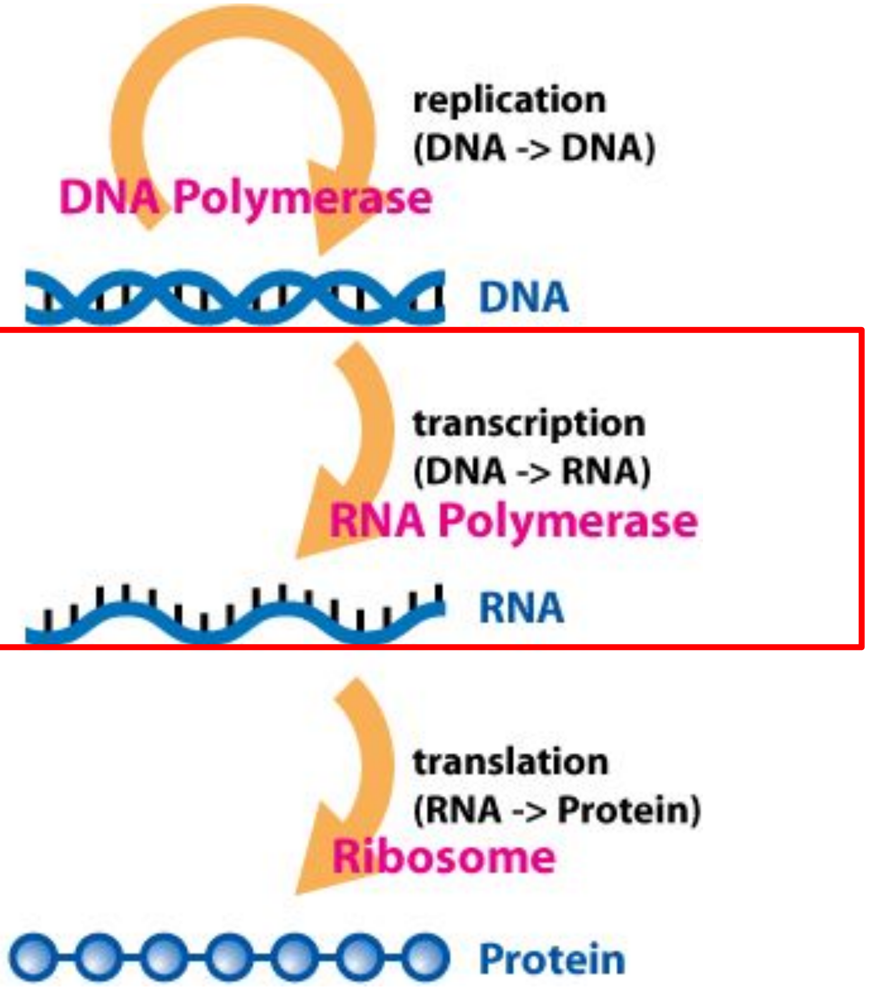
ETH zürich

Splice Site Prediction

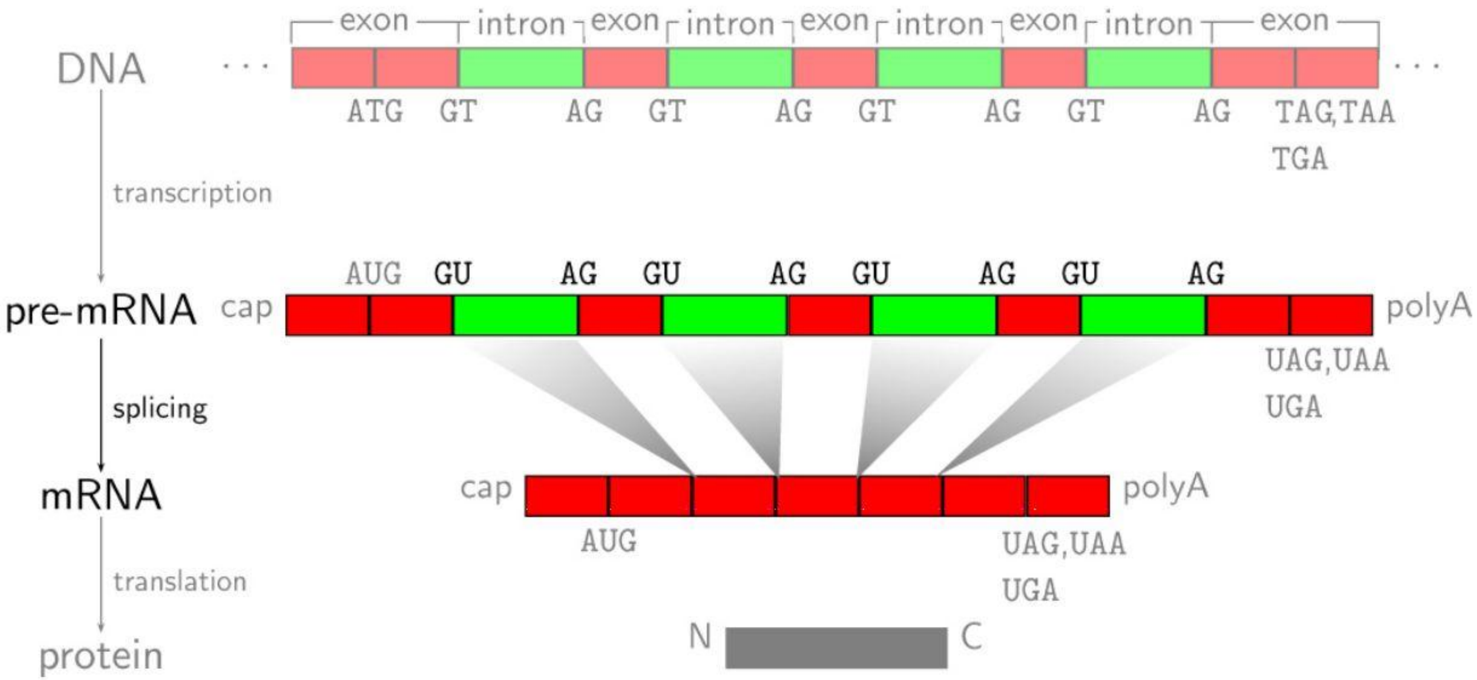
Project 2

Harun Mustafa
23.03.2021

DNA to RNA


replication
(DNA -> DNA)
DNA Polymerase
DNA
transcription
(DNA -> RNA)
RNA Polymerase
RNA
translation
(RNA -> Protein)
Ribosome
Protein

Splice sites



- Almost all *donor splice sites* exhibit GU
- Almost all *acceptor splice site* exhibit AG
- Not all GUs and AGs are used as splice site

Splice site prediction

Binary classification task

- Input: DNA sequences
- Classes: middle of sequence { 1: is an acceptor site; -1: is not an acceptor site }

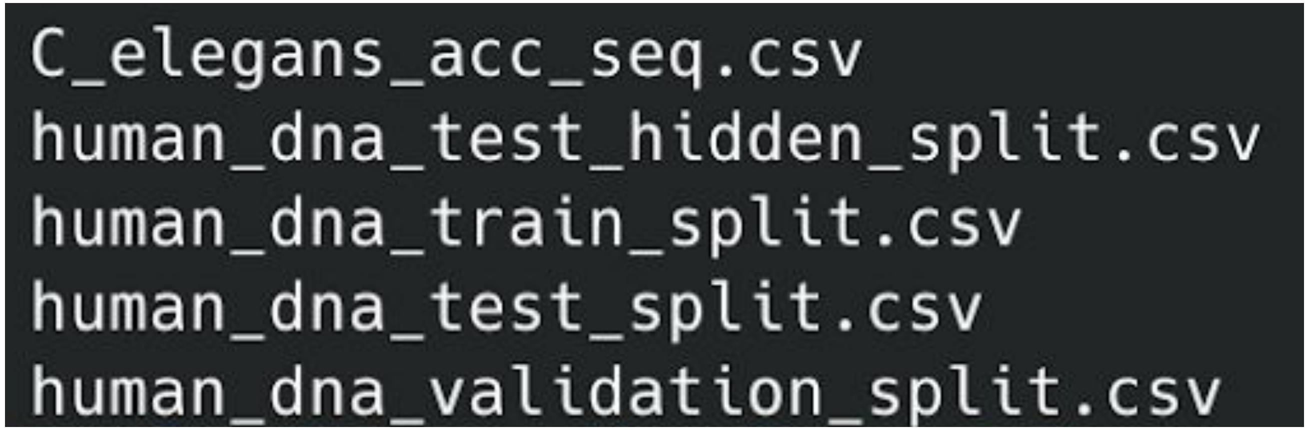
Data sets

- *C. elegans* (roundworm)
- Human

You should train a variety of models for each data set, even simple ones

Data sets

- *C. elegans* has a single CSV
  - split it however you choose
  - remember not to let your test split leak into your training procedure
- Human data provided in several files
  - Train and validation splits
    - Use the validation data however you see fit
  - Test split for your own evaluation
  - Additional test set with no labels (for TAs to use for evaluation)

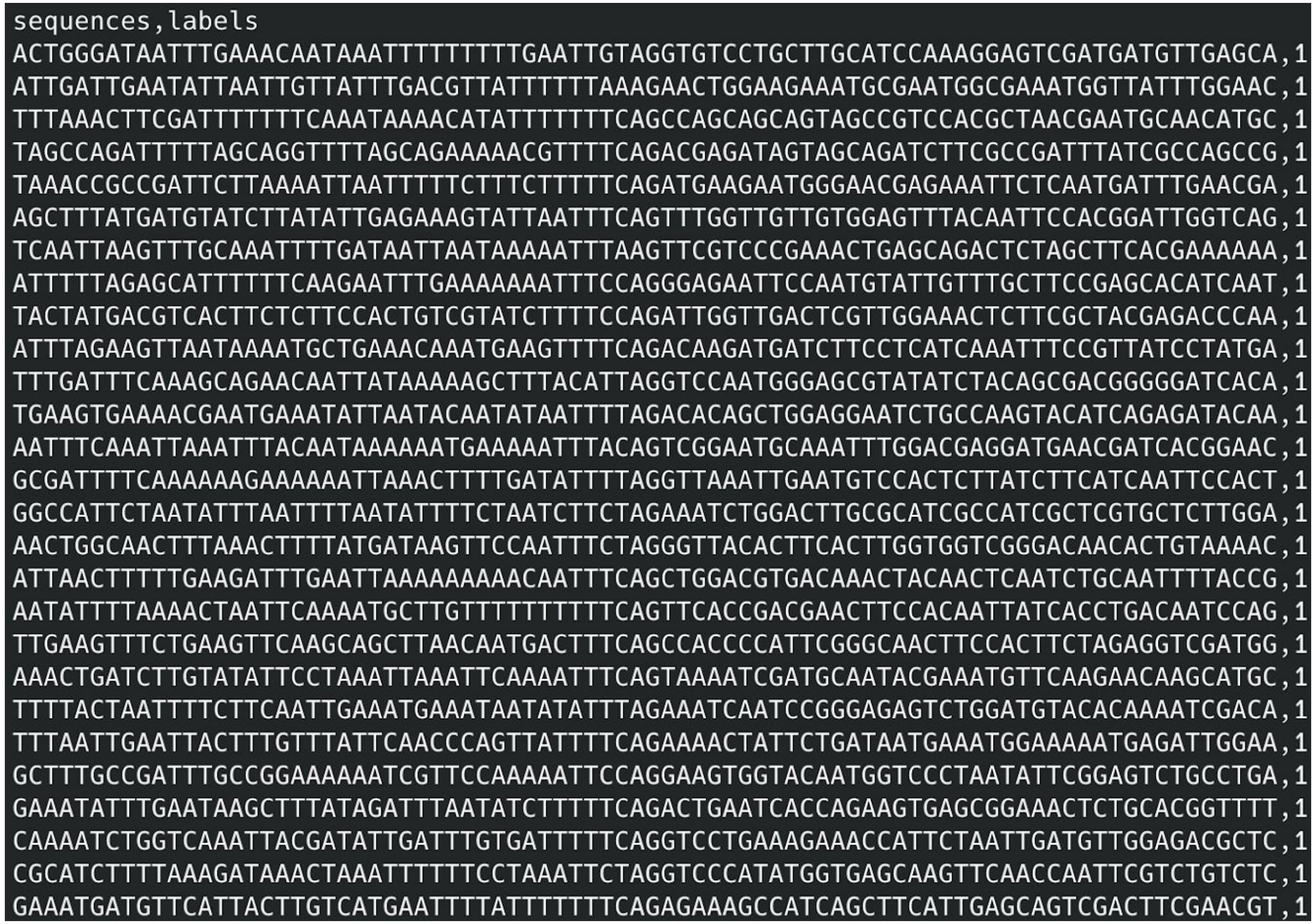


```
C_elegans_acc_seq.csv
human_dna_test_hidden_split.csv
human_dna_train_split.csv
human_dna_test_split.csv
human_dna_validation_split.csv
```

Training data

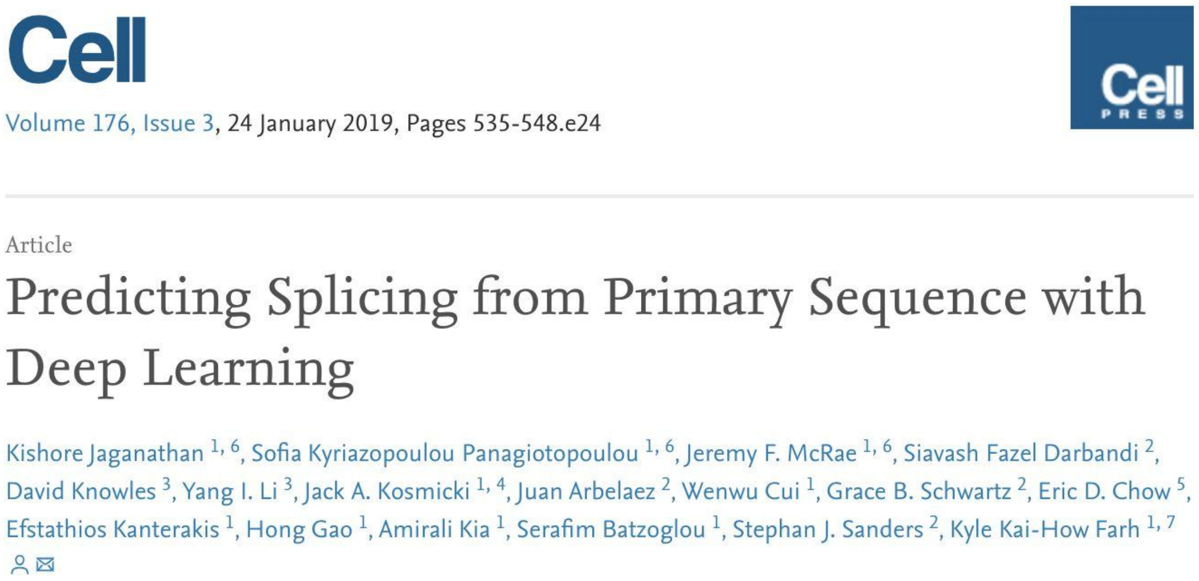
e.g., `C_elegans_acc_seq.csv`

- Sequences derived from DNA, so T instead of U
- Simplified scenario
  - The region in question is in the centre of each sequence
  - Flanking sequence is context
- For a given data set, all sequences are the same length
  - *C. elegans*: 83 bp
  - Human: 398 bp



```
sequences,labels
ACTGGGATAATTTGAAACAATAAATTTTTTTTTGAATTGTAGGTGTCCTGCTTGCATCCAAAGGAGTCGATGATGTTGAGCA,1
ATTGATTGAATATTAATTGTTATTTGACGTTATTTTTTAAAGAACTGGAAGAAATGCGAATGGCGAAATGGTTATTTGGAAC,1
TTTAAACTTCGATTTTTTTCAAATAAAACATATTTTTTTCAGCCAGCAGCAGTAGCCGTCCACGCTAACGAATGCAACATGC,1
TAGCCAGATTTTTAGCAGGTTTTAGCAGAAAAACGTTTTCAGACGAGATAGTAGCAGATCTTCGCCGATTTATCGCCAGCCG,1
TAAACCGCCGATTCTTAAAATTAATTTTTCTTTCTTTTTCAGATGAAGAATGGGAACGAGAAATTCTCAATGATTTGAACGA,1
AGCTTTATGATGTATCTTATATTGAGAAAGTATTAATTTCAGTTTGGTTGTTGTGGAGTTTACAATTCCACGGATTGGTCAG,1
TCAATTAAGTTTGCAAATTTTGATAATTAATAAAAATTTAAGTTCGTCCCGAAACTGAGCAGACTCTAGCTTCACGAAAAAA,1
ATTTTTAGAGCATTTTTTCAAGAATTTGAAAAAAATTTCCAGGGAGAATTCCAATGTATTGTTTGCTTCCGAGCACATCAAT,1
TACTATGACGTCACTTCTCTTCCACTGTCGTATCTTTTCCAGATTGGTTGACTCGTTGGAAACTCTTCGCTACGAGACCCAA,1
ATTTAGAAGTTAATAAAATGCTGAAACAAATGAAGTTTTCAGACAAGATGATCTTCCTCATCAAATTTCCGTTATCCTATGA,1
TTTGATTTCAAAGCAGAACAATTATAAAAAGCTTTACATTAGGTCCAATGGGAGCGTATATCTACAGCGACGGGGGATCACA,1
TGAAGTGAAAACGAATGAAATATTAATACAATATAATTTTAGACACAGCTGGAGGAATCTGCCAAGTACATCAGAGATACAA,1
AATTTCAAATTAAATTTACAATAAAAAATGAAAAATTTACAGTCGGAATGCAAATTTGGACGAGGATGAACGATCACGGAAC,1
GCGATTTTCAAAAAAGAAAAAATTAAACTTTTGATATTTTAGGTTAAATTGAATGTCCACTCTTATCTTCATCAATTCCACT,1
GGCCATTCTAATATTTAATTTTAATATTTTCTAATCTTCTAGAAATCTGGACTTGCGCATCGCCATCGCTCGTGCTCTTGGA,1
AACTGGCAACTTTAAACTTTTATGATAAGTTCCAATTTCTAGGGTTACACTTCACTTGGTGGTCGGGACAACACTGTAAAAC,1
ATTAACTTTTTGAAGATTTGAATTAAAAAAAAACAATTTCAGCTGGACGTGACAAACTACAACTCAATCTGCAATTTTACCG,1
AATATTTTAAAACTAATTCAAAATGCTTGTTTTTTTTTTCAGTTCACCGACGAACTTCCACAATTATCACCTGACAATCCAG,1
TTGAAGTTTCTGAAGTTCAAGCAGCTTAACAATGACTTTCAGCCACCCCATTCGGGCAACTTCCACTTCTAGAGGTCGATGG,1
AAACTGATCTTGTATATTCCTAAATTAAATTCAAAATTTCAGTAAAATCGATGCAATACGAAATGTTCAAGAACAAGCATGC,1
TTTTACTAATTTTCTTCAATTGAAATGAAATAATATATTTAGAAATCAATCCGGGAGAGTCTGGATGTACACAAAATCGACA,1
TTTAATTGAATTACTTTGTTTATTCAACCCAGTTATTTTCAGAAAACTATTCTGATAATGAAATGGAAAAATGAGATTGGAA,1
GCTTTGCCGATTTGCCGGAAAAAATCGTTCCAAAAATTCCAGGAAGTGGTACAATGGTCCCTAATATTCGGAGTCTGCCTGA,1
GAAATATTTGAATAAGCTTTATAGATTTAATATCTTTTTCAGACTGAATCACCAGAAGTGAGCGGAAACTCTGCACGGTTTT,1
CAAAATCTGGTCAAATTACGATATTGATTTGTGATTTTTCAGGTCCTGAAAGAAACCATTCTAATTGATGTTGGAGACGCTC,1
CGCATCTTTTAAAGATAAACTAAATTTTTTCCTAAATTCTAGGTCCCATATGGTGAGCAAGTTCAACCAATTCGTCTGTCTC,1
GAAATGATGTTCATTACTTGTCATGAATTTTATTTTTTTCAGAGAAAGCCATCAGCTTCATTGAGCAGTCGACTTCGAACGT,1
```

For inspiration



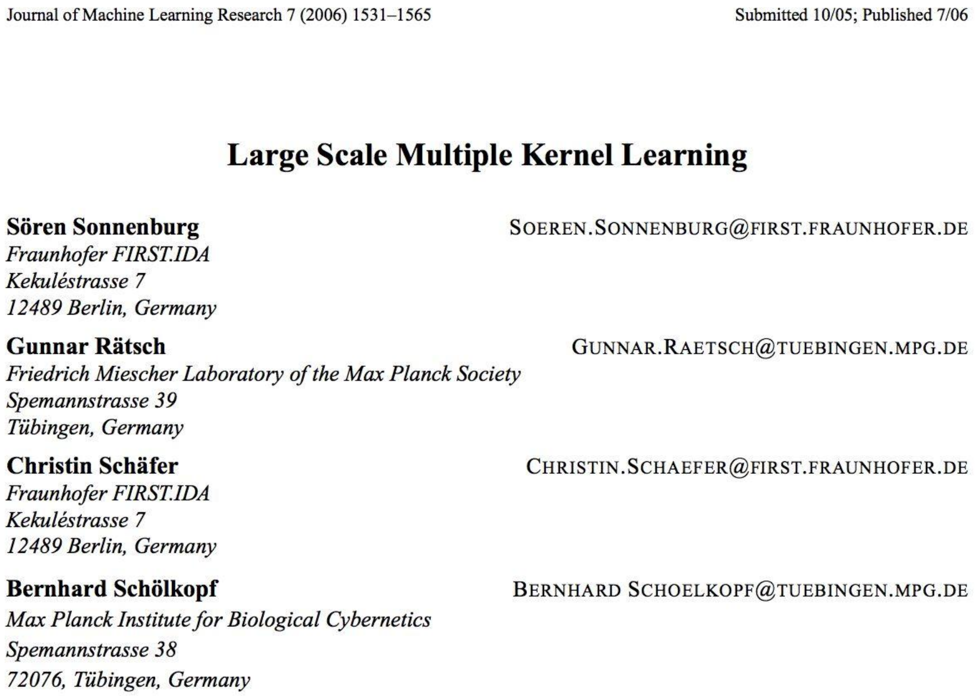
Cell

Volume 176, Issue 3, 24 January 2019, Pages 535-548.e24

Article

Predicting Splicing from Primary Sequence with Deep Learning

Kishore Jaganathan [1, 6], Sofia Kyriazopoulou Panagiotopoulou [1, 6], Jeremy F. McRae [1, 6], Siavash Fazel Darbandi [2], David Knowles [3], Yang I. Li [3], Jack A. Kosmicki [1, 4], Juan Arbelaez [2], Wenwu Cui [1], Grace B. Schwartz [2], Eric D. Chow [5], Efstathios Kanterakis [1], Hong Gao [1], Amirali Kia [1], Serafim Batzoglou [1], Stephan J. Sanders [2], Kyle Kai-How Farh [1, 7]



Journal of Machine Learning Research 7 (2006) 1531–1565 Submitted 10/05; Published 7/06

Large Scale Multiple Kernel Learning

Sören Sonnenburg SOEREN.SONNENBURG@FIRST.FRAUNHOFER.DE
Fraunhofer FIRST.IDA
Kekuléstrasse 7
12489 Berlin, Germany

Gunnar Rätsch GUNNAR.RAETSCH@TUEBINGEN.MPG.DE
Friedrich Miescher Laboratory of the Max Planck Society
Spemannstrasse 39
Tübingen, Germany

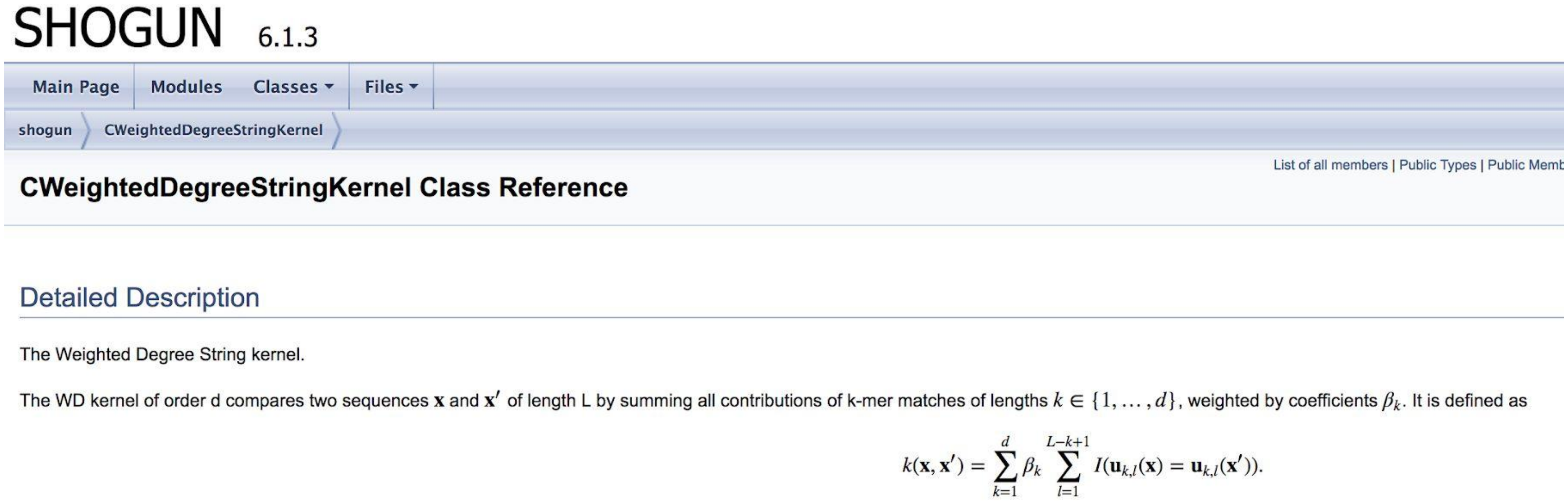
Christin Schäfer CHRISTIN.SCHAEFER@FIRST.FRAUNHOFER.DE
Fraunhofer FIRST.IDA
Kekuléstrasse 7
12489 Berlin, Germany

Bernhard Schölkopf BERNHARD SCHOELKOPF@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
Spemannstrasse 38
72076, Tübingen, Germany

Pitfalls

- Don’t forget to perform hyperparameter searches
- Classes are heavily imbalanced
  - *C. elegans*: 2000 negative samples, 200 positive
  - Human (train + validation): 531,777 negative, 1556 positive
  - Naive methods will classify everything as negative, so explore weighting or resampling schemes
- Feature extraction
  - Depending on your choice of model, one-hot encoding nucleotides may or may not be enough
  - With *k*-mer-based methods, remember that the space has 4^k elements and is sparsely populated, so design your kernels accordingly
- Sequence context is short
  - By comparison, SpliceAI uses a window of size 10k
  - However, sequence in training data are already centred

Tools from various ML toolkits


SHOGUN 6.1.3
Main Page
Modules
Classes
Files
shogun
CWeightedDegreeStringKernel
List of all members | Public Types | Public Memb

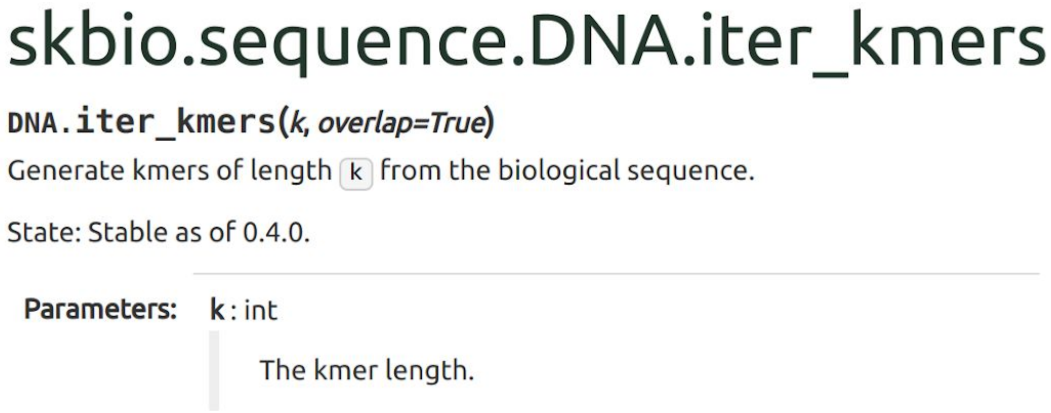
CWeightedDegreeStringKernel Class Reference

Detailed Description

The Weighted Degree String kernel.

The WD kernel of order d compares two sequences $\mathbf{x}$ and $\mathbf{x}'$ of length L by summing all contributions of k-mer matches of lengths $k \in \{1, \ldots, d\}$, weighted by coefficients β_k. It is defined as

$$k(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} I(\mathbf{u}_{k,l}(\mathbf{x}) = \mathbf{u}_{k,l}(\mathbf{x}')).$$



skbio.sequence.DNA.iter_kmers

`DNA.iter_kmers(k, overlap=True)`

Generate kmers of length `k` from the biological sequence.

State: Stable as of 0.4.0.

Parameters: **k** : int

The kmer length.

Deliverables

Train separate set of models for each data set. Provide a data frame which, for each model, reports

- Model name
- Model evaluation
  - ROC (receiver operating characteristic) and PR (precision-recall) curves
  - AUROC and AUPRC (area under ROC and PRC)
- Predicted labels for `human_dna_test_hidden_split.csv`

Submit

- Report describing the methods and detailing each member's contribution
- Conda environment YML
- Clean well-structured code as Jupyter Notebook or Python scripts
- Trained models and evaluation results as data frame in `results.npy`
- README.txt detailing how to run the code and interpret the reported results

Deadline: 26.04.2021 (submit to Moodle, or email harun.mustafa@inf.ethz.ch if too big)

Grading

- 60% of grade will be for the variety of models explored
  - Explore different classifiers, difference sequence encodings, etc.
  - There should be at least 3 with better-than-random performance
  - Diminishing returns after 6 models
- 40% will be based on the performance of the best model