

# **User Guide**

**for**

**T3SEpp**

v1.0

02/03/2020

## Table of Content

1. Introduction .....	3
2. Manual of T3SEpp .....	3
3. Manual of T3SEppML .....	7
4. Manual of T3SEdl .....	10
5. Manual of flBlast.....	12
6. Manual of sigHMM.....	14
7. Manual of effectHMM .....	16

## 1. Introduction

This manual was prepared for the standalone version of T3SEpp (version 1.0) and the related modules. For usage of the T3SEpp webserver, please refer to the HELP page of the T3SEpp website. The link is: <http://www.szu-bioinf.org/T3SEpp>.

This manual demonstrated the usage of T3SEpp and the modules with detailed examples, but did not introduce too much on methodology. For methodological details, parameter optimization and model performance please refer to <http://www.szu-bioinf.org/T3SEpp/modules.html>.

T3SEpp could be run as a whole pipeline. Alternatively, the modules could be used independently. The manual illustrated the usage of T3SEpp in Section 2, and the modules or related models in the remaining sections.

## 2. Manual of T3SEpp

1) **System requirement:** Linux or Mac.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental variable path. Note that in Linux system, the Tensorflow needs to be activated before running T3SEpp (source /root/tensorflow/bin/activate).

Perl 5

R version 3.3 or later

Python3

\*GO (version 1, if source code compilation needed)

BLAST suite version 2.2.30+

HMMER3.1

Numpy 1.18.1

Pandas 0.25.3

Tensorflow 1.3.0

Keras 2.1.2

argparse

### 3) Download and installation of T3SEpp:

The T3SEpp package (T3SEpp.v1.0) for Linux or Mac system could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>.

Decompress the “T3SEpp.tar.gz” and get into ~/T3SEpp/ from terminal. If the pre-compiled version does not work, try to re-compile all the GO scripts in the “codes” sub-folder of ~/T3SEpp/sigHMM/, ~/T3SEpp/cbdHMM/, ~/T3SEpp/effectHMM/, ~/T3SEpp/transHMM/, ~/T3SEpp/flBlast/, ~/T3SEpp/SeqAac/, ~/T3SEpp/SignalP/, ~/T3SEpp/PSORTb/ and ~/T3SEpp/TMHMM/, and replace the binary files in to the “bin” sub-folder of each module with the newly compiled ones, respectively. Compiling GO scripts:

```
$ cd ~/T3SEpp/MODULE_NAME/bin
```

```
$ go build ../codes/xxx.go
```

“xxx.go” is the script in “codes” sub-folder of MODULE\_NAME. Compile all the scripts. Also, compile “T3SEppPrep.go” and “T3SEpp.go” and put the binary result files in the same directory (~/T3SEpp/).

### 4) Input files:

There is a necessary protein sequence file and four optional other input files (promoter sequence file, PSORTb prediction file, SignalP prediction file and TMHMM prediction file). If there is more than one file, the protein or gene name for each file must be consistent, in terms of the total number and identity. The names should not contain a space or an illegal character such as ‘|’, and it is suggested not to use pure numbers as the protein or gene name in case of unexpected errors in different

systems. For the ‘protein sequence file’ and ‘promoter sequence file’, *T3SEpp* requires them to be FASTA-formatted, as exemplified by the demonstrated “test.fa” and “testPromoter.fa” files (Fig 1).

<pre>&gt;T1      protein sequence file - FASTA format VDCSIKNNKKGKNTMEINPIPIGSINILTTQSLTESQSTEEAKIEQ( SYQAKRIISQEPRLSTEFDPFLFNKTERYNARLLKTVFEDTPNT! YQQLWLTPNKQHKLTEEELSTIISKEIKNILIKEQVIKELQTELD' &gt;T2 MGSSHHHHHHSSGLVPRGSHMFLTFPNVAITRDNRIDKLSNDLEI TFKVSFSTTDRAFRRERHIEWQGNAIRLERQLNTGLNVSRG &gt;T3 MLKPICHSGSIKVPYLETDKENAGRTPSSDIQQRNVVEDVPI YGRLFQKQVLAHIHSRCQRDADIIREKALRRISRECGAEIDCALLI SYALGCRPGDLPAYNVGRDSVETKAFELEKLADSPYAPYGQTGGF! &gt;T4 MGHLIPIVLPPhRQLSDVDMLEHIEPTPLFDRVTDSPFLDGLT' MFSNVLSLFSGRALNQGVICSKAVEQNLAALCEGMINDFWVAES' &gt;T5 MPPMNESLKSNDFTLHRQMRQMPLSHFTVEPNAPDYSGIRQSGFF/ HISVQEQVAQAFQALSGLLFSSEDSPIDKWKVTDMARVDQQRVGI RLSEQGIIPGRVPSVDVHPDSWRYISYRNELRSEGGGEMQSQALF</pre>	<pre>&gt;T1      promoter sequence file - FASTA format AGGAACGCTTACGGGCTGTCTTCTCATTTTTAAGCGGCTCTCGG GAGAGCGTGCTTGGCGTGGGGCGTCCCGTCGGCAGCAACTGAAA GCCGTGCGCACCTCGCCGGCTCCTTGTGCCCCCTTCAGAAGAGAT &gt;T2 TCCTGATTATTCAACCACTACAAGGTGCCATGGCAGAAACGCG CGGTCTGTATGTGAGGCTTTAGCCTGTTCTGCGTCTGCCGGCGG TTCCAGTAGCCGTGATTTTGTACCTCCGTTGATGCTCATGAAC &gt;T3 TTGAAAAACGCCCTCAAACCAAGTGTCTTTTTAATCGAAAAATGA AATGTCCACTAATCGATTTCCGGCACAGTACACCATTCCGGCAG GACATTGCCCGTGATTTTATGGAACCGGTACGGCCTACGCCA &gt;T4 TTATTATATTACCTTTGTGTAGCTCTTTTTTCTCTTAGATATC GACTTATGGGGCTAAAAAATTCACGGTAAAGAAGCTTTCTTAAT &gt;T5 AATTCCGATGTCCCCCCTTTTTTTAAACGCCCTGTGAAGG AACAGCGGATTTTATCCAAGTAATTAATGGGAAAAATTTACA</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Fig 1. Input sequence file format - FASTA, name consistency**

The files ‘PSORTb.out.txt’, ‘SignalP.out.txt’ and ‘TMHMM.out.txt’ are optional and generated by submitting the ‘protein sequence file’ to PSORTb 3.0, SignalP 4.1 and TMHMM 2.0 webserver, respectively. The links for the three tools are listed as below:

PSORTb: <https://www.psort.org/psortb/index.html>

SignalP: <https://services.healthtech.dtu.dk/service.php?SignalP-4.1>

TMHMM: <https://services.healthtech.dtu.dk/service.php?TMHMM-2.0>

‘Short Format (tab delimited)’, ‘Short - no graphics’ and ‘One line per protein’ are selected as the output format of PSORTb, SignalP and TMHMM, respectively. The file format for ‘PSORTb.out.txt’, ‘SignalP.out.txt’ and ‘TMHMM.out.txt’ are shown in Fig 2. Once generated, the files are moved into ~/T3SEpp/. For this demonstration, “testPSORTb.txt”, “testSignalP.txt” and “testTMHMM.txt” are the example files of ‘PSORTb.out.txt’, ‘SignalP.out.txt’ and ‘TMHMM.out.txt’ respectively.

PSORTb.out.txt		
T1	Cytoplasmic	8.96
T2	Unknown 2.00	
T3	Cytoplasmic	8.96
T4	Cytoplasmic	8.96
T5	Unknown 2.00	
T6	Cytoplasmic	8.96

SignalP.out.txt											
T1	0.119	67	0.124	34	0.179	10	0.143	0.133	N	0.570	SignalP-noTM
T2	0.141	30	0.123	32	0.164	28	0.092	0.109	N	0.570	SignalP-noTM
T3	0.106	26	0.112	26	0.127	13	0.099	0.106	N	0.570	SignalP-noTM
T4	0.116	66	0.103	33	0.187	13	0.085	0.095	N	0.570	SignalP-noTM
T5	0.164	57	0.119	57	0.160	9	0.100	0.110	N	0.570	SignalP-noTM
T6	0.108	19	0.126	11	0.228	1	0.151	0.138	N	0.570	SignalP-noTM

TMHMM.out.txt					
T1	len=343	ExpAA=2.38	First60=0.17	PredHel=0	Topology=o
T2	len=121	ExpAA=0.14	First60=0.14	PredHel=0	Topology=o
T3	len=374	ExpAA=0.00	First60=0.00	PredHel=0	Topology=o
T4	len=246	ExpAA=0.00	First60=0.00	PredHel=0	Topology=o
T5	len=216	ExpAA=0.00	First60=0.00	PredHel=0	Topology=o
T6	len=523	ExpAA=0.00	First60=0.00	PredHel=0	Topology=o

**Fig 2. Input file format of PSORTb, SignalP and TMHMM prediction**

After the input files have been prepared, get into ~/T3SEpp/ through command line:

```
$ cd ~/T3SEpp/
```

#### 5) Running T3SEpp in one command line:

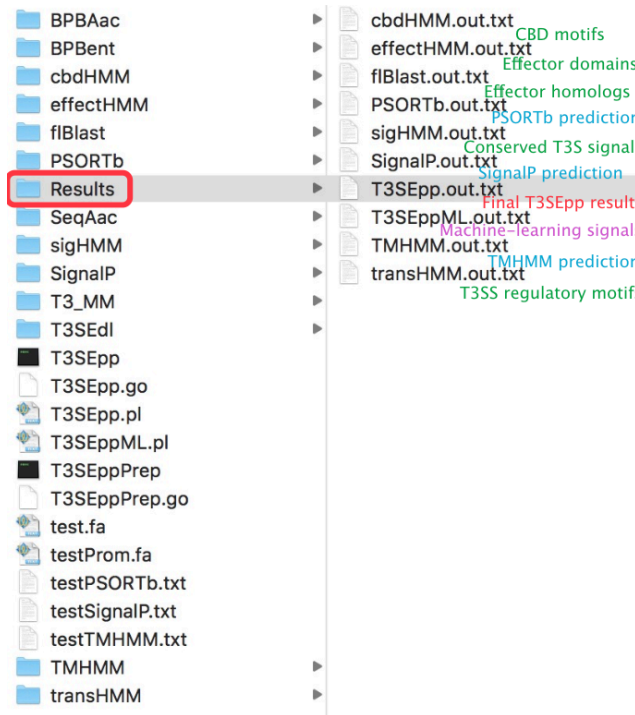
```
$ perl T3SEpp.pl -prot test.fa -prom testProm.fa
    -psortb testPSORTb.txt -signalp testSignalP.txt
    -tmhmm testTMHMM.txt -cutoff 0.2
```

The arguments are explained as below:

-prot	PRPTEIN_SEQUENCE_FILE_TO_BE_PREDICTED	Necessary
-prom	PROMOTER_SEQUENCE_FILE	Optional
-psortb	PSORTb_PREDICTION_FILE	Optional
-signalp	SIGNALP_PREDICTION_FILE	Optional
-tmhmm	TMHMM_PREDICTION_FILE	Optional
-cutoff	CUSTOMIZED_CUTOFF (DEFAULT:0.5)	Optional

#### 6) Output file format:

The prediction results are listed in ~/T3SEpp/Results/. 'T3SEpp.out.txt' gives the overall prediction results of the integrated pipeline, while the other files provide the detailed results from different modules (Fig 3).



**Fig 3. Prediction result files and their details**

‘T3SEpp.out.txt’ summarizes the prediction results of different modules, the general prediction scores and the classification according to the customized or default (0.5) cutoff (Fig 4).

prot	T3SEppML	fiBlast	sigHMM	cbdBHM	effectHMM	transHMM	TMHMM	PSORTb	SignalP	T3SEpp	Pred
T2	0	1	0	1	0	0	1	1	0	0.35	T3S
T3	0	1	0	1	0	1	1	1	0	0.55	T3S
T4	0	1	0	0	0	0	1	1	0	0.28	T3S
T5	0	1	1	0	1	0	1	1	0	0.63	T3S
T6	0	1	1	0	1	0	1	1	0	0.63	T3S
T1	0	1	0	1	0	1	1	1	0	0.55	T3S

**Fig 4. The final prediction result of T3SEpp**

### 3. Manual of T3SEppML

The module *T3SEppML* can also be run independently to screen the proteins with the atypical signal features of T3SEs.

1) **System requirement:** Linux or Mac.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental

variable path. Note that in Linux system, the Tensorflow needs to be activated before running T3SEpp (source /root/tensorflow/bin/activate).

Perl 5

R version 3.3 or later

Python3

\*GO (version 1, if source code compilation needed)

Numpy 1.18.1

Pandas 0.25.3

Tensorflow 1.3.0

Keras 2.1.2

argparse

### 3) Download and installation of T3SEppML:

The *T3SEppML* module for Linux or Mac system could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>.

Decompress the “T3SEppML.tar.gz” and get into ~/T3SEppML/ from terminal. If the pre-compiled version does not work, try to re-compile the GO scripts in ~/T3SEppML/SeqAac/codes/ and replace the binary files in ~/T3SEppML/SeqAac/bin with the newly compiled ones.

Compiling GO scripts:

```
$ cd ~/T3SEppML/SeqAac/bin
```

```
$ go build ../codes/xxx.go
```

“xxx.go” is the script in ~/T3SEppML/SeqAac/codes/. Compile all the scripts.

### 4) Running flBlast in command lines:

```
$ cd ~/T3SEppML/
```

```
$ perl T3SEppML.pl test.fa
```

Input file should be a FASTA-formatted protein sequence file, as exemplified by “test.fa”. The protein name should neither be separated by space nor be with illegal characters such as "|" or purely numbers.



### 5) Output file format:

A single prediction result file will be generated named as “T3SEppML.out.txt”. The format of the prediction result file is shown in Fig 5, containing 3 columns, protein ID, the scores of six models, and the classification with default cutoff of each model, where “1” means “T3S signal” and “0” means “non-T3S signal”.

prot	SeqAac BPBAac BPBent T3_MM T3SEdnn T3SErnn	score	SeqAac BPBAac BPBent T3_MM T3SEdnn T3SErnn
T1	1 0.25 0.20 0.20 0.01 0.97		1 1 1 1 0 1
T2	0 0.31 0.09 0.07 1.00 0.97		0 1 1 1 1 1
T3	1 0.74 0.40 -0.01 1.00 0.21		1 1 1 0 1 0
T4	1 0.23 0.34 -0.02 0.01 0.14		1 1 1 0 0 0
T5	1 0.17 0.27 -0.01 0.01 0.04		1 1 1 0 0 0
T6	1 0.13 0.21 0.09 0.94 0.06		1 1 1 1 1 0

**Fig 5. T3SEppML prediction result file format**

Except for the T3SEd1, we do not provide independent manual for the other machine learning models in this version of UserGuide. However, they can still be run independently.

## 4. Manual of T3SEdl

The module *T3SEdl* can also be run independently to screen the proteins with the atypical signal features of T3SEs with two deep-learning models: *T3SEdnn* and *T3SErnn*.

1) **System requirement:** Linux / Mac / Windows.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental variable path. Note that in Linux system, the Tensorflow needs to be activated before running T3SEpp (source /root/tensorflow/bin/activate).

Python3

Numpy 1.18.1

Pandas 0.25.3

Tensorflow 1.3.0

Keras 2.1.2

argparse

3) **Download and installation of T3SEdl:**

The *T3SEdl* module could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>. Decompress the “T3SEdl.tar.gz” and get into ~/T3SEdl/ from terminal.

4) **Running T3SEdl in command lines:**

i) Predict with T3SEdnn:

```
$ python3 T3SEdl.py -t test.fa -m DNN
```

ii) Predict with T3SEdnn:

```
$ python3 T3SEdl.py -t test.fa -m CNN_LSTM
```

Input file should be a FASTA-formatted protein sequence file, as exemplified by “test.fa”, with each protein sequence 100-aa length (N-terminal 2-101 amino acids from each candidate protein) in one line. The protein name should neither be separated by space nor be with illegal characters such as “|” or purely numbers.

#### 5) Output file format:

Either ‘DNN’ or ‘CNN\_LSTM’ mode generates a “result.csv” file, curating the *T3SEdnn* or *T3SErnn* prediction results, respectively. The result file contains three columns: ID – the name of protein to be predicted, Score and Class (‘1’ – T3S signal or ‘0’ – non-T3S signal). The default cutoff for classification is set as ‘0.5’, i.e.,  $\geq 0.5$  – T3S signal,  $< 0.5$  – non-T3S signal.

## 5. Manual of flBlast

The module *flBlast* can also be run independently to screen the proteins with homology to validated T3SEs.

1) **System requirement:** Linux or Mac or Windows.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental variable path.

\*GO (version 1, if source code compilation needed)

BLAST suite version 2.2.30+

3) **Download and installation of flBlast:**

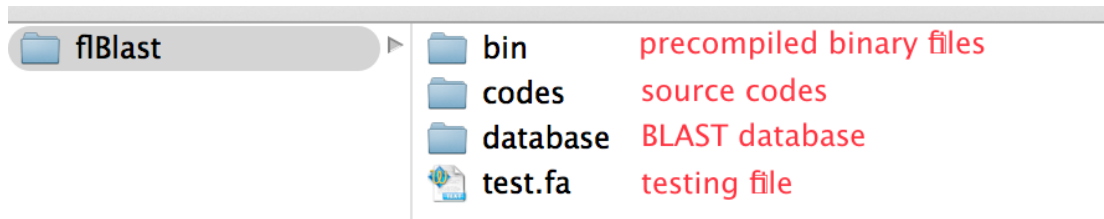
The *flBlast* module for Linux or Mac system could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>.

Decompress the “flBlast.tar.gz” and get into `~/flBlast/` from terminal. There are three sub-folders and a testing file (Fig 6). The flBlast scripts have been precompiled and put in `~/flBlast/bin/`. Users can also compile the source code in `~/flBlast/codes/` and place the binary file in `~/flBlast/bin/`:

```
$ cd ~/flBlast/bin
```

```
$ go build ../codes/xxx.go
```

“xxx.go” is the script in `~/flBlast/codes/`. Compile all the scripts.



**Fig 6. The folder organization of *flBlast* module**

#### 4) Running flBlast in command lines:

```
$ cd ~/flBlast/
$ blastp -query test.fa
  -db ./database/Validated_effectors_20160311 -evalue 1e-5
  -outfmt 6 -out blast.1.txt
$ ./bin/bestpicker blast.1.txt >blast.2.txt
$ ./bin/simcovfilter ./database/Validated_effectors_20160311.length
  ./database/Validated_effectors_20160311.family
  ./database/Validated_effectors_20160311.function blast.2.txt
>flBlast.out.txt
```

Input file should be a FASTA-formatted protein sequence file, as exemplified by “test.fa”. The protein name should neither be separated by space nor be with illegal characters such as “|” or purely numbers.

#### 5) Output file format:

Three files were output and the “flBlast.out.txt” file designated by user contains the final prediction results. The result file has eight columns, representing candidate protein name, the most significantly homologous hit, effector family, functional group, length coverage and similarity for the covered regions (Fig 7).

Query	Subject	Subject_Acc	Family	Function	Coverage	Similarity
T5	<a href="#">Sal_SPI12_SpvC</a>	CCF76774.1	0SPF	Substrate	1.00	0.73
T14	<a href="#">Pse_Pse2_HopAS1</a>	AAZ37064.1	<a href="#">Pse_Pse2_HopAS1</a>	Substrate	1.00	1.00
T15	<a href="#">Pse_Pse2_HopD1</a>	NP_790715.1	XOPB	Substrate	1.00	0.94
T16	<a href="#">Xan_Xan_XopE2</a>	CAJ23957.1	XOPE	Substrate	1.00	0.81
T17	<a href="#">Pse_Pse2_HopAF1</a>	NP_791393.1	<a href="#">Pse_Pse2_HopAF1</a>	Substrate	1.00	1.00
T18	<a href="#">Pse_Pse2_HopAK1</a>	NP_793862.1	<a href="#">Pse_Pse2_HopAK1</a>	Translocon	1.00	0.82
T22	<a href="#">Xan_Xan_XopAL1</a>	AEL06398.1	XOPE	Substrate	0.85	0.35
T23	<a href="#">Erw_Hrp_AvrRpt2</a>	CBA23177.1	AVRPT2	Substrate	1.00	0.99
T24	<a href="#">Bor_Bor_BspR</a>	NP_888184.1	<a href="#">Bor_Bor_BspR</a>	Substrate	0.85	0.98
T26	<a href="#">Shi_Shi_IpaH9.8</a>	YP_406100.1	YOPM	Substrate	0.98	0.75
T27	<a href="#">Shi_Shi_IpaH7.8</a>	YP_406039.1	YOPM	Substrate	1.00	0.63
T28	<a href="#">Bra_Bra_NopM</a>	NP_768544.1	YOPM	Substrate	0.90	0.48
T29	<a href="#">Rhi_Rhi_NopT</a>	NP_444174.1	YOPT	Substrate	1.00	1.00

**Fig 7. flBlast prediction result file format**

## 6. Manual of sigHMM

The module *sigHMM* can also be run independently to screen the proteins with conserved T3S signal sequences.

1) **System requirement:** Linux or Mac.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental variable path.

\*GO (version 1, if source code compilation needed)

HMMER3.1

3) **Download and installation of sigHMM:**

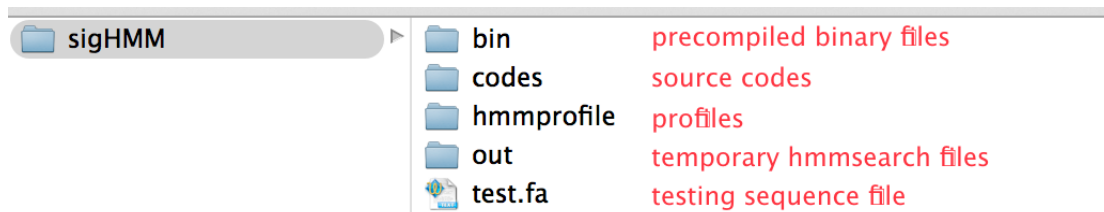
The *sigHMM* module for Linux or Mac system could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>.

Decompress the “sigHMM.tar.gz” and get into `~/sigHMM/` from terminal. There are four sub-folders and a testing file (Fig 8). The sigHMM script has been precompiled and put in `~/sigHMM/bin/`. Users can also compile the source code in `~/sigHMM/codes/` and place the binary file in `~/sigHMM/bin/`:

```
$ cd ~/sigHMM/bin
```

```
$ go build ../codes/xxx.go
```

“xxx.go” is the script in `~/sigHMM/codes/`. Compile all the scripts.



**Fig 8. The folder organization of *sigHMM* module**

#### 4) Running effectHMM in one command line:

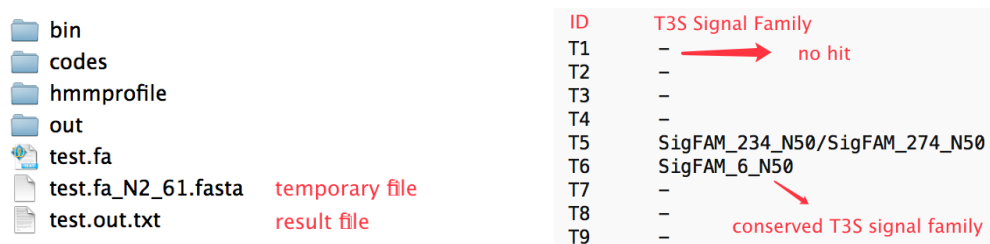
```
$ cd ~/sigHMM/
```

```
$/bin/sigHMM hmmsearch ./hmmprofile test.fa out >test.out.txt
```

Input file should be a FASTA-formatted protein sequence file, as exemplified by “test.fa”. The protein name should neither be separated by space nor be with illegal characters such as “|” or purely numbers.

#### 5) Output file format:

The predictions were output into the result file with self-designated name, e.g., “test.out.txt” in this example (Fig 9, left), with two columns (Fig 9, right). The first column lists the names of proteins to be predicted, and the second column gives the prediction results: ‘-’ represents no hit while ‘SigFAM\_NUMBER\_N50’ indicates the T3S signal family profile that the predicted protein contains.



**Fig 9. *sigHMM* prediction results and output file format**

## 7. Manual of effectHMM

The module *effectHMM* can also be run independently to screen the proteins with known effector domains.

1) **System requirement:** Linux or Mac.

2) **Software or package prerequisites:** The following software or packages should be pre-installed and configured into environmental variable path.

\*GO (version 1, if source code compilation needed)

HMMER3.1

3) **Download and installation of effectHMM:**

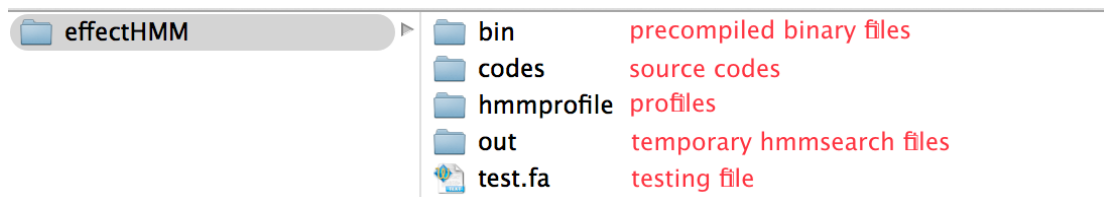
The *effectHMM* module for Linux or Mac system could be downloaded from the website: <http://www.szu-bioinf.org/T3SEpp>.

Decompress the “effectHMM.tar.gz” and get into `~/effectHMM/` from terminal. There are four sub-folders and a testing file (Fig 10). The effectHMM script has been precompiled and put in `~/effectHMM/bin/`. Users can also compile the source code in `~/effectHMM/codes/` and place the binary file in `~/effectHMM/bin/`:

```
$ cd ~/effectHMM/bin
```

```
$ go build ../codes/xxx.go
```

“xxx.go” is the script in `~/effectHMM/codes/`. Compile all the scripts.



**Fig 10. The folder organization of *effectHMM* module**



#### 4) Running effectHMM in one command line:

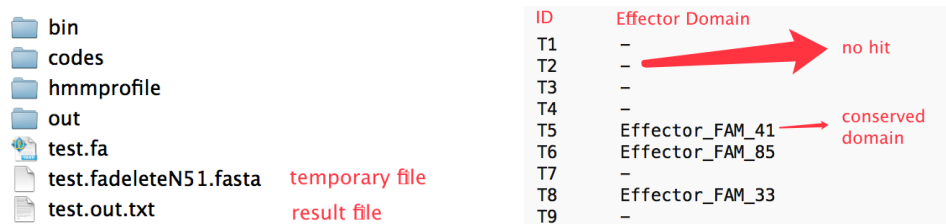
```
$ cd ~/effectHMM/
```

```
$ ./bin/effectHMM hmmsearch ./hmmprofile test.fa out >test.out.txt
```

Input file should be a FASTA-formatted protein sequence file, as exemplified by “test.fa”. The protein name should neither be separated by space nor be with illegal characters such as “|” or purely numbers.

#### 5) Output file format:

The predictions were output into the result file with self-designated name, e.g., “test.out.txt” in this example (Fig 11, left), with two columns (Fig 11, right). The first column lists the names of proteins to be predicted, and the second column gives the prediction results: ‘-’ represents no hit while ‘Effector\_FAM\_NUMBER’ indicates the effector domain family that the predicted protein belongs to.



The figure shows a file explorer on the left and a table of prediction results on the right. The file explorer shows a directory with files: bin, codes, hmmprofile, out, test.fa, test.fadeleteN51.fasta (temporary file), and test.out.txt (result file). The table has two columns: ID and Effector Domain. The rows are T1 to T9. T1, T2, T3, T4, T7, and T9 have a '-' in the Effector Domain column. T5 has 'Effector\_FAM\_41' and T6 has 'Effector\_FAM\_85'. Red arrows point from the '-' in T2 to 'no hit' and from 'Effector\_FAM\_41' in T5 to 'conserved domain'.

ID	Effector Domain
T1	-
T2	-
T3	-
T4	-
T5	Effector_FAM_41
T6	Effector_FAM_85
T7	-
T8	Effector_FAM_33
T9	-

**Fig 11. effectHMM prediction results and output file format**