Tn-seq Explorer 1.3

User guide

1. The purpose of Tn-seq Explorer

Tn-seq Explorer allows users to explore and analyze Tn-seq data for prokaryotic (bacterial or archaeal) genomes. It implements two alternative methods for identification of essential genes and provides additional tools to investigate the Tn-seq data. The primary goal of the data analysis is to study fitness by identifying genes that are essential or beneficial under specific growth conditions.

2. What do you need to use Tn-seq Explorer?

2.1 Hardware and software requirements

Tn-seq Explorer is written in Java and should run on any computer with Java installed. We tested Tn-seq Explorer on Windows 7, Mac OS X, and Redhat Linux. Tn-seq Explorer itself does not require extensive resources and runs adequately fast on a standard PC with an Intel i3 processor (but see below regarding the use of Burrows-Wheeler Aligner). Your computer has to be connected to the Internet to download files required for the data analysis.

2.2 Input data

Presumably, you have performed random transposon mutagenesis, prepared mutant libraries, enriched your library for transposon-chromosome junctions, and sent the DNA for sequencing. Using primers matching the end of the transposon DNA, the sequence reads you received should match a short segment of the genomic DNA just downstream of the transposon. You most likely have these sequence reads in a fastq file that you received from the sequencing center. The first step in the data analysis is to align the reads to the genomic DNA sequence. **Note that you need the annotated genome sequence of the organism you study (reference genome) to use Tn-seq.**

Tn-seq Explorer does not perform the alignment but it will guide you through the installation and use of Burrows-Wheeler Aligner (BWA; http://bio-bwa.sourceforge.net/) (Li and Durbin 2009). BWA runs on UNIX-based systems (Linux or Mac OS X) and if you are using a Windows PC to run Tn-seq Explorer you will need access to a Linux or Mac OS X system to perform the alignment. The alignment has to be stored in the SAM format in order to be readable by Tn-seq Explorer.

The BWA tab in Tn-seq Explorer menu will guide you through the use of BWA. If you are using Windows it will help you design the Linux shell commands that you have to perform on a

Linux system where BWA is installed to obtain the SAM file. If you run Tn-seq Explorer on Linux or Mac OS X it will also attempt to download and install BWA on your system if it is not already installed.

3. Brief explanation of the methodology and motivation

The goal of the method is to identify genes that are essential or advantageous for growth under specific conditions in which the mutant libraries were cultured. The basic premise is that mutants with insertions in essential genes would not be viable. Consequently, essential genes would be characterized by a significantly low density of insertions in the Tn-seq data. Unless the number of mutants is very high, a difficulty arises from comparing insertion densities among genes of different lengths. For example, even zero insertions in a short gene may not be a statistically significant deviation from the expected count of insertions if they were distributed randomly. To overcome this issue, we use a sliding window approach, where we compare insertion counts among overlapping DNA segments (typically hundreds bp in length). When the window size is sufficiently large (depending on the average density of insertions in the genome) the distribution of the insertion counts should be bimodal with low values corresponding to window locations overlapping with essential genes or other essential genomic segments (Sarmiento et al. 2013).

Once the appropriate window size is determined and counts of insertions per window are known, each annotated gene is assigned an essentiality index (EI). For a gene that is larger than the window size, the essentiality index is the largest insertion count in any of the windows embedded in that gene. For genes smaller than the window size, EI is the smallest insertion count among all windows that fully encompass the gene at hand. Note that the sliding window approach does not completely remove the uncertainty affecting short genes because a short gene surrounded by non-essential DNA could lead to all windows overlapping with that gene to have high insertion counts. However, the sliding window can detect clusters of essential short genes (e.g., operons), which could not be reliably identified as essential if the genes were considered separately.

Tn-seq Explorer allows for automatic adjustments of the gene starts and ends. This is because transposon insertions near the 3’ end of the gene are less likely to disrupt the gene function; therefore insertions may be found near the gene 3’ ends even if the gene is essential. Adjustment at the 5’ end can be made to account for possibly mis-annotated translation start sites. The default parameters are set to exclude the 20% of the gene length at the 3’ end and 5% at the 5’ end.

See (Sarmiento et al. 2013) and the supplementary information (http://www.pnas.org/content/suppl/2013/03/01/1220225110.DCSupplemental/pnas.201220225SI.pdf) for detailed description of the method and more extensive justification, as well as example of application.

As an alternative to the sliding window approach, Tn-seq Explorer allows users to assess directly the insertion density in each gene (i.e., the number of insertions within a gene divided by the gene length) (Curtis and Brun 2014; Langridge et al. 2009). Like the insertion counts per window position in the sliding window approach, the insertion density per gene exhibits binomial distribution where essential genes from a separate peak with low insertion densities. A drawback of this approach is that it compares insertion densities in genes of different sizes and the same value of insertion density may not carry the same statistical significance. However, this approach has produced good results in analyses of mutant libraries at high level of saturation.

4. Using Tn-seq Explorer

4.1 Program installation

There is no installation. Simply copy the file Tn-Seq_Explorer.jar to a folder of your choice (we recommend creating a separate folder for Tn-seq Explorer because it will use it to store additional files) and double-click the file icon to start the program.

4.2 Creating a new project

After agreeing to the conditions for using the software you will be asked to select a project. If you want to continue a previously created project, select it in the menu on the left and click ‘ok’. Otherwise click ‘New’, provide the name of the new project, and click ‘ok’. Next, select a folder where you want the project stored and click ‘Save’. A new folder with the name of your project will be created in the folder you select and all files related to the project will be stored in this new folder. By default, the new project folder will be created in the same folder as Tn-seq_Explorer.jar. If you subsequently move the folder the program will not be able to find your project unless you find the project folder using the ‘Browse’ button.

4.3 Preparing a gene annotation file

After selecting a project, a new window will appear. This is the main menu of Tn-seq Explorer. At this point only the ‘Main’ tab will be accessible. In this tab, you have to provide the chromosome length and the annotation data (i.e., gene coordinates and functional description if available). There are three ways to supply the annotation to the program:

a) Download annotation from NCBI FTP server. If the genome you are analyzing is on the NCBI FTP server (ftp://ftp.ncbi.nih.gov/genomes/Bacteria/) select this option, find the genome and the scaffold you want in the list (some genomes include multiple chromosomes or plasmids which are stored as separate scaffolds in different files), and click download. This will download the .ptt and .rnt files, which contain the protein and RNA gene annotations, respectively.

b) Use previously downloaded files from NCBI. Use this option if you already have the .ptt and .rnt files on your hard drive. This can also be useful if you want to edit the annotation to include additional genes or remove some from the list. You can manually edit the .ptt and .rnt files as long as you follow the format of the file.

c) Use previously downloaded files from IMG. This option is provided as an alternative for those who prefer using annotation downloaded from IMG (http://img.jgi.doe.gov/).

Once the files are ready, click ‘Prepare gene file’ (unless you used the option ‘a’ above, in which case the gene file is already created) and you are ready to continue to the next step. All menu tabs should now be accessible.

4.4 BWA

If you have not yet obtained the SAM files from BWA (or a similar program), click the BWA tab and follow the instructions. If you do not know Linux you may need help from someone who does. We embedded the BWA feature in Tn-seq Explorer for the user’s convenience but it may not work under all circumstances depending on the configuration of your system. If you experience difficulties with BWA please refer to the information on BWA website (http://bio-bwa.sourceforge.net/) and perform this part of the analysis outside of Tn-seq Explorer.

4.5 Manage libraries

By ‘library’ we are referring to a set of genomic sequencing reads (short fragments of genomic DNA) that are adjacent to the 3’ end of a transposon and aligned to a reference genome. The DNA reads are obtained by high-throughput sequencing of specific mutant strains (i.e., strains with transposon insertions) grown under a particular set of conditions. For the purposes of Tn-seq Explorer, each library is represented by a SAM file (sequence reads from the mutant library aligned to the genome).[1] To add a new library to the project, click the ‘Browse’ button and navigate to the SAM file, then click ‘Extract’. The following process may take several minutes depending on the size of the SAM file and speed of your computer. When it is finished, you will be asked to provide the name for the new library. The program has now created four new files in the project folder named <Library_name>.inspo, <Library_name>.inspos, <Library_name>.inspou, and <Library_name>.inspous, which contain, respectively, chromosomal locations for every read successfully aligned to the genome (negative coordinates refer to insertions in the ‘-‘ orientation), the same data sorted by chromosomal location, list of unique insertions (including chromosomal location, orientation, and number of reads that correspond to this insertion) sorted by location, and the same data sorted by number of reads. We recommend that you do not edit these files but you may inspect them using Excel or a text editor if you wish. Especially the <Library_name>.inspou and <Library_name>.inspous files provide valuable diagnostic information to detect potential anomalies in distribution of reads

[1] Alternatively, if you have a library file that cannot be converted into the SAM file format you can create the .inspo file manually or by your own script and then feed this file into the program and it will create other files required for data processing. The .inspo file is a plain text file that contains the locations of transposon insertions in the genome for each sequence read, one number per line, with negative values corresponding to insertions in the ‘minus’ orientation (complementary strand).

among unique insertions that could be indicative of biases or other potential problems in the data.

When preparing the data libraries, you may choose to eliminate insertions that are represented by a low number of sequence reads and the associated reads from the data. This feature is motivated by the concern that insertions represented by a single read (or a few reads) could result from reads incorrectly mapped to the genome. By default, this parameter is set to zero, which means that all insertions and all mapped reads are counted.

When you have prepared a library you can click 'Clear the form' and continue to prepare additional libraries.

4.6 Determining the appropriate window size

Tools to determine the appropriate window size are located at the bottom of the 'Manage libraries' tab. This is because the optimal window size depends on the density of insertions and could differ for different libraries and different experiments. The 'Recommend optimal window size' button will attempt to determine the appropriate window size automatically (see the Appendix below for details). You can also use the 'Plot' button to explore the distribution of the unique insertions or reads per window [analogous to Figure 1 in (Sarmiento et al. 2013)]. To do that, select a library, a window length (default 1000 or a previously determined optimal window size) and a step (default 100; determines the number of nucleotide by which the window is shifted each time; that is, window size 1000 and step 100 means that the adjacent widows overlap by 900 bp). Ideally, the resulting plot will have a peak at the left with maximum at or near zero insertions, followed by a shallow valley and a wider peak to the right. Windows comprising the left peak correspond to portions of the genome with a low density of insertions that likely overlap with essential genes whereas the windows of the wider peak and right tail correspond to portions of the genome with a high amount of insertions that are likely nonessential. The number of insertions/reads per window that coincides with the valley between the peaks can be used to determine the essentiality index cutoff for classifying genes as essential, non-essential, or uncertain classification. The best way to determine the most appropriate window size is to use the automatic recommendation (the 'Recommend optimal window size' button) and then manually explore the distribution for slightly smaller or larger windows to decide which one is most appropriate for your particular situation. Larger window sizes will provide more accurate distinction between long essential and nonessential segments but at the cost of being unable to detect essential segments shorter than the window size. The window size determined in this step is stored as a default for the given library but it can be changed in the subsequent data analysis. You can also return to this step at any time and change the default setting.

Once the plot is made you can adjust the ranges of the x and y axes and click 'Re-plot'. The tabulated data is also stored in the project folder in a tab-delimited format (suitable for opening in Excel).

The 'Find essential regions' button at the bottom of the screen creates a table with all chromosomal segments that consist of overlapping windows containing no more than the specified number of insertions (or reads). These “essential” segments are listed in the format “<start>..<end>”, where <start> and <end> refer to the location of the segment in the DNA sequence.

4.7 A caveat concerning small genes

Small essential genes (smaller than the window size) surrounded by nonessential segments may be incorrectly classified as nonessential because all windows overlapping with the gene at hand also overlap with the nonessential segments and therefore can contain high number of transposon insertions. This issue is inherent to using an unsaturated mutant library and generally cannot be resolved by using a smaller window size or counting only insertions within the gene.

4.8 Unique insertions vs. reads

In this and subsequent analyses, you can choose to count the unique insertions per window or sequence reads per window. Unique insertions are defined by unique location or orientation. Excluding errors in the alignment, each unique insertion should represent a unique mutant. In contrast, multiple reads can be obtained from a single mutant. We used unique insertions in our work because of concerns that significant biases may exist in the sequencing, resulting in some mutants more extensively represented among the sequence reads than others and, for the purpose of statistical evaluations, sequence reads may not represent independent observations, whereas unique insertions are more likely to be independent.

The button “Distribution of reads per unique insertion” allows you to investigate how non-random the distribution of sequence read counts per unique insertion is. If each mutant had an equal chance to be sequenced, one would expect this distribution to be approximately normal (or more accurately binomial with n equal to the number of sequence reads and p reciprocal of the number of unique insertions – but the binomial distribution approaches the normal distribution for high n). However, in the data we analyzed, the distribution of reads per unique insertion is more similar to a power law distribution. While there could be biological reasons for the power law distribution (e.g., presence of genes whose deletion might boost fitness under the specific growth conditions), a power law distribution with a long tail could possibly be indicative of experimental artifacts. (note: you may need to change the range of the axes in the plot to see the most relevant part of the distribution when using this feature.) Users who wish to investigate possible biases in the sequence reads distribution in more detail can use the files .inspou and .inspous generated by the Tn-seq Explorer; these are tab-delimited text files stored in the project folder and include for every unique insertion its position in the genome, orientation, and the number of sequence reads mapped to that insertion.

4.9 Manage data tables

This is where you perform the actual data analysis once the libraries are prepared. The ‘Create new’ button creates a new spreadsheet (stored in tab-delimited format, which can be opened in Excel). The spreadsheet is stored in the project folder under the name <table_name>.table.xls. When created, the spreadsheet includes the gene information obtained from the annotation. You can click ‘Open as spreadsheet’ to view the table (you need Microsoft Office or Open Office for this to work[2]). If you edit the file make sure that you do not disrupt the format and that you save it as a tab-delimited text file; if you want to format the table for use outside of Tn-seq Explorer consider making a copy and leaving the original file intact. When you close the spreadsheet click ‘Replace’ to accept the changes or ‘Cancel’ to discard them.

‘Add new data to the table’ opens a new window with several options for data analysis. Each adds a new column to the spreadsheet. In the ‘Add new essentiality indices’ tab, you can calculate essentiality indices for all genes in the table and add them as a new column. You can repeat this step for different libraries and using different parameters (window size, step, gene start and end adjustments), and each time a new column is added. To remove unwanted data open the spreadsheet, delete the columns you do not want, and then save it. The calculation of essentiality indices may take a few minutes.

‘Add insertion densities’ adds a column to the table that contains densities of unique insertions within each gene or density of sequence reads mapped to each gene. The start and end locations of each gene can be adjusted in the same way as in the sliding window approach. You can subsequently plot the distribution of insertion densities per gene to verify that the distribution is bimodal (with essential genes forming a separate peak on the left) and identify the most appropriate cutoff value to classify genes as essential and non-essential. ‘Add insertion counts’ provides the simple count of unique insertions or reads in each gene (again allowing for start and end adjustments).

Two tools are provided to compare the gene essentiality between libraries and help users to identify candidate genes that may be essential in one mutant library but not another. These tools are found under the ‘Compare data’ tab. The ‘Compare’ button adds a new column in the data table which shows differences between values in two previously created columns. Sorting the genes in the output spreadsheet by this column brings genes that are probably essential in library A but not in library B to the top of the list, whereas genes that might be essential in library B but not A are at the bottom. Although this is a very simple feature, we found it both useful and convenient in identifying genes of interest. In our experience, this feature is applicable even if the parameters used to analyze the two libraries (e.g., the window size) are not the same. The ‘Maximum insertions’ parameter caps the essentiality indices at the specified value. This is useful to filter out large differences in EI values when the gene is non-essential (has high EI) in both libraries. Note that the ‘Maximum insertions’ parameter applies only to the ‘Compare’ feature and is ignored by the ‘Plot’ feature described below.

The ‘Plot’ button plots all genes using coordinates from two selected columns in the data table. The result is an interactive plot where the user can select any point in the graph and the program displays the description of the gene represented by that point (caveat: if more than one gene have the same coordinates only one gene is displayed; this can be partially overcome by using the ‘Randomize data’ feature described below). Genes with different essentiality in the two compared libraries are represented by points that are distant from the main diagonal. The ‘Plot’

[2] More accurately, the file opens with a default program designated to open .xls files. For example, you can set the .xls files to be open with text editor if you wish or if you do not have Excel or Open Office.

feature can also be used to explore robustness of the results with respect to changing parameters (e.g., gene start and end adjustments or window sizes in the sliding window approach) or methodology (e.g., essentiality indices vs. insertion density or counting all sequence reads vs. only unique insertions) and to identify genes whose classification is most affected by changing parameters or using different methodology. The ‘Randomize data’ feature adds 1 plus a small random number (between -0.25 and 0.25) to each value before it is plotted. This can be helpful when many data points have the same coordinates and would all be displayed as a single point in the plot. Note that the ‘Randomize data’ feature is intended for comparison of essentiality indices or insertion counts that have discrete values and should not be used when comparing insertion densities.

5. How to cite Tn-seq Explorer

The original description of the sliding window approach and application to *M. maripaludis*:

> Sarmiento F, Mrázek J, Whitman WB (2013) Genome-scale analysis of gene function in the hydrogenotrophic methanogenic archaeon *Methanococcus maripaludis*. Proc. Natl. Acad. Sci. U.S.A. 110:4726-31

Tn-seq Explorer software:

> Solaimanpour, S., Sarmiento, F. Mrázek J. (submitted)

6. Credits and contacts

Jan Mrázek (mrazek@uga.edu)

Designed the methodology, wrote the initial code in C, and supervised the development of the final software.

Sina Solaimanpour (sina@uga.edu)

Converted the code to Java and added a number of new features.

Felipe Sarmiento (felipes@uga.edu)

Supplied the experimental data, extensively tested the software, and provided numerous suggestions in its design

Barny Whitman (whitman@uga.edu)

Posed the problem and provided helpful suggestions in developing the methodology and software.

7. Funding

Development of Tn-seq Explorer was supported by the grant BDI-0950266 from the National Science Foundation

REFERENCES

Curtis PD, Brun YV (2014) Identification of essential alphaproteobacterial genes reveals operational variability in conserved developmental and cell cycle systems. Mol Microbiol 93:713-35

Langridge GC, Phan MD, Turner DJ, Perkins TT, Parts L, Haase J, Charles I, Maskell DJ, Peters SE, Dougan G, Wain J, Parkhill J, Turner AK (2009) Simultaneous assay of every *Salmonella Typhi* gene using one million transposon mutants. Genome Res 19:2308-16

Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25:1754-60

Sarmiento F, Mrázek J, Whitman WB (2013) Genome-scale analysis of gene function in the hydrogenotrophic methanogenic archaeon *Methanococcus maripaludis*. Proc Natl Acad Sci U S A 110:4726-31

8. Appendix: Determining the optimal window size for the sliding window method

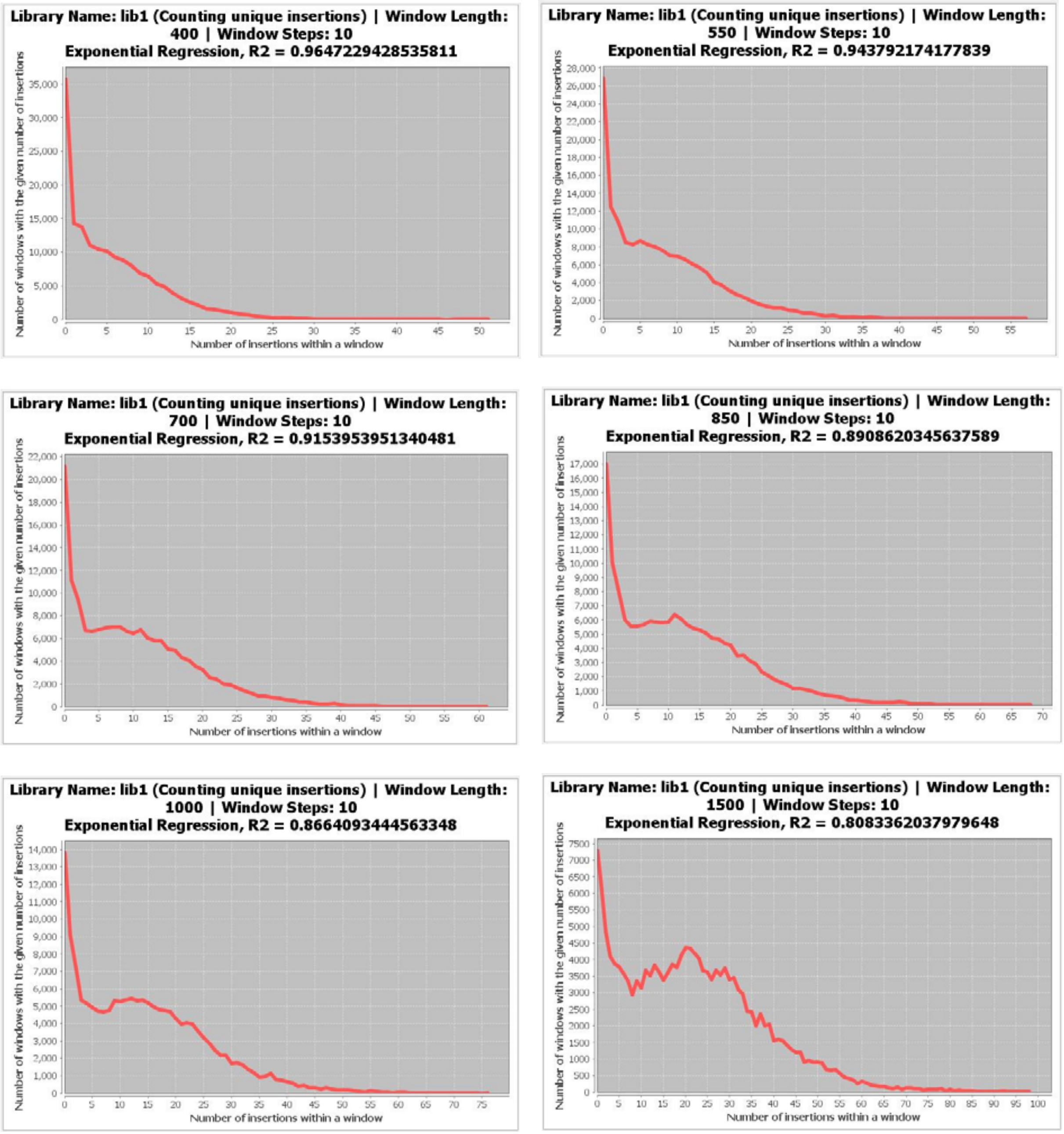
8.1 Background and motivation

We are operating with the premise that the analyzed DNA contains two populations of segments: essential (or strongly beneficial) and non-essential. Transposon insertions can be detected randomly in each population of segments but with a substantially lower rate of insertions in essential segments. Consequently, when counting the number of insertions in windows of a fixed size, these counts should follow a bimodal distribution reflecting the two populations of windows (essential and non-essential). However, the two populations cannot be separated when the window size is too small. Therefore the optimal window size would reflect a compromise between reliably classifying small genes as essential or non-essential (this favors small window size) and reliably differentiating between the two populations of windows (essential and non-essential). In practice, this means finding the least window size that clearly separates the two populations, that is, where the distribution of insertion counts per window is clearly bimodal. In the example in Figure 1, the window sizes 700 or 850 bp appear appropriate while 550 bp is too small to differentiate between essential and non-essential windows and 1000 bp window might be unnecessarily conservative. However, you may choose larger window size to emphasize accuracy of distinction between essential and non-essential windows at the cost of possibly missing small essential genes, or smaller window size to compromise on the other end of the spectrum.

8.2 Automatic recommendation of appropriate window size

Clicking the “Recommend optimal window size” button in Step 2 will attempt to determine automatically the appropriate window size. We tested several procedures to detect the optimal window size (including standard tests for bimodality), with the best results provided by a heuristic method based on exponential regression. This approach was motivated by the observation that for window sizes that are too small, the empirical distribution of insertion counts can be reasonably approximated by exponential function (excluding the right-hand tail). Our algorithm therefore starts with a small window size, performs exponential regression of the distribution of insertion counts, assesses the goodness of fit by the coefficient of determination R^2, and repeats the process with gradually increasing window size (in the increments of 50 bp) until the R^2 drops below a certain cutoff K, which was determined empirically by analyzing several libraries of transposon insertions. The smallest window that yields $R^2<K$ is the recommended optimal window size.

The cutoff K is 0.80 for window sizes 400 bp or smaller, 0.92 for window sizes 1400 bp or larger, and increases linearly between 400 and 1400 bp. Using more stringent cutoff for smaller window sizes is motivated by the reasoning that few genes would be smaller than the window, therefore it is reasonable to increase the emphasis on differentiating accurately essential and non-essential windows. On the other hand, essential genes smaller than the window can be incorrectly classified as non-essential and with increasing window size it is reasonable to shift the emphasis on keeping the window small even at the cost of less reliable distinction between essential and non-essential windows.



Library Name: lib1 (Counting unique insertions) | Window Length: 400 | Window Steps: 10
Exponential Regression, R2 = 0.9647229428535811

Library Name: lib1 (Counting unique insertions) | Window Length: 550 | Window Steps: 10
Exponential Regression, R2 = 0.943792174177839

Library Name: lib1 (Counting unique insertions) | Window Length: 700 | Window Steps: 10
Exponential Regression, R2 = 0.9153953951340481

Library Name: lib1 (Counting unique insertions) | Window Length: 850 | Window Steps: 10
Exponential Regression, R2 = 0.8908620345637589

Library Name: lib1 (Counting unique insertions) | Window Length: 1000 | Window Steps: 10
Exponential Regression, R2 = 0.8664093444563348

Library Name: lib1 (Counting unique insertions) | Window Length: 1500 | Window Steps: 10
Exponential Regression, R2 = 0.8083362037979648

Figure 1. Screenshots of distribution of unique insertions per window for window sizes 400, 550, 700, 850, 1000, and 1500 bp. The analyzed genome (*M. maripaludis* S2) was scanned with a sliding window of the given size shifted by 10 bp in each step, counting the number of unique transposon insertions within each window. The vertical axis shows the number of window positions that yield the insertion count shown by the horizontal axis.