Bioinformatics, 31(18), 2015, 3051–3053
doi: 10.1093/bioinformatics/btv306
Advance Access Publication Date: 18 May 2015
Applications note

OXFORD

Sequence analysis

TRAL: tandem repeat annotation library

Elke Schaper[1,2,3,*], Alexander Korsunsky[4], Jūlija Pečerska[2,3,5], Antonio Messina[6], Riccardo Murri[6], Heinz Stockinger[2], Stefan Zoller[2,3], Ioannis Xenarios[1,2] and Maria Anisimova[2,7]

[1]Vital-IT group, SIB Swiss Institute of Bioinformatics, Quartier Sorge, 1015 Lausanne, Switzerland, [2]SIB Swiss Institute of Bioinformatics, Quartier Sorge, 1015 Lausanne, Switzerland, [3]Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland, [4]Graz University of Technology, Institute of Molecular Biotechnology, 8010 Graz, Austria, [5]Department of Biosystems Science and Engineering, ETH Zürich, 4058 Basel, Switzerland, [6]Services and Support for Science IT, University of Zürich, 8057 Zürich, Switzerland and [7]Institute of Applied Simulations, School of Life Sciences und Facility Management, Zürich University of Applied Sciences, 8820 Wädenswil, Switzerland

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on March 25, 2015; revised on April 24, 2015; accepted on May 8, 2015

Abstract

Motivation: Currently, more than 40 sequence tandem repeat detectors are published, providing heterogeneous, partly complementary, partly conflicting results.

Results: We present TRAL, a tandem repeat annotation library that allows running and parsing of various detection outputs, clustering of redundant or overlapping annotations, several statistical frameworks for filtering false positive annotations, and importantly a tandem repeat annotation and refinement module based on circular profile hidden Markov models (cpHMMs). Using TRAL, we evaluated the performance of a multi-step tandem repeat annotation workflow on 547 085 sequences in UniProtKB/Swiss-Prot. The researcher can use these results to predict run-times for specific datasets, and to choose annotation complexity accordingly.

Availability and implementation: TRAL is an open-source Python 3 library and is available, together with documentation and tutorials via http://www.vital-it.ch/software/tral.

Contact: elke.schaper@isb-sib.ch

1 Introduction

Tandem repeats (TRs) are sequence features, where motifs, or TR units, are found right next to each other, often as imperfect repetitions (Fig. 1A). Currently, more than 40 TR detector (TRD) programs exist, each focusing on different TR types and using different methodologies. We have shown that currently available TRDs do not provide exhaustive detections, and combining TRs from several TRDs is essential for reliable TR annotation (Schaper *et al.*, 2012). Therefore, a researcher interested in TRs needs to handle all of the following tasks: (i) Executing and parsing results of several TRDs, despite no commonly accepted file format; (ii) Validating TR predictions and clustering redundant or overlapping results; (iii) Filtering out false positive TR predictions in a robust statistical framework; (iv) Annotating known TRs homogenously across homologous sequences, and discerning variation among the TRs (Anisimova *et al.*, 2015). Each of these tasks is implemented in TRAL—an open source Python 3 TR annotation library. TRAL is highly modularized, such that a researcher can use the implemented methods or customize them by adding other TRDs, overlap criteria, statistical tests or model-based annotation methods. The software is designed to run efficiently and user-friendly on single machines as well as on large computing clusters. For a variety of workflows, scripts and tutorials are available online.

2 Features and methods

An overview of the structure of TRAL is shown in Figure 1B.

2.1 Annotate with sequence profile models

A common task is to annotate sequences with TRs of a known motif. This allows to study the evolution of the TR across sequence

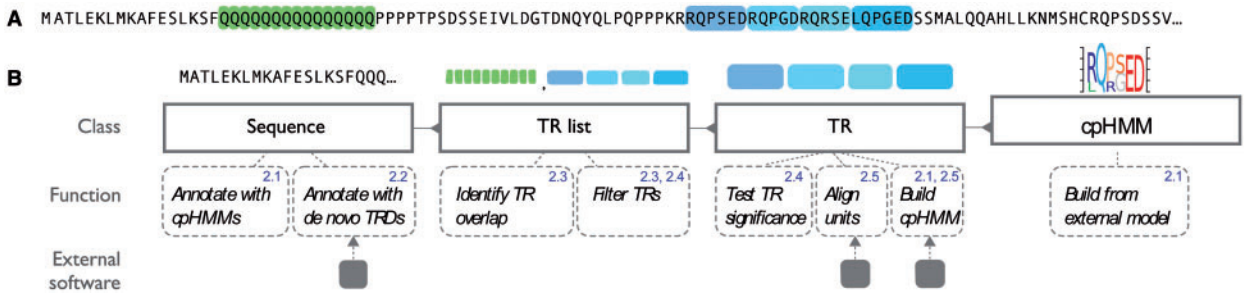

Fig. 1. (**A**) A sequence with two TRs: a poly-Q track (green), and a short TR (blue). (**B**) Illustration of TRAL structure together with the four most important data classes: a **sequence** can contain **TR lists**; a TR list can contain **TRs**; a TR can contain a **cpHMM**. For different use cases, different workflows can be build using these classes. For each class, input/output methods for a variety of formats are available. External software is described in the online documentation

homologues, or to check whether a common motif occurs in tandem within a sequence.

TRAL implements a circular profile hidden Markov model (cpHMM), adapting popular sequence profile models, e.g. from HMMER (Eddy, 1998; Finn *et al.*, 2011) to TRs (Bucher *et al.*, 1996; Schaper and Anisimova, 2015; Schaper *et al.*, 2014; Uricaru *et al.*, 2007). Accurate TR annotation in the maximum likelihood framework is realized with the Viterbi algorithm, such that homologous TRs are homogenously annotated. As all TR annotations are described by the same profile model, they are comparable in terms of their characteristics (TR unit number, length, indels), enabling evolutionary studies (Schaper *et al.*, 2014). cpHMMs can be created from single sequences, TRs or sequence profile models from databases.

2.2 Annotate with *de novo* tandem repeat detectors

For *de novo* annotations, we implemented a scaffold for executing and parsing external TRD software. Six current TRDs are currently integrated:

HHrepID (Biegert and Söding, 2008), Phobos (www.ruhr-uni-bochum.de/ecoevo/cm/cm_phobos.htm), TRED (Sokol *et al.*, 2007), T-REKS (Jorda and Kajava, 2009), TRF (Benson, 1999), TRUST (Szklarczyk and Heringa, 2004) and XSTREAM (Newman and Cooper, 2007). Further TRDs can easily be added to the framework. As we noted that some TRDs sometimes propose TRs that are not part of the input sequence, an automatic sanity check discards these.

Importantly, profile HMMs can be used to refine *de novo* TR annotations. For example, a TRD may correctly identify a TR, but not capture all its TR units or the correct TR unit boundaries. The refinement can then be achieved by re-annotating with cpHMMs.

2.3 Identify and filter overlapping annotations

Different TRDs often predict overlapping TRs, and congruent predictions are very rare (Schaper *et al.*, 2012). The user may be interested in discarding redundant TRs. However, not all overlapping predictions describe a TR redundantly (Anisimova *et al.*, 2015). We included a flexible system to establish overlap and clustering of TRs in TRAL. Two definitions of overlap for a pair of TRs are currently implemented: (i) having at least some characters in common and (ii) having a common ancestry of at least one pair of characters in alignments of multiple TR units for both TRs. In the next step, the clustered TR annotations can be filtered to contain only the best TR representative from a cluster according to user-defined criteria.

2.4 Test and filter for statistical significance

Distinguishing true from false positive TR annotations is another important task, which requires evaluating the statistical significance of a TR annotation. Several *ad hoc* and model-based statistics based on the multiple alignment of TR units have been proposed for this purpose (Schaper *et al.*, 2012). TRAL implements these statistics, as well as their null-distributions for random TRs, such that the statistical significance of each TR (*P*-value) can be calculated.

Table 1. Performance evaluation of TR annotation

Task		Runtime (s)	Memory (MB)
Annotate with cpHMMs		0.3 ± 0.5	1.0 ± 0.9
Annotate with *de novo* TRDs	HHrepID	2.8 ± 2.6	0.5 ± 0.2
	T-REKS	0.90 ± 0.37	0.11 ± 0.06
	TRUST	1.9 ± 2.5	1.1 ± 2.1
	XSTREAM	0.24 ± 0.08	0.07 ± 0.01
Refine *de novo* TRs with cpHMMs		1.1 ± 1.3	0.3 ± 0.4
Identify and filter overlapping TRs		0.00 ± 0.00	0.05 ± 0.02
Test and filter for statistical significance		0.36 ± 0.23	0.22 ± 0.02

TR annotation requirements per sequence on UniProtKB/Swiss-Prot (v11/2014; 547 085 sequences, 1000 sequence per evaluated batch) on standard Intel Xeon hardware, e.g. E5 family. Total computation time including overheads was ~50 days.

2.5 Retrieve tandem repeat characteristics

For each putative TR, TRAL provides access to characteristics such as TR unit alignments, TR unit length, number, divergence and indel distribution. The TR unit alignment can be optimized with attached external multiple sequence alignment software. Finally, TRAL can be used to build cpHMMs from each TR for annotation of homologous TRs on other sequences, or for iterative optimization of the TR annotation (see Section 2.1).

3 Performance evaluation

We evaluated the performance (runtime and memory requirements) of different TR annotation tasks with TRAL for 547 085 sequences in UniProtKB/Swiss-Prot (Table 1). TRs were annotated with four TRDs and with cpHMMs based on PFAM models (Punta *et al.*, 2011), and filtered for statistical significance and overlap. On average, such annotation required ~8 s per protein sequence, mostly depending on the number and type of *de novo* TRDs used. For large annotation projects requiring parallelization, TRAL includes an annotation workflow, which can be run with GC3Pie—an open-source

workflow management system for diverse local, grid, cluster and cloud-based computing resources (Maffioletti and Murri, 2012; https://www.s3it.uzh.ch/software/gc3pie/).

Funding

ZHAW Anschubfinanzierung to M.A. be the SIB Swiss Institute of Bioinformatics is supported by the Swiss State Secretariat for Education, Research, and Innovation. The computation has been performed on the Vital-IT HPC.

Conflict of Interest: none declared.

References

Anisimova,M. *et al*. (2015) Statistical approaches to detecting and analyzing tandem repeats in genomic sequences. *Front. Bioeng. Biotechnol.*, **3**, http://doi.org/10.3389/fbioe.2015.00031.

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.*, **27**, 573–580.

Biegert,A. and Söding,J. (2008) De novo identification of highly diverged protein repeats by probabilistic consistency. *Bioinformatics*, **24**, 807–814.

Bucher,P. *et al*. (1996) A flexible motif search technique based on generalized profiles. *Comput. Chem.*, **20**, 3–23.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Finn,R.D. *et al*. (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, **39**, W29–W37.

Jorda,J. and Kajava,A.V. (2009) T-REKS: identification of Tandem REpeats in sequences with a K-meanS based algorithm. *Bioinformatics*, **25**, 2632–2638.

Maffioletti,S. and Murri,R. (2012) GC3Pie: a Python framework for high-throughput computing. In: *Proceedings of the EGI Community Forum 2012/EMI Second Technical Conference (EGICF12-EMITC2), 26–30 March, 2012*. Munich, Germany.

Newman,A.M. and Cooper,J.B. (2007) XSTREAM: a practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics*, **8**, 382.

Punta,M. *et al*. (2011) The Pfam protein families database. *Nucleic Acids Res.*, **40**, D290–D301.

Schaper,E. and Anisimova,M. (2015) The evolution and function of protein tandem repeats in plants. *New Phytol.*, **206**, 397–410.

Schaper,E. *et al*. (2012) Repeat or not repeat? Statistical validation of tandem repeat prediction in genomic sequences. *Nucleic Acids Res.*, **40**, 10005–10017.

Schaper,E. *et al*. (2014) Deep conservation of human protein tandem repeats within the eukaryotes. *Mol. Biol. Evol.*, **31**, 1132–1148.

Sokol,D. *et al*. (2007) Tandem repeats over the edit distance. *Bioinformatics*, **23**, e30–e35.

Szklarczyk,R. and Heringa,J. (2004) Tracking repeats using significance and transitivity. *Bioinformatics*, **20**, i311–i317.

Uricaru,R. *et al*. (2007) A new type of hidden Markov models to predict complex domain architecture in protein sequences. In: *JOBIM'07*, pp. 97–102.