xpore Documentation

Release 2.0

Ploy N. Pratanwanich

Oct 08, 2021

Contents

xPore is a Python package for identification of differentail RNA modifications from Nanopore sequencing data.

To install the latest release, run:

```
pip install xpore
```

See our *Installation page* for details.

To check the version of xPore, run:

```
xpore -v
```

To detect differential modifications, you can follow the instructions in our *Quickstart page*.

CHAPTER 1

Contents

1.1 Installation

xPore requires Python3 to run.

1.1.1 PyPI installation (recommended)

```
pip install xpore
```

1.1.2 Installation from our GitHub repository

```
git clone https://github.com/GoekeLab/xpore.git
cd xpore
python setup.py install
```

1.2 Quickstart - Detection of differential RNA modifications

Download and extract the demo dataset from our zenodo:

```
wget https://zenodo.org/record/5162402/files/demo.tar.gz
tar -xvf demo.tar.gz
```

After extraction, you will find:

```
demo
|-- Hek293T_config.yml  # configuration file
|-- data
    |-- HEK293T-METTL3-KO-rep1  # dataset dir
```

(continues on next page)

(continued from previous page)

```
    |-- HEK293T-WT-rep1 # dataset dir
|-- demo.gtf # GTF (general transfer format) file for transcript-to-gene mapping
|-- demo.fa # transcriptome reference FASTA file for transcript-to-gene mapping
```

Each dataset under the `data` directory contains the following directories:

- `fast5` : Raw signal, FAST5 files
- `fastq` : Basecalled reads, FASTQ file
- `bamtx` : Transcriptome-aligned sequence, BAM file
- `nanopolish`: Eventalign files obtained from nanopolish eventalign

Note that the FAST5, FASTQ and BAM files are required to obtain the eventalign file with Nanopolish, xPore only requires the eventalign file. See our *Data preparation page* for details to obtain the eventalign file from raw reads.

1. Preprocess the data for each data set using `xpore dataprep`. Note that the `--gtf_or_gff` and `--transcript_fasta` arguments are required to map transcriptomic to genomic coordinates when the `--genome` option is chosen, so that xPore can run based on genome coordinates. However, GTF is the recommended option. If GFF is the only file available, please note that the GFF file works with GENCODE or ENSEMBL FASTA files, but not UCSC FASTA files. We plan to remove the requirement of FASTA files in a future release.(This step will take approximately 5h for 1 million reads):

```
# Within each dataset directory i.e. demo/data/HEK293T-METTL3-KO-rep1 and demo/
↪data/HEK293T-WT-rep1, run
xpore dataprep \
--eventalign nanopolish/eventalign.txt \
--gtf_or_gff ../../demo.gtf \
--transcript_fasta ../../demo.fa \
--out_dir dataprep \
--genome
```

The output files are stored under `dataprep` in each dataset directory:

- `eventalign.index` : Index file to access `eventalign.txt`, the output from nanopolish eventalign
- `data.json` : Preprocessed data for `xpore-diffmod`
- `data.index` : File index of `data.json` for random access per gene
- `data.readcount` : Summary of readcounts per gene
- `data.log` : Log file

Run `xpore dataprep -h` or visit our *Command line arguments* to explore the full usage description.

2. Prepare a `.yml` configuration file. With this YAML file, you can specify the information of your design experiment, the data directories, the output directory, and the method options. In the demo directory, there is an example configuration file `Hek293T_config.yaml` available that you can use as a starting template. Below is how it looks like:

```
notes: Pairwise comparison without replicates with default parameter setting.

data:
    KO:
        rep1: ./data/HEK293T-METTL3-KO-rep1/dataprep
    WT:
        rep1: ./data/HEK293T-WT-rep1/dataprep

out: ./out # output dir
```

See the *Configuration file page* for more details.

3. Now that we have the data and the configuration file ready for modelling differential modifications using `xpore-diffmod`.

```
# At the demo directory where the configuration file is, run.
xpore diffmod --config Hek293T_config.yml
```

The output files are generated within the `out` directory:

- `diffmod.table` : Result table of differential RNA modification across all tested positions
- `diffmod.log` : Log file

Run `xpore diffmod -h` or visit our *Command line arguments* to explore the full usage description.

We can rank the significantly differentially modified sites based on `pval_HEK293T-KO_vs_HEK293T-WT`. The results are shown below.:

```
id                position   kmer  diff_mod_rate_KO_vs_WT  pval_KO_vs_WT  z_score_KO_
↪vs_WT  ...  sigma2_unmod  sigma2_mod  conf_mu_unmod  conf_mu_mod  mod_assignment
↪    t-test
ENSG00000114125  141745412  GGACT               -0.823318  4.241373e-115        -22.
↪803411  ...      5.925238   18.048687       0.968689     0.195429           lower
↪1.768910e-19
ENSG00000159111   47824212  GGACT               -0.828023   1.103790e-88        -19.
↪965293  ...      2.686549   13.820089       0.644436     0.464059           lower
↪5.803242e-18
ENSG00000159111   47824138  GGGAC               -0.757891   1.898161e-73        -18.
↪128515  ...      3.965195    9.877299       0.861480     0.359984           lower
↪9.708552e-08
ENSG00000159111   47824137  GGACA               -0.604056   7.614675e-24        -10.
↪068479  ...      7.164075    4.257725       0.553929     0.353160           lower
↪2.294337e-10
ENSG00000114125  141745249  GGACT               -0.514980   2.779122e-19         -8.
↪977134  ...      5.215243   20.598471       0.954968     0.347174           lower
↪1.304111e-06
```

4. (Optional) We can consider only one modification type per k-mer by finding the majority `mod_assignment` of each k-mer. For example, the majority of the modification means of `GGACT` (`mu_mod`) is lower than the non-modification counterpart (`mu_unmod`). We can filter out those positions whose `mod_assigment` values are not in line with those of the majority in order to restrict ourselves with one modification type per kmer in the analysis. This can be done by running `xpore postprocessing`.

```
xpore postprocessing --diffmod_dir out
```

With this command, we will get the final file in which only kmers with their `mod_assignment` different from the majority assigment of the corresponding kmer are removed. The output file `majority_direction_kmer_diffmod.table` is generated in the `out` directtory. You can find more details in our paper.

Run `xpore postprocessing -h` or visit our *Command line arguments* to explore the full usage description.

1.3 Output table description

Column name	Description
id	transcript or gene id
position	transcript or gene position
kmer	5-mer where modified base sits in the middle if modified
diff_mod_rate_<condition1>_vs_<condition2>	differential modification rate between condition1 and condition2 (modification rate of condition1 - modification rate of condition2)
z_score_<condition1>_vs_<condition2>	z score obtained from z-test of the differential modification rate
pval_<condition1>_vs_<condition2>	significance level from z-test of the differential modification rate
mod_rate_<condition>-<replicate>	modification rate of a replicate in the condition
mu_unmod	inferred mean of the unmodified RNAs distribution
mu_mod	inferred mean of the modified RNAs distribution
sigma2_unmod	inferred sigma^2 of the unmodified RNAs distribution
sigma2_mod	inferred sigma^2 of the modified RNAs distribution
conf_mu_unmod	confidence level of mu_unmod compared to the unmodified reference signal
conf_mu_mod	confidence level of mu_unmod compared to the unmodified reference signal
mod_assignment	lower if mu_mod < mu_unmod and higher if mu_mod > mu_unmod

1.4 Configuration file

The format of configuration file which is one of the inputs for `xpore-diffmod` is YAML.

Only the `data` and `out` sections are required, other sections are optional. Below is the detail for each section.

```
data:
    <CONDITION_NAME_1>:
        <REP1>: <DIR_PATH_TO_DATA_JSON>
        ...

    <CONDITION_NAME_2>:
        <REP1>: <DIR_PATH_TO_DATA_JSON>
        ...

    ...

out: <DIR_PATH_FOR_OUTPUTS>

criteria:
    readcount_min: <15>
    readcount_max: <1000>

method:
    # To speed up xpore-diffmod, you can use a statistical test (currently only t-
↪test is implemented) can be used
    # to remove positions that are unlikely to be differentially modified. So, xpore-
↪diffmod will model only
    # those significant positions by the statistical test -- usually the P_VALUE_
↪THRESHOLD very high e.g. 0.1.
    # If you want xPore to test every genomic/transcriptomic position, please remove␣
↪this prefiltering section.
```

(continues on next page)

(continued from previous page)

```
    prefiltering:
        method: t-test
        threshold: <P_VALUE_THRESHOLD>

    # Here are the parameters for Bayesian inference. The default values shown in <>␣
↪are used, if not specified.
    max_iters: <500>
    stopping_criteria: <0.00001>
```

1.5 Data preparation from raw reads

1. After obtaining fast5 files, the first step is to basecall them. Below is an example script to run Guppy basecaller. You can find more detail about basecalling at Oxford nanopore Technologies:

```
guppy_basecaller -i </PATH/TO/FAST5> -s </PATH/TO/FASTQ> --flowcell <FLOWCELL_ID>␣
↪--kit <KIT_ID> --device auto -q 0 -r
```

2. Align to transcriptome:

```
minimap2 -ax map-ont -uf -t 3 --secondary=no <MMI> <PATH/TO/FASTQ.GZ> > <PATH/TO/
↪SAM> 2>> <PATH/TO/SAM_LOG>
samtools view -Sb <PATH/TO/SAM> | samtools sort -o <PATH/TO/BAM> - &>> <PATH/TO/
↪BAM_LOG>
samtools index <PATH/TO/BAM> &>> <PATH/TO/BAM_INDEX_LOG>
```

3. Resquiggle using nanopolish eventalign:

```
nanopolish index -d <PATH/TO/FAST5_DIR> <PATH/TO/FASTQ_FILE>
nanopolish eventalign --reads <PATH/TO/FASTQ_FILE> \
--bam <PATH/TO/BAM_FILE> \
--genome <PATH/TO/FASTA_FILE \
--signal-index \
--scale-events \
--summary <PATH/TO/summary.txt> \
--threads 32 > <PATH/TO/eventalign.txt>
```

1.6 Data

You can find the links to all preprocessed data used in our manuscript at Zenodo for the SGNEx data and for the others samples. All the raw fast5 and fastq files are also available at ENA and SGNEx. Please refer to our Supplementary Table S7 in our manuscript for full details of data download.

Note that all HEK293T-KO samples can be used as unmodified (m6A) controls for any other data set generated with the same RNA kit (SQK-RNA001). If the cells are genetically different, we recommend to perform variant calling before finalising the list of differentially modified sites in order to remove false positives.

1.7 Command line arguments

We provide 2 main scripts to run the analysis of differential RNA modifications as the following.

1.7.1 xpore-dataprep

- Input

Output files from `nanopolish eventalgin`. Please refer to *Data preparation* for the full Nanopolish command.

Argument name	Required	Default value	Description
–eventalign=FILE	Yes	NA	Eventalign filepath, the output from nanopolish.
–out_dir=DIR	Yes	NA	Output directory.
–gtf_path_or_url	No	NA	GTF file path or url used for mapping transcriptomic to genomic coordinates.
–transcript_fasta_paths_or_urls	No	NA	Transcript FASTA paths or urls used for mapping transcriptomic to genomic coordinates.
–skip_eventalign_indexing	No	False	To skip indexing the eventalign nanopolish output.
–genome	No	False	To run on Genomic coordinates. Without this argument, the program will run on transcriptomic coordinates.
–n_processes=NUM	No	1	Number of processes to run.
–readcount_max=NUM	No	1000	Maximum read counts per gene.
–readcount_min=NUM	No	1	Minimum read counts per gene.
–resume	No	False	With this argument, the program will resume from the previous run.

- Output

File name	File type	Description
eventalign.index	csv	File index indicating the position in the *eventalign.txt* file (the output of nanopolish eventalign) where the segmentation information of each read index is stored, allowing a random access.
data.json	json	Intensity level mean for each position.
data.index	csv	File index indicating the position in the *data.json* file where the intensity level means across positions of each gene is stored, allowing a random access.
data.log	txt	Gene ids being processed.
data.readcount	csv	Summary of readcounts per gene.

1.7.2 xpore-diffmod

- Input

Output files from `xpore-dataprep`.

Argument name	Required	Default value	Description
–config=FILE	Yes	NA	YAML configurtaion filepath.
–n_processes=NUM	No	1	Number of processes to run.
–save_models	No	False	With this argument, the program will save the model parameters for each id.
–resume	No	False	With this argument, the program will resume from the previous run.
–ids=LIST	No	[]	Gene / Transcript ids to model.

- Output

File name	File type	Description
diffmod.table	csv	Output table information of differential modification rates. Please refer to *Output table description* for the full description.
diffmod.log	txt	Gene/Transcript ids being processed.

1.7.3 `xpore-postprocessing`

- Input

The `diffmod.table` file from `xpore-diffmod`.

Argument name	Required	Description
–diffmod_dir	Yes	Path of the directory containing `diffmod.table`.

1.8 Citing xPore

If you use xPore in your research, please cite

Ploy N. Pratanwanich, et al., Identification of differential RNA modifications from nanopore direct RNA sequencing with xPore. *Nat Biotechnol* (2021), https://doi.org/10.1038/s41587-021-00949-w.

Thank you!

1.9 Getting Help

We appreciate your feedback and questions! You can report any error or suggestion related to xPore as an issue on github. If you have questions related to the manuscript, data, or any general comment or suggestion please use the Discussions.

Thank you!

CHAPTER 2

Contacts

If you use xPore in your research, please cite

Ploy N. Pratanwanich, et al.,Identification of differential RNA modifications from nanopore direct RNA sequencing with xPore. *Nat Biotechnol* (2021), https://doi.org/10.1038/s41587-021-00949-w

xPore is maintained by Ploy N. Pratanwanich, Yuk Kei Wan and Jonathan Goeke from the Genome Institute of Singapore, A*STAR.

If you want to contribute, please leave an issue in our repo

Thank you!