ErrorX User Guide

Contents

Overview

ErrorX is a revolutionary software for correction of DNA sequencing errors. In next-generation sequencing, it's common that many of the output sequences are actually errors, different from the input DNA sequence. This can be caused by the way the DNA was prepared, amplification before sequencing, and errors introduced by the sequencing machine itself. Unfortunately, these errors can be very costly to a next-generation sequencing pipeline, as you can never be sure which sequences are true and which are mistakes.

ErrorX solves this problem by using deep neural networks to predict sites where an error has been introduced. Trained on a dataset of tens of millions of sequences, ErrorX can pinpoint with extremely high accuracy positions where an error was likely introduced, saving valuable time and energy by removing these bases from further analysis. In benchmark studies, ErrorX was able to identify errors with an accuracy of 99.9% and a false positive rate of only 0.1%.

ErrorX currently supports processing of antibody and T-cell repertoire sequencing data from human and mice, gathered on Illumina HiSeq and MiSeq instruments.

ErrorX User Guide

Installation

To install ErrorX, you need to request a download link from your Endeavor Bio representative. They will give you a link to the appropriate download based on your operating system. Once you have downloaded the zipped archive (this will have the extension `.tar.gz`), you are ready to unpack the folder and starting running.

Run the following command to extract the files from the archive:

```
     tar xvfz ErrorX-1.0_mac.tar.gz
```

The ErrorX binary is ready to go from there - it is located at `Errorx-1.0/bin/errorx`

Note: the binary relies on data contained in the `ErrorX-1.0` folder. For best results, leave the directory structure of ErrorX intact.

Quickstart

After installing ErrorX you are ready to run your first sequences. Use the following command to run ErrorX prediction:

```
     ErrorX/bin/errorx --species human --format fastq --out
ExampleSequences.tsv --allow-nonproductive
ExampleSequences.fastq
```

This will create a file called `ExampleSequences.tsv` that contains a summary of your input sequences, plus the nucleotide sequences with ErrorX correction applied.

Input

FASTQ

The most common way to run ErrorX is with a FASTQ file. ErrorX will run germline assignment on the sequences and use sequence data and quality information to feed into the error prediction model.

TSV

Alternatively you can provide sequences in TSV format. If you've already run germline assignment with another software, you can save time and only run error prediction in ErrorX. TSV files should have four columns, separated by tabs, with no header:

1. Sequence ID
2. Nucleotide sequence
3. Inferred germline sequence
4. PHRED score.

Output

The output of ErrorX is a TSV file summarizing the input sequences along with a corrected nucleotide sequence, where the predicted errors are replaced by 'N'. If you input a FASTQ sequence, then the TSV will have information on the V, D, and J genes, as well as the level of somatic mutation and CDR3 sequence.

Python interface

Installation

The Python packages are also located in the folder you just extracted. ErrorX supports Python versions 2.7 and 3.6. To install the Python 2.7 package simply run

```
pip install python2_bindings/
```

and for version 3.6 run

```
pip3 install python3_bindings
```

You are now ready to run ErrorX through the Python interface!

Java interface

installation

The ErrorX Java library is also located in the gzipped archive, in the folder `java_bindings`. To install you don't need to do anything special - all the functionality is contained in the `ErrorX.jar` archive within the `java_bindings` folder. When you are compiling your Java application simply add the flag

```
-cp 'ErrorX-1.0/java_bindings/ErrorX.jar'
```

to make sure the JAR is linked, and you are all set!

Note: the JAR relies on data files contained in the `java_bindings` folder. For best results, leave the directory structure of `java_bindings` intact and only link the JAR. Alternatively, if you have to move the JAR file, make sure to move all the other directories in `java_bindings` along with it.