomposition is that the sum of the energy of each individual atom equals the total energy of the system. Bonded energy terms (bonds, angles, dihedrals) are decomposed on a per-atom basis by assigning atom $i$ the total energy for the term involving atom $i$ evenly divided by the number of component atoms in the term $M$,

$$E_{Term}(i, M) = \frac{E_{Term}^{Total}}{M} \tag{11}$$

so e.g. for a Hooke's Law bonded (2 atom) term:

$$E_{Bond}(i, j) = \frac{1}{2} K_B \left(r_i - r_j\right)^2, \; E_{Bond}(i) = \frac{E_{Bond}(i, j)}{2} \tag{12}$$

where $K_B$ is the bond force constant and $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$ are the positions of atoms $i$ and $j$ respectively. In the case of non-bonded terms, one needs to consider whether periodic boundary conditions (PBC) are present or not, and if they are present how to handle the decomposition. The simple case is if periodic boundary conditions are not present; the decomposition is treated the same as the bonded terms (i.e. the total energy between atoms $i$ and $j$ is divided evenly between the two atoms). However, when periodic boundary conditions are present, the energy decomposition needs to be handled more carefully to ensure that the sum over atoms still equals the total energy.

Here we discuss the decomposition of energy with PBC when using the particle mesh Ewald (PME)[9] method to handle electrostatics and a long-range correction for van der Waals interactions calculated via a Lennard-Jones potential[10]. The PME electrostatic energy on each atom $i$ can be broken down into 4 terms:

$$E_{Elec}(i) = E_{Self}(i) + E_{Direct}(i) + E_{Recip}(i) + E_{Adjust}(i) \tag{13}$$

where $E_{Self}$ is the correction for the Gaussian charge distribution of atom $i$ acting on its own site plus the net-neutralizing "plasma", $E_{Direct}$ is the direct space electrostatic energy for atom $i$ within the cutoff, $E_{Recip}$ is the electrostatic energy for atom $i$ outside the cutoff (in reciprocal space), and $E_{Adjust}$ is the correction (also outside the cutoff) for the fact that 1-2, 1-3, and 1-4 atomic interactions are excluded in direct space. This decomposition is similar to how PME was integrated with Grid Inhomogeneous Solvation Theory in CPPTRAJ[11].

The $E_{Self}$ term can be decomposed by atom via:

$$E_{Self}(i) = -\frac{\alpha}{\sqrt{\pi}} q_i^2 + \frac{E_{Plasma}}{N} \tag{14}$$

$$E_{Plasma} = -\frac{1}{2} \frac{\pi}{\alpha^2 V} \left( \sum_{i=1}^{N} q_i \right)^2 \tag{15}$$

where $\alpha$ is the Ewald coefficient, $N$ is the number of atoms, $q_i$ is the charge on atom $i$, and $V$ is the total volume of the system.

The $E_{Direct}$ term can be decomposed by atom via:

$$E_{Direct}(i) = \sum_{j}^{N'} \frac{1}{2} \frac{q_i q_j erfc\left(\alpha * r_{ij}\right)}{r_{ij}} \tag{16}$$

where the summation $N'$ is over all atoms $\boldsymbol{j}$ within the direct space cutoff, *erfc* is the complementary error function and $\boldsymbol{r_{ij}}$ is the minimum imaged distance between atoms $i$ and $\boldsymbol{j}$. The pre-factor of $\frac{1}{2}$ divides the total interaction energy evenly between the two atoms.

The $E_{Recip}$ term can be decomposed by atom via:

$$E_{Recip}(i) = \frac{1}{2} q_i \psi_i \tag{17}$$

where $\psi_i$ is the reciprocal potential at the position of atom $i$, calculated via the helPME library (`https://github.com/andysim/helpme`).

Finally, the $E_{Adjust}$ term can be decomposed by atom via:

$$E_{Adjust}(i) = \sum_{j}^{N''} -\frac{1}{2} \frac{q_i q_j erf\left(\alpha * r_{ij}\right)}{r_{ij}} \tag{18}$$

where the summation $N''$ is over all atoms $j$ outside the direct space cutoff, *erfc* is the error function. Again, the pre-factor of $\frac{1}{2}$ divides the total interaction energy evenly between the two atoms.

The van der Waals/Lennard-Jones energy on each atom i can be broken down into 2 terms:

$$E_{VDW}(i) = E_{LJ}(i) + E_{VDWcorrect}(i) \tag{19}$$

where $E_{LJ}$ is the Lennard-Jones energy for atom $i$ for interactions inside the direct space cutoff, and $\mathrm{E_{VDWcorrect}}$ is the long-range correction to account for interactions outside the cutoff.

The $\mathrm{E_{VDW}}$ term can decomposed by atom via:

$$E_{VDW}(i) = \sum_{j}^{N'} \frac{1}{2} \left( \frac{A}{r_{ij}^{12}} - \frac{B}{r_{ij}^{6}} \right) * f_{Switch}\left(r_{ij}^2\right) \tag{20}$$

where the summation $N'$ is over all atoms $j$ within the direct space cutoff, $A$ and $B$ are the Lennard-Jones coefficients for the interaction of atoms $i$ and $j$ and $f_{Switch}$ is a switching function which ensures the energy goes smoothly to zero outside the direct space cutoff.[12]

The $E_{VDWcorrect}$ term can be decomposed by atom via:

$$E_{VDWcorrect}(i) = \frac{-2\pi * VDWfac}{3V\varepsilon^3} * \frac{VDWterm(i)}{Ntypes_i} \tag{21}$$

$$VDWterm(i) = \sum_{j=1}^{T} Ntypes_i * Ntypes_j * B \tag{22}$$

$$VDWfac = \sum_{i=1}^{T} VDWterm(i) \tag{23}$$

where $V$ is the system volume, $\epsilon$ is the direct space cutoff, $Ntypes_i$ is the total number of atoms that have the same Lennard-Jones type as atom $i$, $B$ is the Lennard-Jones B coefficient for atoms $i$ and $j$, and $T$ is the total number of Lennard-Jones atom types.

## 2.3 Entropy decomposition

### 2.3.1 Separating solute and solvent contributions to the entropy

The entropy of a solvated system of solutes can be represented by:

$$S_{\text{sys}} = -k_B \int P(\mathbf{x}, \mathbf{r}^{N_w}) \, \ln P(\mathbf{x}, \mathbf{r}^{N_w}) \, d\mathbf{x} \, d\mathbf{r}^{N_w} \tag{24}$$

where $k_B$ is Boltzmann's constant, $P$ is the probability of sampling a given configuration, $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is the set of $N$ coordinates that fully specifies the conformation of the solutes, and $\mathbf{r}^{N_w}$ fully describes the positions and orientations of the $N_w$ solvent molecules.

Following the prescription of the Flexi-GIST paper [13], we use a conditional probability to separate the solute and solvent contributions to the entropy. Conditioning the solvent distribution on the solute configuration yields

$$S_{\text{sys}} = -k_B \int P(\mathbf{x}) P(\mathbf{r}^{N_w}|\mathbf{x}) \, \ln P(\mathbf{x}) P(\mathbf{r}^{N_w}|\mathbf{x}) \, d\mathbf{x} \, d\mathbf{r}^{N_w} \tag{25}$$

which separates into two exact contributions,

$$S_{\text{solute}} = -k_B \int P(\mathbf{x}) \, \ln P(\mathbf{x}) \, d\mathbf{x} \tag{26}$$

$$S_{\text{solvent}} = -k_B \int P(\mathbf{x}) P(\mathbf{r}^{N_w}|\mathbf{x}) \, \ln P(\mathbf{r}^{N_w}|\mathbf{x}) \, d\mathbf{x} \, d\mathbf{r}^{N_w} \tag{27}$$

so that $S_{\text{sys}} = S_{\text{solute}} + S_{\text{solvent}}$. The separation is exact and requires no assumption about the coupling between the solutes and the solvent. Our prior work developed the second of these terms, Equation 27. The present work develops an approach to provide an atomistic mapping of the first term, the solute entropy. We note that as written, the entropy due to correlations between the solutes and the solvent, are contained in $S_{\text{solvent}}$, while the correlations among the solutes themselves are contained in $S_{\text{solute}}$.

### 2.3.2 Estimating and Mapping of the Solute Entropy

To estimate the solute configurational entropy, we apply a mutual information expansion (MIE) of the full configurational entropy, truncated at second order, and estimate both the first-order (single-variable) and second-order (pairwise mutual information) terms using a nearest-neighbors approach. The configurational entropy is defined by the expression for the differential entropy:

$$S = -k \int P(x_1, x_2, ..., x_n) \ln P(x_1, x_2, ..., x_n), dx_1 ... dx_n \tag{28}$$

where $x_1, x_2, ....x_n$ is a set of coordinates fully describing the molecular configuration, and $P(x_1, x_2, ....x_n)$ is the corresponding probability distribution in phase space. For a molecule, this configuration space can be fully described by a set of 3N-6 internal coordinates, where N is the number of atoms in the molecule. By applying the Generalized Kirkwood Superposition

Approximation[2] to the probability distribution and expanding the logarithm in equation 28, we produce the MIE[14], here shown to third order:

$$S = \sum_{i=1}^{n} S_1(x_i) - \sum_{i=1}^{n}\sum_{j>i}^{n} I_2(x_i, x_j) + \sum_{i=1}^{n}\sum_{j>i}^{n}\sum_{k>j}^{n} I_3(x_i, x_j, x_k) + ... \tag{29}$$

Equation 29 has the same structure as the energy expansion in Equation 1, in that each term is associated with a specific subset of coordinates and therefore entropy can be divided and assigned to each coordinate in a similar manner:

$$S(x_i) = S_1(x_i) - \frac{1}{2}\sum_{j\neq i}^{n} I_2(x_i, x_j) + \frac{1}{3}\sum_{j\neq i}^{n}\sum_{\substack{k\neq i \\ k>j}}^{n} I_3(x_i, x_j, x_k) + \dots \tag{30}$$

For the calculations presented here, we represent the full configuration of the solute using a bond-angle-torsion (BAT) set of internal coordinates, described in Section 2.4, and truncate the mutual information expansion at second order. Writing the pairwise mutual information as $I_2(x_i, x_j) = S_1(x_i) + S_1(x_j) - S_2(x_i, x_j)$ gives

$$S \approx \sum_{i=1}^{n} S_1(x_i) - \sum_{i=1}^{n}\sum_{j>i}^{n} [S_1(x_i) + S_1(x_j) - S_2(x_i, x_j)] \tag{31}$$

$$S \approx \sum_{i=1}^{n}\sum_{j>1}^{n} S_2(x_i, x_j) - (n-2)\sum_{i=1}^{n} S_1(x_i) \tag{32}$$

where $n = 3N - 6$ is the number of internal coordinates.

We obtain estimates the truncated configurational entropy (31) from the probability distribution of these internal coordinates, sampled from molecular dynamics trajectories, using a nearest-neighbors (NN) algorithm combined with the mutual information expansion described above. The method relies on the nearest-neighbor volume: the volume of a sphere centered on a given data point with radius equal to the distance to its nearest neighbor. The average logarithm of this quantity, with appropriate correction terms, asymptotically converges to the thermodynamic Shannon entropy[15, 16]:

$$S = \frac{d}{n}\sum_{i=1}^{n} ln\left(R_{i,k}\right) + ln\frac{n\pi^{d/2}}{\Gamma\left(\frac{1}{2}d+1\right)} - L_{k-1} + \gamma \tag{33}$$

where $d$ is the dimensionality of the data set, $n$ is the number of data points, $R_{i,k}$ is the distance from data point $i$ to its $k$th nearest neighbor, $L_{k-1}$ is a constant depending on neighbor order $k$, and $\gamma$ is the Euler-Mascheroni constant.

The resulting entropy values for each BAT coordinate are mapped onto individual atoms using the same logic applied to the energy terms above: the entropy of a bond is split evenly between its two atoms, the entropy of an angle among its three atoms, and the entropy of a torsion among its four atoms. For the mutual information (pairwise) terms, the entropy is first divided between the two internal coordinates involved, and each half is then further divided among the atoms of that coordinate, following the same scheme used for the one-dimensional terms. Because both energy and entropy can be partitioned onto atoms using

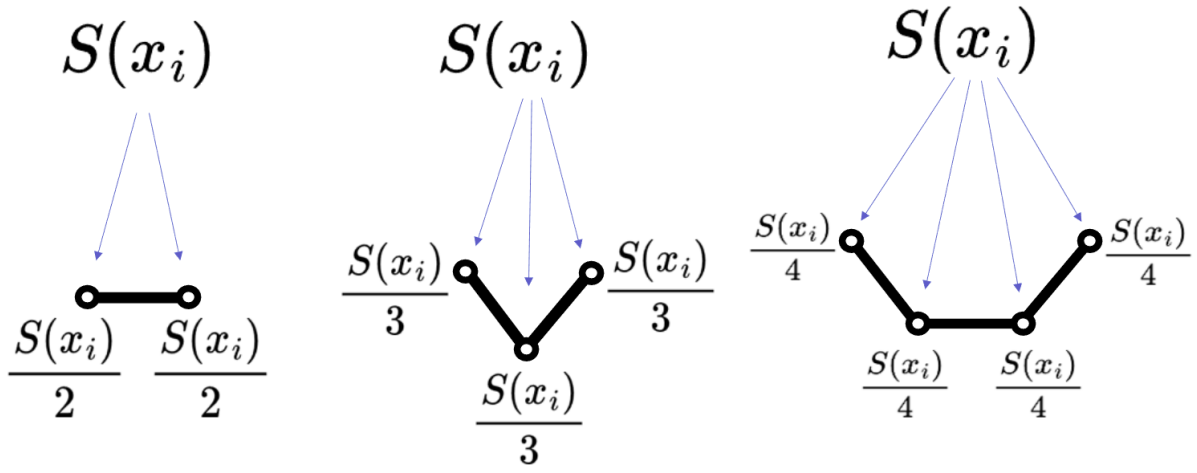



Figure 1: Illustration showing how the entropy of a given internal coordinate is divided among the atoms comprising it. Atomic Energy assignments are done in the same manner.

this consistent bond/angle/torsion scheme, the two together allow the full free energy to be mapped completely onto the atoms each molecule.

## 2.4 Bond-Angle-Torsion List

We generated a Bond-Angle-Torsion set of internal coordinates[17] using the MDAnalysis BAT module[18, 19] which generated 3N-6 total coordinates, where N is the total number of atoms in the solute. Each internal coordinate was saved as a list of atom indices, with bonds having two atoms, angles three, and torsions four. All bonds involving hydrogen atoms were selectively removed from the list due to the SHAKE algorithm utilized in AmberMD, which attempts to keep all bonds to hydrogen at a fixed length throughout the simulation. To reduce the number of torsions, any torsions that shared two central atoms were assumed to have similar angular distributions that only differed by some phase angle. Since their distributions could be characterized by a single torsion, only one of each redundant torsion was retained (the first encountered in the BAT list), and the others were removed, as previously done[17] These changes reduce the total number of internal coordinates in the set to $3N-6-N_H-N_T$, where $N_H$ is the number of hydrogen atoms, and $N_T$ is the number of redundant torsions.

## 2.5 Nearest Neighbor Entropy Calculations

Nearest neighbor entropy calculations were performed separately for one-body and two-body terms. For the one-body terms, each individual internal coordinate was treated as a 1D vector and for the two-body terms each internal coordinate pair was treated as a two-dimensional vector. The core computational engine of the NN algorithm was implemented in Rust for high performance, utilizing the Kiddo KDTree to perform the nearest neighbor searches and Rayon to parallelize the calculations, and exposed as a native Python extension module. The calculated values were then summed using Equation (33).

## 2.6 Simulation Details

### 2.6.1 Protein Preparation

Crystal structures were obtained from the Protein Data Bank (PPARA: 2P54). Proteins were prepared using Schrödinger Maestro Protein Preparation Wizard. Crystallographic Water molecules farther than 5 Å away from the ligand were removed. The preprocessing step included filling missing side chains and loops, adding hydrogen, converting selenomethionine to methionine, capping terminal residues, and assigning bond orders. The prepared protein structures were exported as PDB files and further processed with pdb4amber to ensure compatibility with AMBER, including standardization of atom and residue names of capping residues and removal of extraneous hydrogen.

### 2.6.2 Ligand Preparation

Ligands were extracted from crystal structures using PyMOL and processed with RDKit to assign correct bond orders based on SMILES strings. Hydrogen atoms were added, and ligands were saved as SDF files to preserve bond order information. Protonation states were manually adjusted, with carboxylic acid groups modeled in their deprotonated form (-COO-), yielding a net charge of –1. Final ligand structures were saved as PDB files for parameterization.

### 2.6.3 Force Field Parametrization and Complex Preparation

Ligands were parameterized using GAFF2 with AM1-BCC partial charges calculated via antechamber. Proteins were parameterized with the ff19SB force field. Three systems were prepared independently: the protein, the ligand, and the protein-ligand complex. Systems were solvated in explicit OPC water boxes with 10 Å padding for protein-containing systems and 20 Å padding for the ligand using tleap.

### 2.6.4 Simulation Parameters

Initial minimization was performed for up to 20,000 steps (1,500 steepest descent, remainder conjugate gradient) with strong harmonic restraints (100 kcal/mol·Å$^2$) on all non-water atoms, followed by a second minimization of the same length with restraints reduced to only protein backbone heavy atoms (C$\alpha$, C, N) and bound ions. Systems were then heated from 0 to 300 K over 240 ps in the NVT ensemble with a 1 fs timestep, maintaining 100 kcal/mol·Å$^2$ restraints on backbone heavy atoms, the phosphotyrosine residue, and ions. This was followed by a short NPT equilibration (100 ps, 300 K, 1 atm, 2 fs timestep) in which 100 kcal/mol·Å$^2$ restraints on backbone and carbonyl atoms were held constant under a Berendsen barostat (0.5 ps relaxation time). A subsequent 10 ns NPT equilibration was performed under the same barostat settings, during which restraints on all non-water heavy atoms were gradually reduced from 100 to 2.5 kcal/mol·Å$^2$ over the first 9 ns and then held constant for the final nanosecond. An additional 10 ns equilibration was conducted in the NPT ensemble with constant restraints of 2.5 kcal/mol·Å$^2$ on the same atom selection, followed by a 2 ns fully unrestrained NPT equilibration step. Unrestrained production

simulations were then run for 2 µs in the NPT ensemble at 300 K and 1 atm with a 2 fs timestep, using a Monte Carlo barostat (1.0 ps relaxation time), with coordinates saved every 2 ps for analysis. The final 100 ns(50,000 frames) of the production simulations were saved for further processing. All molecular dynamics simulations used periodic boundary conditions, particle mesh Ewald summation for long-range electrostatics (10 Å cutoff), Langevin dynamics for temperature control, and SHAKE constraints on hydrogen bonds. The simulations were repeated in triplicate, using different starting seeds.

# 3 Software Details

## 3.1 Workflow

The computational workflow for creating the atomic energy and entropy mapping, summarized by Figure 2, is as follows:

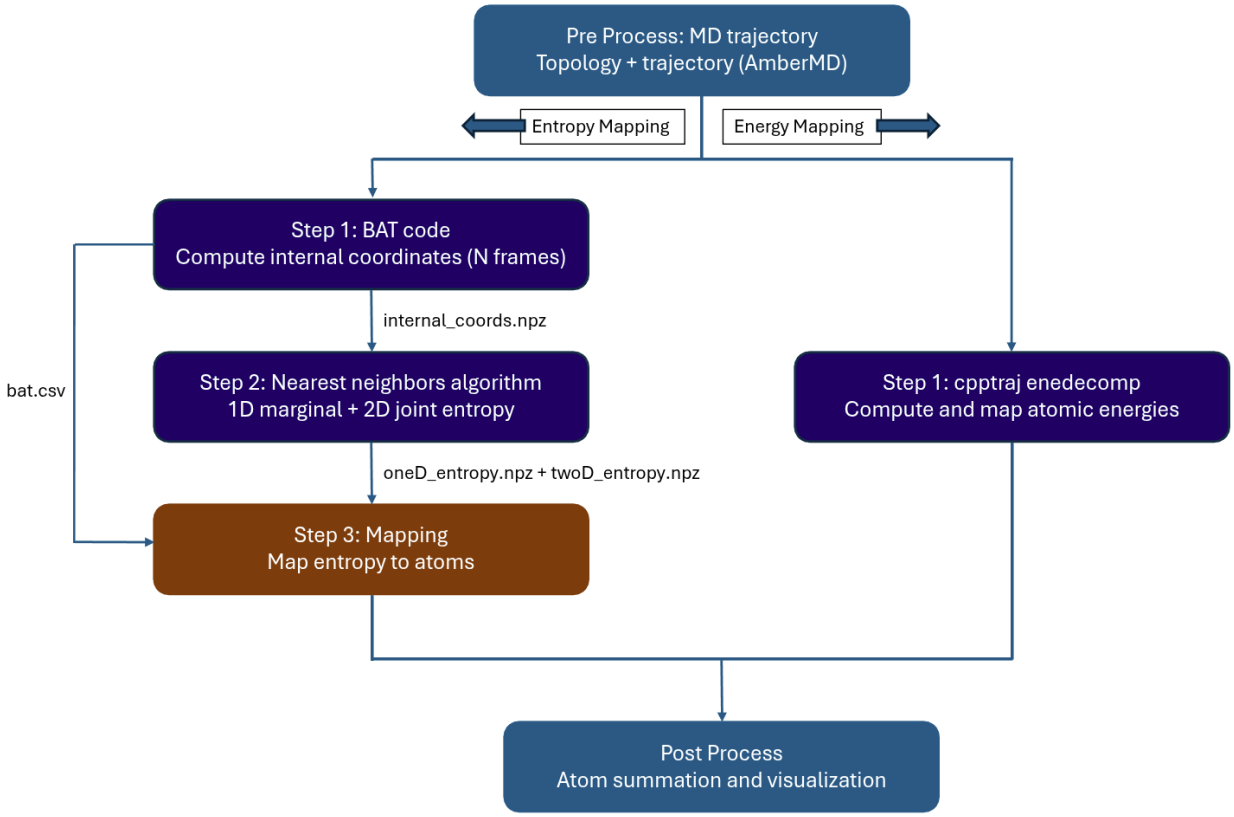



Figure 2: Diagram of the Solute Free Energy Mapping workflow.

### 3.1.1 Generate a System Trajectory

Prepare and simulate the system of interest using AmberMD (see Section 2.6). The BAT code computes internal coordinates directly from the atomic positions, so it will return incorrect

bond lengths, angles, and torsions if the solute is split across the periodic boundary of the simulation box. To avoid this problem, we recommend that the trajectory be 'wrapped' before running the BAT code, so that the solute is whole and fully contained within the periodic boundaries in every frame, using `cpptraj` or an equivalent tool. Save the topology file (prmtop) and the rewrapped trajectory (netcdf) as input for the next step.

### 3.1.2 Create a Bond-Angle-Torsion Trajectory

The BAT code (`BAT.py`) converts the Cartesian coordinates of the solute from the MD trajectory into the internal bond-angle-torsion coordinates used for the entropy calculation. It builds the list of internal coordinates that defines the solute's configuration space, evaluates each of those coordinates for every frame requested, and writes both the coordinate definitions and the resulting internal-coordinate trajectory to disk. The script is written in Python and generates the coordinate list with the MDAnalysis BAT module, which is then reduced as described in Section 2.4, so that bonds constrained by SHAKE and redundant torsions are excluded before any entropy is computed.

The usage of `BAT.py` is: Usage: `BAT.py -i topology.prmtop -x trajectory.nc -n num_frames`

The code takes three inputs: the Amber topology.prmtop file, the trajectory.nc, and the number of frames to analyze. The first `num_frames` frames of the trajectory are processed.

The script outputs two files: `bat.csv` and `internal_coords.npz`.

The first output file, `bat.csv`, is contains the Bond-Angle-Torsion list: one row per internal coordinate, each row containing the topology atom indices that define that coordinate. The number of indices in a row identifies the coordinate type, with two indices for a bond, three for an angle, and four for a torsion. Atom indices begin at zero, following the MDAnalysis convention, so index 0 refers to the first atom in the topology file. The order of the rows in `bat.csv` defines the canonical index of each internal coordinate and is used by every downstream step, so the same file is read again at the mapping stage (Section 3.1.4) to distribute entropies back onto atoms.

The second file, `internal_coords.npz`, contains an $N_f \times N_c$ matrix of the calculated internal coordinates, where $N_f$ is the number of frames analyzed and $N_c$ is the number of internal coordinates retained. Column $j$ of this matrix is the time series of the internal coordinate given in row $j$ of `bat.csv`. Bond lengths are reported in Ångstroms, and bond angles and torsion angles are reported in radians. This matrix is the sole input required by the nearest-neighbors code.

### 3.1.3 Nearest Neighbors Entropy Estimates

The `NN.py` script takes the BAT tractory contained in the internal_coords.npz file and computes the one-dimensional (marginal) entropy of each internal coordinate, and the two-dimensional (joint) entropy of each pair of internal coordinates.

The one-dimensional entropy values are written to oneD_entropy.npz. The entries in this file line up directly with the BAT list: the i-th entropy value is the marginal entropy of the i-th internal coordinate in the BAT list. The two-dimensional entropies are written to twoD_entropy.npz. These entries line up with a "pair list," which enumerates every possible pair of internal coordinates (using their BAT list indices) in lexicographic order, with no duplicates. For example, the first entry in the pair list is [0, 1], so the first value in the two-dimensional entropy list is the joint entropy between the internal coordinates at indices 0 and 1 in the BAT list. More generally, the NN code is input agnostic, capable of calculating the one-body and two-body differential entropies for any $N_f \times N_c$ matrix, where $N_f$ is the number of data points, and $N_c$ is the dimensionality of the data set.

Usage: `NN.py -i internal_coords.npz`
output1: `oneD_entropy.npz`
output2: `twoD_entropy.npz`

### 3.1.4 Mapping

```
Map_entropy_to_atoms.py --bat bat.csv --oned oneD_entropy.npz --twod
twoD_entropy.npz -o atom_entropy.csv
```

Using the bat.csv file and the outputs from the NN code, the calculated entropies are mapped to the atoms according to the scheme outlined in section 2.3.2. The output file atom_entropy.csv contains two columns, the atom index and the entropy mapped to that atom. Once mapped, the atomic contributions can be summed arbitrarily, such as by side-chain, backbone atoms, binding pocket, or chemical moiety.

### 3.1.5 Energy Decomposition with Enedecomp

To run the energy mapping, we utilized the same topology (prmtop) and trajectory (netcdf) from section 3.1.1. These files are processed using the enedecomp action in AmberTools cpptraj. An example script in an example bash script is as follows:

```
cpptraj <<EOF
parm topology.prmtop
trajin trajectory.nc
enedecomp name_of_data_set :1-150 out energy_decomp.out savecomponents pme
run
EOF
```

The `topology.prmtop` and `trajectory.nc` serve as the input files. The enedecomp action has a number of input parameters. `Name_of_data_set` names the cpptraj output data set. The :1-150 serves as an example atom mask, calculating the energy of only those atoms included in the mask. Note that cpptraj utilizes the AmberMD numbering convention, where atom and residue numbering starts at 1. `Energy_decomp.out` serves as the output

file where the computed energies are saved. The `savecomponents` flag tells the enedecomp action to write out the specific contributions for each atom, such as bond, atom, dihedral (torsions), 1-4 van der Waals, 1-4 electrostatics, van der Waals, and electrostatic energies. If not specified, the reported energy for each atom is the sum of these components. If the `pme` flag is specified, the non-bonded terms will use PME for electrostatics and a long-range periodic correction for van der Waals, otherwise a simple model with no cutoff will be used. Enedecomp features a number of input parameters for specifying the details of the PME calculations that were omitted here but can be found in the online documentation. The enedecomp output file contains a number of columns which are the atom index, the average total energy contribution of that atom, the average bond energy, average angle energy, average dihedral(torsion) energy, average 1-4 van der Waals energy, average 1-4 electrostatic energy, average van der Waals energy, and average electrostatic energy.

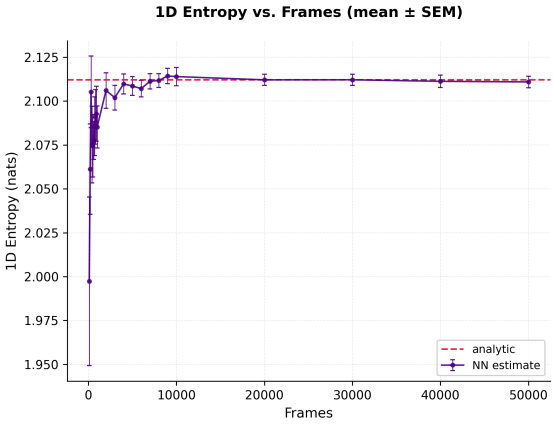
# 4 Results

We show results that validate the NN entropy code for Gaussian probability distribution functions for which analytic results are available, and we demonstrate the application of solute mapping via an end-states analysis of a drug-like ligand (ligand 735) binding to the protein drug target PPAR-$\alpha$ (pdbid:2P54)

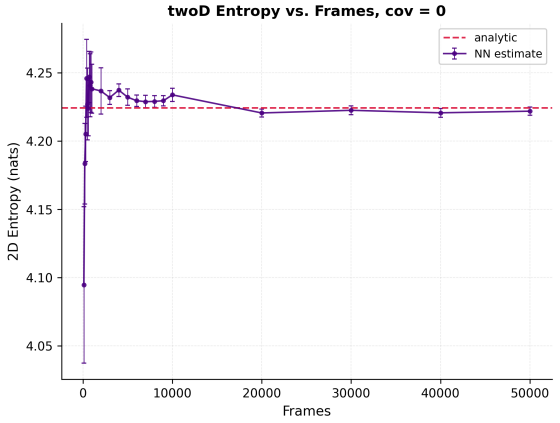
## 4.1 Analytic test cases and convergence properties

We validated the NN estimator by comparing its calculated entropy against the analytical solutions for the differential entropy of Gaussian probability distributions. To test the one-dimensional calculations, a single Gaussian distribution with a standard deviation equal to 2.0 was sampled for 50,000 data points. To test the two-dimensional calculations, bivariate Gaussians with standard deviation equal to 2.0 in both dimensions were sampled for 50,000 data points at each covariance value ranging from 0.0 (no correlation) to 3.6 (highly correlated) in 0.4 increments. For both the one-dimensional and two-dimensional cases, the entropy was calculated using the first N data points from each sample, where N = [100, 200, ..., 1000, 2000, ..., 10000, 20000, ..., 50000] to evaluate quantity convergence as the number of samples increased. This process was then repeated to generate 10 replicates in all. For all of the distributions utilized, the differential entropy can be calculated exactly using equation (28). The random Gaussian samples were generated with the numpy python[20] library.

Figure 3: 1D Convergence 3a and 2D Convergence plot for covariance = 0.0 3b




(a) Estimated entropy (nats) of a univariate Gaussian distribution ($\sigma = 2.0$) as a function of sample size, averaged over 10 trials. The dashed red line indicates the analytical entropy (2.11208 nats) computed from the known distribution parameters. Error bars indicate standard errors of the mean across the ten replicates



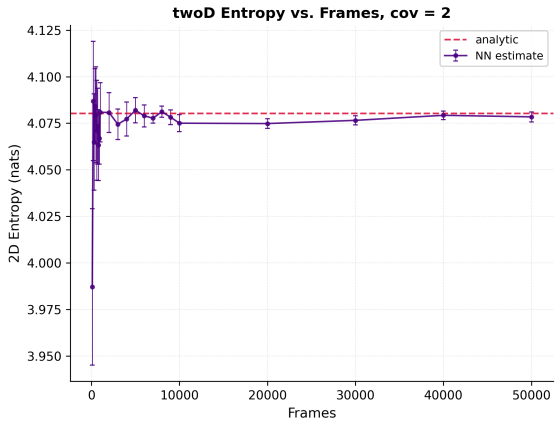

(b) Estimated joint entropy (nats) of a bivariate Gaussian distribution ($\sigma = 2.0$ in both dimensions, covariance = 0.0) as a function of sample size, averaged over 10 trials. The dashed red line indicates the analytical joint entropy (4.22417 nats) computed from the known distribution parameters. Error bars give standard errors of the mean.

In the one-dimensional case, shown in figure 3a, after only 100 data points, the NN estimator was within 5.43% of the analytical solution. By 1000 data points, the estimator was within 1.27% of the analytical solution, and by 10,000 data points, the estimator was within 0.08%. From the bivariate Gaussian calculations, figures 3b,4a,4b, we see even more rapid convergence of the percent error than in the single Gaussian case. Looking at the data for the covariance = 2.0 trial, we see that using just the first 100 data points, the estimator is within 2.28% of the analytical solution. By 200 data points, the estimator is within 0.16% of the analytical solution, and for every value greater than 200 data points, the estimator never has a deviation greater than 0.4% of the analytical solution. Similar convergence was observed at every covariance value tested.

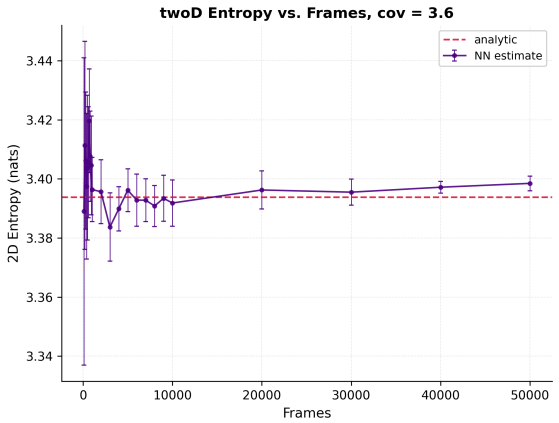
## 4.2 Timing Tests and Computational Costs of the NN Algorithm

To check the code's time dependence versus the number of data points in the sample, a set of 4,000 independent distributions were each sampled for 50,000 data points. The time to calculate the NN entropy of the complete data set was then measured using all of the distributions, but only N number of data points from the set, where N = [10,000, 20,000, 30,000, 40,000, 50,000]. To check the code's time dependence versus the number of dimensions in the data set, a set of 6,000 independent distributions were each sampled for 50,000 data points. The time to calculate the NN entropy of the data set was then measured for every N number of dimensions, where N = [500, 1,000, 1,500, 2,000, 2,500, 3,000, 3,500, 4,000, 4,500, 5,000, 5,500, 6,000]. Computations were performed on compute

Figure 4: 2D Convergence plot for covariance = 2.0 4a and covariance = 3.6 4b




(a) Estimated joint entropy (nats) of a bivariate Gaussian distribution ($\sigma = 2.0$ in both dimensions, covariance = 2.0) as a function of sample size, averaged over 10 trials. The dashed red line indicates the analytical joint entropy (4.08033 nats) computed from the known distribution parameters.



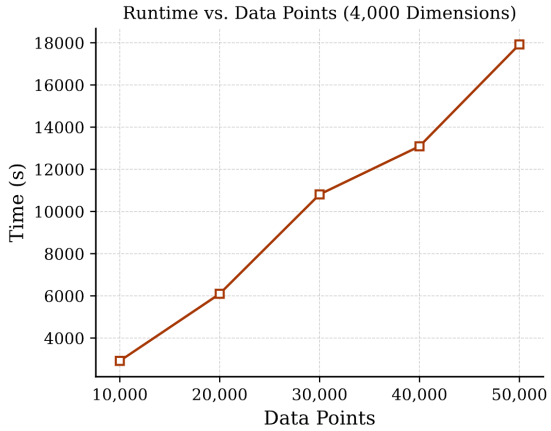

(b) Estimated joint entropy (nats) of a bivariate Gaussian distribution ($\sigma = 2.0$ in both dimensions, covariance = 3.6) as a function of sample size, averaged over 10 trials. The dashed red line indicates the analytical joint entropy (3.39381 nats) computed from the known distribution parameters.

nodes equipped with dual AMD EPYC 9224 processors (48 cores, 96 threads total) and 189 GB RAM, running Rocky Linux (RHEL 9.5, kernel 5.14.0). Only 16 threads were utilized during all trials.

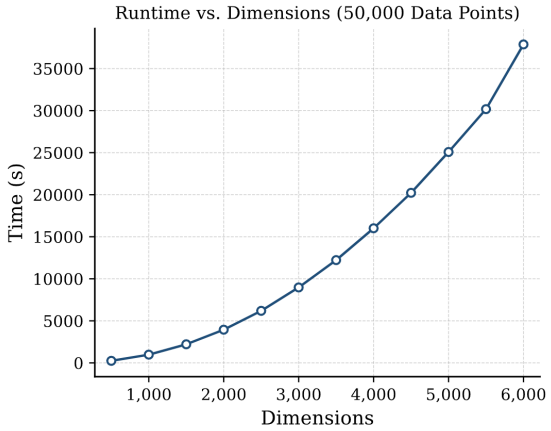
The results of the timing tests can be found in figures 5a and 5b. As the KDTree nearest neighbor algorithm scales $N_f \log(N_f)$, where $N_f$ is the number of data points, we would expect that the time to process scales nearly linearly with the number of frames. This is in agreement with our data, as the coefficient of determination was calculated to be 0.9905, showing a strong linear correlation between time and data points over the tested range. For the dimension time-dependence, we would expect a quadratic relationship between time and the total number of dimensions being considered. This is due to the fact that for $N_c$ one-dimensional nearest neighbor search and summations, the code needs to process $N_c(N_c\text{-}1)/2$ two-dimensional search and summations. When fitting the data to a polynomial of degree 2, we calculate an adjusted $R^2$ value of 0.9988

## 4.3 An illustrative application of Solute Free energy mapping to PPAR-$\alpha$

Here we demonstrate the utility of solute free energy mapping by applying the methodology to the protein-ligand complex of ligand 735 to PPAR-$\alpha$ (pdbid:2P54).




(a) Nearest Neighbor algorithm runtime versus number of data points in the set. Each trial utilized 4,000 internal coordinates, for a total of 8,002,000 distinct nearest neighbor search and summations. $R^2 = 0.99$




(b) Nearest Neighbor algorithm runtime versus number of dimensions. Each trial utilized 50,000 data point samples. Adjusted $R^2 = 0.9988$ when fitting to a polynomial of degree 2

### 4.3.1 Solute energy and entropy mapping of ligand 735 evaluated by chemical group

First we apply the mapping method to characterize how the energetics of ligand 735 change upon binding. We accomplished this by computing the total energy, internal energy(bond-stretch + angle-bend + dihedral + 1-4vdW + 1-4elec terms), van der Waals, and electrostatic energy changes using `enedecomp` (see section 3.1.5 for both the bound ligand in complex and the unbound ligand in free solution in three independent replicas. In each case, per-atom values were summed into five (thiazole core, amide linker, phenyl ring A, phenoxy-isopropyl-acid head group, and $CF_3$-phenyl tail) chemical groups (Figure 6), allowing the binding-energy cost to be attributed to specific functional groups rather than the ligand as a whole.

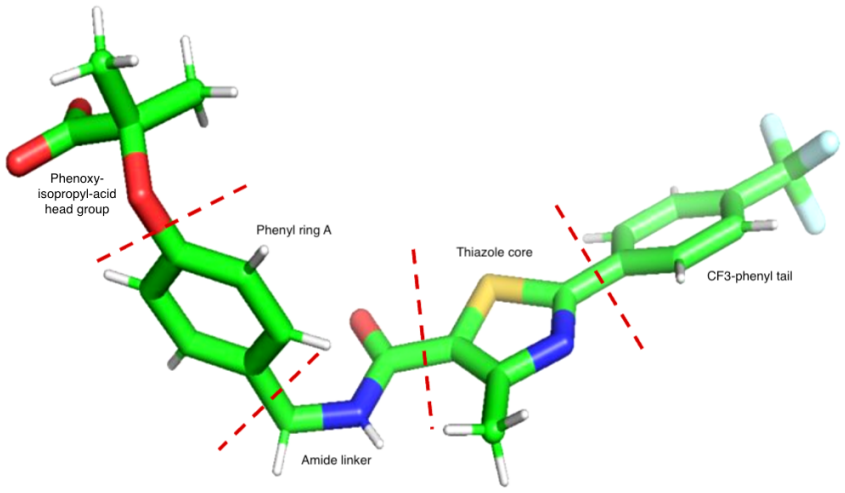



Figure 6: **Ligand chemical-group definitions.** The ligand is divided into five connectivity-defined chemical groups, indicated by red dashed lines: the phenoxy-isopropyl-acid head group (top left), phenyl ring A, the amide linker, the thiazole core, and the $CF_3$-phenyl tail (right). Atoms are colored by element (carbon: green, oxygen: red, nitrogen: blue, sulfur: yellow, fluorine: pale cyan, hydrogen: white). This grouping is used throughout the subsequent per-group energy and entropy decomposition.

Figure 7 shows the total energy and internal strain decomposition. The phenoxy-isopropyl-acid head group consistently dominates the total energy penalty. Figure 8 resolves this further into van der Waals and electrostatic contributions: every chemical group provides a favorable van der Waals contribution and an unfavorableelectrostatical contribution in every replica, and the head group's large electrostatic penalty (12–17 kcal/mol across replicas) accounts for the majority of its net cost, consistent with an electrostatic desolvation penalty rather than a steric clash.

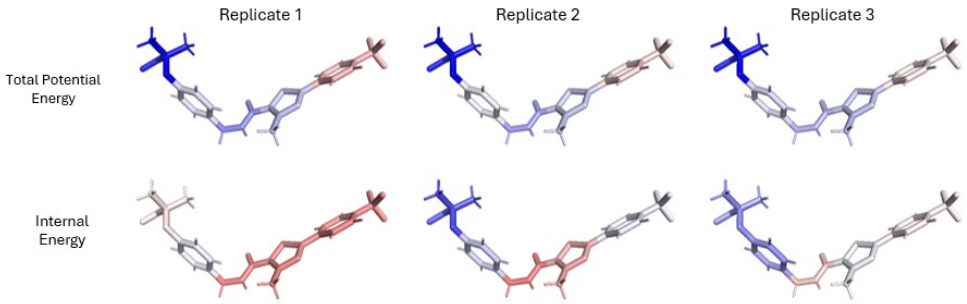



Figure 7: **Total energy (top row) and internal energy (bottom row) mapping of ligand 735 by chemical group across three independent replicas.** Total energy and internal energy is scaled on a -5 to +5 kcal/mol and -0.5 to 0.5 kcal/mol scale respectively. Where red (negative changes) represents favorable upon binding and blue (positive changes) represents unfavorable upon binding.

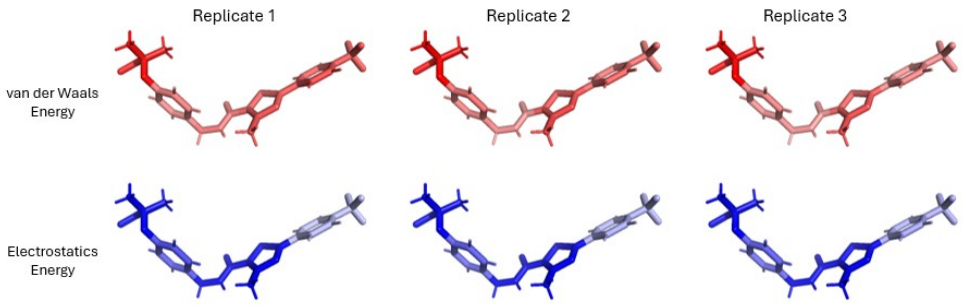



Figure 8: **Van der Waals (top row) and electrostatic (bottom row) mapping of ligand 735 by chemical group across three independent replicas.** Both van der Waals and Electrostatic energy changes share a $-3$ to $+3$ kcal/mol scale, where deep red corresponds to $-3$ kcal/mol and deep blue corresponds to $+3$ kcal/mol. Electrostatic values for all chemical groups exceed this scale's upper bound (all groups are strongly unfavorable, 0.8–17.2$\tilde{\text{k}}$cal/mol across replicas), so the electrostatic row appears uniformly saturated blue, indicating a consistently unfavorable electrostatic contribution across the entire ligand. The phenoxy-isopropyl-acid head group dominates the total energy penalty in every replica.

Ligand total energy, van der Waals, and electrostatic contributions are highly reproducible across the three independent replicate simulations (total energy: $14.3 \pm 2.6$ kcal/mol; van der Waals: $-9.7 \pm 0.2$ kcal/mol; electrostatics: $24.1 \pm 2.6$ kcal/mol), whereas internal strain shows substantially larger relative variability ($0.39 \pm 0.8$ kcal/mol) and is not yet converged at this level of sampling, consistent with the phenoxy-isopropyl-acid group's color-inconsistent strain for replicate-1's shown above.

We performed the solute free energy entropy mapping as described in 2.5 on the bound and unbound simulations of ligand 735. We found when assessing the torsional entropy that every ligand chemical group shows a positive (unfavorable) $-T\Delta S$, consistent with the ligand losing conformational freedom upon complex formation. Figure 9 shows the ligand chemical-group torsional entropy breakdown: the two ring systems (phenyl ring A and the $CF_3$-phenyl tail) show the largest and most reproducible entropy loss across replicates (1.51–1.76 kcal/mol, relative spread 2–13%), while the thiazole core shows the smallest (0.24 kcal/mol).

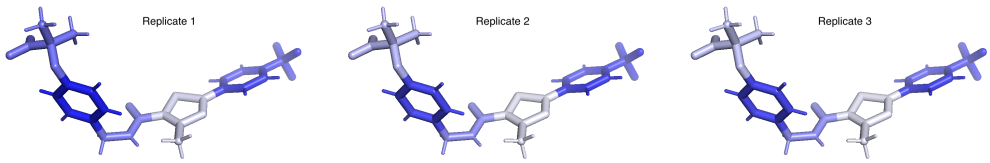



Figure 9: **Torsional entropy mapping of ligand 735 mapped to its chemical groups across three independent replicates.** Colored on a shared $-2.0$ to $+2.0$ kcal/mol scale, where deep red corresponds to $-2.0$ kcal/mol (favorable) and deep blue corresponds to $+2.0$ kcal/mol (unfavorable). Reproducibility varies by group: the $CF_3$-phenyl tail is the most consistent ($1.76 \pm 0.03$ kcal/mol), the phenoxy-isopropyl-acid head group the least ($0.66 \pm 0.33$ kcal/mol); the rank ordering is preserved in every replicate. Values reflect the 1D marginal entropy term only.

### 4.3.2 Solute energy and entropy mapping of PPAR-$\alpha$ binding site residues

We then applied the solute free energy mapping workflow to simulations of the full protein-ligand complex between ligand 735 and PPAR-$\alpha$ (pdbid:2P54). The binding-site potential energy, internal energy (bond-stretch + angle-bend + dihedral + 1-4vdW + 1-4elec), van der Waals, and electrostatic contributions all show substantial cross-replicate variability (total energy: -20 $\pm$ 11 kcal/mol; internal strain: 5.6 $\pm$ 5.7 kcal/mol; van der Waals: -12.5 $\pm$ 4.0 kcal/mol; electrostatics: -13.6 $\pm$ 7.5 kcal/mol), in contrast to the ligand's own energetics, which were largely reproducible aside from internal strain. This binding-site-level variability is consistent with the conformational heterogeneity identified (not shown here) in the apo ensemble across replicates, rather than representing noise specific to any one energy term.

We performed the solute torsional entropy mapping on the bound and unbound simulations of the protein and the majority of binding-site residues show a positive (unfavorable). Figure 14 shows the per-residue binding-site breakdown; residue-level reproducibility is substantially weaker than at the ligand chemical-group level, with the mean cross replicate standard deviation (0.36 kcal/mol) comparable in magnitude to the mean $-T\Delta S$ value itself (0.31 kcal/mol), indicating that individual-residue entropy estimates are not yet converged at this sampling depth.

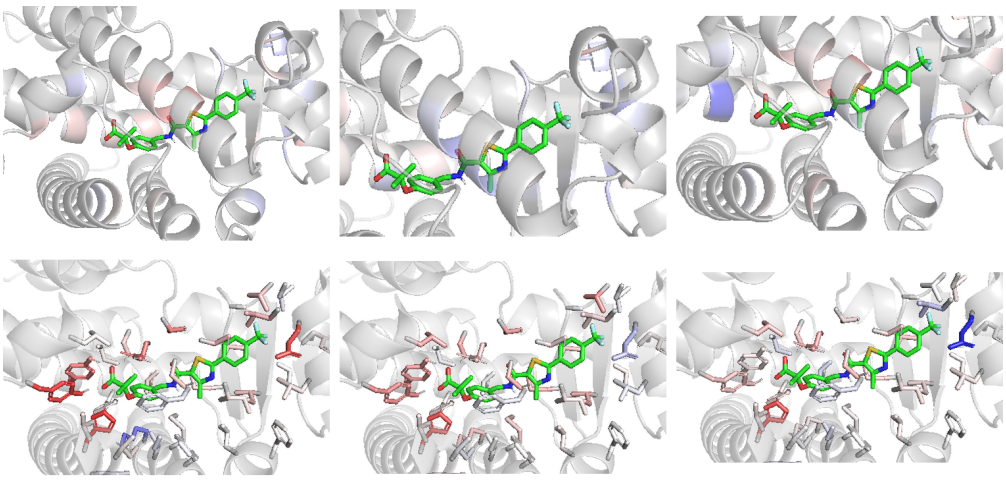


Figure 10: **Total potential energy mapping of binding-site residues split by backbone and side-chain across three independent replicas.** Top row: backbone electrostatic energy change (bound − unbound) per residue. Bottom row: side-chain electrostatic delta per residue. Both rows colored on a shared −5.0 to +5.0 kcal/mol scale, where deep red corresponds to −5 kcal/mol and deep blue corresponds to +5 kcal/mol; values exceeding this range saturate at the corresponding extreme. Several side-chain residues show strongly unfavorable (blue) electrostatic contributions reproducibly across all three replicas. The ligand (green) is shown for spatial reference only and is not itself colored by this mapping.

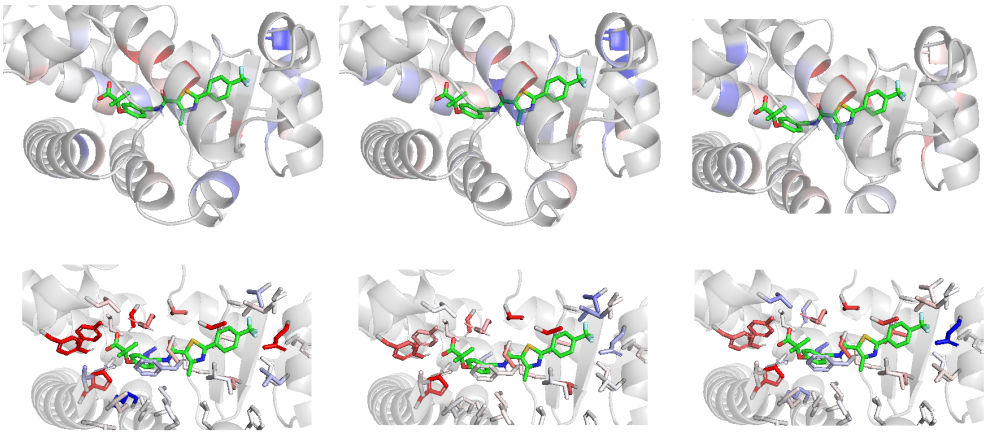


Figure 13: **Electrostatic energy mapping of binding-site residues split by backbone and side-chain across three independent replicas.** Top row: backbone electrostatic energy change (bound − unbound) per residue. Bottom row: side-chain electrostatic delta per residue. Both rows colored on a shared −3.0 to +3.0 kcal/mol scale, where deep red corresponds to −3 kcal/mol and deep blue corresponds to +3 kcal/mol; values exceeding this range saturate at the corresponding extreme. Several side-chain residues show strongly unfavorable (blue) electrostatic contributions reproducibly across all three replicas. The ligand (green) is shown for spatial reference only and is not itself colored by this mapping.

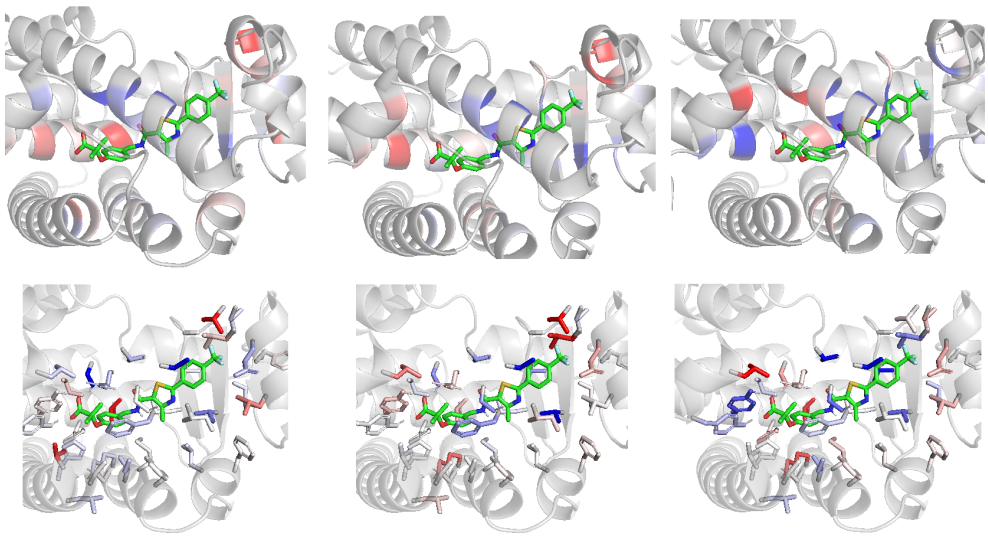


Figure 11: **Internal energy mapping of PPAR-$\alpha$ binding-site residues split by backbone and side-chain across three independent replicas.** Top row: backbone strain energy change (bound − unbound), summed per residue and mapped onto backbone atoms (N, C$\alpha$, C, O). Bottom row: side-chain strain delta summed per residue and mapped onto side-chain heavy atoms. Both rows colored on a shared −5.0 to 5.0 kcal/mol scale, where deep red corresponds to −5.0 kcal/mol and deep blue corresponds to +5.0 kcal/mol. Cys75 induces the largest internal strain among binding-site residues, with a consistently unfavorable (positive) contribution across all three replicates (mean = 6.63 ± 1.90 kcal/mol). The ligand (green) is shown for spatial reference only and is not itself colored by this mapping.

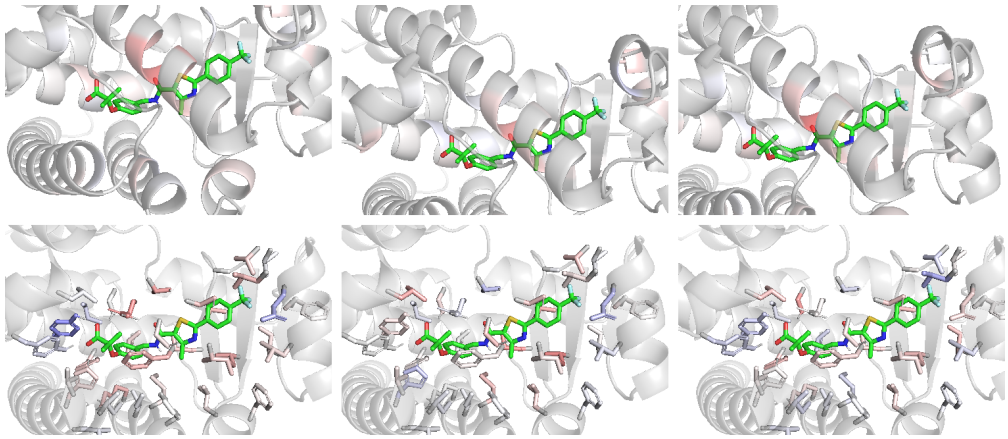


Figure 12: **Van der Waals (steric) energy mapping of binding-site residues split by backbone and side-chain across three independent replicas.** Top row: backbone van der Waals energy change (bound − unbound) per residue. Bottom row: side-chain van der Waals delta per residue. Both rows colored on a shared −3 to +3 kcal/mol scale, where deep red corresponds to −3.0 kcal/mol and deep blue corresponds to +3.0 kcal/mol. The ligand (green) is shown for spatial reference only and is not itself colored by this mapping.

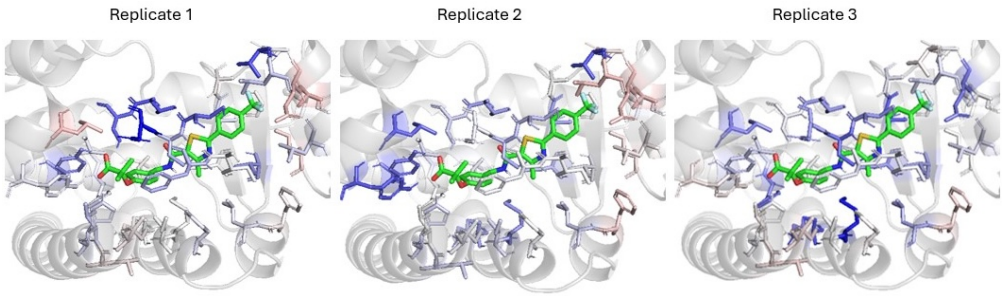



Figure 14: **Torsional entropy mapping of binding-site residues across three independent replicates.** Colored on a shared $-2.0$ to $+2.0$ kcal/mol scale (red favorable, blue unfavorable). Gln77, the residue with the largest mean unfavorable contribution, is also among the least reproducible (std = 0.92 kcal/mol). The ligand (green) is shown for spatial reference only. Values reflect the 1D marginal entropy term only; the second-order pairwise mutual information correction is not yet included.

### 4.3.3 Evaluation of computational runtime of the solute free energy mapping method to PPAR-$\alpha$

To evaluate the computational cost of explicit-solvent PME decomposition relative to a solvent-stripped alternative, the `enedecomp` workflow above was additionally run on solvent-stripped trajectories and topologies (protein and ligand atoms only, no water), using the same PME settings, for all three system types and all three replicates. All calculations were run single-threaded on one CPU core per calculation (Intel Core i7-9700K, 3.60 GHz). Timing results are summarized in figure 15.

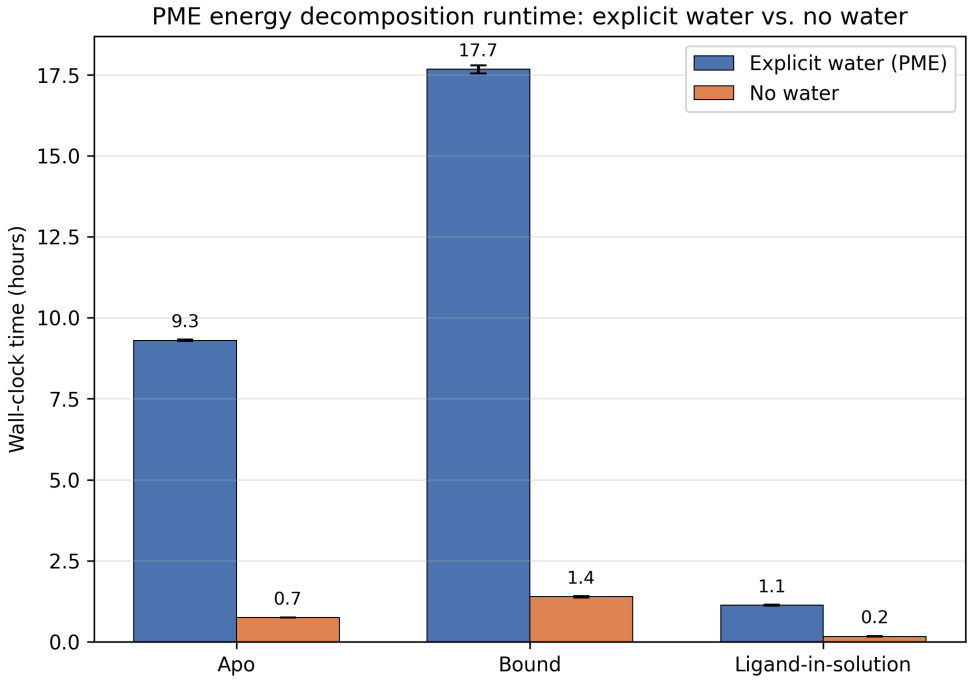



Figure 15: **Energy decomposition runtime, explicit water vs. no water.** Mean wall-clock time (hours) across three replicates; error bars show standard deviation. Full computational details are given above in section 3.1.5.

# 5 Discussion

While free energy methods such as thermodynamic integration and free energy perturbation have been reasonably successful at predicting the change in affinity of a lead molecule upon modification, they provide little to no insight into where on the ligand modifications should be made to achieve gains in affinity. Solvation thermodynamic mapping tools such as GIST and WaterMap have filled this gap, guiding lead discovery and providing an understanding of the solvation contribution to binding affinity and specificity [21, 22, 23]. These methods are generally less accurate than free energy methods, and they give only a partial picture of the thermodynamics, accounting for the solvation contribution alone while leaving the energetic and entropic changes of the protein and ligand themselves unaddressed. What they do provide is an indication of where in the binding cavity and by what type of modification gains (or losses) in binding affinity may be realized, and this has given medicinal chemists concrete guidance as to what modifications to make. No tool has provided the equivalent information for the ligand and protein.

Here, we have addressed this gap by providing the theory, computational methods, and software to decompose the energetic and entropic contributions to the free energy of solutes such as a ligand and a protein, and to map these contributions onto atoms and, by summation, onto any chemically meaningful group of atoms such as ligand functional groups, residues, or side chains.

The approach is robust, and the decomposition is not limited to the separation of the

free energy into its energetic and entropic parts. Because the energy and entropy are each built from well defined constituent terms, these components can be estimated and mapped separately as well. For example, the van der Waals and electrostatic components of the energy can be mapped independently of one another, as can the internal valence terms, allowing the physical origin of a favorable or unfavorable contributions to be identified rather than inferred. We have illustrated the application of the methods and software on the binding of ligand 735 to PPAR-$\alpha$, demonstrating how the contributions can be divided into chemically relevant subgroups and mapped for the thermodynamic end states of an unbound and bound ligand.

We anticipate that atomistic solute thermodynamic mapping will find use across a range of problems in which the thermodynamics of a molecular process must be attributed to specific parts of a structure, including protein design, the analysis of allostery, and the study of conformational change. Drug design is the application we have in mind most immediately, where we expect the approach to prove useful in much the same way that solvation thermodynamic mapping has, guiding a medicinal chemist toward the chemical components of a lead that are worth modifying by identifying which contributions from the ligand and the protein oppose binding and which favor it. When a modification produces an unexpected loss in affinity, the decomposition offers a route to understanding why and where. Consider the addition of a bulky substituent to a lead. Mapping the van der Waals and electrostatic components of the energy onto the new group and onto the surrounding residues distinguishes the case in which the group is in steric conflict with the protein from the case in which the binding cavity restructures to accommodate the larger ligand, paying its cost elsewhere in the protein or in its configurational entropy. The same analysis separates the parts of a ligand that are essential to binding, and should therefore be preserved, from those that contribute little and are the natural candidates for modification.

The atomistic mapping also provides a means of investigating received wisdom about binding. It is generally believed, for example, that strain is induced in both the protein and the ligand upon binding [24, 25, 26, 27]. The energetic and entropic contributions to this strain can be examined separately and mapped, identifying which parts of the ligand and which residues of the binding site bear this strain. This, in turn, could guide drug discovery toward modifications that alleviate ligand strain and induced protein strain, a capability available only through an atomic or chemical group mapping described here.

In summary, we have presented the theory, computational methods, and software for decomposing the energy and entropy of a solute and mapping them onto atoms and chemical groups. The approach complements alchemical free energy methods, which estimate the total binding free energy accurately but do not say where it comes from, trading some accuracy in the total, since per-term errors accumulate, for an interpretable account of its origin. In doing so it supplies for the protein and ligand the kind of spatially resolved thermodynamic information that solvation mapping has long provided for water.

# Author disclosures

MKG has an equity interest in and is a cofounder and scientific advisor of VeraChem LLC. He is also on the scientific advisory boards of Denovicon Therapeutics, In Cerebro, Cold

Start Therapeutics, and Beren Therapeutics.

TK is founder and scientific advisor of Deep Waters NYC, LLC; has an equity interest in Ventus Therapeutics; has an equity interest in and is a director and scientific advisor for InCerebro Inc.; and is an author on U.S. patents 7,756,674; 7,970,580, 7,970,581.

The remaining authors declare no competing interests. The affiliations and financial interests listed above had no role in the conception, design, execution, analysis, or reporting of this work, and the authors affirm that they in no way influenced the scientific content or the conclusions presented here.

# References


[1] Robert W. Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

[2] John G. Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, 3(6):300–313, 1935.

[3] Themis Lazaridis. Inhomogeneous Fluid Approach to Solvation Thermodynamics. 1. Theory. *The Journal of Physical Chemistry B*, 102(18):3531–3541, April 1998.

[4] Themis Lazaridis. Inhomogeneous fluid approach to solvation thermodynamics. 2. applications to simple fluids. *The Journal of Physical Chemistry B*, 102(18):3542–3550, 1998.

[5] Thomas Young, Robert Abel, Byungchan Kim, Bruce J Berne, and Richard A Friesner. Motifs for molecular recognition exploiting hydrophobic enclosure in protein-ligand binding. *Proceedings of the National Academy of Sciences*, 104(3):808–813, 2007.

[6] Crystal N. Nguyen, Tom Kurtzman Young, and Michael K. Gilson. Grid inhomogeneous solvation theory: Hydration structure and thermodynamics of the miniature receptor cucurbit[7]uril. *The Journal of Chemical Physics*, 137(4):044101, 07 2012.

[7] Michael K. Gilson, Joseph Cruz, Kunhuan Liu, Samrat Lohar, Daniel R. Roe, and Tom Kurtzman. A generalized theory for the structural and spatial mapping of energy, entropy, and free energy. *ChemRxiv*, 2026(0630), 2026.

[8] D. A. Case, J. T. Berryman, R. M. Betz, D. S. Cerutti, III T. E. Cheatham, et al. *AmberTools 2025*. University of California, San Francisco, San Francisco, CA, 2025.

[9] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An $n\log(n)$ method for ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, 1993.

[10] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, 1987.

[11] Lieyang Chen, Anthony Cruz, Daniel R. Roe, Andrew C. Simmonett, Lauren Wickstrom, Nanjie Deng, and Tom Kurtzman. Thermodynamic decomposition of solvation free energies with particle mesh ewald and long-range lennard-jones interactions in grid inhomogeneous solvation theory. *Journal of Chemical Theory and Computation*, 17(5):2714–2724, 2021. PMID: 33830762.

[12] Peter J. Steinbach and Bernard R. Brooks. New spherical-cutoff methods for long-range forces in macromolecular simulation. *Journal of Computational Chemistry*, 15:667–683, 1994.

[13] Helmut Carter, Joseph Cruz, Steven Ramsey, and Tom Kurtzman. Flexi-gist: Solvation thermodynamic decomposition and mapping for flexible molecules. 2026.

[14] Hiroyuki Matsuda. Physical nature of higher-order mutual information: Intrinsic correlations and frustration. *Phys. Rev. E*, 62:3096–3102, Sep 2000.

[15] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23(3-4):301–321, 2003.

[16] Vladimir Hnizdo, Eva Darian, Adam Fedorowicz, Eugene Demchuk, Shengqiao Li, and Harshinder Singh. Nearest-neighbor nonparametric method for estimating the configurational entropy of complex molecules. *Journal of Computational Chemistry*, 28(3):655–668, 2007.

[17] Benjamin J Killian and Joslyn Yundenfreund Kravitz. Extraction of configurational entropy from molecular simulations via an expansion approximation. *The Journal of Chemical Physics*, 127(2):024107, 2007.

[18] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. Mdanalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, 32(10):2319–2327, 2011.

[19] Richard J. Gowers, Max Linke, Jonathan Barnoud, Tyler J. E. Reddy, Manuel N. Melo, Sean L. Seyler, Jan Domanski, David L. Dotson, Sebastien Buchoux, Ian M. Kenney, and Oliver Beckstein. Mdanalysis: A python package for the rapid analysis of molecular dynamics simulations. In Tharindu Brahma, Dixie Richards, and Louise Gruber, editors, *Proceedings of the 18th Python in Science Conference*, pages 98–105, 2019.

[20] Charles R. Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[21] Yulin Huang and Robert C. Rizzo. A water-based mechanism of specificity and resistance for lapatinib with erbb family kinases. *Biochemistry*, 51(12):2390–2406, 2012.

[22] David J. Huggins, Woody Sherman, and Bruce Tidor. Rational approaches to improving selectivity in drug design. *Journal of Medicinal Chemistry*, 55(4):1424–1444, 2012.

[23] Daniel D. Robinson, Thomas Bertrand, Jean-Christophe Carry, Frank Halley, Andreas Karlsson, Magali Mathieu, Hervé Minoux, Marc-Antoine Perrin, Benoît Robert, Laurent Schio, and Woody Sherman. Differential water thermodynamics determine PI3K-beta/delta selectivity for solvent-exposed ligand modifications. *Journal of Chemical Information and Modeling*, 56(5):886–894, 2016.

[24] Emanuele Perola and Paul S. Charifson. Conformational analysis of drug-like molecules bound to proteins: an extensive study of ligand reorganization upon binding. *Journal of Medicinal Chemistry*, 47(10):2499–2510, 2004.

[25] John Liebeschuetz, Jana Hennemann, Tjelvar Olsson, and Colin R. Groom. The good, the bad and the twisted: A survey of ligand geometry in protein crystal structures. *Journal of Computer-Aided Molecular Design*, 26(2):169–183, 2012.

[26] Ajay N. Jain, Alexander C. Brueckner, Ann E. Cleves, Mikhail Reibarkh, and Edward C. Sherer. A distributional model of bound ligand conformational strain: From small molecules up to large peptidic macrocycles. *Journal of Medicinal Chemistry*, 66(3):1955–1971, 2023.

[27] Shuo Gu, Matthew S. Smith, Ying Yang, John J. Irwin, and Brian K. Shoichet. Ligand strain energy in large library docking. *Journal of Chemical Information and Modeling*, 61(9):4331–4341, 2021.