XPRESSyourself: Automating and Democratizing High-Throughput Sequencing

Jordan A. Berg,[1] Jonathan R. Belyeu,[2] Alex J. Bott,[1] Jeffrey T. Morgan,[1] Yeyun Ouyang,[1]
Jason Gertz,[3] Michael T. Howard,[2] Aaron R. Quinlan,[2,4,5] Jared P. Rutter[1,6*]

[1]Department of Biochemistry, University of Utah, Salt Lake City, UT, USA, 84112
[2]Department of Human Genetics, University of Utah, Salt Lake City, UT, USA, 84112
[3]Department of Oncological Sciences, University of Utah, Salt Lake City, UT, USA, 84112
[4]USTAR Center for Genetic Discovery, University of Utah, Salt Lake City, UT, USA, 84112
[5]Department of Biomedical Informatics, University of Utah, Salt Lake City, UT, USA, 84112
[6]Howard Hughes Medical Institute, University of Utah, Salt Lake City, UT, USA, 84112

*To whom correspondence should be addressed; E-mail: rutter@biochem.utah.edu.

With the advent of high-throughput sequencing platforms, expression profiling is becoming common-place in medical research. However, for the general user, often someone who ends up outsourcing their bioinformatics, a computational overhead exists. The XPRESSyourself suite aims to remove these barriers and create a tools to help standardize and increase throughput of data processing and analysis. The XPRESSyourself suite is currently broken down into two software packages. The first, XPRESSpipe, automates the pre-processing, alignment, quantification, normalization, and quality control of single-end and paired-end RNAseq, as well as ribosome profiling sequence data. The second, XPRESStools, is a Python toolkit for expression data analysis, compatible with private or public microarray and RNAseq datasets. This software suite is designed where features can easily be modified, and additional packages can be included for processing of other data types in the future, such as CHIPseq or genome alignment. Currently, this package offers several new tools for ribosome profiling and general RNA-seq.

XPRESSyourself is freely available on GitHub: https://github.com/XPRESSyourself

1 Introduction

High-throughput profiling of gene expression data has revolutionized biomedical, industrial and basic science research. Within the last two decades, RNA-seq as found itself the forerunner technology for highest quality expression profiling, as it can measure relative transcript abundance, differential splice variants, sequence polymorphisms, and more. This technology has also been adopted to create technologies such as single-cell RNA-seq, capable of assaying the transcriptional profile cell by cell; and ribosome profiling, which measures ribosome occupancy and translation efficiency.

While vast strides have been made to these technologies, various bottlenecks still exist. For example, while more and more researchers are becoming accustomed to these technologies, learning the bioinformatics portion of sequencing possesses its own learning curve and often efficiency is lacking in how sequencing reads are processed and analyzed. Also for these users, they may not be aware of which tools are accepted as the standard in the field or which analyses they should be perform. They may also lack the experience to process their sequencing libraries rapidly or may not know all the in-between steps that are not always explicitely stated in protocols.

While several pipelines have emerged over the last several years that have been built to tackle various aspects of these bottlenecks, most are not widely used or usable by the average wet-bench researcher. Some are difficult to install or use, often they break easily or do not perform well. Rarely do these tools offer anything new to help overcome emerging challenges in the field.

In response to these issues surrounding the automation and democratization of sequencing technology, we created the XPRESSyourself bioinformatics suite for processing and analyzing high-throughput expression data. In creating this tool, we focused on five aspects in order to create an easy, reliable tool where large barriers-to-entry would be elimiated. These were create a tool that was useful, usable, reliable, efficient, and flexible.

1. We wanted the software we created to be useful for a broad audience, where the bulk of processing and analysis desired by a general user would be covered. We wanted to use pre-existing tools that were fast and accurate. We also wanted to provide additional, new tools that would be of use to the general RNA-seq community, which will be discussed in more detail later.

2. We wanted to create a software package that was easy to use. To do so, we made the tools installable by a single command in the command line interface (CLI) using the Conda and PyPi package managers. We also included thorough external documentation hosted on readthedocs that outlines use and considerations for each tool, as well as provides several examples of how to use each tool. Internally in the CLI-packages, summary documentation has been included by way of the help interface. Jupyter notebooks are also created and installed with the software that provide example analyses that can be easily modified and run.

3. To create a reliable pipeline and analysis package, we use the most current state-of-the-art software tools that have undergone robust benchmarking. We utilize a two-pass RNA-seq alignment process to provide the best coverage around splice sites. We also built the RNA-seq pipeline according to The Cancer Genome Atlas (TCGA) standards. While this technology will no doubt improve over the years, the software is structures in a way for easy modification for addition of tools or substitution of software.

4. In order to make the most efficient package possible, by default XPRESSyourself optimizes use of computing cores to ensure all available are utilized when possible. Additionally, for analysis tools processing large files, we utilize a data matrix chunking method, where a dataset is portioned off into a number equal to the number of cores available, and processes each parallely before rejoining the data chunks.

5. Flexibility is paramount in creating a tool that can be widely used and built upon. The general structure of the software was designed to make it easy to add or remove features. We envision as this suite of tools is more widely adopted by the RNAseq community, modules will be added to handle other sequencing platforms, such as genome sequencing, CHIPseq, and so on.

With XPRESSyourself, the user is provided with a complete suite of software to handle pre-processing, aligning, and quantifying reads, performing quality control via various meta-analyses of pre- and post-processed reads, and tools to perform the bulk of sequence analysis with enough flexibility to generate professional, figure-worthy images.

2 Materials and Methods

2.1 XPRESSpipe

2.1.1 Installation

General installation of XPRESSpipe is handled by the package manager, Anaconda, which is easy to download and install (https://www.anaconda.com/distribution/). XPRESSpipe can be installed using the following command:

```
$ conda install –c bioconda XPRESSpipe
```

Listing 1: curateReference example

XPRESSpipe is built upon several pre-established software packages, listed below:

Package	URL	Purpose
fastp	https://github.com/OpenGene/fastp	Read pre-processing
ucsc-gtftogenepred	http://hgdownload.cse.ucsc.edu/admin/exe/	Create refFlat reference file used in meta-analysis
STAR	https://github.com/alexdobin/STAR	Reference curation and read alignment
samtools	http://www.htslib.org/	Alignment file manipulation
bedtools	https://github.com/arq5x/bedtools2	Alignment file manipulation
deeptools	https://github.com/deeptools/deepTools	Alignment file manipulation
htseq	https://github.com/simon-anders/htseq	Read quantification
fastqc	https://www.bioinformatics.babraham.ac.uk/projects/fastqc/	Quality Control
multiqc	https://multiqc.info/	Quality Control
picard	https://broadinstitute.github.io/picard/	Transcript meta-analysis
plastid	https://github.com/joshuagryphon/plastid	Periodicity meta-analysis

These packages will be automatically installing using the Anaconda package manager; however, if installing manually, these packages will need to be installed by the user.

2.1.2 Inputs

While inputs will vary sub-module to sub-module, and further information can be found in the documentation (https://xpresspipe.readthedocs.io/en/latest/) or by entering "xpresspipe <sub-module name> –help", a few points of guidance are important to consider.

- Single-end reads should end in ".fa", ".fasta", ".txt"
- Paired-end reads should end in ".read1/2.suffix" or ".r1/2.suffix"
- The transcriptome reference file should be a valid GTF file and should be named "transcripts.gtf"
- If specifying a group of fasta files to use for alignment or reference curation, the directory containing this files cannot contain any other files ending in ".txt" or ".fa"

2.1.3 Reference Curation

One of the first preparatory steps of RNAseq alignment is curating a reference transcriptome for the aligner to map reads to. For the purposes of XPRESSpipe, a STAR reference should be created. However, for the purposes of several meta-analyses and more specific mapping, a STAR reference alone is not enough. For example, if one is aligning ribosome profiling reads and wishes to avoid mapping reads to the first 45 nucleotides in order to not quantify the inherent 5' biases in library preparation, a truncated transcriptome reference needs to be used when counting reads to transcripts. Additionally, if one wishes to only quantify reads to the coding portions of the transcriptome, a custom reference file needs to be created. While each of these steps are available as stand-alone sub-modules, all potential reference files, including STAR files, modified transcriptome files, and other meta-analysis files can be created by running the “curateReference” command. An example is shown below for creating a ribosome profiling-ready reference XPRESSpipe directory. The following assumes one is avoiding mapping to the first 45 nucleotides, will only quantify to protein coding regions, and is tailored for mapping 50-bp single-end RNA-seq reads. As this can be a time-consuming process, we will leave the “–max_processors” argument as default in order to utilize all cores available to the computing unit.

```
$ xpresspipe curateReference -o /path/to/output/location/
                             -f /path/to/fasta/genome/files/
                             -g /path/transcripts.gtf
                             --truncate_amount 45
                             --sjdbOverhang 49
```

Listing 2: curateReference example

While current available arguments are limited, the design is simple enough where arguments could be added easily to give more control over STAR reference creation; however, current setting align with TCGA standards.

2.1.4 Read Processing

While all intermediate steps of the pipelines can be run singly, we will describe the outline of the software in the context of the pipelines.

pipeline run singly or all together normalization / batch effect

2.1.5 Outputs

output minimal but enough to get clear picture optional outputs

Figure 1 provides an example of output file scheme for XPRESSpipe.

2.1.6 Quality Control

quality control read distribution meta-gene periodicity

2.1.7 Analyses

prober Deseq

2.2 XPRESStools

2.2.1 Getting Data

2.2.2 Normalizing and Formatting Data

2.2.3 Analyzing Data

2.3 Unit Testing and Code Coverage

New tools will require new tests to maintain code Coverage

2.4 Availability

Open source community GitHub Version Control Singularity

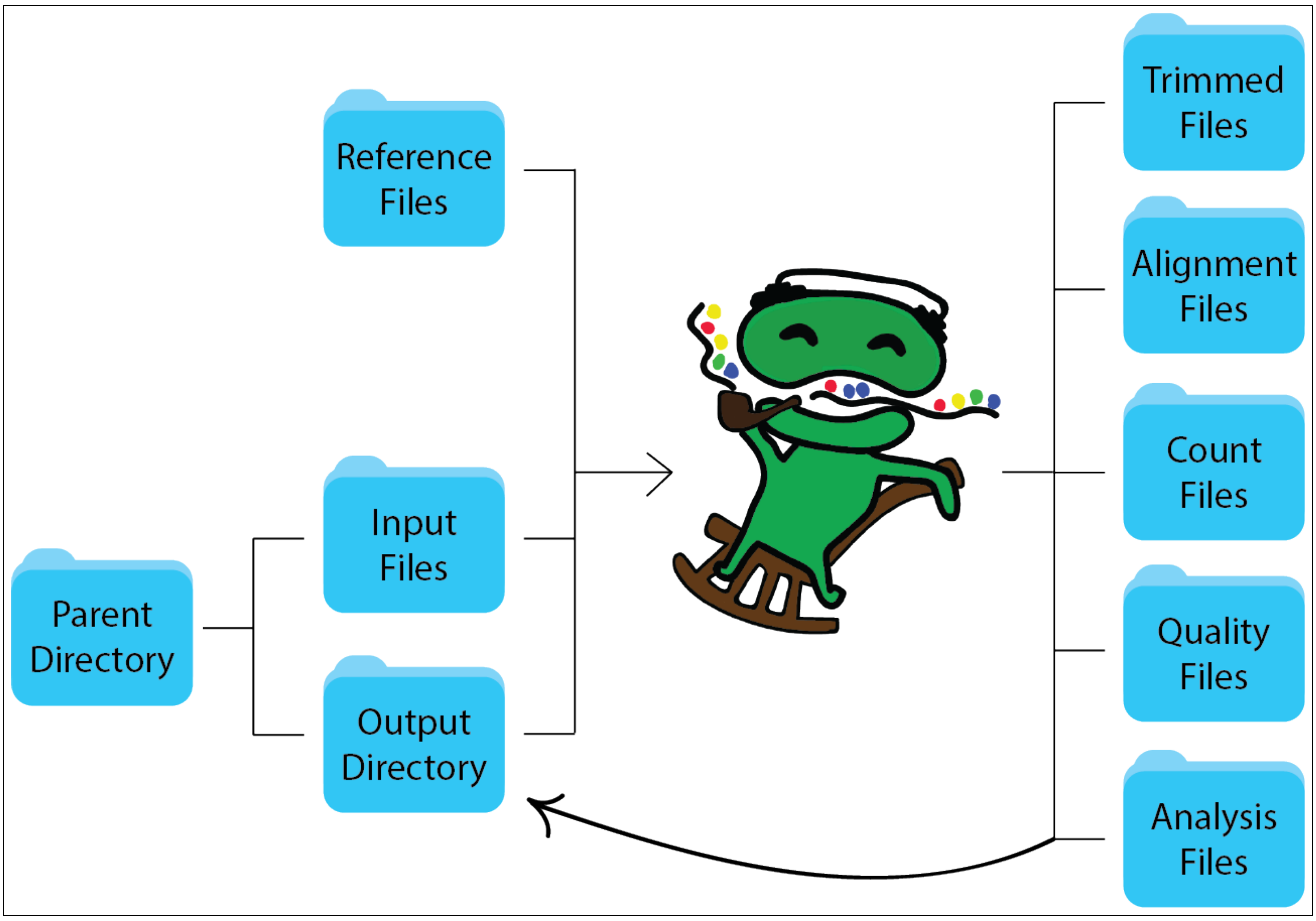

Figure 1: A schematic of XPRESSpipe outputs. Input, output, and often reference directories are specified and each stage of processing or analysis is given a separate directory.

Results and Discussion

2.5 Benchmarking

2.6 Example Data Walkthrough

2.7 Cost Analysis

2.8 Summary

References

Acknowledgments

J.A.B. received support from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) Interdisciplinary Training Grant T32 Program in Computational Approaches to Diabetes and Metabolism Research, 1T32DK11096601 to Wendy W. Chapman and Simon J. Fisher.

Contributions

Conceptualization	J.A.B.
Supervision	M.T.H., J.G., A.R.Q., J.P.R.
Project Administration	J.A.B.
Investigation	J.A.B.
Formal Analysis	J.A.B.
Software	J.A.B.
Methodology	J.A.B.
Validation	J.A.B., A.J.B. Y.O.
Data Curation	J.A.B.
Resources	J.A.B., J.P.R.
Funding Acquisition	J.A.B., J.P.R.
Writing - Original Draft	J.A.B.
Writing - Review & Editing	J.A.B., M.T.H., J.G., A.R.Q., J.P.R.
Visualization	J.A.B.