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

**OVERVIEW**

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

1) **Genome level**: overall display and pattern matches per chromosome, plus search results download

![matches>58 62 45 genes>go>fasta>](image)

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

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

Typical usage goes like this:

- Load data
- Search
- Browse
- Genome level
- Chromosome level
- Gene level
1) Genome level

This level shows the whole dataset as an array of chromosome icons.

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)

Click here to see the full list of matching genes, enriched GO terms or sequences

You can try a working Nucleosee server at http://cpg3.der.usal.es/nucleosee

Or install your own (see install your server below)
2) Chromosome level

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

Once you perform a search (see below), the matching positions in the current chromosome are highlighted in red. You can also search by gene name or GO term (using the go: prefix).

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.

If you load several datasets, they will appear superimposed in this track, in different colors.
3) 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 data with several replicates (see preprocessing data), the variance between replicates will be shown as a shadow below the average line. Hovering over the line, you can check the exact position and coverage. Click on + - symbols to zoom in or out. Click on the side arrows to slide through the genome.

The section matching your search is highlighted in red. Hovering over the line, you can check the exact position and coverage.

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.

If you load several datasets, the gene level will show a separate track for each of them.
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 \( d=3 \) to characterize nucleosomes as aca.
1 **Pattern definition** may include any combination of bin letters and operators + and *, for example abcba*3 or a*5+abcba. You can also search for gene names or GO terms (with the go: prefix).

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 ‘variations’ 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 specific 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.
Further instructions

Data preprocessing

Setting up a server
DATA PREPROCESSING

A 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 Docker Run Parameters.
- .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.

File:
- select your .wig or .bw file.

Data description:
- describe your data for posterior usage.

Organism:
- select your .wig organism or 'None' if unavailable.

Interpolation:
- In case of variableStep .wig files, select the method to infer missing values:
  - Mean

Clipping:
- Set the upper/lower limits to this number of standard deviations.
  - 3

Discretization:
- searches are made based on a discretized version of .wig data. Mean values in a window size range are set into an alphanumerical bin depending on its percentile.
  - Window size: 30
  - Number of bins: 5

See Search for more information.
SETTING UP NUCLEOSEE
A running Nucleosee server is available for tests at http://cpg3.der.usal.es/nucleosee
Nucleosee is developed as a Docker container for easy server setup at custom locations.

1) Install Docker
Visit https://docs.docker.com/install/ to install Docker on your machine.

2) Setup host folders
Download the annotations folder at http://vis.usal.es/rodrigo/nucleosee/annotations.zip
Unzip it at your preferred location (ann_path). You can check its folder structure and add your own organism annotations.

Optionally, you can download some preprocessed examples at
http://vis.usal.es/rodrigo/nucleosee/genomes.zip
Unzip it at your preferred location (gen_path)

3) Run Docker container
docker run -it --rm -p 80:80 -v ann_path:/app/annotations -v gen_path:/app/genomes -e SERVERNAME=hostname efialto/nucleosee
DOCKER RUN PARAMETERS

1) **hostname** is the name of your machine. For example `signus.unas.uk`
2) **ann_path** is the location where you unzipped the example folder in step 2 above, or any other folder you had with the proper structure (see below)
3) **gen_path** is the location where preprocessed data will be stored, along with a file with all your preprocessed data details (`tracks.txt`)

These two last folders (called *volumes* in Docker) will be modified by Nucleosee container itself, and cannot be erased as usual. You should use `docker volume ls` to see which ones have you defined and `docker volume rm volume_name` to delete.

The annotation folder must have the following structure:

```
ann_path
    ├── go
    │    ├── go.obo
    │    └── species_1
    │         ├── fasta
    │         ├── gff
    │         └── goa
    ├── species_n
    └── ...
```

*This OBO file contains the generic GO term details and can be downloaded from [http://geneontology.org](http://geneontology.org)*

You can name each species as you wish, and populate or not each of the three subfolders with single files for:

- Genome sequences in fasta format.
- Gene annotations in gff format.
- GO annotations in gaf format.

*Make sure that gff, fasta and wig files use the same chromosome names. If any file is missing, preprocessing won’t use the corresponding information.*