

Upgrading DiscoTope version 1.1 to 2.0.

Reference version 1.0: <http://www.ncbi.nlm.nih.gov/pubmed/17001032>

Reference version 2.0: <http://www.ncbi.nlm.nih.gov/pubmed/23300419>

Concerning the Algorithm:

The 2.0 version uses modified set of parameters, a novel weighted proximity scoring function and upper half-sphere neighbor to describe the neighborhood and surface exposure respectively of a given residue in the protein 3-dimensional structure. The 1.1 version uses a non-weighted proximity scoring function and full-sphere neighbor count to account for neighborhood epitope likelihood and surface exposure. The table below summarizes the differences:

	1.1	2.0
Proximity scoring function	$PS(r, w, k_{ps}) = \sum_i ls(r_i, w)$	$PS(r, w, k_{ps}) = \sum_i \beta_i \cdot ls(r_i, w)$ $\beta_i = 0.8 \cdot (1 - (d_i / k_{ps})) + 0.2$
Surface exposure measure	Neighbor count $SS(r)$ (alpha-atoms) in a full sphere having radius = 10Å	Neighbor count $SS(r)$ (alpha-atoms) in the upper half sphere having radius = 14Å
DiscoTope score	$DS(r) = -0.5 \cdot SS(r) + PS(r)$	$DS(r) = -\alpha \cdot SS(r) + (1 - \alpha) \cdot PS(r)$
Parameters	$w = 9$ $k_{ps} = 10\text{Å}$	$w = 1$ $k_{ps} = 22\text{Å}$ $\alpha = 0.115$

where r is the query residue for which the log-odds ratio score (PS) is computed, r_i is any residue within k_{ps} distance from r , $ls(r_i, w)$ is the log-odds ratio value of r_i , sequentially averaged over a window of w residues and d_i is the distance between r and r_i .

Concerning the implementation

Both versions use the BioPython PDB module by Thomas Hamelryck to read and extract data from PDB structures. As the 2.0 version uses Upper Half-sphere as a surface exposure measure and the distance to the nearest residues within 22Å contrary 10Å in version 1.1 the running time has increased. To get the upper half-sphere neighbor count and surrounding neighbors in the 22Å sphere version 2.0 utilizes a modified version of the HSExposureListCL.py package kindly provided by Thomas Hamelryck. Contrary the packages used for neighbor count in the 1.1, which has lots of its code implemented in C, this package is purely implemented in python, hence has slower running time.