Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Applications Note

Sequence analysis

SecretSanta: flexible pipelines for functional secretome prediction

Anna Gogleva [1]*, **Hajk-Georg Drost** [1], **Sebastian Schornack** [1]

[1] University of Cambridge, Sainsbury Laboratory, Cambridge, United Kingdom.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The secretome denotes the collection of secreted proteins exported outside of the cell. The functional roles of secreted proteins include the maintenance and remodelling of the extracellular matrix as well as signalling between host and non-host cells. These features make secretomes rich reservoirs of biomarkers for disease classification and host-pathogen interaction studies. Common biomarkers are extracellular proteins secreted via classical pathways that can be predicted from sequence by annotating the presence or absence of N-terminal signal peptides. Several heterogeneous command line tools and web-interfaces exist to identify individual motifs, signal sequences and domains that are either characteristic or strictly excluded from secreted proteins. However, a single flexible secretome-prediction workflow that combines all analytic steps is still missing.
Results: To bridge this gap the *SecretSanta* package implements wrapper and parser functions around established command line tools for the integrative prediction of extracellular proteins that are secreted via classical pathways. The modularity of *SecretSanta* enables users to create tailored pipelines and apply them across the whole tree of life to facilitate comparison of secretomes across multiple species or under various conditions.
Availability: *SecretSanta* is implemented in the R programming language and is released under GPL-3 license. All functions have been optimized and parallelized to allow large-scale processing of sequences. The open-source code, installation instructions, and vignette with use case scenarios can be downloaded from https://github.com/gogleva/SecretSanta.
Contact: anna.gogleva@slcu.cam.ac.uk
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Secreted proteins play a vital role in interactions between neighbouring cells within the same organism as well as between different species engaged in close contact. Their role is pivotal in the context of host-pathogen interactions, where numerous virulence factors are secreted to facilitate recognition and colonization of host cells and tissues. Hence, robust prediction of secreted proteins is usually the first step towards the informed development of efficient infection control strategies.

Secretome prediction requires multiple steps. In the first step, short signal peptides are determined at the N-terminal end of a protein sequence. Next, it is crucial to ensure the absence of specific sequence motifs and domains that target the protein to specific organelles or prevent it from being secreted. Such sequences include transmembrane domains, endoplasmic reticulum (ER) lumen retention signals and mitochondria/plastid targeting signals.

A variety of command line tools and web-interfaces are available to perform predictions of individual motifs and domains, such as SignalP (**?**) for the prediction of signal peptides; TargetP (**?**) and WolfPsort (**?**) for the prediction of subcellular localization; TMHMM (**?**) and TOPCONS (**?**) - for the prediction of transmembrane domains. However, an interface that combines these tools in a single flexible workflow for standardized and efficient secretome prediction was previously unavailable.

SecretSanta bridges this gap by providing a set of optimised wrapper functions around a variety of command line tools. These functions are designed to generate tidy data output to seamlessly integrate into the R data science infrastructure. Users can pipe intermediate outputs between individual methods, run resource-demanding stages in parallel, compare predictions between similar methods and trace back selected steps of the pipeline. Taken together, *SecretSanta* provides a platform to build reproducible and flexible pipelines for scalable secretome prediction. This framework can facilitate comparisons of secretomes across multiple species or under various conditions and can potentially lead to the discovery of novel classes of previously neglected secreted proteins.

2 Implementation

SecretSanta is released under GPL-3 license. The source code along with detailed usage guidelines and unit tests is publicly available at https://github.com/gogleva/SecretSanta. *SecretSanta* requires a number of external dependencies: *SignalP 2.0*, *SignalP 3.0*, *SignalP 4.1*, *TargetP 1.1*, *TMHMM 2.0c*, *WoLF PSORT* which are freely available for academic users. It also depends on several R packages: *Biostrings*, *dplyr*, *parallel*, *reardr*, *tibble*, *stringr* and *biomartr* package (**?**) for automated retrieval of proteomes. The *SecretSanta* package contains wrapper functions around the external dependencies as well as the `check_khdel()` function to scan for C-terminal ER-retention signals, the `m_slicer()` function to generate peptides with alternative translation start sites and the convenient `ask_uniprot()` function to retrieve known information on protein subcellular localisation from UniprotKB. All *SecretSanta* functions are designed to work together by producing standardized output as an instance of the superclass `CBSResult`. Objects of `CBSResult` class contain an `in_fasta` slot storing the submitted sequences and an `out_fasta` slot storing positive candidates after the prediction run. Each particular method then complements `CBSResult` simple structure with relevant slots. For example, outputs of the `signalp()` function are organised in `SignalpResult` objects. In addition to inherited `in_fasta` and `out_fasta` slots, `SignalpResult` objects contain three additional slots, relevant for the prediction of signal peptides:

- `sp_tibble` - parsed SignalP tabular output for positive candidates;
- `mature_fasta` - sequences with cleaved signal peptides;
- `sp_version` - version of SignalP software used.

The underlying class structure allows the user to easily pipe results between individual methods to create flexible custom pipelines and also to compare results generated with similar methods. Piping could be done either explicitly, by specifying output object from the previous analytic step as an input to the next, or - implicitly - by using the pipe operator %>% provided by the *magrittr* package. Finally, processing of large input *fasta* files can be run in parallel making them suitable for HPC usage.

3 Usage

To illustrate the core functionality of the SecretSanta package we will establish a minimal pipeline with the following analytic steps:

- use SignalP-4.1 to predict signal peptides and cleavage sites;
- integrate TOPCONS predictions to ensure that proteins do not contain transmembrane domains; note, the `tmhmm()` function could also be used, we recommend TOPCONS for more accurate results;
- run TargetP on the output to make sure that the set of selected proteins is not targeted to plastids or mitochondria;
- collect the TargetP output and scan for ER-retention signals.

First we read a short sample *fasta* file and initialise an instance of `CBSResult` class:

```
input <- CBSResult(in_fasta =
 readAAStringSet("proteins.fasta"))
```

Next, we start our pipeline by running the `signalp()` function:

```
a <- signalp(input, version = 4, organism = 'euk',
run_mode = 'starter')
```

Now we can use the `a` object as an input to the `topcons()` parser to integrate prediction of transmembrane domains.

```
b <- topcons(a, TM = 0, p_dir = 'rst.zip',
topcons_mode = 'WEB-server')
```

To check for potential mitochondria or plastid targeting signals, we will apply `targetp()` function to the resulting `b` object. Please note, that `targetp()` will use `out_fasta` slot with the full sequence as an input:

```
c <- targetp(b, network = 'N', run_mode = 'piper')
```

Finally, we will make sure that there are no ER-retention signals, defined according to the PROSITE model:

```
d <- check_khdel(c, pattern = 'prosite')
```

More detailed tutorials are available in the online Supplementary material.

4 Conclusion

SecretSanta functions provide an integrated new way to predict and analyze secretomes. The flexibility and modularity of the pipelines grants analytical freedom and computational reproducibility to fuel comparative studies of secreted proteins across the tree of life.

Acknowledgements

We would like to thank Phil Carella and Edouard Evangelisti for providing useful feedback at the early stage of the package.

Funding

This work was supported by the Gatsby Charitable Foundation (RG62472), the Royal Society (RG69135) and by the European Research Council (ERC-2014-STG, H2020, 637537).

References

Drost, H-G. and Paszkowski, J. (2017) Biomartr: genomic data retrieval with R. *Bioinformatics*, **8**, 1216-1217.

Emanuelsson, O. *et al.* (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.*, **300**, 1005-1016.

Horton, P. *et al.* (2006) Protein subcellular localization prediction with WoLF PSORT. *Asian Pacific Bioinformatics Conference, APCB2006*, 39-48.

Nielsen, H., *et al.* (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, **10**, 1-6.

Sonnhammer, E. *et al.* (1998) A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc. Sixth Int. Conf. on Intelligent Systems for Molecular Biology*, 175-182

Tsirigos, K.D. *et al.* (2015) The TOPCONS web server for combined membrane protein topology and signal peptide prediction. *Nucleic Acids Research*, **43**, W401-W407.