predict_ali

Modules

absl numpy os

Functions

main()

mlforensics

Modules

RNA numpy os

Functions

canon_bp(i, j)

get_bp_counts(seqs, structs)

julia_looptypes(seqs, structs)

julia_prediction(seqs, data)

```
Wenn ich richtig verstehe, wird ab Zeile 36 eine 0,1 Matrix gebaut, die 1
Einträge enthält wenn der NN output >0.5 war und der Eintrag der größte auf
der Zeile ist. Kling gut, ausser dass auch pro Spalte höchstens eine 1
stehen darf.  Wird die NN Matrix vielleicht vorher schon symmetrisch
gemacht?

Der code failed auch, wenn der Maximalwert in einer Zeile doppelt vorkommt,
aber das ist hoffentlich selten genug.
```

julia_version(a)

main()

remove_conflicts(a, seq=None)

generate_data

Modules

numpy os random

Functions

main()

```
Generates random sequence data files.

datadir/datatype.fasta
'''
>rseq_{i}_{energy}

sequence

structure

>rseq_{i+1}_{energy}

sequence

structure

'''
```

train (version v0.1)

Modules

absl argparse os

Functions

main()

```
RNAdeep training interface.
python train.py -d l30 -t train_data/fixlen30_n100000.fa-train -v train_data/fixlen30_n100000.fa-valid -m 4 -b 250 -e 20 --
model-log-dir intermediate_models -l intermediate_models/sm4_l30_010/ --epoch0 10 2>> sm4_l30.err >> sm4_l30.out &
```

parse_rnadeep_args(p)

```
Arguments that are used by RNAdeep.
```

training(datatag, ftrain, fvalid, spotmodel=None, basemodel=None, savedir='.', epochs=50, epoch0=0, batch_size=4)

predict

index
rnadeep/examples/predict.py

Modules

absl numpy os

Functions

main()

train_ali (version v0.1)

index
rnadeep/examples/train_ali.py

Modules

absl argparse os tensorflow

Functions

main()

```
RNAdeep training interface.
```

parse_rnadeep_args(p)

```
Arguments that are used by RNAdeep.
```

training(datatag, dbn_dir, ali_dir, spotmodel=None, basemodel=None, savedir='.', epochs=50, epoch0=0, batch_size=4)

alignment_filter

index
rnadeep/rnaconv/alignment_filter.py

Modules

RNA numpy os sys

Functions

filter_alignments(path, rfam_path, max_dbrs_deviation)

```
Filters the alignments generated by the alignment_generator: In each alignment, every sequence is removed that
deviates by over <max_dbrs_deviation> from the given consensus structure. This is achieved using RNAfold to predict
the structure and the base pair distance to compare it to the desired consensus structure. If this results in all
sequences being removed, the whole alignment file with the corresponding consensus structure file copy is removed.

        Parameters:
                path (str): path directory of the generated families. Expects the following folder structure:
                        - alignments
                        - neighbourhoods
                                - ct
                                - dbn
                rfam_path (str): path to the directory of the convered rfam database. Only used to retrieve the consensus
                structure, therefore expects the following folder structure:
                        - seed_neighbourhoods
                                - ct
                                - dbn
                max_dbrs_deviation (int): maximum allowed base pair distance deviation from the consensus structure in
                        percent. Default is 20.
```

main()

obtain_sissi_frequencies(path, rfam_path)

```
Extracts the equilibrium frequencies for unpaired single nucleotides and nucleotide pairs from the generated
alignment and forms the differences to the already extracted equilibrium frequencies of the original rfam
alignments.

        Parameters:
                path (str): path directory of the generated families. Expects the following folder structure:
                        - alignments
                        - neighbourhoods
                                - ct
                                - dbn
                rfam_path (str): path to the directory of the convered rfam database. Only used to retrieve the alignment,
                consensus structure and original equilibrium frequencies, therefore expects the following folder structure:
                        - seed_alignments
                        - seed_frequencies
                                - single
                                - doublet
                        - seed_neighbourhoods
                                - ct
                                - dbn
                                - nei
                                - wuss
```

Data

default_max_dbrs_deviation = 20

rfam_filter

index
rnadeep/rnaconv/rfam_filter.py

Modules

os sys

Functions

filter_rfam_data(rfam_path, max_length)

```
Filters the alignments, consensus structures, frequencies and trees of the converted rfam database:
Every data point (consisting of the four filetypes mentioned above) which alignment exceeds a certain length is
removed.

        Parameters:
                rfam_path (str): path to the directory of the convered rfam database. Expects the following folder structure:
                        - seed_alignments
                        - seed_frequencies
                                - single
                                - doublet
                        - seed_neighbourhoods
                                - ct
                                - dbn
                                - nei
                                - wuss
                        - seed_trees
                                - original
                                - fixed
                                - rescaled
                max_length (int): maximally allowed length of an alignment. 700 by default.
```

main()

Data

default_max_length = 700

family_filter

index
rnadeep/rnaconv/family_filter.py

Modules

RNA numpy os sys

Functions

filter_alignments(path, max_dbrs_deviation)

```
Filters the alignments generated by the family_generator: In each alignment, every sequence is removed that
deviates by over <max_dbrs_deviation> from the desired consensus structure. This is achieved using RNAfold to
predict the structure and the base pair distance to compare it to the desired consensus structure. If this results
in all sequences being removed, the whole alignment file with the corresponding consensus structure file is
removed.

        Parameters:
```

```
                        path (str): path directory of the generated families. Expects the following folder structure:
                                - alignments
                                - neighbourhoods
                                        - ct
                                        - dbn
                                - sequences
                        max_dbrs_deviation (int):
```

main()

Data

default_max_dbrs_deviation = 20

alignment_generator

index
rnadeep/rnaconv/alignment_generator.py

Modules

RNA	os	subprocess	sys

Functions

db_to_ct(dbn, seq)

```
Converts the consensus structures contained in the dot bracket notation input file into the connect table format

        Parameters:
                dbn (str): secondary structure in dot bracket notation
                seq (str): sequence
```

generate_alignment_set(sissi_filepath, n, tree_dirpath, neigh_dirpath, sfreq_dirpath, dfreq_dirpath, ali_dirpath, outpath)

```
Generates n RNA alignments for each combination of tree, consensus structure, single frequency & double frequency
and alignment files with the same name in the respective directories.
For more information, refer to the generate_alignments() function.

        Parameters:
                sissi_filepath (str): path to the compiled sissi099 file
                n (int): The number of alignments to generate
                tree_dirpath (str): path to a directory containing tree files in the newick string format ('.seed_tree')
                neigh_dirpath (str): path to a directory containing neighbourhood files in the sissi01 format ('.nei')
                sfreq_dirpath (str): path to a directory containing files storing a single frequency vector ('.sfreq')
                dfreq_dirpath (str): path to a directory containing files storing a doublet frequency vector ('.dfreq')
                ali_dirpath (str): path to a directory containing alignment files in the clustal format ('.aln')
                outpath (str): The path to which to write the generated alignments
```

generate_alignments(sissi_filepath, n, tree_filepath, neigh_filepath, sfreq_dfilepath, dfreq_filepath, ali_filepath, outpath)

```
Generates n RNA alignments using sissi for given equilibrium frequencies, neighbourhood system and phylogenetic tree.
The raw alignments are used to re-add the indels.
Note: The provided consensus structure will also be copied into one additional file per generated alignment, in order to
create pairs of samples and tags to be used to train a model.

        Parameters:
                sissi_filepath (str): path to the compiled sissi099 file
                n (int): The number of alignments to generate
                tree_filepath (str): path to a tree file in the newick string format ('.seed_tree')
                neigh_filepath (str): path to a neighbourhood file in the sissi01 format ('.nei')
                sfreq_dfilepath (str): path to a file containing a single frequency vector ('.sfreq')
                dfreq_filepath (str): path to a file containing a doublet frequency vector ('.dfreq')
                ali_filepath (str): path to an alignment file in the clustal format ('.aln')
                outpath (str): The path to which to write the generated alignments
```

get_paths(rfam_path)

```
Accepts a path to a converted rfam database and returns the individual paths for the tree, consensus structure,
frequency and alignment files.

        Parameters:
                rfam_path (str): path to a converted rfam database
```

main()

rfam_converter

index
rnadeep/rnaconv/rfam_converter.py

Modules

RNA	os	subprocess	textdistance
numpy	shutil	sys	

Functions

convert_rfam_data(seed_filepath, ali_outpath, neigh_outpath, freq_outpath, tree_path, tree_fixed_outpath, tree_rescaled_outpath)

```
Calls the nessecary functions to convert the whole rfam database into single files, preparing them to be used by
SISSI.

        Parameters:
                seed_filepath (str): path to the Rfam.seed file in STOCKHOLM format, containing the families
                ali_outpath (str): path to the directory in which to save the extracted alignments in the CLUSTAL format
                neigh_outpath (str): path to the directory in which to save the extracted consensus structures in the
                        - wuss
                        - dbn
                        - ct
                        - nei
                        formats, respectively
                freq_outpath (str): path to the directory in which to save the extracted paired and unpaired nucleotide
                        equilibrium frequencies
                tree_path (str): path to the directory in which Rfam tree files in newick format (.seed_tree) files are
                        located
                tree_fixed_outpath (str): path to the directory in which to save the fixed tree newick strings
                tree_rescaled_outpath (str): path to the directory in which to save the rescaled tree newick strings
```

ct_to_nei(filepath, outpath)

```
Converts the consensus structures contained in the connect table input file into the sissi0.1 (.nei) format
Note: SISSI can also use connect table files directly, but this filetype is used for obtaining the equilibrium
frequencies later on.

        Parameters:
                filepath (str): path to the file in connect table format, containing the secondary structure
                outpath (str): path to the directory in which to save the resulting sissi0.1 (.nei) file
```

db_to_ct(filepath, outpath)

```
Converts the consensus structures contained in the dot bracket notation input file into the connect table format

        Parameters:
                filepath (str): path to the file in dot bracket notation format, containing the secondary structure
                outpath (str): path to the directory in which to save the resulting connect table file
```

fix_newick_strings(treedirpath, outpath)

```
Fixes newick strings by replacing every control character (e.g. '(', ')', ',', '.', ':') within a node name with an
underscore.
Additionally, multifurcations are resolved and non-leaf node labels are removed.

These three steps are nessecary for SISSI to be able to parse the Rfam tree files.

        Parameters:
                treedirpath (str): path to the directory containing the tree files in newick string format
                outpath (str): path to the directory in which to save the trees in the fixed newick string format.
```

main()

obtain_equilibrium_frequencies(alidirpath, neighdirpath, outpath)

```
Extracts the equilibrium frequencies for unpaired single nucleotides and nucleotide pairs from an alignment.

It counts the occurences of single nucleotides per unpaired site and saves them in a 4-vector.
Then It counts the occurences of nucleotide pairs per paired site tuple and saves them in a 16-vector.
Then it adds pseudocounts (+1 for each element) and normalizes.

        Parameters:
                alidirpath (str): path to the directory containing the alignment files in CLUSTAL format
                neighdirpath (str): path to the directory containing the alignment consensus structure files in sissi0.1
                        (.nei) format
                outpath (str): path to the directory in which to save the extracted unpaired single and paired nucleotide
                        equilibrium frequencies
```

rescale_newick_strings(treedirpath, alidirpath, outpath)

```
Rescales the tree branch lengths for trees which corresponding sequence alignments sequences are over 95% similar
with respect to their mean pairwise hamming distance, in order to increase the evolution rate when using the tree
for evolutionary simulation.
The rescale factor is 2.

        Parameters:
                treedirpath (str): path to the directory containing the tree files in newick string format
                alidirpath (str): path to the directory containing the alignment files in CLUSTAL format
                outpath (str): path to the directory in which to save the rescaled trees in the newick string format.
```

stockholm_to_alignments(filepath, outpath)

```
Converts the alignments contained in the STOCKHOLM input file into CLUSTAL files

        Parameters:
                filepath (str): path to the Rfam.seed file in STOCKHOLM format, containing the families
                outpath (str): path to the directory in which to save the extracted alignments in the CLUSTAL format
```

stockholm_to_neighbourhoods(filepath, outpath)

```
Converts the consensus structures contained in the STOCKHOLM input file into single files in the following formats:
        - wuss
        - dbn
        - ct
        - nei

        Parameters:
```

```
filepath (str): path to the Rfam.seed file in STOCKHOLM format, containing the families
outpath (str): path to the directory in which to save the extracted consensus structures
```

stockholm_to_wuss(filepath, outpath)

```
Converts the consensus structures contained in the STOCKHOLM input file into single files in the wuss format

        Parameters:
                filepath (str): path to the Rfam.seed file in STOCKHOLM format, containing the families
                outpath (str): path to the directory in which to save the resulting wuss file
```

wuss_to_db(filepath, outpath)

```
Converts the consensus structures contained in the wuss input file into the dot bracket notation format

        Parameters:
                filepath (str): path to the file in wuss format, containing the secondary structure
                outpath (str): path to the directory in which to save the resulting dot bracket notation file
```

family_generator

index
rnadeep/rnaconv/family_generator.py

Modules

RNA	random	sys
os	subprocess	

Functions

db_to_ct(dbn, seq)

```
Converts the consensus structures contained in the dot bracket notation input file into the connect table format

        Parameters:
                dbn (str): secondary structure in dot bracket notation
                seq (str): sequence
```

generate_family(sissi_filepath, n, length, tree_filepath, sfreq_filepath, dfreq_filepath, outpath)

```
Generates n RNA families (consisting of an alignment and a secondary structure) for the given equilibrium frequencies
and phylogenetic tree, using:
- a random ancestral sequence
- RNAfold to predict a consensus structure for that sequence
- SISSI simulate a corresponding homologous sequence alignment (taking the sequence, tree, and equilibrium
frequencies as input).

        Parameters:
                sissi_filepath (str): Path to the compiled sissi099 file
                n (int): The number of families to generate
                length (int): Length of the ancestral sequence used to generate the family
                tree_filepath (str): Path to a tree file in the newick string format ('.seed_tree')
                sfreq_filepath (str): Path to a file containing a single frequency vector ('.sfreq')
                dfreq_filepath (str): Path to a file containing a doublet frequency vector ('.dfreq')
                outpath (str): The path to which to write the generated families
```

generate_family_set(sissi_filepath, n, length, tree_dirpath, sfreq_dirpath, dfreq_dirpath, outpath)

```
Generates n RNA families of a certain length for each combination of tree, single frequency & double frequency files
with the same name in the respective directories.
For more information, refer to the generate_family() function.

        Parameters:
                sissi_filepath (str): Path to the compiled sissi099 file
                n (int): The number of families to generate
                length (int): Length of the ancestral sequences used to generate the families
                tree_dirpath (str): Path to a directory containing tree files in the newick string format ('.seed_tree')
                sfreq_dirpath (str): Path to a directory containing files storing a single frequency vector ('.sfreq')
                dfreq_dirpath (str): Path to a directory containing files storing a doublet frequency vector ('.dfreq')
                outpath (str): Path to which to write the generated families
```

generate_sequence_structure_pair(length=85, min_paired_sites=0)

```
Repeatedly generates a random sequence and predicts its secondary structure using RNAfold, until the structure
has at least min_paired_sites paired sites.

        Parameters:
                length (int, optional): Length of the random sequence
                min_paired_sites (int, optional): Minimal required sites to be paired
```

get_paths(rfam_path)

```
Accepts a path to a converted rfam database and returns the individual paths for the tree and frequency files.

        Parameters:
                rfam_path (str): path to a converted rfam database
```

main()

Data

default_min_paired_sites = 25

lstm_models

index
rnadeep/rnadeep/lstm_models.py

Modules

keras.api._v2.keras.backend numpy os tensorflow

Functions

blstm(lstm_layers=1, lstm_neurons=20)

complex_blstm(lstm_layers=1, lstm_neurons=40)

metrics

index
rnadeep/rnadeep/metrics.py

Modules

keras.api._v2.keras.backend numpy tensorflow

Functions

f1(y_true, y_pred)

focal_loss(gamma=2.0, alpha=0.75)

matthewscorrelation(y_true, y_pred)

mcc(y_true, y_pred)

sensitivity(y_true, y_pred)

specificity(y_true, y_pred)

__init__ (version v0.1)

index
rnadeep/rnadeep/__init__.py

sliding_window

index
rnadeep/rnadeep/sliding_window.py

Modules

keras.api._v2.keras.backend numpy tensorflow

Functions

basic_window(window_size)

basic_window_leakyrelu(window_size)

conv_window(window_size)

Data

absolute_import = _Feature((2, 5, 0, 'alpha', 1), (3, 0, 0, 'alpha', 0), 262144)
division = _Feature((2, 2, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0), 131072)
print_function = _Feature((2, 6, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0), 1048576)
unicode_literals = _Feature((2, 6, 0, 'alpha', 2), (3, 0, 0, 'alpha', 0), 2097152)

models

index
rnadeep/rnadeep/models.py

Modules

keras.api._v2.keras.layers tensorflow

Functions

spotrna_alignment_models(model=1, use_mask=True)

```
Some modifications to Julia's SPOT-RNA implementations.

Supposed to be a reimplementation of the models in the
SPOT-RNA paper. If you find mistakes, please let us know!

Overview:
        - Initial 3x3 convolution layer
        - ResNet blocks
        - Act./Norm.
        - 2D-BLSTM
        - Fully Connected blocks
        - Output masking layer (optional)
        - Output layer

Args:
        model: select the model (0-4)
        use_mask: for padded input/output (defaults to True!)
```

spotrna_models(model=1, use_mask=True)

```
Some modifications to Julia's SPOT-RNA implementations.

Supposed to be a reimplementation of the models in the
SPOT-RNA paper. If you find mistakes, please let us know!

Overview:
        - Initial 3x3 convolution layer
        - ResNet blocks
        - Act./Norm.
        - 2D-BLSTM
        - Fully Connected blocks
        - Output masking layer (optional)
        - Output layer

Args:
        model: select the model (0-4)
        use_mask: for padded input/output (defaults to True!)
```

encoding_utils

index
rnadeep/rnadeep/encoding_utils.py

Modules

numpy

Functions

base_pair_matrix(ss)

binary_encode(structure)

create_windows(sequences, window_size)

encode_padded_alignment_matrix(alignments, max_length=None)

encode_padded_sequence_matrix(sequences, max_length=None)

encode_padded_structure_matrix(structures, max_length=None)

encode_sequence(sequences)

encode_sequence_matrix(sequences)

```
Make a BP probability matrix with one-hot encoding of basepairs.
NOTE: This only works if all sequences have the same length, otherwise
you need to use: encode_padded_sequence_matrix
```

encode_sequence_windows(sequences, window_size)

encode_structure(structures)

encode_structure_matrix(structures)

```
Make a BP probability matrix with one-hot encoding of basepairs.
NOTE: This only works if all sequences have the same length!
```

make_pair_table(ss, base=0, chars=['.'])

```
Return a secondary struture in form of pair table.

Args:
  ss (str): secondary structure in dot-bracket format
  base (int, optional): choose between a pair-table with base 0 or 1
  chars (list, optional): a list of characters to be are ignored, default:
        ['.']

**Example:**
   base=0: ((..)). => [5,4,-1,-1,1,0,-1]
        i.e. start counting from 0, unpaired = -1
        base=1: ((..)). => [7,6,5,0,0,2,1,0]
             i.e. start counting from 1, unpaired = 0, pt[0]=len(ss)

    Returns:
      [list]: A pair-table
```

one_hot_encode(char)

one_hot_matrix(seq)

profile_vec_matrix(ali)

sampling_ali

index
rnadeep/rnadeep/sampling_ali.py

Modules

numpy   os

Functions

draw_ali_sets(ali_directory, dbn_directory, splits=None)

parse_alignment(ali_path, dbn_path, filename)

parse_alignments(ali_directory, dbn_directory)

sampling

index
rnadeep/rnadeep/sampling.py

Modules

numpy   os   random

Functions

draw_sets(fname, splits=None)

generate_random_structures(lengths)

rseq(l)

write_data_file(data, fname, mode='w')

```
Save sequence/structure pairs for the given lengths.
```

write_fixed_len_data_file(seqlen, num, root='')

write_normal_len_data_file(central, std, num, root='')

write_uniform_len_data_file(minlen, maxlen, num, root='')