

Biosequence Analysis in PRISM

Ole Torp Lassen

Research group PLIS: Programming, Logic and Intelligent Systems
Department of Communication, Business and Information Technologies
Roskilde University, P.O.Box 260, DK-4000 Roskilde, Denmark
E-mail: otl@ruc.dk

In this work, we consider probabilistic models that can infer biological information solely from biological sequences such as DNA. Traditionally, computational models for biological sequence analysis have been implemented in a wide variety of procedural and object oriented programming languages [1]. Models implemented using stochastic logic programming (SLP [2–4]) instead, may draw upon the benefits of increased expressive power, conciseness and compositionality. It does, however, pose a big challenge to design efficient SLP models.

We are currently experimenting with the optimization of a simple model for gene finders written in PRISM [4]. This model plays the role of a canonical model, supposed to hold the best knowledge available about genes, non genes and their respective distributions in DNA. We assume that the canonical model is not computationally practical per se.

As a scheme of preprocessing, we propose to divide the sequence to be analyzed into shorter subsequences that can be analyzed individually by distinct components of the canonical model. We achieve this through decomposition of the canonical model M^{canon} into three distinct components:

- $C1$, a canonical model for distribution of genes and non genes in DNA,
- $C2$, a canonical model for genes and
- $C3$, a canonical model for non genes.

We then define a partitioning model M^{chop} consisting of components:

- $C1$
- $A2$, a simplified generalization of $C2$
- $A3$, a simplified generalization of $C3$.

Given a DNA sequence S , canonical model M^{canon} , and partitioning model M^{chop} , the approximating algorithm can be defined as follows:

1. Apply M^{chop} to S to get the most likely approximate partitioning of S into subsequences, $(s_1, t_1), \dots, (s_n, t_n)$, where t_i is the supposed type of subsequence s_i , (i.e.: gene or non gene).
2. For each approximate subsequence (s_i, t_i) , apply the canonical component corresponding to t_i , (i.e.: $C1$ or $C2$), to get an ordered list of most likely canonical subsequence explanations, $E^{sub} = \{(t_1, e_1), \dots, (t_n, e_n)\}$.
3. Apply $C1$ to E^{sub} to combine subsequence explanations into an approximated most likely explanation of the entire sequence S .

An experimental setup of models was implemented using stochastic context free grammars (SCFGs) to allow for sufficient expressive power. In this setup, explanations can be represented by their corresponding parse trees. For the purpose of

evaluation, a good approximation of the canonical analysis of a sequence S is a parse tree similar or equal to the one produced by the canonical model. Assuming that similar parse trees have similar canonical probabilities, a way to avoid explicit comparison of parse trees is to compare their respective probabilities instead. This is not possible for any realistic S because of the assumed complexity of the canonical model. Instead we have been experimenting with an evaluation scheme that compares the probability of a sampled parse of a sequence S with the probability of the best approximating parse of that sequence. This scheme of evaluation by sampling assumes:

- i)* that randomly sampling the canonical distribution produces sequences with high probability canonical explanations with high frequency, only occasionally producing an atypically sequence and
- ii)* that high probability in the canonical model indicates high quality and vice versa and thus that similar explanations have similar probabilities

Experimental results suggest, however, that assumption *i)* is not satisfied by the canonical model that we have been experimenting with. In fact, when sampled, it very rarely produces a sequence with a better canonical explanation than the one provided by the approximating algorithm. Because we have so far refrained from specifying the distribution of the model, the PRISM system defaults to symmetric distributions and this likely blurs the distinction between typical and atypical sequences. While assumption *i)* clearly depends on the proper specification of the canonical distribution, one way to repair the experimental distribution is to restrict the outcomes to sequences with good canonical explanations.

Despite the difficulties of evaluation, the compositional approach to probabilistic modeling described here offers an alternative way of implementing well-known bioinformatical models in a concise and flexible declarative framework while keeping complexity in check. It also has the potential of providing a rigorous generic framework for combining separately developed models and specialized modules. Finally, the framework may generalize to a wide spectrum of applications, both in bioinformatics and beyond.

References

1. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis. Cambridge University Press (1998)
2. Cussens, J.: Loglinear models for first-order probabilistic reasoning. In Laskey, K.B., Prade, H., eds.: UAI, Morgan Kaufmann (1999) 126–133
3. Muggleton, S.: Learning from positive data. In Muggleton, S., ed.: Inductive Logic Programming Workshop. Volume 1314 of Lecture Notes in Computer Science., Springer (1996) 358–376
4. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. *J. Artif. Intell. Res. (JAIR)* **15** (2001) 391–454