The Energy Landscape Library
A Platform for Generic Algorithms

Martin Mann*, Sebastian Will**, and Rolf Backofen

Bioinformatics, University Freiburg, Germany

Abstract. The study of energy landscapes of biopolymers and their models is an important field in bioinformatics [1–6]. For instance the investigation of kinetics or folding simulations are done using methods that are based on sampling or exhaustive enumeration [7–11]. Most of such algorithms are independent of the underlying landscape model. Therefore frameworks for generic algorithms to investigate the landscape properties is needed. Here, we present the Energy Landscape Library (ELL) that allows such a model-independent formulation of generic algorithms dealing with discrete states. The ELL is a completely object-oriented C++ library that is highly modular, easy to extend, and freely available online. It can be used for a fast and easy implementation of new generic algorithms (possibly based on the provided basic method pool) or as a framework to test their properties for different landscape models, which can be formulated straightforward.

1 Introduction

The study of fitness or energy landscapes of biopolymers has gained more and more interest in the field of bioinformatics [1, 2, 6, 10–12]. The kinetics, number and distribution of optima and several other properties of such landscapes are studied and new algorithms have been developed. Further folding pathways and other evolutionary concepts are investigated (e.g. by Monte Carlo approaches or extensions). Most of the methods are very generic and can be applied to any arbitrary landscape model [7, 8]. Therefore a platform for abstract algorithm formulations with a multitude of different landscape models is needed.

Our Energy Landscape Library (ELL) is going to provide such a platform. The core of the library is an abstract class `State` that defines the interface between the underlying landscape model (subclasses of `State`) and the generic algorithms that can be applied. A visualisation of this concept is given in Fig. 1.

Due to the simple interface, a new landscape model can be easily introduced by implementing a corresponding state derived from the abstract superclass. This enables a fast and efficient way to apply the provided methods to the energy landscape in focus. Futhermore, the strict partition of algorithm layer and states allows that a new algorithm, formulated on the abstract `State`, can be tested on

* Supported by the EU project EMBIO (EC contract number 012835)

** Supported by the EU Network of Excellence REWERSE (reference number 506779)

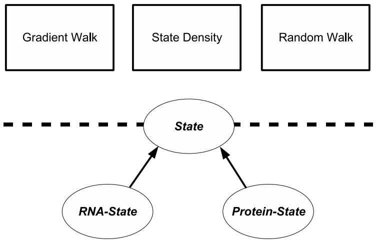

Fig. 1. ELL layer concept: partition of application (rectangle) and state model (elipse).

all landscape types without adaption or reimplementation. Therefore the ELL provides a capable platform for generic algorithms to study different energy landscapes or properties of the methods themselves.

2 The Energy Landscape Library

The ELL is available as C++ library. The implementation and design is completely object-oriented to provide clear interfaces and to be highly modular. By that it is easy to extend and to use.

2.1 The Core - an Abstract State

As mentioned before, the core of the ELL is an abstract `State` class. A `State` provides only its fitness and the possibility to iterate over its neighbors in the landscape in different ways (e.g. ordered or randomly). This is sufficient to implement a number of algorithms that are used to investigate the landscape. The term fitness instead of energy is used because a fitness function can base on more than the energy, e.g. for evolutionary studies that are supported by the ELL too.

The neighbor generation is done via iterators. For a flexible usage and to minimize the memory consumption for the neighborhood enumeration, the neighbors are generated on demand. Additionally, a `State` can be transformed into a compressed form to minimize the memory usage of algorithms that have to collect and manage a huge number of states.

2.2 The Landscape Models

The ELL currently contains completely implemented states for RNA secondary structures [10, 12] and for structures of simple lattice protein models. To determine the free energy of an RNA secondary structure the Vienna RNA package is used [13]. Neighbors only differ in one bond. The simple lattice protein models, like the widespread HP-model [14], are supported for different lattices (square, cubic, face centered cubic). The monomer alphabet and the associated contact energy function are free assignable. To generate the neighborhood of a structure pivot moves are used. They have been proven to be ergodic [15] and allow therefore the transformation of each structure into all others of the same length. Local moves will be available soon.

All available models can be extended in a very simple and clear way (e.g. with new neighbor generators, other fitness functions or new lattices). Also the implementation of new landscape models by their associated `State` subclasses is straight-forward in the ELL and enables a fast incorporation into the library.

2.3 Generic Algorithms

As introduced before, all algorithms are based on the abstract `State` class. Therefore they can be used with all derived subclasses without adaption or reimplementation which yields a very generic pool of methods. These methods can be applied directly to study properties of the landscape model or can be used as modules to build new algorithms. By that the implementation of new generic algorithms becomes faster and the user can focus on the new features instead working on common and often used submethods. An example implementation of a gradient walk using the ELL is given in Fig. 2.

```
// returns a locale optimum (no better neighbor) reached by a gradient walk
State* gradientWalk(const State* const source) {
  State* currState = source.clone(), bestNeighbor = getBestNeighbor(currState);
  while ( currState->getFitness() < bestNeighbor->getFitness()) {
    delete currState;
    currState = bestNeighbor;
    bestNeighbor = getBestNeighbor(currState);
  }
  delete bestNeighbor;
  return currState;  // == local optimum (no neighbor with better fitness)
}

// returns the best state neighbored to 'center'
State* getBestNeighbor(State* center) {
  NeigborListPtr neighbors = center->getNeighborList();
  NeighborList::Iterator itN = neighbors.begin();
  State* best = itN->clone();
  for (++itN; itN != neighbors->end(); ++itN) {
    if (best->getFitness() < itN->getFitness()) {  // found better neighbor
      delete best;
      best = itN->clone();
    }
  }
  return best;
}
```

Fig. 2. Example source code for a gradient walk using the ELL.

Several tool implementations for enumeration and sampling are already available in the ELL. For instance a widely used method to estimate the state density [7], a barrier tree generator [8] or a newly developed barrier tree sampler for landscapes that are too large for exhaustive enumeration.

3 Conclusion

We introduced the ELL, a C++ programming platform for the study of energy landscapes with generic methods. The partition of the algorithms and the underlying landscape model, interfaced by a state abstraction, allows an independent development of both layers. The ELL supports the user with a pool of common methods and some landscape models (e.g. RNA) to provide a fast and efficient base for further investigations. It is highly modular, object-oriented and easy to extend. The ELL is free available under

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

References

1. Reidys, C.M., Stadler, P.F.: Neutrality in fitness landscapes. Appl. Math. and Comput. **117** (2001) 321–350
2. Stadler, P.F.: Fitness landscapes. Biological Evolution and Statistical Physics (2002) 187–207
3. Stadler, P.F., Stephens, C.R.: Landscapes and effective fitness. Comments in Theoretical Biology **8** (2003) 389–431
4. Apaydin, M.S., Brutlag, D.L., Guestrin, C., Hsu, D., Latombe, J.C., Varma, C.: Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. Journal of Computational Biology **10**(3-4) (2003) 257–81
5. Blackburne, B.P., Hirst, J.D.: Three-dimensional functional model proteins: Structure function and evolution. Journal of Chemical Physics **119** (2003) 3453–3460
6. Xia, Y., Levitt, M.: Simulating protein evolution in sequence and structure space. Current Opinion in Structural Biology **14**(2) (2004) 202–7
7. Wang, F., Landau, D.P.: Determining the density of states for classical statistical models: a random walk algorithm to produce a flat histogram. Phys Rev E Stat Nonlin Soft Matter Phys **64** (2001)
8. Flamm, C., Hofacker, I.L., Stadler, P.F., Wolfinger, M.T.: Barrier trees of degenerate landscapes. Z.Phys.Chem **216** (2002) 155–173
9. Wille, L.T., Hansmann, U.H.E.: Global optimization by energy landscape paving. Phys. Rev. Lett. (2002)
10. Wolfinger, M.T., Svrcek-Seiler, W.A., Flamm, C., Hofacker, I.L., Stadler, P.F.: Exact folding dynamics of rna secondary structures. J.Phys.A: Math.Gen. **37** (2004) 4731–4741
11. Wolfinger, M.T., Will, S., Hofacker, I.L., Backofen, R., Stadler, P.F.: Exploring the lower part of discrete polymer model energy landscapes. Europhysics Letters **74**(4) (2006) 725–732
12. Chen, S.J., Dill, K.A.: Rna folding energy landscapes. Proc. Natl. Acad. Sci. USA **97**(2) (2000) 646–51
13. Hofacker, I.L., et al.: Vienna rna package http://www.tbi.univie.ac.at/RNA/.
14. Lau, K.F., Dill, K.A.: A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules **22** (1989) 3986–3997
15. Madras, N., Slade, G.: The Self-Avoiding Walk. Probability and Its Applications. Birkhäuser Boston (1996)