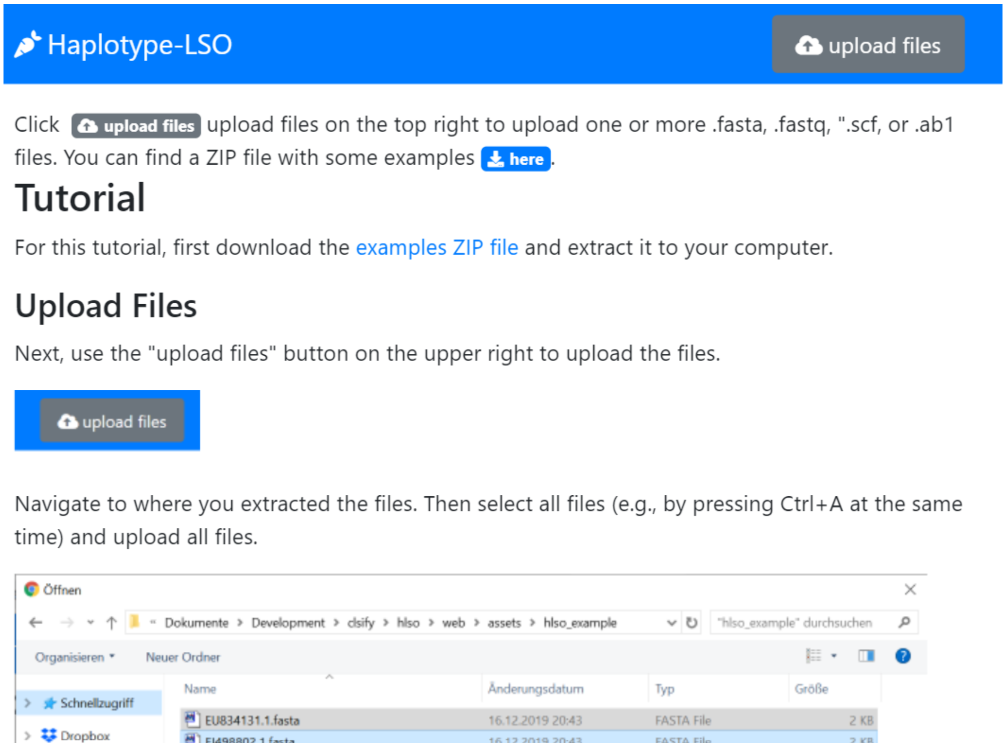
Haplotype-Lso

Release 0.3

Apr 24, 2020

Overview

Haplotype-Lso is a program for the automated determination of *C. liberibacter solanacearum*. For the input, it takes capillary sequencing data from 16S, 16S-23S, and 50S locus enrichment. It then performs a multi-locus sequence typing (MLST) following IPPC (International Plant Protection Convention) standard DP 21: Candidatus Liberibacter solanacearum.



Click upload files upload files on the top right to upload one or more .fasta, .fastq, ".scf, or .ab1 files. You can find a ZIP file with some examples here.

Tutorial

For this tutorial, first download the examples ZIP file and extract it to your computer.

Upload Files

Next, use the "upload files" button on the upper right to upload the files.

Navigate to where you extracted the files. Then select all files (e.g., by pressing Ctrl+A at the same time) and upload all files.

Öffnen
« Dokumente › Development › clsify › hlso › web › assets › hlso_example
"hlso_example" durchsuchen
Organisieren
Neuer Ordner
Schnellzugriff
Dropbox

Name	Änderungsdatum	Typ	Größe
EU834131.1.fasta	16.12.2019 20:43	FASTA File	2 KB
FJ498802.1.fasta	16.12.2019 20:43	FASTA File	2 KB

Fig. 1: The Haplotype-Lso start page.

CHAPTER 1

Public Web App

You can find the publically available version at https://haplotype-los.bihealth.org (hosting is provided by Core Unit Bioinformatics of the Berlin Institute of Health). To get started, you can follow the *Haplotype-Lso Tutorial*.

CHAPTER 2

Running on your Computer

Haplotype-Lso has very low hardware requirements but depends on several external programs (e.g., NCBI BLAST) and must be run on a Linux computer. The easiest way is to install it via Docker:

```
$ docker run quay.io/biocontainers/haplotype-lso:<version>

# for example:

$ docker run quay.io/biocontainers/haplotype-lso:0.3.2--0
```

See here for a list of all versions.

You can also install it via Bioconda. After installing Bioconda:

```
$ conda install haplotype-lso
```

CHAPTER 3

Source Code / Open Source

Haplotype-Lso is written in the Python programming language using Plotly Dash. It is licensed under the permissive MIT license and you can find the source code in the Github project.

3.1 Haplotype-Lso Tutorial

For this tutorial, first download the examples ZIP file and extract it to your computer.

3.1.1 Upload Files

Next, use the “upload files” button on the upper right to upload the files.

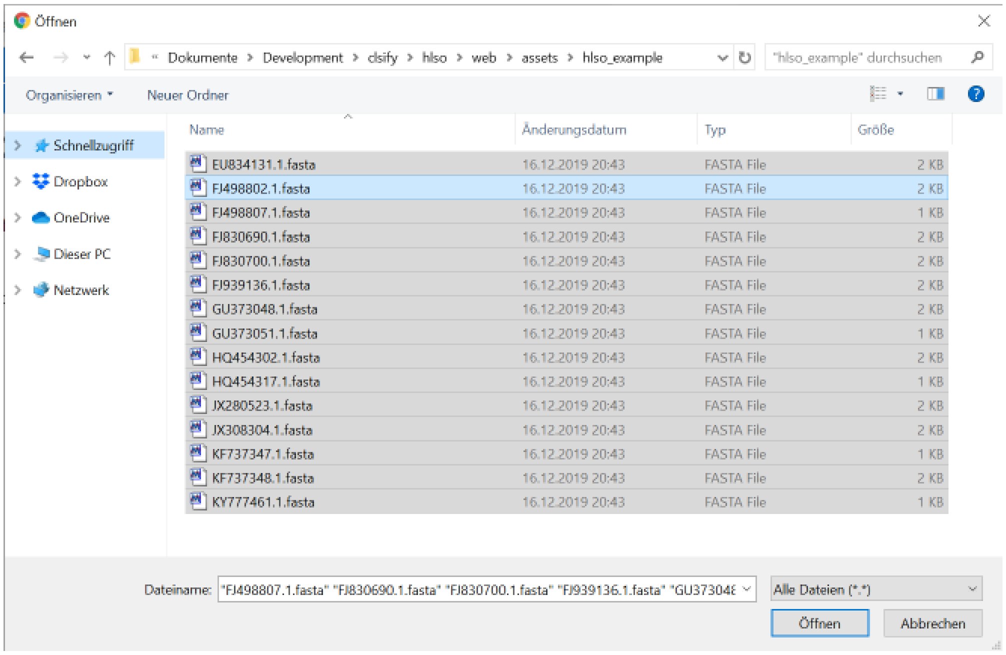
Navigate to where you extracted the files. Then select all files (e.g., by pressing *Ctrl* + *a* at the same time) and upload all files.

Wait for a moment and the haplotyping results will be displayed.

Note: When looking at the example data, you will notice that the file names all follow the same pattern:

The first element of the file name is the sample name, the second element is the name of the target region, separated by a dot. The tool will use this information to group your sequences later and (a) group type information by sample and (b) compute consensus by sample and region. Item (a) allows to determine the haplotype based on multipe regions per sample and (b) allows to use sequence from both forward and reverse primers.

Optionally, you can also add more information (e.g., primer-related) in a third group: `<sample>.<region>.<primer>.fasta`.


Öffnen
« Dokumente › Development › clsify › hlso › web › assets › hlso_example
"hlso_example" durchsuchen
Organisieren ▾
Neuer Ordner
Schnellzugriff
Dropbox
OneDrive
Dieser PC
Netzwerk
Öffnen
Abbrechen

Name	Änderungsdatum	Typ	Größe
EU834131.1.fasta	16.12.2019 20:43	FASTA File	2 KB
FJ498802.1.fasta	16.12.2019 20:43	FASTA File	2 KB
FJ498807.1.fasta	16.12.2019 20:43	FASTA File	1 KB
FJ830690.1.fasta	16.12.2019 20:43	FASTA File	2 KB
FJ830700.1.fasta	16.12.2019 20:43	FASTA File	2 KB
FJ939136.1.fasta	16.12.2019 20:43	FASTA File	2 KB
GU373048.1.fasta	16.12.2019 20:43	FASTA File	2 KB
GU373051.1.fasta	16.12.2019 20:43	FASTA File	1 KB
HQ454302.1.fasta	16.12.2019 20:43	FASTA File	2 KB
HQ454317.1.fasta	16.12.2019 20:43	FASTA File	1 KB
JX280523.1.fasta	16.12.2019 20:43	FASTA File	2 KB
JX308304.1.fasta	16.12.2019 20:43	FASTA File	2 KB
KF737347.1.fasta	16.12.2019 20:43	FASTA File	1 KB
KF737348.1.fasta	16.12.2019 20:43	FASTA File	2 KB
KY777461.1.fasta	16.12.2019 20:43	FASTA File	1 KB

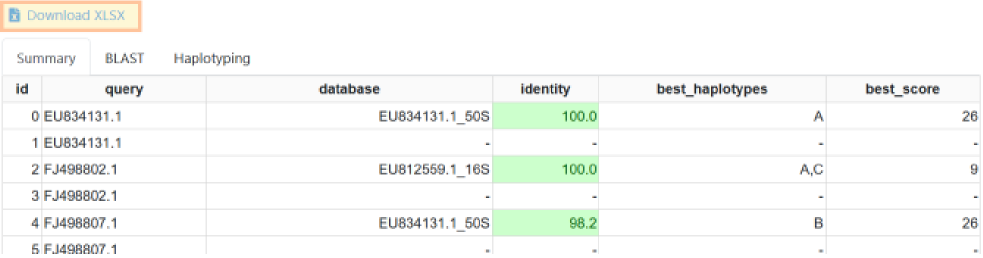
Dateiname: "FJ498807.1.fasta" "FJ830690.1.fasta" "FJ830700.1.fasta" "FJ939136.1.fasta" "GU37304ε

Alle Dateien (*.*)

Fig. 1: Select the files you wan to upload (*Ctrl* + *a* to select all files).

3.1.2 Result Summary Tab

Note that you can download the results information also as an Excel file using the "Download XLSX" links on the top.



id	query	database	identity	best_haplotypes	best_score
0	EU834131.1	EU834131.1_50S	100.0	A	26
1	EU834131.1	-	-	-	-
2	FJ498802.1	EU812559.1_16S	100.0	A,C	9
3	FJ498802.1	-	-	-	-
4	FJ498807.1	EU834131.1_50S	98.2	B	26
5	FJ498807.1	-	-	-	-

Fig. 2: Result summary tab.

The **Summary** tab shows the following information:

query the query file name

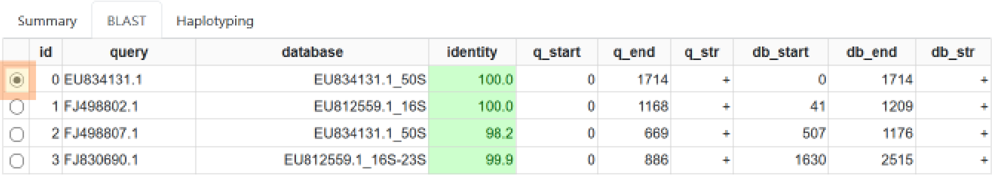
database the ID of the database sequence with the best match. The name will start with the GenBank identifier, followed by an underscore _ and then the name of the region (currently one of 16S, 16S-23S, or 50S).

identity the identity of the BLAST match with the reference in percent.

best_haplotypes the best haplotypes based on the informative values on the reference. NB: if the sequencing error rate is high or the sequence is too short then not all informative positions will be covered. In this case, there can be ambiguity and more than one haplotype can be returned. For example, haplotypes A and C only differ in a single position on the 16S locus.

best_score the score of the best match used for haplotype identification. Concordance with a variant in the haplotyping table contributes a score of "plus one", discordance contributes a "minus one". The sum is the overall score.

3.1.3 Result BLAST Tab



	id	query	database	identity	q_start	q_end	q_str	db_start	db_end	db_str
●	0	EU834131.1	EU834131.1_50S	100.0	0	1714	+	0	1714	+
○	1	FJ498802.1	EU812559.1_16S	100.0	0	1168	+	41	1209	+
○	2	FJ498807.1	EU834131.1_50S	98.2	0	669	+	507	1176	+
○	3	FJ830690.1	EU812559.1_16S-23S	99.9	0	886	+	1630	2515	+

Fig. 3: The BLAST result tab.

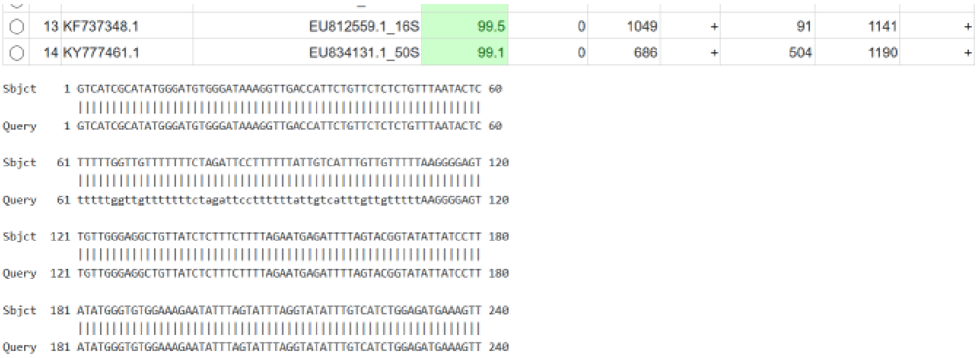
The **BLAST** tab provides the following information:

query, database, identity see above

q_start, q_end, q_str the start and end position of the match in the query and its strand

db_start, db_end, db_start the start end end position of the match in the database and its strand

Further, you can select each match with the little round button on the right. The corresponding BLAST match will be displayed below the results table.



○	13	KF737348.1	EU812559.1_16S	99.5	0	1049	+	91	1141	+
○	14	KY777461.1	EU834131.1_50S	99.1	0	686	+	504	1190	+

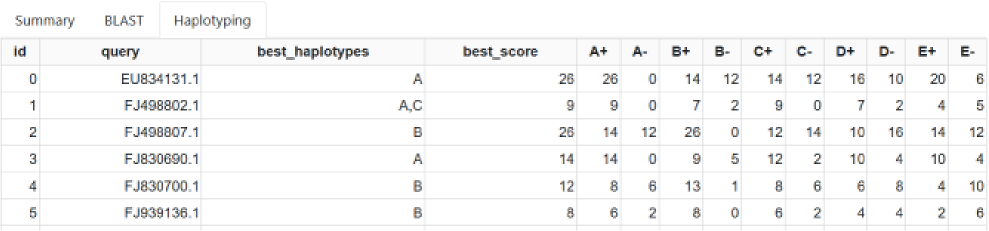
```
Sbjct    1 GTCATCGCATATGGGATGTGGGATAAAGGTTGACCATTCTGTTCTCTCTGTTTAATACTC 60
           ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Query    1 GTCATCGCATATGGGATGTGGGATAAAGGTTGACCATTCTGTTCTCTCTGTTTAATACTC 60

Sbjct   61 TTTTTGGTTGTTTTTTTCTAGATTCCTTTTTTATTGTCATTTGTTGTTTTTAAGGGGAGT 120
           ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Query   61 tttttggttgtttttttctagattccttttttattgtcatttgttgtttttAAGGGGAGT 120

Sbjct  121 TGTTGGGAGGCTGTTATCTCTTTCTTTTAGAATGAGATTTTAGTACGGTATATTATCCTT 180
           ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Query  121 TGTTGGGAGGCTGTTATCTCTTTCTTTTAGAATGAGATTTTAGTACGGTATATTATCCTT 180

Sbjct  181 ATATGGGTGTGGAAAGAATATTTAGTATTTAGGTATATTTGTCATCTGGAGATGAAAGTT 240
           ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Query  181 ATATGGGTGTGGAAAGAATATTTAGTATTTAGGTATATTTGTCATCTGGAGATGAAAGTT 240
```

3.1.4 Result Haplotyping



id	query	best_haplotypes	best_score	A+	A-	B+	B-	C+	C-	D+	D-	E+	E-
0	EU834131.1	A	26	26	0	14	12	14	12	16	10	20	6
1	FJ498802.1	A,C	9	9	0	7	2	9	0	7	2	4	5
2	FJ498807.1	B	26	14	12	26	0	12	14	10	16	14	12
3	FJ830690.1	A	14	14	0	9	5	12	2	10	4	10	4
4	FJ830700.1	B	12	8	6	13	1	8	6	6	8	4	10
5	FJ939136.1	B	8	6	2	8	0	6	2	4	4	2	6

Fig. 4: The haplotyping result tab.

The **Haplotyping** tab shows more details on the haplotyping results.

query see above

best_haplotypes the best haplotype(s) for the given query

best_score the best score of the best haplotype(s)

A+, A-, etc. for each haplotype known to Haplotype-LSO, the number of positive/concordant and negative/discordant position

3.1.5 Phylogenetic Analysis

The **Dendrograms** tab shows results of hierarchical clustering using the UPGMA algorithm for each region. The input of the UPGMA algorithm is based on the pairwise BLAST identities (`1.0 - identity`).

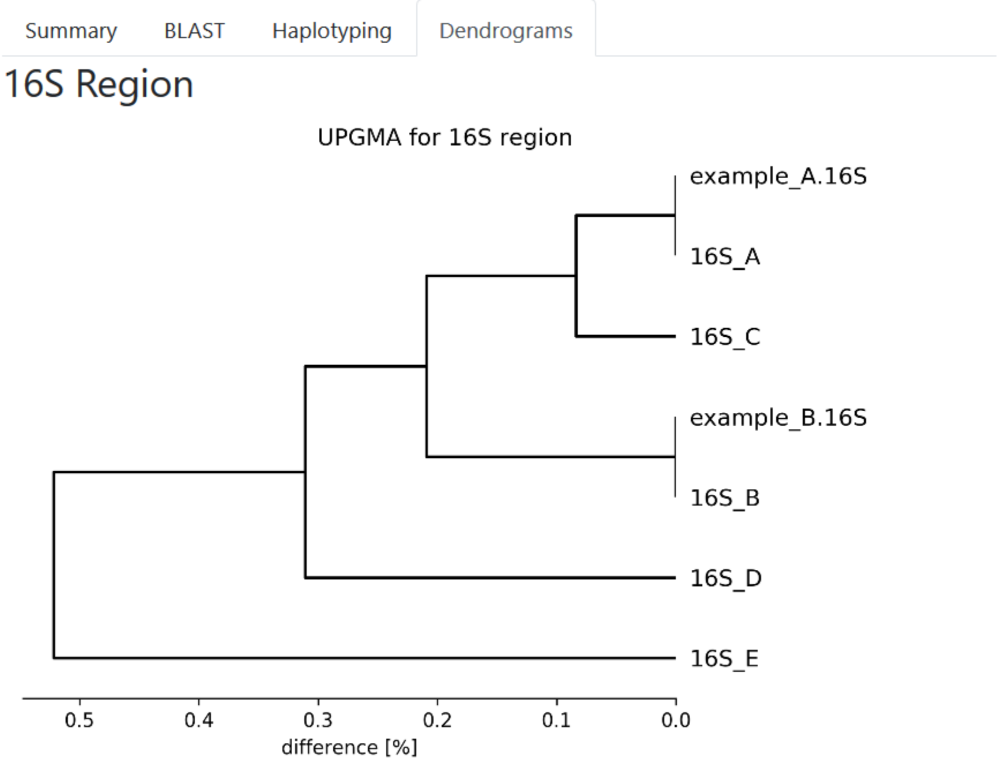


Fig. 5: The dendrograms tab with the phylogenetics analysis.

3.2 Lso Haplotyping

This section explains the haplotyping method implemented in the Haplotype-Lso. It follows the approach described in [DP21]:

> *DP21: Candidatus Liberibacter solanacearum.* International Standards for Phytosanitary Measures 27. Annex 21. 03 Apr 2017. [URL].

3.2.1 Algorithmic Approach

Overall, the algorithmic approach for haplotyping of Candidatus *liberibacter solanacearum* (Lso) is as follows.

User Input

Haplotyping is described for multiple user-provided sequences from multiple genomic regions of a single sample. Following DP21, the sequences would be generated by capillary sequencing of PCR products from predetermined regions. DP21 suggests three primer pairs targeting the 16S, 16S-23S, and 50S regions. However, other regions have been suggested, e.g., in [SwGa2019]. At the moment, Haplotype-Lso provides support for 16S, 16S-23S, and 50S regions as these are widely available. However, extensions to further regions are easily possible.

Reference Dataset

For background information, the haplotyping needs a reference data set (cf. *Reference Construction* on how to rebuild it using the command line intreface of Haplotype-Lso). The reference dataset consists of a reference sequence for each target region `R`. Further, for each known Lso haplotype `H` where the sequence is known for `R` a variant table `T` is given. `T` provides a list of variants with respect to `R` and whether they are present in `H` or whether `H` shows the reference sequence.

For example, the table for the 16S locus following DP21 looks as follows. The reference is given by its GenBank ID, the positions are 1-based and reference and alternative bases are given in VCF syntax (c.f. [DAAA2011] while the description is given in HGVS notation [DDMH2016]. The columns A to E show the allele for each known haplotype.

reference	region	pos	ref	alt	description	A	B	C	D	E
EU812559.1	16S	108	AT	A	n.109delT	AT	AT	AT	AT	A
EU812559.1	16S	115	A	G	n.115A>G	A	A	A	A	G
EU812559.1	16S	116	C	T	n.116C>T	C	C	C	T	C
EU812559.1	16S	151	A	G	n.151A>G	A	A	A	A	G
EU812559.1	16S	212	T	G	n.212T>G	T	G	T	T	T
EU812559.1	16S	581	T	C	n.581T>C	T	C	T	T	T
EU812559.1	16S	959	C	T	n.959C>T	C	C	C	C	T
EU812559.1	16S	1039	A	G	n.1039A>G	A	A	G	G	A
EU812559.1	16S	1040	CC	C	n.1041delC	CC	CC	CC	CC	C
EU812559.1	16S	1073	G	A	n.1073G>A	G	G	G	A	G

Haplotyping Algorithm

The haplotyping algorithm for one query sequence is as follows:

1. For each user sequence `S`, a BLAST search is performed in the reference sequences for the regions defined in the input data sets.
2. The best match is taken for `S`, and variant descriptions are derived (using the normalization approach described in [TAK2015].
3. The call set from the user sequence is compared to all calls that overlap with the BLAST match and a score is computed by summation. For each haplotype from the table each position is considered. If the user input is equal to the haplotype at the position, the score is increased by 1 and it is decreased otherwise.

If the user provides two sequences for a region (e.g., for forward and reverse primer) then only positions are considered where all sequences show the same variant. Finally, the scores for all regions for a sample are aggregated for each haplotype. The haplotype(s) with the highest score are yielded as the final haplotype for the given region.

Final Remarks

Obviously, the described approach is limited to identifying known haploytpes. However, haplotypes A and C only differ in one base on the 16S region and the algorithm is capable of differentiating between them even in the presence of sequencing errors (it is highly unlikely for sequencing errors to show a known variant on the hundreds of reference positions). Also, it is is easy to validate the haplotyping result by manual inspection of the BLAST matches.

Future extensions are:

- Also consider phylogenetic approaches that would allow the identification of new haplotypes (e.g., if the user sequence is an outlier to all known haplotype sequences).
- Perform BLAST searches in larger databases to identify other outliers.

3.3 Reference Construction

This section describes how the reference set (reference sequences and haplotyping position) can be generated.

Note: This section needs extension and more explanation.

3.3.1 Input

The overall input is a TSV file `seeds_accessions` that lists for each haplotype and region a GenBank accession with the prototype sequence.

```
species              haplotype  region   accession   source

Ca. L. solanacearum  A          16S      FJ498802.1  .
Ca. L. solanacearum  B          16S      FJ939136.1  .
Ca. L. solanacearum  C          16S      GU373048.1  .
Ca. L. solanacearum  D          16S      HQ454302.1  .
Ca. L. solanacearum  E          16S      KF737348.1  .

Ca. L. solanacearum  A          16S-23S  FJ830690.1  .
Ca. L. solanacearum  B          16S-23S  FJ830700.1  .
Ca. L. solanacearum  C          16S-23S  JX280523.1  .
Ca. L. solanacearum  D          16S-23S  JX308304.1  .
Ca. L. solanacearum  E          16S-23S  KF737347.1  .

Ca. L. solanacearum  A          50S      EU834131.1  .
Ca. L. solanacearum  B          50S      FJ498807.1  .
Ca. L. solanacearum  C          50S      GU373051.1  .
Ca. L. solanacearum  D          50S      HQ454317.1  .
Ca. L. solanacearum  E          50S      KY777461.1  .
```

3.3.2 Download Seed Sequences

```
$ hlso cli ref_download \
    path/to/seeds_accession.tsv \
    path/to/seeds_paths.tsv
```

This will download sequences by accession, download them next to the `seeds_accession.tsv` file. It will write the file `seeds_paths.tsv` with the names of the downloaded files:

```
species              haplotype  region   accession   path
Ca. L. solanacearum  A          16S      FJ498802.1  FJ498802.1.fasta
Ca. L. solanacearum  B          16S      FJ939136.1  GU373048.1.fasta
Ca. L. solanacearum  C          16S      GU373048.1  GU373048.1.fasta
Ca. L. solanacearum  D          16S      HQ454302.1  HQ454302.1.fasta
Ca. L. solanacearum  E          16S      KF737348.1  KF737348.1.fasta
Ca. L. solanacearum  A          16S-23S  FJ830690.1  FJ830690.1.fasta
Ca. L. solanacearum  B          16S-23S  FJ830700.1  FJ830700.1.fasta
Ca. L. solanacearum  C          16S-23S  JX280523.1  JX280523.1.fasta
Ca. L. solanacearum  D          16S-23S  JX308304.1  JX308304.1.fasta
Ca. L. solanacearum  E          16S-23S  KF737347.1  KF737347.1.fasta
Ca. L. solanacearum  A          50S      EU834131.1  EU834131.1.fasta
```

(continues on next page)

(continued from previous page)

```
Ca. L. solanacearum  B          50S      FJ498807.1  FJ498807.1.fasta
Ca. L. solanacearum  C          50S      GU373051.1  GU373051.1.fasta
Ca. L. solanacearum  D          50S      HQ454317.1  HQ454317.1.fasta
Ca. L. solanacearum  E          50S      KY777461.1  KY777461.1.fasta
```

3.3.3 Performing Seed BLAST Queries

The next step is to perform a BLAST search via NCBI WWWBLAST to obtain sequences similar to the seeds.

```
$ hlso ref_blast path/to/seeds_paths.tsv
```

For each seeed query `accession.fasta`, a file `accession.blast.xml` will be generated with the BLAST results.

3.3.4 Consensus and Table Creation

Finally, compute consensus sequences and the haplotyping table.

```
$ hls ref_consensus path/to/seeds_path.tsv \
    --output-table haplotype_table.txt
```

This will perform a consensus computation of the seeds, generate a haplotype-specific sequence for each region and each haplotype, and create a haplotyping table.

The file `haplotype_table.txt` can then be used for the haplotyping of sequences themselves.

3.4 Command Line Interface

You can alos run Haplotype-Lso locally from the command line.

Note: This section needs some extension.

```
$ hlso cli \
    [--sample-name-from-file] \
    [--sample-regex REGEX] \
    [--output OUTPUT] \
    seq_file [seq_file ...]
```

This will read all sequence files `seq_file` (can be FASTA, FASTQ, AB1, SCF), perform conversion to FASTA (if needed) and then perform a haplotyping. When provided, the result will be written to the XLSX file `OUTPUT`.

You can override the regular expression to extract the sample name and region from the query name with `--sample-regex`.

By default, the query sequence names are taken from their identifier. As this is hard for the binary files AB1 and SCF, you can also configure Haplotype-Lso to use the file names (without extension) as the sample names. This is the behaviour from the web frontend.

3.5 References

3.6 Indices and tables

- genindex
- search

Bibliography

[DAAA2011] Danecek, P., Auton, A., Abecasis, G., Albers, C.A., Banks, E., DePristo, M.A., Handsaker, R.E., Lunter, G., Marth, G.T., Sherry, S.T., McVean, G., Durbin, R. "The variant call format and VCFtools." Bioinformatics 27, no. 15 (2011): 2156-2158.

[TAK2015] Tan, A., Abecasis, G.R., and Kang, H.M. "Unified representation of genetic variants." Bioinformatics 31.13 (2015): 2202-2204.

[DDMH2016] den Dunnen, J.T., Dalgleish, R., Maglott D.R., Hart, R.K., Greenblatt, M.S., McGowan-Jordan, J., Roux, A.-F., Smith, T., Antonarakis, S.E., Taschner, P.E.M.. "HGVS Recommendations for the Description of Sequence Variants: 2016 Update". Humaan Mutation 37.6 (2016): 564-569.

[DP21] *DP21: Candidatus Liberibacter solanacearum.* International Standards for Phytosanitary Measures 27. Annex 21. 03 Apr 2017. [URL].

[SwGa2019] Swisher Grimm, K.D., and Garczynski, S.F. "Identification of a New Haplotype of 'Candidatus Liberibacter solanacearum'in Solanum tuberosum." Plant disease 103.3 (2019): 468-474.