Protein List Comparator (ProLiC)

Quick reference guide, ProLiC version 1.2

1. General remarks and installation

This is a quick reference guide for the utilization of the Protein List Comparator (ProLiC) software. ProLiC is a tool for the comparison of an arbitrary number of files that report results of Proteomics experiments.

In this guide, the properties of ProLiC are described very briefly. We will not go into details, but rather we confine ourselves to the more technical aspects required for installation and usage of ProLiC. However, the necessary information that allows full control of the comparison procedure is given. For a more detailed explanation of the algorithms or for a comparison of ProLiC with similar tools please refer to the ProLiC paper that is currently in preparation. This manual will be updated with the bibliographic information immediately after publication of the manuscript.

In the meantime, please contact the author via the contact information given in section 1.2 if you have further questions.

1.1 List of specific Terms and main Abbreviations

- BSD: Berkeley Software Distribution
- de.NBI: German Network for Bioinformatics Infrastructure
- KNIME: The Konstanz Information Miner
- MIT: Massachusetts Institute of Technology
- MS: Mass spectrometry
- NW: Needleman-Wunsch algorithm
- PID: Percent identity
- PTM: Post translational modification

1.2 Contact information

Dr. Michael Kohl

Group Leader Algorithmic Proteomics

Medizinisches Proteom-Center

Research unit Medical Bioinformatics

Building ZKF E.142 | Universitätsstrasse 150 | D-44801 Bochum

Phone: +49 (0)234 32-25999

Fax: +49 (0)234 32-14554
E-mail: michael.kohl@rub.de
Web site: www.medizinisches-proteom-center.de

1.3 System requirements

Because ProLiC is completely written in Java, the software can be utilized with a variety of operating systems including Microsoft Windows, Mac OS X, Linux and Solaris. It should be possible to run ProLiC on almost every computer if Java 1.7 or higher is installed. However, it is recommend using the latest version of Java.
Concerning RAM requirements, usage of newer personal computers with at least 3 GB RAM should be sufficient in order to carry out comparisons with ProLiC. However, the processing of very big datasets may require more memory and as a consequence the usage of Java 64 bit versions and 64 bit versions of the operating system is required.
Sequence based comparison is the computationally most expensive ProLiC approach. It is therefore (strongly) recommended to restrict the usage of this strategy to comparison of smaller datasets.
Depending on the language settings of your computer, ProLiC supports both English and German output for console messages. Installation of ProLiC is detailed in section 3 of this reference guide.

1.4 License

1.5 Further remarks

ProLiC takes advantage from the following resources (Java libraries / frameworks / source code):

- JGraphT (https://github.com/jgrapht/jgrapht)
- BioJava (http://biojava.org/),
- Log4j 2.3 (http://logging.apache.org/log4j/2.x/download.html)
- IniReader.java, Source code obtained from Stefan Matthias Aust.

Required BioJava libraries from previous releases are part of the ProLiC download.

2. Introduction

Data acquired by high-throughput omics technologies and both the availability and popularity of public repositories enable dissemination of molecular knowledge within the scientific community. For the first time, acquiring real insights into the functioning of the cell as a system sets seams possible by integration of compatible data sets. However, data integration/data comparison is challenging. In order to facilitate this task at least for the proteomics research area, the ProLiC software was developed. The software allows comparison of proteomics identification results. Three comparison methods are supported, namely an accession based, a protein sequence based and a peptide based comparison. These techniques differ in both their level of abstraction, the used information and the computational requirements (i.e. runtime and memory consumption). The accession based method compares two or more lists of accession numbers which were obtained from Proteomics databases. This approach needs no further information but requires at least the usage of the same type of protein accession in each list.

The (protein) sequence based method compares proteomics results on the protein sequence level. In contrast to the comparison of accessions this approach allows also detection of slightly different protein sequences (e.g. protein isoforms). However, ProLiC uses a global sequence alignment approach, which is characterized by high computational requirements. Therefore this opportunity is limited to a few and small data sets.

The peptide based method compares peptide sets. Typically, peptides are identified using a software for the analysis of mass spectrometry data (so-called search engines). Afterwards a peptide set is assigned to a specific protein.

Application of the protein sequence or the peptide based comparison methods leads to definition of protein groups, because of protein of list A may considered similar to more than one protein of list B.

We strongly recommend usage of the peptide based approach, because in this case the most basic information obtained from the mass spectrometry measurements is used. Furthermore, the approach is capable of identifying more similarities than any of the other two methods.

The ProLiC comparison method can be selected via the *compType* parameter of the section *commandLineSettings* in the ProLiC configuration file (cf. Appendix, paragraph a) of this manual).

3. Installation and Usage of ProLiC

ProLiC does not require an extensive installation process. Just unzip the archive in an arbitrary folder and adapt the configuration file (cf. Appendix a) of this manual). For usage of ProLiC, please open the default terminal application of your system (e.g. the 'Command Prompt' application of Microsoft Windows). Change the current directory and go to the directory where ProLiC has been unzipped. Within the terminal application, ProLiC can be started with the command 'java –jar ProLiC.jar'. Because ProLiC is a command line application it is possible to start the software using scripts (for example as part of scheduled tasks).

4. The Format of ProLiC Input Files

ProLiC considers all files with the specified file extension (e.g. .txt or .csv files) located in the input folder as input files for protein list comparison. Both the file extension of the input files and the input folder are settings of the configuration file (cf. Appendix, paragraph a) of this manual).
Currently, ProLiC supports a simple ASCII input (.csv or tab separated) format. Table 1 gives the column names and a description of the data that is stored in these columns as well as the information whether a column is mandatory or not. The first line of an input file is the header line. All information related to a protein is given in a single row of the input file.

Table 1: Header information for the ProLiC input file

Column number	Column header	Description	Mandatory?
1	Accession	(Protein) accession number	mandatory
2	BiomoleculeType	Specifies the type of the considered biological molecule	mandatory

Column number	Column header	Description	Mandatory?
3	Sequence	Sequence information (e.g. protein or gene sequence)	mandatory for sequence based comparison and for some variants of the peptide based comparison.
4-n	NA (= no header)	Peptide sequence(s)	mandatory for peptide based comparison

Please note that for the BiomoleculeType column 'Protein' is currently the only supported entry. Further versions of ProLiC will also support analysis of genomics/transcriptomics data. Starting with column number four the peptide sequences of a specific protein are given. Though this format is quite simple, it is indispensable to observe the specifications given above in order to ensure a consistent usage of the software. The order of the columns is of particular importance. ProLiC is shipped with example data that can be used in order to illustrate the input file format.

5. The ProLiC Result File

The result of a ProLiC protein list comparison is a single text file, which gives the calculated intersections and set differences in a very simple way. Currently the generation of the header of the output data file is rather static. Further versions of ProLiC that consider different biomolecule types (e.g. transcripts) need a more sophisticated preparation of the header, because there is maybe no protein related information. However, the columns, which should be considered within the result file, can be already specified to some extent with the settings in the configuration file (section *genericSettings* of the configuration file, cf. Appendix, paragraph a) of this manual).

5.1 Structure of the ProLiC result file

Within the ProLiC result file the calculated intersections and set differences are marked by a header line, which gives the names of the protein lists involved. These names are obtained from the names of the input files, because it is assumed that the file names are identifiers of a specific protein list. In case of the set differences the header contains the name of a single protein list only.

The next line below this header contains at least the column headings in the following order:

- *1st column* (heading: 'Overlap_Komplement'): This column gives the identifier of the overlap or the set difference, respectively
- *2nd column* (heading: 'BG GroupId'): a unique identification given as consecutive (integer) number
- *3rd column* (heading: 'BG Representative'): the accession number of the biomolecule, which is selected by ProLiC as biomolecule group representative
- *4th column* (heading: 'Accession'): the accession number of the biomolecule group member (currently a protein that is part of this biomolecule group)
- *5th column* (heading: 'Sequence'): the protein sequence for the protein with the accession number specified in the 4th column.
- *6th column* (heading: 'Original List'): the name of the protein list (i.e. the name of the file) that originally contains the information of the protein with the accession number specified in the 4th column.

Here is an example of this structure for an overlap of three lists (List A, C, D) as depicted by the ProLiC result file:

List_A ∩ List_C ∩ List_D[1]

Overlap_Komplement	BG GroupId	BG Representative	Accession	Sequence	Original List
List_A ∩ List_C ∩ List_D	7	Protein_A	Protein_A	AAA	List_A
List_A ∩ List_C ∩ List_D	7	Protein_A	Protein_C	CC	List_C
List_A ∩ List_C ∩ List_D	7	Protein_A	Protein_D	DD	List_D
List_A ∩ List_C ∩ List_D	7	Protein_A	Protein_A	AAA	List_D

Explanation of this overlap:

This overlap contains a single biomolecule group with the group identifier '7', which is a unique number with respect to the biomolecule groups reported in the result file of this comparison. Biomolecule group 7 contains 4 biomolecules that are considered 'identical/similar'. For each of these biomolecules a single row gives the related information. 'Protein_A' is selected as representative of the biomolecule group, because the parameter *selection_representative* is set to *longestSequence* (cf. Appendix, paragraph a) of this manual): The protein sequence of Protein_A comprises three amino acids. The other proteins of this biomolecule group (Protein_C, Protein_D) are shorter (two amino acids). The column 'BG Representative' contains always the same record, because there exists only one representative of a biomolecule group. The last column 'Original List' contains the information about the origin of the protein. Protein_C for example originates from the protein list 'List_C'.

[1] Please note that in some cases (depending on the computer settings) the intersection symbol ∩ is not drawn.

6. Concluding Remarks and Outlook

Current work of ProLiC concerns provision of ProLiC as a node within The Konstanz Information Miner (KNIME (1), http://www.knime.org/) software in order to facilitate creation of structured Proteomics data processing workflows.

ProLiC is also included in the service portfolio of Bioinfra.Prot a service center of the 'German Network for Bioinformatics Infrastructure' (de.NBI, http://www.denbi.de/, Bioinfra.Prot Förderkennzeichen: FKZ 031 A 534A). De.NBI is a national infrastructure supported by the German Federal Ministry of Education and Research (BMBF).

7. References

1. Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., and Wiswedel, B. (2007) KNIME: The Konstanz Information Miner. *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer

Appendix: Configuration of ProLiC

In this section the configuration of ProLiC is outlined. ProLiC uses a file in order to provide access to settings required for program execution and for adaption of the program to the needs of the users. In the following paragraph the configuration file is described in detail.

a) The ProLiC configuration file (ProLiC_conf.ini)

The ProLiC configuration file uses the INI format. It is a simple, human readable text file composed of sections, keys and values. Sections are marked with square brackets. Each section contains lines of key and values pairs, e.g. the line 'compType=acc' assigns the value 'acc' to the key 'compType'. This special key and value pair defines the comparison method that will be used by ProLiC. Here, 'compType=acc' executes an accession based comparison.

In this case, an accession based comparison will be performed. Please note that ProLiC configuration is case sensitive. Therefore, all settings must exactly be given in the notation specified in this section. Table 2 presents an overview of the available sections and the keys therein along with a short parameter description.

Table 2: Overview of the sections of the ProLiC configuration file (ProLiC_conf.ini) and the keys therein. A short explanation of the parameter is given as well.

Key	Description
Section genericSettings	
column_delimiter_input	The column delimiter used in the input file(s).
column_delimiter_output	The column delimiter that should be used for the output file.
file_type	This parameter specifies, which type of file/in-formation should be read from the input directory.
considerProteinNames	This parameter specifies, whether protein name information provided in the input files should be processed.
considerGeneNames	This parameter specifies, whether gene name information provided in the input files should be processed.
writePeptInfo	This parameter specifies, whether peptide information is written in the result file or not.
encoding_output	This parameter specifies the character set for writing ProLiC results to the output file.
Section commandLineSettings	
inputDir	The absolute path of the input directory.
outputDir	The absolute path of the output directory.

file_extension_input	This parameter specifies, which type of file should be read from the input directory.
file_name_out	This parameter specifies the file name of the ProLiC output file (without the file name extension).
file_extension_output	This parameter specifies, which file name extension should be used for the output file.
compType	This parameter specifies the type of comparison (i.e. either accession, sequence or peptide based comparison).
consideration_of_modifications	This parameter specifies whether 'post translational modifications' (PTMs) should be used for comparison of peptide sequences.
regExModification	A regular expression that is used to find a post translational modification in a one letter code protein sequence for subsequent removal.
Section Enzymes	
enzyme1	The parameter value of enzyme1 is a combined string that includes different settings for a theoretical digestion of a protein sequence.
Section grouping	
selection*_representative*	This parameter specifies the method utilized for calculation of a representative biomolecule for a biomolecule group.
Section Seq. based comparison	
PID_method	This parameter specifies the method for calculation of the percent sequence identity (PID).
PID_threshold	This parameter specifies the percent sequence identity threshold, which needs to be passed in order to consider two sequences as identical.
substitution_matrix_id	This parameter specifies an identifier for a software intern generation of the substitution matrix.
alignment_method	This parameter specifies the identifier of the used alignment method.
penalty_gap_open	This parameter specifies the penalty for opening a gap.
penalty_gap_extension	This parameter specifies the penalty for gap extension.
Section Peptide based comparison	
peptideBasedMethod	This parameter specifies the method used for peptide based comparison.

In the following paragraphs each parameter listed in Table 2 is explained in detail. The allowed parameter values or even examples are given as well. Both the keys and the values are shown in italics. The ProLiC jar archive already includes a ProLiC_conf.ini file with reasonable default settings. In many cases it is sufficient to adapt the values for the section 'commandLineSettings' (e.g. the input and output directories) in order to achieve an executable state for ProLiC.

Section 'genericSettings': This section includes several parameters that are necessary for the overall functioning of ProLiC.

- *column_delimiter_input*: This parameter must correspond to the column delimiter used in the input file(s). Since ProLiC process simple text files appropriate (and recommended) column delimiters are:

  ,

  ;

  \t (Regular expression that specifies a tab stop as a column delimiter.)

- *column_*delimiter_*output*: The column delimiter that should be used for the output file. Since ProLiC writes an ASCII formatted output file appropriate column delimiters are:

  ,

  ;

  \t (Regular expression that specifies a tab stop as a column delimiter.)

- *file_type*: This parameter specifies, which type of file/in-formation should be read from the input directory. Currently, ProLiC accepts only files with protein information. Therefore, the only allowed parameter value is *protein_information.*
. It is intended that further versions of ProLiC also support comparison of files with gene related information.

- *considerProteinNames*: This parameter specifies, whether protein name information provided in the input files should be processed. This parameter is implemented for future versions of ProLiC. As a consequence, currently the only allowed parameter value is *false.*

- *considerGeneNames*: This parameter specifies, whether gene name information provided in the input files should be processed. This parameter is implemented for future versions of ProLiC. As a consequence, currently the only allowed parameter value is *false.*

- *writePeptInfo*: This parameter specifies, whether peptide information is written in the result file or not. Allowed values are *true* and *false*.

- *encoding_output*: This parameter specifies the character set for writing ProLiC results to the output file. Allowed values for this parameter are:
  Latin-1
  UTF-8
  Please observe the exact spelling of the parameter values.

Section commandLineSettings: This section includes several parameters that are necessary for ProLiC used from the command line, which is currently the only possibility to use ProLiC. Further software versions will include usage of ProLiC as a node within the KNIME workflow environment.

- *inputDir*: The absolute path of the input directory. Please observe the features of your operation system (e.g. usage of backslashes for Windows style path delimiters, usage of slashes for operation systems of the Unix/Linux family). Please note, that no escape character should be used within the path name.

- *outputDir*: The absolute path of the output directory. Please observe the features of your operation system (e.g. usage of backslashes for Windows style path delimiters, usage of slashes for operation systems of the Unix/Linux family). Please note, that no escape character should be used within the path name.

- *file_extension_input*: This parameter specifies, which file name extension should be considered for the data import. Example: if *txt* is specified as parameter value only .txt files are considered as ProLiC formatted data input files. All other files located in the *outputDir* (e.g. .csv files) are ignored.

- *file_name_out*: This parameter specifies the file name of the ProLiC output file (without the file name extension).

- *file_extension_output*: This parameter specifies, which file name extension should be used for the output file.

- *compType*: This parameter specifies the type of comparison (i.e. either an accession, a sequence or a peptide based comparison). Allowed values for this parameter are:
  acc
  sequence
  peptide
  Please observe the exact spelling of the parameter values.

- *consideration_of_*modifications: This parameter specifies whether or not 'post translational modifications' (PTMs) should be used for comparison of peptide sequences. Allowed values for this parameter are:
  true
  false

- *regExModification*: A regular expression that is used to find a post translational modification in the one letter code peptide sequence. In case of consideration_of_modifications = true, the regular expression is used to remove the PTM from the peptide sequence.

Section Enzymes: This section includes enzyme specific parameter settings. ProLiC uses this values for digestion of a protein sequence into peptide sequences. Currently, the only key in this section is 'enzyme1', because this version of ProLiC does not support a combined digestion, i.e. application of two or more proteases. However, it is possible to specify the necessary information for such a digestion protocol if the involved proteases cleave in the same direction (i.e. all proteases cleave either C-terminal or N-terminal). In this case the necessary information can be given by combining the cleavage sites of the proteases and by adaption of the regular expression term.

- *enzyme1*: The value of this parameter is a combined string. Substrings are separated by underscores. ProLiC expects four substrings for enzyme

definition. This is illustrated by the following example, which specifies the settings for the endoprotease trypsin:

Trypsin_KR_C_(?<=[KR])(?!P)

The first substring (here: 'Trypsin') is the name of the used enzyme.
The second substring (here: 'KR') denotes the cleavage site(s)
given in the amino acid 1-Letter code.
The third substring (here 'C') specifies if the enzyme
cleaves in the C- terminal (specified by the char 'C')
or in the N-terminal (specified by the char 'N') direction
The fourth substring (here '(?<=[KR])(?!P)') gives the
regular expression that is used to calculate the peptides
from a given protein sequence.

Section grouping: This section includes specific settings that are used for grouping of biomolecules.

- *selection_*representative: This parameter specifies the method utilized for calculation of a representative biomolecule for a biomolecule group. Allowed values for this parameter are:
  longestSequence: The protein with the longest protein sequence is selected as representative. If there are proteins with identical sequence lengths the first protein considering the alphabetical order of the unique identifier is selected. The unique identifier is a combination of the protein name and the input file name from which this protein originates.
  random: The representative is selected randomly from the biomolecule group.
  Please observe the exact spelling of the parameter values.

Section Seq. based comparison: This section includes specific settings that are necessary for a sequence based comparison of proteins (sequence alignment).

- *PID_method*: This parameter specifies the method for calculation of the percent sequence identity (PID). Allowed values for this parameter are:
  PID1: The percent identity is calculated by
  $PID1=100*(N_{identPos}/ShortestSequence)$.
  PID2: The percent identity is calculated by
  $PID2=100*(N_{identPos}/N_{alignPos})$.
  PID3: The percent identity is calculated by
  $PID3=100*(N_{identPos}/mean\ (SequenceLength))$.

PID4: The percent identity is calculated by

$$PID4=100*(N_{identPos}/(N_{alignPos}+N_{intGaps})).$$

Here, $N_{alignPos}$ denotes the number of aligned positions, $N_{identPos}$ is the number of identical amino acid positions within the compared sequences, $N_{intGaps}$ gives the number of internal gaps, *mean(SequenceLength)* is the mean value of the lengths of both compared protein sequences and *ShortestSequence* gives the length of the shorter protein sequence.

Please observe the exact spelling of the parameter values.

- *PID_threshold*: This parameter specifies the percent sequence identity threshold, which needs to be passed in order to consider two sequences as identical. This value must be given as a decimal number (e.g. 95.0).

- *substitution_matrix_id*: This parameter specifies an identifier for a software intern generation the substitution matrix. Allowed values for this parameter are:

  BLOSUM30
  BLOSUM35
  BLOSUM40
  BLOSUM45
  BLOSUM50
  BLOSUM55
  BLOSUM60
  BLOSUM62
  BLOSUM65
  BLOSUM70
  BLOSUM75
  BLOSUM80
  BLOSUM85
  BLOSUM90
  BLOSUM100
  PFAM250

  Please observe the exact spelling of the parameter values. Note that per default the BLOSUM62 substitution matrix is used if the value given in the configuration file does not match the allowed values given above.

- *alignment_method*: This parameter specifies the name of the used alignment method. Currently, ProLiC supports only global alignment with an implementation of the Needleman-Wunsch algorithm. Therefore, *NW* is the only allowed value for this parameter.

- *penalty_gap_open*: This parameter specifies the penalty for the opening of a gap. Values must be given as negative integer values.

- *penalty_gap_extension*: This parameter specifies the penalty for gap extension. Values must be given as negative integer values.

Section Peptide based comparison: This section includes specific settings that are necessary for a peptide based comparison of proteins

- *peptideBasedMethod*: This *parameter* specifies the method used for peptide based comparison. Allowed values for this parameter are (please observe the exact spelling of the parameter values):

  includeSequenceInfo: This settings results in a peptide based comparison that considers available protein sequence information for the determination of the identity between two proteins.
  peptidesOnly: This settings results in a peptide based comparison that considers only the *available* peptide sets of the two proteins specified. Proteins are different, if each peptide set of the compared proteins contains at least one peptide that is not part of the peptide set of the other protein.