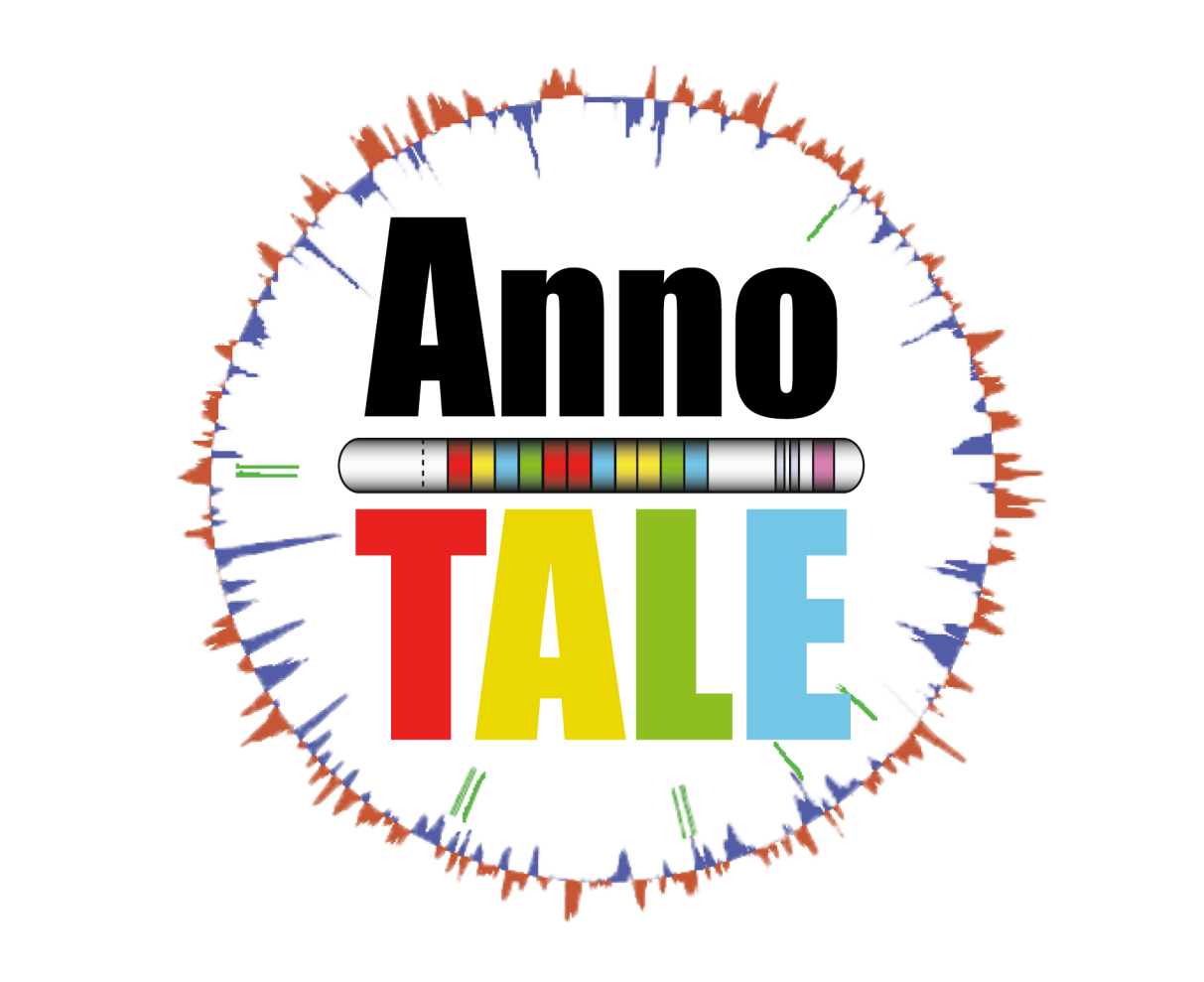


AnnoTALE User Guide

Version 1.0



Predicting, Annotating, and Classifying TALEs from *Xanthomonas* spp.

AnnoTALE User Guide

Maik Reschke, Jens Boch, and Jan Grau

Contact: grau@informatik.uni-halle.de or jens.boch@genetik.uni-hannover.de

AnnoTALE application suite and User Guide can be downloaded at:

<http://www.jstacs.de/index.php/AnnoTALE>

If you use AnnoTALE in your research, please cite:

Jan Grau, Maik Reschke, Annett Erkes, Jana Streubel, Richard D. Morgan, Geoffrey G. Wilson, Ralf Koebnik, and Jens Boch: AnnoTALE: bioinformatics tools for identification, annotation, and nomenclature of TALEs from *Xanthomonas* genomic sequences.

Disclaimer

AnnoTALE is distributed with the intention to be useful, but without any warranty, and without the implied warranty of merchantability or fitness for a particular purpose.

Table of Contents

A Suite of applications - AnnoTALE	3
1) TALE Prediction.....	5
2) TALE Analysis	6
3) TALE Class Builder (optional).....	7
4) Load and View TALE Classes	9
5) TALE Class Assignment	10
6) Rename TALE in File.....	14
7) Predict and Intersect Targets	15
 Quick Start Guide	 18
 References	 19
 Download & Installation	 19

A Suite of applications - AnnoTALE

AnnoTALE is a program suite for predicting, annotating, and classifying TALEs from *Xanthomonas spp.* The suite consists of seven programs that help the user to analyse a *Xanthomonas spp.* genome sequence with respect to its transcription activator-like effectors (TALEs). AnnoTALE can be used for the prediction of *TALE* genes, classifying TALEs based on their RVD sequences, assigning systematic names to TALEs, and for the prediction of possible target genes in a given target organism (e.g. the rice promoterome).

The interface of AnnoTALE consists of three main parts: the **toolbar**, the **data panel** and the **viewer**. (Fig. 1)

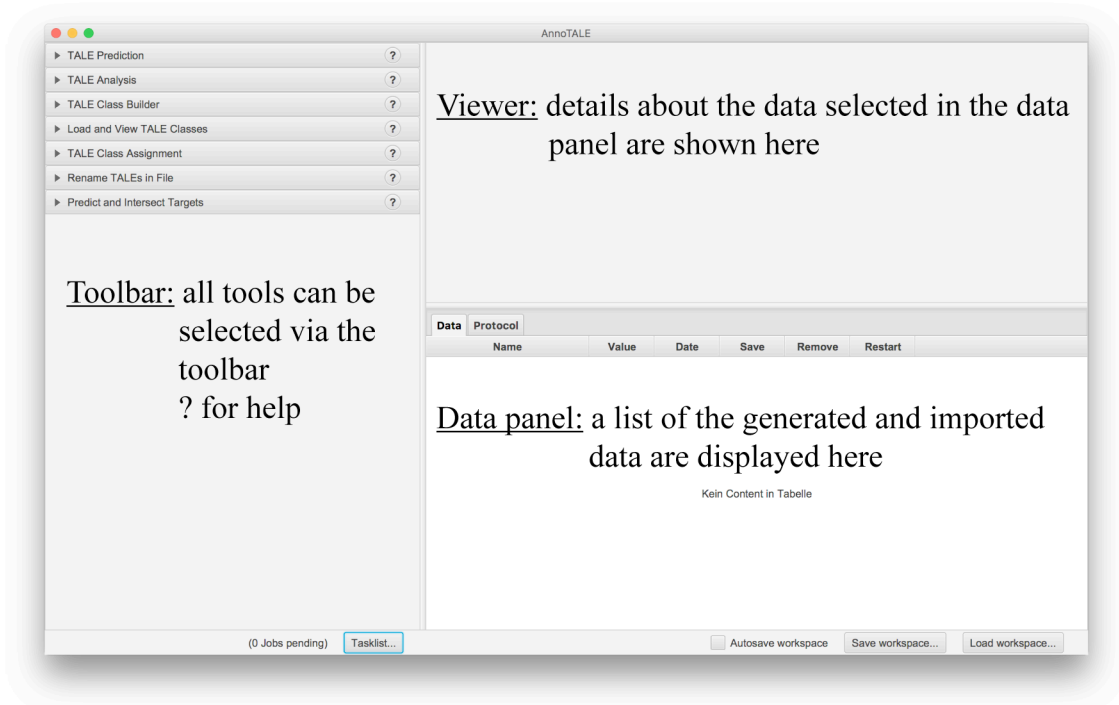


Figure 1: Interface of AnnoTALE after starting the application.

The **toolbar** contains a list of all seven AnnoTALE tools. By clicking on the header of a specific tool, its input parameters are shown and can be specified. The header of each tool includes a button with a **?**, which leads to a separate help window with a short description of the purpose and usage of the corresponding tool.

The **data panel** contains two tabs, **Data** and **Protocol**. The **Data** tab shows all data that have been loaded into AnnoTALE or that have been produced as results of one of the AnnoTALE tools. By clicking on one of the entries in the data panel, the corresponding data are visualized in the **viewer** (Fig. 2).

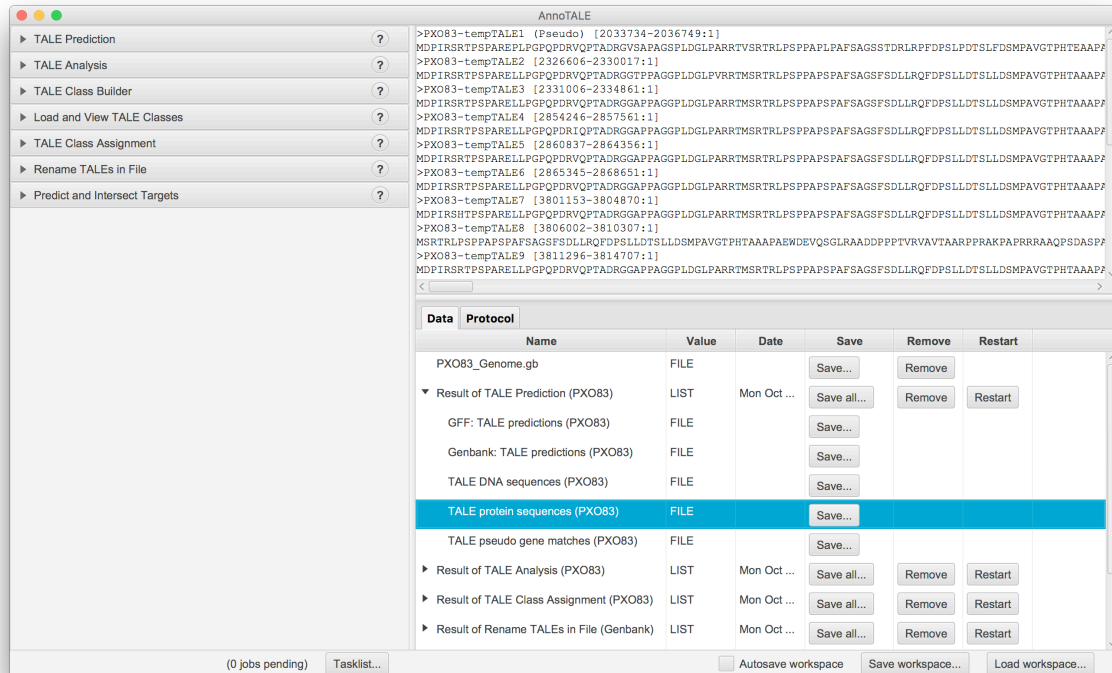


Figure 2: Interface of AnnoTALE showing data in the data panel and viewer

Each of the data entries in the data panel may be saved to disk using the [Save...](#) button in the corresponding row (Fig. 2). Similarly, sets of results may be saved to a directory using the [Save all...](#) button in rows representing aggregate results. In the latter case, the user may specify a target folder and AnnoTALE chooses the names of the output files automatically. In the second tab [Protocol](#), a protocol of current and previous runs of AnnoTALE tools is shown, which contains additional information in case of errors (Fig. 3).

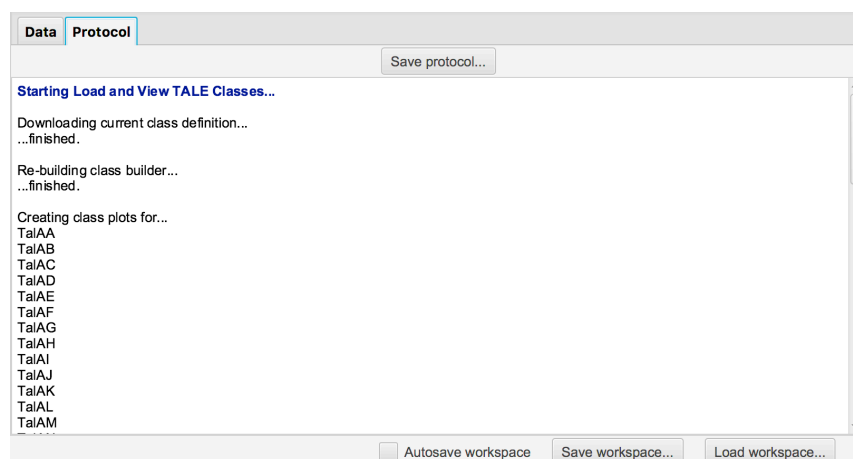


Figure 3: Protocol panel of AnnoTALE

At the **bottom** of the AnnoTALE window (Fig. 2), a bar with further buttons allows the user to access general functionalities. The button [Tasklist...](#) shows a list of all running and scheduled jobs of AnnoTALE. Using the buttons [Save workspace...](#) and [Load workspace...](#),

the current AnnoTALE workspace, i.e., all data shown in the data panel, may be saved to disk in a monolithic file and loaded from disk, respectively. Loading a workspace from disk adds all data entries stored in that workspace file to the current workspace (instead of replacing the current workspace contents). Using the checkbox [Autosave workspace](#), the workspace will be stored to disk automatically after each and every modification of the workspace, and will also restore that workspace after closing and re-opening AnnoTALE. Currently, this feature should be considered experimental and may reduce the responsiveness of the AnnoTALE GUI.

In the remainder of this User Guide, we will describe each of the seven AnnoTALE tools in more detail.

1) TALE Prediction

The tool [TALE Prediction](#) predicts *TALE* genes in a given genomic sequence. With a click on the button [Load from file...](#), a genome sequence (*Xanthomonas spp.*) may be loaded from a file on the local hard drive (Fig. 4). FastA or Genbank files are accepted.

Figure 4: TALE Prediction – input mask

If a Genbank file is used, all existing annotations will be preserved and annotations for the predicted TALE genes will be added to the existing annotation. It is possible to label the input genome with the name of the *Xanthomonas* strain, in this example PXO83, which is useful when successively predicting TALE genes in several *Xanthomonas* genomes. By clicking on the button [Run TALE Prediction...](#), the prediction tool is started and a progress bar is shown in the lower part of the AnnoTALE window.

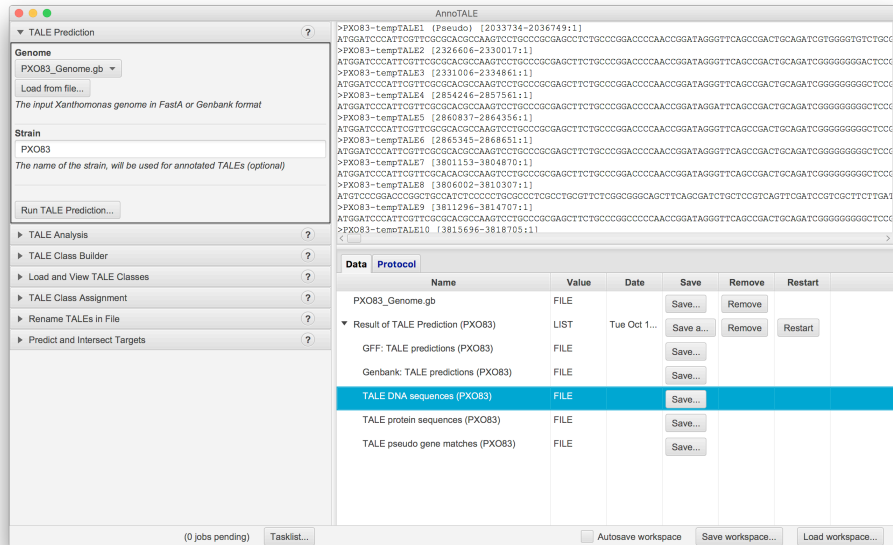


Figure 5: Result of TALE Prediction

After the process has finished, **Result of TALE Prediction** appears in the data panel (Fig. 5). The tool "TALE Prediction" creates a GFF and Genbank file of the genome, where the TALE genes are annotated and labeled with temporary names (tempTALE1...). Furthermore, the program provides FastA files with the DNA sequence, protein sequence and pseudo gene matches of the TALEs in the genome. These data may be used to further analyze the predicted TALEs. GFF and Genbank are standard file formats, which may also be imported into other programs like genome browsers.

2) TALE Analysis

The **TALE Analysis** tool uses as input a set of complete TALE DNA sequences and splits each of these into the N-terminal region, the individual repeats, and the C-terminal region. Splitting the TALEs into repeats is especially useful to distinguish standard repeats from aberrant, short or long, repeats. It also helps to identify those codons in a repeat that code for

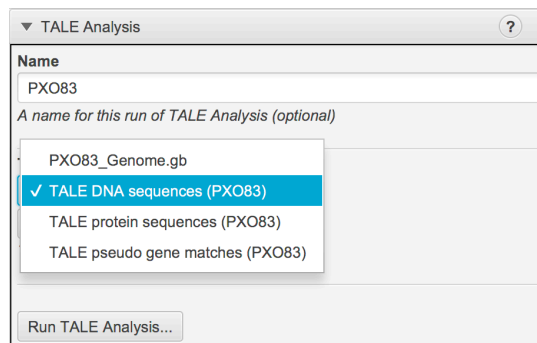


Figure 6: TALE Analysis input mask

its RVD and is, consequently, used to determine the sequence of RVDs for each of the input TALEs. To analyze the TALEs, select the created data set **TALE DNA sequences** created by the **TALE prediction** tool, and click the **Run TALE Analysis...** button (Fig. 6).

The program splits the TALEs of the strain into parts, and creates lists with these parts as a DNA and protein sequence. In addition, the tool automatically extracts the RVD sequence of each TALE (Fig. 7). Further tools in the AnnoTALE pipeline may use the **TALE DNA parts** or **TALE Protein parts** output of the **TALE Analysis** tool.

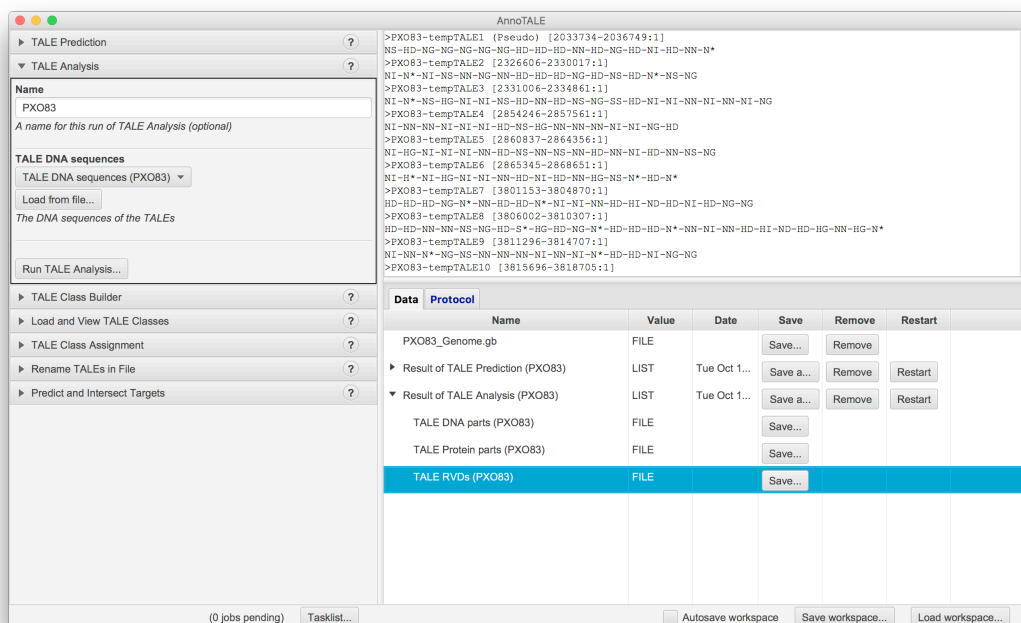


Figure 7: Result of TALE Analysis – TALE RVD sequences

3) TALE Class Builder (optional)

For routine use you may skip this step and proceed to the next tool - **Load and View Classes**. This step is *only* necessary if *no* classification of TALEs exists or to build a new hierarchical classification of TALEs for a specific application.

The tool **TALE Class Builder** is used to build a hierarchy of a given set of TALEs and group them together in classes. It can be used to create a classification of TALEs, e.g. from different pathovars of *Xanthomonas spp.* The tool groups the TALEs into classes on the basis of an RVD sequence comparison. The tool determines a mismatch score from the pairwise alignment of each pair of two TALEs RVD sequences. Afterwards the TALEs are grouped together such that the average mismatch score in a class does not exceed the user-specified threshold (for detailed information see Grau *et al.* 2015). This class definition can be used as a basis for assigning new TALEs to one of these classes using the tool **TALE Class Assignment**.

To create your own TALE classes load TALE DNA or protein sequences from a local FastA file or from the output of the [TALE Prediction](#) tool or use the previously generated [TALE DNA parts](#) or [TALE Protein parts](#) from the “TALE Analysis” tool and click on the [Run TALE Class Builder...](#) button. If creating a new classification, e.g. for a different pathovar, it is possible to use the default settings for cutoff (5.0) and significance level (0.01) or to customize them depending on the specific application. After finishing the classification the [Class builder](#) file, a tree of all classes, as well as a report for the individual classes are shown in the data panel (Fig. 8).

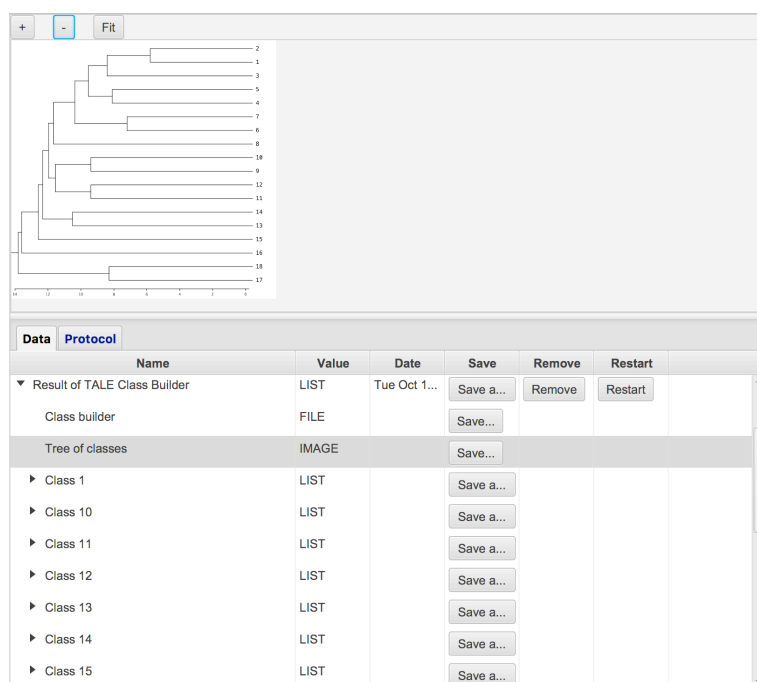


Figure 8: Result of TALE Class Builder – Tree of classes

In the class report of every individual class, information is collected, e.g., the name of the class, RVD sequences of the TALEs in this class, the most likely common binding sequence, the alignment scores, the related classes with significant matches and an RVD sequence alignment of the TALEs from these classes (Fig. 9).

```

Class 1 for (AffineCosts, 5.0, SEMI_GLOBAL)
distance: ==
significance: p=0E0

NI HG NI NI NS HD NN HD HD NS N* N* HD HD NS NS NN NI NG NN NI N* NS N* PX083-tempTALE18

Most likely common binding sequence:
TA T A A A C G C C C A C C C C A A G G A T G A C A C

Class tree:
(PX083-tempTALE18)

Alignment scores:

Related classes with significant matches:
Class 15 with members
PX083-tempTALE19 related to PX083-tempTALE18 with score 14.5 (p=6,25E-3)

Alignments:
PX083-tempTALE19 vs. PX083-tempTALE18:
NI HG NI NI NS HD NN HD HD NS N* N* HD HD NS NS NN NI NG NN NI N* NS N* --
|| || || || :: || || || || || : || : :: || :: :: || : :: || : :: || :
NI HG NI NI HG HD NN HD HD HD NI NI NN NI HD HD HD HG NN NN HD NS NN HD N* NS N*
Cost: 14.5

```

Figure 9: Result of TALE Class Builder – Class report

4) Load and View TALE Classes

The tool **Load and View TALE Classes** loads a given set and hierarchy of TALE classes, termed **Class builder** into AnnoTALE. Typically, the current definition of TALE classes is downloaded from the server via the **Download current definition** option. Alternatively, a local file can be loaded into AnnoTALE using the **Load from file...** button (Fig. 10).

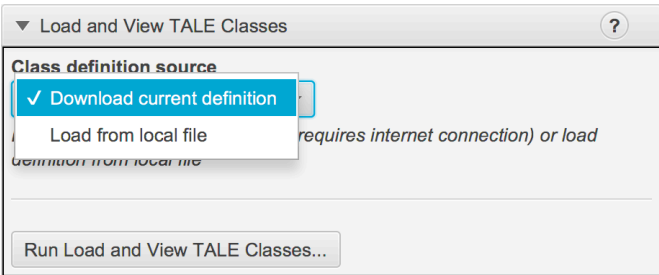


Figure 10: Load and View TALE Classes input mask

The local **Class builder** can be either a class hierarchy that was built using the **TALE Class Builder** tool before or a saved file from a previous AnnoTALE session (e.g. **Augmented Class Builder**, output of the **TALE Class Assignment** tool, see below). After loading, the entry **Result of TALE Class Builder** appears in the data panel, which consists of the **Class builder download** file, the **tree of classes**, and a tree for each single class (Fig. 11).

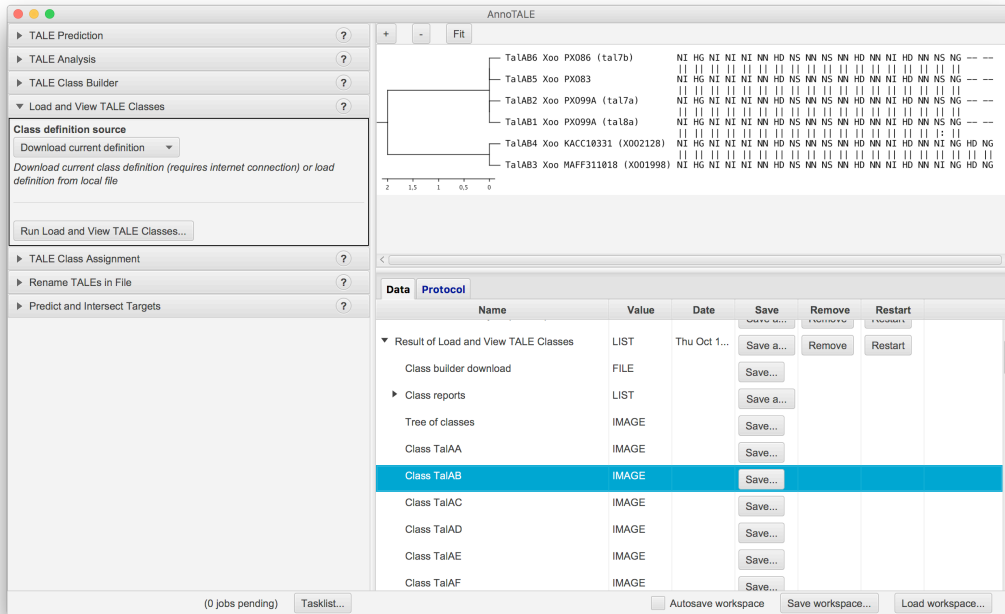


Figure 11: Result of Load and View TALE Classes - overview

It also contains textual [Class reports](#) as well as supplementary files describing the TALEs that are contained in the current class hierarchy and their source strains. The [Class reports](#) contain general information about all TALEs that are included in the [Class builder](#), the TALE protein and DNA sequences, and a list of all TALEs and classes (Fig. 12).

#	ID	Accession	Start	End	Strand	
1	TalAA6 Xoo PXO86 (tal5b)	CP007166.1	2541429	2545185	+1	
2	TalAA5 Xoo PXO83		2305535	2309291	-1	
3	TalAA2 Xoo PXO99A (tal7b)	CP000967.1	2687333	2691088	+1	
4	TalAA1 Xoo PXO99A (tal8b)	CP000967.1	2899420	2903175	+1	
5	TalAA4 Xoo KACC10331 (XOO2276)	AE013598.1	2401988	2405743	+1	
6	TalAA3 Xoo MAFF311018 (XOO2160)	AP008229.1	2387818	2392653	+1	
7	TalAB6 Xoo PXO86 (tal7b)	CP007166.1	2843503	2847022	+1	
8	TalAB5 Xoo PXO83		2860837	2864356	+1	
9	TalAB2 Xoo PXO99A (tal7a)	CP000967.1	2682825	2686343	+1	
10	TalAB1 Xoo PXO99A (tal8a)	CP000967.1	2894912	2898430	+1	
11	TalAB4 Xoo KACC10331 (XOO2128)	AE013598.1	2226672	2230394	-1	
12	TalAB3 Xoo MAFF311018 (XOO1998)	AP008229.1	2205654	2209376	-1	

Name	Value	Date	Save	Remove	Restart
Class reports	LIST		Save a...		
All TALE protein sequences	FILE		Save...		
All TALE DNA sequences	FILE		Save...		
List of TALEs	LIST		Save...		
List of classes	FILE		Save...		
Tree of classes	IMAGE		Save...		
Class TalAA	IMAGE		Save...		
Class TalAB	IMAGE		Save...		
Class TalAC	IMAGE		Save...		
Class TalAD	IMAGE		Save...		

Figure 12: Result of Load and View TALE Classes – List of TALEs

5) TALE Class Assignment

The tool **TALE Class Assignment** assigns a given set of TALEs, e.g., the TALEs that were previously predicted in a *Xanthomonas* genome, to one of the existing TALE classes that were loaded in the previous tool. If no class with sufficient similarity to an individual TALE exists, this TALE is assigned to its own, new class. The assignment into classes is the basis for the systematic nomenclature of TALEs and the **TALE Class Assignment** tool will propose systematic TALE names.

TALE Class Assignment

Class builder
 Class builder download ▾
 Load from file...
TALE class builder definition

TALE sequences
 TALE DNA parts (PXO83) ▾
 Load from file...
The sequences of the TALEs (DNA or protein), or "TALE DNA parts" or "TALE Protein parts" output of "TALE Analysis".

Strain
 PXO83
The name of the strain. (optional)

Accession
 |
The accession number of the genome (if applicable). (optional)

Run TALE Class Assignment...

Figure 13: TALE Class Assignment input mask

As input, the **TALE Class Assignment** tool needs (i) a given set and hierarchy of TALE classes, the **Class builder**, and (ii) a set of TALE sequences (Fig. 13). The TALE sequences may be those generated by the **TALE Prediction** tool, one of the pre-processed **TALE parts** files produced by the **TALE Analysis** tool, or DNA sequences of TALEs stored in a local FastA file. In the latter case, these sequences may now be loaded into AnnoTALE using the **Load from file...** button. Optionally, the user may provide the name of the *Xanthomonas* strain that has been the source of these TALEs and (if available) an accession number, e.g., the accession number of the corresponding genome in NCBI Genbank. If provided, the strain information will be included into the annotation of TALEs with systematic names.

After all parameters have been specified, the assignment is started by clicking on the button **Run TALE Class Assignment...**

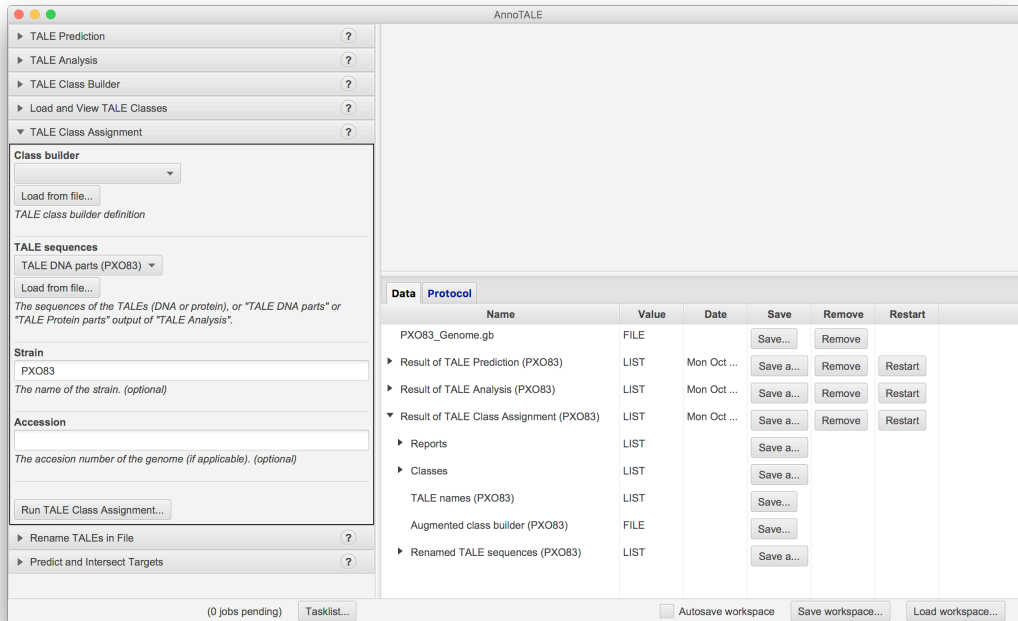


Figure 14: Result of TALE Class Assignment –overview

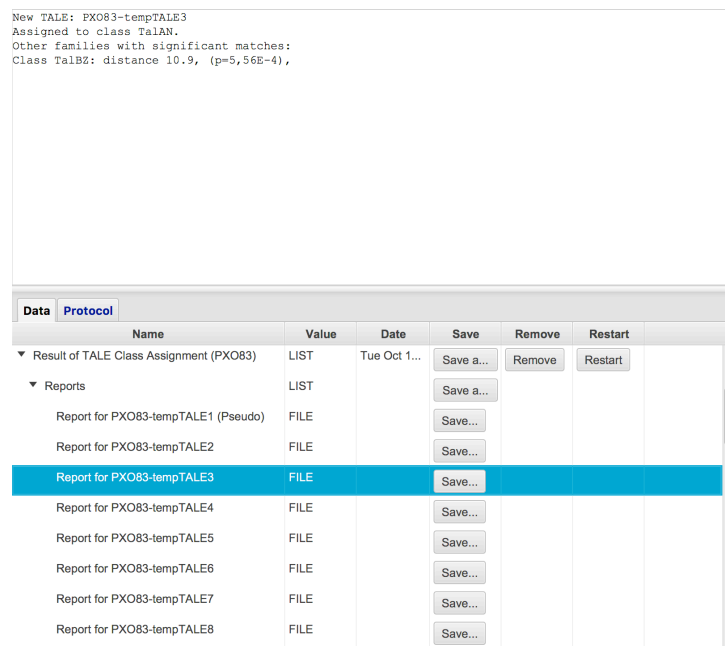


Figure 15: Result of TALE Class Assignment – Class assignment report

In the data panel, section **Result of TALE Class Assignment** reports, classes, TALE names, and Augmented class builder are listed (Fig. 14). The reports give information about the input TALEs in the viewer window, e.g. which class the TALE has been assigned to and possible other families with significant matches (i.e. other families with a similar RVD sequence, Fig. 15). In the section **Classes** all classes are listed that have been modified by the addition of new

TALEs or that have been created, because a TALE did not fit into any of the existing classes. For each of those classes, the tool creates a report and a tree of the members.

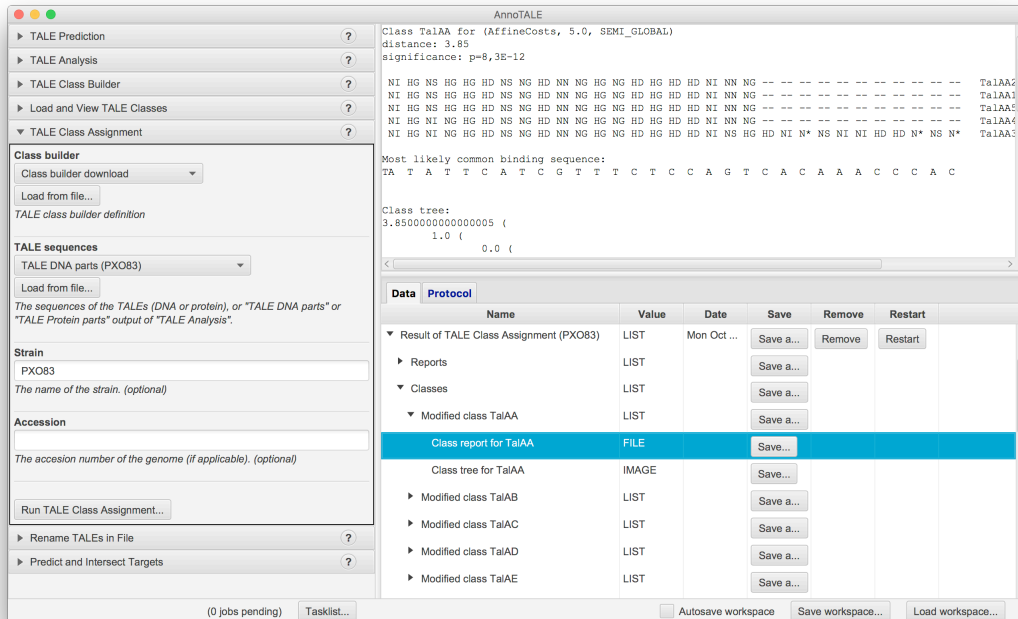


Figure 16: Result of TALE Class Assignment – Class report

The class report (Fig. 16) gives information about the RVD sequences of the TALEs and the alignment score for each TALE pair in the class, the significance of the class assignment and the most likely common binding sequence of all TALEs in this class.

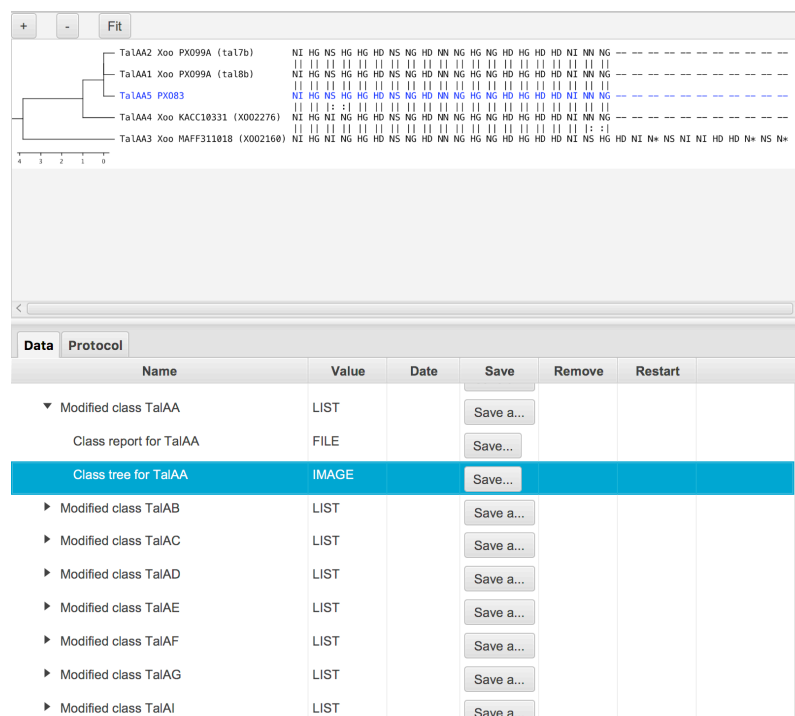
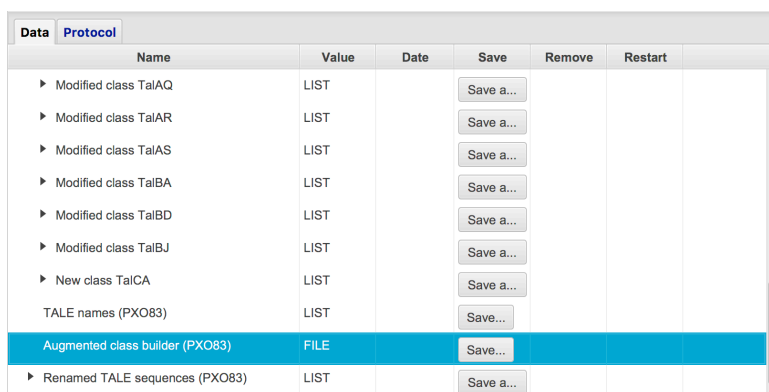


Figure 17: Result of TALE Class Assignment – Class tree

The class tree shows the relationship between the individual TALEs of the class and an RVD alignment of the TALEs. The newly assigned TALE is marked in blue (Fig. 17). Furthermore, the program creates a table, [TALE names](#), which shows the temporary and the proposed systematic name of the TALEs.



Name	Value	Date	Save	Remove	Restart
▶ Modified class TalAQ	LIST		Save a...		
▶ Modified class TalAR	LIST		Save a...		
▶ Modified class TalAS	LIST		Save a...		
▶ Modified class TalBA	LIST		Save a...		
▶ Modified class TalBD	LIST		Save a...		
▶ Modified class TalBJ	LIST		Save a...		
▶ New class TalCA	LIST		Save a...		
TALE names (PXO83)	LIST		Save...		
Augmented class builder (PXO83)	FILE		Save...		
▶ Renamed TALE sequences (PXO83)	LIST		Save a...		

Figure 18: Result of TALE Class Assignment – Augmented class builder

The last file created by the tool is the new class definition termed [Augmented class builder](#) (Fig. 18). This file should be saved and used for further TALE class assignments if TALEs from different *Xanthomonas* strains are added to the class hierarchy successively. To ensure that every TALE gets a unique systematic name it is suggested to send the [Augmented class builder](#) to Jan Grau (grau@informatik.uni-halle.de) for updating the current definition (“Class builder”) on the server. This also ensures that TALE names are reserved, e.g., prior to publication.

6) Rename TALE in File

The tool [Rename TALE in File](#) is used to translate the temporary TALE annotation that has been created by the [TALE Prediction](#) tool in the *Xanthomonas* genome into the new systematic names produced by the [TALE Class Assignment](#) tool.

To rename the TALEs in the GFF or Genbank files from the [TALE Prediction](#) tool specify the [TALE names](#) file created by the [TALE Class Assignment](#) tool and the GFF or Genbank file created by the [TALE Prediction](#) tool, e.g., [Genbank: TALE Predictions](#) and run the process by clicking on [Run Rename TALEs in File](#). The output is either a GFF or Genbank file, depending on the format of the input file, with the annotated *TALE* genes being renamed from temporary to systematic names. This GFF or Genbank file can be saved and used for further analysis in a genome browser (Fig. 19).

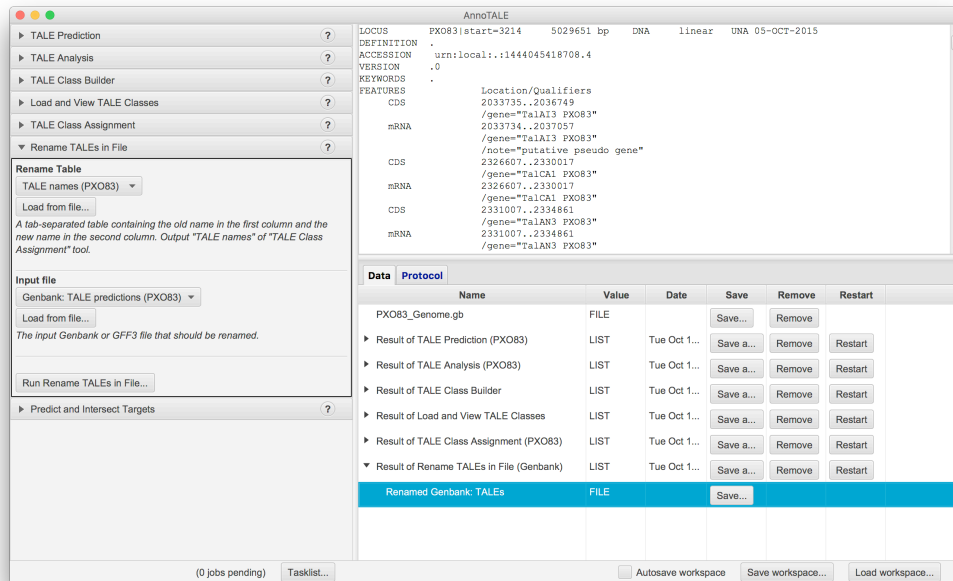


Figure 19: Result of Rename TALEs in File

7) Predict and Intersect Targets

The tool [Predict and Intersect Targets](#) is used to facilitate an initial analysis of putative target genes of all the individual TALEs from a *Xanthomonas* strain and putative common target genes of a class of TALEs. The tool is based on the statistical model of TALgetter (Grau et al. 2013), but is limited to the top 100 target site predictions and does not calculate p-values for single target sites.

For a detailed analysis of putative target sites it is advisable to use TALgetter available at <http://www.jstacs.de/index.php/TALgetter> (Command line application) and <http://galaxy.informatik.uni-halle.de> (Web application).

For predicting possible target sites of TALEs of a *Xanthomonas* strain in a given sequence e.g., the host plant genome or promoterome, it is necessary to load the sequence, that should be analyzed, from a local FastA file via the [Load from file...](#) button in the program. For this example the promoterome of rice (*Oryza sativa*) is used.

Predictions can be done for [TALEs in FastA](#) or for the [Class Builder](#) itself (i.e., all TALEs). Using TALEs in FastA, it is advisable to load [Renamed TALE DNA sequences](#) or [Renamed TALE protein sequences](#), because in these data sets the TALEs have the systematic names instead of temporary names. After loading all necessary data click the [Run Predict and Intersect Targets...](#) button (Figs. 20 and 21).

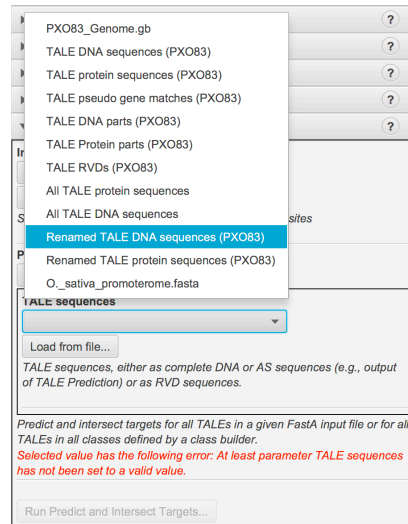


Figure 20: Predict and Intersect Targets input mask, choose renamed TALE sequences

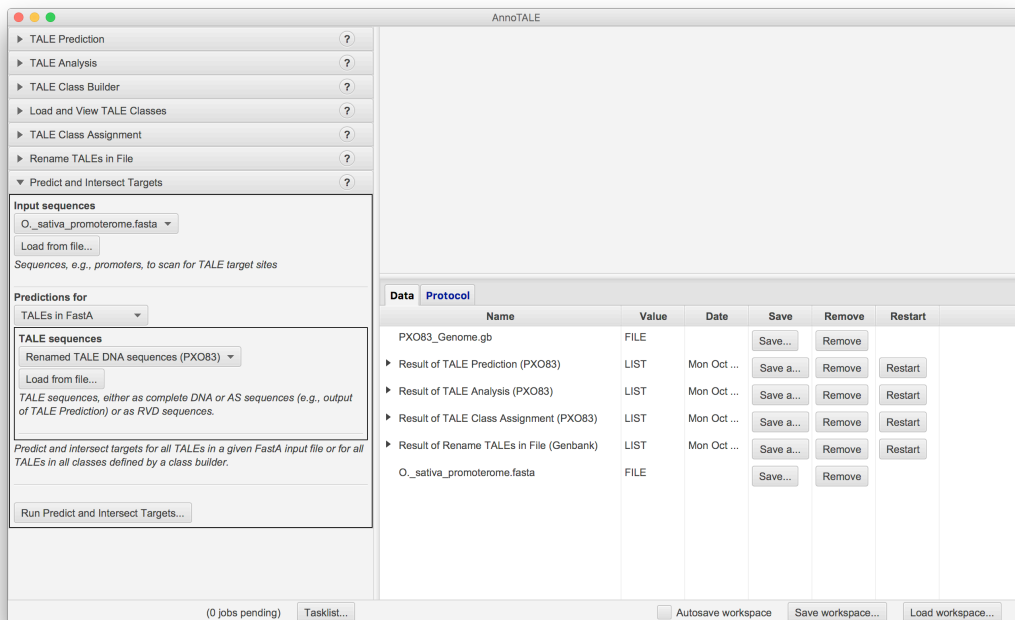


Figure 21: Predict and Intersect Targets – input mask

The tool generates a data set **Overlapping target sites**, where putative target sites, which may be predicted to be targeted by more than one TALE, are listed (Fig. 22). The table in the viewer shows the Sequence ID, the intersection size (i.e. the number of different TALEs which are predicted to target this sequence) and all TALEs with rank and position of the target sites indicated as tuple (**rank, position**).

Furthermore, for every single TALE, predictions of the top 100 putative target sites are generated and displayed in the data panel and viewer, respectively. After selecting the predictions for a TALE a table of the putative targets is shown in the viewer (Fig. 23). The table displays sequence ID and annotation (if available in the input FastA file), the position of the target site in the sequence, the prediction score, the sequence of the target site, and the

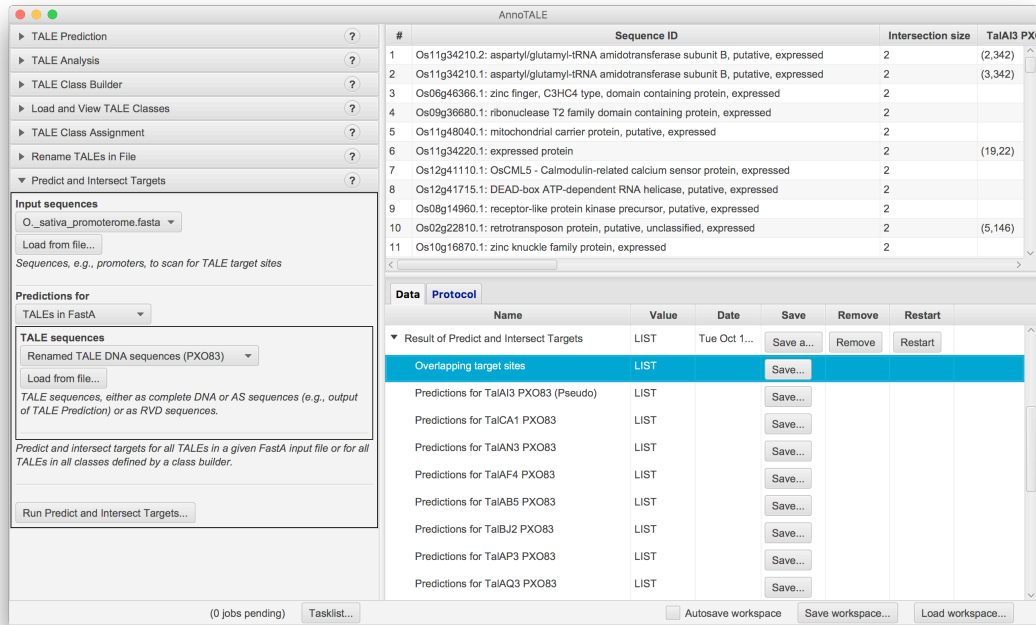


Figure 22: Result of Predict and Intersect Targets – overlapping target sites

match string, which indicates matching RVD-base combinations (M for initial T; l), non perfect matches (:), and mismatches (m for initial T; x).

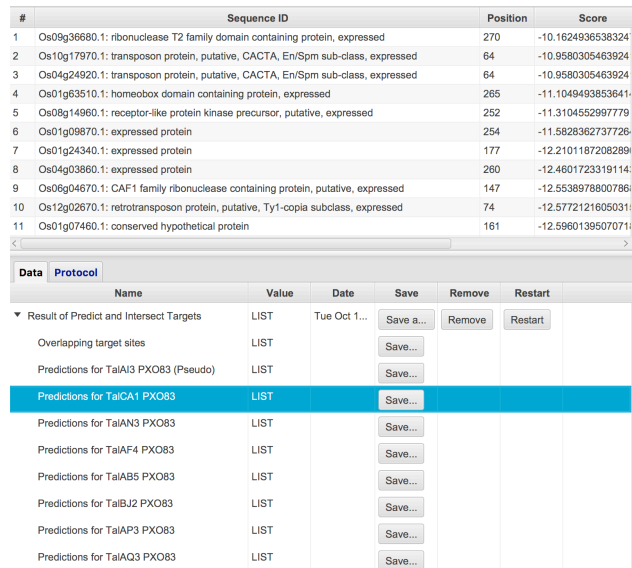


Figure 23: Result of Predict and Intersect Targets – target predictions

For a more global view on TALE targets the full [class builder](#) can be used in the [Predictions for](#) selector. In this case, the prediction is performed for every TALE that is represented in the class builder. Intersections between the predicted sets of targets are determined for all TALEs in common families, which helps to identify putative common targets (and, hence, functions) of TALEs in different *Xanthomonas* strains. Predictions are grouped by the corresponding class.

Quick Start Guide

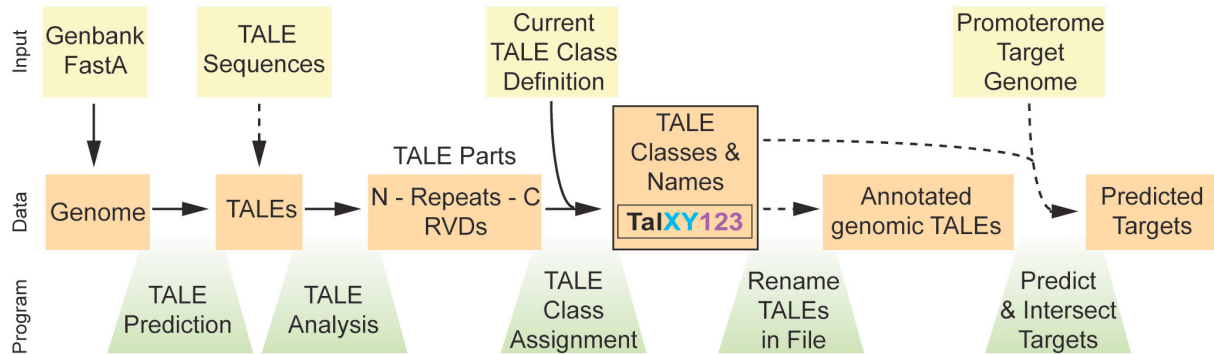


Figure 24: Schematic overview of input parameters, workflow, and program parts.

1) TALE Prediction

Input: Genome file as FastA or Genbank

Output: GFF/Genbank files with annotated TALEs (temporary name); DNA and protein sequences of the TALEs

2) TALE Analysis

Input: TALE DNA or protein sequences from [TALE Prediction](#) or as FastA from file

Output: TALE DNA and protein parts, RVD sequences of all TALEs

3) TALE Class Builder

Input: TALE sequences as FastA, TALE DNA or protein sequences from [TALE Analysis](#)

Output: hierarchy and classification of TALEs, Class Builder file, report and tree of the build classes

4) Load and View TALE Classes

Input: [Download current definition](#) or [Class builder](#) from [TALE Class Builder](#) or [Augmented Class builder](#) (can also be loaded from local file)

Output: Class builder download, report and tree of the pre-defined classes

5) *TALE Class Assignment*

Input: Class builder download or [Class builder](#) from [TALE Class Builder](#) or [Augmented class builder](#) and TALE sequences as FastA, TALE DNA or protein parts from [TALE Analysis](#)

Output: reports for single TALEs; Modified Classes (report + tree); TALE names; Augmented class builder (for further assignments)

6) *Rename TALEs in File*

Input: [TALE names](#) of [TALE Class Assignment](#) or tab separated table and GFF/Genbank file with annotations of [TALE Prediction](#) tool

Output: renamed GFF/Genbank file with annotated TALEs with new systematic name

7) *Predict and Intersect Targets*

Input: sequences to scan, i.e. genome, promoterome as FastA file

Output: predictions of 100 top target sites for all TALEs as table, overlapping target sites of all TALEs

References

Grau, J. *et al.* Computational Predictions Provide Insights into the Biology of TAL Effector Target Sites. *PLoS Comput Biol* **9**, (2013).

Grau, J. *et al.* AnnoTALE: bioinformatics tools for identification, annotation, and nomenclature of TALEs from *Xanthomonas* genomic sequences. (2015)

Download and Installation

AnnoTALE is available at <http://www.jstacs.de/index.php/AnnoTALE> as (i) a runnable Jar file, (ii) a DMG for installation under Mac OS X, and (iii) a Windows installer.

Installation using the runnable Jar file

The runnable Jar file is the preferred version of AnnoTALE if you already have a current version of Java (Java8, update 45 or later) installed on your computer. In this case, no specific installation is required. On each of the three major operating systems (Windows, Linux, Mac OS X), you can just download this Jar file, copy it to any location on your computer you consider appropriate and start AnnoTALE by double-clicking the Jar. If you need to start AnnoTALE with more than the default memory, you need to open a Terminal (or Windows

Command prompt), navigate to the directory containing the Jar file, and start AnnoTALE with the larger memory limits, e.g.

```
java -Xms512M -Xmx6G -jar AnnoTALE-1.0.jar
```

for allowing AnnoTALE to use 512 MB initially and at most 6GB of RAM.

Installation using the Windows installer

The windows installer comes in three versions with different memory requirements. Depending on the main memory (RAM) that is available on your computer, you should choose the largest version (1GB, 2GB, 6GB) that still fits into your computer's RAM. All standard analysis tasks of AnnoTALE may be executed even with the 1GB version, but large workspaces (e.g., when analyzing several genomes in one AnnoTALE run) may lead to a less responsive GUI or even "out of memory" errors. In the latter case, it may be necessary to restart AnnoTALE.

The windows installer contains an appropriate version of Java in addition to AnnoTALE itself. This version of Java is installed together with AnnoTALE and should not interfere with another Java version already installed on your computer.

After you downloaded the windows installer of choice, start the installation process (Fig. 25). After the installation has finished, you find AnnoTALE in your list of Apps.

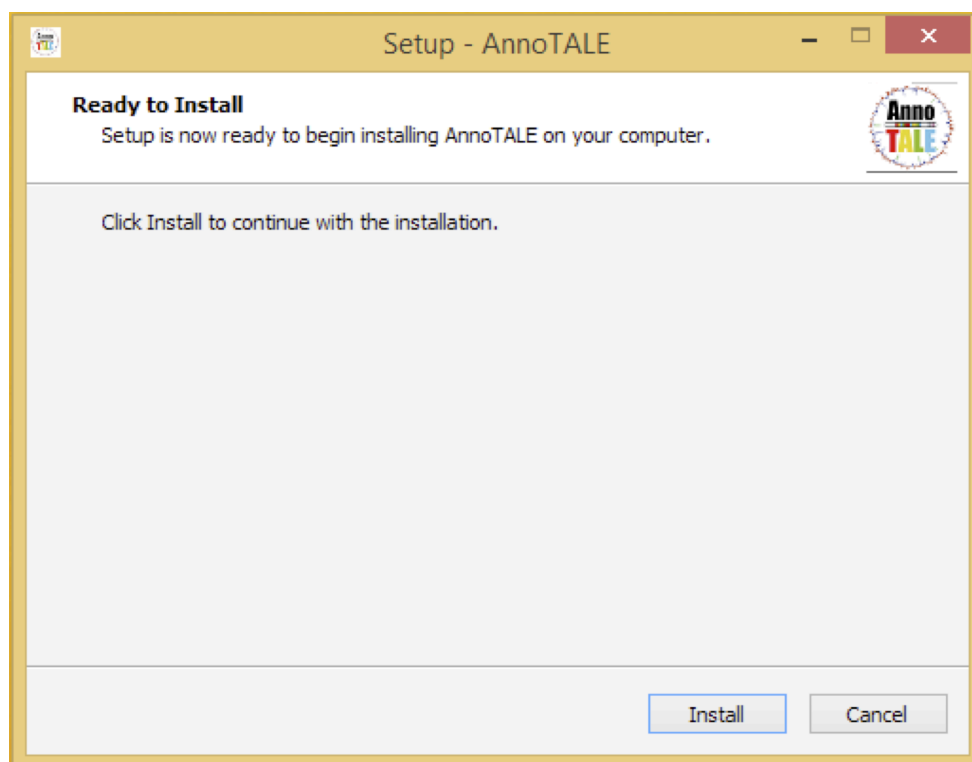


Figure 25: Installation of AnnoTALE using the Windows installer.

Installation from the OS X DMG file

The pre-packaged OS X version of AnnoTALE comes in two version with different memory requirements. Depending on the main memory (RAM) that is available on your computer, you should choose the largest version (2GB, 6GB) that still fits into your computer's RAM. All standard analysis tasks of AnnoTALE may be executed even with the 1GB version, but large workspaces (e.g., when analyzing several genomes in one AnnoTALE run) may lead to a less responsive GUI or even "out of memory" errors. In the latter case, it may be necessary to restart AnnoTALE.

The OS X DMG file contains the AnnoTALE application (Fig. 26), which may be copied to your Applications folder or any other location that you consider appropriate. Afterwards, start AnnoTALE by double-clicking on the AnnoTALE application.

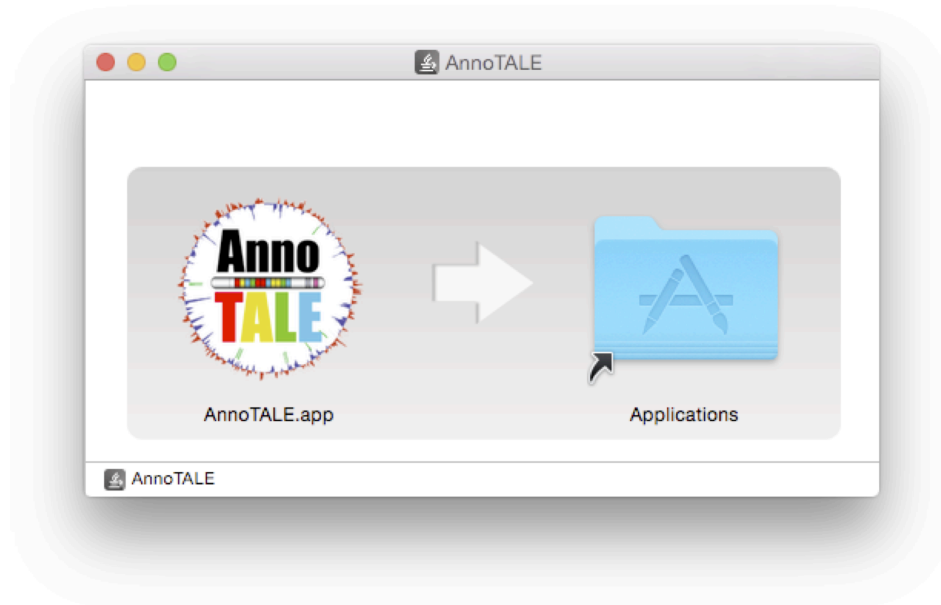


Figure 26: Installation of AnnoTALE for Mac OS X.