Constructing (Almost) Phylogenetic Trees from Developmental Sequences Data

Ronnie Bathoorn and Arno Siebes

Institute of Information & Computing Sciences
Utrecht University
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{ronnie,arno.siebes}@cs.uu.nl

Abstract. In this paper we present a new way of constructing almost phylogenetic trees. Almost since we reconstruct the tree, but without the timestamps. Rather than basing the tree on genetic sequence data ours is based on developmental sequence data. Using frequent episode discovery and clustering we reconstruct the consensus tree from the literature almost completely.

1 Introduction

One of the big open problems in modern biology is the reconstruction of the course of evolution. That is, when did what (class of) species split off another (class of) species and what were the changes? The ultimate goal is to reconstruct the complete tree of life as correct as possible given the current species and the fossil record. Such a tree is often called an *evolutionary tree* or a *phylogenetic tree*.

Most research in this area uses clustering techniques on genetic data. Using knowledge on, e.g., the rate of change, time-stamps are computed for the different splits. In this paper we show how to reconstruct such trees with a different kind of data, viz., the order of events in the development of an animal [1, 2]. Examples of the events are the start of the development of the Heart or of the Eyes. This sequence is fixed for a given species, but varies over species.

2 Method

To construct our tree, we compute the frequent episodes [4, 3] over these event sequences in step 1. Next in step 2 we gather the occurence of these episodes in profiles, which we use to calculate the dissimilarity between species using the Jaccard dissimilarity measure in step 3. Finally in step 4, we use agglomerative hierarchical clustering with complete linkage on this dissimilarity matrix.

The resulting tree has no external information on, e.g., the rate of change, we cannot label splits with their probable occurrence in time. Hence we call our tree an *Almost Phylogenetic Tree*.

3 Experimental Results

The experimental results show that clustering results are almost as good as the Phylogenetic trees found in biological literature as can be seen in Figure 2. This result is

J.-F. Boulicaut et al. (Eds.): PKDD 2004, LNAI 3202, pp. 500–502, 2004.

BUILD_TREE$(data, min_fr, win)$

1 $episodes = FindEpisodes(data, min_fr, win)$
2 $profiles = makeProfiles(episodes)$
3 $dist = makeDistanceMatrix(profiles)$
4 **return** $HierarchicalClustering(dist)$

Fig. 1. Base algorithm

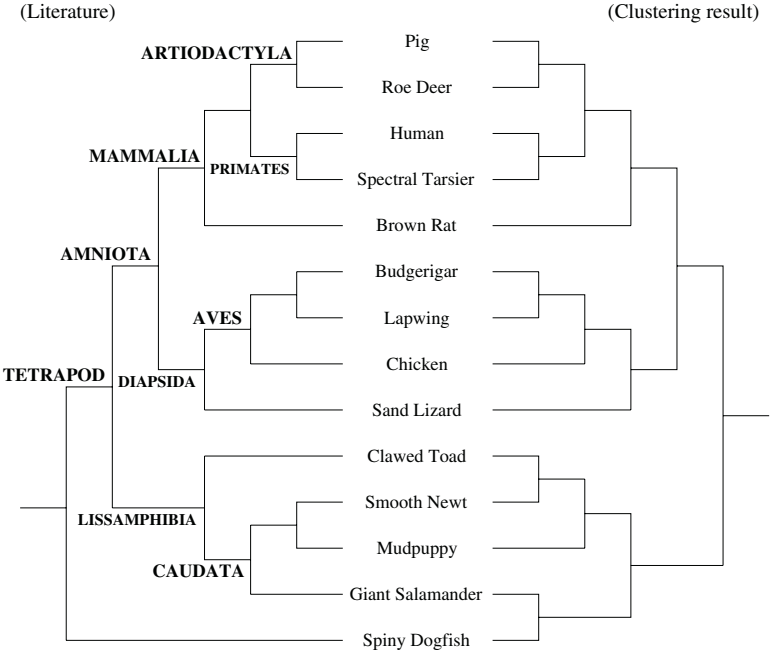

Fig. 2. Comparison of results to the tree from the literature

obtained with a window size of 6 and a minimal frequency of 5%. Lowering the minimal frequency produces trees that become gradualy worse, because the amount of episodes found that have no influence on the distance between species tend to hide the episodes that are key in seperating certain groups of species in the phylogenetic tree. And increasing the window size also produces trees which are worse for the same reason.

Note that the distance in our phylogenetic tree is the distance between species for our distance measure and that it does not mean that the species at the bottom of the tree arose later in evolution. This is because there is no sense of time in these figures. When comparing our experimental result to the Phylogenetic Tree currently accepted in literature we can clearly see that birds and mammals are clustered really well. The amphibians are grouped together but not entirely in the right configuration.

Finally the Spiny Dogfish should have been a group on its own but probably ended up with the amphibians because it was the only creature of its kind in our data set. Which is why episodes being able to separate it from the rest had frequencies which were to low for it to be used in the clustering. That is why we clustered our data with the Spiny Dogfish left out. This gave no changes to the clustering of the rest of the species, giving us reason to believe that patterns able to separate Spiny Dogfish from the rest of the species are not used in the clustering.

4 Conclusions and Future Work

There are a number of reasons that make this application interesting. Firstly, it is a nice illustration of the concept of an *inductive database* [6] given that in the first phase we mine the data for frequent episodes and in the second phase we mine these episodes for the final tree. Moreover, the frequent episodes provide interesting insight in developmental biology by themselves. The second reason is that it illustrates the power of data mining methods. Without expert knowledge we are able to almost reconstruct the consensus tree in the literature, which is based on a lot of biological knowledge. Finally, it shows how much information on evolution is preserved in these developmental event sequences.

This work is part of a larger project that tries to utilize different types of data about the development of different species, such as anatomical data, gene expression data (micro array) and developmental sequence data, e.g., for the construction of evolutionary trees.

References

1. Jonathan E. Jeffery and Olaf R. P. Bininda-Emonds and Michael I. Coates and Michael K. Richardson: Analyzing evolutionary patterns in amniote embryonic development. Evolution & Development Volume 4 Number 4 (2002) 292–302
2. Jonathan E. Jeffery and Michael K. Richardson and Michael I. Coates and Olaf R. P. Bininda-Emonds: Analyzing Developmental Sequences Within a Phylogenetic Framework. Systematic Biology Volume 51 Number 3 (2002) 478–491
3. Mannila, Heikki. and Toivonen, Hannu. and Verkamo, A. Inkeri.: Discovering frequent episodes in sequences. First International Conference on Knowledge Discovery and Data Mining (1995) 210–215
4. Rakesh Agrawal and Ramakrishnan Srikant: Mining Sequential Patterns. International Conference on Data Engineering (1995) 3–14
5. P. W. Holland: The future of evolutionary developmental biology. Nature 402 (1999) 41–44
6. L. De Raedt: A perspective on inductive databases. ACM SIGKDD Explorations Newsletter Volume 4 Issue 2 (2002) 69–77