README for RF models pipeline for classifying epitope specific TCR.

**Steps:** Overall view of the flow of each step in the pipeline. Step1 involves creation of the kmer dictionaries, step2 involves removal of chains and step 3 involves binning and step uses the bins created in step3 to make predictions. ALSO CONSIDER INCLUDING PYTHON SCRIPT FOR AUC.

**Step1: Kmer**

Inputs:

**Data:** The associated data sets used to generate the initial dictionary to train the RF models have. Data sets should be comma separated arrays with columns: V-Gene Alpha, CDR3 Alpha, J-Gene Alpha, V-Gene Beta, CDR3 Beta, J-Gene Beta. V-Gene Alpha, J-Gene Alpha, V-Gene Beta and J-Gene Beta should contain the associated gene in all capitals, e.g. TRAV1 while CDR3 Alpha and CDR3 Beta should contain the associated amino acid sequence.

**Encode/Generate:** Whether a new or existing dictionary will be used for the final arrays.

**Dictionary name:** Final dictionary location name

**Output names:** Final array names containing the kmer positions of associated data sets

Optional Arguments:

**Length\_dependent\_trimming:** Whether the associated kmers with the CDR3 encoding will be given additional positional encoding. The kmer will be associated with Left, Center-Left, Center, Center-Right, or Right given its position within the CDR3 chain as well as the CDR3 length. The number of associated kmers per position for each CDR3 length is provided in the attached supplementary table.

**CDR12\_trimming: number of aas to trim from both end of CDR12 regions**

**CDR3a\_trimming: number of aas to trim from ends of CDR3a region**

**CDR3b\_trimming: number of aas to trim from ends of CDR3b region**

**Use\_CDR3\_length:** Whether to include CDR3 length as an additional variable in the dictionary

**Kmer\_size:** (default 3), the number of amino acids within each kmer

Outputs:

**Dictionary:** Large list containing all unique extracted kmers from provided data.

**Arrays:** Large array where each column is associated with a kmer of the aforementioned dictionary and each row is an associated TCR encoded into a vector of zeros and ones depending on the presence of kmer in specific TCR

**Step2: Remove chains.**

Inputs:

**Dictionary:** An existing dictionary containing CDR1, CDR2, and CDR3 encoded kmers

**Numpy object:** An existing array that contains columns encoding for the matching dictionary

Optional Arguments:

Outputs:

**Adjusted dictionary:** Dictionary with any CDR1 and CDR2 associated kmers removed

**Adjusted numpy object:** Array with any CDR1 and CDR2 associated columns removed

**Step3: Generating RF models:** Separate the positive and negative TCRs and run RF models on subsampled data so that unbiased, unguided selection of positive and negative TCRs on presence of kmers can be done.

Inputs: **Positive array:** First array submitted will be labeled as positive array, where associated kmers will trend towards a higher final RF score

**Negative array:** Second array submitted will be labeled as negative array, where associated kmers will trend towards a lower final RF score

**N\_estimators:** Number of branch points within each tree

**Save\_model:** Whether this model will be saved

**Model\_name:** Name of output finalized RF model

Arguments:

**Runs:** Number of repeated runs to generate from this specific set of positive and negative TCRs

**Max\_features:** amount of randomly sampled kmers from the base dictionary used to create a branch point separating the positive and negative array.

**Min\_samples\_leaf:** minimum number of final TCRs defining the final branches of the RF model

**Do\_imblance match:** Divides the positive and negative array so that an equal amount of rows is used to generate the RF model

**Split\_num:** What bin of subdivided equal populations of positive and negative TCRs will be used to generate the RF model

Outputs.

**Step4: Prediction:** Generate RF scores for a TCR with CDR1,2 and 3 regions based on the RF models within a specific folder. Will generate a final score between 1 and 0, with 1 trending towards the positive set of the RF model and 0 trending towards the negative set of the RF model.

Inputs: **Combined\_models:** Folder containing RF models

**Label:** Label shared by the RF models within said folder

**Test\_encoded:** An array object containing encoded TCRs. Should correspond with the RF models tested.

**RF\_out:** Name of output text file

Outputs:

**List:** A text list of RF scores assigned in order to submitted test encoded TCRs. Each row corresponds with the specific TCR within the given row of the encoded set.