

Overview

The software package HiFiX implements the novel algorithm for **High Fidelity Clustering of Sequences**.

To tackle the challenge of finding proteins families, we model the all-against-all comparison between proteins by a similarity network.

HiFiX combines **network topology** analysis where sequence similarity is seen as a social relationship, **multiple alignment** quality measurement and model **selection theory**.

HiFiX supports **multiprocessing** and is therefore fast enough to be used in practice on large amounts of data.

HiFiX is already included in the experimental pipeline for

[HOGENOM](#) ↗

.



LICENCE

SiLiX is licensed under the [General Public License](#) ↗

DOWNLOAD

You can download the latest version [HERE \(ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/\)](ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/)

SYSTEM REQUIREMENTS & DEPENDENCIES

HiFiX was succesfully tested on Linux (Ubuntu up to 14.04, Debian Wheezy 7.0) and Mac OSX (Darwin 10.5.8)

*HiFiX does **not work with Python 3.x**, **Python 2.X required Python modules** (already installed by default in Python >=2.7):*

- > NumPy
- > [argparse](#) ↗
- > [subprocess](#) ↗
- > [multiprocessing](#) ↗

and

- > [BioPython](#) ↗

(for local install, set the `PYTHONPATH` variable to `/path_to/module`)

Auxiliary software packages (installation required)

> [SiLiX](#)

(**version >= 1.2.10**)

> [MAFFT](#) 

> [HMMER3](#) 

> [Louvain](#) 

: for this program, there is no proper install procedure. First uncompress this

[archive\(ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/Community_BGLL_CPP/Community_BGLL_CPP.zip\)](ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/Community_BGLL_CPP/Community_BGLL_CPP.zip)

and go to the *Community_BGLL_CPP* directory. Compile with *make* (NB: on MacOS only, create a dummy file with *touch malloc.h*, and compile with *make CFLAGS+=-l.*). Then we suggest three solutions:

- > [RECOMMENDED] copy manually the binaries *community*, *convert* and *hierarchy* in your PATH (in */usr/local/bin* for example)
- > add the */path_to/Community_BGLL_CPP* directory to your PATH (export *PATH=\$PATH :/path_to/Community_BGLL_CPP*)
- > [EASY] pass the */path_to/Community_BGLL_CPP* directory to HiFiX by command line (in this way, the above test procedure will fail... don't panic)

INSTALLATION

Compilation and installation are compliant with the Python standard procedure

```
tar zxvf hifix-x.x.x.tar.gz
cd hifix-x.x.x
python setup.py test
python setup.py install
```

(for local install, add the flag *--prefix /path_to/hifix* and set *PYTHONPATH= /path_to/hifix/lib/python2.x/site-packages*)

PROGRAMS USE

(see also Tutorial section)

➤ **THE EASY WAY** (most users):

To get information or help:

```
hifix --help
```

The user provides a fasta file FASTAFILE (***less than 256 characters per line*** and no "U" character [to be replaced by "X"]), the result file of

[SiLiX](#)

clustering FNODESFILE and the network file NETFILE obtained with SiLiX :

```
hifix <FASTAFILE> <NETFILE> <FNODESFILE>
```

If using a multiprocessor machine with NBPROCS cores, it is valuable to specify the option :

```
hifix -t <NBPROCS> ...
```

If you have a /dev/shm directory (shared memory), it is recommended to use it (gains of performance) :

```
hifix -d /dev/shm ...
```

OUTPUT FORMAT

List of pairs in format "family_id sequence_id" where :

- sequence_id are those of FASTAFILE
- family_id are build from the prefix of FASTAFILE (prefix.fasta) followed by a unique tag such as
 - "_i_j" if family j was deduced from SiLiX pre-family i
 - "_i" if SiLiX pre-family i was conserved

Exple:

```
seq_1_1 id1  
seq_1_2 id2  
seq_2 id3
```

CLASSICAL SKETCH

The user provides a fasta file and the result file(s) of a all-against-all BLAST search in tabular format. It is now necessary to use SiLiX as a preliminary step, with `--net` option (see

[Documentation](#)

):

```
silix seq.fasta blastall.out --net > seqSLX.fnodes
```

A file blastall.net has been generated.

NB : if you use the MPI version of SiLiX (with mpirun), you get multiple .net files that you need to concatenate into in single .net file.

The user can now use hifix :

```
hifix seq.fasta blastall.net seqSLX.fnodes > seqHFX.fnodes
```

See also example files in the data/ directory of the package.

MEMORY USE

URL of the page: <https://lbbe.univ-lyon1.fr/fr/Hifix>

Due to mafft-profile requirements, it may be necessary to run HiFiX on computers with more than 2 GB RAM. If SiLiX results displays pre-families larger than 15000 sequences, we recommend using more than 4 GB RAM. Please note that, if HiFiX runs on multiple cores (option -t), the memory requirements are additive with the number of processes.

> **THE HARD WAY** (for very large datasets, if you have cluster or grid facilities, please

[contact us](#)

):

The user must program a pipeline with independent tasks (hifixcore program) distributed and submitted to a scheduler, like this :

```
silix .....
silix-split ....
for all <FASTAFILE> <NETFILE> do
  hifixcore <FASTAFILE> <NETFILE> (job to be submitted)
done
```

References



HiFiX is developed by :

- > [Laurent Duret](#)
- > Daniel Kahn
- > [Vincent Miele](#)
- > [Simon Penel](#)

If you use HiFiX in a published work, please cite the following reference :

Miele,V., Penel, S., Daubin,V., Picard,F., Kahn,D. and Duret,L.,
[High-quality sequence clustering guided by network topology and multiple alignment likelihood](#)
, **Bioinformatics 2012**

HiFiX is also mentioned in :

Dessimoz,C., Gabaldon,T., Roos, D.S., Sonnhammer,E., Herrero,J. and the Quest for Orthologs Consortium (incl. Miele,V.),
[Toward Community Standards in the Quest for Orthologs](#)
, **Bioinformatics 2012**

Contact



For any bugs, information or feedback, please contact:

[Vincent Miele](#)

▶ Download

[multidom.fasta](ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/data/) (ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/data/)

and

[blastmultidom.out](ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/data/) (ftp://pbil.univ-lyon1.fr/pub/logiciel/hifix/data/)

- › *.fasta*: sequences in FASTA format (less than 256 character per line and no "U" character [to be replaced by "X"])
- › *.out*: BLAST results in tabulated format
- › *.fnodes*: list of pairs in format "family_id sequence_id"

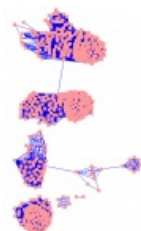
▶ Build pre-families with silix

```
silix multidom.fasta blastmultidom.out --net > multidom_SLX.fnodes
```

- › 8 pre-families displayed in *multidom_SLX.fnodes*



- › *blastmultidom.net* is generated and corresponds to the similarity network represented below



NB : if you use the MPI version of SiLiX (with mpirun), you get multiple *.net* files that you need to concatenate into in single *.net* file.

▶ Build high-quality families with hifix

```
hifix multidom.fasta blastmultidom.net multidom_SLX.fnodes > multidom_HFX.fnodes
```

- › 9 families displayed in *multidom_HFX.fnodes*



- › if *n* are processors available, use option

```
-t n
```