Chisel Manual

Husam Abdulnabi

Installation

```
pip install poseydon-chisel
```

Requirements

(will automatically install if not satisfied):

'numpy>=1.20.0' , 'pandas>=1.3.0', 'pysam>=0.18.0', 'pybigwig', 'biopython>=1.79', 'joblib>=1.1.0', 'matplotlib>=3.5.0', 'denseweight==0.1.2'

Usage

```
from poseydon_chisel import *
```

Notes

- Sections are not alphabetically sorted to preserve relationships. Use find (ctrl+f) or the navigation pane.
- All arrays are numpy arrays unless otherwise specified

Symbols

s = Sequence .

S = Collection of sequence(s).

z = Sub-sequence in S.

r = Region. sub-location in Z. **rs** = Regions

Arr = Array. A numpy array unless otherwise specified.

V = sub-array / slice

H = Signal(s).

A = Area(s).

C# = Column.

Abbreviations

B*	Boolean (True or False)
Int	Integer
seq	Sequence.
LoL	List of lists.
Bw/bw	Bigwig.
Bg/bg	Bedgraph.
RC	Reverse Complement

OHE	One Hot Encoded
OHERC	OHE + RC.
PWM	Position Weight Matrix.
np	NumPy

Formats

Track	List; each element (**sub-Track**) pertaining to a sub-sequence. $[z_1, z_2 \ldots z_N]$.
Tracks	List of Tracks.
Sequence Track (SeqTrack)	Track; each element a string.
Signal Track (**SigTrack)**	Track; each element an Arr with the shape $(len(z),\ H)$. Signal can be from one or multiple sources. Each value an int or float.
Binary Track (BinTrack)	SigTrack; values are either 0 or 1.
Filter Track (FilterTrack)	BinTrack; 0s are positions to be excluded from analysis.
Profile	A transposed slice of a SigTrack. An Arr with shape $(H,\ len(r))$.
Area	Integration of a Profile. An Arr with shape $(H,\ 1)$.
Markers:	Pandas Arr with shape $(L, 2-4+)$. Each row is a location in the sequence. C1 is the sub-sequence identifier; C2 is the sub-sequence index; C3 (optional) is the strand where 0 is the positive (+) strand and 1 is the negative (-) strand; C-1 (the last column) (optional) is the size of the region, centered on C2. Additional columns (optional) can include genomic attributes.
Nucleotide Array (**NucArr**)	Arr with shape $(len(s),\ 4\ or\ 8)$. Each row is a position in the sequence. C1-4 are 'A', 'C', 'G', 'T'/'U' respectively. C5-8 (optional) is the same as C1-4 but represents the reverse complement.
Pack	Arr with shape $(rs,\ *V.shape)$.
SeqPack	Pack of sequences. Arr with shape $(rs,)$
NucArrPack	Pack of NucArr. Arr with shape $(rs,\ len(s),\ 4\ or\ 8)$.
Profile	A transposed slice of a SigTrack. An Arr with shape $(H,\ len(r))$.
Area	Integration of a Profile. An Arr with shape $(H,\ 1)$.
ProfilePack	Pack of Profiles. Arr with shape $(rs,\ H,\ len(R))$.
AreaPack	Pack of Area(s). Arr with shape $(rs,\ H,\ A)$.
TraitPack	Pack of Trait(s).
FlattenedPack	Arr with shape $(rs,\ F)$, where F is the product of V shape.

Common Arguments

** all arguments that take in lists can also take in 1dim arrays.*

fileloc	Location of the file.

seq	String.
seqs	List of seqs.
BaseSeq (**BS**)	SeqTrack. Reference seq. E.g., a genome where each sub-seq is a chromosome.
BS_ids	List of strings. Ids of each sub-seq. E.g., chromosome names.
select_BS_ids	List of strings or None. **BS_ids** to include in output. If None, all **BS_ids** are used.
BS_sizes	List of ints. Lengths of each sub-seq. I.e. chromosome lengths.
reso	Int. The current resolution of the file/SigTrack.
newreso	Int or None. New resolution of output track.
resomode	Function to convert reso to newreso. Commonly either np.mean or np.max.
Msizes	List or None. Marker size. If None, 1bp will be used. If length of Msizes is 1, this size will be applied to all. Otherwise, must match length of Markers. Could use Markers[:, -1] if sizes column present.
expand_dim	Int or None. Dimension to expand output.
PWM	NucArr.
PWMScores	SigTrack of scores for each PWM window in sub-seqs in BaseSeq.
window	Int or None. Window size.
stride	Int or None. Stride size. Positions to skip for every slide of a window.
Vsize	Int. size of slice to retrieve centered at Marker.
exact	B*. When reso > 1, retrieving a slice can be from a rounded to reso Marker (False) or exact (True). The latter involves interpolating the signal.
Vpad	Int or None. Padding added to slice for exact = True.
stranded	B*. If True, will take strand information to retrieve sequence. Markers must have strand column.
seqmode	None or function used on SeqPack. Can be Seq2OHE or Seq2OHERC.
xlabels	None or List of strings. Labels for x axis.
labels	None or List of strings. Labels for y axis or subplot title.
suptitle	None or string. Title of plot.
figsize	Tuple of Ints. Figure size.
fontsize	Int. font size.
cmap	String. Matplotlib color palette.
threshold	Int or None. If int, values equal to or greater than are taken.
threshsign	Function where the values of are compared to the threshold; threshsign(values, threshold). Can be np.greater, *np.greater_equal*, *np.less*, *np.less_equal*, *np.equal*, *np.not_equal*.
mode	Function.
center	Int or None. If int, will pad the beginning window//2 by the int value such that values from windowing represent the center of the window instead of the start.

extend	Int or None. If int, will extend subTrack after windowing to match the template length by the int value.
skiprows	Int. Number of rows to skip in file.
sep	String. File delimiter. Default is: '\t'.
addcols	List of int(s) or None. Additional columns: columns to add to Markers.
addcols_names	List of string(s) or None. Names for additional columns.

Functions

In the following format:

Function

Description

(Argument [= Default])

Argument	Argument description. Refer to Common Arguments if not included here.

- Return Product(s)

Fasta2Seqs

Converts a fasta file into SeqTrack and ids (optional).

(fileloc, idents = False)

idents	Output fasta description if True.

- SeqTrack; a list of ids if idents = True.

TrackSelect

Selects sub-Track(s) based on sub-seq identifiers.

(Track, BS_ids , select_BS_ids)

- Track

ListLengths

Gets the length of each element in a list.

(x)

x	List.

- List.

Rounder

Rounds a number/numbers to the nearest base.

(x, base = 5)

x	Int or 1dim NumPy Arr

- Int or 1dim NumPy Arr with dtype = int.

Sequence Functions

Seqs2kmer

Finds regions in SeqTrack that match k-mer.

(seqs, k = 6)

k — Int. Size of *k*-mers.

- List of *k*-mers.

KmerFinder

Finds regions in SeqTrack that match kmer.

(SeqTrack, kmer, RC = None, center = None, extend = None)

kmer — String. The pattern to search SeqTrack for. It is case sensitive.

RC — B*. If True, searches the reverse complement of each *k*-mer.

- BinTrack.

Seqs2OHE

Converts list of seqs to OHE.

(Seqs, expand_dim = None, customdict = None)

customdict — dictionary that maps (capital) letter to one hot encoding. If None, default is used.

- OHE.

Seqs2OHERC

Converts a list of seqs to OHERC.

(Seqs, expand_dim = None, customdict = None)

customdict — *see Seqs2OHE*.

- OHERC.

CM2PWM

Converts a count matrix to a PWM.

(CM, pseudo = 1, bkg = [0.25, 0.25, 0.25, 0.25], base = 2):

CM — NucArr. Count Matrix.

pseudo — Int. Pseudo count to be added to each element of the CountMatrix.

bkg — List. Background probabilities of each nucleotide.

base — Int. Base of log for log scaled normalized frequency values.

- PWM.

Seq2PWMScore

Scores PWM fit on seq.

(seq, PWM, pieces = None, RC = True, center = None, extend = None)

pieces	Int or None. Breaks up seq into multiple equal length pieces. They are overlapping so to provide a score for every position in the seq as if 1 piece. Useful for memory consumption.
RC	B*. If True, will score the sequence and its reverse complement and keeps the greater score.

- List of scores for each PWM window in seq.

PWMScorer

Scores PWM on BaseSeq.

(PWM, BaseSeq, BS_ids, select_BS_ids = None, parallel = None, pieces = None, RC = True, center = None, extend = None):

parallel:	None or int. Number of processors to use for parallel multiprocessing. If None, no parallel processing.
pieces	*See Seq2PWMScore*.
RC	*See Seq2PWMScore*.

- SigTrack.

Track Functions

Bw2Track

Converts a .bigwig file into a SigTrack.

(fileloc, BS_sizes, BS_ids , select_BS_ids = None, newreso = 20, resomode = np.mean, window = None)

window	Int or None. maximum size of interval to include score from.

- SigTrack.

Bg2Track

Converts a .bedgraph file into a SigTrack.

(fileloc, BS_sizes, BS_ids , select_BS_ids = None, newreso = 20, resomode = np.mean, window = None, sep = '\t', skiprows = None)

window	*See Bw2Track*.

- SigTrack.

Bam2Track

Converts a .bedgraph file into a SigTrack.

(fileloc, BS_sizes, BS_ids , select_BS_ids = None, newreso = 20, resomode = np.mean, stranded = True, read_length = None, paired = False, make_index = False)

stranded	B*. Reads are mapped to respective strand on a SigTrack. If False, will sum signals to a single signal SigTrack.
read_length	Int or None. If None, read_length is inferred from bam file. If int, read will be extended to read_length strand specifically.
paired	B*. If True, mapped mates will be used to extend the read of the first mate. If False, reads are treated independently and the size by read_length.
make_index	B*. If True, will create a index file for bam file in source directory.

- SigTrack.

TrackInterpolator

Interpolates each individual signal of SigTrack by values that pass a threshold.

(SigTrack, reso, window = None, threshold = 0, threshsign = np.greater)

window	Int or None. If gap between indices greater than threshold is greater than window, gap is not interpolated.

- SigTrack.

Tracks2Track

Stacks a list of SigTracks to a single SigTrack,

(SigTracks)

- SigTrack.

TrackMerger

Merges multi signal SigTrack into a single signal SigTrack.

(SigTrack, mode = np.mean)

- SigTrack.

TrackModifier

Applies a function with a Modifier to SigTrack. Most used for multiplying a SigTrack with a FilterTrack to get a filtered SigTrack.

(SigTrack, modifier, mode = np.multiply)

Modifier	a list that should match length of SigTrack. Each sub-modifier is applied to each subTrack of a SigTrack. Sub-modifiers must be able to broadcast over the subTrack.

- SigTrack.

TrackTransformer

Min-max normalizes or standardizes each individual signal of a SigTrack.

(SigTrack, minmax = (0, None), standardize = False)

minmax — a tuple where the first and second position are minimum and maximum values for normalization. If a value is None, it is found from the SigTrack. Is not used if *standard* is True.

standardize — B*. If True, values will be standardized instead of minmax normalized.

- SigTrack.

LowerResTrack

Lowers the resolution of a SigTrack.

(SigTrack, reso, newreso, resomode = np.mean)

- SigTrack.

TrackExpander

Extends a single signal SigTrack to size of reference sequence by repeating values and adding filler at ends.

(SigTracks, reso, BS_size, BS_ids, select_BS_ids = None, filler = 0)

filler — Int. Adds this value to the end to match reference sequence size.

- SigTrack.

BinTrackInverter

Inverts binary values; 0s become 1s and vice-versa.

(BinTrack)

- BinTrack.

TrackWindower

Applies a function over windows of a SigTrack.

(SigTrack, reso = 1, mode = np.mean, window = None, center = True, extend = True)

window — Int or None. size of window to apply *mode* over. If None, window is equal to reso.

- SigTrack.

TrackThresholder

Finds values in SigTrack that pass a threshold. Can return either a BinTrack or a SigTrack that includes those values that passed while others are zeroed. .

(SigTrack, threshold = 0, threshsign = np.greater, binary = False**)**

binary — B*. If True, will return a BinTrack else will return a SigTrack.

- SigTrack.

TrackPadder

Finds a value in SigTrack and expands it to surrounding values. Useful for expanding 0s or 1s in BinTracks.

(SigTrack, reso = 1, pad = 10, shift = 0, value = 1, base = 0)

pad — Int. number of surrounding bp on either side to expand value to.

shift — Int. can be negative. Number of bp to shift pad window. Useful if padding is not centered.

value — Int or float. Value to be expanded over base.

base — Int or float. Background values.

- SigTrack.

TrackFlagger

Identifies flags in SigTrack.

(SigTrack, reso = 5, window = 1000, stride = 50, flagmode = np.argmax, double = False)

flagmode — Function to be applied to window to find index (*flag*) within window.

double — B*. If True, only keeps best *flag* within a window.

- SigTrack.

Marker Functions

Bed2Markers

Converts a bed file to Markers.

(fileloc, sep = '\t', header = None, skiprows = None, strandcol = None, addcols = None, addcols_names = None)

strandcol — Int or None. Column with strand information.

- Markers

MarkersFilter

Filters Markers by SigTrack values.

(Markers, SigTrack, BS_ids, select_BS_ids = None, reso = 1, exact = False, Msizes = None, threshold = None, threshmode = np.sum, threshsign = np.greater)

reso — Int. SigTrack reso.

- Markers.

Markers2BinTrack

Converts Markers to BinTrack

(Markers, BS_sizes, BS_ids , select_BS_ids = None, Msizes= None, reso = 1, inverse = False)

inverse — B*. If False, Markers are 1s and rest are 0s, else inversed.

reso — Int. reso of output BinTrack.

- BinTrack.

SigTrack2Markers

Produces Markers from SigTrack.

(SigTrack, BS_ids, select_BS_ids = None, reso = 1, select = None, select_mode = None, select_mode_args = None)

select	Int or None. If None, will make Markers from all non-zero values. If specified, will return that number of Markers.
select_mode	Function. If None and select is an int, will randomly select from non-zero values. Currently can be either Top, RevDistro, or Harpoon.
mode_args	Dictionary or None. Arguments for select_mode function, only used if select_mode is not None. If None, will use select_mode defaults.

- Markers.

Top

Returns indices of the largest or smallest values.

(vals, select, smallest = False)

vals	List or 1-dim array of int or floats.
select	Int. Number of indices to return.
smallest	B*. If False, larger values are taken for top/threshold; if True, smaller values.

- Indices.

RevDistro

Returns indices of weighted (DenseWeight) random selections.

(vals, select, alpha = 1.0)

vals	List or 1-dim array of int or floats.
select	Int. Number of indices to return.
alpha	Int or float. The alpha parameter in DenseWeights (Steininger et al., 2021).

- Indices.

Harpoon

Returns indices of a uniformized distribution.

(vals, select, bins = 100)

vals	List or 1-dim array of int or floats.
select	Int. Number of indices to return.
bins	Int. Number of bins to use for binning values.

- Indices.

IdxFlat2Track

Converts list of indices from flattened SigTrack to lists of indices per sub-Track.

(SigTrack, flatidx)

flatidx	List or 1-dim array of int.

- Track Indices.

Txt2Markers

Produces Markers from txt file.

(fileloc, addcols, sep = '\t', header = None, skiprows = None, addcols_names = None)

- Markers.

AddTraits

Transfers Traits from one Markers to another.

(Markers_A, Markers_B, Msizes_A = None, Msizes_B = None)

Markers_A	Markers to transfer traits to.
Markers_B	Markers to transfer traits from.
Msizes_A	Msizes for Markers_A.
Msizes_B	Msizes for Markers_B.

- Markers.

Markers4Packs

Edits Markers for Pack production.

(Markers, select_BS_ids = None, opposite = None, ends = None, BS_ids = None, BS_sizes = None)

opposite	B*. If True, will include the opposite strand for each Marker then removes duplicates.
ends	Int or None. Markers falling within the ends of sub-seq are removed.

- Markers.

Slice, Pack and Split Functions

SeqTrack2Slice

Gets a Slice at Marker of a SeqTrack.

(SeqTrack, Marker, BS_ids, select_BS_ids, Vsize = 1000, seqmode = None, stranded = False)

- Slice.

SeqTrack2Pack

Gets a Pack at Markers of a SigTrack.

(SeqTrack, Markers, BS_ids, select_BS_ids = None, Vsize = 1000, seqmode = None, stranded = False)

- Markers (Filtered and modified), Pack.

SigTrack2Slice

Gets a Slice at Marker of a SigTrack.

(SigTrack, Marker, BS_ids, select_BS_ids = None, Vsize = 1000, reso = 5, newreso = None, resomode = np.mean, exact = True, Vpad = 2, stranded = False)

Vpad	Int. Adds Vpad*reso to the Slice for interpolation (when exact = True).

- Slice.

SigTrack2Pack

Gets a Pack at Markers of a SigTrack.

(SigTrack, Markers, BS_ids, select_BS_ids, Vsize = 1000, reso = 5, newreso = None, resomode = np.mean, areas = None, exact = False, stranded = False)

areas — list of ints. Sizes to integrate the signal over. Largest must be smaller or equal to the Vsize.

- Markers (Filtered and modified), Pack.

LowerResProfilePack

Lowers the resolution of a ProfilePack.

(ProfilePack, reso, newreso, resomode = np.mean)

- ProfilePack.

Traits2Pack

Gets a Pack from traits of a Markers.

(Markers, cols = None, vectorize =None)

cols — list of ints or None. Columns of Markers to retrieve traits from.

vectorize — list of ints or None. Columns of Markers to vectorize. Must be in cols. Vectorize applies pandas get_dummies and returns alphabetically sorted vectorizations.

- TraitPack.

PackShaper

Modifies shape of a Pack.

(Pack, expand_dim = None, squeeze = False, twoD = False, flatten = False)

squeeze — B*. If True, will call np.squeeze to remove 1 length dimensions.

twoD — B*. If True, will reshape pack into 2 dimensions of (observations, features).

flatten — B*. If True, will call np.flatten to convert pack to 1 dimension.

- Pack.

PackShaper

Concatenates Packs of the same dimensions and the same first dimension.

(Packs)

Packs — List of packs. Must be of the same dimensions and the same first dimension.

- Pack.

Splitter

Generates Splits.

(X, num_splits = 3, proportions = [0.7, 0.3], random = False, cut = True)

X	Int. Length of Markers.
num_splits	Int. Number of Split to generate.
proportions	List of int or float. Proportion of indices in each sub-split.
random	B*. When True, will randomize the distribution of indices.
cut	B*. When True, will apply cut method like card shuffling.

- Splits.

PackSplit

Divides a Pack based on a Split.

(Pack, Split)

- List of Packs for each sub-split.

Visuals Functions

GridPlot

(Arr, bounds = (None, None), xlabels = None, labels = None, suptitle = None, figsize = (15,5), fontsize = 10, cmap = 'viridis')

bounds	Tuple of Int or None. Sets the lower and upper limit for coloring respectively.

- GridPlot

FilledLinePlot

(Arr, bounds = (None, None), xlabels = None, labels = None, suptitle = None, figsize = (15,5), fontsize = 10, cmap = 'viridis')

bounds	Tuple of Int or None. Sets the lower and upper limit y-axis.

- FilledLinePlot

LinePlot

(Arr, bounds = (None, None), xlabels = None, labels = None, suptitle = None, figsize = (15,5), fontsize = 10, cmap = 'viridis')

bounds	Tuple of Int or None. Sets the lower and upper limit y-axis.

- LinePlot

Track2Visual

Produces visual of a Slice in SigTrack.

(SigTrack, Marker, BS_ids, select_BS_ids = None, reso = 5, Vsize = 2000, exact = False, vismode = GridPlot, bounds = (None, None), xlabelsnum = 3, labels = None, suptitle = None, figsize = (15,5), fontsize = 10, cmap = 'viridis')

- vismode

ProfilePack2Visual

Creates a visual from ProfilePack.

(ProfilePack, combomode = np.mean, vismode = GridPlot, Vsize = 1000, bounds = (None, None), xlabelsnum = 3, labels = None, suptitle = None, figsize = (15,5), fontsize = 10, cmap = 'viridis')

combomode Function to use to collapse profile signals to a 2D array. Either np.mean or np.sum.

- vismode

Common Internal Variables

resoratio = newreso // reso

z = sub-seq

ind = index

m = mask, ms = masks

T = Track, nT = new Track

I = interpolated

r = region

st = strand

l* = len(*)

si = list of sizes

sp = list of spacing

t = threshold

V = slice

ud = updown

c = cent = center

iro, ro = index, row for pandas iterrows