TreeVal

Usage Cheatsheet

Input YAML

```
assembly:
 assem_level: {scaffold|chromosome}
 assem_version: {Version number of the assembly}
 sample_id: {Name of the assembly}
 latin_name: {Scientific Name}
 defined_class: {User defined class of the input assembly}
 project_id: {Project ID of assembly} #Optional
reference_file: {Path to .f{a|n|asta}{.gz} formatted input}
map_order: length
assem_reads:
 read_type: hifi
 read_data: {Folder containing longread data in .fasta.gz
format}
hic_data:
 hic_cram: {Folder containing HiC reads in cram format with
.crai}
 hic_aligner: {minimap2|bwamem2}
kmer_profile:
 kmer_length: {Default to 31}
 dir: {Path to pre-existing FKPROF files if they exist}
#Optional
alignment:
 data_dir: {Path to the top level gene_alignment_data folder}
 geneset_id: {A csv delimited list of geneset data to align}
self_comp:
 motif_len: 0
 mummer_chunk: 10
intron:
 size: "50k"
telomere:
 teloseq: {The expected telomeric sequence}
synteny:
 synteny_path: {Folder of FASTA files used for alignments}
 synteny_genomes: {Specify files in the above path}
busco:
 lineages_path: {Path to the busco database e.g. /busco/v5}
 lineage: {The odb10 lineage to use}
```

Run:

```
nf-core run sanger-tol/treeval -r 1.1.0 –input TreeVal.yaml –
profile {singularity|docker} –output {OUTDIR} –entry RAPID
```

or

```
nf-core run sanger-tol/treeval -r 1.1.0 –input TreeVal.yaml –
profile {singularity|docker} –output {OUTDIR}
```

Output files

```
{OUTDIR}
├── treeval_upload/
│  ├── my.genome                # GENERATE_GENOME*
│  ├── coverage.bw              # READ_COVERAGE
│  ├── coverage_log.bw          # READ_COVERAGE
│  ├── *_repeat_density.bw      # REPEAT_DENSITY
│  ├── *gap.bed.gz              # GAP_FINDER
│  ├── *gap.bed.gz.tbi          # GAP_FINDER
│  ├── *telomere.bed.gz + .tbi # TELO_FINDER
│  ├── *_buscogene.bigbed       # BUSCO_ANALYSIS*
│  ├── *_ancestral.bigbed       # BUSCO_ANALYSIS*
│  ├── *.gff.gz + .tbi          # GENE_ALIGNMENT-PEPTIDE*
│  ├── *_cdna.bigBed            # GENE_ALIGNMENT-NUCLEAR*
│  ├── *_cds.bigBed             # GENE_ALIGNMENT-NUCLEAR*
│  ├── *_rna.bigBed             # GENE_ALIGNMENT-NUCLEAR*
│  ├── BSPQ1.bigBed             # INSILICO_DIGEST*
│  ├── BSSS1.bigBed             # INSILICO_DIGEST*
│  ├── DLE1.bigBed              # INSILICO_DIGEST*
│  ├── *_selfcomp.bigBed        # SELFCOMP*
│  ├── *.paf                    # SYNTENY*
│  ├── *.ref.spectra-cn.ln.png  # KMER*
│  ├── *_{kmer_size}_.bw        # KMER_COVERAGE
│  └── punchlists
│    ├── halfcoverage.bigbed    # READ_COVERAGE
│    ├── zerodepth.bigbed       # READ_COVERAGE
│    ├── maxdepth.bigbed        # READ_COVERAGE
│    ├── *_pep_punchlist.bed    # GENE_ALIGNMENT-PEPTIDE*
│    ├── *_cdna_punchlist.bed   # GENE_ALIGNMENT-NUCLEAR*
│    ├── *_cds_punchlist.bed    # GENE_ALIGNMENT-NUCLEAR*
│    └── *_rna_punchlist.bed    # GENE_ALIGNMENT-NUCLEAR*
├── hic_files
│  ├── *gap.bed                 # GAP_FINDER
│  ├── *_repeat_density.bw      # REPEAT_DENSITY
│  ├── *.mcool                  # HIC_MAPPING
│  ├── *_pretext_normal.pretext # HIC_MAPPING + PRETEXT_INGESTION
│  ├── *_pretext_hr.pretext     # HIC_MAPPING + PRETEXT_INGESTION
│  └── *telomere.bed            # TELO_FINDER
└── pipeline_info
   ├── TreeVal_Runs*.txt        # TreeValProject.Summary (Groovy Function)
   ├── execution*{html|txt}     # STANDARD OUTPUT
   ├── pipeline*{html|txt}      # STANDARD OUTPUT
   └── software_versions.yml # STANDARD OUTPUT
```