BIOINFORMATICS APPLICATIONS NOTE
Vol. 26 no. 19 2010, pages 2466–2467
doi:10.1093/bioinformatics/btq445

Structural bioinformatics
Advance Access publication August 4, 2010

SPRINT: side-chain prediction inference toolbox for multistate protein design

Menachem Fromer[1,*], Chen Yanover[2], Amir Harel[1], Ori Shachar[1], Yair Weiss[1] and Michal Linial[3]

[1]School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel, [2]Program in Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, WA, USA and [3]Department of Biological Chemistry, Institute of Life Sciences, The Hebrew University of Jerusalem, Israel

Associate Editor: Burkhard Rost

ABSTRACT

Summary: SPRINT is a software package that performs computational multistate protein design using state-of-the-art inference on probabilistic graphical models. The input to SPRINT is a list of protein structures, the rotamers modeled for each structure and the pre-calculated rotamer energies. Probabilistic inference is performed using the belief propagation or A* algorithms, and dead-end elimination can be applied as pre-processing. The output can either be a list of amino acid sequences simultaneously compatible with these structures, or probabilistic amino acid profiles compatible with the structures. In addition, higher order (e.g. pairwise) amino acid probabilities can also be predicted. Finally, SPRINT also has a module for protein side-chain prediction and single-state design.

Availability: The full C++ source code for SPRINT can be freely downloaded from http://www.protonet.cs.huji.ac.il/sprint

Contact: fromer@cs.huji.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 11, 2010; revised on July 29, 2010; accepted on July 30, 2010

1 INTRODUCTION

The objective of engineering a protein to perform a particular biological function is called protein design. The design problem is usually restricted to the search for an amino acid sequence that assumes a target 3D structure (Kuhlman *et al.*, 2003), presuming that it will possess a corresponding function. This paradigm typically assumes a fixed protein backbone, and the amino acid side chain conformations to be considered are taken from a library of energetically favorable empirical observations termed rotamers (Dunbrack and Karplus, 1993). Lastly, pairwise atomic energy functions are used to assign pseudo-physical energetic values to pairs of rotamer atoms (Gordon *et al.*, 1999), and thus employed to score the compatibility of a particular sequence to the structure.

Recently, researchers have moved toward more realistic modeling of proteins in their cellular environment by generalizing the design concept to explicitly seek sequences that adopt *multiple* functional states, (Ambroggio and Kuhlman, 2006; Havranek and Harbury, 2003). We have achieved this goal by formulating protein design using probabilistic graphical models (Fromer *et al.*, 2010). Here, we present the software built to this end, SPRINT (side-chain prediction inference toolbox for multistate protein design). SPRINT accepts as input multiple protein structures, their respective rotamers and their pre-calculated energies. Furthermore, it is explicitly built to provide the user with a global view of the sequence space compatible with these multiple structures, since we have shown the significance of predicting both multiple low-energy sequences (Fromer and Shifman, 2009; Fromer and Yanover, 2009) and amino acid profiles (Fromer and Yanover, 2008).

2 METHODS

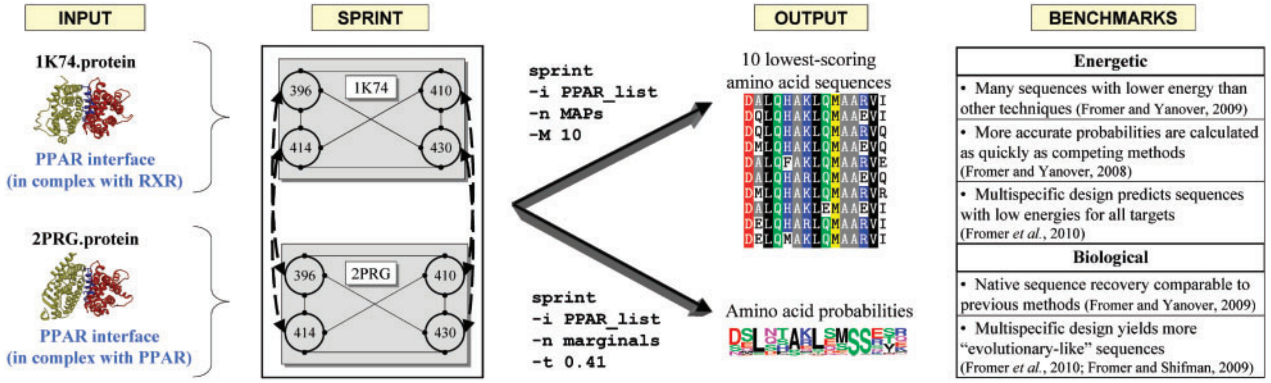
SPRINT is an open-source C++ software package that uses structural data to design functional protein sequences. The probabilistic inference core of SPRINT is based on the FastInf package (Jaimovich *et al.*, 2010). Protein structures are cast as probabilistic graphical models, and inference is performed to obtain amino acid sequences compatible with the multiple input structures (Fromer *et al.*, 2010). Multiple sequences can be predicted using the type-specific Best Max-Marginal First algorithm (Fromer and Yanover, 2009), and amino acid profiles can be calculated using the sum-product belief propagation (BP) algorithm (Fromer and Yanover, 2008). See Figure 1 for an overview of the SPRINT procedure.

3 RESULTS

SPRINT can be downloaded and installed on the user’s machine. In addition, due to their object-oriented nature, SPRINT modules can be extended to provide user-specific functionalities, such as atomic energy function calculations for input PDB structures.

Figure 1 demonstrates a run of the SPRINT package on the peroxisome proliferator-activated receptor (PPAR) interface design problem from Fromer *et al.* (2010). As input, rotamer–rotamer energies must be pre-computed (obtained here using the RosettaDesign package; Kuhlman *et al.*, 2003) for each protein structure and stored in the FastInf format (see http://www.protonet.cs.huji.ac.il/sprint/energies.php for details). The energy files, along with the relevant rotamer lists, are the input that SPRINT uses to construct a probabilistic graphical model. Since it has been shown that state-of-the-art atomic energy functions can still benefit from significant improvement (Potapov *et al.*, 2009), we did not choose a particular energy function to be included in the package but leave this choice to the user, thus separating the energetic aspects of the design process from the search process.

[*]To whom correspondence should be addressed.



```
sprint
-i PPAR_list
-n MAPs
-M 10
```

```
sprint
-i PPAR_list
-n marginals
-t 0.41
```

```
DALQHAKLQMAARVI
DQLQHAKLQMAAEVI
DQLQHAKLQMAARVQ
DMLQHAKLQMAAEVQ
DALQFAKLQMAARVE
DALQHARLQMAAEVQ
DMLQHAKLQMAARVR
DALQHAKLEMAAEVI
DELQHARLQMAARVI
DELQMAKLQMAARVI
```

Energetic
• Many sequences with lower energy than other techniques (Fromer and Yanover, 2009)
• More accurate probabilities are calculated as quickly as competing methods (Fromer and Yanover, 2008)
• Multispecific design predicts sequences with low energies for all targets (Fromer *et al.*, 2010)
Biological
• Native sequence recovery comparable to previous methods (Fromer and Yanover, 2009)
• Multispecific design yields more "evolutionary-like" sequences (Fromer *et al.*, 2010; Fromer and Shifman, 2009)

Fig. 1. The SPRINT input consists of a set of protein structures (e.g. 1K74.protein), their rotamer lists and pre-calculated interaction energies. These are converted into a probabilistic graphical model (illustrated here for four positions) and a probabilistic inference algorithm is run. The user can choose to output multiple low-energy amino acid sequences (top) or globally consistent amino acid probabilities (bottom). In this example, we demonstrate a run of SPRINT on the interface of PPAR, where the goal is to design PPAR interface sequences compatible with binding both the retinoic acid receptor (RXR, PDB code 1K74) and another PPAR monomer (PDB code 2PRG) (Fromer *et al.*, 2010). The SPRINT command-line options for running these examples are shown on the arrows, and the data files can be found at http://www.protonet.cs.huji.ac.il/sprint/#examples. The extensive benchmarking results for the algorithms implemented in the open-source SPRINT package are shown at far right; detailed lists can be found in Supplementary Tables S1–S4.

SPRINT builds a graphical model to represent all protein structures in the multistate design by connecting corresponding positions in the structures and requiring that they choose the same amino acid. In Figure 1, SPRINT will design the PPAR interface to be optimal for binding both RXR and another PPAR monomer. Probabilistic inference is performed using either the BP (Fromer *et al.*, 2010) or A* algorithms (Leach and Lemon, 1998), and type-dependent dead-end elimination can be applied as pre-processing for the prediction of low-energy sequences (Yanover *et al.*, 2007). We have previously shown that BP-based approaches outperform other methods in predicting sequences with low energies and computing more accurate sequence profiles (Fromer and Yanover, 2008, 2009); see Figure 1, Supplementary Tables S1–S4, and accompanying references for details.

The user can choose to predict either multiple low-energy sequences suitable for the input protein structures, or to predict amino acid probabilities for each position (or pair of positions). Figure 1 shows both options, with the 10 interface sequences most suited to bind both RXR and PPAR (top), and positional probabilities (bottom) calculated by approximating a statistical evaluation of *all* possible sequences weighted by the Boltzmann distribution of their side-chain conformational free energies (Fromer and Yanover, 2008). In Fromer *et al.* (2010), it was found that the sequences predicted to bind *both* targets better match evolutionary PPAR profiles than those optimized to bind only one target.

To this date, we have successfully used SPRINT for numerous single-state design problems (7 to 92 designed positions, rotamer space 10^{24} to 10^{200}, Fromer and Yanover, 2009) and hundreds of multistate design problems (20 positions, up to three states with combined rotamer space 10^{200}, Fromer and Shifman, 2009), where in all cases all amino acids were permitted at each design position. Lastly, we note that SPRINT also has a module for protein side-chain placement and single-state design; see web site for details.

ACKNOWLEDGEMENTS

We thank Ariel Jaimovich and Ofer Meshi for help with FastInf.

Funding: Sudarsky Center for Computational Biology (to M.F.).

Conflict of Interest: none declared.

REFERENCES

Ambroggio,X.I. and Kuhlman,B. (2006) Computational design of a single amino acid sequence that can switch between two distinct protein folds. *J. Am. Chem. Soc.*, **128**, 1154–1161.

Dunbrack,R.L. and Karplus,M. (1993) Backbone-dependent rotamer library for proteins application to side-chain prediction. *J. Mol. Biol.*, **230**, 543–574.

Fromer,M. and Shifman,J.M. (2009) Tradeoff between stability and multispecificity in the design of promiscuous proteins. *PLoS Comput. Biol.*, **5**, e1000627.

Fromer,M. and Yanover,C. (2008) A computational framework to empower probabilistic protein design. *Bioinformatics*, **24**, i214–i222.

Fromer,M. and Yanover,C. (2009) Accurate prediction for atomic-level protein design and its application in diversifying the near-optimal sequence space. *Proteins Struct. Funct. Bioinform.*, **75**, 682–705.

Fromer,M. *et al.* (2010) Design of multispecific protein sequences using probabilistic graphical modeling. *Proteins Struct. Funct. Bioinform.*, **78**, 530–547.

Gordon,D.B. *et al.* (1999) Energy functions for protein design. *Curr. Opin. Struc. Biol.*, **9**, 509–513.

Havranek,J.J. and Harbury,P.B. (2003) Automated design of specificity in molecular recognition. *Nat. Struct. Mol. Biol.*, **10**, 45–52.

Jaimovich,A. *et al.* (2010) FastInf. Available at http://jmlr.csail.mit.edu/papers/v11/jaimovich10a.html (last accessed date August 25, 2010).

Kuhlman,B. *et al.* (2003) Design of a novel globular protein fold with atomic-level accuracy. *Science*, **302**, 1364–1368.

Leach,A.R. and Lemon,A.P. (1998) Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. *Proteins Struct. Funct. Genet.*, **33**, 227–239.

Potapov,V. *et al.* (2009) Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Eng. Des. Sel.*, **22**, 553–560.

Yanover,C. *et al.* (2007) Dead-end elimination for multistate protein design. *J. Comput. Chem.*, **28**, 2122–2129.