Tutorial MAJIQ/Voila (v1.0.x)

Introduction

What are *MAJIQ* and *Voila* ?

MAJIQ and Voila are two software packages that together define, quantify, and visualize local splicing variations (LSV) from RNA-Seq data. Conceptually, MAJIQ/Voila can be divided into three modules:

- **MAJIQ Builder**: Uses RNA-Seq (BAM files) and a transcriptome annotation file (GFF/GTF) to define splice graphs and known/novel Local Splice Variations (LSV).
- **MAJIQ Quantifier**: Quantifies relative abundance (PSI) of LSVs and changes in relative LSV abundance (delta PSI) between conditions with or without replicates.
- **Voila**: A visualization package that combines the output of MAJIQ Builder and MAJIQ Quantifier using interactive D3 components and HTML5. Voila creates interactive summary files with gene splice graphs, LSVs, and their quantification.

The above three modules are designed to be executed in sequence with one module's output feeding into the other. In most usage cases, the Builder will be executed only once for a given set of RNA-Seq experiments, and then the Quantifier and Voila may be executed on top of it multiple times for different analysis tasks. Note that for samples to be analyzed by the Quantifier **ALL samples analyzed must come from the same execution** of the Builder.

There are two main modes of executing the quantifier: Quantifying the relative inclusion levels of LSVs in a given experimental condition (also known as *"percent selected index"*, PSI, or Ψ), and quantifying changes of LSVs inclusion levels between two experimental conditions (aka delta PSI or $\Delta\Psi$).

Voila has two main modes of visualizing the Quantifier's results, whether these are PSI or delta PSI quantifications. The first is a (possibly long) table view of LSVs that can be filtered and ordered by different attributes (columns). The second mode is gene based, in which case each gene's splice graph and matching LSVs are grouped together. In both cases, the experimental condition can be either a single experiment or a set of replicates. In all cases the output is an interactive HTML5 that can be opened in a web browser. There is also an option to dump the output as a tab delimited text file for further analysis with other tools/scripts.

Below there is more information describing what are LSVs, how are they quantified and visualized, and what MAJIQ can (and equally important - cannot) do. You can either go through those or jump directly to the Quick Start guide.

What is an LSV?

LSV stands for "local splicing variation". Briefly, exons that are spliced together can be represented using a *splice graph* Heber et al. 2002 such as this:
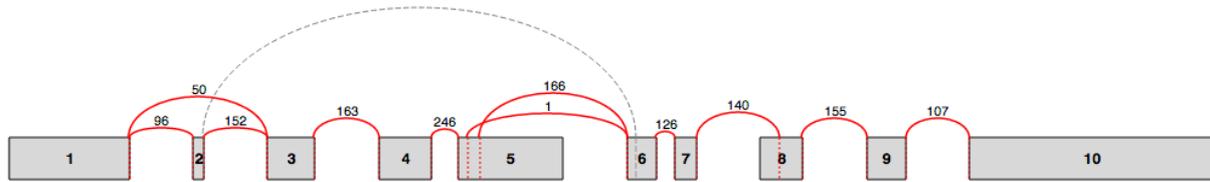
> In Voila Splice graphs, exons are represented by rectangles and junctions (or edges) by arcs. The raw number of reads spanning a junction is also displayed. For a more detailed description see splice graphs description.

LSVs involve an exon (or node in the splice graph) from which splits in the graph originate (single source LSV, or SS-LSV) or an exon into which several graph edges converge (single target LSV, or ST-LSV). An illustration of a splice graph is shown below with several SS-LSV and ST-LSV marked. The "local" aspect of LSVs definition stems from the fact they involve only a single source or single target exon. For a more formal definition please see Vaquero-Garcia et al., 2016.
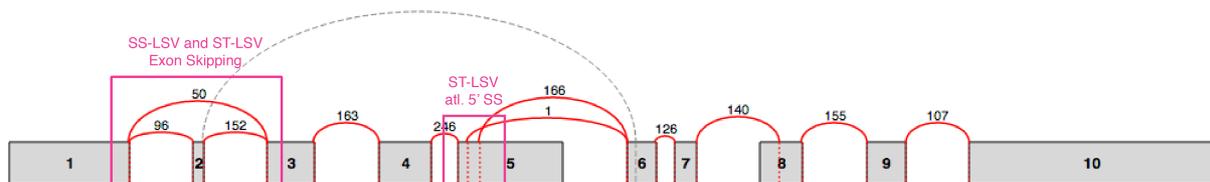
The terminology of LSVs generalizes that of alternative splicing (AS) "events" and AS events "types". The most common AS types in mammals are skipped exons, alternative 3' splice site, and alternative 5' splice site (Wang et al., *Nature* 2008). These types can all be seen as specific cases of simple or binary LSVs, i.e. LSVs that involve only two way graph splits (see figure above). However, we find that the transcriptome contains many other types of LSVs that involve different combinations of 3' and 5' splice site choices in different exons (see figures above/below, or just run your data through MAJIQ/VOILA…). Consequently, the LSV terminology helps us define and quantitate more accurately the spectrum of local splice variations observed in the transcriptome, many of which are complex.

Conceptually, LSVs are aimed to fill the gap between previously defined AS "types" described above, and full transcripts/isoforms. Ideally, we would like to identify and quantify all existing isoforms of each gene in a given RNA-Seq experiment. However, the complexity of gene isoforms combined with the shortness of current sequencing reads (typically ~100 nt long) makes isoform quantification from RNA-Seq reads a challenging problem. In contrast, LSVs can arguably still capture a lot of useful information about transcriptome variability while being deduced directly from RNA-Seq reads that span across splice junctions.

What is LSV quantification?

MAJIQ's LSV quantification is based on estimating the relative inclusion level of each junction in the LSV. For simple, binary, cases such as skipped exons, LSV quantification is equivalent to estimating the exon's percent spliced in (PSI, or Ψ). For more complex LSVs that involve three or more splice graph edges (*i.e.*, exon joining options), *MAJIQ* computes the marginal inclusion level, or PSI, per junction. Computing only these marginals allows *MAJIQ* to handle complex LSVs, keeping computational cost linear with the number of edges while still delivering estimates for the interesting biological question of "*how much is each junction used?*".

When estimating PSI for a LSV's junctions, *MAJIQ* produces a complete posterior distribution over possible PSI values. This distribution takes into account the number of reads observed at each junction, their distribution across genomic positions, GC content bias, and some possible mapper or technical artifacts. Intuitively, the deeper and smoother the coverage of an LSV, the more concentrated the PSI posterior would be (i.e. the more "sure" *MAJIQ* is about the "true" PSI value), while lower and less even coverage would result in higher variance of the PSI estimate.

Similarly, *MAJIQ*'s quantification of LSV's differential inclusion when comparing two conditions is based on estimating a posterior distribution for the change in each junction's relative inclusion level, termed delta PSI ($\Delta\Psi$). Naturally, this distribution lies in the range of -100% to +100% (or -1 to +1 when using fractions instead of percentages).

Note: For a thorough description of *MAJIQ*'s quantification algorithm for Ψ and $\Delta\Psi$ and the various parameters that control it see Vaquero-Garcia et al., 2016.

Voila's visualization of LSV Quantification uses several different techniques. In all cases, colors are used to represent the different junctions in the LSV. *Voila* uses *violin plots* to represent the posterior probability distribution for Ψ or $\Delta\Psi$. Examples of the violin plots for single Ψ are shown below.
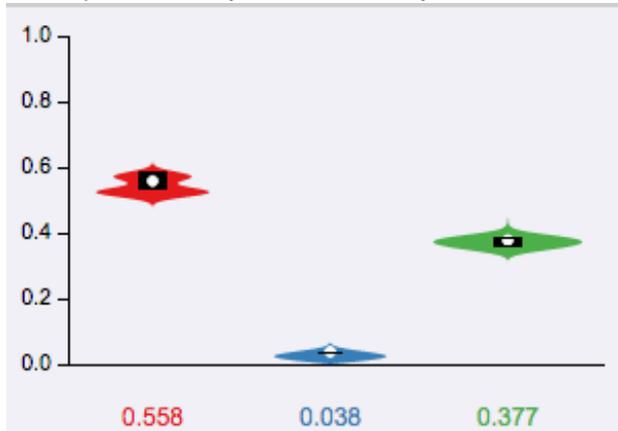
PSI

Compact view
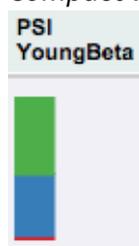
When displaying lists of LSVs *Voila* uses a compact stacked bar chart representation. The **height** of each bar represent the **expected PSI ($E[\Psi]$)** which naturally add to 100% over all

the LSV's junctions. Clicking over the bars will open a more detailed representation of the PSI distribution.

Violin plots (binary and multi-way LSV)

> Violin plots are *boxplots* plotted over the original distributions. The *box* goes from the 25th to the 75th percentile, with a white horizontal line indicating the 50th percentile (median). The tails represents the 10th and 90th percentile. Additionally, the expected PSI or $E(\Psi)$ is marked with a white circle.

Delta PSI

Compact view
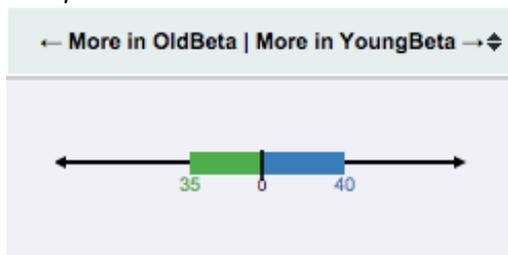
For compact visualization of $\Delta\Psi$ quantification, each colored bar represents the percentage of the expected differential inclusion for the matching edge in the splice graph. The arrows indicate the preference for one condition versus another. In this example, the blue junction is shown to have 40% more inclusion in *YoungBeta* compared with *OldBeta*. In contrast the green junction is expected to be more included in *OldBeta* by a 35% difference. Note that in this case the numbers do not generally sum to 1 or 100% as they reflect MAJIQ's expected change for each junction separately. In addition, users can use the more detailed violin plots to gain other information/statistics such as the confidence and probability distribution per junction.

What is MAJIQ?

MAJIQ is a software package that allows researchers to define and quantify both known and novel Local Splice Variations (LSVs) in genes from RNA-Seq data.

MAJIQ's main features

MAJIQ takes as input a set of RNA-Seq experiments (sorted, indexed BAM files) and previous genome annotation (GFF3 files) and produces the following:

- Splice graph for each gene based on both known transcripts annotation and de-novo junctions detected.
- All detected (known + novel) single source and single targets LSVs per gene.
- Quantification of LSVs from a given RNA-Seq experiment (w/wo replicates).

New Majiq functionalities (v1.0.x)

- **Simplifier** : The simplifier is a pre-quantification filter that allows the user to eliminate from the splice graphs junctions which are lowly utilized yet still pass MAJIQ's thresholds (de novo, reliable, quantifiable). Especially for high coverage experiments or highly expressed genes the simplifier can help users "clear out" the splicing picture, leaving the more used variants and removing those more likely to be technical or biological noise variants.
- **Weights** : This is an outlier detector for biological replicates. If you suspect your data may have outliers, it can help diagnose those. It also allows MAJIQ to assign weights to specific samples or LSVs suspected as outliers, making it more robust to outliers without sacrificing statistical power from these samples. See more details about the new underlying algorithm here: https://doi.org/10.1101/104059

What MAJIQ is not

There are many RNA-Seq analysis tasks for which *MAJIQ* was *not* designed or is currently not structured to address. Some examples include:

- Gene/isoform expression estimation: *MAJIQ* uses expression levels when it quantifies LSVs. For example, LSVs that are not present in the data will not be quantified and those with lower coverage will result in lower confidence for the relative inclusion of matching RNA segments. However, *MAJIQ* only computes *relative* inclusion of junctions in an LSV (e.g., 80% inclusion of an alternatively skipped exon). Consequently, computing the

expression of genes or isoforms and comparing those in the same experiment or between experiments is not supported.

- Relative isoform abundance: *MAJIQ* only operates at the level of local splice variations (LSVs). It does not assume the full spectrum of gene isoforms is known and does not quantify those.
- Novel gene/non coding RNA detection: *MAJIQ* requires a transcriptome annotation file (GFF3). It supplements by identifying both known and novel splice junctions within the bounds of existing loci in the provided annotation. Putative isoforms of new loci are not inferred during this process.
- Alternative transcription start/end
- Alternative polyadenylation (APA) identification/quantification.

What is Voila?

Voila is a package to interactively visualize splice variations in RNA-Seq data. It is written in Python and produces summary files in HTML5 that can be opened and interactively explored with any modern browser*. It has been conceived as the visual component of MAJIQ for analysis of Local Splice Variants (LSVs).

Voila has been tested on Google Chrome [recommended], Firefox, and Safari.

How to cite us?

Primary Publication
If you use MAJIQ in your published work, please cite this publication:
Vaquero-Garcia J, Barrera A, Gazzara MR, et al. A new view of transcriptome complexity and regulation through the lens of local splicing variations. eLife, 5, e11752.
http://doi.org/10.7554/eLife.11

Quick start

Pre MAJIQ

Select a GFF3 annotation file

The general feature format (gene-finding format, generic feature format, GFF) is a file format used for describing genes and other features of DNA, RNA, and protein sequences. The format specification for the gff version 3 can be found at GFF3 format.

In our case we use some of these features in order to define genes, transcripts and exons. An example of this format is shown below

```
chr1 protein_coding gene 107399655 107452689 . + .
Name=Serpinb7;ID=ENSMUSG00000067001;Name=ENSMUSG00000067001
chr1 protein_coding mRNA 107399655 107435399 . + .
Parent=ENSMUSG00000067001;Name=Serpinb7-002;ID=ENSMUST00000154538
chr1 protein_coding exon 107399655 107399724 . + .
Parent=ENSMUST00000154538;ID=exon:ENSMUST00000154538:1
chr1 protein_coding exon 107428231 107428416 . + .
Parent=ENSMUST00000154538;ID=exon:ENSMUST00000154538:2
chr1 protein_coding exon 107434736 107434786 . + .
Parent=ENSMUST00000154538;ID=exon:ENSMUST00000154538:3
chr1 protein_coding exon 107435327 107435399 . + .
ID=exon:ENSMUST00000154538:4;Parent=ENSMUST00000154538
chr1 protein_coding five_prime_UTR 107399655 107399724 . + .
Parent=ENSMUST00000154538;ID=five_prime_UTR:ENSMUST00000154538:1
chr1 protein_coding five_prime_UTR 107428231 107428248 . + .
ID=five_prime_UTR:ENSMUST00000154538:2;Parent=ENSMUST00000154538
chr1 protein_coding start_codon 107428249 107428251 . + 0
Parent=ENSMUST00000154538;ID=start_codon:ENSMUST00000154538:1
chr1 protein_coding CDS 107428249 107428416 . + 0
ID=CDS:ENSMUST00000154538:1;Parent=ENSMUST00000154538
chr1 protein_coding CDS 107434736 107434786 . + 0
Parent=ENSMUST00000154538;ID=CDS:ENSMUST00000154538:2
…
```

It is important to note that MAJIQ makes some assumptions when parsing the hierarchical GFF3 file and currently has some specific requirements:

- We only consider sequence features with the type (column 3) "gene"
- For every gene, we only consider isoforms of a gene with a type of "mRNA" or "transcript"
- All entries (except for genes) should have a parent attribute
- All genes should have a unique ID attribute
- Within a gene all entries should have a unique ID attribute.
- A gene can have a Name attribute, otherwise the ID will be used instead in the output.

Keeping these in mind will be important for analyzing the types of transcripts you care about and modifying your GFF3 annotation file may be necessary.

In order to obtain this format, we recommend the use of some of the most well known online DB. They provide the annotation files in some format like *GTF*, and you can transform this file to *GFF3* using a script, like this *script*

You can also download the annotation files used in Vaquero-Garcia et al., 2016 for the Ensembl hg19 or mm10 genome builds here.

Study configuration file
MAJIQ has a set of parameters needed for its execution. Several of them depend of the RNA-Seq study. This configuration file should include this information in order to be able to pass it the the MAJIQ Builder. Secondly,
it is useful to keep the info of the study ready and accessible.

This is an example of the configuration file, divided in two blocks, *info* and *experiments*:

```
[info]
readlen=76
samdir=/data/MGP/ERP000591/bam
genome= mm10
genome_path=/data/WASP_DATA/Genomes/goldenPath/mm10
type=strand-specific
[experiments]
Hippocampus=Hippocampus1,Hippocampus2
Liver=Liver1,Liver2
```

Info
This is the study global information needed for the analsysis. The mandatory fields are:

- *readlen*: Length of the RNA-seq reads. MAJIQ *can* handle experiments with multiple read lengths, just indicating the longest read length
- *samdir*: Path where the bam files are located
- *genome*: Genome assembly
- *genome_path*: Path to the genome fasta files. This path is needed for the GC content

normalization procedure.

- *type=strand-specific*: When using strand specific RNA-seq data with negative strand as reference. [Omit this parameter otherwise].

Experiments

This section defines the experiments and replicates that are to be analyzed. Each line defines a condition and its name can be customized in the following way:

```
<group name>=<experiment file1>[,<experiment file2>]
```

where the experiment file is the sorted bam filename inside the samdir directory (excluding extension *.bam*). MAJIQ expects to find within the same directory bam index files for each experiment with the format `<experiment file>.bam.bai` .

Multiple replicates within an experiment should be comma separated with no spaces.

MAJIQ Builder

MAJIQ Builder is the part of MAJIQ tool where RNA-Seq data is analyzed in order to detect LSV candidates.

All conditions and replicates that will be analyzed with MAJIQ PSI or delta PSI should be executed TOGETHER in a single Builder execution.

```
majiq build <transcript list> -conf <configuration file> --nthreads NT --output <build outdir>
```

- **Transcriptome annotation**: This is the file with the annotation database. Currently, we accept only GFF3 format. For a better description, see the annotation file section.
- **Configuration file**: This is the configuration file for the study. This file should define the files and the paths for the bam files, the read length, the genome version, and some other information needed for the builder. For a more detailed information, please check the configuration file section.
- **NT**: Number of threads to use.
- **Build outdir**: Directory where the output will be placed. MAJIQ builder has a set of output files including one *.majiq.hdf5* for each bam file and one *splicegraph.hdf5*. These files will be the input files in the next steps of the analysis.

MAJIQ Builder has several arguments in order to tweak its analysis and performance. Please check the MAJIQ parameters section for a more detailed explanation.

Outlier detection

In MAJIQ v1.0.x a new feature was introduced, weights. Sometimes RNA-Seq replicates are not as biologically similar as one might expect, and the outlier replicates can affect your PSI/DeltaPSI estimates. For that reason MAJIQ v1.0.x is able to automatically detect such outliers and correct for them. This allows MAJIQ to still extract statistical power from outliers, without letting it bias the results. This feature can be used in both PSI and deltaPSI analysis with new weights arguments that configure this feature.

A new command has been added too. In this case it will run the outlier analysis and return a global experiment weight, indicating how different each sample is from the rest of the group.

> *NOTE* To use the new weights features, an experimental group must consist of at least **3 replicates**

```
majiq weights <build outdir>/<replicate1>.majiq.hdf5 <build
outdir>/<replicate2>.majiq.hdf5 <build outdir>/<replicate3>.majiq.hdf5 --
nthreads NT --output <psi outdir> --name <cond_id>
```

This command will output each replicate sample with a corresponding float value, indicating its weight. The smaller the weight the higher chances it is considered an outlier.

Please check the MAJIQ parameters section for a more detailed explanation of all the arguments.

PSI Analysis

PSI quantification
MAJIQ PSI quantifies the LSV candidates given by the Builder. In order to improve its accuracy and reproducibility, it allows the use of biological replicates.

```
majiq psi <build outdir>/<replicate1>.majiq.hdf5 [<build
outdir>/<replicate2>.majiq.hdf5 ...] --nthreads NT --output <psi outdir>
--name <cond_id>
```

- ***.majiq.hdf5 file[s]**: the path to the *.majiq.hdf5* file(s) that were created by the MAJIQ Builder execution.
- **cond_id**: group identifier that you want to use for this execution

- **NT**: Number of threads to use.

  Please check the MAJIQ parameters section for a more detailed explanation of all the arguments.

Visualize results with VOILA

The package VOILA allows the user to generate interactive summaries to display MAJIQ computations and quantifications in the browser. All the information is also provided in TAB-delimited files that can be easily parsed for further analysis.

```
voila psi <psi outdir>/<cond_id>.psi.voila --splice-graph <build
outdir>/splicegraph.hdf5 -o <voila outdir>
```

- **<cond_id>.psi.voila** is the output file from MAJIQ PSI computation
- **splicegraph.hdf5** contains information about the genes and splice variants identified in the MAJIQ Builder

In the output directory **<voila outdir>** you will find:

- **index.html**: HTML file with a table containing all genes and LSVs identified and analyzed.
- **summaries/xx_<cond_id>.psi.html** files: interactive HTML5 summaries with MAJIQ quantifications, where **xx** is the page counter. Ten genes are displayed per page
- **<cond_id>.psi.tsv**: A tab-delimited file with all LSV information (expected PSI value, variance, exon coordinates, junction coordinates, etc.) and genomic information (chromosome, strand and coordinates).
- **static** folder: needed for the correct visualization of the `index.html` file.
- **voila.log**: log file with the execution information of Voila.

For more information see VOILA section.

For additional command line arguments use `majiq psi -h` or `voila psi -h`

Delta PSI Analysis

Delta PSI quantification

Majiq Delta PSI quantifies the differential splicing between two different groups (or conditions). Like PSI, Delta PSI is able to use replicates for each group in order to improve its accuracy and reproducibility.

```
majiq deltapsi -grp1 <build outdir>/<cond1_rep1>.majiq.hdf5 [<build
outdir>/<cond1_rep2>.majiq.hdf5 ...] -grp2 <build
```

```
outdir>/<cond2_rep1>.majiq.hdf5 [<build outdir>/<cond2_rep2>.majiq.hdf5
...] --nthreads NT --output <dpsi outdir> --names <cond1_id> <cond2_id>
```

- **-grp1 *.majiq.hdf5* file[s]**: Set of *.majiq.hdf5* file[s] for the first condition,
- **-grp2 *.majiq.hdf5* file[s]**: Set of *.majiq.hdf5* file[s] for the second condition,
- **–name cond_id1 cond_id2**: group identifiers for *grp1* and *grp2*, respectively, used for naming output files
- **NT**: Number of threads to use.

Please check the MAJIQ parameters section for a more detailed explanation of all the arguments.

Visualize results with VOILA

To visualize deltapsi quantification with Voila execute:

```
 voila deltapsi <dpsi outdir>/<cond1_id>_<cond2_id>.deltapsi.voila --
splice-graphs <build outdir>/splicegraph.hdf5 -o <voila outdir>
```

- **<dpsi_outdir>/<cond1_id>_<cond2_id>.deltapsi.voila** is the output file from delta PSI computation,
- **splicegraph.hdf5** contains information about the genes and splice variants identified in the MAJIQ Builder

In the output directory `<voila outdir>` you will find:

- **index.html**: HTML file with a table containing all genes and LSVs identified and analyzed and links to more detailed summaries.
- **summaries/xx_<cond1_id>_<cond2_id>.deltapsi.html** files: interactive HTML5 summaries with MAJIQ quantifications.
- **<cond1_id>_<cond2_id>.deltapsi.tsv**: A tab-delimited file with all the genes and LSV quantifications and genomic information.

By default VOILA uses a threshold of a change of |dPSI| >= 0.2 (20%) between conditions. To change this threshold you can use the option `--threshold` and specify a fraction from 0 to 1. To show all LSVs `--show-all` can be used instead.

For additional command line arguments use `majiq deltapsi -h` or `voila deltapsi -h`

MAJIQ

Majiq Parameters

In the quick start section above we described a general execution pipeline for MAJIQ, but those three commands have many other parameters that can be adjusted to modify MAJIQ's behavior.

Builder

```
majiq build [-h] [--nthreads NTHREADS] --output OUTDIR [--logger LOGGER]
[--silent] [--debug DEBUG] [--min_experiments MIN_EXP] -conf CONF [--
nogc]
[--pcr PCR_FILENAME] [--gff_output GFF_OUTPUT] [--min_denovo MIN_DENOVO]
[--minreads MINREADS] [--min_intronic_cov MIN_INTRONIC_COV]
[--minpos MINPOS] [--only_rna] [--non_denovo] [--only_gather] [--
permissive_ir]
[--markstacks MARKSTACKS] [--simplify all|denovo|annotated FRACTION] [--
prebam]
transcripts
```

Mandatory arguments:

- `transcripts` : Transcriptome file with the annotation database. Currently, we accept only GFF3 format. For a better description, see the annotation file section.
- `-conf CONFIG_FILE` : This is the configuration file for the study. This file should define the files and the paths for the bam files, the read length, the genome version, and some other information needed for the Builder. For a more detailed information, please check the configuration file section.
- `--output OUTDIR` : Directory where the output will be placed. MAJIQ Builder has a set of output files *.majiq.hdf5* per each bam file and one *splicegraph.hdf5*. These files will be the input files in the next steps of the analysis.

Optional arguments:

- `-h, --help` : Show help message and exit
- `--nthreads NTHREADS` : Number of threads to use.
- `--prebam` : prebam option will assume that the bam analysis was done before in the specified output folder. Causes the Builder to skip redoing this step and look for the temporary files that should be already generated.
- `--gff_output GFF_OUTPUT` : Filename where a gff with the detected LSV will be

stored. [Default: None, No file generated]
- `--simplify all|denovo|annotated FRACTION` : These are the values used for the simplification process (check simplifier). First value is the set of junction we want the simplifier to check, FRACTION is a [0..1] float indication the threshold where less used junctions will be removed.
- `--non_denovo` : Avoid *de novo* detection of junction, splice-sites, and exons. This will speedup the execution but reduce the number of LSVs detected.
- `--only_rna` : Use only junctions detected in the RNA-seq data for detecting LSVs. If an exon has only one junction with coverage, it is not going to be detected as an LSV.
- `--minreads MINREADS` : Minimum number of reads threshold combining all positions in a LSV to consider that the LSV "exists in the data". [Default: 3]
- `--minpos MINPOS` : Minimum number of start positions with at least 1 read in a LSV to consider that the LSV "exists in the data" [Default: 2]
- `--min_intronic_cov MIN_INTRONIC_COV` : Minimum number of reads on average in intronic sites, only for intron retention. [Default: 1.5]
- `--min_experiments MIN_EXP` : Used to lower the threshold for group filters. min_experiments is the minimum number of experiments where the different filter checks in order to pass an lsv or junction.
- `--min_denovo MIN_DENOVO` : Minimum number of reads threshold combining all positions in a LSV to consider that denovo junction is real". [Default: 2]
- `--nogc` : deactivate GC content normalization [Default: GC content normalization activated]
- `--markstacks MARKSTACKS` : Mark stack positions. Expects a p-value. Use a negative value in order to disable it. [Default: 1e-07]

Logger arguments:

- `--logger LOGGER_PATH` : Path for the logger. Default is output directory
- `--silent` : Boolean argument used to silence the logger.
- `--debug` : Activate this flag to activate debug messages.

PSI

```
majiq psi [-h] [--nthreads NTHREADS] --output OUTDIR --name NAME
[--logger LOGGER] [--silent] [--debug DEBUG]
[--only_bootstrap] [--k K] [--m M]
[--min_experiments MIN_EXP] [--minreads MINREADS] [--minpos MINPOS]
[--weights WEIGHTS] [--weights_alpha WEIGHTS_ALPHA]
[--weights_threshold WEIGHTS_THRESHOLD] [--weights_local LOCAL]
files [files ...]
```

Mandatory arguments:

- `files` : *.majiq.hdf5* file[s] that were created by the MAJIQ Builder execution
- `--name NAME` : The name that identifies the quantification group.
- `--output OUTDIR` : PSI output directory. It will contain the *psi.voila* file once the execution is finished.
Optional arguments:

- `-h, --help` : Show help message and exit

- `--nthreads NTHREADS` : Number of threads to use.
- `--minreads MINREADS` : Minimum number of reads to pass the quantifiable threshold combining all positions in a LSV to considered. [Default: 10]
- `--minpos MINPOS` : Minimum number of start positions with at least 1 read in a LSV to considered. [Default: 3]
- `--min_experiments MIN_EXP` : Use to alter the threshold for group filters. min_experiments is the minimum number of experiments where the different filter checks must be met in order to consider LSV or junction quantifiable.
- `--only_bootstrap`
- `--weights WEIGHTS` : Defines weights for each one of the replicates, for group1 and group2. The expected value is –weights [Auto|None|$w1[, w2, ..]$] Auto will make majiq calculate the best weights, None will use uniform weights. Select the weights manually requires specifying one weight for each replicate or an error will be triggered (check weights).
- `--weights_alpha WEIGHTS_ALPHA` : Dispersion hyperparameter (Default: 15.00)
- `--weights_threshold WEIGHTS_THRESHOLD` : Threshold hyperparameter (Default: 0.75)
- `--weights_local LOCAL` : Window for computation of local weights. If negative, uses a parametric approximation instead. (Default: 0.00)
- `--k K` : Number of positions to sample per iteration. [Default: 50]
- `--m M` : Number of bootstrapping samples. [Default: 100]

- `--trimborder TRIMBORDER` : Trim the borders when sampling (keeping the ones with reads). [Default: 5]

- `--nodiscardb` : Skip discarding the b from the NB polynomial function, since we expect our fit to start from x=0, y=0
- `--discardzeros DISCARDZEROS` : Discarding zeroes, up to a minimum of N positions per:junction. [Default: 5]

Logger arguments:

- `--logger LOGGER_PATH` : Path for the logger. Default is output directory
- `--silent` : Boolean argument used to silence the logger.
- `--debug` : Activate this flag to activate debug messages.

DeltaPSI

```
majiq deltapsi -grp1 FILES1 [FILES1 ...] -grp2 FILES2 [FILES2 ...]
--names NAMES [NAMES ...] --output OUTDIR
[-h] [--nthreads NTHREADS] [--logger LOGGER] [--silent] [--debug DEBUG]
[--min_experiments MIN_EXP] [--minpos MINPOS] [--minreads MINREADS]
[--binsize BINSIZE] [--extra_bootstrap]
[--default_prior] [--priorminreads PRIORMINREADS]
[--priorminnonzero PRIORMINNONZERO] [--iter ITER] [--breakiter BREAKITER]
[--k K] [--m M] [--trimborder TRIMBORDER] [--nodiscardb]
[--discardzeros DISCARDZEROS] [--weights WEIGHTS WEIGHTS]
[--weights_alpha WEIGHTS_ALPHA] [--weights_threshold WEIGHTS_THRESHOLD]
[--weights_local LOCAL]
```

Mandatory arguments:

- `-grp1 FILES1 [FILES1 ...]` : Set of *.majiq.hdf5* file[s] for the first condition
- `-grp2 FILES2 [FILES2 ...]` : Set of *.majiq.hdf5* file[s] for the second condition
- `--names NAMES [NAMES ...]` : *cond_id1 cond_id2*: group identifiers for grp1 and grp2 respectively.
- `--output OUTDIR` : PSI output directory. It will contain the *deltapsi.voila* file once the execution is finished.

Optional arguments:

- `-h, --help` : Show help message and exit
- `--nthreads NTHREADS` : Number of threads to use [Default: 4].
- `--minreads MINREADS` : Minimum number of reads to pass the quantifiable threshold combining all positions in a LSV to considered. [Default: 10]
- `--minpos MINPOS` : Minimum number of start positions with at least 1 read in a LSV to considered. [Default: 3]
- `--min_experiments MIN_EXP` : Use to alter the threshold for group filters. min_experiments is the minimum number of experiments where the different filter checks must be met in order to consider LSV or junction quantifiable.
- `--binsize BINSIZE` : The bins for PSI values. With a BINSIZE of 0.025 (default), we have 40 bins
- `--default_prior` : Use a default prior instead of computing it using the empirical data
- `--priorminreads PRIORMINREADS` : Minimum number of reads combining all positions in a junction to be considered (for the 'best set' calculation). [Default: 20]
- `--priorminnonzero PRIORMINNONZERO` : Minimum number of positions for the best set.
- `--iter ITER` : Max number of iterations of the EM

- `--breakiter BREAKITER` : If the log likelihood increases less than this flag, do not do another EM step
- `--extra_bootstrap`
- `--weights WEIGHTS` : Defines weights for each one of the replicates, for group1 and group2. The expected value is –weights [Auto|None|$w1[, w2, ..]$] Auto will make MAJIQ calculate the best weights, None will use uniform weights. Select the weights manually requires specifying one weight for each replicate or an error will be triggered (check weights).

  - `--weights_alpha WEIGHTS_ALPHA` : Dispersion hyperparameter (Default: 15.00)
  - `--weights_threshold WEIGHTS_THRESHOLD` : Threshold hyperparameter (Default: 0.75)
  - `--weights_local LOCAL` : Window for computation of local weights. If negative, uses a parametric approximation instead. (Default: 0.00)
  - `--k K` : Number of positions to sample per iteration. [Default: 50]
  - `--m M` : Number of bootstrapping samples. [Default: 100]
  - `--trimborder TRIMBORDER` : Trim the borders when sampling (keeping the ones with reads). [Default: 5]
  - `--nodiscardb` : Skip discarding the b from the NB polynomial function, since we expect our fit to start from x=0, y=0
  - `--discardzeros DISCARDZEROS` : Discarding zeroes, up to a minimum of N positions per:junction. [Default: 5]

Logger arguments:

- `--logger LOGGER_PATH` : Path for the logger. Default is output directory
- `--silent` : Boolean argument used to silence the logger.
- `--debug` : Activate this flag to activate debug messages.

Weights

```
majiq weights [-h] [--nthreads NTHREADS] --output OUTDIR
[--logger LOGGER] [--silent]
[--debug DEBUG] [--min_experiments MIN_EXP] [--k K]
[--m M] [--minreads MINREADS] [--minpos MINPOS]
[--trimborder TRIMBORDER] [--nodiscardb]
[--discardzeros DISCARDZEROS]
[--weights_alpha WEIGHTS_ALPHA]
```

```
[--weights_threshold WEIGHTS_THRESHOLD]
[--weights_local LOCAL] --name NAME
files [files ...]
```

Mandatory arguments:

- `files` : *.majiq.hdf5* file[s] that were created by the MAJIQ Builder execution
- `--name NAME` : The name that identifies the quantification group.
- `--output OUTDIR` : The output directory where the log files will be written.
  Optional arguments:

- `-h, --help` : Show help message and exit

- `--nthreads NTHREADS` : Number of threads to use.
- `--minreads MINREADS` : Minimum number of reads threshold combining all positions in a LSV to considered. [Default: 10]
- `--minpos MINPOS` : Minimum number of start positions with at least 1 read in a LSV to considered. [Default: 3]
- `--min_experiments MIN_EXP` : Lower threshold for group filters. min_experiments is the minimum number of experiments where the different filters check in order to pass an lsv or junction.
- `--weights WEIGHTS` : Defines weights for each one of the replicas, for group1 and group2. The expected value is –weights [Auto|None|$w1[, w2, ..]$] Auto will make majiq calculate the best weights, None will use uniform weights. Select the weights manually requires specifying one weight for each replica or an error will be triggered (check weights).
- `--weights_alpha WEIGHTS_ALPHA` : Dispersion hyperparameter (Default: 15.00)
- `--weights_threshold WEIGHTS_THRESHOLD` : Threshold hyperparameter (Default: 0.75)
- `--weights_local LOCAL` : Window for computation of local weights. If negative, uses a parametric approximation instead. (Default: 0.00)
- `--k K` : Number of positions to sample per iteration. [Default: 50]
- `--m M` : Number of bootstrapping samples. [Default: 100]
- `--trimborder TRIMBORDER` : Trim the borders when sampling (keeping the ones with reads). [Default: 5]
- `--nodiscardb` : Skip biscarding the b from the NB polynomial function, since we expect our fit to start from x=0, y=0
- `--discardzeros DISCARDZEROS` : Discarding zeroes, up to a minimum of N positions per:junction. [Default: 5]
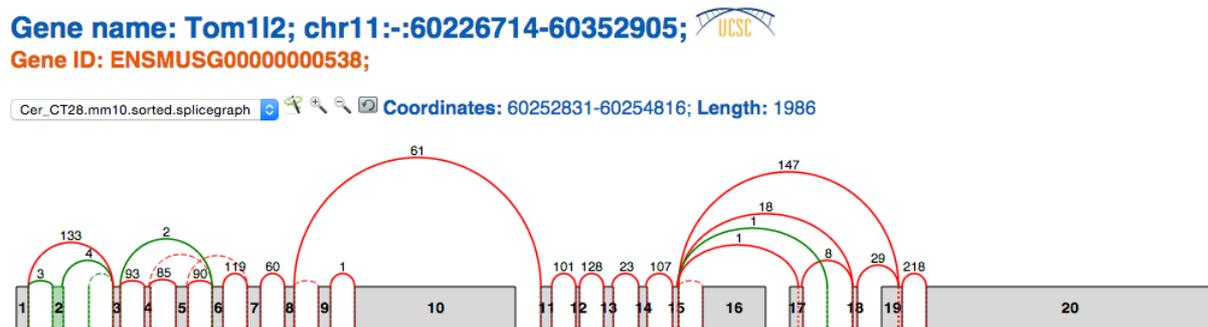
Logger arguments:

- `--logger LOGGER_PATH` : Path for the logger. Default is output directory
- `--silent` : Boolean argument used to silence the logger.

- `--debug` : Activate this flag to activate debug messages.


VOILA

HTML5 Summaries

Splice Graphs

The splice graph gadget included in VOILA summarizes all the splice variants found in a gene by MAJIQ. Splice Graphs include the following features:

- Easy differentiation between exons and junctions annotated (red) and *de novo* detected (green) in RNA-Seq data,
- Contextual information about the coordinates for each exon and intron (hovering over exons/junctions).
- Raw reads counts for each junction. (*Note* MAJIQ applies several normalization factors to these raw values)
- Scaled view of the gene and the splice graph 🪄. By default, introns are trimmed to obtain a more compact representation of the splice graph. Switching between scaled and default view is accomplished by clicking on the wand.
- Zoom in/out to explore complex splice graphs.
- Possibility to switch between replicates (condition members) when more than one splice graph is available via the dropdown box.

All *Gene Summaries* include a descriptive legend of what you might find in the Splice Graphs:
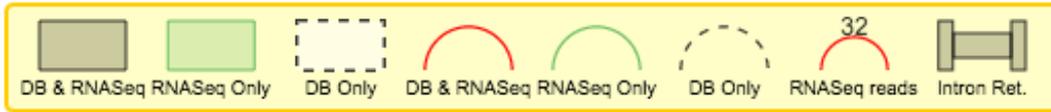
DB refers to exons and junctions annotated in the GFF3 file and *RNASeq* to exons and junctions found in RNA-Seq data. *RNASeq reads* alludes to the raw reads found in RNA-Seq data. Please, note that retained introns (narrow rectangles connecting two exons) do not appear currently in the legend.

Note that when MAJIQ creates the splice graphs that VOILA visualizes, it considers the bounds of each individual exon in all transcripts containing that exon and the longest version is represented in the splice graph. So in the above example, the annotation database had longer versions of exons 17 and 19 corresponding to alternative transcription start sites for this gene and thus the starting positions are extended to reflect this.

PSI Summary

VOILA PSI Index file offers an overview of all the LSVs detected in a table, providing links to detailed summaries of LSVs and genes. Clicking over a gene or LSV ID opens up a new tab with a summary containing interactive splice graphs, distributions of PSIs per junction and links to the UCSC. Below is an example of PSI summaries for *Tpd52l1*:
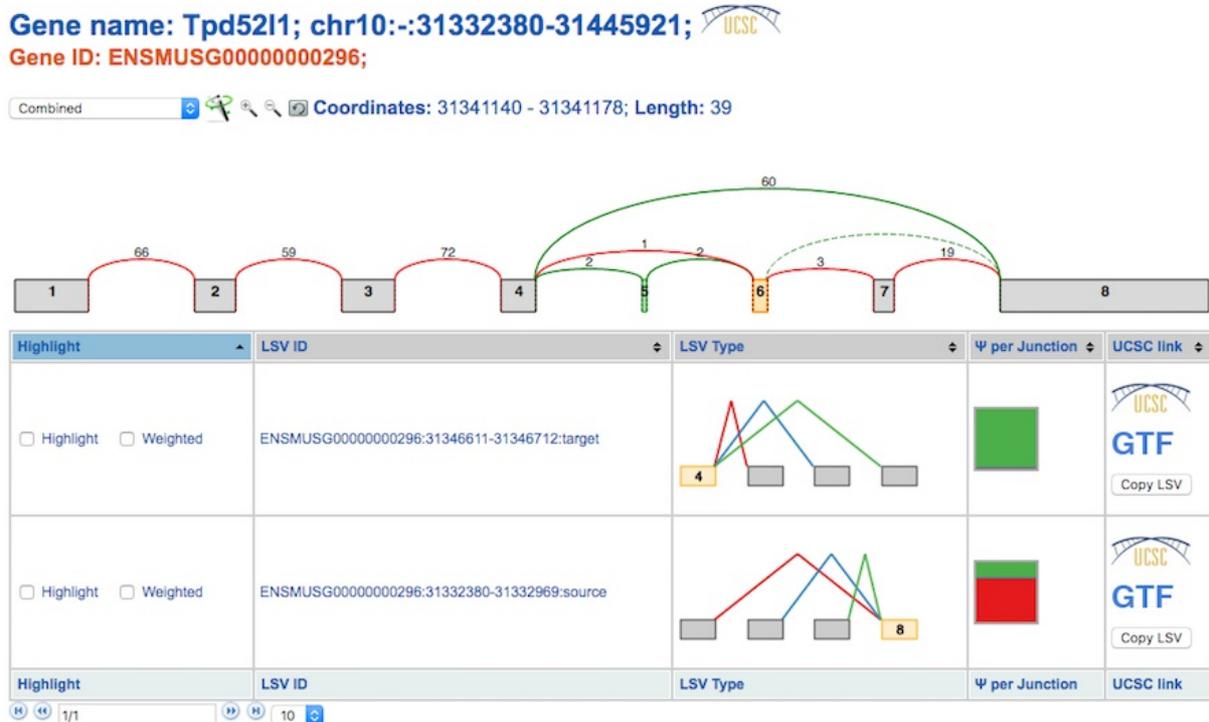
> **Tip**: PSI Summaries can be navigated through the *Previous* and *Next* links, without having to go back to the *index* file.

The information is broken into genes (10 per page), each of them with an interactive splice graph and an associated table with LSV quantification data. The table has the following information about the LSV:

- Hightlight:
    - **Hightlight**. To move the visualization of the LSV to the splice graph, select the "highlight" checkbox in the LSV table. All junctions will be hidden except the ones included in the LSV. The visible junctions will be colored to match the highlighted LSV.
    - **Weighted**. To weight the junctions of the highlighted LSV, check the "weighted" checkbox. The junctions for this LSV will now be weighted proportionally to the LSV's expected PSI value for this experiment.

- LSV ID: a unique identifier for the LSV.
- LSV type: a thumbnail representing the splicing event. Each junction has a different color.
- PSI per junction: the expected Percent Selected Index (PSI) per junction. It has two *views*

that can be switched between by clicking on them:

- ○ ***Compact view***. Initially, the PSI per junction is represented by the height of a colored box. Each color refers to the expected inclusion of a particular junction. The taller the rectangle, the more included the junction is expected to be. Clicking on the rectangle (zoom in pointer) will open the *expanded view*.
- ○ ***Expanded view***. Distributions of probabilities of PSI per LSV junction (violin boxplots). The *white* dot represents the expected PSI, whereas the box plot indicates the 10, 25, 50, 75 and 90 percentile of the distribution. Consistently with the Compact view, each color refers to a particular junctions of the LSV.

- • LSV links:
  - ○ **GTF**. Link to the GTF file associated with the LSV.
  - ○ **UCSC**. Coordinates link to explore the LSV on UCSC Genome Browser (when available).
  - ○ **Copy LSV**. The new VOILA supports copy-paste functionality into a new web-tool, MAJIQ-SPEL. SPEL allows users to connect the LSV quantification to the various gene isoforms associated with those, automatically design RT-PCR primers for validating the LSV (with optional control over primer design parameters), and map the LSV to protein domains on the genome browser for functional integration and downstream analysis. See more details: http://biorxiv.org/content/early/2017/05/09/136077

Lastly, the panel *LSV filters* allows the user to screen out LSVs with certain properties like having alternative 5-prime splice sites, involve Exon Skipping or contain a certain amount of exons and junctions.

Delta PSI Summary

#	Gene	LSV ID	LSV Type	← More in Cer | More in Adr →	Coordinates Links
1	Rapgef1	ENSMUSG00000039844:29722224-29722309:target			UCSC
2	Clip1	ENSMUSG00000049550:123630225-123631169:target			UCSC
3	Dlgap4	ENSMUSG00000061689:156745835-156746196:target			UCSC
4	Fubp3	ENSMUSG00000026843:31598538-31598700:target			UCSC
5	Clta	ENSMUSG00000028478:44025449-44025560:source			UCSC
#	Gene	LSV ID	LSV Type	← More in Cer | More in Adr →	Coordinates Links
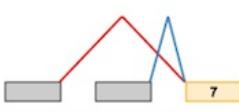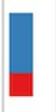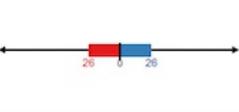
1/1   All

Delta PSI index file is identical to PSI index except for the *PSI per junction* column that represents the estimated differential inclusion levels as bars leaning towards condition1 or condition2 (see *Compact view* below). Clicking over LSV IDs and genes opens up individual summaries.

The results are broken into 10-genes per page summaries, with LSV quantifications grouped by gene. Unlike in single PSI summaries, there are 2 splice graphs (one per condition) which facilitate quick visual inspection of possible differences. If multiple replicates were used in a condition, the splice graphs for each replicate can be switched between through the drop down box located next to the wand. The LSV information is displayed as follows:

- Hightlight:
    - **Hightlight**. To move the visualization of the LSV to the splice graph, select the "highlight" checkbox in the LSV table. All junctions will be hidden except the ones included in the LSV. The visible junctions will be colored to match the highlighted

LSV.

- **Weighted**. To weight the junctions of the highlighted LSV, check the "weighted" checkbox. The junctions for this LSV will now be weighted proportionally to the LSV's expected PSI value for this experiment.

- LSV ID: a unique identifier for the LSV.
- LSV type: a thumbnail representing the splicing event. Each junction has a different color.
- PSI *condition[1|2]*: individual Ψ in *condition[1|2]*, similar to the Ψ *per junction* column in PSI summary, with compact and expanded views.
- More in *condition 1* | More in *condition 2*: the expected delta PSI for all junctions. It has two *views* which can be switched between by clicking on them:
  - **Compact view**. Initially, the observed delta PSI preference for a certain condition (more included) per junction. A bar going towards *condition 1* (left) with a value of *26* means that the junction is 26% more included in condition 1 than in condition 2. Each color refers to a particular junction of the LSV. In the above example, the red junction is 26% more included in cerebellum compared to adrenal gland.
  - **Expanded view**. Distributions of probabilities of Delta PSI per LSV junction (violin boxplots). The *white* dot represents the expected PSI, whereas the box plot indicates the 10, 25, 50, 75 and 90 percentile of the distribution. Consistent with the Compact view, each color refers to a particular junction of the LSV.

- LSV links:
  - **GTF**. Link to the GTF file associated with the LSV.
  - **UCSC**. Coordinates link to explore the LSV on UCSC Genome Browser (when available).
  - **Copy LSV**. The new VOILA supports copy-paste functionality into a new web-tool, MAJIQ-SPEL. SPEL allows users to connect the LSV quantification to the various gene isoforms associated with those, automatically design RT-PCR primers for validating the LSV (with optional control over primer design parameters), and map the LSV to protein domains on the genome browser for functional integration and downstream analysis. See more details: http://biorxiv.org/content/early/2017/05/09/136077

Tab-delimited file

VOILA provides a tab-delimited text file to allow users to parse MAJIQ results and further analyze particular LSVs or genes of interest. Most fields are shared between single PSI and delta PSI computations for the expected values and the confidence measures (variance in the case of

single PSI and the probability of |delta psi| > 0.2 (or your specified alternative threshold) in delta PSI analysis). The common fields are: Gene name; LSV ID; LSV Type; LSV attributes (A5SS, A3SS, ES, Num. Junctions and Num. Exons); chromosome; strand; LSV coordinates (junctions and exons coordinates); and finally, if additional evidence is required to determine what is the start/end of an LSV, a list with all possible alternative starts and ends is provided.

FAQ

In VOILA gene summaries, what is that *number at the beginning* of the HTML file?

To achieve a better performance, VOILA creates HTML files of up to 10 genes. Therefore, if MAJIQ detected and quantified LSVs from N genes, there will be N/10 pages (always rounded to the upper integer limit).
For example, let say that we executed `voila psi data/Liver1.majiq_psi.pickle --genes-exp1 data/Liver1.splicegraph -o psi_gene_out/` and 182 genes were detected. There will be 182/10=19 pages (starting with 0):
0_Liver1.majiq_psi_lsv_single_gene.html, 1_Liver1.majiq_psi_lsv_single_gene.html, …, 18_Liver1.majiq_psi_lsv_single_gene.html.