LatFoldVec - Manual

Martin Mann - University Freiburg

http://www.bioinf.uni-freiburg.de/

LatPack Tools Package
Version 1.9.1

1 Description

LatFoldVec models the co-translational (or vectorial) folding. This is done via a folding simulation of a periodically growing amino acid chain. Therefore, it implements a combination of a Monte-Carlo simulation utilizing a Metropolis criterion (see [4]) and a chain growth algorithm. The implementation is based on the Energy Landscape Library [5]. It supports:

1. various lattices (see Sec. 3)
2. arbitrary energy functions (see Sec. 4)
3. ...

2 Method

2.1 Co-translational Folding Simulation

Given:

$S = S_1, \ldots, S_n$	:	monomer sequence from alphabet A to fold
$E(S, P)$	:	energy function (see Sec. 4)
$M(L)$	:	the set of neighboring structures of L in the energy landscape
Δt	:	the folding time between single chain elongations
N	:	the neighboring vectors of the lattice model to use

Result:

$L = L_1, \ldots, L_n$	:	3D coordinates of the energetically best sequential placement of S in the lattice

Method:

The simulations follows a structure-elongating chain-growth approach in combination with a "global" folding simulation:

1: $L = L_1 = (0, 0, 0)$ ▷ initialized by placing the first monomer to $(0, 0, 0)$
2: **for** $i = 2 \dots n$ **do**

3: **for** $t = 0 \dots \Delta t$ **do** ▷ folding simulation of chain with length i
4: Select random neighbor $m_r \in M(L)$
5: $r \in [0, 1]$ ▷ get random number in interval $[0, 1]$
6: **if** $(r \leq e^{-\frac{E(S,m_r)-E(S,L)}{kT}})$ **then**
7: $L \leftarrow m_r$ ▷ go to neighboring structure
8: **else** ▷ keep current structure for this step
9: **end if**
10: **end for**

11: $E(L) = \emptyset$ ▷ will contain all elongations of L
12: **for all** $\vec{v} \in N$ **do**
13: **if** $L_{(i-1)} + \vec{v} \notin \{L_1, \dots, L_{(i-1)}\}$ **then** ▷ selfavoidingness
14: $L' \leftarrow (L_1, \dots, L_{(i-1)}, (L_{(i-1)} + \vec{v}))$
15: $E(L) \leftarrow E(L) \cup \{L'\}$ ▷ store extension
16: **end if**
17: **end for**
18: $Z(E(L)) = \sum_{e \in E(L)} e^{\frac{-E(S,e)}{kT}}$ ▷ partition function of E(L)

19: select random elongation $e_r \in E(L)$ according to its probability in the ensemble of all elongations $P(e) = \frac{e^{\frac{-E(S,e)}{kT}}}{Z(E(L))}$

20: $L = e$ ▷ set choosen elongation
21: **end for**
22: report L with energy $E(S, L)$

Note: Since we are simulating the folding of a *growing* amino acid chain, we have to ensure that the ends of the intermediate chains (namely monomer L_i) has to have a free neighbored position that can be used for elongation. Therefore, only a reduced set of neighbored structures $M'(L) \subseteq M(L)$ is used in line 4.

3 Available Lattices

Several lattice models can be used to fold a structure.

The currently supported lattice models and the corresponding neighboring vectors are:

ID	Name	Neighborhood vectors	#
SQR	Square	$\{\pm(1,0,0), \pm(0,1,0)\}$	4
CUB	Cubic	$\{\pm(1,0,0), \pm(0,1,0), \pm(0,0,1)\}$	6
FCC	Face Centered Cubic	$\left\{ \begin{smallmatrix} \pm(1,1,0), \pm(1,0,1), \pm(0,1,1), \\ \pm(1,-1,0), \pm(1,0,-1), \pm(0,1,-1) \end{smallmatrix} \right\}$	12

4 Energy Functions

LatFoldVec supports arbitrary energy functions that are based either on contacts or on distance intervals. The specification of an energy function has to be given in text format and defines the allowed sequence alphabet as well.

In general, the energy of a sequence S of length n with structure coordinates P is determined by

$$E(S,P) = \sum_{1 \le i+1 < j \le n} e(S_i, S_j, P_i, P_j). \tag{1}$$

Here, $e(S_i, S_j, P_i, P_j)$ is a placeholder for the specific evaluation function that is given for the different types in the following.

4.1 Contact Based Energy Function

A contact based energy function for an alphabet A is defined by an energy table $E^c : |A| \times |A| \to \mathcal{R}$ such that

$$e_c(S_i, S_j, P_i, P_j) = \begin{cases} E^c[S_i, S_j] & \text{if } P_i \text{ and } P_j \text{ are neighbored} \\ 0 & \text{else} \end{cases} \tag{2}$$

For example, a function like this was used by Lau and Dill to define the widely used HP-model [1].

Text File Encoding

The LatFoldVec text file enconding of a contact based energy function consists of two parts: the alphabet elements and the energy table. A consecutive string of the alphabet elements in the first line determines the allowed protein sequence characters (the alphabet) and the dimensions of the energy table that is read from the remaining file.

An example energy file for the HPNX-model is:

```
HPNX
-4.0  0.0  0.0  0.0
 0.0 +1.0 -1.0  0.0
 0.0 -1.0 +1.0  0.0
 0.0  0.0  0.0  0.0
```

4.2 Distance Interval Based Energy Function

A distance insterval based energy function for an alphabet A is defined by a consecutive set of k distance intervals with the upper bounds $d^{up}_{1\ldots k}$ and an energy table $E^i_{1\ldots k} : |A| \times |A| \to \mathcal{R}$ for each of them. Given the distance to interval index function idx we define the evaluation function

$$e_i(S_i, S_j, P_i, P_j) = E^i_{idx(P_i,P_j)}[S_i, S_j] \quad (3)$$

$$idx(P_i, P_j) = \arg\min_k \left(|P_i - P_j| \leq d^{up}_k\right) \quad (4)$$

Text File Encoding

The LatFoldVec text file enconding of a distance interval based energy function consists of three parts: the alphabet elements, the upper bounds of the intervals and the energy tables for the interval. A consecutive string of the alphabet elements in the first line determines the allowed protein sequence characters (the alphabet) and the dimensions of the energy tables. The second line contains a whitespace separated list of the upper interval bounds. Their number sets the number of energy tables read from the remaining file. The interval bounds are expected to be given in Ångstroems. For a correct scaling of the bounds it is necessary to give the average distance of two consecutive C_α-atoms in the underlying model to LatFoldVec (see input parameters in Sec. 5).

An example energy file that encodes the HPNX-model using a distance interval based energy function is:

```
HPNX
3.7 3.9 999999

 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0

-4.0  0.0  0.0  0.0
 0.0 +1.0 -1.0  0.0
 0.0 -1.0 +1.0  0.0
 0.0  0.0  0.0  0.0

 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0
```

The number 999999 is used as a placeholder for $+\infty$, i.e. the upper bound of the last distance interval. It is important to know that the average C_α-distance in the underlying model was 3.8 Å. Therefore, the resulting interval energy function corresponds to the contact based energy function of the previous section; only distances close to the average C_α-distance are taken into account.

5 Program Parameters

Input

-seq The sequence to fold globally. It has to be conform to the alphabet given by the energy file (see **-energyFile**).

-energyFile A file that encodes the used alphabet and the specific energy function (see Sec. 4 for format details).

-energyForDist If present, the input of **-energyFile** will be interpreted as distance interval energy function. Otherwise a contact based energy functino is expected.

-energyCalphaDist Specifies the average distance of two consecutive C_α-atoms in the underlying model. This value is needed to scale the intervals of a distance interval based energy function onto the C_α-distances of the lattice model in use.

Simulation Settings

-kT the relative temperature to calculate the Boltzmann weight for a given structure within the formula $\exp(-E/kT)$. Thus, the parameter kT sets the product of temperature (T) and the energy function specific scaling constant (k).

-maxSteps each intermediate subchain simulation ends after this number of simulation steps before proceeding to next elongation

-minE simulation ends if energy gets below or equal the given value

-seed seed for random number generator (uses a system independent linear congruent generator)

-runs number of independent folding simulations to perform

Lattice Settings

-lat The lattice model to use for the sequential folding. The available list of lattice identifiers is given in Sec. 3.

-moveSet Sets the move set to use for the present lattice:

PullM Pull-moves definied by Lesh et al. [2]

Output

-out The output mode along the folding simulation:

N no additional output is done (default)

E the energy of the structure of each simulation step is printed

S structure and energy of each simulation step is given

-outFile Specifies where to write the output of simulations to. If equal to 'STDOUT' it is written to standard output, otherwise the given string is assumed to be the filename to write to.

-outTiming If present, the used cpu-time is printed.

Miscellaneous

-v Give verbose output during computation.

-vv Give extra verbose output during computation.

-help Prints the available program parameters.

6 Contact

Martin Mann
Bioinformatics Group
University Freiburg, Germany

http://www.bioinf.uni-freiburg.de/

References

[1] Kit Fun Lau and Ken A. Dill: **A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins**, *Macromolecules* 1989, **22**(10):3986–3997

[2] Lesh, N., Mitzenmacher, M., and Whitesides, S.: **A complete and effective move set for simplified protein folding**, In *Proceedings of the seventh annual international conference on Research in computational molecular biology (RECOMB'03)* 2003, 188–195.

[3] Madras, N. and Sokal, A. D.: **The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk**, *Journal of Statistical Physics* 1988, **50**, 109–186.

[4] Mann, M., Maticzka, D., Saunders, R., and Backofen, R.: **Classifying protein-like sequences in arbitrary lattice protein models using LatPack**, *HFSP Journal* 2008, **2**(6), 396. Special issue on protein folding: experimental and theoretical approaches

[5] Mann, M., Will, S., and Backofen, R.: **The Energy Landscape Library - A Platform for Generic Algorithms**, In *Proceedings of the 1st international Conference on Bioinformatics Research and Development (BIRD'07)* 2007, OCG **217**, 83-86.