Tpms: a Tree Pattern-matching Utility for Querying Gene Trees Collections

Thomas BIGOT, Vincent DAUBIN and Guy PERRIÈRE
Laboratoire de Biométrie et Biologie Évolutive, UMR 5558 CNRS, Université Claude Bernard – Lyon 1, 43 bd. du 11 Novembre 1918, 69622, Villeurbanne Cedex, France
{thomas.bigot,vincent.daubin,guy.perriere}@univ-lyon1.fr

Abstract Tpms is a portable C++ program allowing to retrieve gene trees from large collections, this according to tree patterns defined by the users. It can be used for different purposes such as orthologs search or horizontal gene transfers identification. Documentation, source code, as well as Linux and MacOSX binaries can be freely downloaded at ftp://pbil.univ-lyon1.fr/pub/mol_phylogeny/tpms/.

Keywords phylogenetic trees, tree pattern-matching, orthologs, horizontal gene transfers.

1 Introduction

Comparative genomics is a common approach in sequence analysis, and many biological results have been obtained through its use. Among the different programs and packages developed for comparative genomics, those using the information contained in phylogenetic trees are of special interest. Indeed, orthology detection methods using phylogenetic trees usually perform better than the simpler (and easier to use) methods based on best reciprocal hits of sequence similarity scores [1]. In that context we developed tpms, a program allowing to retrieve gene trees from a tree collection, this according to patterns defined by the users. Those patterns usually include some kind of constraints, such as node nature (duplication, speciation), or subtree content. Therefore this program can be used for orthologs search, but also for any studies that require to retrieve sets of gene families matching constraints in their corresponding phylogenetic tree (e.g., gene duplications identification or horizontal gene transfers prediction).

2 System and methods

The tree pattern-matching algorithm used in tpms is a C++ version of the one from the RAP program implemented in FamFetch [2]. It requires the Bio++ [3, 4] and Boost [5] libraries to be run. This new implementation consists in a command-line standalone binary and is not embedded into a graphical interface. Moreover, it is also no longer dependant on the use of the HOVERGEN, HOGENOM and HOMOLENS gene families databases [6], and it can be used on collections build by the users. Binaries of the program are provided for Linux and MacOSX (Intel architectures only), as well as the source code at ftp://pbil.univ-lyon1.fr/pub/mol_phylogeny/tpms/.

3 Program use

At first, the user needs to build a gene trees collection in the RAP format [7]. This collection will be then accessed by the program when performing pattern searches. Collection construction can be done easily through the use of tpms_mkdb program, distributed with tpms. This tool uses a reference species tree, also in RAP format, and a set of individual gene trees in standard Newick format. The species tree can contain unresolved nodes (multifurcations), but not the individual gene trees.

Tree patterns have to be written in an extended Newick format. The simplest constraints that can be introduced are represented by the taxa found on the leaves of the pattern. The labels can stand for a given species (e.g., Homo sapiens) or larger taxonomic groups (e.g., Primates). For instance, the pattern:

((Homo sapiens, Pan troglodytes),Rodentia)
will allow to find all the gene trees in which a subtree with sequences from \textit{H. sapiens} and \textit{Pan troglodytes} species are grouped, while sequences from any rodents are located outside of this group.

The programs also allows to specify constraints on subtrees: one can ask that a subtree from the entered pattern will only contains genes from a defined set of species. This set can be one species, one taxon, or a list with a combination of species and taxa. This makes it possible to exclude a subgroup of a larger group. Elements of the list are bracketed, and indication of addition or removal is done through + or − operators. For example, the pattern:

\[
((\text{Homo sapiens}, \text{Pan troglodytes}), \text{Mammalia}\{-\text{Primates}\})
\]

will allow to find all the gene trees in which a subtree with sequences from \textit{H. sapiens} and \textit{P. troglodytes} species are grouped, while any sequences from mammals – excluding primates – is located outside this group.

A third kind of constraint can be set on nodes if the program is running on a reconciled trees collection: it is possible to search specifically for speciation or duplication nodes. This kind of nodes can be specified in the pattern by the use of letters \textit{S} or \textit{D}. In the following example:

\[
(\text{Homo sapiens}, \text{Mus musculus})(\text{D})
\]

will allow to find all the gene trees in which a subtree with sequences from \textit{H. sapiens} and \textit{M. musculus} species are grouped, while the node that groups them is a duplication node.

In order to search for orthologs, the program can perform queries in the gene trees collection with a pattern extracted from the reference species tree. Orthologs can be then identified from the subparts in the gene trees that match that pattern. Another possibility is to search for all gene trees in which there is a subtree containing a list of taxa defined by the user, whatever the topology of the subtree is, and giving the fact that all taxa are present in single copy. This later approach is suited only for the detection of 1:1 orthology relationships, but it is extremely fast and can be used on very large collections containing thousands of trees.

Search for horizontal gene transfers is also straightforward, as it will only require to enter anomalous patterns, this relatively to the reference species tree.

References