ASPeak

Version 2.0

A Peak Calling Tool for CLIP- / RIP- Seq Data

User Manual

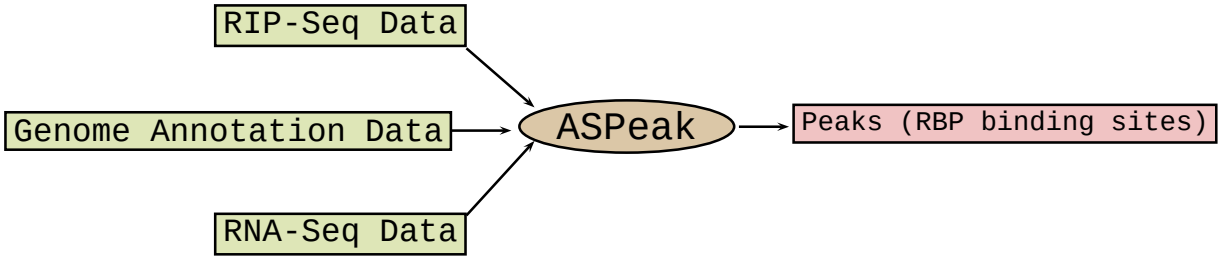
Contents

What is ASPeak?

ASPeak is a free command-line software tool for finding clusters (peaks) in CLIP- / RIP- Seq data. It is implemented in Perl.

ASPeak uses an abundance sensitive peak detection algorithm to detect clusters (peaks) of RNA-binding protein (RBP) sites in CLIP- / RIP- Seq data at nucleotide resolution.

ASPeak takes CLIP- / RIP- Seq, RNA-Seq and genome annotation data as input and it outputs peaks, namely RBP binding sites. Also, when control data is provided, ASPeak outputs the false discovery rates of the found peaks.



We previously applied this algorithm to detect genome-wide binding sites of the multi-protein Exon Junction Complex (EJC). In that study, we analyzed eIF4AIII-Y14 RIPiT sequencing libraries. RIPiT is an extension of the RIP-Seq approach posing identical bioinformatics challenges. Briefly, instead of using a single antibody against a protein of interest, a tandem immunoprecipitation approach is used. In the case of EJC, there were 4 known core factors of the complex. The first and second immunoprecipitations target two of these core factors ensuring a more stringent pull down of the protein complex of interest. ASPeak applied to this dataset facilitated discovery of a role for the EJC in mRNA compaction via extensive interactions with SR and other RNA-binding proteins (Singh et al., 2012).

Requirements

ASPeak is designed to run on Unix clones. It has been tested on Linux (Mint, Fedora, Ubuntu and CentOS) and OS X. Users of other operating systems can install VirtualBox, freely available at https://www.virtualbox.org, and install and run a Linux distribution virtually.

Perl comes with many Unix clones out of the box. ASPeak can run on Perl 5.8.8 or any later version. To learn the Perl version installed in your system, in the terminal run the command

```
$> perl -v
```

Throughout this document, '$>' denotes the command prompt of the terminal. It is not meant to be typed in the command line.

The only non-standard perl module required to run ASPeak is Math::CDF. This module can be installed via CPAN

```
$> cpan -i Math::CDF
```

You can verify the installation of Math::CDF by

```
$> perl -e "require Math::CDF"
```

The module is properly installed if there are no error messages.

ASPeak requires wigToBigWig for file conversion from bedGraph to BigWig. You can obtain it from the appropriate directory at http://hgdownload.cse.ucsc.edu/admin/exe/

If your input is in BAM, SAM or BOWTIE format, you need SAMTools (available at http://samtools.sourceforge.net/) and BEDTools (available at http://code.google.com/p/bedtools/) installed in your system.

If you have root access in your system, simply copy the executables of these programs to a standard bin directory such as /usr/bin. If you don't have root access, you need to define the paths of these programs. As an example solution, create a directory called bin in your home directory.

```
$> mkdir ~/bin
```

Copy the executables of these programs to ~/bin. Now, add the directory ~/bin to your PATH. For this, you need to modify your bash startup file. Type

```
$> ls ~/ -a
```

and look for a file named .bashrc or .bash_login or .bash_profile. Then append the line "export PATH=$PATH:~/bin" to your bash startup file. For example, if your bash startup file is .bashrc, then

```
$> echo export PATH=$PATH:~/bin >> ~/.bashrc
```

To make the changes take effect, make bash read the startup file

```
$> source ~/.bashrc
```

If, for example, you put bedtools executable in ~/bin, then you should see the version of bedtools by

```
$> bedtools --version
```

Getting ASPeak

The latest version of ASPeak is available at https://sourceforge.net/projects/as-peak/

Configuring ASPeak

First, review the Section Requirements and make sure that you have the necessary software and cpan package installed in your system. Move the tar.gz file to an appropriate directory. Extract it

```
$> tar -zxvf ASPeak_v2_0_0.tar.gz
```

ASPeak is ready to run. The main script to execute is ASPeak.pl which is under the scripts directory.

Running ASPeak

This section explains how to prepare your system and how to run ASPeak and its steps. You can find a walk-through of ASPeak in the section A Case Study.

You can make sure that ASPeak is working properly by testing it on the sample data that comes with ASPeak. It has RIP-Seq, RNA-Seq and Control libraries in bowtie format. It has annotation data for the regions UTR3 and introns in BED format. The sample data is in the folder called example. To run ASPeak with this data set, change the directory to the main directory of ASPeak and type

```
$> ./scripts/ASPeak.pl -lib example/RIPSeq/sampleRIPSeq.bowtie \
-beddir example/annotation -rnaseq example/RNASeq/sampleRNASeq.bowtie \
-control example/control/sampleControl.bowtie -outdir exampleOut
```

Upon successful execution, ASPeak displays the output folders and steps executed. You can see the found peaks by

```
$> more exampleOut/foundpeaks/sampleRIPSeq/peaks/sampleRIPSeq.peaks
```

Input Preparation

Before running ASPeak with your own data, prepare your CLIP- / RIP- Seq, RNA-Seq, Control (if you there is a control library) and genome annotation data files according to the following file formats. More information on file formats can be found in the section File Formats. ASPeak accepts BAM, SAM, BOWTIE (or BOWOUT) and BED file formats for CLIP- / RIP- Seq and RNA-Seq data. Note that ASPeak recognizes the formats of the input files by their extension. So BAM files should have the extension .bam, SAM files .sam, BOWTIE files .bowtie (or .bowout) and BED files .bed

Annotation data must be provided in BED format. ASPeak calls peaks for each region of the genome separately. *Regions* (for example introns, exons, UTR3 and etc) are defined by the user in BED files. For each region, create a BED file. Each row of the BED file is a contiguous part of the genome. We call these parts *intervals*. So, for example, if we define exons in a BED file, say exons.bed, then each row of exons.bed corresponds to an individual exon. Create your genome annotation files in BED format and make them have .bed extension. Put all annotation files in one directory. For each region, that is for each BED file in this directory, peaks will be called separately. Consequently, those peaks, i.e., protein binding sites, which are not in any of the regions given to ASPeak, will not be reported. For detailed information on genome annotation, regions and intervals, see the subsection Genome Annotation: Regions & Intervals. In the subsection Example: Creating a Genome annotation file, we describe how to create a bed file for the region UTR3' as an example.

Note that in most cases RNA-Seq measures the abundance, i.e., expression levels, of intervals. The user is free to choose between RIP-input or any other appropriate RNA-seq data to estimate the expression level. For simplicity, we used RNA-seq as our measure of expression levels. The user can provide the expression level data (whether it comes from RNA-Seq or RIP-Seq) in the -rnaseq argument.

Checklist before running ASPeak	
Genome Annotation Data	Have your annotation data in BED format ready as described above. Each BED file defines a region. Put all your annotation files in one directory.
CLIP- / RIP- Seq Data	Have your data ready in BAM, SAM, BOWTIE or BED format.
RNA-Seq Data	Have your data ready in BAM, SAM, BOWTIE or BED format.
Is there a control library?	If you have a control library, have your data ready in BAM, SAM, BOWTIE or BED format. This input is optional.

Command Line Arguments

The executable scripts of ASPeak are in the scripts sub-folder. Change the current directory to the main directory. The main script is ASPeak.pl which is in the scripts folder. All other scripts are executed by this script. For help, i.e., to view the plain old documentation, type

```
$> ./scripts/ASPeak.pl -help
```

The essential parameters needed to run ASPeak are as follows.

Essential Parameters	
-beddir ⟨ **BED files directory** ⟩	Genome annotation files
-lib ⟨ **lib1path:**$\cdots$**:libNpath** ⟩	CLIP- / RIP- Seq library paths separated by colons.
-control ⟨ **control Library Path** ⟩	Control library path. Using a control library is optional.
-rnaseq ⟨ **RNA-Seq library path** ⟩	RNA-Seq library path.

So, ASPeak can be used with the default parameters, with one RIP-Seq library and without the control library, by

```
$> ./scripts/ASPeak.pl -bed path -lib path/filename -RNASeq path/filename
```

If you have a control library, you can use it by specifying the -control parameter

```
$> ./scripts/ASPeak.pl --bed path -lib path/filename \
-RNASeq path/filename -control path/filename
```

You can override the default parameters in two ways: using a parameter file or command line arguments. ASPeak comes with a default parameter file that it gets the system parameters from. The name of the file is `default.txt` and it is in `scripts` directory. You can make a copy of this file and modify it according to your needs. Then you can run ASPeak using the new parameters by providing the new parameter file with the command line argument `-param path/filename`.

```
$> ./scripts/ASPeak.pl -bed path -lib path/filename \
-RNASeq path/filename -param path/filename
```

Detailed information on the parameter file can be found in the subsection Parameter File.

Another way of overriding the default parameters is specifying them in the command line arguments. All command line arguments of ASPeak are given in Table 1.

Table 1: Complete command line parameters of ASPeak

Main Parameters		
-beddir <path>	:	The directory of the genome annotation files. All annotation files must be in BED format and they must have .bed extension.
-outdir <output directory>	:	ASPeak will create this directory, if it doesn't exist, and put the output and LOG files in it.
-lib <lib1file:lib2file:...>	:	CLIP- / RIP- Seq Library file(s). Accepted formats are BAM, SAM, BOWTIE and BED.
-rnaseq <file1:file2:...>	:	RNA-Seq library file(s). Accepted formats are BAM, SAM, BOWTIE and BED.
Optional Parameters		
-param <parameter file>	:	Provides system parameters. System parameters are overridden by command line arguments (see the rest of this table) if there are any.
-type <run type>	:	See the table below to see all the run types. If no type is given, ASPeak will execute the steps specified in the parameter file.
-region <region name>	:	Specify region name
-nornaseq	:	Set this parameter if you don't have RNA-Seq data. (Not recommended.)
-libname <name1:name2:...>	:	Library name(s) (separated by colons).
-libformat <sam|bam|bed|bowtie>	:	Specify the input format of the RIP-Seq libraries. If not given, format is recognized by the file extension.
-control <lib1file:lib2file:...>	:	Control library file(s). Accepted formats are BAM, SAM, BOWTIE and BED.
-controlname <name1:name2:...>	:	Control library names separated by colons
-controlformat <sam|bam|bed|bowtie>	:	Specify the input format of the control libraries. If not given, format is recognized by the file extension.
-rnaseqname <name1:name2:...>	:	RNA-Seq library name(s)
-rnaseqformat <sam|bam|bed|bowtie>	:	Specify the input format of the RNA-Seq libraries. If not given, format is recognized by the file extension.
-s	:	If given, peak calling is done for each strand separately.
-estimationradius	:	The number of intervals above and below the current interval in determining the parameters of the NB distribution. Its default value is 500.
-gapnumber	:	The maximum number of tolerated consecutive positions having relatively low reads (gaps) in the peaks. Its default value is 2.
-cluster	:	When given, jobs will be submitted to the nodes of Sun Grid Engine (SGE). You should have access to a SGE to use this feature.
-node <max node number>	:	The maximum number of processors/cores/nodes that ASPeak can run on. Use this when -cluster is set.
-queue <cluster node>	:	Specify the cluster node to submit the jobs to. If it is set to ANY, jobs will be submitted to any available cluster node. Use this when -cluster is set
-jobnum <job number>	:	When running on cluster, each submission of a particular script is called a job. This sets the initial job number that increments with each submission.
-help	:	Display plain old documentation.
-version	:	Print the program version.

Running ASPeak Manually

If you run ASPeak as above, in other words without specifying the -type parameter, the system will automatically execute every step as given in the parameter file, having a nonzero flag, in the given order. This type of execution is recommended for most users. Advanced users may also run a particular step by providing the -type parameter in the command line arguments. Note that the order of these steps is important as the output of a former step is usually input of following steps. You can execute a particular step with the default parameters by

```
$> ./ASPeak.pl -bed path-lib path/filename -RNASeq path/filename \
-type StepName
```

Execution types are are given in Table 2. If no type is specified, ASPeak automatically executes those steps in the parameter file having value 1. By default all steps are executed which is recommended for most users.

Table 2: Run Types of ASPeak

-type	**action**
StepCheckParams	Check the system parameters and input for consistency.
StepGetChroms	Get the names of chromosomes from the input data.
StepConvert2Bed	Convert all accepted input file formats to the BED format.
StepCountMappedReads	Get the number of mapped reads for both the positive and the negative strand.
StepRegionSeparation	Partition input into regions (e.g. utr3, utr5, introns, exons and etc.)
StepChromSeparation	Partition bed files into chromosomes for parallelization and low memory consumption purposes.
StepPrepInput	For each input file, determine the number of reads and RPKM values of each interval and prepare bedGraph files using centers.
StepMakeRPKM	Find the RPKM values of each interval using the RNA-Seq data. If there are two or more RNA-Seq libraries, combine them.
StepNBParameters	Estimate the parameters of the Negative binomial distribution for each interval.
StepPeak	Finds the peaks in the RIP-Seq data.
StepCombinePeaks	Combine the found peaks for each region into one.
StepCalcFDR	If control data is provided, calculate the False Discovery Rate (FDR) of the peaks.
StepPeaks2Bed	Convert the output of StepPeak, i.e., found peaks, to BED file format.
StepMakeBedGraph	Convert the found peaks to bedGraph for visualization.
StepWigToBigWig	Convert the bedGraph file of the found peaks to BigWig format.
StepSubmitJobs	Execute the above steps in the given order if their flag is 1 in the parameter file.

Output Files

Output files are put in the directory called `foundpeaks` under the output directory specified by the user. For each library, there will be a subdirectory of `foundpeaks` having the name of the library. Found peaks are reported in four different ways.

PEAK FILE: This file is in the directory `peaks` and it has the extension `.peaks`. Its format is explained in the subsection Output File (Peaks). The most detailed information about the found peaks, such as chromosome, strand, interval, p-Value, position, RPKM value and etc., is given in this file. Each line of this file corresponds to one peak. A peak file having the name of the corresponding RIP-Seq library is created. This file contains all peak information. Besides, peak files having the name of the defined regions exist in the same directory. They have only the peaks that are in the corresponding region.

BED FILE: All input formats of RIP-Seq and Control are internally converted into `BED` file format by `ASPeak` before calling the peaks. The rows of those files that cover the found peaks are reported in this BED file.

bedGraph FILE: Peaks are plotted in `bedGraph` file format so that they can be visualized in a genome browser. Input data (whether it be in `BAM`, `SAM`, `BED` or `BOWTIE`) of RIP-Seq, RNA-Seq and Control libraries consist of fragments of nucleotide sequences. Those fragments that cover the found peaks are converted into bedGraph format. More precisely, all input formats of RIP-Seq and Control are internally converted into `BED` file format by `ASPeak` in the preprocessing step. Only the fragments in `BED` file that cover the peaks are converted into `bedGraph`.

SPAN FILE: In the peak file, peaks are reported with a start and end position. We call this the *span* of the peak. Span of the peaks are reported in BED format in the span file. Thus, for each peak, there will be one row having the same start and end position with the corresponding peak.

Running `ASPeak` on a High Performance Computing Center

Though `ASPeak` runs standalone by default, parallelization on High Performance Computing Centers (HPC) is offered as an experimental feature. In particular, `ASPeak` has been tested on a Sun Grid Engine (SGE) and related job submission procedures have been implemented in `ASPeak` in the script `submitJobs.pl`.

In order to run `ASPeak` on a SGE, check your system documentation and make sure that SGE is available. Check that you have the submission program `qsub` and cluster status monitoring program `qstat` available.

```
$> qsub -help
```

```
$> qstat -help
```

If your system has a different architecture, please see your system documentation. Then according to your system, you can modify the script `submitJobs.pl` in the `scripts` directory and the function `StepSubmitJobs` in `functions.pm` which is in `lib` directory.

In the command line arguments, set the parameter -cluster to enable parallel running. You can specify the cluster node with the -queue option. You can set the maximum number of nodes to be used via the -node option. For example, let's say you want to run ASPeak using the example data that comes with it, on a SGE. You want to submit the jobs to the cluster node called chassis01.q. You want to use at most 20 nodes once at a time. Then, you can run ASPeak by

```
$> ./scripts/ASPeak.pl -lib example/RIPSeq/sampleRIPSeq.bowtie \
-beddir example/annotation -rnaseq example/RNASeq/sampleRNASeq.bowtie \
-outdir exampleOut -cluster -queue chassis01.q -node 20
```

When running in parallel mode, ASPeak can process each chromosome file, for each region, independently by submitting jobs to the cluster node given in the parameter file. Especially if you have a large data set, it is recommended to run ASPeak on a HPC running on SGE and take advantage of parallel processing when possible.

An important note on using ASPeak without RNA-Seq data

ASPeak can still be used in the absence of RNA-Seq data but **it is highly recommended that RNA-Seq data is provided for obtaining much more reliable results.**

The main advantage of the peak calling algorithm used in ASPeak is that it takes the initial abundance of the intervals into account in determining the background noise. Without RNA-Seq data, one doesn't have this advantage which is crucial for calling the peaks. RIP-Seq and CLIP-Seq experiments are based on an enrichment strategy. However, in these experiments, there is some noise that arises from experimental methods used such as ligation bias, sequencing quality, mapping and etc. This noise is mostly correlated to the expression levels of the target RNA sites. For example, say there are two intervals A and B where B is expressed 1000 times more than A. Suppose that, in reality, A is a binding site where the protein binds to the site in almost all cases. On the other hand, B does not have a binding site but accidentally (as a result of experimental or other errors mentioned), in about every 1 out of 10 cases, we see read counts on B. So, in the RIP-Seq data, we will observe that B has read counts 10 times more than A. But, then, a comparison of the read counts on A and B using only *only* RIP-Seq data would suggest calling B a binding site and A a non-binding site. ASPeak overcomes this problem by taking the expression levels of A and B into account. It finds a background noise for A by estimating the parameters p and r, of the NB distribution, from the intervals having similar expression levels to that of A and does the same thing for B. Then it uses a statistical test to distinguish signal from noise. This way, highly expressed parts of RNA and less expressed parts of RNA are considered separately in computing the background noise. This gives us a more robust determination of the protein binding sites namely peaks.

In the peak finding step, ASPeak needs two parameters p and r to statistically test the significance of the reads. NB parameters file is produced by running the StepNBParameters step. One of the inputs of this step is the RPKM file which requires RNA-Seq data. So, if there is no RNA-Seq data, there can't be an RPKM file and hence there can't be a NB parameters file. In this case, ASPeak can still be used by skipping the previous steps StepPrepInput, StepMakeRPKM and StepNBParameters. For this, prepare yourCLIP- / RIP- Seq and annotation data as described above. In the command line arguments, set the -nornaseq parameter. Alternatively, in the parameter file, you can set the flag nornaseq to 1. This will tell ASPeak

not to look for a NB parameters file. ASPeak will estimate the parameters of the NB distribution for each position locally. The read counts in frames of given radii, whose center is the current position, are taken as sample to estimate the parameters. Hence, parameters will be obtained from Rip / Clip Seq data not RNA-Seq data. This is done as follows. There is a parameter called frameRadii in the parameter file. It holds several frame radius values separated by commas. For each position, for each radius value, the read counts are averaged in a frame of this radius centered at the current position. So, for each radius i, this will give a pair of parameters p_i and r_i. Let p_{i_0} and r_{i_0} be the parameters maximizing the the expected value of the corresponding Negative Binomial distribution. Then p_{i_0} and r_{i_0} is assigned to be the parameters of the current position.

Genome Annotation: Regions & Intervals

ASPeak computes RPKM, estimates the parameters of the Negative Binomial distribution and calls peaks on a predefined part of the genome that we call *region*. Some typical examples of regions are introns, exons, three prime untranslated region and so on. So, for instance, all exons constitute a region of the genome. This region is defined in a bed file to ASPeak. Users can define as many regions as they wish by providing a BED file in the bed directory. In the subsection Example: Creating a Genome annotation file, we describe how to obtain a BED file that describes a region using the annotations in the UCSC Genome Bioinformatics Site. ASPeak gets the region definitions by reading all the BED files in the directory given in -beddir parameter. The format of BED file is given in the subsection Bed File. Given a library (RIP-Seq, RNA-Seq or Control), for a region, ASPeak first intersects the input with that region and then calls the peaks. Consequently, peaks that don't reside on any region will not be reported by ASPeak.

Each row of a region is a contiguous part on the genome defined by start and end positions. We call these parts *intervals*. For example, let exons.bed be the BED file that defines the exons region. Then each row of exons.bed is an exon. The start position of an interval is inclusive whereas the end position is exclusive. Peaks are called on each interval using interval specific RPKM values and the Negative Binomial distribution parameters.

Example: Creating a Genome annotation file

ASPeak calls peaks for each region separately. Regions can be defined by the user in Bed files or they can be obtained from elsewhere. Each region must be defined in a separate Bed file with .bed extension. Here, as an example, we explain how to get the region definitions of three prime untranslated exon regions from the UCSC Genome Bioinformatics Site.

1. Go to the UCSC web site at https://genome.ucsc.edu/index.html.
2. Click on the "Tables" link at the top.
3. Select the following fields: genome→human, assembly→hg18, group→Gene and gene prediction tracks, track→RefSeq Genes, table→RefGene. Write "utr3.bed" in the output file box. Click "get output". See Figure 1.
4. Choose "3' UTR Exons". See Figure 2. Click "Get BED". This will download the bed file to your computer.

Figure 1: Getting 3‘ UTR from UCSC: Step 1

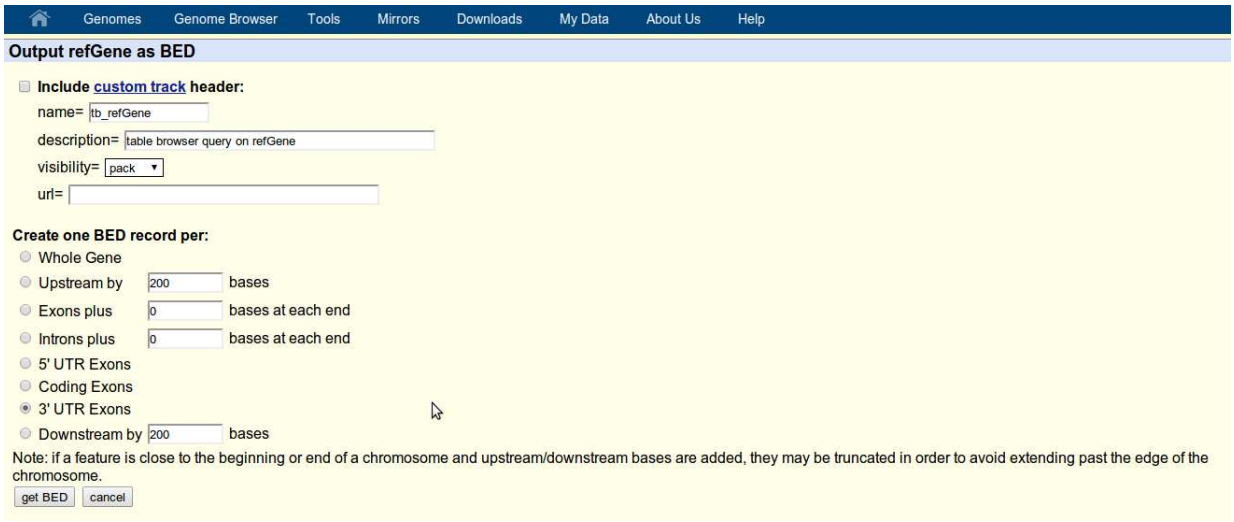


Figure 2: Getting 3‘ UTR from UCSC: Step 2

A Case Study

Say Jane has a RIP-Seq data to analyze. She got a copy of ASPeak and extracted it under her home directory in /home/jane/ASPeak. She ran ASPeak with the example data set. She has ASPeak functioning.

Jane changes the working directory to the main directory of the ASPeak.

```
$> cd /home/jane/ASPeak
```

She checks the working directory

```
$> pwd
/home/jane/ASPeak
```

and its contents.

```
$> ls /home/jane/ASPeak
example LICENSE.txt README.txt
scripts MANUAL.pdf
```

Under this directory, she creates the directory named mydata in which she is going to put all her data.

```
$> mkdir mydata
```

She is interested in the peaks on introns and exons. She gets the annotation data from the UCSC Genome Bioinformatics Site. She puts her annotation data, which are in the files introns.bed and exons.bed, in the directory /home/jane/ASPeak/mydata/annotation. Jane checks the annotation directory and makes sure that the file names and their extension are correct.

```
$> ls mydata/annotation
exons.bed introns.bed
```

For example, the first three lines of exons.bed, are as follows.

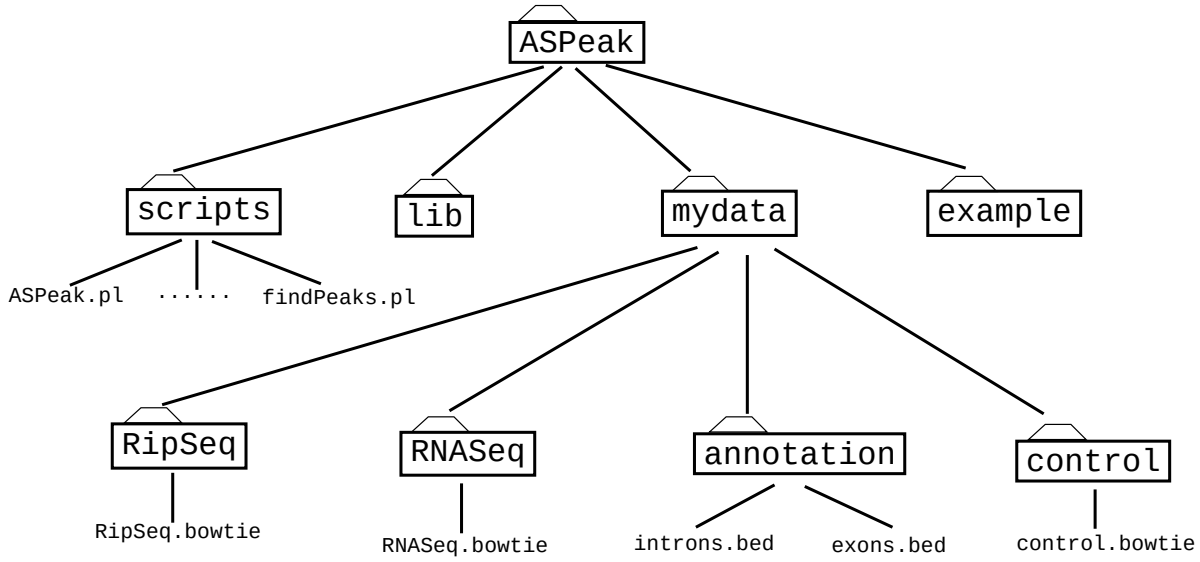
```
chr1 67051159 67052451 NM_024763_exon_0_0_chr1_67051160_r 0 -
chr1 67060631 67060788 NM_024763_exon_1_0_chr1_67060632_r 0 -
chr1 67065090 67065317 NM_024763_exon_2_0_chr1_67065091_r 0 -
```

Jane aligned her RIP-Seq row data using bowtie. As output, she got one file named RipSeq.bowtie. Jane puts her RIP-Seq data under the directory /home/jane/ASPeak/mydata/RipSeq/. Jane has RNA-Seq data and control data as well and she also aligned them using bowtie. She puts the RNA-Seq file RNASeq.bowtie in /home/jane/ASPeak/mydata/RNASeq/ and control file in /home/jane/ASPeak/mydata/control/. The rows of her BOWTIE files look like the following.

```
1012_1273_802_F3 - chr1 7800 CTCCGCCGGCGCAGGCGCGCAGGA :::::::::::::::::::: 0
1012_1273_803_F3 - chr1 7932 TCCGCCGGCGCAGGCGA ::::::::::::::::: 0
1012_1273_804_F3 - chr1 7951 CCGCCGGCGCAGGCGAGAGC :::::::::::::::::: 0
```

Her directory structure should look like Figure 3.

Figure 3: Directory Structure of Jane before running ASPeak



She changes the directory to the main directory of ASPeak.

```
$> cd /home/jane/ASPeak
```

Now Jane is all set. She wants to run ASPeak on her won computer so she doesn't set the cluster parameter. She wants the output in the directory called myout. So she will set the parameter -outdir to myout. On the terminal, she types the following to run ASPeak.

```
$> ./scripts/ASPeak.pl -lib mydata/RIPSeq/RIPSeq.bowtie \
-beddir mydata/annotation -rnaseq mydata/RNASeq/RNASeq.bowtie \
-control mydata/control/control.bowtie -outdir myout
```

Note that she could run ASPeak without the control library simply by

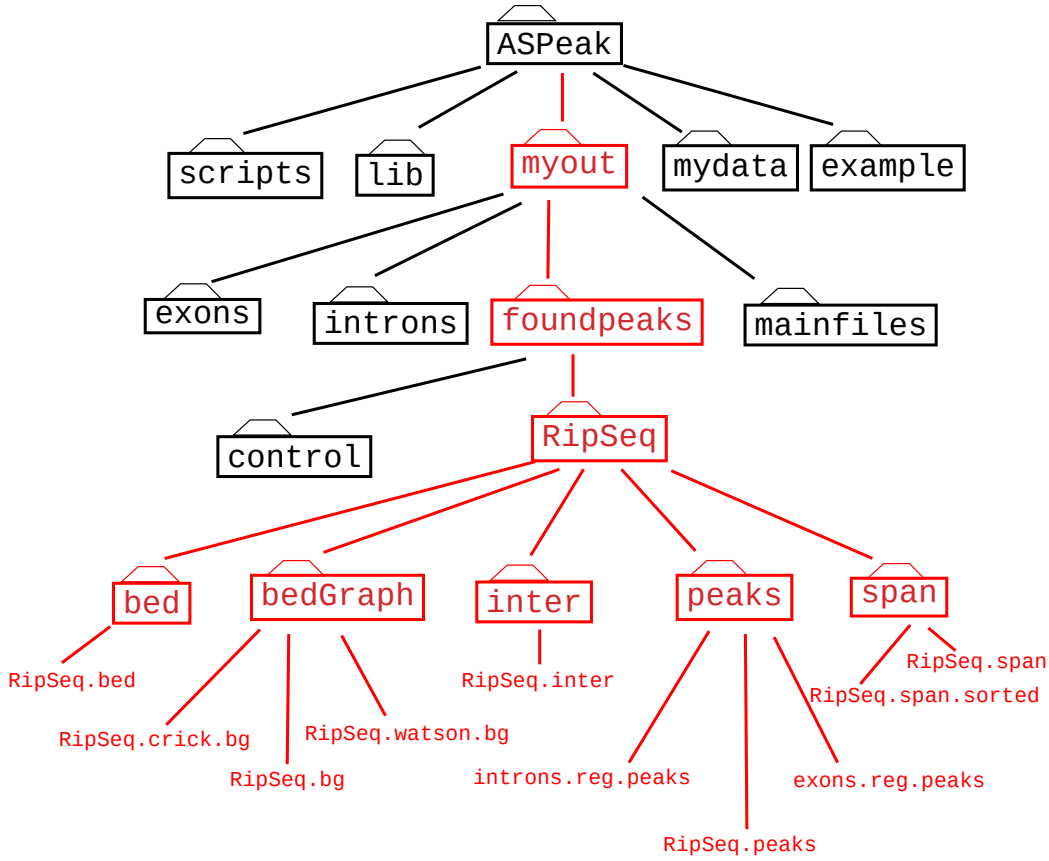
```
$> ./scripts/ASPeak.pl -lib mydata/RIPSeq/RIPSeq.bowtie \
-beddir mydata/annotation -rnaseq mydata/RNASeq/RNASeq.bowtie \
-outdir myout
```

Also, Jane could run ASPeak without the RNA-Seq library though this is highly discouraged.

```
$> ./scripts/ASPeak.pl -lib mydata/RIPSeq/RIPSeq.bowtie \
-beddir mydata/annotation -nornaseq -outdir myout
```

After running ASPeak with the RIP-Seq, RNA-Seq and control libraries, she gets the output files in the directory /home/jane/ASPeak/myout/foundpeaks. Her final directory structure is given in Figure 4.

Figure 4: Directory Structure After Running ASPeak



Jane can see all the peaks in detail by reading the file
`/home/jane/ASPeak/myout/foundpeaks/RipSeq/peaks/RipSeq.peaks`

```
$> more /home/jane/ASPeak/myout/foundpeaks/RipSeq/peaks/RipSeq.peaks
```

The rows of this file looks like the following.

```
chr1 + NM_4777_intron_chr1_300064_f 96774 11 5 3 300064 300067 -1 300065 0 7.17e-06 <0.00e+00
chr1 - NR_0245_exon_0_chr1_7778_r 147 6 6 1 7814 7815 140.95 7814 1 2.43e-05 < 2.50e-01
chr2 + NM_0649_exon_0_chr2_262259_f 223 12 6 3 262315 262319 120.23 262318 0 7.54e-05 < 2.50e-01
```

Some important attributes of this file are as follows. The third column is the identifier of the interval which also contains region information. Total tag count of the peak is given in the fifth column. The maximum hight of the peak is in the sixth column. The eighth and ninth columns are the start and end positions of the peak. Column 10 contains the RPKM value of the interval. When there is no RNA-Seq data available for the interval, there will be no abundance information and hence no RPKM value. In order to indicate this, ASPeak will put -1 on the RPKM field. For example, the first peak in the above table has no abundance information and therefore its RPKM value is -1. Column 13 is the p-Value of the peak where this value comes from the Negative Binomial distribution. The last column is the false discovery rate of the peak. This value is obtained by comparing the peaks in the RIP-Seq library with the control library. Note that, Jane wouldn't see the last column had she run ASPeak without

a control library. More detailed information on the peaks file can be found in the subsection Output File (Peaks).

Peaks are also reported in the directory `home/jane/ASPeak/myout/foundpeaks/span` in `BED` format. For each peak, the start and end positions in the bed file and the peaks file coincide.

Alternatively, Jane can view the peaks by uploading the bedGraph file `/home/jane/ASPeak/myout/foundpeaks/bedGraph/RipSeq.bg` to a genome browser. Also in the directory `/home/jane/ASPeak/myout/foundpeaks/bedGraph/`, bedGraph plots of the peaks on the Watson and Crick strands are given separately.

Moreover, the part of the RIP-Seq data, covered by the peaks, in `BED` format, is in the file `/home/jane/ASPeak/myout/foundpeaks/bed/RipSeq.bg`.

Abundance Sensitive Peak Detection Algorithm

In this section, we describe our abundance sensitive peak detection algorithm and explain the components of `ASPeak`.

Motivation

We observed a significant correlation between the overall RIP signal and transcript abundance as measured by RNA-seq. This correlation is expected as RNA immunoprecipitation is an enrichment strategy. Consider two mRNAs that differ in abundance by 100-fold such that there are 10 molecules of RNA X and 1000 of RNA Y in each cell. Even if all molecules of X compared to $1/10$th of Y are bound by the protein of interest, we will obtain 10 times more signal from Y. If one does not take into account the initial abundance of the two molecules, the signal on RNA X might not be detected leading to biased results. To address this limitation, we developed a RNA abundance sensitive peak-detection algorithm for RIP-seq data that uses RNA-Seq from the same cell type to inform the noise in the RIP-seq data.

The Algorithm

Our algorithm computes an expression-sensitive background for each interval (see Genome Annotation: Regions & Intervals for the definition of interval), having a positive RPKM value but uses expression-insensitive backgrounds for intervals having no abundance information, i.e., for intervals with 0 RPKM. We modeled the read counts in each interval as a Negative Binomial distribution, parametrized by interval specific parameters p and r. In order to estimate p and r for each interval, we used the method of moments as follows. First, abundance of each interval (namely interval RPKM) is calculated from the RNA-Seq library. All intervals are then ranked by their RPKM values. For each interval, 1000 intervals with the closest RPKM level are taken. Then these intervals are ranked based on their $\frac{\text{interval count}}{\text{interval length}}$ value, where interval counts are coming from the RIP-Seq library. The intervals contained in the top and bottom 5% of this list are removed to minimize outlier effects.

As the next step, we use the estimators of p and r, coming from the method of moments, using the number of reads at each position of the remaining 900 intervals as follows. Let μ_1 and μ_2 be the first and second moments, around the origin, of the Negative Binomial distribution respectively. Let n_{ij} be the read count at the j^{th} position of the i^{th} interval. Let ℓ_i be the length of the i^{th} interval. Then we can estimate μ_1 and μ_2 by

$$\mu_1 \approx m_1 = \frac{\sum_{i=1}^{900}\sum_{j=1}^{\ell_i} n_{ij}}{\sum_{i=1}^{900} \ell_i} \qquad \text{and} \qquad \mu_2 \approx m_1 = \frac{\sum_{i=1}^{900}\sum_{j=1}^{\ell_i} n_{ij}^2}{\sum_{i=1}^{900} \ell_i}$$

Then, using m_1 and m_2, the parameters p and r can be estimated as follows

$$p = \frac{m_1}{m_2 - m_1^2} \qquad \text{and} \qquad r = \frac{pm_1}{1-p}$$

We would like to emphasize that the parameters p and r are interval specific.

Next, peaks on each interval are detected using interval specific p and r in each interval using the Negative Binomial (NB) distribution whose probability mass function is given by

$$f(x, p, r) = \binom{r + x - 1}{x} p^r (1 - p)^x.$$

where x is the number of read counts. The function $f(x, p, r)$ gives the probability of observing exactly x reads.

The probability distribution function of the Negative Binomial distribution is given by

$$F(x, p, r) = \sum_{i=0}^{x} f(i, p, r), \tag{1}$$

which is the probability of observing x or less reads.

Using $F(x, p, r)$, a p-value ($pVal$) is obtained that reflects the probability of observing x or more reads at the given position given the parameters p and r where

$$pVal = 1 - F(x - 1, p, r). \tag{2}$$

We scan the positions of each chromosome in ascending order. When in the state of searching, a peak is defined to begin in a position having $pVal \leq 0.01$. Once a peak is found, the positions following the peak are tested using the NB distribution for extension of the peak. Let $pVal_{peak}$ be the overall p-value of the peak defined in the subsection p-alue Calculation. Let i be the index of the current position in the peak. We test the proceeding $L+1$ positions, i.e., $i+1$,$i+2$,$\ldots$,$i+L+1$, as described in the next paragraph. If the test turns out to be positive, we include the next position, i.e., $i+1$, in the peak and move on to the position $i+1$. The peak is defined to end when the result of this test is negative. This way, we allow L gaps (positions having relatively lower reads) in the peaks. The default value of L is 5. The user can change this value by setting the commmand line option `-gapnumber`.

For the i^{th} position, the proceeding L positions are tested as follows. We temporarily include the position $i+1$ in the peak. If the p-value of the position $i+1$ and the p-value of the peak are both less than 0.01 then the result of the test is positive. Otherwise, we move to the position $i+2$ and repeat this procedure. If negative, then we repeat this procedure till $i+L+1$. The result of the procedure is negative if the p-value of the position or the p-value of the peak is greater than 0.01 in every case. Otherwise, the result is positive. A pseudocode for this procedure is given in Table 3.

```
includeNextPosition := FALSE;
temporaryPeak := peak;

for k:=1 to L+1
  temporaryPeak := temporaryPeak ∪ {position_{i+k}};

  if pVal_{position_i} ≤ 0.01    and    pVal_{temporaryPeak} ≤ 0.01
    includeNextPosition := TRUE;
  end if

end for
```

Table 3: Pseudocode that tests the $i+1^{th}$ position for inclusion in the peak. Suppose that a peak is already found and contains the i^{th} position as the rightmost position. Here we test the $i+1^{th}$ position for inclusion in the peak. in the end, if the logical variable `includeNextPosition` is `TRUE` the $i+1^{th}$ position is included in the peak. Else, the i^{th} position is marked as the end position of the peak, the peak is saved and search for a new peak begins.

For intervals with 0 RPKM, i.e., intervals having no RNA-Seq data, we use a local window approach, similar to that of MACS (Zhang et al., 2008), as follows. For a position that belongs to such an interval, and for each of the given radius values (default: 1k, 5k and 10k) we estimate the parameters p and r using the read counts in the frame. Then we assign the parameters, giving the maximum expected value of the random variable of the Negative Binomial distribution, to be the p and r values of the given position.

We also implement a filtering step to remove artificial peaks arising from over-amplification during PCR. Specifically, we remove peaks with one nucleotide species. These peaks are marked as "block" peaks, in the `.peaks` file, and likely represented PCR jackpots and should not be used in subsequent analysis in our opinion.

The output in the `.peaks` file includes interval specific read counts in the RNA-Seq library, abundance values quantified by reads per kilobase per million (RPKM). Also an overall p-value of the peak, as defined in subsection p-value Calculation, is saved in the `.peaks` file. ASPeak can carry out this computation in parallel where the data and annotation is split by chromosome and region. Also it has further parallelization abilities that run each chromosome separately and merge them after the runs are finished. Although the implementation has four main steps, all of them can be executed with a single command thus streamlining the application of the software. Additionally, detailed debugging information about the individual steps of the implementation are reported with testing options. Finally, when multiple RIP experiments are analyzed with a common RNA-seq data, the abstraction into four distinct steps in the implementation prevents unnecessary repetition in the calculations. Specifically, first three steps can be skipped when using the same RNA-seq data and the final peak calling step of the algorithm can be executed.

p-value Calculation

In this subsection, we explain how an overall p-value for a peak is calculated. Let X be a random variable having a NB distribution with parameters p and r. Then the p-value (denoted

by $pVal$) of observing x or more reads, i.e., the probability of $X >= x$, is given in (2). We can extend this definition to reads coming from many positions. Let's assume that there is a peak spanning the positions $s, s+1, \ldots, e$ and having read counts $x_s, x_{s+1}, \ldots, x_e$. Also assume that the corresponding random variables have the same NB parameters p and r. Then the p-value of this peak is defined to be

$$pVal_{\text{peak}} = 1 - F\left(\sum_{i=s}^{i=e} x_i \ - 1,\ p,\ (e-s+1)r\right) \tag{3}$$

which is the probability of observing $\sum_{i=s}^{i=e} x_i$ or more reads, totally, in the positions $s, s+1, \ldots, e$.

Let $X_1, X_2, \ldots, X_k$ be independent random variables having NB distribution. Suppose that, for all $1 \leq i \leq k$, the random variable X_i has parameters p and r_i. Then, it is well-known that the random variable $X = \sum_{i=1}^{i=k} X_i$ has also NB distribution with parameters p and $r = \sum_{i=1}^{i=s} r_i$. This justifies the definition (3).

Note that for each random variable X_i above, the probability of observing $X_i = x_i$ is

$$P(X_i = x_i) = \binom{r_i + x_i - 1}{x_i} p^{r_i}(1-p)^{x_i}.$$

Then the probability of observing $X = x$, where $r = \sum_{j=1}^{k} r_j$ and $x = \sum_{j=1}^{k} x_j$, is

$$\begin{aligned}
P(X=x) &= \sum_{\sum x_i = x} P(X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k) \\
&= \sum_{\sum x_i = x} P(X_1 = x_1) \cdot P(X_2 = x_2) \cdots P(X_k = x_k) \\
&= \sum_{\sum x_i = x} \left(\prod_{i=1}^{k} \binom{r_i + x_i - 1}{x_i} p^{r_i}(1-p)^{x_i} \right) \\
&= \sum_{\sum x_i = x} \left(\prod_{i=1}^{k} \binom{r_i + x_i - 1}{x_i} \right) p^{\sum_{j=1}^{k} r_j} (1-p)^{\sum_{j=1}^{k} x_j} \\
&= \sum_{\sum x_i = x} \left(\prod_{i=1}^{k} \binom{r_i + x_i - 1}{x_i} \right) p^{r} (1-p)^{x} \\
&= p^r (1-p)^x \sum_{\sum x_i = x} \left(\prod_{i=1}^{k} \binom{r_i + x_i - 1}{x_i} \right) \\
&= p^r (1-p)^x \binom{r + x - 1}{x}
\end{aligned}$$

This proves that the random variable X has NB distribution with parameters p and $r = \sum_{j=1}^{k} r_j$.

False Discovery Rate Calculation

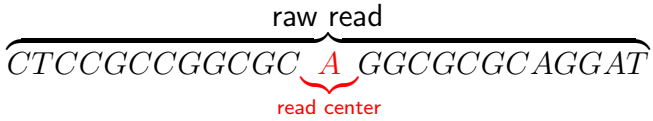
When control library data is provided along with CLIP- / RIP- Seq data, `ASPeak` computes the False Discovery Rates of the found peaks. FDR's are reported on the last column of the `.peaks` file.

FDR calculation is done as follows. The peaks are sorted by their p-values. For each p-value, say pVal, all the peaks, both in the control library and the RIP-Seq library, having p-value equal to or less than pVal are taken. Let V be the number of those peaks in the control library having p-value less than or equal to pVal. These peaks are called *false discoveries*. Let S be the number of the peaks in the CLIP- / RIP- Seq library having p-value less than or equal to pVal. These peaks are called *true discoveries*. For pVal, the false discovery rate is the ratio of false discoveries to all discoveries. That is

$$\text{FDR}_{pVal} = \frac{V}{V+S}$$

A Note on the Independence of Random Variables

In CLIP- / RIP- Seq and RNA-Seq experiments, aligned data comes in nucleotide sequences of varying length. We expect that read centers of these sequences reflect the positions of the proteins of interest on the given RNA most accurately. So raw data is internally converted to `bedGraph` format by `ASPeak` as follows. Each fragment is counted as one read **at the center** of the fragment. Say, there is a read of consecutive nucleotide sequences of length N begining at position i. This read is interpreted as a read count **only** at the position $i + \lfloor N/2 \rfloor$, where $\lfloor x \rfloor$ is the largest integer less than or equal to x. For example, suppose that the sequence `CTCCGCCGGCGCAGGCGCGCAGGAT` was aligned on a region of interest on the genome. Say the first nucleotide of this sequence is on the 10001^{th} position. Then, in the corresponding bedgraph file, we only increment the nucleotide position corresponding to the center of the read, which is 10013 in this case.



Bed Graph File Before Processing the read `CTCCGCCGGCGCAGGCGCGCAGGAT`

⋮			⋮
chr1	10011	10012	26
chr1	10012	10013	27
chr1	10013	10014	30
chr1	10014	10015	25
chr1	10015	10016	25
chr1	10016	10017	23
⋮			⋮

Bed Graph File After Processing the read `CTCCGCCGGCGCAGGCGCGCAGGAT`

⋮			⋮
chr1	10011	10012	26
chr1	10012	10013	27
chr1	10013	10014	31
chr1	10014	10015	25
chr1	10015	10016	25
chr1	10016	10017	23
⋮			⋮

Consequently, in the `bedGraph` file, the existence of reads in each position are independent of each other. Hence the Negative Binomial test can be applied on this data to call the peaks.

System Overview

Our peak finding algorithm consists of four major steps. The remaining steps mentioned above are for preprocessing the input and post-processing the output. The four main steps of `ASPeak`, in the running order, are as follows.

Count	Finds the interval lengths and interval counts in the CLIP- / RIP- Seq data.
RPKM	Finds the abundance, i.e., RPKM values, of the intervals.
Parameter Estimation	Estimates the NB parameters of each interval.
Peak	Using the NB parameters and the CLIP- / RIP- Seq data, finds the peaks.

A latter step uses the output of former steps as shown in Figure 5.

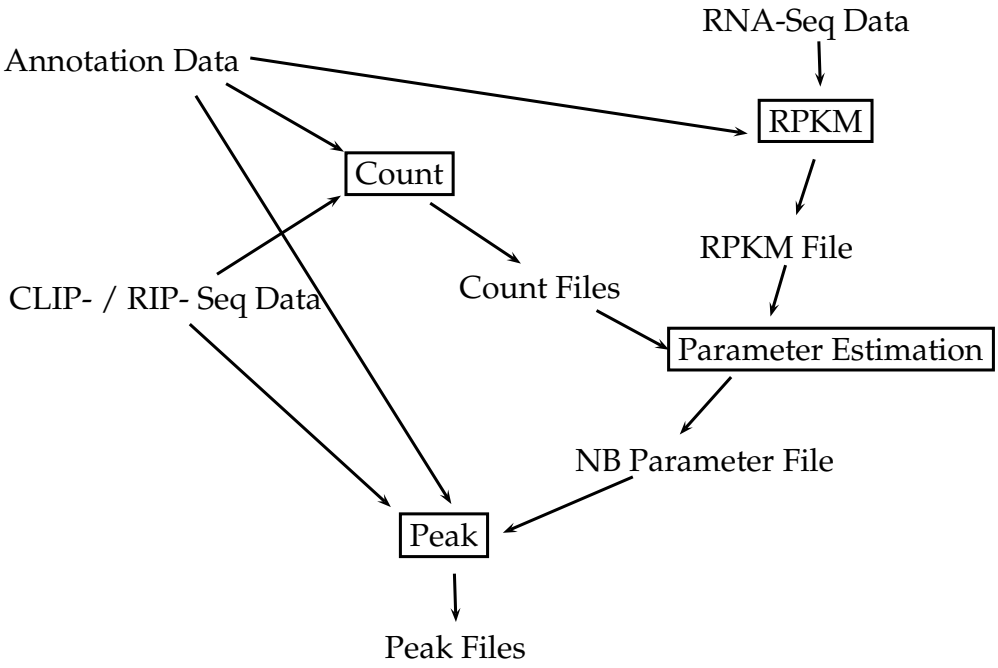


Figure 5: Data Flow Diagram of `ASPeak`

Count: This part is implemented in `findCounts.pl` file. It can be run in parallel for each chromosome file.

RPKM: This part is implemented in `makeRPKM.pl` file. It can be run in parallel for each chromosome file.

Parameter Estimation: This part is implemented in `estimateParameters.pl` file.

Peak: This part is implemented in `findPeaks.pl` file. It can be run in parallel for each chromosome file.

File Formats

Parameter File

ASPeak comes with a default parameter file named default.txt in the scripts foder. If no -param argument is given in the command line, this parameter file, that is default.txt, will be used to get the settings. Users can provide their own parameters by specifying -param in the command line. Though, for most users, we recommend using the default parameter file default.txt.

The parameter file is read by the main script ASPeak.pl and its contents are used to determine the command line arguments of the other components of the system. ASPeak has a default parameter file which is suitable for most users. Power users can use their own parameter file with -p parameter_file option. If the command line argument -p is not specified, then the default file will be used by ASPeak. Also, the name and location of the default parameter file that comes with ASPeak should not be changed.

The lines starting with # are comment lines and they will be ignored by ASPeak. The parameters are defined as

```
parameter_name = value
```

There has to be a space between parameter_name and = and, between = and value. Both parameter_name and value are case **sensitive**.

The flags from StepCheckParams to StepSubmitJobs determine the steps to be executed by ASPeak. If the flag is 0, the corresponding step will not be executed. If the flag of a step is nonzero, the corresponding step will be executed. Note that the order steps is important as output of a former step is input of latter steps. If you don't have a through knowledge of these steps, set them all to 1.

The entries of the parameter file are as follows.

Parameter File Entries

MainDir	:	The main directory containing all the system directories of ASPeak.
ExecDir	:	The directory containing the scripts of ASPeak.
RunInCluster	:	Set this to 0 if you want to run ASPeak stand-alone. Set this to 1 to run it in the cluster.

Parameter File Entries		
Queue	:	The cluster node that ASPeak scripts are going to be submitted to. Please consult your system documentation to get a list of the available cluster nodes. This parameter has effect only if the parameter RunInCluster is set to 1. If the value of this parameter is ANY, then jobs will be submitted to any cluster node available.
StepCheckParams	:	Set to 1 if you want to run parameter file checking step.
StepGetChroms	:	Set this to 1 to get the chromosomes from the input data.
StepConvert2Bed	:	Set this to 1 to convert the input to bed file format.
StepCountMappedReads	:	Set this to 1 to find the counts of the mapped reads.
StepRegionSeparation	:	Set this to 1 to partition the genome into regions such as introns, exons, utr3 and etc.
StepChromSeparation	:	Set this to 1 to partition the input into chromosomes.
StepPrepInput	:	Set this to 1 to find the sum of counts and sum of the square of the counts.
StepMakeRPKM	:	Set this to 1 to get the RPKM of the intervals from the RNASeq data.
StepNBParameters	:	Set this to 1 to estimate the parameters of the Negative Binomial distribution.
StepPeak	:	Set this to 1 to find the peaks.
StepCombinePeaks	:	Set this to 1 to combine the peaks in different regions into one file.
StepCalcFDR	:	Set this to 1 to find the false discovery rate for each peak (requires control data).

Parameter File Entries (continued)

StepPeaks2Bed	:	Set this to 1 to convert output (peak file(s)) to bed format.
StepMakeBedGraph	:	Set this to 1 to plot the peaks in bedGraph file format
StepWigToBigWig	:	Set this to 1 to visualize the peaks on the genome by converting the output to bigWig format (requires WigToBigWig).
StepSubmitJobs	:	Set this to 1 to submit when running ASPeak in cluster to run jobs in parallel.
Explanations	:	Set this to 1 to run in verbose mode, 0 to run in silent mode.
Debug	:	Set this to 1 to see the debug information. Otherwise set this to 1.
Strand	:	Set this to 1 to call peaks for each strand, namely Watson and Crick, spearately. Set this to 0 to call the peaks for both strands together.
db	:	The name of the annotation database
nornaseq	:	Set this to 0 if you have RNASeq data. Else, set this to 0.
InputDir	:	The main directory containing input data.
BedDir	:	The directory containing the genome annotation files. The extensions must be .bed
LibName	:	RipSeq library name
Lib	:	RipSeq library path
LibFormat	:	RipSeq library format, e.g. bam, sam, bed, bowtie

Parameter File Entries (continued)

`ControlName`	:	Control library name
`Control`	:	Control library path
`ControlFormat`	:	Control library format
`RNASeqName`	:	RNASeq library name
`RNASeq`	:	RNASeq library path
`RNASeqFormat`	:	RNASeq library format
`OutDir`	:	The main directory that contains the output files and related subdirectories.
`LogDir`	:	For each run, the execution summary is held in this directory.
`RefDir`	:	RefSeq files are in this directory.
`frameRadii`	:	In order to determine the parameters of the Negative Binomial (NB) distribution for each interval, for each radius specified in this parameter, the read counts of the neighboring positions are used to estimate the parameters of the NB. The parameters that give the maximu expected value of NB are taken as the estimated parameter for the articular position. The radii must be separated by commas without any whitespace in between.
`estimationRadius`	:	The number of intervals above and below the current interval in determining the parameters of the NB distribution.
`gapNumber`	:	The maximum number of tolerated consecutive positions having relatively low reads (gaps) in the peaks.

Counts File

This file is the output of the step called `StepPrepInput`. It is required for estimating the parameters in the step called `StepNBParameters` to estimate the parameters of the Negative Binomial distribution.. This file contains the interval counts information of the Rip / CLIP-Seq data. For each interval, there is a separate line in the Counts File telling the chromosome of the interval, the name of the interval, the length of the interval, total tag count of the interval, the sum of the squares of tag counts and RPKM value of the interval. For example, suppose that for the interval under consideration, the tag count of the i^{th} nucleotide positio is n_i. Then Column 4 is $\sum_i n_i$ and Column 5 is $\sum_i n_i^2$.

Counts File Columns

Column 1	:	Chromosome
Column 2	:	Interval name
Column 3	:	Interval length
Column 4	:	Sum of the reads
Column 5	:	Sum of the square of the reads
Column 6	:	RPKM value of the interval

RPKM File

RPKM file is needed to estimate the parameters of the intervals for the Negative Binomial distribution. This file is the output of the `StepMakeRPKM` step.

In `StepNBParameters`, this file is used as follows. The intervals are sorted with respect to their rpkm values. Intervals having similar RPKM values are grouped and their counts are averaged to determine the RPKM value. More precisely, 500 intervals above and below the current interval, in the sorted list are taken. Outliers are removed by discarding the 50 intervals having the highest read count averages and 50 intervals having the lowest read count averages. The columns of the rpkm file are as follows:

RPKM File Columns

`Column 1`	:	The name of the chromosome (eg. chr1, chrX)
`Column 2`	:	Interval Name
`Column 3`	:	RPKM

bedGraph File

CLIP- / RIP- Seq and control libraries are processed and then kept internally in bedGraph format by ASPeak. The peak finding step, namely `findPeaks.pl`, uses bedGraph as input.

A very important point about the bedGraph file is that, **the positions must be sorted in ascending order**. Each line of the wig file has 4 columns.

bedGraph File Columns		
Column 1	:	Chromosome Number
Column 2	:	Beginning of the Read Position
Column 3	:	Column 2 plus 1
Column 4	:	Read Count

Negative Binomial Parameters File

Lambda file is needed for calling the peaks in the Rip / Clip Seq data in "type=peak" step. It is the output of the step "type=lambda". It contains the lambda value of the exons. For each exon, there is a line in the lambda file. Note that, if nornaseq option is set to 1, then the peaks will be called without the lambda file though this is not recommended. The lambda file format is as follows.

NB Parameters File Columns		
Column 1	:	Interval name
Column 2	:	p parameter
Column 3	:	r parameter
Column 4	:	RPKM value of the interval
Column 5	:	Interval Count
Column 6	:	Interval Length

Bed File

Genome annotation data should be provided in Bed file format to ASPeak. Also, CLIP- / RIP- Seq data can be given in Bed format. Moreover, ASPeak uses different formats to represent the peaks for different purposes. One of these formats is Bed. More precisely, ASPeak outputs only the parts of the input bed files that correspond to peaks. The columns

must be separated by tabs in Bed files. Detailed information on Bed format can be found at https://genome.ucsc.edu/FAQ/FAQformat.html#format1

BAM & SAM Files

CLIP- / RIP- Seq, RNASeq and control libraries can be given to ASPeak in BAM or SAM file formats. Detailed information on BAM and SAM formats can be found at http://genome.sph.umich.edu/wiki/SAM or at http://samtools.sourceforge.net/SAM1.pdf
You need to have BEDTools and SAMTools installed in your system to be able to use these formats.

Output File (Peaks)

This file is the output of the step "type=peak". For each found peak, there is a separate line on the output file giving detailed information about the peak. For each chromosome, a separate output file is created. This file is also the input of the step "type=makewig". The columns of the output file are as follows. Note that the last column (Column 14) exists only when there is a control library. That is, the peak files will have only the first 13 columns when ASPeak is run without control data.

Output File (Peak) Columns		
Column 1	:	Chromosome number
Column 2	:	Strand information
Column 3	:	Interval Identifier
Column 4	:	The length of the interval
Column 5	:	The number of reads in the peak
Column 6	:	Maximum height for a detected peak
Column 7	:	Foot-size of the peak
Column 8	:	Start position of the peak
Column 9	:	End position of the peak
Column 10	:	Interval RPKM. This value is -1 when there is no RNA-Seq data available.
Column 11	:	Weighted center for a peak that is a position which is found by summing up the heights from the beginning of a peak for each position till to pass the half of the # all sums of the heights in the peak.
Column 12	:	pVal of the peak. The minimum pVal is 1e-45
Column 13	:	Upper bound for the False Discovery Rate (FDR) of the peak. Note that this column exists only when there is a control library.

Bug Reporting

If you find a bug, please send an email to hakan.ozadam@umassmed.edu or alper.kucukural@umassmed.edu.

Before reporting a bug, please make sure that you have the latest version running. Please provide the minimal input data to us that causes the bug.

How to Cite

Alper Küçükural, Hakan Özadam, Guramrith Singh, Melissa Moore, Can Cenik, *ASPeak: An Abundance Sensitive Peak Detection Algorithm for RIP-Seq*, preprint.

Authors

Can Cenik, PhD
Alper Küçükural, PhD
Hakan Özadam, PhD

Acknowledgements

ASPeak uses the CPAN package Math::CDF, freely available at http://search.cpan.org/ callahan/Math-CDF-0.1/CDF.pm, for computing the probability distribution of the Negative Binomial distribution.

This work is supported by NIH grant # GM53007.

License and Copyright

Supplementary Material

Parameter Estimation of the Negative Binomial Distribution

Since CLIP- / RIP- Seq experiments and sequencing are prone to errors, there is some background noise in CLIP- / RIP- Seq data. So, the major problem in finding peaks is distinguishing signal from noise in the given data set. Roughly, a peak is defined to exist in those positions having *significantly higher* reads than the background noise. Identifying significantly higher reads is done using a statistical test where the number of reads in each nucleotide position is considered to be a random variable, say X. Experimental data suggest that the distribution of X can be modeled using the Negative Binomial distribution. For this reason, a hypothesis testing is implemented in `ASPeak` that uses the Negative Binomial distribution. The parameters of the Negative Binomial distribution is determined from the data using the method of moments. In this subsection, the details of this procedure are explained rigorously.

There are different conventions on the definition of the Negative Binomial (NB) distribution in the literature. In some definitions, x starts from 0 in some others x starts from r. In some definitions, p denotes the probability of success whereas in some others p denotes the probability of failure. Consequently, they have different mean, variance and moments. Here we use the following definitions.

Definition 1. Let p be the probability of observing success and $1-p$ be the probability of observing failure in one trial. The probability of observing x successes until observing r failures is given by

$$p(x) = \binom{r+x-1}{x} p^x (1-p)^r. \tag{4}$$

Definition 2. Let p be the probability of observing success and $1-p$ be the probability of observing failure in one trial. The probability of observing x failures until observing k successes is given by

$$p(x) = \binom{r+x-1}{x} p^r (1-p)^x. \tag{5}$$

Note that the transition from Definition 1 to Definition 2 is achieved by substituting p for $p-1$ in (4). Hence, if we have one property for Definition 1, we can get the same property for Definition 2 by this substitution.

If a random variable X has the probability mass function as defined in (1) or (2), X is said to have Negative Binomial distribution. The parameters of $p(x)$ are p and r.

`ASPeak` uses the CPAN module Math::CDF to compute the values of the NB distribution and Math::CDF is implemented according to Definition 2. So we will be using Definition 2 in determining the parameters of the NB distribution.

The expected value of a random variable X with probability mass function $p(x)$ is defined to be

$$\sum_x x p(x)$$

For the NB distribution, the expected value of X, w.r.t. Definition 1 , is given by

$$\mu_1 = E(X) = \frac{rp}{1-p} \tag{6}$$

So, w.r.t. Definition 2, it can be obtained by substituting p for $1-p$ in (6). Hence

$$\mu_1 = E(X) = \frac{r(1-p)}{p} \tag{7}$$

The expected value of X^2, w.r.t Definition 1, is given by

$$\mu_2 = E(X^2) = \frac{pr + p^2r^2}{(1-p)^2} \tag{8}$$

By the same substitution we get the expected value of X^2, w.r.t Definition 2 as

$$\mu_2 = E(X^2) = \frac{(1-p)r + (1-p)^2r^2}{p^2} \tag{9}$$

The Method of Moments

Given a sample, the method of moments can be used to estimate the parameters of the probability distribution function.

The i^{th} moment of the random variable X is defined to be the expected value of X^i.

Let $s_1, s_2, \ldots, s_N$ be a sample whose probability distribution is known. Then we can estimate the i^{th} moment, i.e., $E(X^i)$ as follows.

$$E(X^i) \approx m_i = \frac{s_1^i + s_2^i + \cdots + s_N^i}{N}$$

Then using the relation between the parameters and the moments, we can estimate the parameters of the probability function.

Let X be a random variable having NB distribution with respect to Definition 2. Then using (7) and (9), we have

$$\mu_2 = \frac{\mu_1}{p} + \mu_1^2 = \frac{\mu_1 + p\mu_1^2}{p}.$$

So we get

$$\begin{aligned} p\mu_2 &= \mu_1 + p\mu_1^2 \\ p(\mu_2 - \mu_1^2) &= \mu_1 \\ p &= \frac{\mu_1}{\mu_2 - \mu_1^2} \end{aligned}$$

By (7), we have

$$r = \frac{p}{1-p}\mu_1 \tag{10}$$

So if we have the estimations $\mu_1 \approx m_1$ and $\mu_2 \approx m_2$, then we can first estimate p as

$$p \approx \frac{m_1}{m_2 - m_1^2} \tag{11}$$

Then, we can estimate r as

$$r \approx \frac{p}{1-p} m_1 \tag{12}$$

Derivation of the Moments

Here we show how to obtain the first and second moments, namely (6) and (8). Note that these moments are with respect to Definition 1. The moments of Definition 2, i.e., (7) and (9), can easily be obtained by the substitution $p \to 1-p$ as explained above.

Clearly, we have

$$1 = \sum_{x=0}^{\infty} \binom{r+x-1}{x} p^x (1-p)^r \tag{13}$$

since all probabilities add up to 1. Differentiating both sides with respect to p, we get

$$0 = \sum_{x=0}^{\infty} x \binom{r+x-1}{x} p^{x-1}(1-p)^r - \sum_{x=0}^{\infty} r \binom{r+x-1}{x} p^x (1-p)^{r-1}$$

So

$$\sum_{x=0}^{\infty} x \binom{r+x-1}{x} p^{x-1}(1-p)^r = \sum_{x=0}^{\infty} r \binom{r+x-1}{x} p^x (1-p)^{r-1}.$$

Multiplying both sides by $p(1-p)$ we get

$$(1-p) \sum_{x=0}^{\infty} x \binom{r+x-1}{x} p^x (1-p)^r = pr \sum_{x=0}^{\infty} \binom{r+x-1}{x} p^x (1-p)^r.$$

Using (13), we get

$$\begin{aligned} (1-p)E(x) &= pr \cdot 1. \\ \text{So} & \\ E(x) &= \frac{pr}{1-p} \end{aligned}$$

This proves (6).

The second moment can be obtained by carrying out a similar computation.

We have showed above that

$$\sum_{x=0}^{\infty} x \binom{r+x-1}{x} p^x (1-p)^r = r \sum_{x=0}^{\infty} \binom{r+x-1}{x} p^{x+1} (1-p)^{r-1}.$$

Differentiating both sides with respect to p, we get

$$\begin{aligned}
&\sum_{x=0}^{\infty} x\binom{r+x-1}{x}\left[xp^{x-1}-rp^x(1-p)^{r-1}\right] \\
= \; & r\sum_{x=0}^{\infty}\binom{r+x-1}{x}\left[(x+1)p^x(1-p)^{r-1}-(r-1)p^{x+1}(1-p)^{r-2}\right]
\end{aligned}$$

Rearranging the terms yields

$$\begin{aligned}
&\sum_{x=0}^{\infty} x^2\binom{r+x-1}{x}p^{x-1}(1-p)^r - r\sum_{x=0}^{\infty} x\binom{r+x-1}{x}p^x(1-p)^{r-1} \\
= \; & r\sum_{x=0}^{\infty}\binom{r+x-1}{x}p^x(1-p)^{r-1} + r\sum_{x=0}^{\infty} xp^x(1-p)^{r-1} \\
& -r(r-1)p\sum_{x=0}^{\infty}\binom{r+x-1}{x}p^x(1-p)^{r-2}.
\end{aligned}$$

Using the definition of the moments $E(x)$ and $E(x^2)$, we get

$$\frac{E(x^2)}{p}-\frac{rE(x)}{1-p}=\frac{r}{1-p}+\frac{rE(x)}{1-p}-\frac{r(r-1)p}{(1-p)^2}$$

So

$$\begin{aligned}
\frac{E(x^2)}{p} &= \frac{r+2rE(x)}{1-p}-\frac{r(r-1)p}{(1-p)^2} \\
&= \frac{r+2rE(x)-(r-1)E(x)}{1-p} \\
&= \frac{r+rE(x)+E(x)}{1-p}
\end{aligned}$$

Then

$$\begin{aligned}
E(x^2) &= \frac{pr+prE(x)+pE(x)}{1-p} \\
&= \frac{pr+\frac{p^2r^2}{1-p}+\frac{p^2r}{1-p}}{1-p} \\
&= \frac{pr-p^2r+p^2r^2+p^2r}{1-p} \\
&= \frac{pr+p^2r^2}{1-p}.
\end{aligned}$$

This shows (8).

Comparison with Other Peak Calling Tools

We evaluated the performance of ASPeak on four different datasets (2 RIP-Seq: (Singh et al., 2012) and (Myers et al., 2011); 2 CLIP-Seq: (Saulière et al., 2010) and (Cho et al., 2012)). In each case, we compared ASPeak to two other peak calling tools: Piranha (Uren et al., 2012) and MACS (Zhang et al., 2008). In all cases, ASPeak gave comparable or better results (see Figure 6).

Since there are currently no gold-standard datasets allowing a rigorous precision-recall or receiver-operating curve based evaluation of the existing approaches, we relied on the specific knowledge about the biology of the protein complex of interest to interpret the "correctness" of peaks. For this, we considered the ratio of the peaks on the Exon Junction Complexes (EJCs) to all of the peaks found by each tool, where higher ratios correspond to better performance. The results are given in Table 4. The ratios are also plotted in Figure 6.

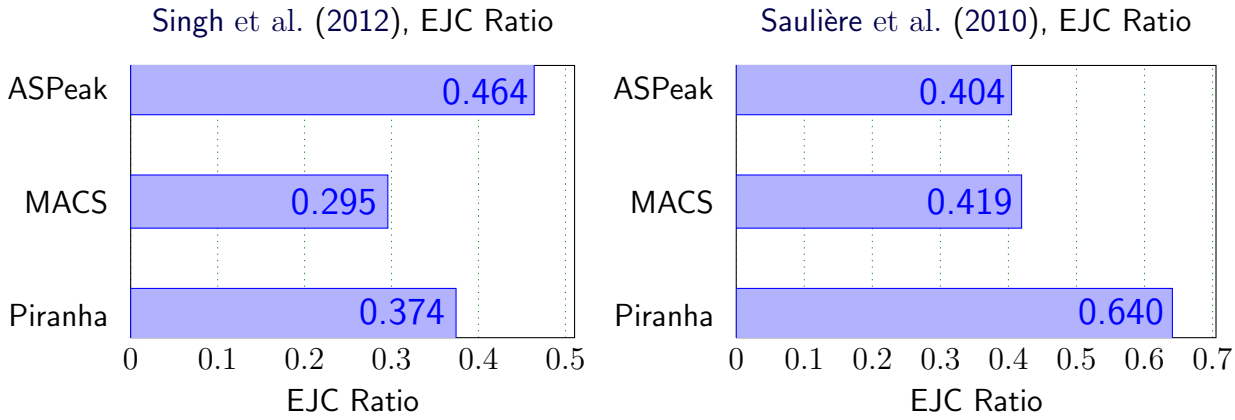


Figure 6: EJC Ratios of the Peak Calling Tools: ASPeak, Piranha and Macs have been run on the data sets used in (Singh et al., 2012) and (Saulière et al., 2010). Each bar represents the ratio of the peaks on the EJC site to all of the peaks called by the individual tool.

	Singh et al. (2012)		**Saulière et al. (2010)**	
	All Peaks	**EJC Peaks**	**All Peaks**	**EJC Peaks**
ASPeak	94999	44091	14931	6041
MACS	264518	78232	8670	3636
Piranha	80838	30251	908	582

Table 4: Number of all peaks and number of peaks on EJCs found by each peak calling tool on the data sets used in (Singh et al., 2012) and (Saulière et al., 2010).

In Table 5, we give the number of found peaks on all of the data sets (two RIP-Seq and two CLIP-Seq) we used. The Venn Diagrams of these peaks are given in Figure 8.

We also compare the three peak calling tools in precision plot in Figure 7.

	RIP		CLIP	
	Singh et al. (2012)	**Myers et al. (2011)***	**Saulière et al. (2010)**	**Cho et al. (2012)**
ASPeak	94999	30381	14931	34082
MACS	264518	90433	8670	8788
Piranha	80838	5714	908	1105

Table 5: Number of found peaks by each peak calling tool on RIP and CLIP data sets.

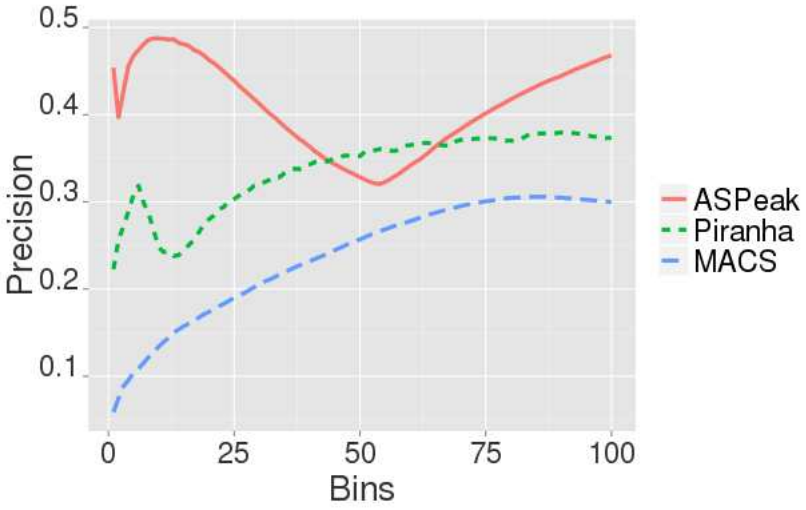


Figure 7: Precision plot: We ranked the called peaks by their program-specific scores from the highest to lowest confidence. Then, we separated each list into 100 equal size bins and calculated the ratio of EJC peaks to all peaks (precision) in each bin.

For an unbiased comparison of the tools ASpeak, MACS and Piranha, we used the same command line parameters whenever possible and interpreted the output as follows.

- The p-value cut off was set to 0.01 in all peak callers.
- Since ASPeak calls the peaks for given annotated regions(intronic, utr5, utr3, cds), called peaks in intergenic regions were removed in other peak callers.
- Since footprint size distribution varies in different algorithms, to asses an overlap in between peaks from different algorithms, we merged found peaks close to each other. In every output, each found peak region was extended 50 nucleotide to either side of the peak. Overlapping peaks in this extended region were merged and considered as the same peak.

Command line options of ASPeak

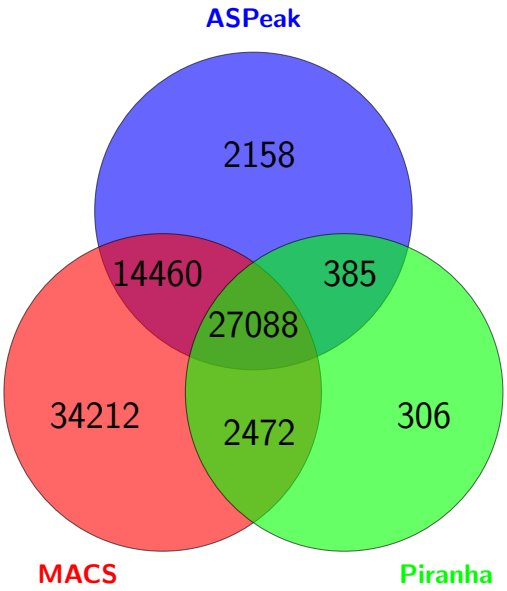
ASPeak was run with BAM and BED alignment files of CLIP- / RIP- Seq data. Default parameters were used.

Command line options of MACS

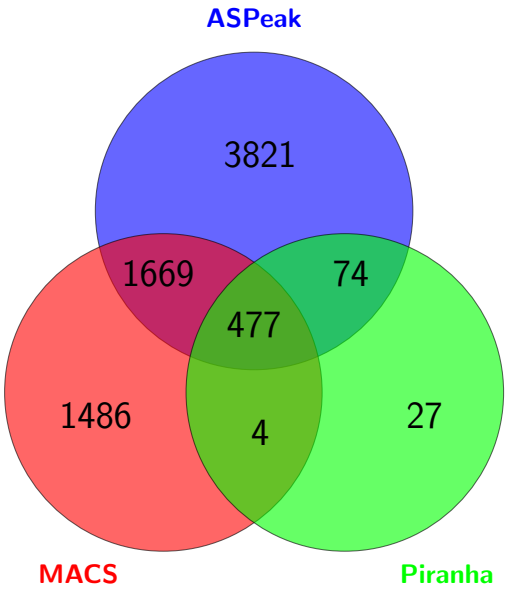
MACS (version 1.4.1) was run with BAM and BED alignment files for CLIP- / RIP- Seq data. The `-nomodel` and `-shiftsize=1` options were set, since we were calling the peaks on CLIP- / RIP- Seq data not CHIP-Seq. These methods don't require shifting of the enriched reads, since CLIP- / RIP- Seq libraries come from single-stranded RNA, whereas CHIP-Seq data comes from double-stranded DNA. The genome size `--gsize` parameter was set to the size of the species. We also used `--call-subpeaks` option to split the peaks to more refined sub peaks for better comparisons. The remaining parameters were set to system defaults.

Command line options of Piranha

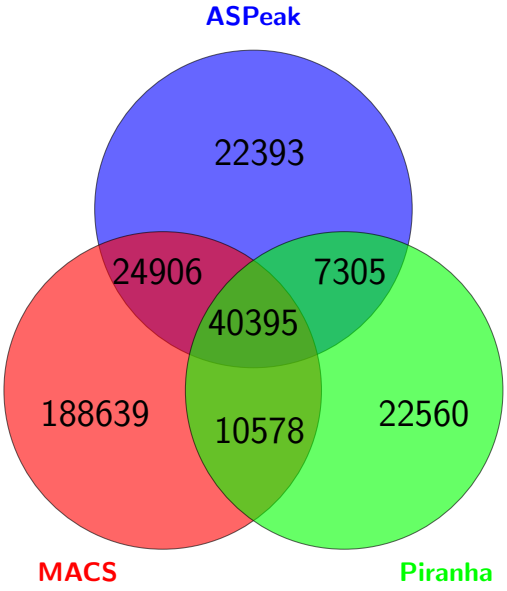
Piranha (version 1.2.0) was run with BED alignment files of CLIP- / RIP- Seq data where the `-b` parameter (bin-size) was set to 5. The remaining parameters were set to system defaults.



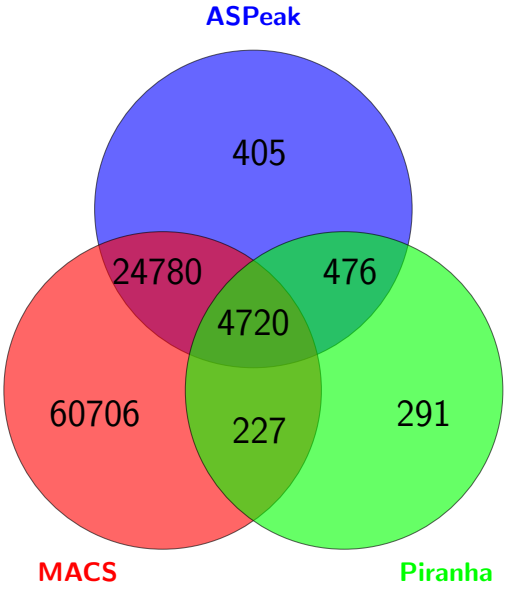
(a) Singh et al. (2012), EJC Peaks



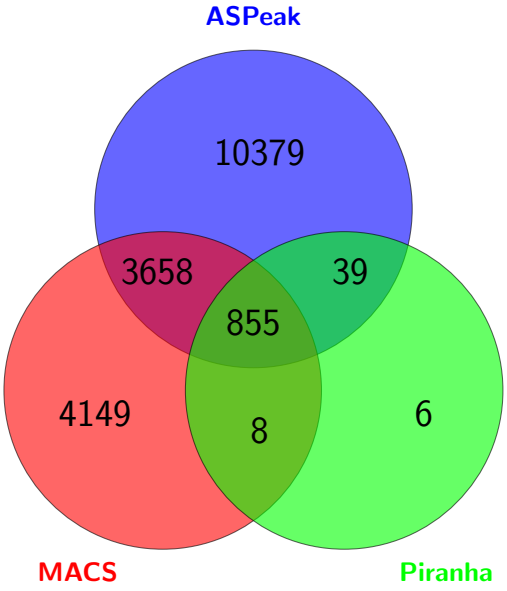
(b) Saulière et al. (2010), EJC peaks



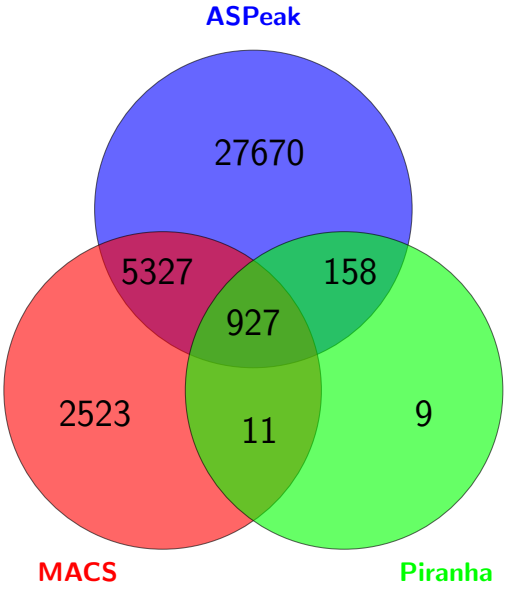
(c) Singh et al. (2012), RIP



(d) Myers et al. (2011), RIP



(e) Saulière et al. (2010), CLIP



(f) Cho et al. (2012), CLIP, Mouse

Figure 8: Venn Diagrams of the peaks called by ASPeak, MACS and Piranha. The diagrams 8b and 8a consist of only EJC peaks. The diagrams 8c-8f consist of all peaks. All genomes, except for 8f, are human genomes and 8f is mouse genome.

References

Cho, J. et al. (2012). LIN28A is a suppressor of ER-associated translation in embryonic stem cells. Cell *151*, 765–777.

Myers, R. M. et al. (2011). A user's guide to the encyclopedia of DNA elements (ENCODE). PLoS Biol. *9*, e1001046.

Saulière, J. et al. (2010). The exon junction complex differentially marks spliced junctions. Nat. Struct. Mol. Biol. *17*, 1269–1271.

Singh, G. et al. (2012). The cellular EJC interactome reveals higher-order mRNP structure and an EJC-SR protein nexus. Cell *151*, 750–764.

Uren, P. J. et al. (2012). Site identification in high-throughput RNA-protein interaction data. Bioinformatics *28*, 3013–3020.

Zhang, Y. et al. (2008). Model-based analysis of ChIP-Seq (MACS). Genome Biol. *9*, R137.