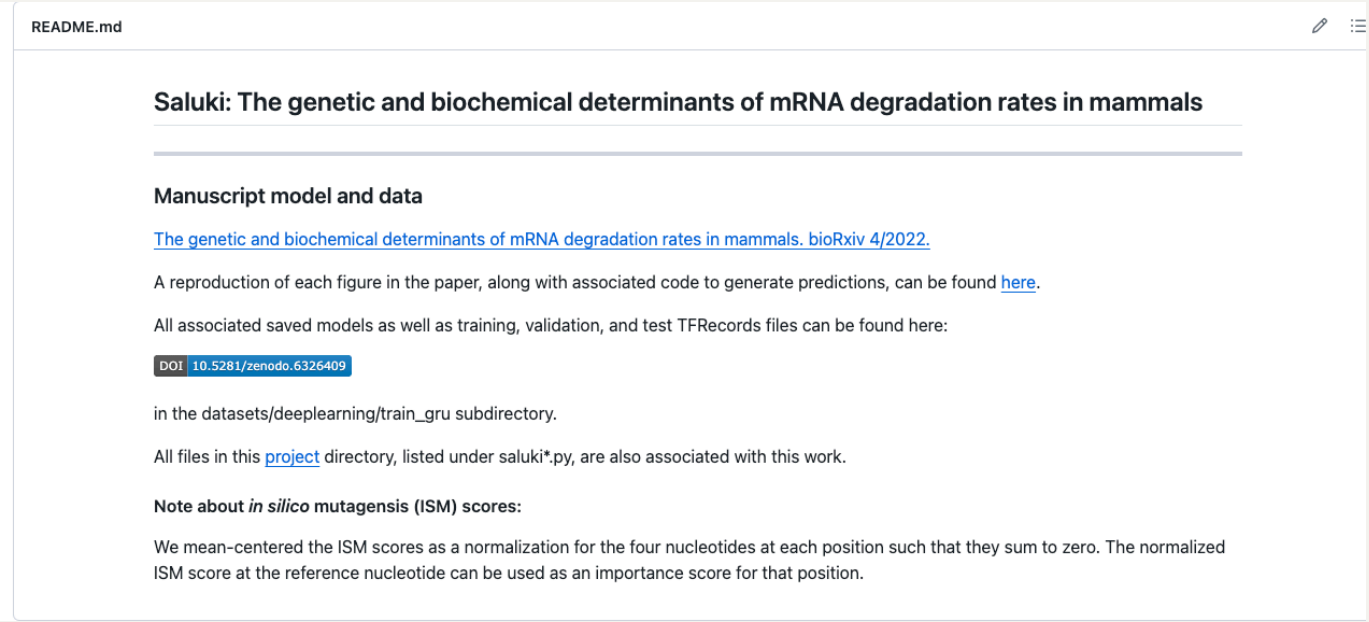
Saluki for mRNA half life

1. Intro


README.md

Saluki: The genetic and biochemical determinants of mRNA degradation rates in mammals

Manuscript model and data

The genetic and biochemical determinants of mRNA degradation rates in mammals. bioRxiv 4/2022.

A reproduction of each figure in the paper, along with associated code to generate predictions, can be found here.

All associated saved models as well as training, validation, and test TFRecords files can be found here:

DOI 10.5281/zenodo.6326409

in the datasets/deeplearning/train_gru subdirectory.

All files in this project directory, listed under saluki*.py, are also associated with this work.

Note about *in silico* mutagensis (ISM) scores:

We mean-centered the ISM scores as a normalization for the four nucleotides at each position such that they sum to zero. The normalized ISM score at the reference nucleotide can be used as an importance score for that position.

https://github.com/calico/basenji/tree/master/manuscripts/saluki

The first link is https://github.com/vagarwal87/saluki_paper, it introduced how to generate figs in the paper. (https://github.com/vagarwal87/saluki_paper/tree/main/Fig5_S6)

But it mainly focused on how to draw plots after we get the results. eg. (https://github.com/vagarwal87/saluki_paper/blob/main/Fig5_S6/Fig5bc.R)

saluki_paper / Fig5_S6 / Fig5bc.R

vagarwal87 Updating files

Code Blame Executable File · 19 lines (15 loc) · 528 Bytes Code 55% faster with GitHub Copilot

```
1  library(ggplot2)
2  library(glmnet)
3  library(LSD)
4
5  a=read.csv("experimentResults.csv")
6  round(a,3)
7
8  results = data.frame(tests = c("BC3MS vs Saluki human","BeEM vs Saluki human","BC3MSD vs Saluki mouse"), pvals =
9  c(t.test(a[,1], a[,3], paired=T)$p.value,
10 t.test(a[,2], a[,3], paired=T)$p.value,
11 t.test(a[,4], a[,5], paired=T)$p.value))
12 results
13
14 meltData <- reshape::melt(a)
15
16 pdf("png/violinplots.pdf",width=4,height=4) #Fig5b
17 p<-ggplot(meltData, aes(x=variable, y=value)) + geom_violin(position=position_dodge(1))
18 print(p)
19 dev.off()
```

For the last link (https://github.com/calico/basenji/tree/master/bin), we can check all the code listed as saluki*.py.

2. Train

In https://github.com/calico/basenji/blob/master/bin/saluki_train.py, we know the input of how to train the model:

```
################################################################################
def main():
  usage = 'usage: %prog [options] <params_file> <data1_dir> ...'
  parser = OptionParser(usage)
  parser.add_option('-o', dest='out_dir',
      default='train_out',
      help='Output directory for test statistics [Default: %default]')
  (options, args) = parser.parse_args()

  if len(args) < 2:
    parser.error('Must provide parameters and data directory.')
  else:
    params_file = args[0]
    data_dirs = args[1:]

  # read model parameters
  with open(params_file) as params_open:
    params = json.load(params_open)
  params_model = params['model']
  params_train = params['train']
  os.makedirs(options.out_dir, exist_ok=True)
  if params_file != '%s/params.json' % options.out_dir:
    shutil.copy(params_file, '%s/params.json' % options.out_dir)

  # read datasets
  train_data = []
  eval_data = []

  for data_dir in data_dirs:
    # load train data
    train_data.append(dataset.RnaDataset(data_dir,
      split_label='train',
      batch_size=params_train['batch_size'],
      shuffle_buffer=params_train.get('shuffle_buffer', 1024),
```

It has 2 required parameters and 1 optional parameter (output path).

For the required parameters, we should have the parameter file of the model and the input data path. For example,

The parameter json:

ying@methylome3: /mnt/data1/ying/saluki_paper/datasets/deeplearning/train_gru/f1_c1/train

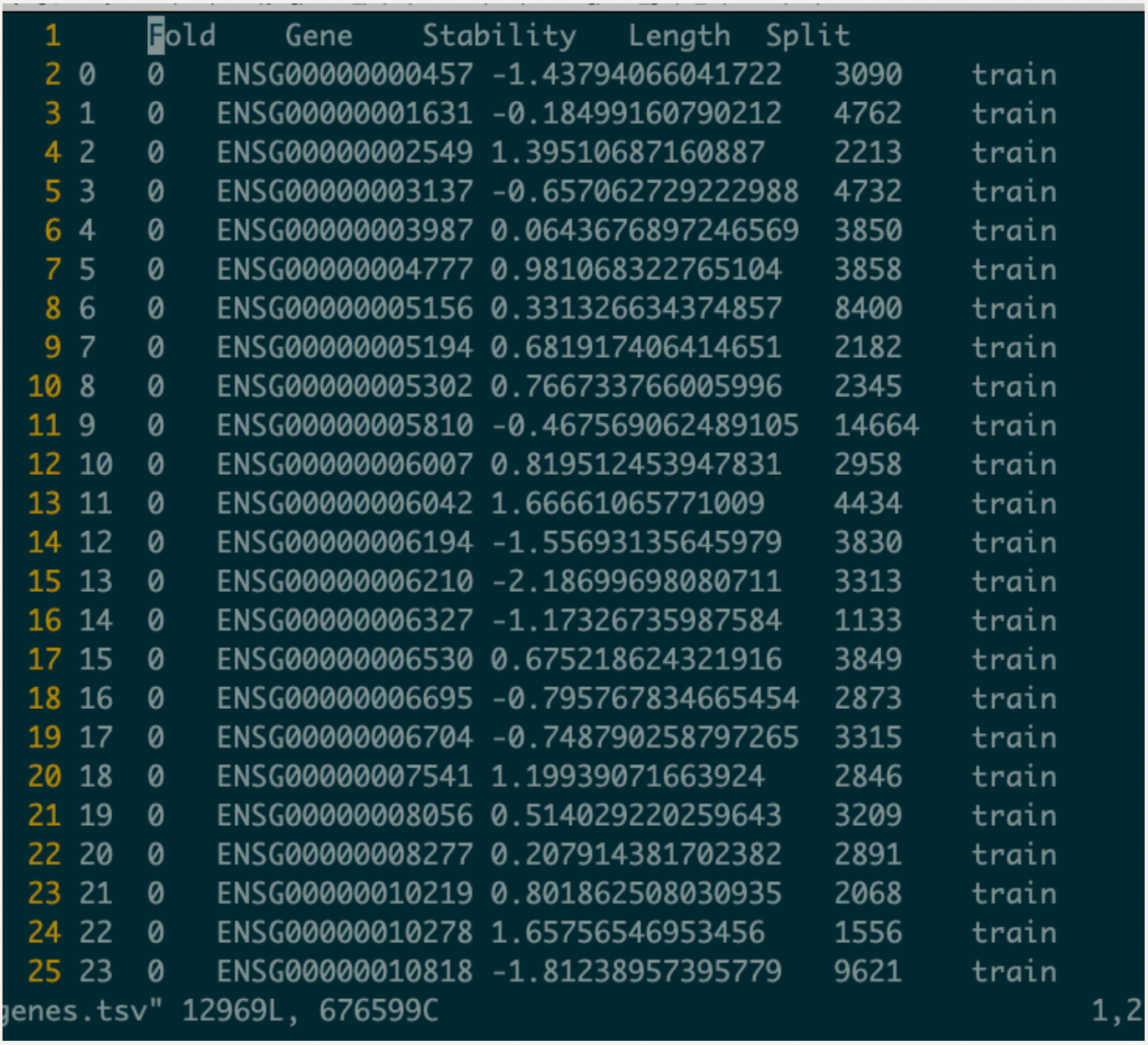
```
{
    "train": {
        "batch_size": 64,
        "optimizer": "adam",
    "loss": "mse",
        "learning_rate": 0.0001,
        "adam_beta1": 0.90,
        "adam_beta2": 0.998,
        "global_clipnorm": 0.5,
        "train_epochs_min": 100,
    "train_epochs_max": 250,
        "patience": 25
    },
    "model": {
        "activation": "relu",
    "rnn_type": "gru",
    "seq_length": 12288,
    "augment_shift": 3,
    "num_targets": 1,
    "heads": 2,
        "filters": 64,
        "kernel_size": 5,
        "dropout": 0.3,
        "l2_scale": 0.001,
        "ln_epsilon": 0.007,
        "num_layers": 6,
        "bn_momentum": 0.90
    }
```

The input path should have these files:

```
ying@methylome3:/mnt/data1/ying/saluki_paper/datasets/deeplearning/train_gru/f1_c1/
data0$ ls
genes.tsv  statistics.json  tfrecords
```

Genes.tsv (file), statistics.json (file). tfrecordes (directory)

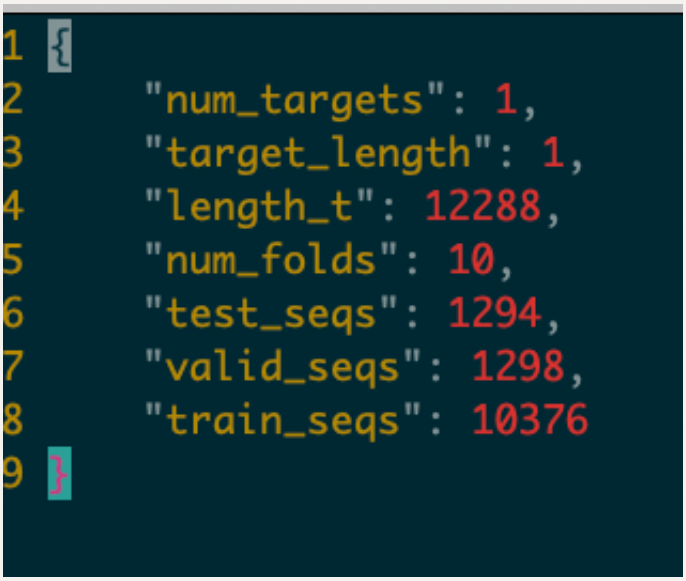
genes.tsv:



		Fold	Gene	Stability	Length	Split
2	0	0	ENSG00000000457	-1.43794066041722	3090	train
3	1	0	ENSG00000001631	-0.18499160790212	4762	train
4	2	0	ENSG00000002549	1.39510687160887	2213	train
5	3	0	ENSG00000003137	-0.657062729222988	4732	train
6	4	0	ENSG00000003987	0.0643676897246569	3850	train
7	5	0	ENSG00000004777	0.981068322765104	3858	train
8	6	0	ENSG00000005156	0.331326634374857	8400	train
9	7	0	ENSG00000005194	0.681917406414651	2182	train
10	8	0	ENSG00000005302	0.766733766005996	2345	train
11	9	0	ENSG00000005810	-0.467569062489105	14664	train
12	10	0	ENSG00000006007	0.819512453947831	2958	train
13	11	0	ENSG00000006042	1.66661065771009	4434	train
14	12	0	ENSG00000006194	-1.55693135645979	3830	train
15	13	0	ENSG00000006210	-2.18699698080711	3313	train
16	14	0	ENSG00000006327	-1.17326735987584	1133	train
17	15	0	ENSG00000006530	0.675218624321916	3849	train
18	16	0	ENSG00000006695	-0.795767834665454	2873	train
19	17	0	ENSG00000006704	-0.748790258797265	3315	train
20	18	0	ENSG00000007541	1.19939071663924	2846	train
21	19	0	ENSG00000008056	0.514029220259643	3209	train
22	20	0	ENSG00000008277	0.207914381702382	2891	train
23	21	0	ENSG00000010219	0.801862508030935	2068	train
24	22	0	ENSG00000010278	1.65756546953456	1556	train
25	23	0	ENSG00000010818	-1.81238957395779	9621	train

genes.tsv" 12969L, 676599C 1,2

Statistics.json:




```
!python saluki_train.py /content/half_life/params.json
/content/half_life/
```

3. Test

https://github.com/calico/basenji/blob/master/bin/saluki_test.py

```
def main():
      help='Save targets and predictions numpy arrays [Default: %default]')
  parser.add_option('--shifts', dest='shifts',
      default='0',
      help='Ensemble prediction shifts [Default: %default]')
  parser.add_option('-t', dest='targets_file',
      default=None, type='str',
      help='File specifying target indexes and labels in table format')
  parser.add_option('--split', dest='split_label',
      default='test',
      help='Dataset split label for eg TFR pattern [Default: %default]')
  # parser.add_option('--tfr', dest='tfr_pattern',
  #     default=None,
  #     help='TFR pattern string appended to data_dir/tfrecords for subsetting [Default: %default]')
  (options, args) = parser.parse_args()

  if len(args) != 3:
    parser.error('Must provide parameters, model, and test data HDF5')
  else:
    params_file = args[0]
    model_file = args[1]
    data_dir = args[2]

  if not os.path.isdir(options.out_dir):
    os.mkdir(options.out_dir)
```

Required parameters:

- parameter file
- model.h5
- Data_dir (the genes.tsv all contains the `test` genes)

```
!python saluki_test.py /content/half_life/params1.json
train_out/model_best.h5 /content/half_life/
```

Then we'll have the loss, pearsonr, and r2 of testing data.

```
 dense (Dense)                (None, 64)                4160

 dropout_6 (Dropout)          (None, 64)                0

 batch_normalization_1 (Batc  (None, 64)                256
 hNormalization)

 re_lu_8 (ReLU)               (None, 64)                0

 dense_1 (Dense)              (None, 1)                 65

=================================================================
Total params: 155,777
Trainable params: 155,521
Non-trainable params: 256
_________________________________________________________________
None
21/21 [==============================] - 72s 3s/step - loss: 1.3386 - pearsonr: 0.6007 - r2: 0.3585

Test Loss:          1.33857
Test PearsonR:      0.60071
Test R2:            0.35855
```