PRACTICAL TRAINING - DAY 1

Login to a Linux image in VirtualBox

- username: ltr
- password: copia

Troubleshooting Protocols 1-4

Protocol 1

- Copy protocol history to your account on RepeatExplorer server
    - login to your account on https://repeatexplorer-elixir.cerit-sc.cz/
    - copy and paste this URL to the browser: https://repeatexplorer-elixir.cerit-sc.cz/galaxy/u/kavonrtep/h/protocol-1--single-species
    - click on **[+]** (import history)
- The downloaded history is also available from the Linux image at
    - `~/Desktop/data/histories/protocol_1`
- manual correction of repeat annotations and recalculation of repeat proportions in the genome:
    - edit `CLUSTER_TABLE.csv` (in `~/Desktop/data/histories/protocol_1`) and save as `CLUSTER_TABLE_corrected.csv`
    - upload to the Galaxy server
    - use the tool "RepeatExplorer Utilities -> Repeat proportions from CLUSTER_TABLE"

Protocol 2

- Copy protocol history to your account on RepeatExplorer server
    - login to your account on https://repeatexplorer-elixir.cerit-sc.cz/
    - copy and paste this URL to the browser: https://repeatexplorer-elixir.cerit-sc.cz/galaxy/u/kavonrtep/h/protocol-2--comparative-analysis
    - click on **[+]** (import history)
- The downloaded history is also available from the Linux image at
    - `~/Desktop/data/histories/protocol_2`

- visualization of comparative analysis in Galaxy
  - edit `CLUSTER_TABLE.csv` (in `~/Desktop/data/histories/protocol_2`) and save as `CLUSTER_TABLE_corrected.csv`
  - upload `CLUSTER_TABLE_corrected.csv` and `COMPARATIVE_ANALYSIS_COUNTS.csv` to the Galaxy server
  - upload also `comparative_analysis_genome_sizes.txt` from `~/Desktop/data/examples/`
  - use the tool “RepeatExplorer Utilities -> Visualization of comparative clustering”

Running RepeatExplorer2 from a command line

Installation of RepeatExplorer2

This tutorial assume that you have working conda/mamba environment. Mambaforge was installed using following command:

```
# conda
curl -L -O
"https://github.com/conda-forge/miniforge/releases/latest/download/
Mambaforge-$(uname)-$(uname -m).sh"
bash Mambaforge-$(uname)-$(uname -m).sh
```

See miniforge repository for details (https://github.com/conda-forge/miniforge)

Installation of Singularity

```
mamba create -n singularity -c conda-forge singularity=3.6.3
```

Get RepeatExplorer2 Singularity image:

singularity image is available in https://github.com/repeatexplorer/repex_tarean/releases

```
conda activate singularity
cd
mkdir repeatexplorer && cd repeatexplorer
singularity pull
https://github.com/repeatexplorer/repex_tarean/releases/download/0.3.8/
repex_tarean_0.3.8.sif
singularity build  --sandbox repex_tarean repex_tarean_0.3.8.sif
# Verify build
singularity exec repex_tarean seqclust --help
```

Run clustering on test data:

```
# Download small pair-end dataset:
wget
https://bitbucket.org/petrnovak/repex_tarean/raw/devel/test_data/LAS_paired_1
0k.fas
# inspect data
head LAS_paired_10k.fas
seqkit stat LAS_paired_10k.fas
# run repeatexplorer
```

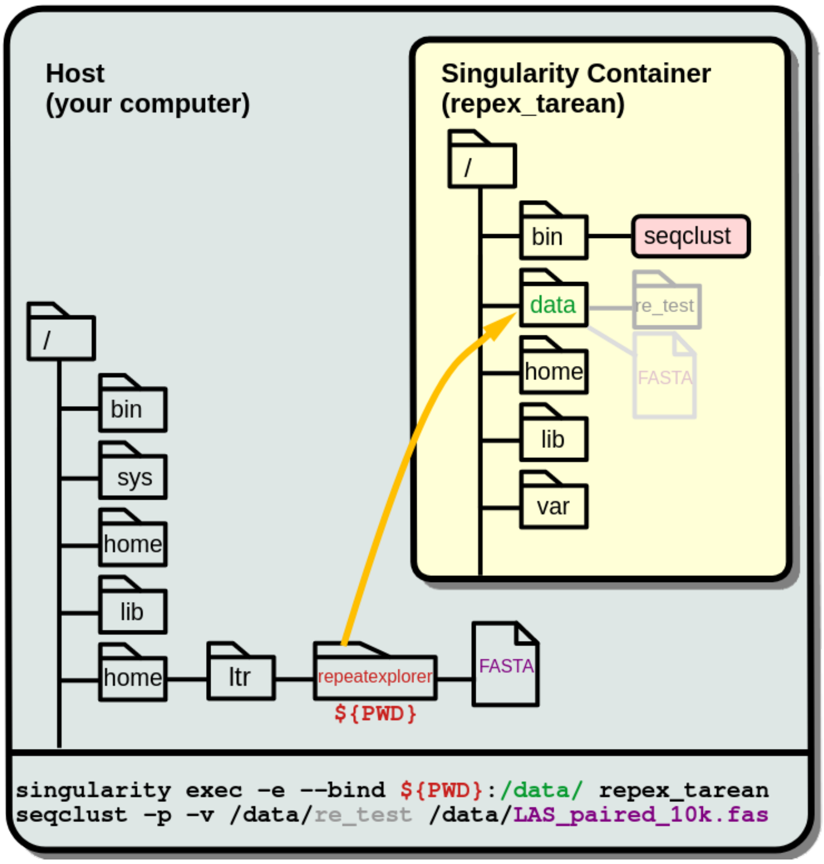
```
singularity exec -e --bind ${PWD}:/data/ repex_tarean  seqclust  -p -v
/data/re_test /data/LAS_paired_10k.fas
```

A Singularity container is a portable, self-contained unit that contains everything needed to run the program. `--bind ${PWD}:/data/` is creating a bridge between a folder on your system (`${PWD}`) and a folder inside the container (`/data/`). The syntax is `--bind src:dest`, where `src` is the source directory on the host system, and `dest` is the destination directory inside the container. The `${PWD}` part represents the current working directory you're in.

As an alternative to the above command, you can also make the paths the same inside and outside the container, like this:

```
singularity exec -e --bind ${PWD}:${PWD} repex_tarean  seqclust  -p -v
re_test /data/LAS_paired_10k.fas
```

In this setting, directory path will be the same in the singularity container and on the host. If there is a conflict in file paths (i.e., the same directory path exists both inside and outside of the container), the bind mount will overwrite the contents of the destination directory inside the container with the source directory from the host system. To avoid conflict, we bind unique directory path `/data` which is guarantied to be empty on `repex_tarean` singularity container.



```
singularity exec -e --bind ${PWD}:/data/ repex_tarean
seqclust -p -v /data/re_test /data/LAS_paired_10k.fas
```

Sharing data between a host and Singularity container:

Example of clustering including data pre-processing

Data:

Get data and check the quality of reads using FastQC program:

```
# Clustering example 1
cd ~/repeatexplorer
mkdir single_species
cd single_species
# Sample of T.cacao pair-end Illumina reads
wget
https://github.com/kavonrtep/example_data/raw/master/SRR089356_1.fastq.gz
wget
https://github.com/kavonrtep/example_data/raw/master/SRR089356_2.fastq.gz
fastqc *.fastq.gz
```

When FastQC run finishes, inspect html reports.

Quality filtering

Use **Trimmomatic** program to remove low quality reads and trim read ends. Trimmomatic will be used in paired-end mode. Trimmomatic options are:

```
TrimmomaticPE  [-threads threads] [-phred33 | -phred64] [-trimlog logFile] \
paired_output_1 unpaired_output_1 paired_output_2 unpaired_output_2
step_1 ...

The current trimming steps are:

- ILLUMINACLIP:<fastaWithAdaptersEtc>:<seed mismatches>:<palindrome clip
threshold>:<simple clip threshold>

- *fastaWithAdaptersEtc*: specifies the path to a fasta file containing all
the adapters, PCR sequences etc. The naming of the various sequences within
this file determines how they are used.

-  *seedMismatches*: specifies the maximum mismatch count which will still
allow a full match to be performed

- *palindromeClipThreshold*: specifies how accurate the match between the two
adapter ligated´ reads must be for PE palindrome read alignment.

- *simpleClipThreshold*: specifies how accurate the match between any adapter
etc. sequence must be against a read.

- SLIDINGWINDOW:<windowSize>:<requiredQuality>
       windowSize: specifies the number of bases to average across
       requiredQuality: specifies the average quality required.

- LEADING:<quality>
       quality: Specifies the minimum quality required to keep a base.

- TRAILING:<quality>
       quality: Specifies the minimum quality required to keep a base.

- CROP:<length>
       length: The number of bases to keep, from the start of the read.
```

```
- HEADCROP:<length>
       length: The number of bases to remove from the start of the read.

- MINLEN:<length>
       length: Specifies the minimum length of reads to be kept.

Trimming occurs in the order which the steps are specified on the command
line. It  is recommended  in  most  cases  that  adapter clipping, if
required, is done as early as possible.
```

```
# Adapter sequences are located in /usr/share/trimmomatic/
cp  /usr/share/trimmomatic/*.fa .

# Remove first 10 nt, min length must be 90
TrimmomaticPE -phred33 SRR089356_1.fastq.gz SRR089356_2.fastq.gz \
 SRR089356_1_clean.fastq.gz SRR089356_1_unpaired.fastq.gz \
 SRR089356_2_clean.fastq.gz SRR089356_2_unpaired.fastq.gz \
 ILLUMINACLIP:NexteraPE-PE.fa:2:40:15 SLIDINGWINDOW:4:10 CROP:100 HEADCROP:10
MINLEN:90

# Check statistics of fastq files:
seqkit stats *fastq.gz
# Run fastqc on clean data:
fastqc *clean*.fastq.gz
```

Sample to required number of reads:

```
# Paired end read sampling:
seqtk sample -s 10  SRR089356_1_clean.fastq.gz 5000 >
SRR089356_1_clean_sample.fastq
seqtk sample -s 10  SRR089356_2_clean.fastq.gz 5000 >
SRR089356_2_clean_sample.fastq
```

Interleaved pairs into single file:

```
# Make interleaved FASTQ
seqtk mergepe SRR089356_1_clean_sample.fastq SRR089356_2_clean_sample.fastq
> SRR089356_clean_sample_merged.fastq
# Convert to FASTA
seqtk seq -A SRR089356_clean_sample_merged.fastq >
SRR089356_clean_sample_merged.fasta
```

Run RepeatExplorer with default settings:

```
# Run clustering with default settings
cd ~/repeatexplorer
singularity exec -e --bind ${PWD}:/data/ repex_tarean  seqclust  -p -v
/data/re_output_run1 /data/single_species/SRR089356_clean_sample_merged.fasta
```

NOTE : current directory (`$PWD`) is /data directory in singularity container.

Command line options:

```
seqclust  [-h] [-p] [-A] [-t] [-l LOGFILE] [-m {float range 0.0..100.0}] [-M
{0,float range 0.1..1}] [-o {float range 30.0..80.0}] [-c CPU]
                [-s SAMPLE] [-P PREFIX_LENGTH] [-v OUTPUT_DIR] [-r
MAX_MEMORY] [-d DATABASE DATABASE] [-C] [-k] [-a {2,3,4,5}]
                [-tax
{VIRIDIPLANTAE3.0,VIRIDIPLANTAE2.2,METAZOA2.0,METAZOA3.0}]
                [-opt
{ILLUMINA,ILLUMINA_DUST_OFF,ILLUMINA_SENSITIVE_MGBLAST,ILLUMINA_SENSITIVE_BLA
STPLUS,OXFORD_NANOPORE}]
                [-D {BLASTX_W2,BLASTX_W3,DIAMOND}]
                sequences

RepeatExplorer:
    Repetitive sequence discovery and clasification from NGS data



positional arguments:
  sequences

optional arguments:
  -h, --help            show this help message and exit
  -p, --paired
  -A, --automatic_filtering
  -t, --tarean_mode     analyze only tandem reapeats without additional
                        classification
  -l LOGFILE, --logfile LOGFILE
                        log file, logging goes to stdout if not defines
  -m {float range 0.0..100.0}, --mincl {float range 0.0..100.0}
  -M {0,float range 0.1..1}, --merge_threshold {0,float range 0.1..1}
                        threshold for mate-pair based cluster merging,
                        default 0 - no merging
  -o {float range 30.0..80.0}, --min_lcov {float range 30.0..80.0}
                        minimal overlap coverage - relative to longer
                        sequence length, default 55
  -c CPU, --cpu CPU     number of cpu to use, if 0 use max available
  -s SAMPLE, --sample SAMPLE
                        use only sample of input data [by default max reads
                        is used]
  -P PREFIX_LENGTH, --prefix_length PREFIX_LENGTH
                        If you wish to keep part of the sequences name,
                        enter the number of characters which should be
                        kept (1-10) instead of zero. Use this setting if
                        you are doing comparative analysis
  -v OUTPUT_DIR, --output_dir OUTPUT_DIR
  -r MAX_MEMORY, --max_memory MAX_MEMORY
                        Maximal amount of available RAM in kB if not set
                        clustering tries to use whole available RAM
  -d DATABASE DATABASE, --database DATABASE DATABASE
                        fasta file with database for annotation and name of
                        database
  -C, --cleanup         remove unncessary large files from working directory
  -k, --keep_names      keep sequence names, by default sequences are renamed
  -a {2,3,4,5}, --assembly_min {2,3,4,5}
                        Assembly is performed on individual clusters, by
                        default clusters with size less then 5 are not
                        assembled. If you need assembly of smaller cluster
                        set *assembly_min* accordingly
  -tax {VIRIDIPLANTAE3.0,VIRIDIPLANTAE2.2,METAZOA2.0,METAZOA3.0}, --taxon
{VIRIDIPLANTAE3.0,VIRIDIPLANTAE2.2,METAZOA2.0,METAZOA3.0}
                        Select taxon and protein database version
  -opt
{ILLUMINA,ILLUMINA_DUST_OFF,ILLUMINA_SENSITIVE_MGBLAST,ILLUMINA_SENSITIVE_BLA
STPLUS,OXFORD_NANOPORE}, --options
{ILLUMINA,ILLUMINA_DUST_OFF,ILLUMINA_SENSITIVE_MGBLAST,ILLUMINA_SENSITIVE_BLA
STPLUS,OXFORD_NANOPORE}
```

```
                        ILLUMINA : standard option, all-to-all similarity
                        search is performed using mgblast, threshold for hits
                        is 90 percent identity over 55 percent of the
                        sequence length, word size is 18

                        ILLUMINA_SENSITIVE_MGBLAST : all-to-all search is
                        performed using mgblast, with  word size 8 and
                        threshold for hits is 80 percent identity over
                        55 percent of the sequence length

                        ILLUMINA_SENSITIVE_BLASTPLUS : all-to-all search is
                        performed using blastn,with  word size 6 and
                        threshold for hits is 80 percent identity over 55
                        percent of the sequence length

                        OXFORD_NANOPORE: experimental option, all-to-all
                        search is performed using lastal program
  -D {BLASTX_W2,BLASTX_W3,DIAMOND}, --domain_search
                        Detection of protein domains can be performed by
                        either blastx or diamond program. options are:
                          BLASTX_W2 - blastx with word size 2 (slowest, the
                          most sesitive)
                          BLASTX_W3 - blastx with word size 3 (default)
                          DIAMOND   - diamond program (significantly faster,
                          less sensitive)

                        To use this option diamond program must be installed
                        in your PATH
```

Running comparative analysis

Data:

```
# Get data from comparative analysis
cd ~/repeatexplorer
mkdir comparative
cd comparative
wget
https://github.com/kavonrtep/example_data/raw/master/SRR9938304_1.fastq.gz
wget
https://github.com/kavonrtep/example_data/raw/master/SRR9938304_2.fastq.gz
wget
https://github.com/kavonrtep/example_data/raw/master/SRR089356_1.fastq.gz
wget
https://github.com/kavonrtep/example_data/raw/master/SRR089356_2.fastq.gz
seqkit stats *.fastq.gz
```

Quality control and filtering:

```
fastqc *.fastq.gz
cp  /usr/share/trimmomatic/*.fa .
TrimmomaticPE -phred33 SRR089356_1.fastq.gz SRR089356_2.fastq.gz \
 SRR089356_1_clean.fastq.gz SRR089356_1_unpaired.fastq.gz \
 SRR089356_2_clean.fastq.gz SRR089356_2_unpaired.fastq.gz \
 ILLUMINACLIP:NexteraPE-PE.fa:2:40:15 SLIDINGWINDOW:4:10 CROP:100 HEADCROP:10
MINLEN:90

TrimmomaticPE -phred33 SRR9938304_1.fastq.gz SRR9938304_2.fastq.gz \
 SRR9938304_1_clean.fastq.gz SRR9938304_1_unpaired.fastq.gz \
 SRR9938304_2_clean.fastq.gz SRR9938304_2_unpaired.fastq.gz \
 ILLUMINACLIP:NexteraPE-PE.fa:2:40:15 SLIDINGWINDOW:4:10 CROP:100 HEADCROP:10
MINLEN:90
```

Sample to required coverage:

```
seqtk sample -s 10  SRR089356_1_clean.fastq.gz 5000 >
SRR089356_1_clean_sample.fastq
seqtk sample -s 10  SRR089356_2_clean.fastq.gz 5000 >
SRR089356_2_clean_sample.fastq

seqtk sample -s 10  SRR9938304_1_clean.fastq.gz 5000 >
SRR9938304_1_clean_sample.fastq
seqtk sample -s 10  SRR9938304_2_clean.fastq.gz 5000 >
SRR9938304_2_clean_sample.fastq
```

Interleave:

```
seqtk mergepe SRR089356_1_clean_sample.fastq SRR089356_2_clean_sample.fastq
seqtk > SRR089356_clean_sample_merged.fastq
seqtk mergepe SRR9938304_1_clean_sample.fastq SRR9938304_2_clean_sample.fastq
seqtk > SRR9938304_clean_sample_merged.fastq
# Convert to FASTA
seqtk seq -A SRR089356_clean_sample_merged.fastq >
SRR089356_clean_sample_merged.fasta
seqtk seq -A SRR9938304_clean_sample_merged.fastq >
SRR9938304_clean_sample_merged.fasta
```

Add prefix and concatenate :

```
# Add prefixes CA, CB
seqtk rename SRR089356_clean_sample_merged.fasta CA >
prefix_SRR089356_clean_sample_merged.fasta
seqtk rename SRR9938304_clean_sample_merged.fasta CB >
prefix_SRR9938304_clean_sample_merged.fasta
cat prefix* > CA_CB_final.fasta
```

Comparative clustering:

```
cd ~/repeatexplorer
singularity exec -e --bind ${PWD}:/data/ repex_tarean  seqclust --paired --
prefix_length 2  -v /data/re_output_comparative
/data/comparative/CA_CB_final.fasta
```

Specifying TEMP directory:

During its operation, RepeatExplorer produces a significant number of temporary files. It's important to ensure that the directory designated for these temporary files has sufficient storage capacity.

Singularity, by default, uses the `/tmp` directory on your host system to store temporary data, unless you specify otherwise. To adjust this default setting, you can define the `TEMP` variable and select your desired directory using the `--bind` option. This is demonstrated in the command below:

```
singularity exec --no-home --env TEMP=/tmp  --bind /mnt/tmp:/tmp  --bind $
{PWD}:/data/  repex_tarean  seqclust  -p -v /data/re_test
/data/LAS_paired_10k.fas
```

In this example, the `/mnt/tmp` directory on your system will serve as the storage area for temporary files. Please ensure that this directory has sufficient storage to accommodate the temporary files generated during the operation.

PRACTICAL TRAINING - DAY 2

TideCluster

(NOTE - the tool is still under development; see https://github.com/kavonrtep/TideCluster for the most recent version and updated info)

Running individual analysis steps separately with default parameters

- open terminal, switch to the tidecluster directory and activate conda environment:

```
cd ~/tidecluster/
conda activate tidecluster
```

- execute TideHunter

```
TideCluster.py tidehunter -f ~/Desktop/data/examples/CEN6_ver_220406.fasta -
pr CEN6_default -c 4
```

- initialize Integrative Genomics Viewer (IGV) and import TideHunter output gff3 file
  - Genomes -> Load Genome from File (~/Desktop/data/examples/CEN6_ver_220406.fasta)
  - File -> Load from File (~/tidecluster/CEN6_default_tidehunter.gff3 AND ~/tidecluster/CEN6_default_chunks.bed)
  - change CEN6_default_chunks.bed track to "Squished" (right-click and select "Squished") and CEN6_default_tidehunter.gff3 to "Expanded"
  - save the IGV session (File -> Save Session…)
- run similarity-based clustering of the identified tandem repeats

```
TideCluster.py clustering -f ~/Desktop/data/examples/CEN6_ver_220406.fasta -
pr CEN6_default -c 4
```

- inspect resulting gff3 file in IGV
  - File -> Load from File (~/tidecluster/CEN6_default_clustering.gff3)
- perform tandem repeat annotation using a reference database (should be RepeatMasker-formatted)

```
TideCluster.py annotation -pr CEN6_default -l
~/Desktop/data/examples/reference_db_SATELLITES_Fabeae.RM_format -c 4
```

- inspect resulting gff3 file in IGV
  - File -> Load from File (~/tidecluster/CEN6_default_annotation.gff3)
- update gff3 file to show the annotations
  - open “CEN6_default_annotation.tsv” in LibreOffice Calc, edit:
    - Data -> Sort -> Column C -> Descending; remove lines with values < 0.5
    - delete Column C
    - save as “CEN6_default_annotation_refDB.csv”
- run the script updating names of selected clusters in gff3 file

```
update_gff3.py -g CEN6_default_annotation.gff3 -t
CEN6_default_annotation_refDB.csv -o CEN6_default_annotation_refDB.gff3
```

- inspect resulting gff3 file in IGV
  - File -> Load from File (~/tidecluster/CEN6_default_annotation_refDB.gff3)
- run TAREAN to get consensus sequences

```
TideCluster.py tarean -f ~/Desktop/data/examples/CEN6_ver_220406.fasta -pr
CEN6_default -c 4
```

- inspect TAREAN summary “CEN6_default_tarean_report.html” (open in web browser)

Running all TideCluster steps automatically

In this example, we demonstrate automatic execution of all analysis steps, together with using custom settings to detect tandem repeats with short monomers.

- run the whole pipeline using the command:

```
TideCluster.py run_all -f ~/Desktop/data/examples/CEN6_ver_220406.fasta -pr
CEN6_short_monomers -l
~/Desktop/data/examples/reference_db_SATELLITES_Fabeae.RM_format -c 4 -T "-p
10 -P 39 -c 5 -e 0.25"
```

- inspect resulting gff3 files in IGV
  - File -> Load from File (~/tidecluster/CEN6_short_monomers_tidehunter.gff3 AND CEN6_short_monomers_annotation.gff3)
- *note that there was no TAREAN output generated as there was no tandem repeat so abundant to pass the minimal total length cutoff for this analysis (50 kb by default)*

Import additional information tracks to IGV

- File -> Load from File (~/Desktop/data/examples/CEN6_SAT_manual_annotation.gff)
- File -> Load from File (~/Desktop/data/examples/CEN6_CENH3_ChIP-seq_C1P23_C1K_bs200.bigwig)
- save IGV session
  - File -> Save Session…

Design of hybridization probes and primers for PCR

Design of oligonucleotide probe

- Inspect the TAREAN report for tidecluster repeat TRC_15
  - open ~/tidecluster/CEN6_default_tarean_report.html in web browser
  - check graph shape, k-mer coverage score, k-mer based graph, and sequence logo
- open terminal and switch to the probe directory

```
cd ~/probes/
```

- copy monomer and dimer consensus sequences of TRC_15 from the output of tidecluster

```
cp ~/tidecluster/CEN6_default_tarean/TRC_15.fasta_tarean/consensus.fasta
TRC_15_consensus.fasta

cp
~/tidecluster/CEN6_default_tarean/TRC_15.fasta_tarean/consensus_dimer.fasta
TRC_15_consensus_dimer.fasta
```

- Select only those consensus sequences that have k-mer coverage score ≤ 0.1 (first 8 sequences; see ~/tidecluster/CEN6_default_tarean/TRC_15.fasta_tarean/report.html)

```
seqkit head -n 8 TRC_15_consensus_dimer.fasta >
TRC_15_consensus_dimer_f0.1.fasta
```

- Compare the sequences using dotter

```
dotter TRC_15_consensus_dimer_f0.1.fasta TRC_15_consensus_dimer_f0.1.fasta
```

- Select the first, i.e. the most representative consensus sequence (monomer and dimer)

```
seqkit head -n 1 TRC_15_consensus.fasta >
19_1_sc_0.764035_l_1213_monomer.fasta

seqkit head -n 1 TRC_15_consensus_dimer.fasta >
19_1_sc_0.764035_l_1213_dimer.fasta
```

- Compare the best monomer sequence with the entire arrays of TRC_15

```
dotter 19_1_sc_0.764035_l_1213_monomer.fasta
~/tidecluster/CEN6_default_tarean/fasta/TRC_15.fasta
```

- Identify regions that are not suitable for probes (inverted repeats and short simple repeats)
    - Manually using dotter

```
dotter 19_1_sc_0.764035_l_1213_monomer.fasta
19_1_sc_0.764035_l_1213_monomer.fasta
```

    - Find inverted repeats using einverted (program of EMBOSS package)

```
cat 19_1_sc_0.764035_l_1213_monomer.fasta | einverted -filter -threshold 0
```

    - Find low-complexity sequences using dustmasker

```
dustmasker -level 30 -in 19_1_sc_0.764035_l_1213_monomer.fasta
```

- Design oligonucleotide probes using perl script
    - Print help for the perl script

```
~/Desktop/data/scripts/Design_probe_using_TAREAN_ppm_01.pl -h
```

    - Copy the input csv file from the TAREAN folder

```
cp ~/tidecluster/CEN6_default_tarean/TRC_15.fasta_tarean/ppm_19mer_1.csv ./
```

    - Run the script; Example 1: get all oligonucleotides in the specified range of length (50-60 nt) and print the output on the screen. The coordinates in the parameter “-I” are from einverted and dustmasker.

```
cat ppm_19mer_1.csv |
~/Desktop/data/scripts/Design_probe_using_TAREAN_ppm_01.pl -l 50 -L 60 -F 50
-N 0.390 -I "258-293 367-332 722-861 875-906 1201-1211"
```

    - Run the script; Example 2: get all oligonucleotides in the specified range of length (50-60 nt) and print the output to a file.

```
cat ppm_19mer_1.csv |
~/Desktop/data/scripts/Design_probe_using_TAREAN_ppm_01.pl -l 50 -L 60 -F 50
-N 0.390 -I "258-293 367-332 722-861 875-906 1201-1211" >
19_1_sc_0.764035_l_1213_probes50-60.tsv
```

    - Run the script; Example 3: get all oligonucleotides in the specified range of length

(50-60 nt) and Tm (51-53 °C), print the output on the screen, and sort it based on the Coverage_score

```
cat ppm_19mer_1.csv |
~/Desktop/data/scripts/Design_probe_using_TAREAN_ppm_01.pl -l 50 -L 60 -F 50
-N 0.390 -t 51 -T 53 -I "258-293 367-332 722-861 875-906 1201-1211" | grep -v
"Ignore!" | sort -k3,3n
```

  - Select the last oligonucleotide sequence from the Example 3 and save it as 19_1_sc_0.764035_l_1213_selected_probe.fasta
    - use shift-ctrl-c to copy the sequence to the clipboard
    - create and open the file 19_1_sc_0.764035_l_1213_selected_probe.fasta

```
nano 19_1_sc_0.764035_l_1213_selected_probe.fasta
```

    - use shift-ctrl-v to paste the sequence into the file
    - manually add the name line on the first row: “>19_1_sc_0.764035_l_1213_selected_probe”
    - use ctrl-o to save the file and ctrl-x to close the file

- Verify that the probe has no similarity to the other tandem repeats
  - Get the sequences of all tandem repeat arrays from tidecluster

```
cat ~/tidecluster/CEN6_default_tarean/fasta/TRC_*.fasta >
TRC_arrays_all.fasta
```

  - Make blast database

```
makeblastdb -in TRC_arrays_all.fasta -input_type fasta -dbtype nucl
```

  - Get names of TCR_15 array sequences TCR15

```
cat ~/tidecluster/CEN6_default_tarean/fasta/TRC_15.fasta | grep ">"
```

  - Run blastn with and without subsequent filtering of TRC_15 array sequences

```
blastn -db TRC_arrays_all.fasta -query
19_1_sc_0.764035_l_1213_selected_probe.fasta -word_size 20 -outfmt 6

blastn -db TRC_arrays_all.fasta -query
19_1_sc_0.764035_l_1213_selected_probe.fasta -word_size 20 -outfmt 6 | grep -
v -P "CEN6_ver_220406_87863497_87875968|CEN6_ver_220406_87897437_87941877"
```

- For probes ≤ 50, estimate the Tm using dnaMATE

    - Use the perl script to extract and analyze all 35-50mers

```
cat ppm_19mer_1.csv |
~/Desktop/data/scripts/Design_probe_using_TAREAN_ppm_01.pl -l 35 -L 50 -F 0 -
N 0.390 -I "258-293 367-332 722-861 875-906 1201-1211" | sort -k1,1n >
19_1_sc_0.764035_l_1213_probes35-50.tsv
```

    - Prepare input for dnaMATE (cut -f2 selects the 2nd column; grep -v removes the first line which contains the word "Sequence")

```
cat 19_1_sc_0.764035_l_1213_probes35-50.tsv | cut -f2 | grep -v "Sequence" >
19_1_sc_0.764035_l_1213_probes35-50_for_dnaMATE.txt
```

    - Run dnaMATE (Note: -o is a probe concentration [M] and -s is a salt concentration [M]. dnaMATE is also available at http://melolab.org/dnaMATE/tm-pred.html.)

```
dnaMATE 19_1_sc_0.764035_l_1213_probes35-50_for_dnaMATE.txt -s 0.39 -o
0.000001 > 19_1_sc_0.764035_l_1213_probes35-50_dnaMATE_output.tsv
```

    - Merge the outputs of the perl script and dnaMATE horizontally using "paste" command. The output will be a tab-delimited file which can be inspected e.g. in LibreOffice Calc or Excel.

```
paste 19_1_sc_0.764035_l_1213_probes35-50.tsv
19_1_sc_0.764035_l_1213_probes35-50_dnaMATE_output.tsv >
19_1_sc_0.764035_l_1213_probes35-50_mergedTm.tsv
```

Design of PCR primers using primer3

- In web browser, go to https://primer3.ut.ee/

    - As an input sequence use the **dimer** consensus sequence in the file ~/probes/19_1_sc_0.764035_l_1213_dimer.fasta
    - It is convenient to use saved settings. Compare default and modified settings at ~/Desktop/data/examples/primer3/
        - to push primer3 to design primers in a dimer, all TEMPLATE_MISPRIMING values must be set to -1
        - to amplify entire monomer, the PRIMER_PRODUCT_SIZE_RANGE should be set close to the monomer size
        - Minimum, maximum, and optimum Tm should match the PCR conditions
    - Upload the settings (use ~/Desktop/data/examples/primer3/primer3_settings_01.txt)
    - If you need to place the primers into a particular region, set either SEQUENCE_INCLUDED_REGION or SEQUENCE_EXCLUDED_REGION

        - examples are provided in ~/Desktop/data/examples/primer3/primer3_settings_01.txt
    - When the form is filled, click on “Pick Primers”
    - Notes:
        - Never use the PCR product(s) amplified from genomic DNA as a probe for hybridization (even in cases where there is a single product of the expected size). All probes should be cloned and sequenced.
        - If the cloned sequence contains short sequence repeats, these should be avoided in the probe.
        - Check the similarity of the probe with other repeats to avoid misinterpretation of results.

Annotation of repeats in genome assembly

Input data

The assembly of a 177.6 Mb region of chromosome 6 of *Pisum sativum* including its 81.6 Mb centromere (CEN6) and adjacent chromosome arms

Import input data

1. Import data to a new history using *Shared Data →Data Libraries → workshop_2023_DATA*. Select dataset *CEN6_ver_220406.fasta* from the table above and click on *Export to History → as Datasets*. Create a new history with the name “Assembly Annotation”.
2. Click on the green popup window in upper right corner to switch to newly created history

Identification of TE protein domains using DANTE

3. Run detection of TE conserved domains using DANTE→*Domain based ANnotation of Transposable Elements – DANTE*

Input Parameter	Value
Choose the type of sequence data	Fasta
Sequences in fasta format	CEN6_ver_220406.fasta imported in step 1
Select taxon and protein domain database version (REXdb)	Viridiplantae_version_3.0
Select scoring matrix	BLOSUM80
Run iterative search	No

Run time ~ 10 minutes

This tool creates three output datasets:

- *DANTE on data 1, full output:* annotation in the GFF3 format.
- *DANTE on data 1, filtered output*: annotation in the GFF3 format with low quality hits filtered out (filtering criteria are minimum 35% identity, 45% similarity, alignment length at least 80% over database sequence and maximum 3 interruptions per 100 amino acids)
- *DANTE on data 1, protein domains, filtered output :* FASTA with protein sequences

4. Summarize DANTE output using *DANTE → Summarize gff3 output from DANTE*

Input Parameter	Value
Input GFF	DANTE on data 1, filtered output
select categories to summarize	Classification

Identification of full-length LTR transposable elements

5. Run *Assembly Annotation Tools→DANTE_LTR retrotransposon identification*

Input Parameter	Value
GFF3 output from DANTE pipeline - full output	DANTE full output from step 3
Reference sequence matching DANTE output	CEN6_ver_220406.fasta imported in step 1
Maximum number of missing protein domains to tolerate in full length retrotransposon	1

Run time ~ 16 minutes

Output from DANTE_LTR contains the LTR retrotransposon annotation in the GFF3 format and also summary table. Detected elements are divided into 5 groups (Ranks):

Rank	Annotation
DLTP	Elements with identified protein **D**omains, **L**TRs, **T**SD and **P**BS
DLP	Elements with identified protein **D**omains, **L**TRs and **P**BS (TSD was not found)
DLT	Elements with identified protein **D**omains, **L**TRs and **T**SD (PBS was not found)
DL	Elements with protein **D**omains, **L**TRs (PBS and LDS were not found)
D	Elements with protein **D**omains from the same lineage, no LTRs detected

Please be aware that the subsequent two steps are time-intensive. To expedite this process, we will bypass them and load the history containing pre-calculated results from steps 6 to 7 To do so, navigate to "Shared Data" and then to "Histories." From there, import the history titled "*Workshop 2023 - Assembly Annotation Complete.*" by clicking on **+** symbol in the top right corner.

6. (DO NOT RUN) Filter out potentially chimeric elements using *Assembly Annotation Tools→DANTE_LTR retrotransposons filtering*:

Input Parameter	Value
GFF3 output from DANTE_LTR retrotransposon identification pipeline	LTR retrotransposons annotation in GFF3 format from previous step.
Reference sequence matching input GFF3	CEN6_ver_220406.fasta imported in step 1
Run time ~ 60 minutes	

This step creates several datasets:

Dataset	Note
Validated LTR retrotransposons annotation (GFF3)	Annotation in GFF3 format
Non-redundant library of LTR retrotransposons (FASTA)	Sequences of LTR retrotransposons in FASTA format with redundant sequences removed
Library of LTR retrotransposons (FASTA)	Sequences in FASTA format to be used for library based assembly annotation. This dataset will be used in the next step for library based assembly annotation
Library of 5'LTRs (FASTA)	
Library of 3'LTRs (FASTA)	
LTR retrotransposons lengths summary	Graphical summary

Using full-length LTR retrotransposons to annotate genome assembly

7. (DO NOT RUN) The full-length LTR retrotransposons in FASTA format from the previous step will be used as a custom library to annotate retrotransposons in the genome assembly. Select the tool *Assembly Annotation Tools→Library Based Assembly Annotation*:

Input Parameter	Value
Genome/Assembly to annotate	CEN6_ver_220406.fasta imported in step 1
RepeatExplorer based Library of Repetitive Sequences	Library of LTR retrotransposons (FASTA) from previous step
Sensitivity	Default sensitivity
Run time ~ 3 hours	

Cleaning up LTR retrotransposons annotation

Given that certain satellites are derived from LTR retrotransposons, there may be inaccuracies in the annotation of some satellite regions due to their similarity to some LTR retrotransposons from the library. Therefore, in the following steps, we will refine our LTR retrotransposon annotations by removing regions that overlap with tandem repeat annotations identified by TideHunter.

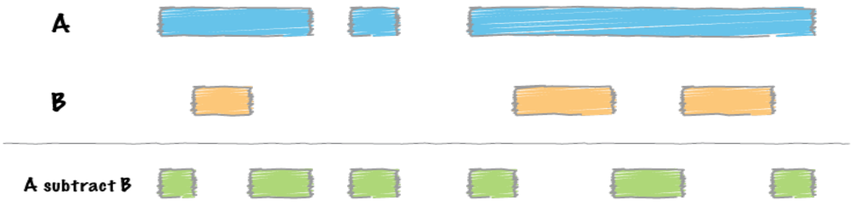
8. Upload annotation of tandem repeats from TideHunter. Use file *~/tidehunter/CEN6_default_tidehunter.gff3*

9. Remove overlapping LTR retrotransposons annotation using the tool: *BED → bedtools SubtractBed:*

Input	Value
BED/bedGraph/GFF/VCF/EncodePeak file (-a)	Repeat Annotation (GFF3) on data 10 and data 1 from step 7
BED/bedGraph/GFF/VCF/EncodePeak file (-b)	CEN6_default_tidehunter.gff3

Keep other parameters in default settings

This tool subtracts *B* track from the *A* track:



10. Calculate annotation summary from GFF3 file using the tool *Assembly Annotation Tools → Create summary on GFF3 attribute:*

Input	Value
Input GFF	bedtools SubtractBed on data 16 and data 14
Name of attribute to summarize	Name

Data download and visualization in IGV

11. Download resulting summary table from previous step to *~/dante_ltr/te_annotation_summary.csv*

12. Download all resulting GFF3 outputs to directory *~/dante_ltr*. Click on *search datasets window* on top of history panel and type GFF3 and enter. This will show only GFF3 dataset. For each dataset click on download icon and navigate top *~/dante_ltr* directory. Use following filenames:

Dataset	New File Name
DANTE on data 1, full output	DANTE_full.gff3
DANTE on data 1, filtered output	DANTE_filtered.gff3
LTR retrotransposons annotation (GFF3) based on DANTE annotation 2 and reference 1	DANTE_LTR_full.gff3
Validated LTR retrotransposons annotation (GFF3) based on annotation 6 and reference 1	DANTE_LTR_filtered.gff3
Repeat Annotation (GFF3) on data 10 and data 1, cleaned gff	RM_DANTE_LTR.gff3
bedtools SubtractBed on data 16 and data 14	RM_DANTE_LTR_clean.gff3

13. Calculate the genomic proportion of individual lineages of LTR retrotransposons from *~/dante_ltr/te_annotation_summary.csv* file:
    - Open *~/dante_ltr/te_annotation_summary.csv* in Libreoffice Calc
    - Delete all columns except *Name* and *total_length*
    - Label the third column *Genomic Proportion* and bellow fill in the formula to calculate the genomic proportion as *total_length/genome_size*. Use genome size 177,603,725

14. Visualize GFF3 tracks in IGV together with tandem repeat annotations.
    - Open IGV
    - Load previous session with satellite annotation from `~/tidecluster/igv_session.xml`
    - For clarity, delete all tracks except:
        - `CEN6_CENH3_ChIP-seq_C1P23_C1K_bs200.bigwig`
        - `CEN6_default_tidehunter.gff3`
        - `CEN6_SAT_manual_annotation.gff`
        - `CEN6_default_annotation_refDB.gff3`
    - Load all additional GFF3 tracks from `~/dante_ltr` directory

15. Explore these genomic locations:
    - CEN6_ver_220406:132,000,000-136,000,000
    - CEN6_ver_220406:132,746,518-132,789,180

PRACTICAL TRAINING - DAY 3

REPET pipeline

- https://github.com/repeatexplorer/workshop

Preparing Illumina-like PE input data from long reads

- NOTE – at present, only PacBio HiFi reads have error rate small enough to allow efficient clustering with RepeatExplorer
- copy example data to your history at the Galaxy server:
    - Shared Data → Data Libraries → workshop_2022_DATA → HiFi reads example (P. sativum 1x)
- analyze read length distribution:
    - FASTA manipulation -> Compute sequence length
    - Visualization -> Histogram of a numeric column (plot as counts)
- sample long reads to get 0.1x genome coverage (1C=4300 Mb)
    - Experimental Tools -> Create sample of long reads [set Total length to 430,000,000]
- extract Illumina-like paired end reads, sample 0.1x coverage (→ final coverage will be 0.01x):
    - Experimental Tools -> Get pseudo short paired end reads from long reads [Insert_length: 800, read_length: 200, coverage: 0.1]
- OPTIONAL: set up RepeatExplorer2 clustering:
    - RepeatExplorer2 → RepeatExplorer2 clustering [use all reads and select basic queue]