

WITCH

Presenter: Chengze Shen
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign

Multiple sequence alignment challenges

- Many sequences need to be aligned (thousands to millions)
 - High rates of evolution
 - Sequence length heterogeneity, and especially short sequences
-
- All the challenges affect runtime and accuracy
 - High accuracy is possible with existing methods using divide-and-conquer, if no sequence length heterogeneity
 - If sequence length varies, one approach is **incrementally adding sequences into a growing alignment**

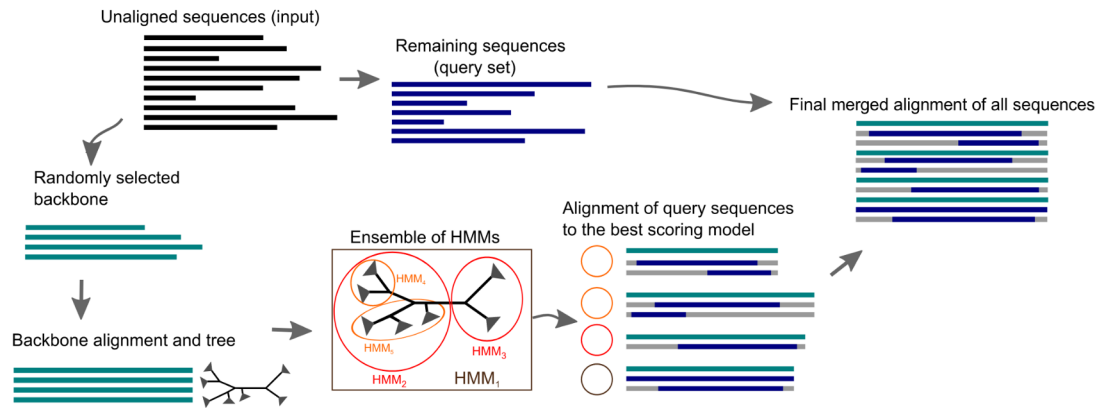
UPP: Prior MSA method given sequence length heterogeneity

UPP (Nguyen et al., Genome Biology, 2015)

Can be used for MSA on datasets with sequence length heterogeneity

- Divide dataset into full-length and other
- Construct MSA on full-length sequences
- Add remaining sequences one-by-one using “**ensemble of Hidden Markov Models**”

UPP



Each sequence X is added to backbone by:

1. Pick the top HMM (using bitscore).
2. Use the selected HMM to add X to a copy of backbone alignment.
3. Merge all copies.

UPP is a divide-and-conquer framework.

Improving UPP

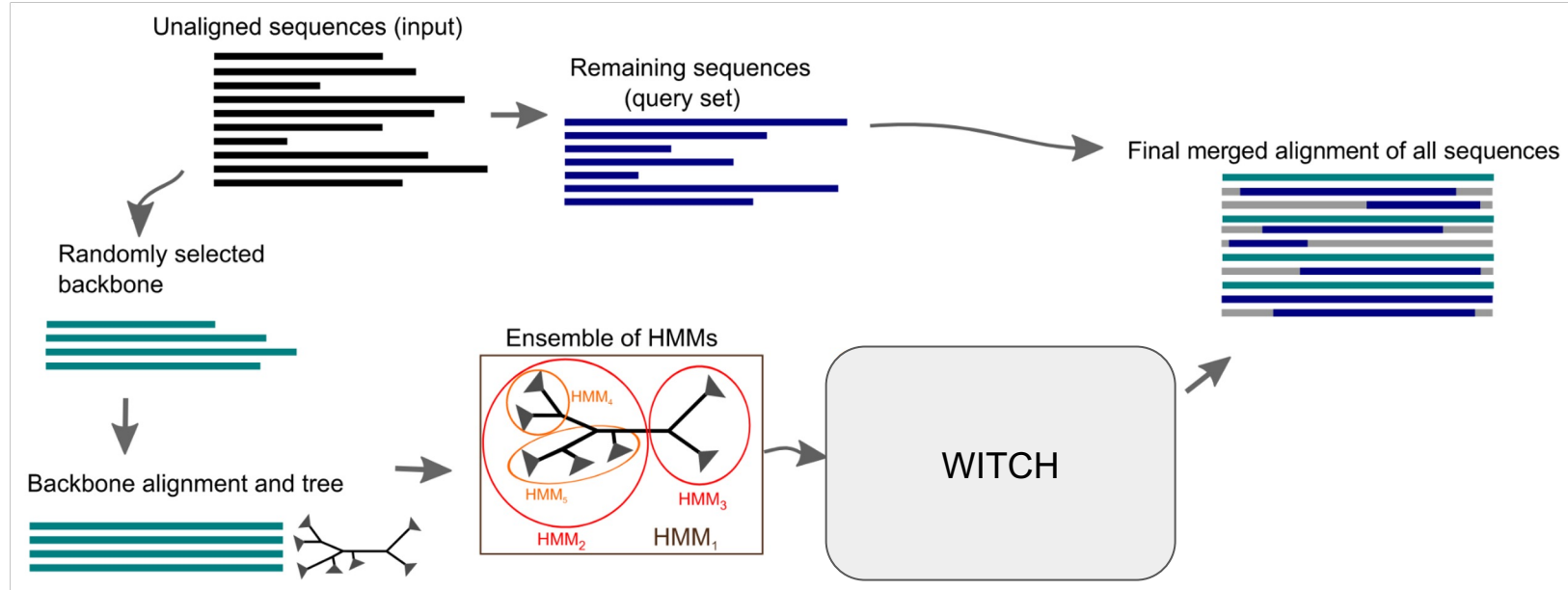
UPP adds sequences one-by-one (independently)

Each sequence picks the single best HMM in the ensemble

Improved accuracy could be obtained by:

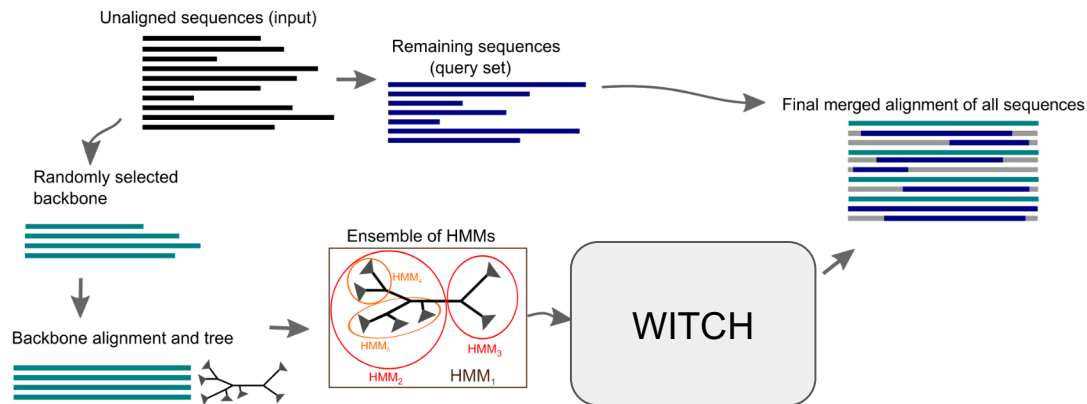
- Using the ensemble more intelligently (WITCH, Shen et al., J Comp Biol 2022), especially for adding short sequences
- Adding sequences in groups (EMMA, Shen et al., Algs Mol Biol, 2023), especially for adding full-length sequences

WITCH is built on UPP



WITCH changes on how new (query) sequences are aligned.

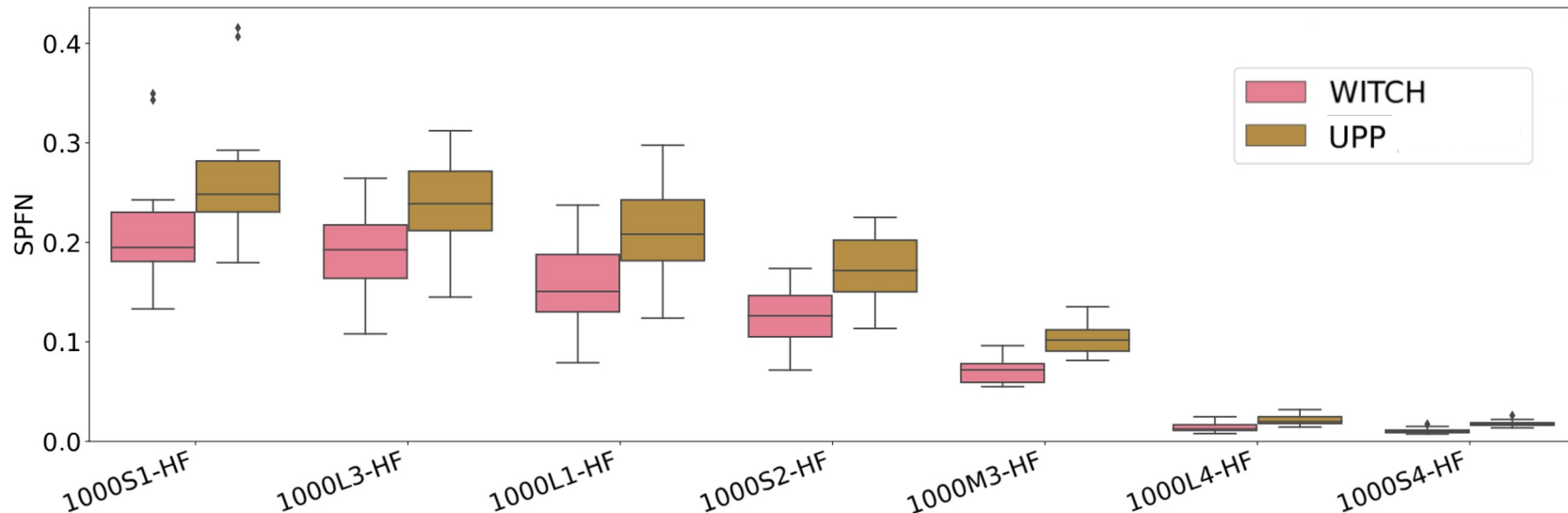
WITCH is built on UPP



Each sequence X is added to backbone by:

1. Pick the top **k** HMMs using adjusted bitscore (**k** is parameter).
2. Use the selected **k** HMMs to add X to a copy of backbone alignment.
 - a. A “consensus” alignment.
3. Merge all copies.

Comparison when aligning with fragmentary sequences



> WITCH and UPP have the same backbone.

> Modified version of Figure 6 from Shen et al., J Comp Bio, 2022.

> Average SPFN (% missing homologies), lower the better.

> “1000XX-HF”: 1000 sequences, some highly fragmentary, simulated data, varying rates of evolution.

WITCH runtime and scalability

- WITCH has been able to run on laptops to align datasets with 100,000 sequences
- *de novo* alignment, average runtime
 - ~1.3 hours aligning 5000 sequences, ~1000bp each
 - ~4.3 hours aligning the largest 10 Homfam datasets (15k-94k AA sequences)
- Runtime can improve further by limiting the number of HMMs in the ensemble
 - A parameter (-A) to modify in the software
 - May come with some cost of alignment accuracy

WITCH code

WITCH - WelgHTed Consensus Hmm alignment

pyPI [v1.0.8](#) python [3.7](#) | [3.8](#) | [3.9](#) | [3.10](#) | [3.11](#) | [3.12](#) build [passing](#) license [GPL-3.0](#) CHANGELOG DOI [10.1089/cmb.2021.0585](#)

Developer: Chengze Shen

Table of contents

- [News](#)
- [Method Overview](#)
 - [Note and Acknowledgement](#)
- [\(Important\) Software Output Explanation](#)
- [Installation](#)
 - [Install with PyPI \(`pip` \)](#)
 - [Install from the source file](#)
 - [Requirements](#)
 - [Installation Steps](#)
 - [main.config](#)
 - [user-specified config file](#)
- [Usage](#)
 - [Examples](#)
 - [Scenario A](#)
 - [Scenario B](#)
 - [Scenario C](#)
 - [Scenario D - additional options](#)
 - [Scenario E - with user-specified config file](#)
- [TODO list](#)

- Implemented with Python
- Available on GitHub and PyPI:
<https://github.com/c5shen/WITCH>

Installation and Usage: WITCH

```
# 1. Install with pip (--user if no root access)
```

```
pip3 install witch-msa [--user]
```

```
# 2. After installation, users can run WITCH with either "witch-msa" or "witch.
```

```
#    First time running WITCH will create the config file at ~/.witch_msa/main.
```

```
witch-msa [-h]    # or,
```

```
witch.py [-h]
```



I will demonstrate some of the use cases in the following slide

Scenarios to use: WITCH

1. Scenario 1: *de novo* alignment

```
witch.py -i [input fasta file]
```

(can also be in gzip format)

1. Scenario 2: Adding new sequences to an existing alignment (without a phylogenetic tree)

```
witch.py -b [existing alignment file] -q [new sequences file]
```

1. Scenario 3: Adding new sequences to an existing alignment with its phylogenetic tree

```
witch.py -b [existing alignment file] -e [existing phylogeny file] -q [new sequences file]
```

Sample logging

3. Scenario 3: Adding new sequences to an existing alignment with its phylogenetic tree

```
witch.py -b [existing alignment file] -e [existing phylogeny file] -q [new sequences file]
```

```
Decomposing the backbone tree...  
Running...: 100%|██████████████████████████████████████| 141/141 [00:01<00:00, 91.41it/s]  
  
Performing all-against-all HMMSearches between the backbone and queries...  
Running...: 100%|██████████████████████████████████| 564/564 [00:27<00:00, 20.16it/s]  
Running...: 100%|██████████████████████████████████| 141/141 [00:00<00:00, 1564.16it/s]  
  
Reading and ranking bit-scores from HMMSearch files...  
Running...: 100%|██████████████████████████████████| 141/141 [00:00<00:00, 1193.59it/s]  
  
Calculating weights (adjusted bit-scores)...  
Running...: 100%|██████████████████████████████████| 500/500 [00:00<00:00, 5760.35it/s]  
  
Performing GCM alignments on query subsets...  
Running...: 100%|██████████████████████████████████| 500/500 [00:17<00:00, 29.27it/s]  
  
All GCM subproblems finished! Doing merging with transitivity...  
  
All done! WITCH finished in 47.47548985481262 seconds...
```

Sample output

3. Scenario 3: Adding new sequences to an existing alignment with its phylogenetic tree

```
witch.py -b [existing alignment file] -e [existing phylogeny file] -q [new sequences file]
```

Output to directory (default) → `witch_output/*`

Output Alignment files → `witch_output/aligned.fasta`

→

`witch_output/aligned.masked.fasta`

Summary: WITCH

- Both add sequences into existing alignments
 - WITCH is best when there are short sequences to add
 - EMMA is best when all the sequences are close to full-length
 - Both improve on UPP in terms of alignment accuracy
-
- I maintain these codes, and can be reached at zhazhashen@gmail.com or chengze5@illinois.edu

This work was supported by U.S. National Science Foundation grant 2006069 (to Tandy Warnow).