Retrieve Taxonomy

April 15, 2019

0.1 Crea tassonomia

- Scegli una qualsiasi istanza ad un rank qualsiasi della tassonomia
- Trova tutti i suoi nodi discendenti, includendo i nodi intermedi.
- Crea un dataframe con un record per ogni discendente con i seguenti attributi:
  - tax id (identificativo univoco in NCBI)
  - rank
  - tax id del nodo padre
  - lineage
  - una colonna per ogni livello tassonomia allineata

```
In [2]: from ete2 import NCBITaxa
        from Bio import Entrez
        from collections import OrderedDict
        import pandas as pd
        import numpy as np
        import json
        import re
```

```
In [3]: ncbi = NCBITaxa()
```

Update taxonomy database, might takes a few minutes...

```
In [4]: #ncbi.update_taxonomy_database()
```

Insert the root of the taxonomy to start retrieving information from

```
In [5]: organism = "Mollusca"
```

Main (general) taxonomy of reference Used to align all retrieved organisms to a common lineage

```
In [6]: TAXONOMY = ("root", # vitae
                    "domain",
                    "superkingdom", "kingdom",
                    "phylum", "subphylum", # or division, "subdivision"
                    "class", "subclass", "infraclass",
                    "superorder", "order", "suborder", "infraorder",
                    "superfamily", "epifamily", "family", "subfamily", "infrafamily",
                    "tribe", "subtribe", "infratribe",
                    "genus", "subgenus",
                    "species", "subspecies"
                   )
```

```
In [7]: taxid2name = ncbi.get_name_translator([organism])
        taxid2name
```

```
Out[7]: {'Mollusca': [6447]}
```

```
In [8]: organism_taxid = taxid2name[organism][0]
        organism_taxid
```

```
Out[8]: 6447
```

Available methods - NCBITaxa.get_rank() - NCBITaxa.get_lineage() - NCBITaxa.get_taxid_translator() - NCBITaxa.get_name_translator() - NCBITaxa.translate_to_names()

```
In [9]: descendants = ncbi.get_descendant_taxa(organism, intermediate_nodes=True)
        print("Alcuni discendenti di {} sono:\n{}".format(organism,
                                                          "\n".join(ncbi.translate_to_names(des
```

```
Alcuni discendenti di Mollusca sono:
Cancellaria reticulata
Vespericola sp. 5 MG-2018
Brocchinia clenchi
Planorbella subcrenata
Marcia recens
Ardeadoris scottjohnsoni
Ardeadoris cf. scottjohnsoni SU-2008
Delectopecten fosterianus
Lima zealandica
Veprichlamys jousseaumei
```

```
In [10]: print("Ci sono {} nodi nella tassonomia dei {}".format(len(descendants), organism))
```

```
Ci sono 31675 nodi nella tassonomia dei Mollusca
```

```
In [11]: ancestor_ranks = ncbi.get_lineage(organism_taxid)
         ancestor_ranks
```

```
Out[11]: [1, 131567, 2759, 33154, 33208, 6072, 33213, 33317, 1206795, 6447]
```

Build dictionary of taxid with its rank

```
In [12]: # Rank of every organism fetched, including ancestors
         full_ranks = ncbi.get_rank(ancestor_ranks + descendants)
         full_ranks[1] = u'root' # if not it is 'no rank'

         #ranks = ncbi.get_rank(descendants)
         ranks = ncbi.get_rank(descendants + [organism_taxid])# include self
```

Dictionary of ranks is structured this way

```
{
    ...
     395969: 'no rank',
     1051332: 'species',
     1813622: 'species',
     2759: 'superkingdom',
     1813623: 'species',
     1813625: 'species',
     1813626: 'species',
     2231019: 'no rank',
     87862: 'superfamily',
     2053944: 'species',
     87865: 'genus',
     87866: 'species',
     87867: 'genus',
     2053948: 'species',
     87869: 'genus',
     87870: 'species',
     87871: 'superfamily',
     87872: 'superfamily',
    ...
 }
```

Build dictionary of taxid with its name

```
In [13]: taxid_translator = {}
         for taxid in full_ranks:
             taxid_translator[taxid] = ncbi.get_taxid_translator([taxid])[taxid]
```

Dictionary of taxid_translator is structured is this way:

```
{
    ...
    395969: 'unclassified Protobranchia',
    1051332: 'Galba pervia',
    765046: 'Provanna laevis',
    2759: 'Eukaryota',
    765047: 'Provanna macleani',
    765049: 'Provanna variabilis',
    765050: 'Provanna sculpta',
    2231019: 'unclassified Galeommatidae',
    87862: 'Helicoidea',
    2053944: 'Pterygioteuthis sp. DP0009X',
    87865: 'Coniglobus',
    87866: 'Coniglobus mercatorius',
    1267515: 'Lasaea sp. LHK07',
    1267516: 'Lasaea sp. LHK06',
    87869: 'Satsuma',
    87870: 'Satsuma japonica',
    1267519: 'Lasaea sp. LHK04',
    87872: 'Polygyroidea',
    ...
}
```

Create a dictionary of ordered lineage for each taxid It is structured in the following way: TAXID: {RANK_TASSONOMICO: ISTANZA} for each rank e.g.

```
"1441792": <-- tax_id
{
    'root' : 'root',
    'sub_root' : 'cellular organisms',
    'superkingdom' : 'Eukaryota',
    'sub_superkingdom' : 'Opisthokonta',
    'kingdom' : 'Metazoa',
    'sub_kingdom' : 'Eumetazoa',
    'sub_kingdom_1' : 'Bilateria',
    'sub_kingdom_2' : 'Protostomia',
    'sub_kingdom_3' : 'Lophotrochozoa',
    'phylum' : 'Mollusca',
    'class' : 'Bivalvia',
    'subclass' : 'Protobranchia',
    'sub_subclass' : 'unclassified Protobranchia',
}
```

NOTE: Taxonomonic groups that start with "sub_" were originally assigned a 'no rank' value in NCBI database. We name them after their father altough it could also be the case that the most appropriate name is for example 'sovra_' and the son's name (like sovra_class and not sub_phylum) but we use the simpler and faster approach.

The following code uses a json file as support to write the taxonomy, otherwise for very large taxonomy (kingdom and above) storing everything in a dicitonary would be too much and the kernel would crash

```
In [14]: # with open(organism + '_lineageTaxonomy.json', 'a') as file:
         #     for taxid, rank in ranks.items():
         #         taxid_lineage = {}
         #         taxid_lineage[taxid] = OrderedDict()
         #         count_consecutive_noranks = 1 # e' il primo no rank consecutivo --> e' la p
         #         was_norank = False
         #         for i, ancestor_id in enumerate(ncbi.get_lineage(taxid)):
         #             lineage_level_name = taxid_translator[ancestor_id] # u'Teuthida, u'Ceph
         #             lineage_instance = full_ranks[ancestor_id] # order, suborder, ...
         #             # do not override no rank keys !
         #             if lineage_instance == u'no rank': # first instance is never no rank, e
         #                 # if the previous ancestor is not on the same level then reset coun
         #                 if not was_norank:
         #                     lineage_instance = u'sub_' + taxid_lineage[taxid].items()[i-1][
         #                 else:
         #                     # take the upper common ancestor
         #                     lineage_instance = u'sub_{}_{}'.format(taxid_lineage[taxid].ite
         #                                                                         count_consecutive_noran
         #                     count_consecutive_noranks += 1
         #                 was_norank = True
         #             else:
         #                 count_consecutive_noranks = 1
         #                 was_norank = False
         #             taxid_lineage[taxid][lineage_instance] = lineage_level_name # set e.g.
         #         json.dump(taxid_lineage, file, indent = 4)
```

This version is for lighter taxonomies (that is for lower rank organisms) and only uses a dictionary (no json file)

```
In [15]: # PSEUDOCODICE
         #inizializza dizionario
         #per ogni ancestor
         #    inizializza dizionario ordinato di quell'ancestor
         #    count_consecutive_noranks = 1
         #    se ho un no_rank
         #        se was_norank == False
         #            chiamalo sotto_livello
         #        altrimenti
         #            chiamalo sotto_livello_${count_consecutive_noranks}
         #            count_consecutive_noranks += 1
         #        was_norank = True
         #    altrimenti (se non ho un no_rank)
         #        was_norank = False

         taxid_lineage = {}
         for taxid, rank in ranks.items():
             taxid_lineage[taxid] = OrderedDict()
             count_noranks = 0
             consecutive_noranks = 1 # e' il primo no rank consecutivo --> e' la prima volta c
             was_norank = False
             for i, ancestor_id in enumerate(ncbi.get_lineage(taxid)):
                 lineage_level_name = taxid_translator[ancestor_id] # u'Teuthida, u'Cephalopod
```

```
        lineage_instance = full_ranks[ancestor_id] # order, suborder, ...
        # do not override no rank keys !
        if lineage_instance == 'no rank': # first instance is never no rank, else cod
            # if the previous ancestor is not on the same level then reset counter
            if not was_norank:
                lineage_instance = u'sub_' + taxid_lineage[taxid].items()[i-1][0]  # t
            else:
                # take the upper common ancestor
                lineage_instance = u'sub_{}_{}'.format(taxid_lineage[taxid].items()[i
                                                        consecutive_noranks)
                consecutive_noranks += 1
            was_norank = True
        else:
            consecutive_noranks = 1
            was_norank = False
        taxid_lineage[taxid][lineage_instance] = lineage_level_name # set e.g. {u'sup
#taxid_lineage
```

0.1.1 Globally align taxonomy columns

Manual ordering is not 'interoperable'! Try automated ordering instead, based on the taxonomy declared at the beginning.

```
In [16]: # First create the dataframe
         dd = pd.DataFrame.from_dict(taxid_lineage, orient = "index")
         dd.iloc[np.r_[0:6, -6:0]]
```

```
Out[16]:          root            sub_root superkingdom sub_superkingdom  kingdom  \
         6447     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6448     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6451     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6452     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6453     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6454     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558352  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558353  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558354  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558355  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558356  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558877  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa

                 sub_kingdom sub_kingdom_1 sub_kingdom_2   sub_kingdom_3    phylum  \
         6447      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
         6448      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
         6451      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
         6452      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
         6453      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
         6454      Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
```

```
2558352   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
2558353   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
2558354   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
2558355   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
2558356   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca
2558877   Eumetazoa     Bilateria   Protostomia  Lophotrochozoa  Mollusca

         ... sub_suborder sub_genus sub_subfamily sub_species sub_superfamily  \
6447     ...          NaN       NaN           NaN         NaN             NaN
6448     ...          NaN       NaN           NaN         NaN             NaN
6451     ...          NaN       NaN           NaN         NaN             NaN
6452     ...          NaN       NaN           NaN         NaN             NaN
6453     ...          NaN       NaN           NaN         NaN             NaN
6454     ...          NaN       NaN           NaN         NaN             NaN
2558352  ...          NaN       NaN           NaN         NaN             NaN
2558353  ...          NaN       NaN           NaN         NaN             NaN
2558354  ...          NaN       NaN           NaN         NaN             NaN
2558355  ...          NaN       NaN           NaN         NaN             NaN
2558356  ...          NaN       NaN           NaN         NaN             NaN
2558877  ...          NaN       NaN           NaN         NaN             NaN

        tribe sub_phylum_1 sub_superorder sub_order_1 sub_subclass_3
6447      NaN          NaN            NaN         NaN            NaN
6448      NaN          NaN            NaN         NaN            NaN
6451      NaN          NaN            NaN         NaN            NaN
6452      NaN          NaN            NaN         NaN            NaN
6453      NaN          NaN            NaN         NaN            NaN
6454      NaN          NaN            NaN         NaN            NaN
2558352   NaN          NaN            NaN         NaN            NaN
2558353   NaN          NaN            NaN         NaN            NaN
2558354   NaN          NaN            NaN         NaN            NaN
2558355   NaN          NaN            NaN         NaN            NaN
2558356   NaN          NaN            NaN         NaN            NaN
2558877   NaN          NaN            NaN         NaN            NaN

[12 rows x 43 columns]
```

Note that column are not taxonomically ordered, we need to resort them the way we want.

```
In [17]: # Then create list of ordered columns and reorder dataframe
         # PSEUDOCODE
         # inizializza lista colonne ordinate
         # inizializza lista colonne del database ancora da matchare in ordine alfabetico
         # per ogni rank nella tassonomia generale
         #     se esiste un match perfetto con una colonna del df
         #         aggiungi quella colonna (il match) alla lista ordinata
         #         rimuovi la colonna aggiunta dalla lista di ricerca
         #     altrimenti se esiste un match con "sub_"
```

```python
#         do
#             aggiungi il primo match trovato alla lista ordinata
#             rimuovi la colonna aggiunta dalla lista di ricerca
#         finche' c'e' un match con "sub_"
columns_ordered = () # tuples maintain order, not strictly necessary this time.\
df_coltomatch = dd.columns.to_list()
df_coltomatch.sort() # sort columns to avoid some checks
for rank in TAXONOMY:
    if rank in str(df_coltomatch):
        columns_ordered += (rank,)
        df_coltomatch.remove(rank)
        while True:
            if not any(s.startswith('sub_' + rank) for s in df_coltomatch):
                break
            for e in df_coltomatch:
                # list to match is ordered !
                if e.startswith('sub_' + rank):
                    columns_ordered += (e,)
                    df_coltomatch.remove(e)
        # Deal with 'species group'
        for e in df_coltomatch:
            # list to match is ordered !
            if e.startswith(rank):
                columns_ordered += (e,)
                df_coltomatch.remove(e)


print("Ordered columns:\n{}".format("\n".join(columns_ordered)))
print("\n\nDid I miss any column ? {}".format(len(df_coltomatch) != 0))
```

```
Ordered columns:
root
sub_root
superkingdom
sub_superkingdom
kingdom
sub_kingdom
sub_kingdom_2
sub_kingdom_1
sub_kingdom_3
phylum
sub_phylum
sub_phylum_1
class
sub_class
subclass
sub_subclass
sub_subclass_2
```

```
sub_subclass_1
sub_subclass_3
infraclass
sub_infraclass
superorder
sub_superorder
order
sub_order
sub_order_1
suborder
sub_suborder
infraorder
superfamily
sub_superfamily
family
sub_family
subfamily
sub_subfamily
tribe
genus
sub_genus
subgenus
species
sub_species
species group
subspecies


Did I miss any column ? False
```

```
In [18]: dd = dd[list(columns_ordered)] # reorder columns
         dd.iloc[np.r_[0:6, -6:0]]
```

```
Out[18]:          root           sub_root superkingdom sub_superkingdom  kingdom  \
         6447     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6448     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6451     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6452     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6453     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         6454     root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558352  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558353  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558354  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558355  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558356  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
         2558877  root  cellular organisms    Eukaryota     Opisthokonta  Metazoa
```

```
        sub_kingdom sub_kingdom_2 sub_kingdom_1   sub_kingdom_3    phylum  \
6447      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
6448      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
6451      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
6452      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
6453      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
6454      Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558352   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558353   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558354   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558355   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558356   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca
2558877   Eumetazoa   Protostomia     Bilateria  Lophotrochozoa  Mollusca

         ... subfamily sub_subfamily tribe      genus sub_genus subgenus  \
6447     ...       NaN           NaN   NaN        NaN       NaN      NaN
6448     ...       NaN           NaN   NaN        NaN       NaN      NaN
6451     ...       NaN           NaN   NaN        NaN       NaN      NaN
6452     ...       NaN           NaN   NaN   Haliotis       NaN      NaN
6453     ...       NaN           NaN   NaN   Haliotis       NaN      NaN
6454     ...       NaN           NaN   NaN   Haliotis       NaN      NaN
2558352  ...       NaN           NaN   NaN  Planorbis       NaN      NaN
2558353  ...       NaN           NaN   NaN     Anisus       NaN      NaN
2558354  ...       NaN           NaN   NaN     Anisus       NaN      NaN
2558355  ...       NaN           NaN   NaN     Anisus       NaN      NaN
2558356  ...       NaN           NaN   NaN  Planorbis       NaN      NaN
2558877  ...       NaN           NaN   NaN        NaN       NaN      NaN

                      species sub_species species group subspecies
6447                      NaN         NaN           NaN        NaN
6448                      NaN         NaN           NaN        NaN
6451                      NaN         NaN           NaN        NaN
6452                      NaN         NaN           NaN        NaN
6453       Haliotis corrugata         NaN           NaN        NaN
6454       Haliotis rufescens         NaN           NaN        NaN
2558352  Planorbis sp. DCLF41         NaN           NaN        NaN
2558353     Anisus sp. DCLF73         NaN           NaN        NaN
2558354     Anisus sp. DCLF74         NaN           NaN        NaN
2558355     Anisus sp. DCLF77         NaN           NaN        NaN
2558356  Planorbis sp. DCLF80         NaN           NaN        NaN
2558877                   NaN         NaN           NaN        NaN

[12 rows x 43 columns]
```

```
In [19]: filter_col = [col for col in dd if col.startswith('sub_')]
         for f in filter_col:
             print("Column '{}' has {} unique value(s): {}\n".format(f, len(dd[f].dropna().unic
```

```
Column 'sub_root' has 1 unique value(s): [u'cellular organisms']
Column 'sub_superkingdom' has 1 unique value(s): [u'Opisthokonta']

Column 'sub_kingdom' has 1 unique value(s): [u'Eumetazoa']

Column 'sub_kingdom_2' has 1 unique value(s): [u'Protostomia']

Column 'sub_kingdom_1' has 1 unique value(s): [u'Bilateria']

Column 'sub_kingdom_3' has 1 unique value(s): [u'Lophotrochozoa']

Column 'sub_phylum' has 3 unique value(s): [u'Aplacophora' u'environmental samples' u'unclassi

Column 'sub_phylum_1' has 1 unique value(s): [u'unclassified Aplacophora']

Column 'sub_class' has 6 unique value(s): [u'unclassified Bivalvia' u'unclassified Gastropoda'
 u'environmental samples' u'Gastropoda incertae sedis'
 u'unclassified Neomeniomorpha' u'unclassified Polyplacophora']

Column 'sub_subclass' has 13 unique value(s): [u'Sorbeoconcha' u'Euthyneura' u'Caenogastropoda
 u'lower Heterobranchia' u'Hypsogastropoda' u'unclassified Protobranchia'
 u'unclassified Caenogastropoda' u'unclassified Patellogastropoda'
 u'unclassified Neoloricata' u'unclassified Pteriomorphia'
 u'unclassified Heterobranchia' u'unclassified Vetigastropoda'
 u'unclassified Neritimorpha']

Column 'sub_subclass_2' has 3 unique value(s): [u'Hygrophila' u'Sacoglossa' u'unclassified Euo

Column 'sub_subclass_1' has 4 unique value(s): [u'Euopisthobranchia' u'Panpulmonata' u'unclassi
 u'unclassified Sorbeoconcha']

Column 'sub_subclass_3' has 2 unique value(s): [u'unclassified Sacoglossa' u'unclassified Hygro

Column 'sub_infraclass' has 1 unique value(s): [u'unclassified Euheterodonta']

Column 'sub_superorder' has 2 unique value(s): [u'unclassified Decapodiformes' u'unclassified 

Column 'sub_order' has 25 unique value(s): [u'Sigmurethra' u'Orthurethra' u'unclassified Neogas
 u'unclassified Octopoda' u'unclassified Teuthida'
 u'unclassified Chitonida' u'unclassified Ostreoida'
 u'unclassified Nudibranchia' u'unclassified Mytiloida'
 u'unclassified Arcoida' u'unclassified Pholadomyoida'
 u'unclassified Littorinimorpha' u'unclassified Nuculanoida'
 u'unclassified Limoida' u'unclassified Pterioida'
 u'unclassified Veneroida' u'unclassified Myoida'
 u'unclassified Stylommatophora' u'unclassified Systellommatophora'
 u'unclassified Sepiida' u'unclassified Acochlidiacea'
 u'unclassified Cavibelonia' u'unclassified Lucinoida'
 u'environmental samples' u'unclassified Gadilida']

Column 'sub_order_1' has 1 unique value(s): [u'unclassified Sigmurethra']

Column 'sub_suborder' has 5 unique value(s): [u'Cladobranchia incertae sedis' u'unclassified Gy
 u'unclassified Thecosomata' u'unclassified Myopsina'
 u'environmental samples']

Column 'sub_superfamily' has 10 unique value(s): [u'Triophidae' u'Plutoniidae' u'environmental
 u'unclassified Acteonoidea' u'unclassified Architectonicoidea'
 u'Seguenzioidea incertae sedis' u'unclassified Seguenzioidea'
 u'unclassified Galeommatoidea' u'unclassified Truncatelloidea'
 u'Strophocheilidae']

Column 'sub_family' has 112 unique value(s): [u'unclassified Lymnaeidae' u'unclassified Eulimic
 u'Hydrobiidae incertae sedis' u'unclassified Unionidae'
 u'unclassified Teredinidae' u'unclassified Helicarionidae'
 u'Thiaridae incertae sedis' u'Mytilidae incertae sedis'
 u'unclassified Camaenidae' u'Hygromiidae incertae sedis'
 u'unclassified Vesicomyidae' u'unclassified Mytilidae'
 u'unclassified Hydrobiidae' u'unclassified Lepidochitonidae'
 u'unclassified Aglajidae' u'unclassified Corbulidae'
 u'unclassified Psammobiidae' u'unclassified Conidae'
 u'unclassified Chromodorididae' u'unclassified Terebridae'
 u'unclassified Cardiidae' u'unclassified Turridae'
 u'unclassified Viviparidae' u'unclassified Trochidae'
 u'environmental samples' u'unclassified Octopodidae'
 u'unclassified Lepetellidae' u'unclassified Turbinidae'
 u'unclassified Lucinidae' u'unclassified Drilliidae'
 u'unclassified Solemyidae' u'unclassified Ancylidae'
 u'unclassified Veneridae' u'unclassified Peltospiridae'
 u'unclassified Liotiidae' u'unclassified Skeneidae'
 u'unclassified Solariellidae' u'unclassified Calliostomatidae'
 u'unclassified Fasciolariidae' u'unclassified Charopidae'
 u'unclassified Clenchiellidae' u'unclassified Facelinidae'
 u'unclassified Aeolidiidae' u'unclassified Buccinidae'
 u'unclassified Iravadiidae' u'unclassified Cerithiidae'
 u'unclassified Caecidae' u'unclassified Cerithiopsidae'
 u'unclassified Succineidae' u'unclassified Ranellidae'
 u'unclassified Dialidae' u'unclassified Rissoinidae'
 u'unclassified Discodorididae' u'unclassified Cranchiidae'
 u'unclassified Dotidae' u'unclassified Neopilinidae'
 u'unclassified Cocculinidae' u'unclassified Acochlidiidae'
 u'unclassified Veronicellidae' u'unclassified Pleuroceridae'
 u'unclassified Thyasiridae' u'unclassified Gastropteridae'
 u'unclassified Planorbidae' u'unclassified Planaxidae'
 u'unclassified Tritoniidae' u'unclassified Costellariidae'
 u'unclassified Mitridae' u'unclassified Physidae'
 u'unclassified Capulidae' u'unclassified Cyclophoridae'
 u'unclassified Dreissenidae' u'unclassified Urocoptidae'
 u'unclassified Trapezidae' u'unclassified Gastrochaenidae'
 u'unclassified Tellinidae' u'unclassified Colloniidae'
 u'unclassified Olividae' u'unclassified Sphaeriidae'
 u'unclassified Fissurellidae' u'unclassified Semelidae'
 u'unclassified Columbellidae' u'unclassified Pneumodermatidae'
 u'unclassified Proneomeniidae' u'unclassified Clausiliidae'
 u'unclassified Streptaxidae' u'unclassified Subulinidae'
 u'unclassified Prochaetodermatidae' u'unclassified Acanthomeniidae'
 u'unclassified Anomiidae' u'unclassified Limacidae'
 u'unclassified Haplotrematidae' u'unclassified Vertiginidae'
 u'unclassified Zonitidae' u'unclassified Agriolimacidae'
 u'unclassified Hygromiidae' u'unclassified Lottiidae'
 u'unclassified Arionidae' u'unclassified Tateidae'
 u'unclassified Amphimeniidae' u'unclassified Galeommatidae'
 u'unclassified Pruvotinidae' u'unclassified Simrothiellidae'
 u'unclassified Dondersiidae' u'unclassified Gymnomeniidae'
 u'unclassified Philomycidae' u'unclassified Architectonicidae'
 u'unclassified Littorinidae' u'unclassified Ficidae'
 u'unclassified Personidae' u'unclassified Bathysciadiidae'
 u'unclassified Pseudococculinidae' u'unclassified Clavatulidae']

Column 'sub_subfamily' has 11 unique value(s): [u'unclassified Cantharidinae' u'unclassified St
 u'environmental samples' u'unclassified Bathymodiolinae'
 u'unclassified Photinae' u'unclassified Belgrandiinae'
 u'unclassified Lophomeniinae' u'unclassified Umboniinae'
 u'unclassified Triculinae' u'unclassified Heteroteuthidinae'
 u'unclassified Halomeniinae']

Column 'sub_genus' has 16 unique value(s): [u'unclassified Loligo' u'unclassified Mytilus' u'un
 u'unclassified Conus' u'environmental samples'
 u'Conasprella incertae sedis' u'Turbo incertae sedis'
 u'unclassified Turbo' u'unclassified Tergipes'
 u'Trochulus hispidus complex' u'unclassified Cerithidea'
 u'Cochlostoma incertae sedis' u'unclassified Chilostoma'
 u'unclassified Choriplax' u'unclassified Hemiarthrum'
 u'unclassified Cochlostoma']

Column 'sub_species' has 12 unique value(s): [u'Bathymodiolus brevior Lau back arc basin'
 u'Bathymodiolus brevior Kairei vent' u'Bathymodiolus brevior Edmond vent'
 u'Corbicula javanica form B' u'Patelloida pygmaea form conulus'
 u'Lunella cinerea A STW-2006' u'Lunella cinerea B STW-2006'
 u'Benthoctopus eureka 1 JMS-2006' u'Benthoctopus normani 2 JMS-2006'
 u'Benthoctopus normani 1 JMS-2006' u'Benthoctopus eureka 2 JMS-2006'
 u'Benthoctopus normani 3 JMS-2006']
```

```
In [20]: #select no_rank columns rooting (starting from) at the chosen organism i.e. avoid anc
         organism_rank = ncbi.get_rank([organism_taxid])[organism_taxid]
         try:
             idx_filter = filter_col.index("sub_" + organism_rank)
         except: # if the there is no no_rank below the organism, root at the organism
             idx_filter = filter_col.index(organism_rank)
         norank_col = filter_col[idx_filter:]
         norank_col
```

```
Out[20]: [u'sub_phylum',
          u'sub_phylum_1',
          u'sub_class',
          u'sub_subclass',
          u'sub_subclass_2',
          u'sub_subclass_1',
          u'sub_subclass_3',
          u'sub_infraclass',
          u'sub_superorder',
          u'sub_order',
          u'sub_order_1',
          u'sub_suborder',
          u'sub_superfamily',
          u'sub_family',
          u'sub_subfamily',
          u'sub_genus',
          u'sub_species']
```

```
In [21]: # dataframe with only those organism that have at least one no rank in the lineage
         norank_df = dd[dd[norank_col].notnull().any(axis = 1)]
```

0.1.2 Create dataset of taxid with name, rank and lineage

```
In [22]: # First build a dictionary...
         df = {}
         for taxid in descendants + [organism_taxid]:
             df[taxid] = {}

             specie = ncbi.translate_to_names([taxid])
             rank_dict = ncbi.get_rank([taxid])
             lineage_id = ncbi.get_lineage(taxid)
             names = ncbi.get_taxid_translator(lineage_id)
             lineage_name = [names[taxid] for taxid in lineage_id]

             df[taxid]['name'] = specie[0]
             df[taxid]['rank'] = rank_dict[taxid]
             df[taxid]['lineage_id'] = '//'.join([str(char) for char in lineage_id])
             df[taxid]['lineage_name'] = '//'.join(lineage_name)
         #    df[taxid]['lineage_complete'] = taxid_lineage[taxid]
```

```
In [23]: #print(json.dumps(df, indent = 2))
```

```
In [24]: # ... then convert the dictionary to dataframe
         data = pd.DataFrame.from_dict(data=df, orient="index")
         data.iloc[np.r_[0:3, -3:0]]
```

```
Out[24]:                                                  lineage_id  \
         6447     1//131567//2759//33154//33208//6072//33213//33...
         6448     1//131567//2759//33154//33208//6072//33213//33...
         6451     1//131567//2759//33154//33208//6072//33213//33...
         2558355  1//131567//2759//33154//33208//6072//33213//33...
         2558356  1//131567//2759//33154//33208//6072//33213//33...
         2558877  1//131567//2759//33154//33208//6072//33213//33...

                                  name     rank  \
         6447                 Mollusca   phylum
         6448               Gastropoda    class
         6451               Haliotidae   family
         2558355       Anisus sp. DCLF77  species
         2558356    Planorbis sp. DCLF80  species
         2558877  unclassified Tateidae  no rank

                                                        lineage_name
         6447     root//cellular organisms//Eukaryota//Opisthoko...
         6448     root//cellular organisms//Eukaryota//Opisthoko...
         6451     root//cellular organisms//Eukaryota//Opisthoko...
         2558355  root//cellular organisms//Eukaryota//Opisthoko...
         2558356  root//cellular organisms//Eukaryota//Opisthoko...
         2558877  root//cellular organisms//Eukaryota//Opisthoko...
```

Add ancestor relationship

```
In [25]: data['sonof_id'] = None
         data['sonof_name'] = None
         for index, row in data.iterrows():
             row['sonof_id'] = row['lineage_id'].split('//')[-2] # take father node
             row['sonof_name'] = row['lineage_name'].split('//')[-2] # take father node
             #row['son_of_(rank_name)'] = data[index, 'son_of']

         # Reorder columns
         data = data[["name", "rank", "sonof_id", "sonof_name", "lineage_id", "lineage_name"]]
         data.sort_values(by=['lineage_id'], inplace=True) # order rows by lineage id
         data.iloc[np.r_[0:5, -5:0]]
```

```
Out[25]:                           name    rank sonof_id          sonof_name  \
         6447                  Mollusca  phylum  1206795       Lophotrochozoa
         32584               Scaphopoda   class     6447             Mollusca
         32585               Dentaliida   order    32584           Scaphopoda
```

```
120450                  Rhabdidae   family    32585             Dentaliida
120451                    Rhabdus    genus   120450              Rhabdidae
2230179     Mollusca sp. IOP_0179  species   696338  unclassified Mollusca
2230263     Mollusca sp. IOP_0387  species   696338  unclassified Mollusca
2230264     Mollusca sp. IOP_0390  species   696338  unclassified Mollusca
2230281     Mollusca sp. IOP_0450  species   696338  unclassified Mollusca
696312   cf. Mollusca sp. DH-2009  species   696338  unclassified Mollusca

                                                lineage_id  \
6447     1//131567//2759//33154//33208//6072//33213//33...
32584    1//131567//2759//33154//33208//6072//33213//33...
32585    1//131567//2759//33154//33208//6072//33213//33...
120450   1//131567//2759//33154//33208//6072//33213//33...
120451   1//131567//2759//33154//33208//6072//33213//33...
2230179  1//131567//2759//33154//33208//6072//33213//33...
2230263  1//131567//2759//33154//33208//6072//33213//33...
2230264  1//131567//2759//33154//33208//6072//33213//33...
2230281  1//131567//2759//33154//33208//6072//33213//33...
696312   1//131567//2759//33154//33208//6072//33213//33...

                                              lineage_name
6447     root//cellular organisms//Eukaryota//Opisthoko...
32584    root//cellular organisms//Eukaryota//Opisthoko...
32585    root//cellular organisms//Eukaryota//Opisthoko...
120450   root//cellular organisms//Eukaryota//Opisthoko...
120451   root//cellular organisms//Eukaryota//Opisthoko...
2230179  root//cellular organisms//Eukaryota//Opisthoko...
2230263  root//cellular organisms//Eukaryota//Opisthoko...
2230264  root//cellular organisms//Eukaryota//Opisthoko...
2230281  root//cellular organisms//Eukaryota//Opisthoko...
696312   root//cellular organisms//Eukaryota//Opisthoko...
```

Create dataframe for full taxonomy (include everything)

```
In [26]: full_taxonomy = data.join(dd) # join with distinct column of taxonomy
         print(full_taxonomy.shape)
         full_taxonomy.iloc[np.r_[0:7, -7:0]]
```

```
(31676, 49)
```

```
Out[26]:                              name      rank sonof_id            sonof_name  \
         6447                     Mollusca   phylum  1206795        Lophotrochozoa
         32584                  Scaphopoda    class     6447              Mollusca
         32585                  Dentaliida    order    32584            Scaphopoda
         120450                  Rhabdidae   family    32585            Dentaliida
         120451                    Rhabdus    genus   120450             Rhabdidae
         120452            Rhabdus rectius  species   120451               Rhabdus
```

```
192396                Gadilinidae   family    32585           Dentaliida
2230141     Mollusca sp. IOP_0029  species   696338  unclassified Mollusca
2230142     Mollusca sp. IOP_0030  species   696338  unclassified Mollusca
2230179     Mollusca sp. IOP_0179  species   696338  unclassified Mollusca
2230263     Mollusca sp. IOP_0387  species   696338  unclassified Mollusca
2230264     Mollusca sp. IOP_0390  species   696338  unclassified Mollusca
2230281     Mollusca sp. IOP_0450  species   696338  unclassified Mollusca
696312   cf. Mollusca sp. DH-2009  species   696338  unclassified Mollusca

                                                lineage_id  \
6447     1//131567//2759//33154//33208//6072//33213//33...
32584    1//131567//2759//33154//33208//6072//33213//33...
32585    1//131567//2759//33154//33208//6072//33213//33...
120450   1//131567//2759//33154//33208//6072//33213//33...
120451   1//131567//2759//33154//33208//6072//33213//33...
120452   1//131567//2759//33154//33208//6072//33213//33...
192396   1//131567//2759//33154//33208//6072//33213//33...
2230141  1//131567//2759//33154//33208//6072//33213//33...
2230142  1//131567//2759//33154//33208//6072//33213//33...
2230179  1//131567//2759//33154//33208//6072//33213//33...
2230263  1//131567//2759//33154//33208//6072//33213//33...
2230264  1//131567//2759//33154//33208//6072//33213//33...
2230281  1//131567//2759//33154//33208//6072//33213//33...
696312   1//131567//2759//33154//33208//6072//33213//33...

                                              lineage_name  root  \
6447     root//cellular organisms//Eukaryota//Opisthoko...  root
32584    root//cellular organisms//Eukaryota//Opisthoko...  root
32585    root//cellular organisms//Eukaryota//Opisthoko...  root
120450   root//cellular organisms//Eukaryota//Opisthoko...  root
120451   root//cellular organisms//Eukaryota//Opisthoko...  root
120452   root//cellular organisms//Eukaryota//Opisthoko...  root
192396   root//cellular organisms//Eukaryota//Opisthoko...  root
2230141  root//cellular organisms//Eukaryota//Opisthoko...  root
2230142  root//cellular organisms//Eukaryota//Opisthoko...  root
2230179  root//cellular organisms//Eukaryota//Opisthoko...  root
2230263  root//cellular organisms//Eukaryota//Opisthoko...  root
2230264  root//cellular organisms//Eukaryota//Opisthoko...  root
2230281  root//cellular organisms//Eukaryota//Opisthoko...  root
696312   root//cellular organisms//Eukaryota//Opisthoko...  root

                   sub_root superkingdom sub_superkingdom  ... subfamily  \
6447     cellular organisms    Eukaryota     Opisthokonta  ...       NaN
32584    cellular organisms    Eukaryota     Opisthokonta  ...       NaN
32585    cellular organisms    Eukaryota     Opisthokonta  ...       NaN
120450   cellular organisms    Eukaryota     Opisthokonta  ...       NaN
120451   cellular organisms    Eukaryota     Opisthokonta  ...       NaN
120452   cellular organisms    Eukaryota     Opisthokonta  ...       NaN
```

```
192396   cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230141  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230142  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230179  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230263  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230264  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
2230281  cellular organisms    Eukaryota     Opisthokonta  ...       NaN
696312   cellular organisms    Eukaryota     Opisthokonta  ...       NaN

        sub_subfamily tribe    genus sub_genus subgenus  \
6447              NaN   NaN      NaN       NaN      NaN
32584             NaN   NaN      NaN       NaN      NaN
32585             NaN   NaN      NaN       NaN      NaN
120450            NaN   NaN      NaN       NaN      NaN
120451            NaN   NaN  Rhabdus       NaN      NaN
120452            NaN   NaN  Rhabdus       NaN      NaN
192396            NaN   NaN      NaN       NaN      NaN
2230141           NaN   NaN      NaN       NaN      NaN
2230142           NaN   NaN      NaN       NaN      NaN
2230179           NaN   NaN      NaN       NaN      NaN
2230263           NaN   NaN      NaN       NaN      NaN
2230264           NaN   NaN      NaN       NaN      NaN
2230281           NaN   NaN      NaN       NaN      NaN
696312            NaN   NaN      NaN       NaN      NaN

                          species sub_species species group subspecies
6447                          NaN         NaN           NaN        NaN
32584                         NaN         NaN           NaN        NaN
32585                         NaN         NaN           NaN        NaN
120450                        NaN         NaN           NaN        NaN
120451                        NaN         NaN           NaN        NaN
120452            Rhabdus rectius         NaN           NaN        NaN
192396                        NaN         NaN           NaN        NaN
2230141     Mollusca sp. IOP_0029         NaN           NaN        NaN
2230142     Mollusca sp. IOP_0030         NaN           NaN        NaN
2230179     Mollusca sp. IOP_0179         NaN           NaN        NaN
2230263     Mollusca sp. IOP_0387         NaN           NaN        NaN
2230264     Mollusca sp. IOP_0390         NaN           NaN        NaN
2230281     Mollusca sp. IOP_0450         NaN           NaN        NaN
696312   cf. Mollusca sp. DH-2009         NaN           NaN        NaN

[14 rows x 49 columns]
```

Create taxonomy for organisms that have at least a no_rank associated

```
In [27]: norank_taxonomy = data.join(norank_df, how='right')
         print(norank_taxonomy.shape)
         norank_taxonomy.iloc[np.r_[0:7, -7:0]]
```

```
(15751, 49)
```

```
Out[27]:                             name     rank sonof_id          sonof_name  \
         6470                 Potamididae   family    69597        Cerithioidea
         6471                  Cerithidea    genus     6470         Potamididae
         6472     Cerithidea rhizophorarum  species     6471          Cerithidea
         6496            Euopisthobranchia  no rank   216307          Euthyneura
         6497                    Aplysiida    order     6496   Euopisthobranchia
         6498                   Aplysiidae   family   216318          Aplysioidea
         6499                      Aplysia    genus     6498          Aplysiidae
         2547865        Conus sp. 2 NP-2019  species  2071698  unclassified Conus
         2558352       Planorbis sp. DCLF41  species    55738           Planorbis
         2558353          Anisus sp. DCLF73  species   271028              Anisus
         2558354          Anisus sp. DCLF74  species   271028              Anisus
         2558355          Anisus sp. DCLF77  species   271028              Anisus
         2558356       Planorbis sp. DCLF80  species    55738           Planorbis
         2558877     unclassified Tateidae  no rank  1345660            Tateidae

                                                        lineage_id  \
         6470     1//131567//2759//33154//33208//6072//33213//33...
         6471     1//131567//2759//33154//33208//6072//33213//33...
         6472     1//131567//2759//33154//33208//6072//33213//33...
         6496     1//131567//2759//33154//33208//6072//33213//33...
         6497     1//131567//2759//33154//33208//6072//33213//33...
         6498     1//131567//2759//33154//33208//6072//33213//33...
         6499     1//131567//2759//33154//33208//6072//33213//33...
         2547865  1//131567//2759//33154//33208//6072//33213//33...
         2558352  1//131567//2759//33154//33208//6072//33213//33...
         2558353  1//131567//2759//33154//33208//6072//33213//33...
         2558354  1//131567//2759//33154//33208//6072//33213//33...
         2558355  1//131567//2759//33154//33208//6072//33213//33...
         2558356  1//131567//2759//33154//33208//6072//33213//33...
         2558877  1//131567//2759//33154//33208//6072//33213//33...

                                                      lineage_name  root  \
         6470     root//cellular organisms//Eukaryota//Opisthoko...  root
         6471     root//cellular organisms//Eukaryota//Opisthoko...  root
         6472     root//cellular organisms//Eukaryota//Opisthoko...  root
         6496     root//cellular organisms//Eukaryota//Opisthoko...  root
         6497     root//cellular organisms//Eukaryota//Opisthoko...  root
         6498     root//cellular organisms//Eukaryota//Opisthoko...  root
         6499     root//cellular organisms//Eukaryota//Opisthoko...  root
         2547865  root//cellular organisms//Eukaryota//Opisthoko...  root
         2558352  root//cellular organisms//Eukaryota//Opisthoko...  root
         2558353  root//cellular organisms//Eukaryota//Opisthoko...  root
         2558354  root//cellular organisms//Eukaryota//Opisthoko...  root
         2558355  root//cellular organisms//Eukaryota//Opisthoko...  root
```

```
2558356  root//cellular organisms//Eukaryota//Opisthoko...  root
2558877  root//cellular organisms//Eukaryota//Opisthoko...  root

                   sub_root superkingdom sub_superkingdom  ... subfamily  \
6470     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6471     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6472     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6496     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6497     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6498     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
6499     cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2547865  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558352  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558353  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558354  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558355  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558356  cellular organisms    Eukaryota    Opisthokonta  ...       NaN
2558877  cellular organisms    Eukaryota    Opisthokonta  ...       NaN

        sub_subfamily tribe       genus           sub_genus subgenus  \
6470              NaN   NaN         NaN                 NaN      NaN
6471              NaN   NaN  Cerithidea                 NaN      NaN
6472              NaN   NaN  Cerithidea                 NaN      NaN
6496              NaN   NaN         NaN                 NaN      NaN
6497              NaN   NaN         NaN                 NaN      NaN
6498              NaN   NaN         NaN                 NaN      NaN
6499              NaN   NaN     Aplysia                 NaN      NaN
2547865           NaN   NaN       Conus  unclassified Conus      NaN
2558352           NaN   NaN   Planorbis                 NaN      NaN
2558353           NaN   NaN      Anisus                 NaN      NaN
2558354           NaN   NaN      Anisus                 NaN      NaN
2558355           NaN   NaN      Anisus                 NaN      NaN
2558356           NaN   NaN   Planorbis                 NaN      NaN
2558877           NaN   NaN         NaN                 NaN      NaN

                          species sub_species species group subspecies
6470                          NaN         NaN           NaN        NaN
6471                          NaN         NaN           NaN        NaN
6472     Cerithidea rhizophorarum         NaN           NaN        NaN
6496                          NaN         NaN           NaN        NaN
6497                          NaN         NaN           NaN        NaN
6498                          NaN         NaN           NaN        NaN
6499                          NaN         NaN           NaN        NaN
2547865       Conus sp. 2 NP-2019         NaN           NaN        NaN
2558352      Planorbis sp. DCLF41         NaN           NaN        NaN
2558353         Anisus sp. DCLF73         NaN           NaN        NaN
2558354         Anisus sp. DCLF74         NaN           NaN        NaN
2558355         Anisus sp. DCLF77         NaN           NaN        NaN
```

```
2558356      Planorbis sp. DCLF80         NaN           NaN         NaN
2558877                        NaN         NaN           NaN         NaN

[14 rows x 49 columns]
```

Create complete taxonomy (that is the difference between full and no_ranks df)

```
In [28]: complete_taxonomy = full_taxonomy.loc[full_taxonomy.index.difference(norank_taxonomy.i
         complete_taxonomy.dropna(axis=1, how = 'all', inplace=True) # remove now columns with
         print(complete_taxonomy.shape)
         complete_taxonomy.iloc[np.r_[0:7, -7:0]]
```

```
(15925, 31)
```

```
Out[28]:                                  name     rank  sonof_id       sonof_name  \
         6447                         Mollusca   phylum   1206795  Lophotrochozoa
         6448                       Gastropoda    class      6447         Mollusca
         6451                       Haliotidae   family    216276      Haliotoidea
         6452                         Haliotis    genus      6451       Haliotidae
         6453               Haliotis corrugata  species      6452         Haliotis
         6454               Haliotis rufescens  species      6452         Haliotis
         6455              Haliotis cracherodii  species      6452         Haliotis
         2547906  Corbicula sp. 'Form B' AH-2019  species     45948        Corbicula
         2547907  Corbicula sp. 'Form C' AH-2019  species     45948        Corbicula
         2547908  Corbicula sp. 'Form D' AH-2019  species     45948        Corbicula
         2548440              Villorita cornucopia  species   1176410        Villorita
         2550841                         Yaukthwa    genus   1659700    Rectidentinae
         2555875                        Volegalea    genus      6480     Melongenidae
         2555876              Volegalea cochlidium  species   2555875        Volegalea

                                                       lineage_id  \
         6447     1//131567//2759//33154//33208//6072//33213//33...
         6448     1//131567//2759//33154//33208//6072//33213//33...
         6451     1//131567//2759//33154//33208//6072//33213//33...
         6452     1//131567//2759//33154//33208//6072//33213//33...
         6453     1//131567//2759//33154//33208//6072//33213//33...
         6454     1//131567//2759//33154//33208//6072//33213//33...
         6455     1//131567//2759//33154//33208//6072//33213//33...
         2547906  1//131567//2759//33154//33208//6072//33213//33...
         2547907  1//131567//2759//33154//33208//6072//33213//33...
         2547908  1//131567//2759//33154//33208//6072//33213//33...
         2548440  1//131567//2759//33154//33208//6072//33213//33...
         2550841  1//131567//2759//33154//33208//6072//33213//33...
         2555875  1//131567//2759//33154//33208//6072//33213//33...
         2555876  1//131567//2759//33154//33208//6072//33213//33...

                                                     lineage_name  root   \
         6447     root//cellular organisms//Eukaryota//Opisthoko...  root
         6448     root//cellular organisms//Eukaryota//Opisthoko...  root
         6451     root//cellular organisms//Eukaryota//Opisthoko...  root
         6452     root//cellular organisms//Eukaryota//Opisthoko...  root
         6453     root//cellular organisms//Eukaryota//Opisthoko...  root
         6454     root//cellular organisms//Eukaryota//Opisthoko...  root
         6455     root//cellular organisms//Eukaryota//Opisthoko...  root
         2547906  root//cellular organisms//Eukaryota//Opisthoko...  root
         2547907  root//cellular organisms//Eukaryota//Opisthoko...  root
         2547908  root//cellular organisms//Eukaryota//Opisthoko...  root
         2548440  root//cellular organisms//Eukaryota//Opisthoko...  root
         2550841  root//cellular organisms//Eukaryota//Opisthoko...  root
         2555875  root//cellular organisms//Eukaryota//Opisthoko...  root
         2555876  root//cellular organisms//Eukaryota//Opisthoko...  root

                            sub_root superkingdom sub_superkingdom  ... suborder  \
         6447     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6448     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6451     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6452     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6453     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6454     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         6455     cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2547906  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2547907  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2547908  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2548440  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2550841  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2555875  cellular organisms    Eukaryota     Opisthokonta  ...      NaN
         2555876  cellular organisms    Eukaryota     Opisthokonta  ...      NaN

                    superfamily        family      subfamily tribe      genus subgenus  \
         6447               NaN           NaN            NaN   NaN        NaN      NaN
         6448               NaN           NaN            NaN   NaN        NaN      NaN
         6451       Haliotoidea    Haliotidae            NaN   NaN        NaN      NaN
         6452       Haliotoidea    Haliotidae            NaN   NaN   Haliotis      NaN
         6453       Haliotoidea    Haliotidae            NaN   NaN   Haliotis      NaN
         6454       Haliotoidea    Haliotidae            NaN   NaN   Haliotis      NaN
         6455       Haliotoidea    Haliotidae            NaN   NaN   Haliotis      NaN
         2547906  Corbiculoidea  Corbiculidae            NaN   NaN  Corbicula      NaN
         2547907  Corbiculoidea  Corbiculidae            NaN   NaN  Corbicula      NaN
         2547908  Corbiculoidea  Corbiculidae            NaN   NaN  Corbicula      NaN
         2548440     Cyrenoidea     Cyrenidae            NaN   NaN  Villorita      NaN
         2550841     Unionoidea     Unionidae  Rectidentinae   NaN   Yaukthwa      NaN
         2555875    Buccinoidea  Melongenidae            NaN   NaN  Volegalea      NaN
         2555876    Buccinoidea  Melongenidae            NaN   NaN  Volegalea      NaN

                                         species species group subspecies
```

```
6447                                 NaN          NaN         NaN
6448                                 NaN          NaN         NaN
6451                                 NaN          NaN         NaN
6452                                 NaN          NaN         NaN
6453                  Haliotis corrugata          NaN         NaN
6454                  Haliotis rufescens          NaN         NaN
6455                Haliotis cracherodii          NaN         NaN
2547906  Corbicula sp. 'Form B' AH-2019           NaN         NaN
2547907  Corbicula sp. 'Form C' AH-2019           NaN         NaN
2547908  Corbicula sp. 'Form D' AH-2019           NaN         NaN
2548440             Villorita cornucopia          NaN         NaN
2550841                              NaN          NaN         NaN
2555875                              NaN          NaN         NaN
2555876             Volegalea cochlidium          NaN         NaN

[14 rows x 31 columns]
```

0.1.3 Save all dataframes

```
In [ ]: full_taxonomy.to_csv(organism + "_taxonomy_full.csv", index_label = 'taxid')
        norank_taxonomy.to_csv(organism + "_taxonomy_norank.csv", index_label = 'taxid')
        complete_taxonomy.to_csv(organism + "_taxonomy_complete.csv", index_label = 'taxid')
```

0.1.4 Merge taxonomy with dataset of sequences/genes

```
In [ ]: genes = pd.read_csv("merge-test-a aaaa.csv", sep=";")
        genes.head()
```

```
In [ ]: full_sequences = pd.merge(genes, full_taxonomy, left_on='tax_id', right_index=True)
        norank_sequences = pd.merge(genes, norank_taxonomy, left_on='tax_id', right_index=True
        complete_sequences = pd.merge(genes, complete_taxonomy, left_on='tax_id', right_index=
```

```
In [ ]: full_sequences.to_csv("sequences_full.csv", index = False)
        norank_sequences.to_csv("sequences_norank.csv", index = False)
        complete_sequences.to_csv("sequences_complete.csv", index = False)
```

Remove lineage common to all entries (i.e. until Teuthida included)

```
In [ ]: #common_lineage_to_remove = r"root//.*//" + organism
        #data.replace(to_replace = common_lineage_to_remove,
        #              value = "", inplace = True, regex = True)
        #data.head()
```

Create dataframe of lineage of taxonomy ranks for each taxidThat is (taxid:“279107”, rank_lineage: “order//suborder//family//genus//species”)

```
In [ ]: id_taxidLineage = data.lineage_id
        id_taxidLineage.head()
```

```
In [ ]: # Root the lineage starting from the organism of interest
        # That is split the lineage by the organism taxid and take the second part
        id_taxidLineage = str(organism_taxid) + id_taxidLineage.str.split(str(organism_taxid),
        id_taxidLineage.iloc[np.r_[0:10, -10:0]]
```

```
In [ ]: #id_rankorder = data.rank # rank is a function of dataframes
        id_rankorder = data['rank']
        id_rankorder.iloc[np.r_[0:10, -10:0]]
```

```
In [ ]: id_rankLineage = pd.Series()
        for idx, lineage_list in id_taxidLineage.str.split("//").iteritems():
            rank_list = []
            for lin_id in lineage_list:
                lin_rank = id_rankorder[int(lin_id)]
                rank_list.append(lin_rank)
            id_rankLineage[str(idx)] = rank_list
        id_rankLineage.head()
```

```
In [ ]: rank_lin_df = id_rankLineage.to_frame(name = "rank_lineage")
        rank_lin_df
```

```
In [ ]: rank_lin_df = rank_lin_df.assign(rank_lineage = lambda x: x.rank_lineage.str.join("//"
        rank_lin_df.head()
```

```
In [ ]: # Merge original dataframe to the new one with lineage rank
        rank_lin_df.index = rank_lin_df.index.map(int)
        df = data.join(rank_lin_df)
        df.rename(columns = {"rank_lineage": "lineage_rank"}, inplace = True)
        df = df[['name', 'rank', 'lineage_name', 'lineage_rank', 'lineage_id', 'sonof_id', 'so
        df.head()
```

```
In [ ]: #df.to_csv(path_or_buf = 'taxonomy_teuthida.csv', index_label = 'taxid')
```