

## Gradient of a weighted RMSD harmonic energy, in arbitrary coordinate space

The idea is that we have a score that depends on the RMSD between some reference and mobile structures,  $R$  and  $M$ , respectively, with the mobile one aligned onto the reference one. It is easy to calculate the gradient of RMSD (and hence the score) with respect to Cartesian coordinates of the mobile structure  $\vec{x}$ . But what if we are interested in the gradient with respect to an alternative coordinate system  $\vec{c}$ ? Obviously, the number of coordinates in the two sets has to be the same and the phase spaces described by these have to be equivalent (so there has to be some one-to-one mapping between the two coordinate sets). Let the score be  $E = w \cdot r^2$ , where  $w$  is some constant weight factor and  $r$  is the best-fit RMSD. We then have:

$$\frac{\partial E}{\partial c_i} = 2wr \frac{\partial r}{\partial c_i} = 2wr \sum_j \frac{\partial r}{\partial x_j} \frac{\partial x_j}{\partial c_i}$$

where  $c_i$  is some coordinate from the coordinate system  $\vec{c}$  and the summation is over coordinates  $x_j$  of the coordinate system  $\vec{x}$ . The last step is simply application of the multi-variable chain rule.

Next, let's consider the case where the weighting itself is RMSD-dependent (adaptive weighting). In particular, let's say there are  $n$  mobile structures,  $M_1$  through  $M_n$ , for the one reference structure  $R$ , and that the total score is the sum of scores from each mobile-to-reference alignment, Boltzmann-weighted by the corresponding RMSD. That is:

$$E = \sum_{k=1}^n w_k v_k E_k = \sum_{k=1}^n w_k \frac{e^{-\beta r_k^2}}{Z} r^2 = \sum_{k=1}^n w_k \frac{e^{-\beta r_k^2}}{\sum_{j=1}^n e^{-\beta r_j^2}} r^2$$

where  $\beta$  is some temperature factor and  $w_k$  is some constant weight pre-factor associated with the  $k$ -th mobile structure,  $v_k$  is the variable (adaptive) Boltzmann weight, and  $Z$  is the associated partition function. Then, the gradient of the total score is just the sum of gradients of the parts of the score, so let us just compute the gradient of each  $E_k$ :

$$\begin{aligned} \frac{\partial E_k}{\partial c_i} &= \frac{w_k}{Z^2} \left[ \left( e^{-\beta r_k^2} \cdot (-2\beta r_k) \cdot r_k^2 \frac{\partial r_k}{\partial c_i} + e^{-\beta r_k^2} \cdot 2r_k \frac{\partial r_k}{\partial c_i} \right) \cdot Z - e^{-\beta r_k^2} \cdot r_k^2 \frac{\partial Z}{\partial c_i} \right] \\ &= w_k \frac{e^{-\beta r_k^2}}{Z^2} r_k \left[ \left( 2 \frac{\partial r_k}{\partial c_i} - 2\beta r_k^2 \frac{\partial r_k}{\partial c_i} \right) \cdot Z - r_k \frac{\partial Z}{\partial c_i} \right] = w_k \frac{e^{-\beta r_k^2}}{Z} r_k \left[ 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial c_i} - r_k \frac{1}{Z} \frac{\partial Z}{\partial c_i} \right] \\ &= w_k v_k r_k \left[ 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial c_i} - r_k \frac{1}{Z} \frac{\partial Z}{\partial c_i} \right] = w_k v_k r_k \left[ 2(1 - \beta r_k^2) \sum_{j=1}^N \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - r_k \frac{1}{Z} \sum_{j=1}^N \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} \right] \end{aligned}$$

where the last step again is just the application of the multi-variable chain rule. Then, the gradient of the total score is then:

$$\frac{\partial E}{\partial c_i} = \sum_{k=1}^n \frac{\partial E_k}{\partial c_i} = \sum_{k=1}^n w_k v_k r_k \left[ 2(1 - \beta r_k^2) \sum_{j=1}^N \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - r_k \frac{1}{Z} \sum_{j=1}^N \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} \right]$$

To make this more convenient to calculate, we can further rearrange a little:

$$\begin{aligned} \frac{\partial E}{\partial c_i} &= \sum_{k=1}^n w_k v_k r_k \left[ 2(1 - \beta r_k^2) \sum_{j=1}^N \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - r_k \frac{1}{Z} \sum_{j=1}^N \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} \right] = \sum_{k=1}^n w_k v_k r_k 2(1 - \beta r_k^2) \sum_{j=1}^N \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - \\ &\sum_{k=1}^n w_k v_k r_k r_k \frac{1}{Z} \sum_{j=1}^N \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} = \sum_{k=1}^n \sum_{j=1}^N w_k v_k r_k 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - \sum_{k=1}^n \sum_{j=1}^N w_k v_k r_k r_k \frac{1}{Z} \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} = \\ &\sum_{j=1}^N \sum_{k=1}^n w_k v_k r_k 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - \sum_{j=1}^N \sum_{k=1}^n w_k v_k r_k r_k \frac{1}{Z} \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} = \sum_{j=1}^N \sum_{k=1}^n w_k v_k r_k 2(1 - \end{aligned}$$

$$\beta r_k^2) \frac{\partial r_k}{\partial x_j} \frac{\partial x_j}{\partial c_i} - w_k v_k r_k r_k \frac{1}{Z} \frac{\partial Z}{\partial x_j} \frac{\partial x_j}{\partial c_i} = \sum_{j=1}^N \left( \sum_{k=1}^n w_k v_k r_k 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial x_j} - w_k v_k r_k r_k \frac{1}{Z} \frac{\partial Z}{\partial x_j} \right) \frac{\partial x_j}{\partial c_i} =$$

$$\sum_{j=1}^N \left( \sum_{k=1}^n w_k v_k r_k \left( 2(1 - \beta r_k^2) \frac{\partial r_k}{\partial x_j} - r_k \frac{1}{Z} \frac{\partial Z}{\partial x_j} \right) \right) \frac{\partial x_j}{\partial c_i}$$

It only remains to calculate the derivative of  $Z$  with respect to Cartesian coordinates:

$$\frac{1}{Z} \frac{\partial Z}{\partial x_i} = \frac{1}{Z} \sum_{j=1}^n e^{-\beta r_j^2} \cdot (-2\beta r_j) \cdot \frac{\partial r_j}{\partial x_i} = -2 \sum_{j=1}^n v_j \beta r_j \frac{\partial r_j}{\partial x_i}$$

where we include the pre-factor  $1/Z$  for convenience. Computationally, one would need to first compute the partition function  $Z$  (in the process storing the numerators of Boltzmann weights), then compute the weights (in the process computing the gradient of  $Z$ ), and then finally compute the gradient of each component of the score. Specifically, pseudo-code for this would be ( $N$  is the dimensionality of the coordinate space  $\vec{x}$  or  $\vec{c}$ ; for simplicity, we will omit the constant weights  $v$ ):

```

Z = 0                                % partition function Z
w = zeros(1, n)                      % weights of each mobile structure
gradZ = zeros(1, N)                  % gradient of the Z (in  $\vec{c}$ ), divided by Z
r = zeros(1, n)                      % best-fit RMSD of each mobile structure
gradRMSD = zeros(n, N)               % gradients of the above RMSDs, in  $\vec{c}$ 
grad = zeros(1, N)                   % gradient of the total score, in  $\vec{c}$ 

% compute RMSDs, RMSD gradients, weight numerators, and Z
for i = 1:n
    (r[i], grad) = compute_RMSD_and_gradient(Mi, R)
    gradRMSD(:, i) = grad
    w[i] = exp(-beta*r[i]*r[i]) % beta is the temperature factor parameter
    Z += w[i]
end

% compute weights and the gradient of Z
for j = 1:n
    w[j] = w[j]/Z
    for i = 1:N
        gradZ[i] -= 2*w[j]*beta*r[j]* gradRMSD (j, i)
    end
end

% compute the gradient of the total score
for k = 1:n
    for j = 1:N
        for each alternative coordinate ci that depends on x[j]:
            grad[ci] += (w[k]*r[k]*(2*(1 - beta*r[k]*r[k])*gradRMSD[k, j] -
                                r[k]*gradZ[j]))*
                        (partial derivative of x[j] with respect to ci)
        end
    end
end

```

If there is not one, but a set of structures  $R$ , each with its own corresponding group of mobile structures (as is often the case in Fuser), then the above pseudo-code simply needs to be applied to each group separately and the contribution to the gradient of the total score (which will now be a sum over groups) simply added.