Baltica

Thiago Britto Borges

Table of contents

(https://github.com/dieterich-lab/Baltica/tree/master/docs/index.md)

Baltica: integrated splice junction usage analysis *(https://github.com/dieterich-lab/Baltica)*¶

Baltica is a framework that facilitates the execution and enables the integration of results from multiple differential junction usage (DJU) methods. The core of the framework is Snakemake workflows [1], a python command-line interface, and R/Bioconductor scripts for analysis [2][3][4][5]. The workflows are include methods for RNA-Seq quality control [6][7][8], four DJU methods: RMATs [9] JunctionSeq [10], Majiq [11] and Leafcutter [12]. We use Stringtie2 [13] *de novo* transcriptome assembly to re-annotate the results. Baltica's main goal is to provide an integrative view of the results of these methods. To do so, Baltica produces an RMarkdown report with the integrated results and links to UCSC GenomeBrowser for further exploration.

Features¶

```
- Snakemake workflows for DJU: junctionseq, majiq, rmats, and leafc
- Snakemake workflow for de novo transcriptome annotation with stri
- Process, integrate and annotate the results from the methods
- Summarise AS class of differently spliced junctions
- DJU method benchmarks
- Report on the integrative analysis
```

To get started, use the menu on the left-hand side or search function to navigate over this documentation.

Citation¶

Thiago Britto-Borges, Volker Boehm, Niels H. Gehring and Christoph Dieterich (2020) **Baltica: integrated splice junction usage analysis**. Manuscript in preparation.

Baltica is based on the work of many scientists and developers. Thus, if you use the results of their tools in your analysis, consider citing their work.

License¶

Baltica is free, open-source software released under an MIT License *(https://github.com/dieterich-lab/Baltica/blob/master/LICENSE)*.

Contact¶

Please get in touch with us the GitHub issue tracker *(https://github.com/dieterich-lab/Baltica/issues)*.

References¶

1. Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, and et al. Sustainable data analysis with snakemake. *F1000Research*, 10:33, Apr 2021. URL: http://dx.doi.org/10.12688/f1000research.29032.2 *(http://dx.doi.org/10.12688/f1000research.29032.2)*, doi:10.12688/f1000research.29032.2 *(https://doi.org/10.12688/f1000research.29032.2)*.

2. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL: https://www.R-project.org/ *(https://www.R-project.org/)*.

3. Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T. Morgan, and Vincent J. Carey. Software for computing and annotating genomic ranges. *PLoS Computational Biology*, 9(8):e1003118, Aug 2013. URL: http://dx.doi.org/10.1371/journal.pcbi.1003118 *(http://dx.doi.org/10.1371/journal.pcbi.1003118)*, doi:10.1371/journal.pcbi.1003118 *(https://doi.org/10.1371/journal.pcbi.1003118)*.

4. M. Lawrence, R. Gentleman, and V. Carey. Rtracklayer: an r package for interfacing with genome browsers. *Bioinformatics*, 25(14):1841–1842, May 2009. URL: http://dx.doi.org/10.1093/bioinformatics/btp328 *(http://dx.doi.org/10.1093/bioinformatics/btp328)*, doi:10.1093/bioinformatics/btp328 *(https://doi.org/10.1093/bioinformatics/btp328)*.

5. Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, and et al. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, Nov 2019. URL: http://dx.doi.org/10.21105/joss.01686 *(http://dx.doi.org/10.21105/joss.01686)*, doi:10.21105/joss.01686 *(https://doi.org/10.21105/joss.01686)*.

6. Liguo Wang, Shengqin Wang, and Wei Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics*, 28(16):2184–2185, June 2012. URL: https://doi.org/10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*, doi:10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*.

7. Simon Andrews, Felix Krueger, Anne Segonds-Pichon, Laura Biggins, Christel Krueger, and Steven Wingett. FastQC. Babraham Institute, January 2012. URL: https://qubeshub.org/resources/fastqc *(https://qubeshub.org/resources/fastqc)*.

8. Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, Jun 2016. URL: http://dx.doi.org/10.1093/bioinformatics/btw354 *(http://dx.doi.org/10.1093/bioinformatics/btw354)* , doi:10.1093/bioinformatics/btw354 *(https://doi.org/10.1093/bioinformatics/btw354)*.

9. Shihao Shen, Juw Won Park, Zhi-xiang Lu, Lan Lin, Michael D. Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. Rmats: robust and flexible detection of differential alternative splicing from replicate rna-seq data. *Proceedings of the National Academy of Sciences*, 111(51):E5593–E5601, Dec 2014. URL: http://dx.doi.org/10.1073/pnas.1419161111 *(http://dx.doi.org/10.1073/pnas.1419161111)* , doi:10.1073/pnas.1419161111 *(https://doi.org/10.1073/pnas.1419161111)*.

10. Stephen W. Hartley and James C. Mullikin. Detection and visualization of differential splicing in RNA-seq data with JunctionSeq. *Nucleic Acids Research*, pages gkw501, June 2016. URL: https://doi.org/10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*, doi:10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*.

11. Jorge Vaquero-Garcia, Alejandro Barrera, Matthew R Gazzara, Juan Gonzalez-Vallinas, Nicholas F Lahens, John B Hogenesch, Kristen W Lynch, and Yoseph Barash. A new view of transcriptome complexity and regulation through the lens of local splicing variations. *eLife*, February 2016. URL: https://doi.org/10.7554/elife.11752 *(https://doi.org/10.7554/elife.11752)*, doi:10.7554/elife.11752 *(https://doi.org/10.7554/elife.11752)*.

12. Yang I. Li, David A. Knowles, Jack Humphrey, Alvaro N. Barbeira, Scott P. Dickinson, Hae Kyung Im, and Jonathan K. Pritchard. Annotation-free quantification of RNA splicing using LeafCutter. *Nature Genetics*, 50(1):

151–158, December 2017. URL: https://doi.org/10.1038/s41588-017-0004-9 *(https://doi.org/10.1038/s41588-017-0004-9)*, doi:10.1038/s41588-017-0004-9 *(https://doi.org/10.1038/s41588-017-0004-9)*.

13. Sam Kovaka, Aleksey V. Zimin, Geo M. Pertea, Roham Razaghi, Steven L. Salzberg, and Mihaela Pertea. Transcriptome assembly from long-read rna-seq alignments with stringtie2. *Genome Biology*, Dec 2019. URL: http://dx.doi.org/10.1186/s13059-019-1910-1 *(http://dx.doi.org/10.1186/s13059-019-1910-1)* , doi:10.1186/s13059-019-1910-1 *(https://doi.org/10.1186/s13059-019-1910-1)*.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/intro.md)

Introduction¶

For motivation and review of state of the art, please check:

Thiago Britto-Borges, Volker Boehm, Niels H. Gehring and Christoph Dieterich (2020) Baltica: integrated splice junction usage analysis. Manuscript in preparation.

Tips on RNA-Seq aiming differential splicing detection¶

If you aim to resolve mRNA isoforms with relatively low abundance, you should design the RNA-seq experiment accordingly. The expert suggestion *(https://support.illumina.com/bulletins/2017/04/considerations-for-rna-seq-read-length-and-coverage-.html)* is to sequence around 40 to 60 million reads pairs. This parameter is particularly relevant for complex RNA libraries, but it can be insufficient to saturate novel SJ, in our experience. Read length and paired-end reads are also critical for SJ identification, and longer reads offer more coverage of the exons boundaries. Thus, the target read length should be around 100 nucleotides for Illumina RNA-seq to maximize the read overhang length and, consequently, maximize the quality of the alignments.

In the Baltica manuscript, we propose an approach to integrate DJU results from Illumina to DJU results from third-generation sequencing.

Also, databases such as the CHESS *(http://ccb.jhu.edu/chess/)* can provide additional evidence for splice sites absent in the annotation.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/setup.md)

Getting started¶

Quick example:¶

If Baltica dependencies, baltica configuration and cluster configuration are available, use:

```
baltica <workflow> <config> --use-singularity
```

- workflow:
  - all: run end-to-end wokflows
  - qc: run quality control
  - stringtie: run *de novo* and guided transcriptome assembly
  - rmats
  - junctionseq
  - majiq
  - leafcutter
  - analysis: run scripts for integration, annotation and reporting
- config: project configuration file

Use `--profile <cluster>` with a Snakemake cluster profile *(https://snakemake.readthedocs.io/en/stable/executing/cli.html#profiles)* or set the number of avaiable cores with `--cores`. For more Snakemake parameters, check their documentation *(https://snakemake.readthedocs.io/en/stable/executing/cli.html)*.

Warning

Baltica is under active development. Please contact us *(https://github.com/dieterich-lab/Baltica/issues)* if you have any issues with this documentation.

Software environment:¶

Baltica framework is based on:
- A python command-line interface
- Snakemake *(https://snakemake.readthedocs.io/en/stable/)* workflows
- Docker containers used with Singularity *(https://sylabs.io/singularity/)*
- R scripts for processing, integrating, annotating, assigning biological features, and reporting
- a Rmarkdown report

We have developed it on the following computer environments: - Linux version 4.19.0-16-amd64 Debian 4.19.181-1 (2021-03-19)

- gcc version 8.3.0
- Python version 3.7.7
- Singularity version 3.7.3
- Snakemake version 6.4.1
- Git version 2.20.1

These versions should not matter because the workflows are run within Docker containers, as long **Snakemake version > 6** and a **recent Singularity version**.

Baltica depends on python3, Singulary, and Snakemake:
- How to install Singularity *(https://sylabs.io/guides/3.0/user-guide/installation.html)*
- How to install Snakemake *(https://snakemake.readthedocs.io/en/stable/getting_started/installation.html)*

Installation¶

```
git clone https://github.com/dieterich-lab/baltica
cd baltica
pip install .
```

Will install Baltica and its python dependencies. You may want to create a virtual environment *(https://realpython.com/python-virtual-environments-a-primer/)* before installing Baltica.
All other requirements are resolved with singularity containers. Baltica store its singularity containers at `$HOME/.baltica/singularity/`.

Note

We plan to submit Baltica to the Python Package Index.

Warning

majiq requires an Academic or Commercial license for use. Users are required to obtain their licenses. *(https://majiq.biociphers.org/app_download/)*.

Executing Baltica¶

Use baltica `cli` for current help documentation:

```
baltica --help
```

Baltica executor takes a single optional argument `--verbose`, to detail its execution. Every other option is passed to Snakemake.

Test dataset¶

Baltica ships with a test dataset, located at the `data/` directory. There is a configuration file for the test dataset. **Users are required to update this configuration file**. Please see the Baltica project configuration for further details.

Cluster profile¶

Snakemake supports distributed workflow execution in many different high-performance computer clusters, as detailed here *(https://snakemake.readthedocs.io/en/stable/executing/cluster.html?highlight=profile#cluster-execution)*. We recommend using cluster profiles *(https://snakemake.readthedocs.io/en/stable/executing/cli.html#profiles)* and using it like:

```
baltica <workflow> <config> --use-singularity --profile <cluster>
```

(Advanced) Baltica workflows directly from Snakemake¶

Baltica workflows can be used directly with Snakemake without installation. However, there is limited support for it.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/proj-config.md)

Baltica project configuration:¶

Baltica requires a project configuration file as input. For a template see here *(https://raw.githubusercontent.com/dieterich-lab/Baltica/master/baltica/config.yml)*. For a programmatic solution to generate the configuration, use the script baltica/write_new_config.R *(https://github.com/dieterich-lab/Baltica/blob/8bc5fe5f71e948b3971ea5db5f1456b2d9e2f838/baltica/write_new_config.R)*. Method specific parameters are detailed in the Workflow implementation page.

Warning

You are required to update the configuration. Use the full path to files.

Note

The `Required` column flags parameters without default.

General parameters description¶

Parameter	Description	Required
path	project path	✓
sample_path	path to the parent directory for aligment files	✓
samples	sample name and directory formated as "{sample_name}:{path_to_sample}	✓
contrasts	list of contrasts names (format)	✓
assembly	assembly name on UCSC Browser	
strandness	one of fr-firststrand, fr-secondstrand or none (unstranded)	✓
read_len	maximun read length	✓
ref	path to reference annotation in the GTF format	✓
ref_fa	path to reference annotation in the FASTA format	✓
project_authors	project author name, used in the report	
project_title	project title name, used in file names and report	
orthogonal_result	result from Nanopore-seq in GFF or BED with a valid score column, and optionally a comparisons column with contrasts	

Pairwise comparisons¶

```
contrasts:
  {case1}-vs-{control}:
    - {case1}
    - {control}
  {cas2}-vs-{control}:
    - {case2}
    - {control}
```

Note

junctionseq and leafcutter support more complex experimental designs, which were not implemented in Baltica.

majiq specific paramers¶

majiq manual *(https://biociphers.bitbucket.io/majiq/MAJIQ.html#builder)*

Parameter	Original parameter	Description
rule majiq_build: (`majiq deltapsi`)		
majiq_min_experiments	--min-experiments	minimum number of experiments to filter with --minreads (default 1.0 - all experiments)
majiq_minreads	--minreads	Discard SJ with less than `--minreads` reads
majiq_min_denovo	--min-denovo	Discard novel SJ with less than `--min-denovo` reads
rule majiq_deltapsi (`majiq deltapsi`)		
majiq_minreads	--minreads	same as above
rule majiq_voila (`voila tsv`)		
majiq_non_changing_threshold	--non-changing-threshold	
majiq_threshold	--threshold	

junctionseq specific paramers¶

qorts manual *(https://hartleys.github.io/QoRTs/doc/QoRTs-vignette.pdf)* junctionseq manual *(http://hartleys.github.io/JunctionSeq/doc/JunctionSeq.pdf)*

Parameter	Original parameter	Description
rule junctionseq_qc (`qorts QC`)		
is_single_end	--singleEnded	Flag single end libraries
rule junctionseq_merge (`qorts mergeNovelSplices`)		
junctionseq_mincount	--minCount	Discard SJ with less than `--minCount` reads

leafcutter specific paramers¶

regtools manual *(https://regtools.readthedocs.io/en/latest/)* leafcutter manual *(http://davidaknowles.github.io/leafcutter/articles/Usage.html)*

Parameter	Original parameter	Description
rule leafcutter_bam2junc (`regtools junctions extract`)		
leafcutter_minimum_anchor_length	-a	Discard reads with overanging length lower than `-a`
leafcutter_minimum_intron_size	-i	Minimum intron size
leafcutter_maximum_intron_size	-I	Maximum intron size
rule leafcutter_differential_splicing		
leafcutter_min_coverage	--min_coverage	desc
leafcutter_min_samples_per_group	-g	Discard SJ used in less than `-g` samples
leafcutter_min_samples_per_intron	-i	Discard SJ with less than `--min_coverage` in `-i` samples in each group

rmats specific paramers¶

rmats manual *(https://github.com/Xinglab/rmats-turbo#all-arguments)*

Parameter	original parameter	Description
rule rmats_run (`rmats.py`)		
rmats_allow_clipping	--allow-clipping	Allow clipped reads
rmats_variable_read_length	--variable-read-length	Allow reads with variable-length
rmats_novel_ss	--novelSS	Allow the detecting unannotated SJ
rmats_extra	none	Pass extra arguments to `rmats.py`

(https://github.com/dieterich-lab/Baltica/tree/master/docs/workflows.md)

Workflow implementation¶

This chapter details the implementation and usage of each workflow in Baltica.

Baltica comprises a collection of Snakemake workflows (SMK files). Each file determines a series of sub-tasks (rules). The sub-tasks run in a specific order; once the output of every rule is complete, the workflow is considered successful. We implemented the workflows following instructions and parameters suggested by the methods authors unless otherwise noted.

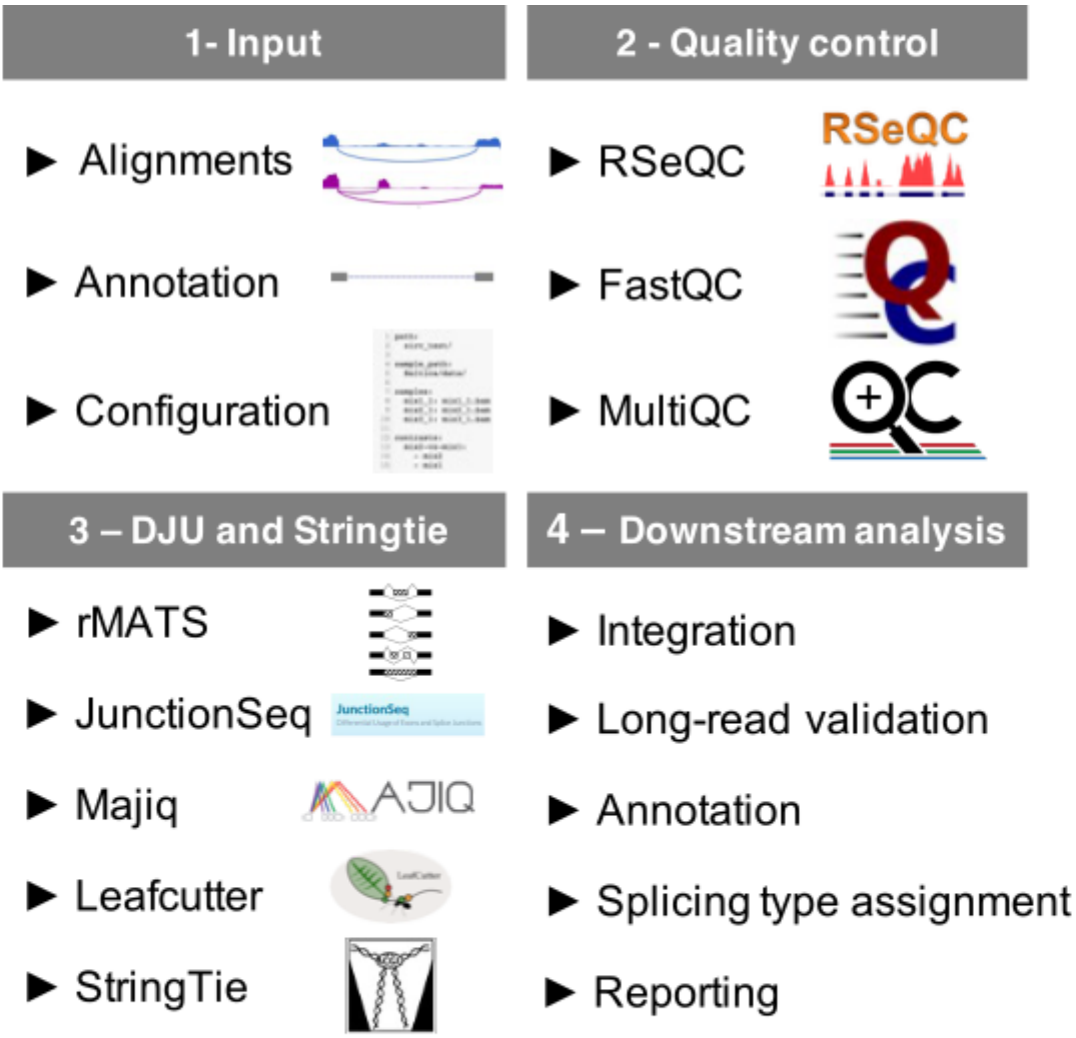

Fig. 1 - **Baltica overview**: Baltica is a framework to execute and integrate differential junction usage (DJU) analysis and further investigation enabled by the data integration. 1 -- **Input**: Baltica takes as input RNA-seq alignments, reference annotation, and a configuration file. 2 -- **Quality control**: As the first step of the pipeline, Baltica performs quality control of alignments with `RSeQC` and `FastQC`, which is reported by `MultiQC`. 3 -- **DJU and transcriptome assembly**: Next, Baltica computes DJU with `JunctionSeq`, `Majiq`, and `Leafcutter`, and uses `Stringtie2` to detected new transcripts and exons in the dataset. 4 -- **Downstream analysis**: Finally, we integrate the results from the DJU method. Optionally, Baltica can include an extra piece of evidence for DJU, such

as DJU obtained from third-generation sequencing, to the integrated table. The set of introns is re-annotated using information from *de novo* transcriptome annotation, and splice types between SJ and exons are assigned. Finally, a Baltica compiles a report with the most relevant information.

Quality control workflow¶

Executed with:

```
baltica qc <config> --use-singularity
```

The first workflow comprises the quality control of the read alignments. This step aims to determine the success of sequencing and alignment. Baltica includes workflows for RSeQC [1] and FastQC [2]. MultiQC [3] summarizes the output from both tools. In addition, users can use parameters from QC workflow into Baltica, such as maximum read length and library type.

Beyond the quality control, this step may help to identify differences among the RNA libraries. For example, RSeQC provides the proportion of reads per feature in the input annotation. Differences between case vs. control, such as enrichment of reads aligned to introns, may suggest technical artifacts or global changes in splicing. In addition, RSeQC provides the `junction_saturation.py` method, which quantifies the abundance of known and novel SJ in the RNA-seq alignments, and diagnoses if the alignment coverage detects known and novel splice junctions in sub-samples for alignments. Thus, users can use this functionality to identify the saturation of annotated and unannotated SJ, or a higher level of coverage is needed. In conclusion, the quality control step serves to identify potential problems with the RNA-Seq library alignment and, potentially, direct on further troubleshooting and downstream analysis.

Software dependencies *(https://github.com/dieterich-lab/Baltica/blob/master/envs/qc.yml)*

RMATs workflow¶

Executed with:

```
baltica rmats <config> --use-singularity
```

RMATs [4] workflow is done in two steps: - Determine the experimental groups.
- Run `rmats.py`.

Running RMATs prep and post tasks separately *(https://github.com/Xinglab/rmats-turbo/tree/8a2ad659717a1ccd6dbecd593dc1370ba7c30621#running-prep-and-post-separately)* and paired statistical test were not implemented in Baltica.

Software dependencies *(https://github.com/dieterich-lab/Baltica/blob/master/envs/rmats.yml)*

JunctionSeq workflow¶

Executed with:

```
baltica junctionseq <config> --use-singularity
```

JunctionSeq [5] workflows starts by junction read counts extraction done with QoRTs [6]. In Baltica implementation for JunctionSeq workflow, we only consider reads that span multiple exons (splice junction reads, SJ) for annotated and unannotated introns, ignoring exon counts. JunctionSeq uses disjoint genomic bins to flatten the transcriptome annotation. To test the hypothesis that features are differently expressed in experimental groups, JunctionSeq fits a generalized linear model, as described in DEXSeq [7], but reporting a test statistic at the genomic feature and gene level. Unlike other DJU methods, JunctionSeq does not group the introns or S in AS events, so it does not compute PSI events but rather log fold change.

Baltica parses the `*_sigGenes.results.txt.gz` (located at `junctionseq/analysis/`) and discard entries that were flagged as not testable.

Software dependencies *(https://github.com/dieterich-lab/Baltica/blob/master/envs/junctionseq.yml)* and docker image recipe *(https://github.com/dieterich-lab/Baltica/blob/master/docker/junctionseq/1.16.0/Dockerfile)*.

Majiq workflow¶

Executed with:

```
baltica majiq <config> --use-singularity
```

Majiq workflow includes the following steps: 1. Create a configuration file (`majiq/build.ini`) 1. Converts the reference annotation from gtf to gff with `gtf2gff3.pl` 1. **majiq build** generates the Splice Graph database with exons and SJ from the RNA-Seq experiment and the reference annotation 1. **majiq deltapsi**: computes ψ and Δψ and tests if the Δψ significantly changes between comparisons. Introns are called significant if the probability of Δψ > `threshold` is higher than `non-changing-threshold`, where `threshold` and `non-changing-threshold` are the `--threshold` and `--non-changing-threshold` parameters, respectively 1. **voila tsv**: filter and outputs the Majiq result to a tab-separated value file

Majiq visualization methods, such as `voila view`, are not currently implemented in Baltica but can be used independently.

Baltica parses the `{comparison}_voila.tsv` files - one per comparison, located at `majiq/voila/`.

Docker image recipe *(https://github.com/dieterich-lab/Baltica/blob/master/docker/majiq/2.2/Dockerfile)*.

Leafcutter workflow¶

Executed with:

```
baltica leafcutter <config> --use-singularity
```

Leafcutter uses Regtools[8] to extract SJ reads from the BAM files. Next, introns with at least `mincluereads` reads clustered. The clustering procedure iteratively discards introns supported by less than `mincluratio` reads within a cluster. Finally, Leafcutter fits a Dirichlet-Multinominal model, which determines the SJ usage for each cluster the usage (proportion) of a giving SJ within a cluster and compare this usage among conditions

The relevant output files from Leafcutter have the `_cluster_significance.txt` and `_effect_sizes.txt` suffix, computed for each comparison.

Column description:

`*_cluster_significance.txt`: 1. `cluster`: `{chromosome}:{intron_start}:{intron_end}` 1. `Status`: is this cluster testable? 1. `loglr`: the log-likelihood ratio between the null model and alternative 1. `df`: degrees of freedom, equal to the number of introns in the cluster minus one (assuming two groups) 1. `p` unadjusted p-value dor the under the asymptotic Chi-squared distribution

`*_effect_sizes.txt`: 1. `intron`: intron identifier on the format `chromosome:intron_start:intron_end:cluster_id` 1. `es`: effect size 1. `{cond_1}`: fitted junction usage in condition `cond_1` 1. `{cond_2}`: fitted junction usage in condition `cond_2` 1. `deltapsi`: difference between usage in the two conditions

Software dependencies *(https://github.com/dieterich-lab/Baltica/blob/master/envs/leafcutter.yml)* and Docker image recipe *(https://github.com/dieterich-lab/Baltica/blob/master/docker/leafcutter/0.2.7/Dockerfile)*.

Stringtie workflow¶

Baltica uses the gene, transcript, and class code assignments from the Stringtie output to the SJ from the DJU method outputs. In addition, exons defined by this annotation are used for assignments of splicing event types.

We process *de novo* transcriptomic workflow with Stringtie [9]. First, we merge the alignment files from biological replicates. Next, we compute *de novo* annotation with Stringtie, using the parameters `-c 3 -j 3 -f 0.01`. Finally, the merge the multiple annotation with `gffcompare -r {reference_annotation.gtf} -R -V`. Details on the parameter selection are in the Integration chapter.

Docker image recipe *(https://github.com/dieterich-lab/Baltica/blob/master/docker/stringtie/2.1.5/Dockerfile)*

References¶

1. Liguo Wang, Shengqin Wang, and Wei Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics*, 28(16):2184–2185, June 2012. URL: https://doi.org/10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*, doi:10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*.

2. Simon Andrews, Felix Krueger, Anne Segonds-Pichon, Laura Biggins, Christel Krueger, and Steven Wingett. FastQC. Babraham Institute, January 2012. URL: https://qubeshub.org/resources/fastqc *(https://qubeshub.org/resources/fastqc)*.

3. Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, Jun 2016. URL: http://dx.doi.org/10.1093/bioinformatics/btw354 *(http://dx.doi.org/10.1093/bioinformatics/btw354)* , doi:10.1093/bioinformatics/btw354 *(https://doi.org/10.1093/bioinformatics/btw354)*.

4. Shihao Shen, Juw Won Park, Zhi-xiang Lu, Lan Lin, Michael D. Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. Rmats: robust and flexible detection of differential alternative splicing from replicate rna-seq data. *Proceedings of the National Academy of Sciences*, 111(51):E5593–E5601, Dec 2014. URL: http://dx.doi.org/10.1073/pnas.1419161111 *(http://dx.doi.org/10.1073/pnas.1419161111)* , doi:10.1073/pnas.1419161111 *(https://doi.org/10.1073/pnas.1419161111)*.

5. Stephen W. Hartley and James C. Mullikin. Detection and visualization of differential splicing in RNA-seq data with JunctionSeq. *Nucleic Acids Research*, pages gkw501, June 2016. URL: https://doi.org/10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*, doi:10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*.

6. Stephen W. Hartley and James C. Mullikin. QoRTs: a comprehensive toolset for quality control and data processing of RNA-seq experiments. *BMC Bioinformatics*, Jul 2015. URL: http://dx.doi.org/10.1186/s12859-015-0670-5 *(http://dx.doi.org/10.1186/s12859-015-0670-5)*, doi:10.1186/s12859-015-0670-5 *(https://doi.org/10.1186/s12859-015-0670-5)*.

7. S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10):2008–2017, June 2012. URL: https://doi.org/10.1101/gr.133744.111 *(https://doi.org/10.1101/gr.133744.111)*, doi:10.1101/gr.133744.111 *(https://doi.org/10.1101/gr.133744.111)*.

8. Kelsy C. Cotto, Yang-Yang Feng, Avinash Ramu, Zachary L. Skidmore, Jason Kunisaki, Megan Richters, Sharon Freshour, Yiing Lin, William C. Chapman, Ravindra Uppaluri, and et al. Regtools: integrated analysis of genomic and transcriptomic data for the discovery of splicing variants in cancer. pre-print, Oct 2018. URL: http://dx.doi.org/10.1101/436634 *(http://dx.doi.org/10.1101/436634)* , doi:10.1101/436634 *(https://doi.org/10.1101/436634)*.

9. Mihaela Pertea, Geo M Pertea, Corina M Antonescu, Tsung-Cheng Chang, Joshua T Mendell, and Steven L Salzberg. Stringtie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*, 33(3):290–295, Feb 2015. URL: http://dx.doi.org/10.1038/nbt.3122 *(http://dx.doi.org/10.1038/nbt.3122)*, doi:10.1038/nbt.3122 *(https://doi.org/10.1038/nbt.3122)*.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/integration.md)

DJU methods result integration

Parsing the results of the method¶

The first step in the analysis workflow is parsing and processing the DJU methods' output with `scripts/parse_{method}_output.R` scripts as follows:

1. The resulting text output from the DJU methods is parsed and loaded as R data frames
2. The data frames are pivoted in a longer format to have one junction and one comparison per row

> **Warning**
>
> RMATs, Majiq, and Leafcutter use AS events to test for DJU, so metrics are associated with a group and not the SJ. In Baltica, we split these groups in SJ, and multiple SJ may have the same metric, for example, test statistics.

Result integration¶

One challenge for the integration of DJU results is that the methods use different genomic coordinate systems. The coordinates system's differences are due to the method implementation: methods can be 0-indexed (BED format) versus 1-indexed (GTF format) or use the exonic versus intronic coordinates to represent the SJ genomic position.
We propose a `filter_hits_by_diff` function to find overlapping features and then discard any overlaps with more than two bp differences to account for the multiple genomic coordinates system. The multiple hits form a graph, which is then partitioned into the clusters, and each cluster represents an intron. This feature enables the reconciliation of the multiple DJU results.

Annotating the results¶

We annotate the results with information from genes and transcripts hosting the SJ. For this, we use the *de novo* transcript annotation at `stringtie/merged/merged.combined.gtf`. Commonly, multiple transcripts share an intron so that a single intron may be annotated with multiple transcripts.

These are the columns assigned after the annotation:

Table 1: Annotation description Column name | Description | ------------|-------------| comparison | pairwise comparison as `{case}_vs_{control}` | chr | seqname or genomic contig | start | intron start position for the SJ | end | intron end position | strand | RNA strand that encodes that gene | gene | the gene symbol | e2 - e1| acceptor and donor exons number, if in + strand else the

inverse | tx_id | transcript identifier from the combined annotation | transcript_name | transcript name | class_code | association between reference transcript and novel transcript (seq fig1 for details *(https://doi.org/10.12688/f1000research.23297.1)*) |

Selecting optimal parameters for *de novo* transcriptome assembly¶

Warning

The section below was obtained in a previous Baltica release, using stringtie v1.2.X, but we don't expect major changes in current version.

We found that the parameters used to obtain the *de novo* transcriptome are critical for maximum integration between the GTF and the SJ from DJU methods. Fig 1 shows a parameter scan where we vary the group, `-j` (minimum junction coverage), `-c` (minimum coverage), and `-f` (minimum isoform proportion) and compute the number of transcripts that match with SJ called significantly. As expected, the merged annotation and not the group-specific annotation have the highest rate of annotated introns. The crucial result here is the dependency of the `-f` parameter, which is also associated with an increased number of annotated introns. As we confirmed this behavior in other datasets, we decided to use `-c 3 -j 3 -f 0.01` as default values in Baltica. The higher coverage (`-c` and `-j`) values counter the potential noise of transcripts with low abundance.

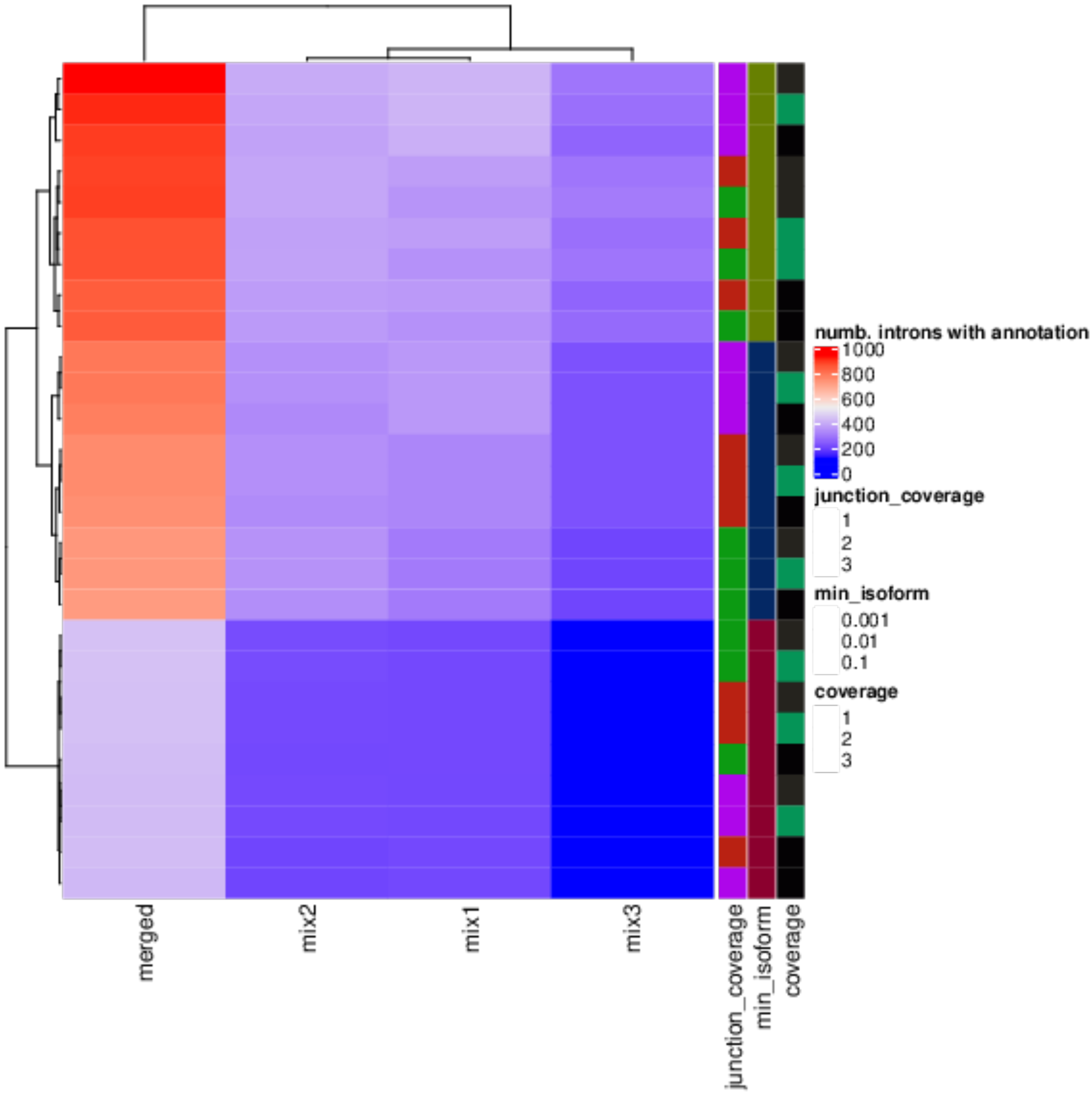

Fig 1:Parameter scan to maximize the number of introns annotated. We have run Stringtie with multipleparameters of merged annotation or group annotation; junction coverage of 1, 2, or 3; coverage of 1, 2, 3, and minimum isoform fraction of 0.1, 0.01, or 0.001. The result shows a dependency of the minimum isoform fraction parameter, which needs to be minimized to increase the proportion of annotated SJ, as expected.

Assigning AS type¶

Biological motivation¶

Identifying the type of AS is critical to understand a potential molecular mechanism for AS events. SRSF2 *(https://www.uniprot.org/uniprot/Q01130)* is a relevant example in this context. SRSF2 is splicing factors from the SR family that are known for auto-regulation. In certain conditions, the SRSF2 transcript can activate the nonsense-mediated decay by either including a new exon containing a premature stop codon or an intron in 3' UTR. These changes lead to transcript degradation and overall reduction of gene expression. Thus, the reduction of SRSF2 protein level leads to widespread exon skipping. Identifying such patterns is critical to understanding which splicing regulators are driving the observed splicing changes.

Implementation¶

In Baltica, we use a geometric approach to define AS in three classes: - ES, for exon skipping - A3SS, for alternative 3' splice-site - A5SS, for alternative 5' splice-site

Figure 2 details how we use the distance between features start and end to determine the AS type.

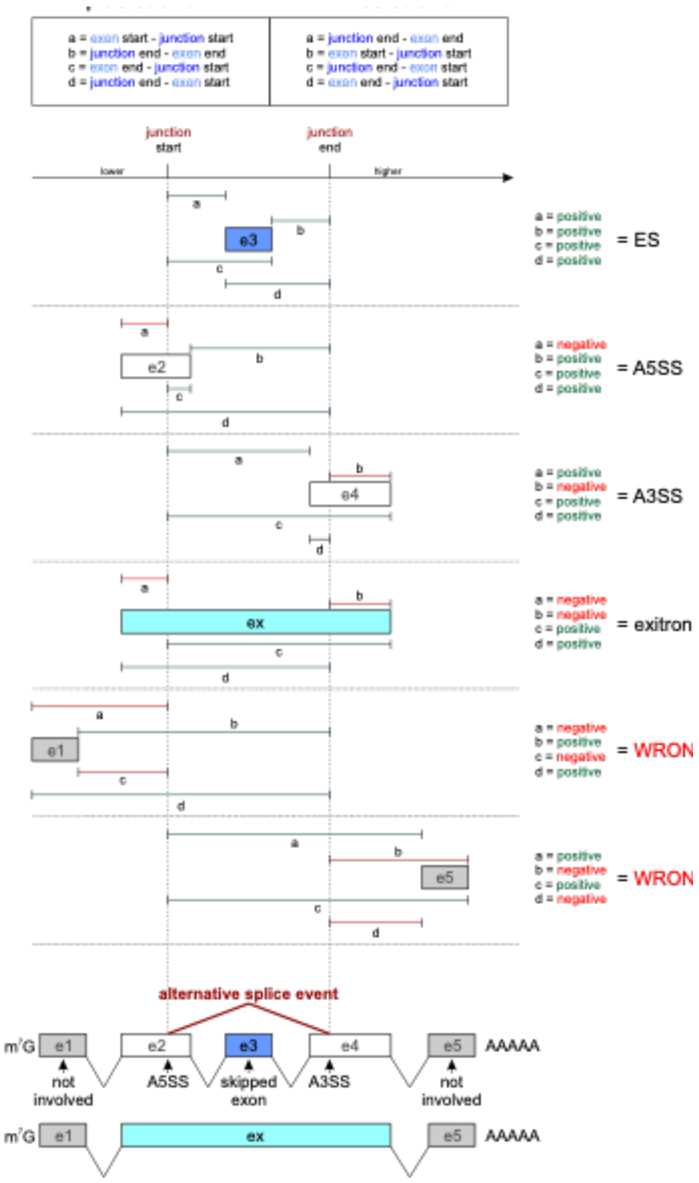

Fig 2: AS type assignment in Baltica. Baltica uses the genomic coordinates from the SJ and its overlapping exons to assigning AS type to SJ and its overlapping exons. Because many exons may be affected, multiple assignments are output. For example, donor and acceptor exons are assigned as JS and JE, respectively.

Simplify the AS event¶

Because most of the final users are only interested in the list of genomic ranges, gene names, or event types, we offer a simplified output that removes redundant information. This step helps generate a final report.

References¶

(https://github.com/dieterich-lab/Baltica/tree/master/docs/dev_guide.md)

Development guidelines:¶

Contributing to the documentation¶

For the docs, we use MkDocs *(https://www.mkdocs.org/)* because of its flexibility:
- mkdocs-material *(https://squidfunk.github.io/mkdocs-material/getting-started/)*: look and feel
- mkdocs-bibtex *(https://github.com/shyamd/mkdocs-bibtex)*: literature reference
- MkPDFs *(https://comwes.github.io/mkpdfs-mkdocs-plugin/getting-started.html)*: PDF version

Modify any of the doc files¶

```
vi docs/setup.md
```

Test the changes locally¶

```
mkdocs serve
```

If everything looks fine you can submit a patch or pull-request.

Deploy changes¶

This requires permissions from the GitHub organization.

```
mkdocs gh-deploy
```

Setting up mkdocs¶

```
# osx specific settings
conda install pango cairo

pip install mkdocs
pip install mkdocs-material
pip install mkdocs-bibtex
pip install -e git+https://github.com/jwaschkau/mkpdfs-mkdocs-plugi

# osx specific settings
```

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
```

Updating docker containers¶

The dockerfiles for containers reside at the `docker/` directory. Some of the environments use conda recipes, which reside in the `envs/` directory. After updating a recipe, to build and upload its container, one should use:

```
cd Baltica/
docker build -f <dockerfile> --tag <name>:<tag> .
docker push <tag>
```

For example, - dockerfile: `docker/baltica/1.0/Dockerfile` - name: `tbrittoborges/baltica` - tag: `1.0`

Once the container is updated, its tag version (tag) should be updated as well as it the container directive in the snakemake workflows.

docker hub *(https://hub.docker.com/repository/docker/tbrittoborges/)* hosts the container and can change this location at the *container* directive at the snakefiles.

Testing Baltica¶

Baltica's continuous integration testing suite is under development.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/faq.md)

Frequently asked questions

What you mean with `baltica provides a interface between users and DJU methods`:¶

There are many specificities to the DJU methods, and while running one method is not too complicated, figuring out how to run multiple methods is time-demanding. Baltica aims to facilitate this task so that methods results can be produced and compared.

Snakemake `Error: Directory cannot be locked.`:¶

This error happens when there is an error or failure during the workflow execution, and Snakemake's process does not have the opportunity to unlock the directory. Use `baltica <workflow> <config> --unlock` to resolve it. See more here *(https://snakemake.readthedocs.io/en/stable/project_info/faq.html#how-does-snakemake-lock-the-working-directory)*

ERROR lines in majiq_gtf_to_gff:¶

Most of the errors in this rules are not fatal. See here *(https://manpages.debian.org/unstable/gbrowse/gtf2gff3.1p.en.html)* for the diagnostics.

rmats empty or mostly empty outputs¶

This error can be either issue with: - The read length parameter *(https://github.com/Xinglab/rmats-turbo/issues/95)*. To resolve it, increase the read length parameter or use `--variable-read-length` in Baltica configuration. - Or an error with the stack size limit *(https://github.com/Xinglab/rmats-turbo/issues/91)*. To resolve it increase the stack size in bash with `ulimit -c unlimited`.

Junctionseq bpapply error:¶

This error occurs in the `junctionseq_analysis` rule, which uses multiple threads with BiocParallel. One can overcome this issue by setting threads to 1 on the said rule.

Junctionseq analysis thread error¶

`Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index xxx out of bounds for length xxx` It is complaining the maximum read length is longer than the read length input. First, check the maximum read length in the quality control report and then increase the `read_len` parameter on Baltica config.

How does Baltica compute the score for each DJU method?¶

The different DJU methods are pretty different in many aspects, including how they compute the final test statistic, and we use the following rule to compute the score for the Baltica table (higher is better):
- majiq score = 1 - non-changing-threshold (probability of |ΔΨ| > 0.2, by default)
- leafcutter = 1 - p.adjust
- junctionseq = 1 - padjust
- rmats = 1 - FDR

I see a message: `/bin/bash: /root/.bashrc: Permission denied`. What is wrong?¶

This error message is benign and, in our experience, does not affect workflow execution.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/tutorial.md)

Step by step tutorial with the sample data set¶

Setup¶

Make sure you have Singularity and Snakemake up and running. We have experienced problems with `singularity pull` and docker images when using a temporary directory (TMPDIR) in the shared file system. Setting `TMPDIR=/tmp/` resolves this issue.

Installation¶

Follow Installation guide

Configuration¶

Open `Baltica/data/config.yml` and replace `/beegfs/homes/tbrittoborges/` or `/home/tbrittoborges` to your desired path. The **path** parameter specifies where the project will be located. There is no needs to change **baltica_path**, as Baltica resolves it.

Execution¶

Execute Baltica with `baltica all /Baltica/config.yml --use-singularity`. Add **--quiet** to reduce the logging level or **--verbose** to increase it. Use a cluster profile (**--profile**) to take advantage of the cluster scheduler. See more useful parameters for snakemake below.

You should expect a `results/` directory containing the most relevant files by the end of the run. The report and excel table, `results/baltica_report{project_title}.html` and `results/baltica_table{project_title}.xlsx`, are the most relevant files. Intermediate files are kept in directories for each method (named for the methods), and can be used or deleted.

Important snakemake parameters¶

- **--cores**: only required if you are not using a cluster scheduler. Use `baltica ... --cores all` to specify the maximum number of cores available.
- **--profile**: setup a cluster configuration profile *(https://snakemake.readthedocs.io/en/stable/executing/cli.html#profiles)*.

- **--dry-run**: only computes the DAG, does not execute rules. But execution order may not reflect the actual order of execution.
- **--unlock**: unlock the directory if a previous run had problems.
- **--list-untracked**: list files that are not tracked by workflows, useful if you want to clean up the directory to save disk space after a successful run.
- **--quiet**: less verbose output.
- **--reason**, **--printshellcmds**, **--verbose**: verbose output. `Simple use baltica ... --verbose` to get maximum debug information (you may want to redirect it to a file).

See more detail at the Snakemake docs *(https://snakemake.readthedocs.io/en/stable/executing/cli.html#useful-command-line-arguments)*.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/bibliography.md)

References

1: Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, and et al. Sustainable data analysis with snakemake. *F1000Research*, 10:33, Apr 2021. URL: http://dx.doi.org/10.12688/f1000research.29032.2 *(http://dx.doi.org/10.12688/f1000research.29032.2)*, doi:10.12688/f1000research.29032.2 *(https://doi.org/10.12688/f1000research.29032.2)*. 2: R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL: https://www.R-project.org/ *(https://www.R-project.org/)*. 3: Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T. Morgan, and Vincent J. Carey. Software for computing and annotating genomic ranges. *PLoS Computational Biology*, 9(8):e1003118, Aug 2013. URL: http://dx.doi.org/10.1371/journal.pcbi.1003118 *(http://dx.doi.org/10.1371/journal.pcbi.1003118)*, doi:10.1371/journal.pcbi.1003118 *(https://doi.org/10.1371/journal.pcbi.1003118)*. 4: M. Lawrence, R. Gentleman, and V. Carey. Rtracklayer: an r package for interfacing with genome browsers. *Bioinformatics*, 25(14):1841–1842, May 2009. URL: http://dx.doi.org/10.1093/bioinformatics/btp328 *(http://dx.doi.org/10.1093/bioinformatics/btp328)*, doi:10.1093/bioinformatics/btp328 *(https://doi.org/10.1093/bioinformatics/btp328)*. 5: Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, and et al. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, Nov 2019. URL: http://dx.doi.org/10.21105/joss.01686 *(http://dx.doi.org/10.21105/joss.01686)*, doi:10.21105/joss.01686 *(https://doi.org/10.21105/joss.01686)*. 6: Liguo Wang, Shengqin Wang, and Wei Li. RSeQC: quality control of RNA-seq experiments. *Bioinformatics*, 28(16):2184–2185, June 2012. URL: https://doi.org/10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*, doi:10.1093/bioinformatics/bts356 *(https://doi.org/10.1093/bioinformatics/bts356)*. 7: Simon Andrews, Felix Krueger, Anne Segonds-Pichon, Laura Biggins, Christel Krueger, and Steven Wingett. FastQC. Babraham Institute, January 2012. URL: https://qubeshub.org/resources/fastqc *(https://qubeshub.org/resources/fastqc)*. 8: Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, Jun 2016. URL: http://dx.doi.org/10.1093/bioinformatics/btw354 *(http://dx.doi.org/10.1093/bioinformatics/btw354)*, doi:10.1093/bioinformatics/btw354 *(https://doi.org/10.1093/bioinformatics/btw354)*. 9: Shihao Shen, Juw Won Park, Zhi-xiang Lu, Lan Lin, Michael D. Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. Rmats: robust and flexible detection of differential alternative splicing from replicate rna-seq data. *Proceedings of the National Academy of Sciences*, 111(51):E5593–E5601, Dec 2014. URL: http://dx.doi.org/10.1073/pnas.1419161111 *(http://dx.doi.org/10.1073/pnas.1419161111)*, doi:10.1073/pnas.1419161111 *(https://doi.org/10.1073/pnas.1419161111)*. 10: Stephen W. Hartley and James C. Mullikin. Detection and visualization of differential splicing in RNA-seq data with JunctionSeq. *Nucleic Acids Research*, pages gkw501, June 2016. URL: https://doi.org/10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*, doi:10.1093/nar/gkw501 *(https://doi.org/10.1093/nar/gkw501)*. 11: Jorge Vaquero-Garcia, Alejandro Barrera, Matthew R Gazzara, Juan

Gonzalez-Vallinas, Nicholas F Lahens, John B Hogenesch, Kristen W Lynch, and Yoseph Barash. A new view of transcriptome complexity and regulation through the lens of local splicing variations. *eLife*, February 2016. URL: https://doi.org/10.7554/elife.11752 *(https://doi.org/10.7554/elife.11752)*, doi:10.7554/elife.11752 *(https://doi.org/10.7554/elife.11752)*. 12: Yang I. Li, David A. Knowles, Jack Humphrey, Alvaro N. Barbeira, Scott P. Dickinson, Hae Kyung Im, and Jonathan K. Pritchard. Annotation-free quantification of RNA splicing using LeafCutter. *Nature Genetics*, 50(1):151–158, December 2017. URL: https://doi.org/10.1038/s41588-017-0004-9 *(https://doi.org/10.1038/s41588-017-0004-9)*, doi:10.1038/s41588-017-0004-9 *(https://doi.org/10.1038/s41588-017-0004-9)*. 13: Sam Kovaka, Aleksey V. Zimin, Geo M. Pertea, Roham Razaghi, Steven L. Salzberg, and Mihaela Pertea. Transcriptome assembly from long-read rna-seq alignments with stringtie2. *Genome Biology*, Dec 2019. URL: http://dx.doi.org/10.1186/s13059-019-1910-1 *(http://dx.doi.org/10.1186/s13059-019-1910-1)*, doi:10.1186/s13059-019-1910-1 *(https://doi.org/10.1186/s13059-019-1910-1)*. 14: Jennifer V Gerbracht, Volker Boehm, Thiago Britto-Borges, Sebastian Kallabis, Janica L Wiederstein, Simona Ciriello, Dominik U Aschemeier, Marcus Krüger, Christian K Frese, Janine Altmüller, and et al. Casc3 promotes transcriptome-wide activation of nonsense-mediated decay by the exon junction complex. *Nucleic Acids Research*, 48(15):8626–8644, Jul 2020. URL: http://dx.doi.org/10.1093/nar/gkaa564 *(http://dx.doi.org/10.1093/nar/gkaa564)*, doi:10.1093/nar/gkaa564 *(https://doi.org/10.1093/nar/gkaa564)*.

(https://github.com/dieterich-lab/Baltica/tree/master/docs/release-notes.md)

Release notes

Change log¶

Master July 23, 2021 (unreleased)¶

- Add rmats workflow
- Add scrips for parsing for rmats and updated analysis to support the method
- Create the benchmark with the ONT Nanopore-seq
- Update benchmaks, included difference comparison for SIRV benchmark
- Splite annotation and AS type assigment functions
- Update baltica table algorithm
- Add support for singularity container via snakemake, with container recipes `baltica qc config.yaml --use-singularity`
- Add parsing method for gffcompare tracking output
- Update configuration file to expose important parameters from the DJU methods
- Add end-to-end analysis with `baltica all config`
- Experiment with meta-score (gradient boosted trees)
- Add baltica report and improved on report summaries
- Add orthogonal dataset use-case, to integrate third generation sequencing to the baltica table
- Change strand parameter to "fr-firststrand": "reverse", "fr-secondstrand": "forward" or unstranded, fix error in rmats strand

1.1 September 17, 2020¶

- Add `is_novel` column, indication introns not into the reference annotation
- Remove unitended columns (X1, ...) from merge

1.0 - July 23, 2020¶

- First public release comprises of DJU methods Leafcutter, Junctionseq and Majiq. Stringtie for *de novo* transcriptomics assembly. FastQC and MultiQC (#1).