MUPEXI USER MANUAL 1.1

Anne-Mette Bjerregaard[1]
[1]Center for Biological Sequence Analysis, Department of Systems Biology, Technical University of Denmark, Lyngby, Denmark.

September 27, 2016

TABLE OF CONTENTS

General Description

Personalization of immunotherapies such as cancer vaccines and adoptive T cell therapy will depend on identification of patient-specific neo-epitopes that can be specifically targeted. MuPeXI, the Mutant Peptide Extractor and Informer, is a program that uses somatic mutation data, HLA binding prediction, self-similarity comparison and gene expression to identify and assess potential neo-epitopes as targets for immunotherapy.

MuPeXI.py extracts peptides of user-defined lengths around missense variant mutations, indels and frameshifts. Information from each mutation is annotated together with the mutant and normal peptides in the file output.

Dependencies

To run MuPeXI the following software and packages must be installed:

Required software:

- Python 2.7
- NetMHCpan 2.8
- Variant Effect Predictor (VEP)

Python packages:

- Biopython
- pandas
- numpy

These packages are included if downloading python trough Anaconda, the full anaconda package list can be found here. Python 2.7 standard library list can be found here. If installing Anaconda it should not be necessary to do the second step of the installation guide.

Installation

1. Install all software listed above
2. Install Biopython, pandas and nympy with the following command:

```
pip install biopython
pip install pandas
pip install numpy
```

3. Download the repository to your local system.
4. If not already on your system, reference files from GRCh38 should be obtained. These include cDNA, peptide and cosmic referneces, look under references for detailed description.
5. Fill the config.ini file:

Instructions on how to fill the file is found within the file. The `config.ini` file is automatically detected in the same directory as `MuPeXI.py` script but can also be placed elsewhere and referred to by the `-c` option.

State the path to netMHCpan 3.0 and VEP in the config.ini file.

```
[netMHC]
# Please specify the netMHCpan 3.0 binary path
MHC = your/path/to/netMHCpan-3.0/netMHCpan

[EnsemblVEP]
# Please specify binary path to Ensembls Variant effect predictor (VEP), and the directory
of containing the cache database.
VEP = your/path/to/ensembl-tools-release-
85/scripts/variant_effect_predictor/variant_effect_predictor.pl
VEPdir = your/path/to/ensembl-tools-release-85/scripts/variant_effect_predictor
```

You should be aware that the version of VEP library you use should match the references used (peptide and cDNA) eg. In the example above is used version/release 85 of GRCh38.

The path to reference files should also be stated in the config.ini file, together with FULL path to the small C script named “pepmatch_db” downloaded with MuPeXI.

All reference peptides of the defined length, extracted from the proteome reference, can be defined as a file in the fonfig.ini file. This will save time, avoiding chop-up of the proteome reference for each run. As many files as peptides length should be stated, if not stated MuPeXI runs the chop-up for the peptide lengths where no peptide reference file is provided.

```
[References]
# Please specify the binary path to the references used (optional)
cDNA = your/path/to/human_GRCh38/cDNA/Homo_sapiens.GRCh38.78.cdna.all.fa
pep = your/path/to/human_GRCh38/pep/Homo_sapiens.GRCh38.78.pep.all.fa
cosmic = your/path/to/cosmic/Census_allWed_Feb_17_09-33-40_2016.tsv
pep9 = your/path/to/reference_peptide_9.txt

[PeptideMatch]
PM = your/path/to/MuPeXI/apps/pepmatch_db
```

If the user wishes for MuPeXI to do a lift-over from GRCh37 / HG19 to GRCh38 this is possible if a local version of picard tools 2.5 is installed. The paths to java8 and picard-tools should also be stated in the config.ini file.

```
[LiftOver]
fasta = your/path/to/references/GRCh38.fa
chain = your/path/to/references/HG19toGRCh38.chain
java8 = your/path/to/java8
picard = your/path/to/picard-tools-2.5.0
```

USAGE

After installation MuPeXI is called as follows. Here is an example where the reference files are stated in the config.ini file and netMHCpan-3.0 is run for the HLA types HLA-A01:01 and HLA-B08:01, with the transcript expression values annotated in the expression file.

The config.ini file is stated using the `-c` option and the expression file with the `-e` option.

```
path/to/MuPeXI.py -v mutation_call_file.vcf -a HLA-A01:01,HLA-B08:01 -c path/to/config.ini –e
expression_file.tsv
```

MuPeXI can be used for both peptide extraction, giving immunogenicity information for peptide selection, and for printing of FASTA file mutant-peptide-library for optimal mass spectrometry peptide detection.

All options can be explored using the usage information with the `-h` option:

```
> path/to/MuPeXI.py -h

    MuPeXI - Mutant Peptide Extractor and Informer
    version 2016-08-11

    The current version of this program is available from
    https://bitbucket.org/ambj/mupexi

    MuPeXI.py accepts a VCF file describing somatic mutations as input, and from this
    derives a set of mutated peptides of specified length(s). These mutated peptides
    are returned in a table along with various annotations that may be useful for
    predicting the immunogenicity of each peptide.

    Usage: ./MuPeXI.py -v <VCF-file> [options]

                                                                                DEFAULT
    Required arguments:
    -v, --vcf-file <file>   VCF file of variant calls, preferably from
                            MuTect (only SNVs) or MuTect2 (SNVs and indels)

    Recommended arguments:
    -a, --alleles           HLA alleles, comma separated.                       HLA-A02:01
    -l, --length            Peptide length, given as single number,             9
                            range (9-11) or comma separated (9,10,11).
    -e, --expression-file   Expression file, tab separated
                            ((ENST*/ENSG*)   mean)

    Optional arguments affecting output files:
    -o, --output-file       Output file name.                                   <VEP-file>.mupexi
    -d, --out-dir           Output directory - full path.                       current directory
    -p, --prefix            Prefix for output files - will be applied           <VCF-file>
                            to all (.mupexi, .log, .fasta) unless specified
                            by -o or -L.
    -L, --log-file          Logfile name.                                       <VCF-file>.log

    Other options (these do not take values)
    -f, --make-fasta        Create FASTA file with long peptides
                            - mutation in the middle
    -c, --config-file       Path to the config.ini file                         current directory
    -t, --keep-temp         Retain all temporary files
    -m, --mismatch-only     Print only mismatches in normal peptide sequence
                            and otherwise use dots (...AA.....)
    -w, --webface           Run in webserver mode
    -g, --hg19              Perform liftover HG19 to GRCh38.
                            Requires local picard installation with paths
                            stated in the config file.
    -E, --expression-type   Setting if the expression values in the expression  transcript
                            files are determined on transcript or gene level
                            (transcript/gene)
    -h, --help              Print this help information

    REMEMBER to state references in the config.ini file
```

INPUT FILES

MuPeXI accepts a VCF file of somatic mutation calls optimally obtained from either MuTect or MuTect2. The VCF files does not need to be file handled, the “raw” output VCF file can be put directly into MuPeXI.

VCF FILE

Compact example of a VCF file:

```
##fileformat=VCFv4.2
##GATKCommandLine.MuTect2=<ID=MuTect2,Version=3.5-0-g36282e4
##SAMPLE=<ID=NORMAL,SampleName=TCGA-XV-A9W5_N
##SAMPLE=<ID=TUMOR,SampleName=TCGA-XV-A9W5_T
##reference=file:///home/projects/pr_46630/data/references/human_GRCh38/GCA_000001405.15_GRCh38_f
ull_analysis_set.fa
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  TUMOR   NORMAL
chr1    948711  .   C   G   .   germline_risk . . . .
chr1    1657358 .   T   TA  .   alt_allele_in_normal . . . .
chr1    1986752 rs4233028   A   G   .   germline_risk . . . .
chr1    3431704 rs2493274   G   C   .   germline_risk . . . .
chr1    3631978 rs2244942   T   C   .   germline_risk . . . .
chr1    3839305 rs1891940   T   C   .   clustered_events;germline_risk  . . . .
```

A full example of a VCF file can be found on the MuPeXI webserver here.

EXPRESSION FILE

It is optional, but preferable, to provide a file with expression values as input to add the expression of each transcript where a mutated peptide was extracted.

The expression files used for testing MuPeXI were generated from raw RNA-seq data using Kallisto. The files should be tab separated and include Ensembl transcript ID (ENST) and mean expression WITHOUT a header.

```
ENST00000456328.2   0.868567715
ENST00000450305.2   0
ENST00000488147.1   2.72373575
ENST00000619216.1   0
ENST00000473358.1   0
ENST00000469289.1   0
ENST00000607096.1   0
```

A full example of an expression file can be found on the MuPeXI webserver here.

It should be noted that MuPeXI takes both expression values determined on transcript and gene level, though transcript is preferable. If gene level is used (ENSG…) the `-E gene` option should be used.

REFERENCES

The following references are required for MuPeXI to run:

- Peptide
  The peptide reference is a FASTA file containing all peptides of the human proteome.
- cDNA
  The cDNA reference is a FASTA file containing all nucleotide sequences of the human proteome.

These reference can be acquired from Ensembles website. The 85 release can be found here.

The following reference are optional but preferable:

- Cosmic
  TSV file containing known cancer driver genes. The cancer gene census can be downloaded from the COSMIC website.

OUTPUT FILES

MuPeXI can output up three files.

The output files are the following:

1. .mupexi
   The main output file containing a TSV file with the extracted mutated peptides and all the information needed to choose the wanted peptides.
2. . log
   The log file containing information about the run.
3. .fasta
   The FASTA file with the long peptides, before they are copped up into user defined length, with the mutation centered. (use the option -F)

COLUMN EXPLANATION

The prediction output (.mupexi) for each peptide pair consists of the following columns:

- **HLA allele** Allele name
- **Normal peptide** Peptide from reference corresponding to the mutant peptide.
- **Normal MHC affinity** Predicted binding affinity of normal peptide in nanoMolar units.
- **Normal MHC % rank** %Rank of prediction score for nomal peptides.
- **Mutant peptide** The extracted mutant peptide.
- **Mutant MHC affinity** Predicted binding affinity of mutant peptide in nanoMolar units.
- **Mutant MHC % rank** %Rank of prediction score for mutant peptides.
- **Gene ID** Ensembl gene ID
- **Transcript ID** Ensembl transcript ID
- **Amino acid change** Amino acid change annotated in VEP file.
- **Allele Frequency** Genomic allele frequency detected by MuTect2.
- **Mismatches** Mismatches between normal and mutant peptide.
- **Peptide position** Position of amino acid change in the peptide. Can be a range in the case of insertions and frameshifts.
- **Chr** Chromosome position annotated in the VEP file.
- **Genomic position** Genome nucleotide position annotated in the VEP file.
- **Protein position** Amino acid position annotated in the VEP file.
- **Mutation cons.** The consequence annotated in the VEP file translated into single letter

abbreviations:

M	Missense variant
I	In-frame insertion
D	In-frame deletion
F	Frameshift variant

- **Gene symbol** HUGO symbol corresponding to the Ensembl transcript id.
- **Cancer driver gene** Yes if the HUGO symbol is in the cosmic reference list, No if it is not.
- **Expression Level** Expression of the transcript which the mutant peptide was extracted from.
- **Mutant affinity score** Calculated binding affinity score of the mutant peptide, based on a negative logistic function of the mutant MHC %Rank score. This is used to calculate the final prioritization score.
- **Normal affinity score** Calculated binding affinity score of the normal peptide, based on a positive logistic function of the normal MHC %Rank score. This is used to calculate the final prioritization score.
- **Expression score** Calculated Expression score of the transcript expression level. This is used to calculate the final prioritization score.
- **Priority score** Calculated prioritization dependent on HLA binding, gene expression, normal and mutant peptide binding ratio and allele frequency.

NetMHCpan output:
%Rank of prediction score to a set of 200.000 random natural 9mer peptides. For more information go to NetMHCpan2.8 output format.

TEST EXAMPLE

To run the provided test files with MuPeXI the following command can be run:

```
path/to/MuPeXI.py –v test.vcf -c path/to/config.ini –e expression_test.tsv
```

For additional fasta file output:

```
path/to/MuPeXI.py –v test.vcf -c path/to/config.ini –e expression_test.tsv -f
```

Print only the miss-match amino acid for the normal peptide:

```
path/to/MuPeXI.py –v test.vcf -c path/to/config.ini –e expression_test.tsv -m
```

DATA PREPARATION

Raw sequencing data should be obtained and the following steps is a recommendation on how to processes this data prior to using MuPeXI.

RECOMMENDED PREPROCESSING OF NEXT GENERATION SEQUENCING (NGS) DATA

The following data should be obtained:

- WXS (or WGS) of tumor and corresponding normal sample
- RNAseq of tumor sample

Raw sequencing data is processed to extract variant calls, and optionally gene expression values. Therefore, either whole exome (WXS) or whole genome (WGS) sequencing data is required from both tumor and matching normal sample. For optimal utilization of MuPeXI, RNA sequencing (RNAseq) data should be obtained as well from the tumor sample.

DATA CLEANUP

First, it is important to check the quality of both WXS and RNAseq data. This can be done by running the wrapper tool Trim Galore, which combines the functions of Cutadapt and FastQC: trimming the reads below an average phred score of 20 (default value), cutting out standard adaptors, such as those from Illumina, and in the end running FastQC to display the FastQC report. This enables the user to check if the quality is satisfactory and if non-standard adapters are seen in the overrepresented sequences.

WXS DATA

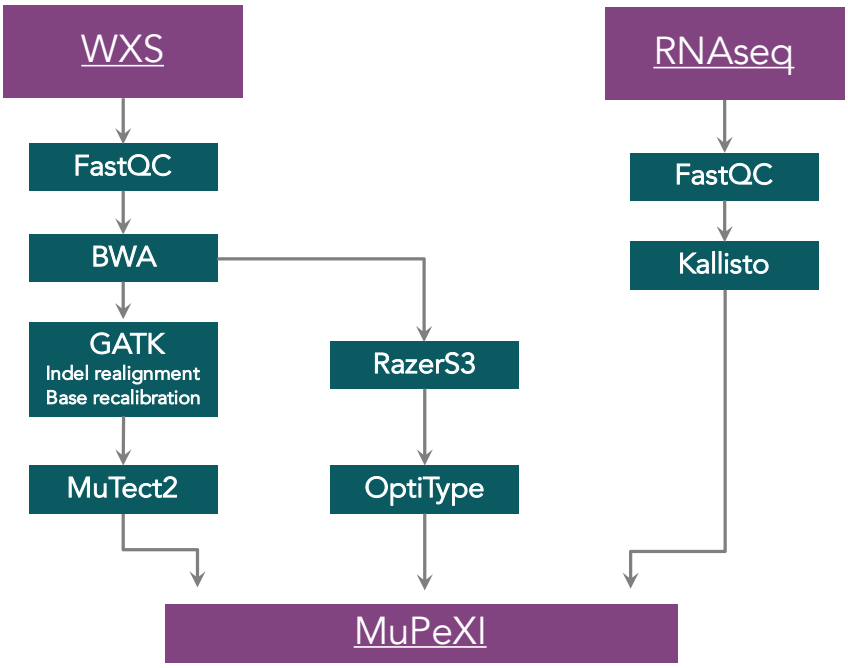
For the WXS pipeline the preprocessing steps for variant calling should be done following the Genome Analysis Toolkit (GATK) best practice guidelines. The reads are aligned using default Burrows-Wheeler Aligner mem options; here it is important to provide the Read Group for the following steps to be possible. Then indel realignment and base recalibration are done to reduce false positive variant calls; a detailed description of these steps can be found on the GATK website.
Single nucleotide variant (SNV) and indel calling should be done using MuTect2. MuTect2 is designed to call variants, both SNVs and indels, from tumor samples evaluating reads from matched tumor and normal samples.
The standard output format from MuTect2 is a VCF file.
If MuTect or Mutect2 is not used for variant calling the genomic allele frequency will not be taken into account in the MuPeXI priority score.

RNASEQ

After quality check the expression scores are obtained by pseudo-aligning the raw reads using Kallisto. Since this is very fast, the runs are bootstrapped (eg. 500 times) giving a variance of each expression score obtained. The expression score is normalized by Kallisto in transcripts per million (TPM).

HLA TYPING

If the HLA alleles for the sample(s) have not been typed by other means, they can be inferred from sequencing data by several different algorithms. Among these are ATHLATES , Polysolver, OptiType and PHLAT. We used OptiType with default settings, first filtering the reads with RazerS3.
The outputs files from the NGS analysis; VCF file, HLA alleles and expression file, are now ready to be processed by MuPeXI.



Pre-processing of raw sequencing data. Flowchart of an example NGS pipeline to process raw sequencing data and make it ready to predict neo-epitopes with MuPeXI. BWA: Burrows-Wheeler Alignment. GATK: Genome Analysis Tool Kit. WXS: Whole exome sequencing.

CONTACT

Anne-Mette Bjerregaard
ambj@cbs.dtu.dk

or

Aron Charles Eklund
eklund@cbs.dtu.dk

Center for Biological Sequence Analysis
Technical University of Denmark
Department of Bio and Health Informatics
Kemitorvet building 208, room 007
2800 Lyngby, Denmark