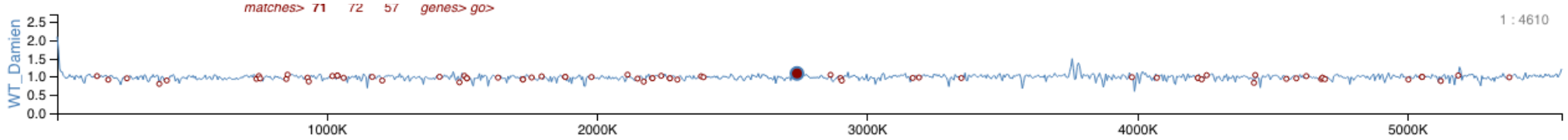
**Nucleosee** is a pattern-search oriented genome browser

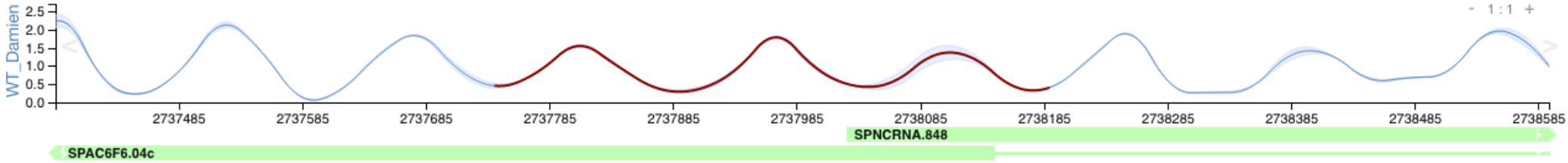
## OVERVIEW

Nucleosee has *three levels* which unfold with the pattern search narrative

**1) Chromosome level**: whole single chromosomes for overview and search context




**2) Gene level**: single search matches are represented on a 1:1 pixel-nucleotide scale base



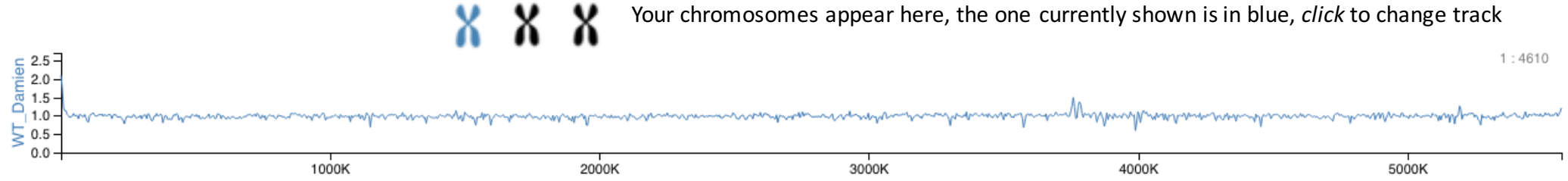

**3) Nucleotide level**: current browsing on level 2 and other search matches are represented at this level


Typical usage goes like this:

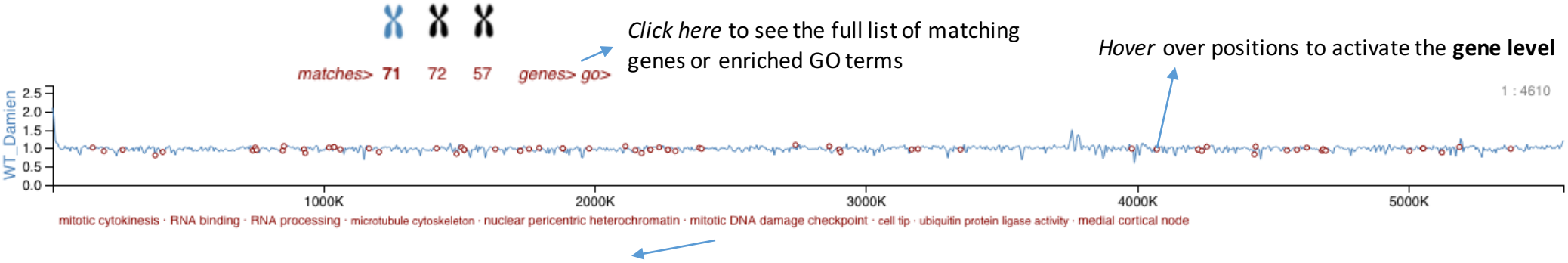
## 1) CHROMOSOME LEVEL

This level shows whole tracks (usually chromosomes) as defined in the wig files
The scale is therefore large, as it only provides the context for searches

To load this level, **select data** already preprocessed (see **preprocessing data**) at 


Once you perform a **search** (see below), the number of matches on each chromosome, as well as the positions in the current chromosome are highlighted in red.
You can also search by gene name, by GO term (with the prefix `go:`) or by interval (`a-b`)




If there are enriched GO terms (FDR<$10^{-3}$), they are shown here (the larger font the more enriched)
*Hover* them to see the p-value and the number of annotated genes matching the pattern respect to the total number of annotated genes
*Click* on them to highlight the annotated genes in the chromosome track.

## 2) GENE LEVEL

This level shows a single match on the current search.

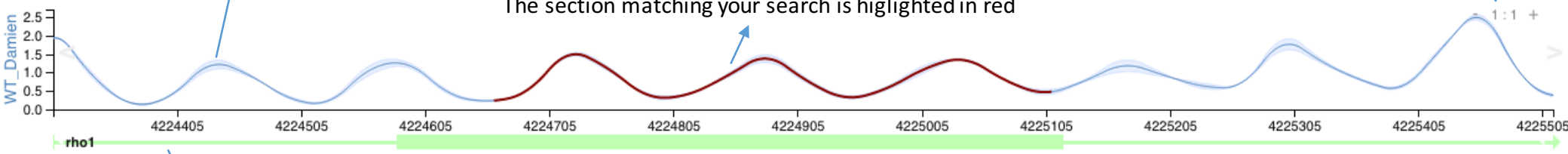
It has a 1:1 scale, so each pixel usually represents only one or a few nucleotides:

If you are using several batched data (see preprocessing data), the variance between replicates will be shown as a shadow below the average line

*Click* on + - symbols to zoom in or out.

*Click* on the side arrows to slide through the genome

The section matching your search is higlighted in red




Gene annotations are shown here. The wide part corresponds to CDS. The arrow marks the sense. *Hover* over the name to see gene details

*Click* on a gene name to zoom fit to its length.

*Hovering* over the line, the nucleotide level is activated

## 3) NUCLEOTIDE LEVEL

This level shows the actual sequence of the hovered interval at gene level

It also shows the sequence of some of the remaining search matches at the same interval

In **bold**, the most frequent motif on all the matches

The last line shows the consensus motif, letters are darker if the consensus is high, and are underlined if it is above 90%

**SEARCH**

Nucleosee uses BWT to index .wig or .bw data and then perform quick searches.
During preprocessing, numerical levels are split into *windows* and then discretized into *percentile* bins. Each of these bins is represented by a letter (`a`, `b`, `c`, etc.)

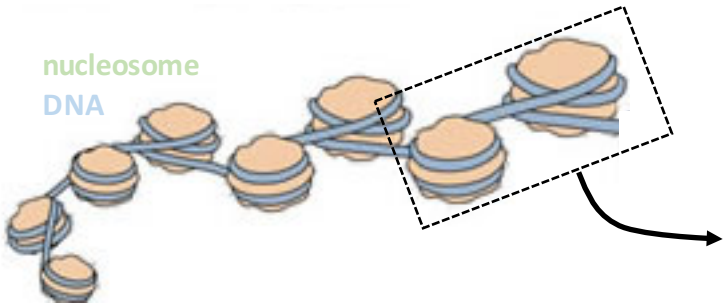
*Example 1*:

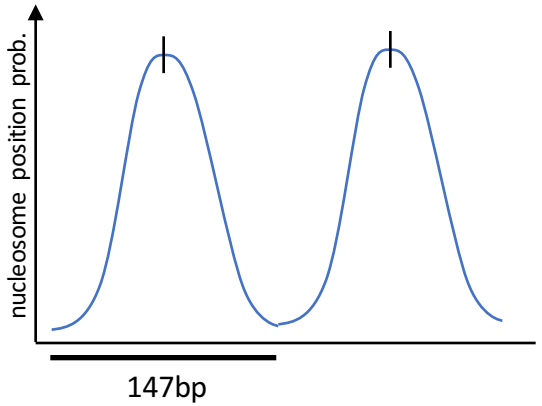
Let be the 10-nucleotide sequence of abundance levels: `1 4 8 9 9 7 6 5 3 0`
A 2-nucleotide window will average it as: `2.5 8.5 8.0 5.5 1.5`
A 3-bin discretization will do: `a c c b a`

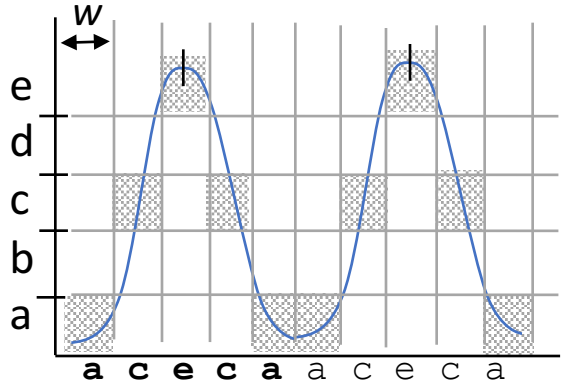
`a` represents a window value between percentile 0% and 33%, `b` between 33% and 66% and so on.
*(with 3 levels, it can be easily memorized as below, around and above average)*

*Example 2*:

For S pombe nucleosome maps, a good window choice can be *w*=30 bp and *d*=5 percentiles
This way, a perfectly positioned nucleosome will be represented as `aceca`, and NDRs as `poly-a`
Or you could simplify the model to *w*=50 and 3 percentiles to characterize nucleosomes as `aca`



## SEARCH OPTIONS


1 **Pattern definition** may include any combination of bin letters and operators `+` and `*`, por example `abcba*3` or `a*5+abcba`. You can also search for gene names, go terms (with the prefix `go:`) or intervals (`start-end`)

2 **Pattern combination:** in the case of several loaded data sets, we can combine searches of different patterns on them, with different join actions (and, or, not)

3 **Mutations:** number of allowed 1-letter 'violations' from the pattern. As in BWT alignment, a high number of mutations severely affects performance
*Soft* mutations are changes to a *contiguous* character (`a` to `b` but not `a` to `e`)

4 **Restriction** to especific annotations (genes, UTRs, intergenic regions, etc.) can be applied.
*Fully inside* restrictions imply that the pattern must fall totally inside the given annotation.

5 **Draw grid** shows the percentile thresholds that separate bins

6 **Motif size** determines the length for motif searches on the sequences at the locations of the matching patterns

# Instructions for admins

Setting up the server
Setting up the client
Preprocessing data
Installing annotations

**SETTING UP NUCLEOSEE**

Nucleosee is comprised of two parts to be installed in the same or different machines:

**1) Server**

Nucleosee server (also called seqview, folder `py_server`) is written in python 2.7 and requires the following non-standard libraries:

- `flask` for web service support
- `numpy` for numerical analysis
- `fisher` for statistical enrichment (https://pypi.python.org/pypi/fisher/)

It can be run by simply using `python analysis.py &` at `py_server` although it is not recommended except for tests (bad performance, security issues, no concurrency)
For production it is recommended ***to install it in apache*** *(see below)*

**2) Client**

Seqview client (folder `client`) is written in html/css+javascript. It requires libraries `bootstrap`, `d3` and `jquery`, but are all self-included (folder `svc/js/libs`)

Just make sure the `client` folder is in a `public_html` path to make it accessible
NOTE: you will see an 'Internal Server Error' when accessing it if the server is not set up

# 1) SETTING UP SEQVIEW SERVER IN APACHE

**0) copy seqview folder** to your server. It should contain folders `py_server` and `client` and files `seqview.conf` and `seqview.wsgi`

**1) seqview.conf**: move this file to `/etc/apache2/sites-available`

- You must create a `seqview_user` user on your system, or change the .conf file to your desired user. It's advisable (more secure) to use a specific user for Nucleosee.
- The file has some lines you might need to depending on your configuration:
  - `Listen xxxx`: in case the server is not set to promiscuous listening, you need to set the server to listen to the port for seqview (default 2750)
    - This is the same port that must appear at `<VirtualHost *:xxxx>`
  - `Options FollowSymLinks`: in case that you put a symbolic link as path to your .wsgi file in `WSGIScriptAlias`
  - `Require all granted` can substitute `Order deny, allow` and `Allow from all` in Apache 2.4

**2) seqview.wsgi:** move this file to the path you set in `WSGIScriptAlias` in seqview.conf

- Configure `sys.path.insert` to the path of your seqview root folder

*Now you should restart apache and test in a web browser if you can access `http://yourhostname:xxxx/test` and get as response `Seqview server correclty configured`*

**3) annotations and genomes**: seqview server is run from the home folder of the user set in `WSGIDaemonProcess` in seqview.conf

- If it is a different path from the one where you installed seqview, move `py_server/annotations` and `py_server/genomes` folders to that home folder
- For a clean data list, remove all the `.pic` files in `py_server/genomes` and clear `tracks.txt`

For more information about setting up a Flask server into Apache please read http://flask.pocoo.org/docs/0.11/deploying/mod_wsgi/

## 2) SETTING UP SEQVIEW CLIENT IN A SERVER

You need to configure index.html at line 558 so line:

```
Server.connect(true, verifyConnection, "", "", "http://127.0.0.1:2750/");
```

directs to your previously configured server.

Now you can run the client from your local machine just by clicking on `client/index.html`

Or you can also publish the `client` folder on a web server by moving it to a `public_html` path. It can be the same machine that the server or a different one.

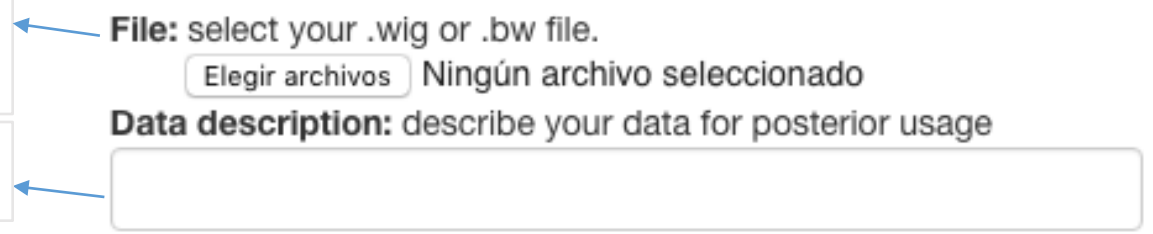
Load file

# PREPROCESSING DATA

A small link at the top-right of the browser allows the user to load and index .wig or .bw files
It's recommended to be done by the person that will oversee the server

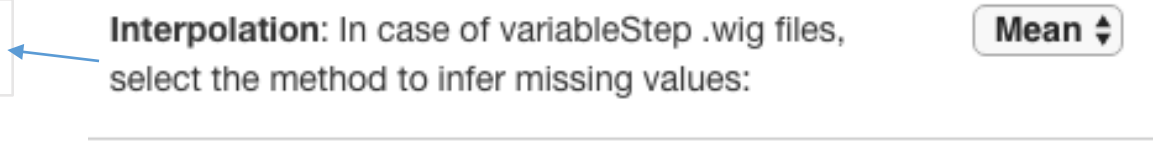
You can select one or more files, providing they have the same track names and sizes
They will be batched together and averaged for visualization

This is the description by which your processed data will be presented to the users when loading data

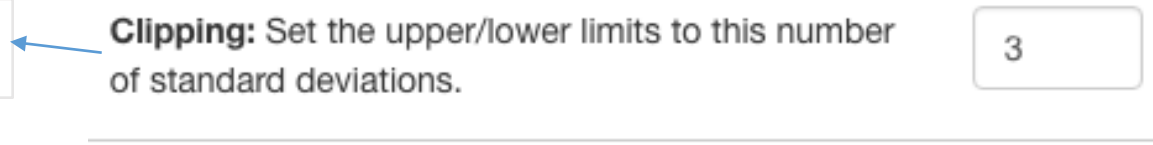



You must select an organism for annotations, enrichment and sequences. You must have previously loaded the organism annotations as explained in **Installing Annotations**

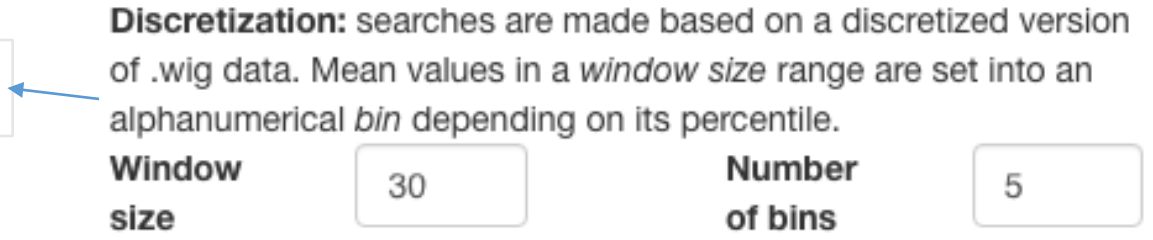

.wig files might have variable steps. In such a case, you can choose a way to interpolate missing values




To deal with outliers, you can clip data to remove values above/below a given number of standard deviations




Window size and number of bins refer to the indexing done for searches. See **Searching** for more infomation



**INSTALLING ANNOTATIONS**

In order to add or update annotations for an organism, you must make a new folder in `annotations` (see 'setting up seqview server', section 3) with a name which can be identified with the organism (it's the name that will appear in the *organism* section when **Preprocessing data**).

The folder should have three subfolders:

- `gff`: must contain a single gff-format file with the gene annotations for the organism. If it is missing, no annotations will be available at the gene level.
- `goa`: must contain a single gaf-format file with the GO annotations for the file. If it is missing, no functional enrichment will be available.
- `fasta`: must contain the fasta or fsa files with the sequences. There should be one file per chromosome, named with the chromosome name. If it is missing, the nucleotide level will not be available

Any or all of the folders can be missing, and you can select no organism when preprocessing data, but then the corresponding functionalities will not be available