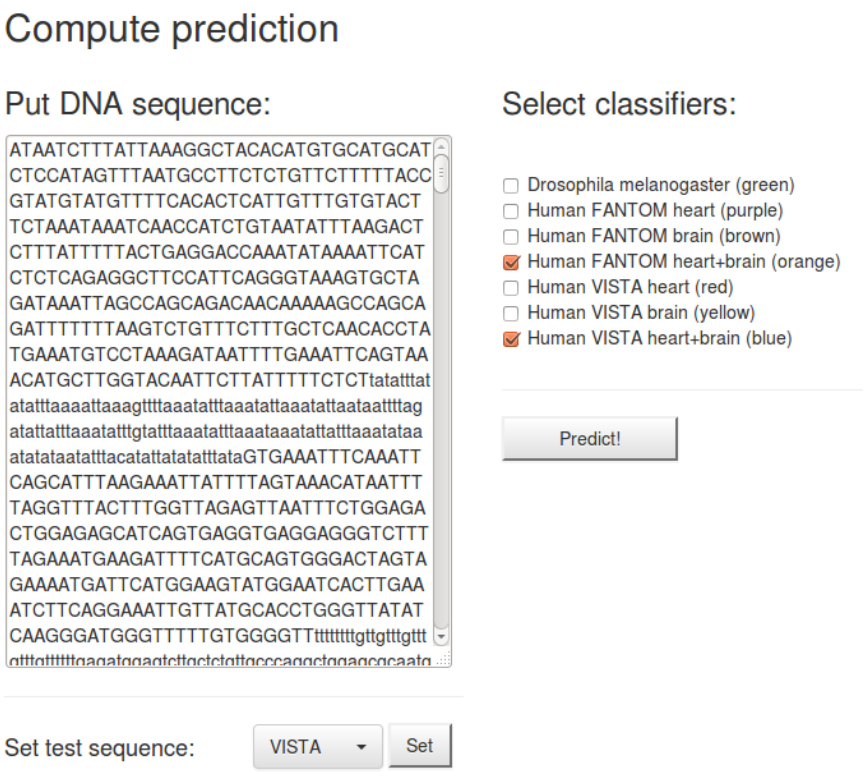
MANUAL

1. Compute prediction

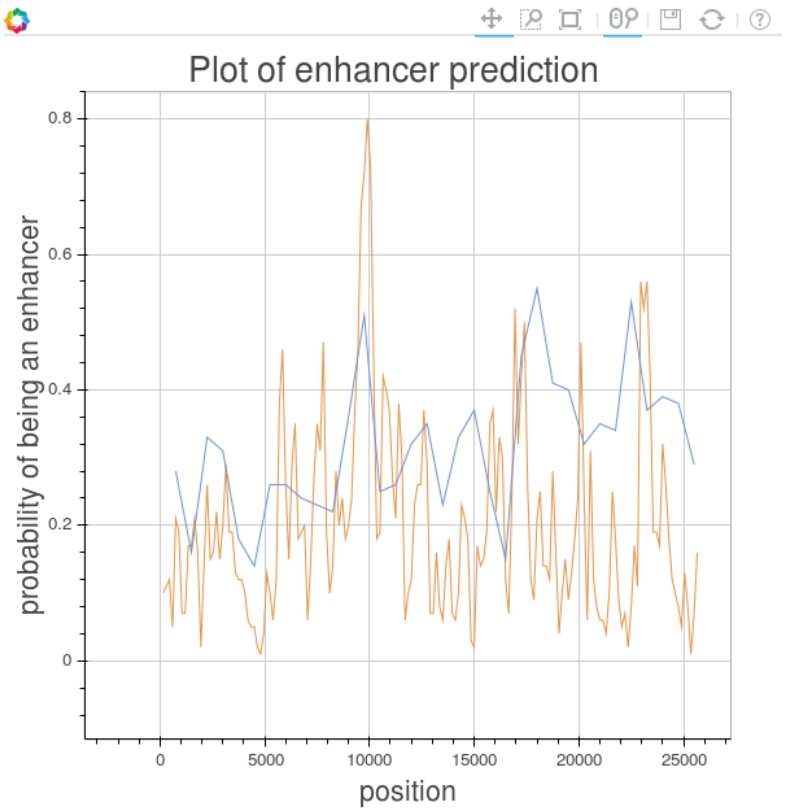
This tool is used to compute the prediction of being an enhancer. It uses classifiers based on Random Forest method, trained on 4-mers frequency of DNA sequence.



You can put both uppercase and lowercase nucleotides characters and white characters (new lines, spaces). We were prepared for testing 3 sample DNA sequences, which were chosen specifically for the presented classifiers. Classifiers were trained on fixed length DNA sequences, hence it is required to note minimum length of the typing DNA sequence:

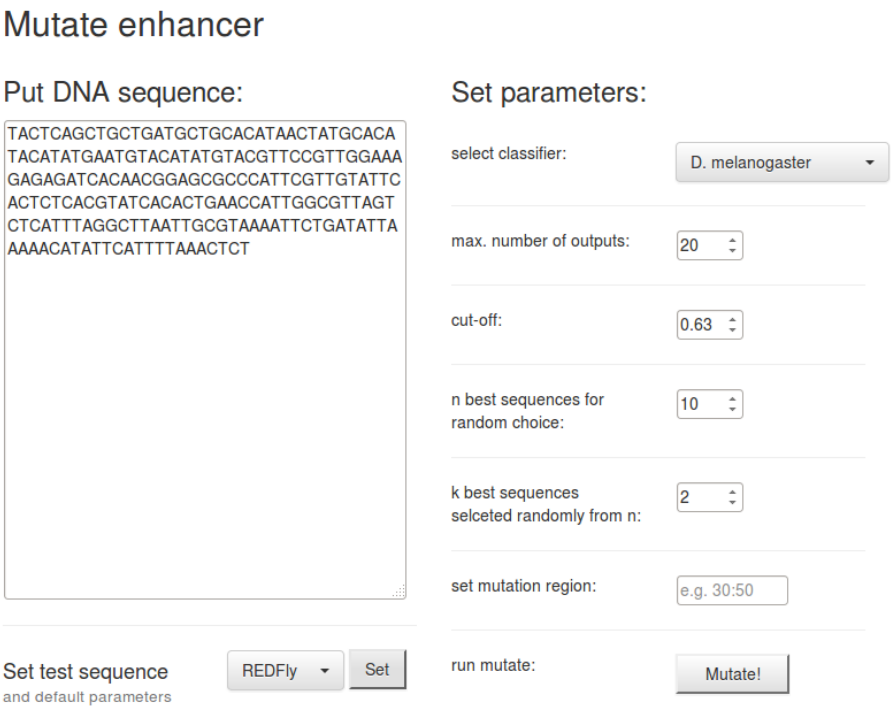
classifier	frame length	step length
Drosophila melanogaster	200 n.	100 n.
Human FANTOM	300 n.	150 n.
Human VISTA	1500 n.	750 n.

Results of prediction are visualized by an interactive graphic containing line charts using the library Bokeh (http://bokeh.pydata.org/en/latest/):



2. Mutate

This tool is used to find a point mutations which decreases prediction value of the DNA sequence to the level of potentially enhancer inactivating. It uses classifiers based on Random Forest method, trained on 4-mers frequency of DNA sequence.



You can put both uppercase and lowercase nucleotides characters and white characters (new lines, spaces). We were prepared for testing 3 sample DNA sequences, which were chosen specifically for the presented classifiers (with default parameters). Classifiers were trained on fixed length DNA sequences and it is required to type sequence with frame lengths provided below:

classifier	**frame length**	**step length**
Drosophila melanogaster	200 n.	100 n.
Human FANTOM	300 n.	150 n.
Human VISTA	1500 n.	750 n.

Before mutation computing you can set following parameters:

- **max. number of outputs**
- **cutoff:** prediction value of which is considered a potential inactivation of the enhancer sequence
- **n best sequences for random choice:** number of mutated sequences with the lowest prediction value of which will be selected randomly sequences to the next iteration
- **k best sequences selected randomly from n:** number of random sequences from the *n best sequences for random choice* taken to the next iteration
- **set mutation region** proper format of a region in which the sequence will be mutated: *start_position:end_position*, e.g. *20:200* (default: entire length of the sequence)

Proposition of mutated sequence is visualized with highlighted of modified nucleotides:

Sequence mutated: 1

```
         10         20         30         40         50
TACTCAGCTG CTGATGCTGC ACATAACTAT GCACATCCAT ATGAATGTAC
         60         70         80         90         100
ATATGTACGT TCCGTTGGAA AGAGAGATCA CAACGGAGCC CCCATTCGTT
        110        120        130        140        150
GTATTCACTC TCACGTATCA CACTGAACCA TTGGCGTTAG TCTCATTTAG
        160        170        180        190        200
GCTTAATTGC TTAAAATTCT GATATTAAAA ACATATTCAT TTTAAACTCT
```

Positions:

```
1) A -> C on positions [37]
2) G -> C on positions [90]
3) G -> T on positions [161]
```

Prediction value:

original sequence	mutated sequence
0.93	0.63

Sequence mutated: 2

```
         10         20         30         40         50
TACTCAGATG CTGATGCTGC ACATAACTAT GCACATACAT ATGAATGTAC
         60         70         80         90         100
ATATGTACGT TCCGTTGGAA AGAGAGATCA CAACGGAGCC CCCATTCGTT
        110        120        130        140        150
GTATTCACTC TCACGTATCA CACTGAACCA TTGGCGTTAG TCTCATTTAG
        160        170        180        190        200
GCTTAATTGC TTAAAATTCT GATATTAAAA ACATATTCAT TTTAAACTCT
```

Positions:

```
1) C -> A on positions [8]
2) G -> C on positions [90]
3) G -> T on positions [161]
```

Prediction value:

original sequence	mutated sequence
0.93	0.63