# README - Instructions for set up and use of LVQ-KNN

**Authors:** Ariane Belka, Dirk Hoeper, Mareike Fischer, Martin Beer, Anne Pohlmann


If you use this program/method to create data/results in any publication please cite:

> LVQ-KNN: Composition-based DNA/RNA classification of short nucleotide sequences utilizing a prototype-based k-nearest neighbor approach. 2018. A. Belka, M. Fischer, A. Pohlmann, M. Beer and D. Höper.(doi...)

## 1 SOFTWARE DEPENDENCIES

1. *EMBOSS/compseq:* for computation of oligonucleotide frequency rates
2. *R and packages:* for creating data and prototype sets and for classification
   (a) "seqinr" and all dependencies
   (b) "class" and all dependencies
3. *Perl and packages:* for creating prototype sets via LVQ (if chosen; faster than LVQ.R)
   (a) List::Util
   (b) Time::Duration
4. *Operating System:* Linux (validated with CentOS 6/7)

## 2 SET UP

1. unzip L-KNN archive
2. change paths "path-name" in Config.txt and Main.sh to corresponding software tools and the program
3. change path to Config.txt in *.sh files (* wildcard)

## 3 DATA STRUCTURE

1. If reference sequences are used:
   The name of the reference sequences (refseqs) should contain the words "Bacteria", "DNA" or "RNA" to generate the coding information. The sequences have to be in fastA format, the files need the file extention: *.fasta or *.fna, e.g.:

$Refseq Bacteria.fasta$, containing all bacterial refseqs

$Refseqds DNA.fasta$, containing all double-stranded DNA virus refseqs

$Refseqss\text{-}RNA.fasta$, containing all negative-sense single-stranded RNA virus refseqs,

...

2. If refseq oligonucleotide information is provided:
   The data frame need the following structure, tab-separated, decimal mark '.' (e.g. for dinucleotides):

| | AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT | DoRNA |
|---|---|---|
| ref1 | ..... | 1 |
| ref2 | ..... | 2 |
| ... | | |

   NOTE: 1 = DNA, 2 = RNA; In order to avoid errors, the data frame should be ordered by class-coding. File name and extention: e.g. $training * .txt$

3. If prototype datasets are provided:
   The data frame should have the same structure as for references in 2. File name and extention: e.g. $proto\_di\_ls\_10\_apk\_500.txt$

4. If test sequences are used:
   The sequences have to be in fastA format, the file need the file extention: *.fasta or *.fna

5. If test sequence oligonucleotide information is provided:
   The data frame need the following structure, tab-separated, decimal mark '.' (e.g. for dinucleotides):

| | AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT |
|---|---|
| query1 | ..... |
| query2 | ..... |
| ... | |

   File name and extention: e.g. $test * .txt$

   NOTE:

   The prototype and the query sequence information need to be the same (e.g. both dinucleotides), otherwise the classification fails.

   Training oligonucleotide file names: $training * .txt$

   Test oligonucleotide file names: $test * .txt$

## 4 USAGE

If you want to do the whole analysis (training data, prototypes, test data and classification) run Main.sh with your specified parameter settings. Alternatively both scripts can be run individually.

1. Whole analysis program running, using ./Main.sh

   -o working directory, where the data of the analysis should be stored

   -p project name of the current analysis (default: analysis_output)

   -t should a training dataset be created (default: y)

   if y: reference sequences files are copied to /wdir/projectname/trainingdata

   if n: reference oligonucleotide information file is copied to /wdir/projectname/trainingdata

   -q should a query dataset be created (default: y)

   if y: test sequences files are copied to /wdir/projectname/testdata

   if n: test oligonucleotide information file is copied to /wdir/projectname/testdata

   -l should prototype datasets be computed (default: y)

   if y: LVQ method is employed to trainingdata to compute prototypes

   if n: prototype dataset should be stored in /wdir/projectname/prototypes by the USER or can be delivered by option -r

   -k should classification results be created (default: y)

   if y: K-NN method is employed to testdata and prototype sets to classify the testdata

   if n: Nothing is done.

   -r path to the reference file/s or the prototype set

   -u path to query files

   -c which oligonucleotide should be used for classification (default: 2)

   **EXAMPLES:**

   Main.sh -o /Analysis/ -p Test -t y -l y -r /References/ -u /Queries/ -c 3

   The analysis results are stored in /Analysis/Test/ (-o and -p).

   The training dataset and prototypes are created/computed (-t y and -l y).

   The analysis is based on trinucleotides (-c 3).

   The classification is running by default (-q y and -k y).

2. Training dataset creation and Prototype dataset computation with LVQ using ./Prototype-computing/training_prototypes.sh

   -o working directory, where the data of the analysis should be stored

   -p project name of the current analysis

   -t should a training dataset be created

   if y: refseqs files are copied to /wdir/projectname/trainingdata

   if n: reference oligonucleotide information file is copied to /wdir/projectname/trainingdata

   -l should prototype datasets be computed

   if y: LVQ method is employed to trainingdata to compute prototypes

   if n: prototype dataset should be stored in /wdir/projectname/prototypes by the USER or can be delivered by option -f

-f path to the reference file/s or the prototype set

-c which oligonucleotide should be used for classification

**EXAMPLES:**

training_prototypes.sh -o /Analysis/ -p Test -t y -l y -f /References/ -c 3

The Analysis results are stored in /Analysis/Test/ (-o and -p).

The training dataset and prototypes are created/computed (-t y -l y).

The analysis is based on trinucleotides (-c 3).

training_prototypes.sh -o /Analysis/ -p Test -t n -l y -f /References/

Only prototypes are computed, training dataset already exist(-t n -l y).

The analysis is based on dinucleotides, because -c is default 2.

3. Test dataset creation and K-NN classification using ./Prototype-computing/test-classification.sh

-o working directory, where the data of the analysis should be stored

-p project name of the current analysis

-t should test datasets be created

if y: test sequences files are copied to /wdir/projectname/testdata

if n: test oligonucleotide information file is copied to /wdir/projectname/testdata

-k should classification results be created

if y: K-NN method is employed to testdata and prototype sets to classify the testdata

if n: Nothing is done.

-f path to the test file/s

-c which oligonucleotide should be used for classification

**EXAMPLES:**

test-classification.sh -o /Analysis/ -p Test -t y -k y -f /Testfiles/ -c 3

The Analysis results are stored in /Analysis/Test/ (-o and -p).

The test dataset and the classification are created (-t y, -k y).

The analysis is based on trinucleotides (-c 3).

test-classification.sh -o /Analysis/ -p Test -t n -k y -f /Testfiles/

Only classification is executed, test dataset already exist(-t n, -l y).

The analysis is based on dinucleotides, because -c is default 2.

Have fun with the program. If any errors occur do not hesitate to report them to

ariane.belka@fli.de

PROGRAM - FILE - STRUCTURE:

- LVQ-KNN (dir)
  - k-NN-classification (dir)
    - k-nn.R
    - results_compact.R
    - test-classification.sh
    - testdata.R
  - Prototype-computing (dir)
    - LVQ.pl
    - LVQ.R
    - training_prototypes.sh
    - trainingdata.R
  - Config.txt
  - Hilfsprogramme.R
  - input_info.txt
  - Main.sh
  - README
  - README.pdf