Building a pipeline for automated CRISPR construct design in *Drosophila*

Background: Transcription factors (TFs) are proteins that bind to the DNA and regulate the transcription of nearby target genes. To study the expression and function of TFs, specific antibodies are required. However, antibodies are only available for a small subset of *Drosophila* TFs. Alternatively, a small tag that is detectable with antibodies can be added to the protein of interest using transgenesis. We have recently developed a tool to tag proteins in *Drosophila* at their endogenous location with an exchangeable tag using CRISPR/Cas9. For this, two plasmids have to be injected into fly embryos, one for expression of a single guideRNA (sgRNA) to induce a DNA double strand break in the genome near the protein of interest and one as a template for homology-directed repair (HDR) to introduce the tag to the genome. Two primers are required for the cloning of each sgRNA plasmid, and two for each of the two homology arms of the HDR plasmid (i.e., six primers in total for each tag). For a large-scale project, TFTag, we are going to tag all 753 *Drosophila* TFs at all annotated protein termini. This requires the design of about 1,900 HDR constructs and sgRNAs for transgenesis.

Aim: To create a script to automate sgRNA and primer design for the TFTag project.

We have manually designed a first set of primers. The information collected during the process (up for discussion, maybe we don't need all of it) can be seen in Tagginglist.xlsx/Tagginglist.txt:

- A-B: TF identifiers
- C: CRISPR cut site location (Cas9 will cut between the two coordinates)
- D: direction of the gene relative to the reference genome
- E: relative location of the tag (N or C terminus)
- F: which isoforms share this terminus
- G: is this isoform a read-through of a stop codon (there are surprisingly many, not sure if physiologically relevant)
- H: does tagging this terminus disrupt the coding sequence of a different isoform or gene
- I-K: sgRNA sequence and primers
- L-N, Q-S: primers and sequence of HAL and HAR
- O, T: length of HAL and HAR
- P, U: mutations introduced in HAL / HAR to destroy sgRNA recognition sites
- X, Y: validation primers

Homology arms

Homology arms will be amplified from genomic DNA using PCR. They each should have a length of 900-1000 bp.

Guide RNAs

The Cas9 protein induces a double strand break in genomic DNA where the 20 bp sgRNA is followed by a primary adjacent motif (PAM), which has the sequence NGG. sgRNAs are designed by selecting one that is as close as possible to the start / stop codon that we want to tag. Ideally the sgRNA spans the terminus so that neither homology arm (HA) has > 15 bp overlap with 3' of sgRNA (including PAM) as this could result in the HDR vector being cut by Cas9 (see Fig. 1). If this is not possible, blocking mutations need to be introduced in the HA primers to destroy the recognition site (see Fig

2). Blocking mutations are preferably made as silent coding mutations and preferably mutate one of the Gs of the PAM. If this is not possible, introduce two silent mutations as close as possible to the PAM (but not the N in NGG as this has no effect.). https://doi.org/10.1038/s41598-021-98965-y has evaluated the effect of the position of the blocking mutations relative to the PAM.

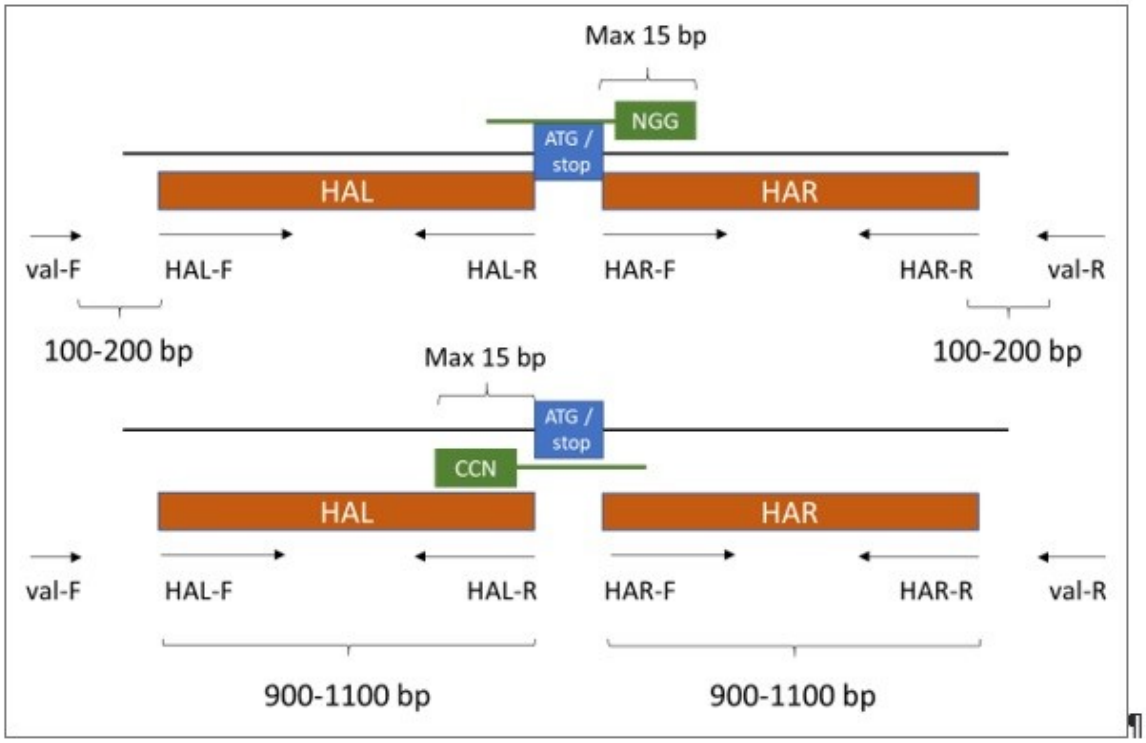

Fig.·1:·Ideal·configuration¶

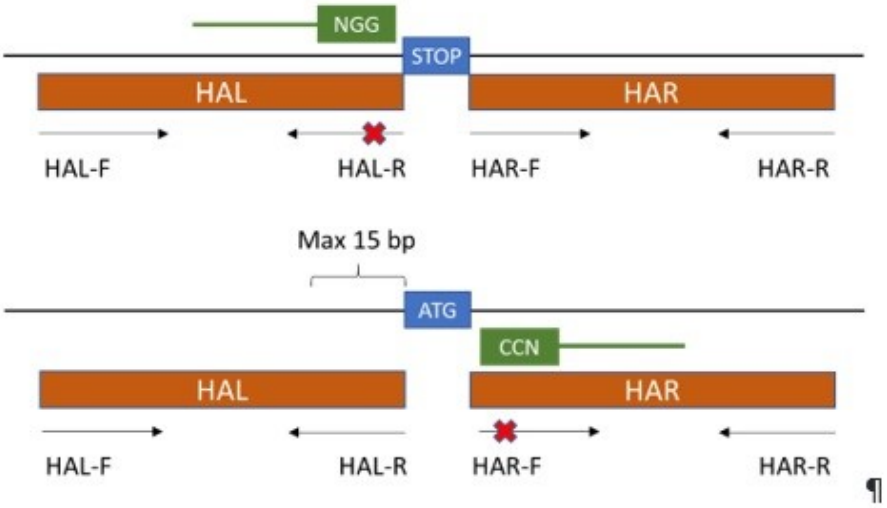

Fig.·2:·Also·acceptable¶

flyrnai (https://www.flyrnai.org/crispr3/web/) holds a file with all possible guide RNA sequences in the genome. We can download the entire set as a gff file here:
http://flybase.org/reports/FBlc0003481.html

Some modules / functions that we will need

1) Function to interact with the *Drosophila* genome annotations

   Input: a list of flybase ids of TFs (TFs.xlsx/TFs.txt)

   Outputs:

   different start and stop codons for each gene (new line for each unique terminus, indicate if it is a unique start or stop codon).

   orientation of gene

   for each start / stop codon, the protein variants that use these

   for each start or stop codon, the genomic coordinates. (not sure if this can be directly lifted from flybase or if this requires blasting of CDSs against the *Drosophila* genome)

   nucleotides -1 - -30 relative to start or stop to design HAL-R

   nucleotides -900 - -1000 relative to start or stop to design HAL-F

   nucleotides -100 - -200 relative to the 5' of HAL-F to design val-F

   nucleotides +4 +34 relative to start or stop to design HAR-F

   nucleotides +900 - +1000 relative to start or stop to design HAR-R

   nucleotides +100 - +200 relative to the 5' of HAR-R to design val-R

2) Function to interact with the sgRNA gff file. The tar.gz file unzips in several gff files with varying levels of off target stringency. You probably need to cycle through them, if the most stringent one has no suitable sgRNA, check the next level down.

   Input:

   genomic coordinates of termini

   is it N- or C-terminus

   Check:

   Does it overlap start / stop codon?

   Output:

   Sequence of closest sgRNA

   Coordinates of closest sgRNA

   Does this sgRNA require us to mutate the HA?

3) Function to interact with primer3. You should be able to recycle this from someone else, e.g. https://pypi.org/project/primer3-py/. There isn't much wiggle room for HAL-R and HAR-F as they **must** start at a particular location (before and after the start/stop codon). Manually, one would copy 30 nucleotides up (HAL-R) or downstream (HAR-F) of the start or stop codon, plug this into primer3 and see if any of the primers start at position 1 (HAR-F) or

position 30 (HAL-R). If not, I would start relaxing various parameters until I get one. If none come up, I just take a primer that doesn't start at the correct position and manually elongate it, so it does start correctly. So probably again cycle through different versions with decreasing stringency until a primer is found (see parameters below). If the primer needs to be mutated, it should be longer to compensate for the mismatch. This is probably easiest to achieve by increasing the minimum annealing temp and the maximum primer length.

Input:

nucleotide sequence outputted by Function 1

will the primer be mutated?

Output: primer sequence

4) Function to mutate the primers. Any ideas on how to implement this are welcome.

Input:

Primer sequence

Reading frame info

Position of the PAM

Check:

Can we mutate pam Gs with silent mutation

If not, which are the closest nucleotides to the PAM that we can mutate

Output: mutated primer