A pipeline to analyse PASP data

Reference

This repository contains the pipeline and raw results described in the manuscript:

- Botond Sipos, Adrian M. Stütz, Greg Slodkowicz, Tim Massingham, Jan Korbel, Nick Goldman: *PASP - a whole-transcriptome poly(A) tail length determination assay for the Illumina platform*.

Click here for more details on the wetlab experiment.

Using the pipeline

The pipeline can be used by invoking the following `make` targets:

- Fetch raw data from ENA: `make fetch`
- Generate transcriptome from SGD annotation: `make transcriptome`
- Align and parse reads: `make parse`
- Test for differential polyadenylation: `make test` or `make lsf_test`
- Parse spike-in reads (`make parse_spikeins`) and build “error model” (`make error_model`)

- Filter test results by G-tail coverage: `make gtail_cov_filter`
- Plot correlation between technical replicates: `make gtail_tech_corr`
- Plot and cluster tail length distributions: `make classify_tail_dists`
- Correlate thresholded tail lengths with PASTA and PAL-seq: `make corr_with_studies`

Index of selected raw results

Illumina reads

Sequencing data are available in the ArrayExpress database under accession number E-MTAB-2456.

Alignment

Sample	Alignment log	Alignment report
WT1A	http://bit.ly/PAT-seq-WT1A_aln_log	http://bit.ly/PAT-seq-WT1A_align_pdf
WT1B	http://bit.ly/PAT-seq-WT1B_aln_log	http://bit.ly/PAT-seq-WT1B_align_pdf
WT1C	http://bit.ly/PAT-seq-WT1C_aln_log	http://bit.ly/PAT-seq-WT1C_align_pdf`

WT1D	http://bit.ly/PAT-seq-WT1D_aln_log	http://bit.ly/PAT-seq-WT1D_align_pdf
WT2A	http://bit.ly/PAT-seq-WT2A_aln_log	http://bit.ly/PAT-seq-WT2A_align_pdf
WT2B	http://bit.ly/PAT-seq-WT2B_aln_log	http://bit.ly/PAT-seq-WT2B_align_pdf
WT2C	http://bit.ly/PAT-seq-WT2C_aln_log	http://bit.ly/PAT-seq-WT2C_align_pdf
WT2D	http://bit.ly/PAT-seq-WT2D_aln_log	http://bit.ly/PAT-seq-WT2D_align_pdf
MUT1A	http://bit.ly/PAT-seq-MUT1A_aln_log	http://bit.ly/PAT-seq-MUT1A_aln_pdf
MUT1B	http://bit.ly/PAT-seq-MUT1B_aln_log	http://bit.ly/PAT-seq-MUT1B_aln_pdf
MUT1C	http://bit.ly/PAT-seq-MUT1C_aln_log	http://bit.ly/PAT-seq-MUT1C_aln_pdf`
MUT1D	http://bit.ly/PAT-seq-MUT1D_aln_log	http://bit.ly/PAT-seq-MUT1D_aln_pdf
	http://bit.ly/PAT-seq-	http://bit.ly/PAT-seq-

MUT2A	MUT2A_aln_log	MUT2A_aln_pdf
MUT2B	http://bit.ly/PAT-seq-MUT2B_aln_log	http://bit.ly/PAT-seq-MUT2B_aln_pdf
MUT2C	http://bit.ly/PAT-seq-MUT2C_aln_log	http://bit.ly/PAT-seq-MUT2C_aln_pdf
MUT2D	http://bit.ly/PAT-seq-MUT2D_aln_log	http://bit.ly/PAT-seq-MUT2D_aln_pdf

Parsing alignments

Sample	Parse log	Parse report
WT1A	http://bit.ly/PAT-seq-WT1A_parse_log	http://bit.ly/PAT-seq-WT1A_parse_pdf
WT1B	http://bit.ly/PAT-seq-WT1B_parse_log	http://bit.ly/PAT-seq-WT1B_parse_pdf
WT1C	http://bit.ly/PAT-seq-WT1C_parse_log	http://bit.ly/PAT-seq-WT1C_parse_pdf
WT1D	http://bit.ly/PAT-seq-WT1D_parse_log	http://bit.ly/PAT-seq-WT1D_parse_pdf
WT2A	http://bit.ly/PAT-seq-	http://bit.ly/PAT-seq-

	WT2A_parse_log	WT2A_parse_pdf
WT2B	http://bit.ly/PAT-seq-WT2B_parse_log	http://bit.ly/PAT-seq-WT2B_parse_pdf
WT2C	http://bit.ly/PAT-seq-WT2C_parse_log	http://bit.ly/PAT-seq-WT2C_parse_pdf
WT2D	http://bit.ly/PAT-seq-WT2D_parse_log	http://bit.ly/PAT-seq-WT2D_parse_pdf
MUT1A	http://bit.ly/PAT-seq-MUT1A_parse_log	http://bit.ly/PAT-seq-MUT1A_parse_pdf
MUT1B	http://bit.ly/PAT-seq-MUT1B_parse_log	http://bit.ly/PAT-seq-MUT1B_parse_pdf
MUT1C	http://bit.ly/PAT-seq-MUT1C_parse_log	http://bit.ly/PAT-seq-MUT1C_parse_pdf
MUT1D	http://bit.ly/PAT-seq-MUT1D_parse_log	http://bit.ly/PAT-seq-MUT1D_parse_pdf
MUT2A	http://bit.ly/PAT-seq-MUT2A_parse_log	http://bit.ly/PAT-seq-MUT2A_parse_pdf
MUT2B	http://bit.ly/PAT-seq-MUT2B_parse_log	http://bit.ly/PAT-seq-MUT2B_parse_pdf

MUT2C	http://bit.ly/PAT-seq-MUT2C_parse_log	http://bit.ly/PAT-seq-MUT2C_parse_pdf
MUT2D	http://bit.ly/PAT-seq-MUT2D_parse_log	http://bit.ly/PAT-seq-MUT2D_parse_pdf

Quantifying tail length slippage using spike-in standards

- Tail run lengths until the first 1-5 non-A bases in reads mapped to spike-in poly(A) tracts PDF

Testing differences between wild type and mutant tail runs

Comparison	Test log	Test report	Results
WT1 vs. MUT1	TXT	PDF	CSV
WT2 vs. MUT2	TXT	PDF	CSV

Tail run distributions from all transcripts with G-tail coverage > 1000

- WT1
- WT2
- MUT1

- MUT2

Cross-study correlation

| WT1 vs. MUT1 | http://bit.ly/PAT-seq-TEST_WT1_vs_MUT1_log | http://bit.ly/PAT-seq-TEST_WT1_vs_MUT1_pdf | http://bit.ly/PAT-seq-TEST_WT1_vs_MUT1_trs_tab | | WT2 vs. MUT2 | http://bit.ly/PAT-seq-TEST_WT2_vs_MUT2_log | http://bit.ly/PAT-seq-TEST_WT2_vs_MUT2_pdf | http://bit.ly/PAT-seq-TEST_WT2_vs_MUT2_trs_tab |

Tail run distributions from all transcripts with G-tail coverage > 1000

- WT1: http://bit.ly/PAT-seq-CLS_WT1_pdf
- WT2: http://bit.ly/PAT-seq_CLS_WT2_pdf
- MUT1: http://bit.ly/PAT-seq-CLS_MUT1_pdf
- MUT2: http://bit.ly/PAT-seq-CLS_MUT2_pdf

Cross-study correlation

- PAL_total vs. WT1: http://bit.ly/PAT_seq_PAL_total_vs_WT1_pdf
- PAL_total vs. WT2: http://bit.ly/PAT-seq_PAL_total_vs_WT2_pdf
- PAL_total vs. PASTA: http://bit.ly/PAT-seq-PAL_total_vs_PASTA_pdf

Dependencies

- Platform LSF
- Python 2.x
- numpy >= 1.6.2
- matplotlib >= 1.1.0
- scipy >= 0.10.1
- biopython >= 1.60
- Bowtie2 >= 2.1.0
- samtools >= 0.1.19+
- wget

Using the analysis tools

The analysis tool can be found under `patsy/`:

patsy-align - classify read pairs and align them using Bowtie2

```
usage: patsy-align [-h] -1 fq1 -2 fq2 -f ref [-o outdir] [-s stats_pickle]
                   [-l gtail_sig] [-G gtag_min] [-N max_N] [-I min_fsize]
                   [-X max_fsize] [-p nr_threads] [-r report]

Align PASP reads using Bowtie2 (1.1).

optional arguments:
  -h, --help         show this help message and exit
  -1 fq1             First FASTQ file.
  -2 fq2             Second FASTQ file.
  -f ref             Reference fasta.
  -o outdir          Output directory.
  -s stats_pickle    Stats pickle file.
  -l gtail_sig       Portion of read start/end used for G-tail classification
                     (14).
  -G gtag_min        Minimum G-tag length(3).
  -N max_N           Maximum number of Ns in the first -l bases (6).
  -I min_fsize       Minimum fragment size (0).
  -X max_fsize       Maximum fragment size (500).
  -p nr_threads      Number of threads to use (1).
  -r report          Report PDF.
```

patsy-parse - parse classified and aligned PASP read pairs

```
usage: patsy-parse [-h] -g gtail_sam -n nvtr_sam -d dataset_id -f ref
                   [-l gtail_sig] [-G gtag_min] [-N max_N] [-e err_tol]
                   [-o out_pickle] [-i tr_list] [-q min_q] [-r report] [-t]
```

```
Parse classified and aligned PASP read pairs (1.1).

optional arguments:
  -h, --help     show this help message and exit
  -g gtail_sam   SAM file containing G-tail alignments.
  -n nvtr_sam    SAM file containing NVTR alignments.
  -d dataset_id  Dataset identifier.
  -f ref         Reference fasta.
  -l gtail_sig   Portion of read start/end used for G-tail
classification.
  -G gtag_min    Minimum G-tag length(3).
  -N max_N       Maximum number of Ns in the first -l bases
 (6).
  -e err_tol     Number of errors tolerated in the tail.
  -o out_pickle  Output pickle file.
  -i tr_list     List of transcripts considered.
  -q min_q       Mapping quality treshold (30).
  -r report      Report PDF.
  -t             Plot per-transcript coverage reports.
```

patsy-test - test for differential polyadenylation in PASP data

```
usage: patsy-test [-h] -a [a_pickles [a_pickles ...]] -na a
_name -b
                  [b_pickles [b_pickles ...]] -nb b_name [-
i lrt_list]
```

```
                  [-P lik_penalty] [-M min_size_U] [-s sig_
level]
                  [-op out_pickle] [-ot out_trs] [-og out_g
lob]
                  [-otr out_runs_prefix] [-orr out_rep_pref
ix] [-r report]
                  [-t]

Test for differential polyadenylation in PASP data (1.1).

optional arguments:
  -h, --help            show this help message and exit
  -a [a_pickles [a_pickles ...]]
                        Parsed read pickles - group A.
  -na a_name            Name of data group A.
  -b [b_pickles [b_pickles ...]]
                        Parsed read pickles - group B.
  -nb b_name            Name of data group B.
  -i lrt_list           Transcripts to be tested with ancho
rs LRT.
  -P lik_penalty        Log-likelihood penalty for data poi
nts outside valid
                        range.
  -M min_size_U         Minimum sample size when performing
 Mann-Whitney U
                        test (30).
  -s sig_level          Significance level.
  -op out_pickle        Output pickle file.
  -ot out_trs           Output tabular file: transcript properties.
  -og out_glob          Output tabular file: global results.
  -otr out_runs_prefix  Output tabular file: tail runs prefix.
  -orr out_rep_prefix   Output tabular file: tail means per replicate.
  -r report             Report PDF.
  -t                    Plot reports for all transcripts.
```

patsy-spike - estimate the number of sequencing errors in runs of bases.

```
usage: patsy-spike [-h] -n spike_sam -f ref [-w window] [-m max_errors_plot]
                   [-o out_pickle] [-q min_q] [-r report] [-l read_len]
                   [-pk pickle]

Estimate the number of sequencing errors in runs of bases (1.0).

optional arguments:
  -h, --help          show this help message and exit
  -n spike_sam        SAM file containing NVTR alignments.
  -f ref              Reference fasta with the spike-in seq
```

```
uences.
  -w window             Size of the flanking sequence around
the run of As.
  -m max_errors_plot  Maximum number of errors for which to
 plot the length
                      distribution.
  -o out_pickle       Output pickle file.
  -q min_q            Mapping quality treshold (30).
  -r report           Report PDF.
  -l read_len         Read length.
  -pk pickle          Result pickle file.
```