Oct 27, 2022

Determining MLST allele sequences in novel STs

Varun Shamanna[1]

[1]Central Research Laboratory, Kempegowda Institute of Medical Sciences, Bengaluru, India

1 Works for me Share

dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

Varun Shamanna

ABSTRACT

Steps required to determine Novel allele sequence of MLST from the assembly files using MLSTaR R package

DOI

dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

DOCUMENT CITATION

Varun Shamanna 2022. Determining MLST allele sequences in novel STs. **protocols.io** https://dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

LICENSE

CREATED

Oct 27, 2022

LAST MODIFIED

Oct 27, 2022

DOCUMENT INTEGER ID

71866

ABSTRACT

Steps required to determine Novel allele sequence of MLST from the assembly files using MLSTaR R package

Citation: Varun Shamanna Determining MLST allele sequences in novel STs https://dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

There are many methods both manual and programmatic for achieving this. Here is just one method using an existing software tool - MLSTar (https://github.com/iferres/MLSTar)

1. First install the blast dependency. It is recommended to perform this using a conda package
   conda install -c bioconda blast
2. MLSTar is an R package so install R if not already installed (RStudio recommended). Then within the console in R/RStudio
3. Install the devtools package
   install.packages("devtools")
4. Install the MLSTar package
   devtools::install_github('iferres/MLSTar')
5.

6. This package will only work "out of the box" with MLST schemes on the PubMLST database/website, so for other schemes (e.g Klebsiella on the Pasteur website) you need to download the profiles and allele sequences.
   For klebsiella the profiles can be found
   https://bigsdb.web.pasteur.fr/cgi-bin/bigsdb/bigsdb.pl?db=pubmlst_klebsiella_seqdef
   allele sequences can be found here
   https://bigsdb.web.pasteur.fr/cgi-bin/bigsdb/bigsdb.pl?db=pubmlst_klebsiella_seqdef&page=downloadAlleles

   I have pre-downloaded them and zipped them up here

Please note If providing your own profiles and sequences check that the header names in the profiles TSV match the names of the sequence file prefixes and that there are no trailing tabs in either the header or profile rows.

e.g
sed -i 's/[[:space:]]*$//' kpn_profiles.txt

on Mac OSX
sed -i '' 's/[[:space:]]*$//' kpn_profiles.txt

1. Then to find the profiles and extract the sequences we run the following commands
2. Find supported organisms
   listPubmlst_orgs()
3. Find schemes for a supported organism
   listPubmlst_schemes(org = "escherichia")
   There are many schemes including some very long ones which are the cgMLST schemes, however you will see 2 familiar ones for E.coli
   $scheme_1

```
[1] "adk"  "fumC" "gyrB" "icd"  "mdh"  "purA" "recA"
attr(,"Desc")
[1] "MLST (Achtman)"
$scheme_2
[1] "dinB" "icdA" "pabB" "polB" "putP" "trpA" "trpB" "uidA"
```

Citation: Varun Shamanna Determining MLST allele sequences in novel STs
https://dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

```
attr(,"Desc")
[1] "MLST (Pasteur)"
```

1. To call MLST and write alleles to a file

```
    results <- doMLST(
    c("G18000002.fasta", "G18000051.fasta"),
    org = "escherichia",
    scheme = 1,
    write = "all")
    results$result
          adk fumC gyrB icd mdh purA recA  ST
G18000002  92    4   87  96  70   58    2 648
G18000051  53   40   47  13  36   28   29 131
```

1. The allele sequences are found in a directory as shown below. In this case there are no new alleles and so there is no need to examine these on the linux terminal (not R console)

```
    ls -l alleles_escherichia_1/
total 28
-rw-rw-r-- 1 biouser biouser 1196 Nov  2 11:07 adk.fasta
-rw-rw-r-- 1 biouser biouser 1061 Nov  2 11:07 fumC.fasta
-rw-rw-r-- 1 biouser biouser 1044 Nov  2 11:07 gyrB.fasta
-rw-rw-r-- 1 biouser biouser 1160 Nov  2 11:07 icd.fasta
-rw-rw-r-- 1 biouser biouser 1026 Nov  2 11:07 mdh.fasta
-rw-rw-r-- 1 biouser biouser 1081 Nov  2 11:07 purA.fasta
-rw-rw-r-- 1 biouser biouser 1145 Nov  2 11:07 recA.fasta
```

1. For other schemes the paths to the profiles and allele sequences need to be provided (in this case they are in a directory called mlst_scheme) and you'll need to provide a dummy organism e.g test since "klebsiella" is not an officially supported organism

```
    results <- doMLST(
    c("G18583057.fasta", "G18583075.fasta"),
    org = "test",
    scheme = 1,
    schemeFastas = c(
"mlst_scheme/gapA.fas",
"mlst_scheme/infB.fas",
"mlst_scheme/mdh.fas",
"mlst_scheme/pgi.fas",
"mlst_scheme/phoE.fas",
"mlst_scheme/rpoB.fas",
"mlst_scheme/tonB.fas"),
schemeProfile = "mlst_scheme/kpn_profiles.txt",
write = "all")
```

When looking at the result this time you will see that the profiles have an unknown allele 'u1'

```
results$result
          gapA infB mdh pgi phoE rpoB tonB ST
G18583057    2    5  u1   1    4    1    4 NA
G18583075    2    5  u1   1    4    1    4 NA
```

Citation: Varun Shamanna Determining MLST allele sequences in novel STs
https://dx.doi.org/10.17504/protocols.io.36wgqj7kovk5/v1

Or

If you have novel profile instead of a unknown allele 'u1' the result will be a profile with all allele numbers assigned but ST still NA since the alleles present represent a new combination.

```
gapA infB mdh pgi phoE rpoB tonB  ST
G18250048    3   3  1  1   1   1   4  NA
```

1. The novel allele sequences can be found in the output file
   cat alleles_test_1/mdh.fasta

```
>mdh_u1;G18583057;NODE_2_length_734790_cov_17.426302
catcgacaaggtcgccgacccgccgcccgctttcgcttccacgacttcggtaccggcgtt
ctgaatacgtttagtcaggtcggcaatttcctgatcgctaaagctgacgccggggatctg
cgacagtaaaggcagaatggtgaccccggagtgaccaccaatgaccgggacttccacctc
ggttgccgatttacctttcagctccgccacaaaggtattggaacggatgatgtcaagcgt
ggtaacgccgaacagtttgtttttatcgtacacgccggctttttcagtacttcggcggc
gatagccacggtggtattcaccgggttggtgataatgccgatgcaggcctgcgggcaggt
tttggcaatctgctgcacgaggttcttcacgatacccgcattcacattaaacaggtcgga
acgatccatgccgggcttacgcgccacgcccgcggagatcagcactacatccgcg
>mdh_u1;G18583075;NODE_2_length_734148_cov_16.523686
cgcggatgtagtgctgatctccgcgggcgtggcgcgtaagcccggcatggatcgttccga
cctgtttaatgtgaatgcgggtatcgtgaagaacctcgtgcagcagattgccaaaacctg
cccgcaggcctgcatcggcattatcaccaacccggtgaataccaccgtggctatcgccgc
cgaagtactgaaaaaagccggcgtgtacgataaaaacaaactgttcggcgttaccacgct
tgacatcatccgttccaatacctttgtggcggagctgaaaggtaaatcggcaaccgaggt
ggaagtcccggtcattggtggtcactccggggtcaccattctgcctttactgtcgcagat
ccccggcgtcagctttagcgatcaggaaattgccgacctgactaaacgtattcagaacgc
cggtaccgaagtcgtggaagcgaaagcgggcggcgggtcggcgaccttgtcgatg
```