Bio-related utilities

- Bio file utilities
  - Convert bed12 to gtf
  - Convert gtf to bed12
  - Merge bed12 files and remove duplicate entries
  - Merge fasta files and remove duplicate sequences
  - Get read length distributions
- Bio plotting utilities
  - Plotting read length distributions
- Other utilities
  - Download reads from the Short Read Archive

Bio File Utilities

Convert bed12 to gtf

Convert a bed12 file into an equivalent gtf file. In particular, it uses the `thick_start` and `thick_end` fields to determine the CDS gtf entries. It only creates `exon` and `CDS` gtf entries. The bed columns after the standard 12 are included as attributes in the gtf file. The `id` field in the bed file is used for the `transcript_id` attribute to link exons and CDs gtf entries. The output is sorted by `seqname`, `start` and `end`.

```
bed12-to-gtf <bed> <out> [-s/--source <source>] [-p/--num-cpus <num_cpus>] [--add-gene-id]
```

Command line options

- `bed`. The bed12 file. It must conform to the style expected by `misc.bio_utils.bed_utils`.
- `out`. The output gtf file. It will conform to the style dictated by `misc.bio_utils.gtf_utils`.
- `--source`. A string to use for the `source` column in the gtf file. Default: "`.`"
- `--num-cpus`. The number of parallel processes to use to split the bed entries. The operation of converting from exon blocks to genome-coordinate gtf entries is not optimized and can be somewhat slow.

- `--add-gene-id`. If this flag is given, then the `id` field will be used as the `gene_id`.

Convert gtf to bed12

Convert a gtf file into an equivalent bed12 file. It created bed entries based on the `exon` features and `transcript_id` field. It then uses the `CDS` features to determine the `thick_start` and `thick_end` values for the bed file.

```
gtf-to-bed12 <gtf> <out>  [--chr-name-file <chr_name_file>] [--exon-feature <exon_feature>]
```

Command line options

- `gtf`. The gtf file
- `out`. The bed12 file
- [`--chr-name-file`]. If given, the order in this file will be used to sort the entries' `seqname`s. Presumably, this will be the `chrName.txt` file created by STAR. Default: `seqname`s are sorted alphabetically.
- [`--exon-feature`]. The `feature`s to count as "exons" for determing the transcript structures. Default: `exon`
- [`--cds-feature`]. The `feature`s to count as "coding" regions for determining the "thick" parts of the bed entries. Default: `CDS`

Merge bed12 files and remove duplicate entries

The `remove-duplicate-bed-entries` script concatenates a list of bed12+ files and removes the redundant entries. It uses the following fields to identify duplicates:

- `seqname`
- `start`
- `end`
- `strand`
- `num_exons`
- `exon_lengths`
- `exon_genomic_relative_starts`

All of those fields must match exactly for two entries to count as "duplicates". The precedence among duplicates is arbitrary. The output is sorted by `seqname`, `start`, `end` and `strand`.

```
remove-duplicate-bed-entries <bed_1> [<bed_2> ...] -o/--out <out> [--add-gene-id] [--compres
```

Command line options

- `bed_i`. The input bed12+ files.
- `out`. The output non-redundant bed12+ file
- [`--add-gene-id`]. If this flag is given, then the `id` field will be used as the `gene_id` for the transcript.
- [`--compress`]. If this flag is given, then the output will be gzipped. **N.B.** The extension of `out` will not be changed, so it should already include the `gz`.

Merge fasta files and remove duplicate sequences

The `remove-duplicate-sequences` script merges a list of fasta files and removes the redundant sequences. “Redundant” here means “exactly the same.” There is no approximate string matching, etc.

```
remove-duplicate-sequences <fasta_1> [<fasta_2> ...] -o/--out <out> [--compress] [-l/--lowe
```

Command line options

- `fasta_i`. The input fasta files.
- `out`. The output fasta file
- [`--compress`]. If this flag is given, then the output will be gzipped. **N.B.** The extension of `out` will not be changed, so it should already include the `gz`.
- [`--lower-precedence-re`]. If this is given, then identifers which *no not* match this regular expression will be kept when two sequences are found to be duplicates. For example, this could be used to prefer identifiers based on Ensemble annotations rather than *de novo* assemblies (e.g., the re could be “`TCONS.*`”).

Merge isoforms

Merge groups of gtf features into non-overlapping entries. If the groups are based on merging CDSs of genes, then this operation is equivalent to merging all transcript isoforms of that gene into a “super” transcript which comprises all isoforms of the gene.

```
merge-isoforms <gtf> <out> [--feature-type <feature_type>] [--group-attribute <group_attrib
```

Command line options

- `gtf`. The gtf file
- `out`. The (output) gtf file with specified features of the specified groups merged.
- [`--feature-type`]. The type of `feature` (third column) to merge. Default: `CDS`. Other reasonable choices: `exon`
- [`--group-attribute`]. The attribute used to create the groups. Default: `gene_id`. Other reasonable choices: `transcript_id`, `gene_name`.
- [`--id-format-str`]. A python string to use for the identifiers. The first `{}` will be replaced with the value from the `group_attribute`. Default: `{}.merged`.
- [`--chr-name-file`]. If given, the order in this file will be used to sort the entries' `seqname`s. Presumably, this will be the `chrName.txt` file created by STAR. Default: `seqname`s are sorted alphabetically.
- [`--add-exons`]. If this flag is given, then all features will be duplicated but with the `feature` type `exon`. Presumably, this should be given when `CDS` features are merged and the resulting gtf file will be used by STAR (or anything else expecting `exon`s).

Get read length distributions

Count the number of unique reads in a set of files. All of the files must have the same type (see below for valid types). In the case of bam files, the script only counts primary alignments. Thus, it does not double-count multimappers, and unmapped reads are not included in the distribution.

```
get-read-length-distribution <file_1> [<file_2> ...] -o/--out <out> [-f/--file-type <file_ty
```

Command line options

- `file_i`. The files for which read length distributions will be found.
- `out`. The output (csv.gz) file which will contain the length and counts of each file. In particular, it will have the columns: `basename`, `length`, `count`, where `basename` is the name of the file, excluding the final extension.
- [`--file-type`]. The type of the files. All files must be of the same type. If `AUTO` is given, then the type is guess based on the extension of the first file. Please use the `--help` flag to see more information about how the file types are guessed. Default: `AUTO`. Choices: `AUTO`, `bam`, `fasta` or `fastq`

Bio plotting utilities

Plotting read length distributions

Create bar charts of the length distributions created by `get-read-length-distributions`.

```
plot-read-length-distribution <length_distribution> <basename> <out> [--title <title>] [--mi
```

Command line options

- `length_distribution`. The file created by `get-read-length-distributions`
- `basename`. The “basename” of the sample to plot, as given in `length_distribution`. Alternatively, `ALL` can be given, and the plot will include all of the samples as a factor plot.
- `out`. The output filename. The extension should be something which matplotlib can interpret, such as “pdf” or “png”.
- [`--title`]. An optional title included at the top of the plot.
- [`--{min,max}-read-length`]. Optionally, reads lengths above or below the given thresholds will not be shown. Default: All read lengths are shown.
- [`--ymax`]. The maximum for the y-axis in the bar charts. Default: the maximum values will be selected based on the maximum count in the data.
- [`--fontsize`]. The size of the fonts in the plots.

Other Utilities

Download reads from the Short Read Archive

The `download-srr-files` script can be used to retrieve sequencing runs from the SRA (or ENA). It can download from either the NCBI or EBI sites. Primarily, the script needs the `SraRunInfo.csv` file for the relevant “Bio Project”. In particular, the “Run” accessions are required for downloading files.

The easiest (maybe) way to find this file is as follows.

1. Browse to the BioProject page on NCBI or EBI.
2. Select “SRA” from the “Related Information” box.
3. Check the boxes for all of the desired samples.
4. Click the “Send to:” link at the bottom of the page.

5. Choose “File” as the destination and “RunInfo” as the format.
6. Click “Create File”.

In addition to downloading the sequence files in `sra` format, the script extracts them to `fastq.gz` files and removes the `sra` files.

This script requires the `fastq-dump` program from the SRA toolkit to be in the `$PATH`.

The main advantage of this script compared to the Aspera client `ascp` used by default by the SRA toolkit is that it works over standard FTP. Many firewalls block non-standard ports, so `ascp` can have difficult connecting from many (or, at least, my) university, etc. settings.

Also, by using the SraRunInfo.csv (or similar) file, lengthy command line calls can be avoided, which improves reproducibility.

```
download-srr-files <run_info> <outdir> [-a/--accession-field <accession_field>] [-p/--pairec
```

Command line options

- `run_info`. The SraRunInfo.csv file, or any other file which includes the run accessions. These are typically of the form “SRR…”
- `outdir`. The location for the `fastq.gz` files. This path should already exist.
- [`--accession-field`]. The field (column) containing the run accessions. Default: “Run”
- [`--paired-field`]. The field indicating whether the sample is paired-end. Default: “LibraryLayout”
- [`--paired-values`]. The values in `paired_field` which indicate that the sample is paired-end. Default: [“PAIRED”]
- [`--source`]. The remote server from which the files will be downloaded. Default: “ebi”
- [`--overwrite`]. By default, files which already exist will not be downloaded again. If this flag is present, all files will be re-downloaded.
- [`--sep`]. The separator in the `run_info` file. Default: “͡”
- [`--num-cpus`]. The number of simultaneous connections. Each connection runs as a separate process. Default: 1