main

July 14, 2021

1 Prepare GCT files for the gtex eqtl pipeline

Apuã Paquola

Edited by Kynon J Benjamin

- Inputs:
  - raw counts
  - sample table
- Outputs:
  - GCT files of counts and tpm for selected samples and genes
  - A lookup table of sample_ids and brain_ids
  - A list of chromosomes to use

```python
[1]: import pandas as pd
```

```python
[2]: def to_gct(filename, df):
         description_df = pd.DataFrame({'Description': df.index.values}, index=df.
     ↪index)
         dfo = pd.concat([description_df, df], axis=1)
         dfo.index.name = 'Names'
         with open(filename, "wt") as out:
             print("#1.2", file=out)
             print(df.shape[0], df.shape[1], sep="\t", file=out)
             dfo.to_csv(out, sep="\t")
```

1.1 Load data

```python
[3]: fam_df = pd.read_csv("/ceph/projects/v4_phase3_paper/inputs/genotypes/_m/
     ↪LIBD_Brain_TopMed.fam",
                          sep="\t", header=None, names=["BrNum", "ID", "V2", "V3",␣
     ↪"V4", "V5"])
     pheno_df = pd.read_csv("/ceph/projects/v4_phase3_paper/inputs/phenotypes/_m/
     ↪merged_phenotypes.csv")
     pheno_df = pheno_df[(pheno_df["Dx"].isin(["CTL", "SZ"])) &
                         (pheno_df["Race"].isin(["AA", "EA"])) &
                         (pheno_df["Age"] > 13)].copy()
     pheno_df["ids"] = pheno_df.RNum
     pheno_df.set_index("ids", inplace=True)
```

```python
tpm_df = pd.read_csv("/ceph/projects/v4_phase3_paper/inputs/counts/
 ↪text_files_counts/tpm/_m/hippocampus/transcript/tpm.csv", index_col=0)
counts_df = pd.read_csv("/ceph/projects/v4_phase3_paper/inputs/counts/
 ↪text_files_counts/_m/hippocampus/tx_counts.txt",
                          sep="\t", index_col=0)
```

1.2 Select individuals

```python
[4]: samples_rnum = list(set(pheno_df.index).intersection(set(counts_df.columns)))
samples = list(set(pheno_df.loc[samples_rnum,:].BrNum).intersection(set(fam_df.
 ↪BrNum)))
new_fam = fam_df[(fam_df["BrNum"].isin(samples))].
 ↪drop_duplicates(subset="BrNum")
new_fam.to_csv("keepFam.txt", sep='\t', index=False, header=False)
new_fam.shape
```

```python
[4]: (394, 6)
```

```python
[5]: new_pheno = pheno_df.loc[(pheno_df.RNum.isin(samples_rnum)), ["RNum", "BrNum"]]\
                    .reset_index().set_index("BrNum")\
                    .loc[new_fam.BrNum].reset_index().set_index("ids")
print(new_pheno.shape)
new_pheno.head(2)
```

```
(394, 2)
```

```python
[5]:        BrNum   RNum
     ids
     R5527  Br822   R5527
     R2725  Br823   R2725
```

```python
[6]: interaction_df = pheno_df.loc[(pheno_df.RNum.isin(samples_rnum)), ["RNum",
 ↪"BrNum", "Sex"]]\
                            .reset_index().set_index("BrNum")\
                            .loc[new_fam.BrNum]
interaction_df["Sex"] = interaction_df.Sex.astype("category").cat.codes
interaction_df.loc[:, ["Sex"]].to_csv("sex_interaction_list.txt", sep='\t')
```

1.3 Select genes

```python
[7]: genes = list(set(counts_df.index).intersection(set(tpm_df.index)))
len(genes)
```

```python
[7]: 197311
```

1.4 Output files

```
[8]: to_gct("counts.gct", counts_df.loc[genes,new_pheno.index])
     to_gct("tpm.gct", tpm_df.loc[genes,new_pheno.index])
     new_pheno.loc[:, ["RNum", "BrNum"]].to_csv("sample_id_to_brnum.tsv", sep="\t",
      ↪index=False)
```

```
[9]: pd.DataFrame({'chr':['chr'+xx for xx in [str(x) for x in range(1,23)] +
      ↪['X']]})\
       .to_csv('vcf_chr_list.txt', header=False, index=None)
```

```
[ ]:
```