summarize_selected_probes

July 30, 2020

0.0.1 Summarize selected probes

- Get the probes selected in Dmel and add properties from current version of the probe designer
- Also output a fasta file to use for blasting against the Dmel transcriptome

[1]:

```
#Imports
import sys
import pandas as pd
import os
import primer3
from Bio.SeqUtils import MeltingTemp as mt
import primer3
from Bio.Seq import Seq

sys.path.append('../scripts/')
from plot_helpers import *
sys.path.append(os.path.join(probe_designer_dir, 'scripts'))
import screen_kmers
import choose_probes

%matplotlib inline
%load_ext autoreload
%autoreload 2
```

[2]:

```
outdir = '../figures/F1/'
os.makedirs(outdir, exist_ok = True)
```

[3]:

```
#Get the probe sequences, and add to the df
qpcr_dir = os.path.join(results_dir, 'qPCR_data')
probe_seqs = os.path.join(qpcr_dir, 'probe_seqs.csv')
seq_df = pd.read_csv(probe_seqs, index_col = 'probe_name')
#Now get the properties for the probes selected for Drosophila
#chosen probes, these are the ones that were included in the Ribo-Pop mix for␣
 ↪sequencing
chosen_probes_18S = [12, 18, 21, 24, 28]
chosen_probes_28S = [36, 37, 38, 39, 40, 41, 42, 43, 44, 45]
chosen_18S = seq_df.loc[seq_df['probe_num'].isin(chosen_probes_18S),␣
 ↪['sequence']].copy()
```

```
chosen_28S = seq_df.loc[seq_df['probe_num'].isin(chosen_probes_28S),␣
 ↪['sequence']].copy()
chosen_18S['target_name'] = '18S'
chosen_28S['target_name'] = '28S'
chosen_df = pd.concat([chosen_18S, chosen_28S])
chosen_df.reset_index(drop = True, inplace = True)
chosen_df['probe_num'] = chosen_df.index + 1
chosen_df['length'] = chosen_df['sequence'].apply(lambda x: len(x))
chosen_df['unique_id'] = chosen_df.apply(lambda x: '%s_%s' % (x['target_name'],␣
 ↪x['probe_num']), axis = 1)
chosen_df.set_index('unique_id', drop = False, inplace = True)
```

[4]:

```
#Get the values calculated from the probe design pipeline and add
dmel_18S = os.path.join(results_dir, 'probe_design_results/dmel_200729/
 ↪probe_design/18S/potential_probes_filt.csv')
dmel_28S = os.path.join(results_dir, 'probe_design_results/dmel_200729/
 ↪probe_design/28S/potential_probes_filt.csv')
dmel_18S_df = pd.read_csv(dmel_18S)
dmel_28S_df = pd.read_csv(dmel_28S)
allfilt_df = pd.concat([dmel_18S_df, dmel_28S_df])

cols2write = ['Tm','sequence', 'target_start', 'target_end','passed_excluded',␣
 ↪'hairpin_dG', 'homodimer_dG', 'passed_structure',
              'GC_content', 'A_content', 'C_content', 'GC_content_rule',␣
 ↪'A_composition_rule', 'C_composition_rule',
              '4xA_stack_rule', '4xC_stack_rule', 'earlyCs_rule', 'any5_rule',␣
 ↪'rolling_Tm_quantile_co']


col_order = ['probe_num', 'sequence', 'target_name', 'target_start',␣
 ↪'target_end', 'length', 'unique_id', 'Tm', 'GC_content', 'A_content',
'C_content', 'rolling_Tm_quantile_co', 'hairpin_dG', 'homodimer_dG',␣
 ↪'dimer_dG', 'dimer_partner', 'GC_content_rule',
'A_composition_rule', 'C_composition_rule', '4xA_stack_rule', '4xC_stack_rule',␣
 ↪'earlyCs_rule', 'any5_rule']

annotated_df = pd.merge(chosen_df, allfilt_df[cols2write], left_on =␣
 ↪'sequence', right_on = 'sequence', how = 'left')
annotated_df.set_index('unique_id', inplace = True)
annotated_df[['dimer_dG', 'dimer_partner']] = choose_probes.
 ↪calc_dimer(annotated_df)
annotated_df.reset_index(inplace = True)
annotated_df[col_order].round(2).to_csv(os.path.join(outdir,␣
 ↪'Dmel_selected_properties.csv'), index = False)
```

```
[5]: #write fasta file of the probes -- note that we expect matches to align to the␣
      ↪negative strand
     with open(os.path.join(outdir, 'Dmel_probes.fa'), 'w') as g:
         for i in annotated_df.itertuples():
             g.write('>%s\n%s\n' % (i.unique_id, i.sequence))
```

```
[ ]:
```