Mappings

July 3, 2017

1 Mapping Objects

Create mappings (hash tables / dictionaries) used in downstream analyses.

Inputs

- TCGA/GTEx Metadata
  - syn9962462
- ENSEMBLE Genes
  - syn10156423

```
In [6]: import pickle
        import os

        import pandas as pd
        from mygene import MyGeneInfo
```

```
In [7]: df = pd.read_csv('inputs/tcga_gtex_metadata_intersect.tsv', index_col=0, sep='\t')
```

1.1 Tissue Map

Map samples to tissues

```
In [8]: tissue_map = {}
        for sample in df.index:
            tissue_map[sample] = df.loc[sample].tissue
        with open('pickles/tissue_map.pickle', 'wb') as f:
            pickle.dump(tissue_map, f)
```

1.2 Type Map

Map samples to "type", which for TCGA is disease and for GTEx is long-form tissue

```
In [9]: type_map = {}
        for sample in df.index:
            type_map[sample] = df.loc[sample].type
        with open('pickles/type_map.pickle', 'wb') as f:
            pickle.dump(type_map, f)
```

1.3 Gene Map

Maps ENSMBL Gene IDs to Gene names

```
In [10]: genes = [x.strip() for x in open('inputs/ENS_genes.txt', 'r').readlines()]

In [11]: mg = MyGeneInfo()

In [12]: gene_map = {}
         unmapped_genes = []
         for gene in genes:
             g = gene.split('.')[0]  # remove ENS version tag
             q = mg.query(g)
             if q['hits']:
                 h = q['hits']
                 if len(h) > 2:
                     print h
                     break
                 else:
                     gene_map[g] = h[0]['symbol']
             else:
                 unmapped_genes.append(g)
                 gene_map[g] = g
         print '{} genes unmapped of {} total genes.'.format(len(unmapped_genes), len(gene_map))
         print '{}% Mapped'.format((len(gene_map) - 212) *1.0 / len(gene_map))

212 genes unmapped of 19797 total genes.
0.989291306764% Mapped

In [13]: with open('pickles/gene_map.pickle', 'wb') as f:
             pickle.dump(gene_map, f)
```