mutations_context_distribution

August 26, 2018

1 Analizing the contexts in which mutations occur

Not all mutations are the same and here we analyze how are the common mutation consequences distributed,

```
In [20]: %matplotlib inline

         import matplotlib.pyplot as plt
         import seaborn as sns

         sns.set()
```

We first map genes to the number of mutations they harbor (read from a random sample of 100,000 mutations)

```
In [9]: from collections import Counter
        from ICGC_data_parser import SSM_Reader

        consequences = Counter()

        mutations = SSM_Reader(filename='data/ssm_sample.vcf')
        CONSEQUENCE = mutations.subfield_parser('CONSEQUENCE')

        for record in mutations:
            consequence_types = [c.consequence_type for c in CONSEQUENCE(record)]
            consequences.update(consequence_types)

        sorted(consequences.items(),
               key=lambda a: a[1],
               reverse=True )
```

```
Out[9]: [('intron_variant', 179634),
         ('intergenic_region', 51253),
         ('downstream_gene_variant', 25290),
         ('upstream_gene_variant', 24513),
         ('missense_variant', 4898),
         ('exon_variant', 3922),
         ('synonymous_variant', 2204),
```

```
('3_prime_UTR_variant', 1933),
('splice_region_variant', 494),
('5_prime_UTR_variant', 425),
('stop_gained', 303),
('frameshift_variant', 292),
('splice_acceptor_variant', 117),
('splice_donor_variant', 110),
('5_prime_UTR_premature_start_codon_gain_variant', 69),
('intragenic_variant', 16),
('inframe_deletion', 15),
('disruptive_inframe_deletion', 11),
('stop_retained_variant', 5),
('disruptive_inframe_insertion', 4),
('start_lost', 3),
('stop_lost', 3),
('inframe_insertion', 2)]
```