BROT_Neo4j

November 9, 2022

1 Projet Neo4j sur la base de données BROT

Yannis BARBA

1.0.1 Contexte

Pour ce projet visant à travailler avec une base de données sous forme de graphe avec neo4j, j'ai souhaité mettre à profit cette technologie dans le domaine de l'écologie (étude des populations et des écosystèmes).

Au cours de mes recherches, j'ai découvert une base de données recensant un grand nombre d'espèce méditérranénnes ainsi que leur traits. Les traits étant des charactéristiques phénotypiques, génétiques ou morphologiques d'un organisme (1). Les auteurs de cette base de données sont Çağatay Tavşanoğlu & Juli G. Pausas. Ils ont référencé 2457 taxons méditérranéens et un total de 44 traits, pour obtenir au final 25 764 individus, un individu étant la mesure d'un trait pour un taxon donné (tous les traits ne sont pas recensés pour chaque taxons). Les auteurs ont rédigé un article (2) accompagnant cette base de données afin d'apporter des compléments d'informations qui m'ont été fort utiles pour comprendre les données.

Cette base de données a été constituée de plusieurs manières notamment grâce à des références bibliographiques et des observations expérimentales. Ainsi, pour chaque paire taxon-trait observé, donc pour chaque ligne, il y a également une source bibliographique.

La base de données est composée en 4 fichiers csv :

- BROT2_dat.csv (toutes les paires taxon-trait observé)
- BROT2_sou.csv (toutes les sources)
- BROT2_tax.csv (toutes les informations supplémentaires sur les taxons)
- BROT2_syn.csv (tous les synonymes des taxons)

J'ai également décidé de créer un fichier **traits.csv**, permettant d'avoir le nom complet d'un trait et des explications le concernant. Ces informations sont disponibles dans l'article sur lequel j'ai basé mon projet. J'ai décidé de créer ce fichier afin d'avoir l'information directement dans le résultat de mes requêtes et ne pas avoir à me référer systématiquement à l'article pour analyser mes résultats.

1.1 Problématique

A partir de ce jeu de données, les problématiques possibles sont très nombreuses. D'après l'article ce type de base de données (sur les traits) est très prometteur pour les biologistes afin de comprendre la répartition de populations et la formation d'écosystèmes.

Une base de donnée sous forme de graphe semble tout à fait adaptée, puisqu'elle permet de visualiser assez aisément les appartenances des taxons aux différentes familles, les taxons possédant tel ou tel trait etc. Par ailleurs au cours de mes recherhces, j'ai pu constater que les bases de données de traits étaient souvent sous forme de graphe.

Les sujets de réflexions étant vastes, j'ai décidé de me concentrer sur certaines questions que je me suis posé au premier abord en lisant l'article.

Dans un premier temps, comme l'auteur a identifié des traits en lien avec la résillience des plantes au incendie, j'ai décidé de me concentrer sur l'étude de ces traits. En effet, la présence du feu en méditérranée a eu un impact sur le développement des populations y compris végétales. Certaines espèces ont développé des traits traduisant une adaptation à cette pression environnementale particulière.

Je me suis donc demandé si l'on pouvait identifier un pattern de plantes adpatées à la présence du feu en méditérranée. Comme nous allons le voir par la suite, certains traits sont des mesures directes de l'adaptation aux incendies. Mais je voulais savoir si parmis les plantes qui possèdent ces traits, nous pouvions mettre en évidence d'autres traits qui ne serait pas directement associés à l'adaptation au feu.

1.2 Arborescence du projet

- un fichier ***BROT__Neo4j.ipynb*** pour les analyses et le déroulé du projet
- un pdf du fichier précédent (***BROT__Neo4j.pdf***)
- Un Dossier **datasets-tests** avec l'ensemble des fichiers cités dans le contexte
- Un fichier ***db__create.py*** pour créer la base de données
- un fichier ***db__request.py*** pour certains fonctions de requêtes de bases
- Le PDF de l'article utilisé pour ce projet

Connexion à la base de données

[80]:
```
from py2neo import *
import pandas as pd
from pandas import DataFrame
import matplotlib.pyplot as plt
graph = Graph("bolt://localhost:7687", auth=("neo4j", "root"))
```

1.3 Adaptation aux incendies

Afin de répondre à la première probématique posée, il nous faut définir ce que sont les taxons présentants des adaptations aux incendies. A partir de la description des traits dans l'article, j'ai pu définir un ensemble de trait définissant une adaptation aux incendies, la modalité choisie est indiquée en ().

- Fire-stimulated flowering **FireStimFlower** (yes)
- Resprouting capacity after fire **RespFire** (high)
- Chemical germination cues **ChemCues** (stimulation)
- Heat-stimulated germination **HeatStimGerm** (high|LMH)
- Post-fire seedling emergence **SeedlEmerg** (high)

- Post-fire seedling survival **SeedlSurv** (high)

Afin de connaître la description de chacun des traits choisis, nous pouvons interoger la base de données avec une première requête simple.

[112]:
```
fireTraits = ["FireStimFlower", "RespFire", "ChemCues", "HeatStimGerm",␣
 ↪"SeedlEmerg", "SeedlSurv"]

def traitExplanation(traits):

    for trait in traits:
        rq = '''match(t:Trait {name : $trait}) return t.name as name, t.
 ↪explanation as explanation'''
        res = graph.run(rq, trait=trait).to_data_frame()
        print("######################")
        print(f"Trait : {res['name'][0]} \n {res['explanation'][0]} \n \n")

traitExplanation(fireTraits)
```

```
######################
Trait : FireStimFlower
 Fire-stimulated ï¬owering in post-ï¬re resprouters.


######################
Trait : RespFire
 Resprouting capacity after one year when most of the plant has been scorched
(average proportion of
adult plants that resprout as percentage). Not reported for annual plants (which
can be assumed to be
RespFire = no).


######################
Trait : ChemCues
 Germinative response to smoke (smk), ash (ash), charcoal (cha), nitrogenous
compounds like KNO 3
(NC1), NaNO 2 (NC2), NH 4 Cl (NC3), NH 4 HOC 3 (NC4), NH 4 NO 3 (NC5), or
response to karrikins
(KAR), cyanohydrins (CYN, including several cyanides, mandelonitrile, and
glyceronitrile). The response
in indicated before the vertical bar, and the chemical cue tested, after the
vertical bar (e.g. stimulation|
Smk).


######################
Trait : HeatStimGerm
```

```
 The highest intensity in heat treatments (i.e., seed exposition to dry heat > =
50 Â°C) that produce higher
germination than the control (i.e., stimulated germination). Heat intensity
deï¬ned as: Low (L: o100 Â°C
during o = 5 min.), Moderate (M:o100 Â°C during >5 min. or > = 100 Â°C during o
= 5 min.), High
(H: > = 100 Â°C during >5 min.) or unknown (unk). Note that in many cases, post-
treatment seed viability
is not considered or not speciï¬ed. The heat intensities tested for each
experiment are indicated after
the vertical bar (category|LMH), with an underscore when the corresponding heat
intensity is not tested
(e.g., category|L_H).


######################
Trait : SeedlEmerg
 Average density of seedlings per pre-ï¬re mature individuals emerged during
the ï¬rst year after ï¬re. This
is a ratio (number of seedlings / number of mature plants)


######################
Trait : SeedlSurv
 Proportion of seedlings surviving the ï¬rst dry season after ï¬re (%).
```

Pour des fonctions réutilisées régulièrement, j'ai décidé de créer un fichier db_request

[31]:
```
from db_request import *
```

[117]:
```
def allTaxons():
    rq = ''' match(tx:Taxon) return tx.family as family, tx.genus as genre, tx.
 ↪species as espece'''
    return graph.run(rq).to_data_frame()

allTaxons=allTaxons()
allFamilies=pd.unique(allTaxons["family"])

print(f"Nombre de taxons : {len(allTaxons)}")
print(f"Nombre famille : {len(allFamilies)}")
```

```
Nombre de taxons : 2457
Nombre famille : 119
```

1.3.1 Identification des familles dont certaines espèces sont adaptées aux incendies

Regardons dans un premier temps tous les taxons ayant au moins une modalité d'un des traits identifié précédemment. C'est à dire les taxons présentant au moins une adaptation au feu.

la modalité *stimulation/smk* pour le trait **ChemCues**, signifie que le taxon voit sa germination stimulée par la fumée.

[33]:
```
fireTraits = {
    "FireStimFlower": "yes",
    "RespFire": "high",
    "ChemCues" : "stimulation|smk",
    "HeatStimGerm" : "high|LMH",
    "SeedlEmerg" : "high",
    "SeedlSurv" : "high"
}

def allTaxonAdaptedTo(fireTraits):
    rq = ""
    maxUnion = 0
    for key, value in fireTraits.items():
        rq += f"match(tx:Taxon)<-[:TRAIT_OF_TAXON]-(d:Data {{data:␣
 ↪'{value}'}})-[:TYPE_OF_TRAIT]->(t:Trait {{name: '{key}'}}) return tx.taxon␣
 ↪as taxon, t.name as trait, d.data as data"
        if(maxUnion < len(fireTraits)-1):
            rq += " UNION "
            maxUnion+=1
    return graph.run(rq).to_data_frame()

allTaxonAdaptedToFire = allTaxonAdaptedTo(fireTraits)
print(allTaxonAdaptedToFire)
print(f"{allTaxonAdaptedToFire.shape[0]} taxons présentent au moins une␣
 ↪adaptation au feu")
print(f"soit {round((allTaxonAdaptedToFire.shape[0]/len(allTaxons))*100, 2)}%␣
 ↪des taxons")
```

```
                    taxon           trait  data
0       Asphodelus ramosus  FireStimFlower   yes
1    Cyclamen hederifolium  FireStimFlower   yes
2        Muscari racemosum  FireStimFlower   yes
3         Urginea maritima  FireStimFlower   yes
4      Narcissus triandrus  FireStimFlower   yes
..                     …             …   …
382    Euphorbia characias       SeedlSurv  high
383   Genista acanthoclada       SeedlSurv  high
384        Fumana laevipes       SeedlSurv  high
385     Micromeria juliana       SeedlSurv  high
386    Trifolium campestre       SeedlSurv  high
```

```
[387 rows x 3 columns]
387 taxons présentent au moins une adaptation au feu
soit 15.75% des taxons
```

N'ayant pas de connaissance préalable sur le sujet (les adaptations des plantes au feu), j'ai décidé d'aborder dans un premier temps une approche dite "naïve", c'est à dire que j'étudie dans un premier temps les différents trait identifés précédemment.

1.3.2 Focus sur tous les traits lié aux incendies

Nous pouvons maintenant regarder pour l'ensemble des traits identifiés comme étant une adadaptation au feu, lequel est le plus représenté.

[34]:
```
for trait, modality in fireTraits.items():
    taxonWithTraitAndMod = allTaxonWithTraitAndModality(trait, modality)
    nbreTaxon = len(taxonWithTraitAndMod)
    perc = round((nbreTaxon/allTaxonAdaptedToFire.shape[0])*100, 2)
    print("############")
    print(f"{perc}% des taxons présentant une adaptation au feu ont la modalité␣
 ↪{modality} pour le trait {trait}")
```

```
############
6.2% des taxons présentant une adaptation au feu ont la modalité yes pour le
trait FireStimFlower
############
30.75% des taxons présentant une adaptation au feu ont la modalité high pour le
trait RespFire
############
22.22% des taxons présentant une adaptation au feu ont la modalité
stimulation|smk pour le trait ChemCues
############
15.5% des taxons présentant une adaptation au feu ont la modalité high|LMH pour
le trait HeatStimGerm
############
47.29% des taxons présentant une adaptation au feu ont la modalité high pour le
trait SeedlEmerg
############
10.59% des taxons présentant une adaptation au feu ont la modalité high pour le
trait SeedlSurv
```

1.3.3 Focus sur le trait SeedEmerg

C'est le trait **SeedEmerg** qui est le trait le plus représenté parmi les taxons possédant au moins un trait d'adaptation au feu. C'est pourquoi j'ai décidé de me concentrer d'abord sur ce trait.

Dans un premier temps, déterminons si les taxons possédant ce trait possédent également d'autres traits d'adaptation au feu.

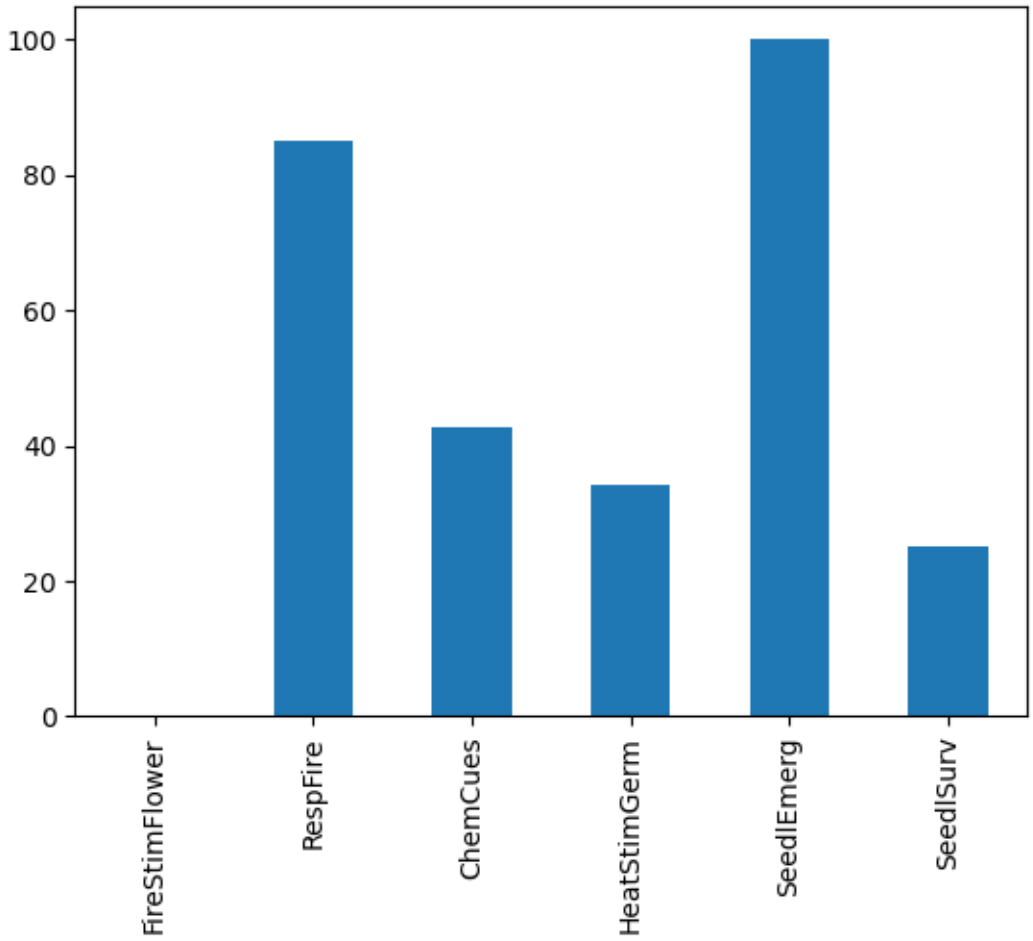
[35]:
```
colname = [key for key, value in fireTraits.items()]
```

```
seedEmergTaxon = allTaxonWithTraitAndModality("SeedlEmerg", "high")

seedEmergTaxonWithTraits = pd.DataFrame(columns=colname)

def ContingenceMatrixToCompare(baseTrait, baseModality, traitsAndModality):
    """Cette fonction retourne une matrice de contingence permettant de␣
 ↪déterminer si des taxons possédant
    un trait et une modalité donnée possède d'autres traits et modalités"""

    colname = [key for key, value in traitsAndModality.items()]

    taxonFromBaseTrait = allTaxonWithTraitAndModality(baseTrait, baseModality)

    matrix = pd.DataFrame(columns=colname)

    for tax in taxonFromBaseTrait["taxon"]:
        mod = {key:0 for key,value in traitsAndModality.items()}
        rq = '''match(tx:Taxon {taxon: $tax})<-[:TRAIT_OF_TAXON]-(d:Data)-[:
 ↪TYPE_OF_TRAIT]->(t:Trait) return t.name as trait, d.data as modality'''
        allTraits = graph.run(rq, tax=tax).to_data_frame()
        for trait in colname:
            if allTraits["trait"].isin([trait]).any():
                if allTraits["modality"][allTraits["trait"][allTraits["trait"]␣
 ↪== trait].index[0]]:
                    mod[trait] = 1

        matrix.loc[tax] = [value for key, value in mod.items()]

    return matrix

seedlEmergContingenceMatrix = ContingenceMatrixToCompare("SeedlEmerg", "high",␣
 ↪fireTraits)
```

[47]:
```
seedlEmergRatio = round((seedlEmergContingenceMatrix.sum(axis="rows")/
 ↪seedlEmergContingenceMatrix.shape[0])*100,2)
print(type(seedlEmergRatio))
seedlEmergRatio.plot(kind="bar")
```

```
<class 'pandas.core.series.Series'>
```

[47]: <AxesSubplot:>



D'après le graphe ci-dessus, nous pouvons dire que les taxons possédant le trait **SeedlEmerg** et la modalité *high* , c'est à dire une forte densité d'individu mature un an après un incendie:

- 85% d'entre eux ont la modalité *"high"* pour le trait **RespFire** qui correspond à la capacité de repousse 1 an après un incendie. Cela semble cohérent, car s'il y a une forte densité de population, c'est bien que les individus ont une forte capacité à la repousse. Cette forte corrélation est donc logique.
- 42% d'entre eux ont la modalité *"stimulation/smk"* pour le trait **ChemCues**, ce qui veut dire que leur germination est stimulée par le feu, ce qui encore une fois est plutôt logique, mais nous pourrions également regarder quelles sont les modalités pour ce trait afin d'apporter plus de finesse dans l'analyse.
- 34 % d'entre eux ont la modalité *"high/LMH"*, pour le trait **HeatStimGerm** c'est à dire que la germination est stimulée lors d'une chaleur très importante, c'est à dire supérieure à 100°C pendant plus de 5min. (Attention, dans le cadre de ce projet, j'ai choisi de me concentrer sur une seule modalité par soucis de simplification pour ce trait, mais comme nous pouvons le constater, les modalités sont plus nombreuses, les résultats sont donc à prendre avec des pincettes).

- 25% ont la modalité *"high"* pour le trait **SeedlSurv**, trait qui correspond à la proportion de graines ayant survécus la première saison sèche après l'incendie.
- 0 % d'entre eux n'ont la modalité *"yes"* pour le trait **FireStimFlower**, soit ce trait n'a pas été renseigné pour ces espèces, soit, un taxon à une adaptation au feu au niveau de la germination OU au niveau de la floraison mais pas pour les deux en même temps.

Il me semble maintenant intéressant de regarder pour tous les taxons possédant le trait **SeedlEmerg** et la modalité *high*, quels sont les traits et leur modalités les plus représentés, afin de comprendre s'il y a des liens entre les traits d'adaptation au feu et les autres.

[59]:
```
def allTraitsOfTaxons(taxons):
    """Pour une liste de taxons donné, cette fonction retourne un dictionnaire␣
 ↪avec l'ensemble des traits observés
    et le nombre de fois que ces trais ont été observé pour tous ces taxons␣
 ↪ainsi que le nombre de fois où chacune
    des modalités a été observé"""
    traits = {}

    for tax in taxons:
        rq = '''match(tx:Taxon {taxon: $tax})<-[:TRAIT_OF_TAXON]-(d:Data)-[:
 ↪TYPE_OF_TRAIT]->(t:Trait) return t.name as trait, d.data as modality'''
        allTraits = graph.run(rq, tax=tax).to_data_frame()
        for trait in allTraits["trait"].unique():
            modality =␣
 ↪allTraits["modality"][allTraits["trait"][allTraits["trait"] == trait].
 ↪index[0]]
            if trait in traits:
                traits[trait]["somme"] +=1
                if modality in traits[trait]:
                    traits[trait][modality] = traits[trait][modality] + 1
                else :
                    traits[trait].update({modality: 1})
            else :
                traits.update({trait: {"somme": 1, modality: 1}})
    return traits

#test = allTraitsOfTaxons(["Ononis pusilla"])
allTraitsForAllTaxonWithSeedlEmerg = allTraitsOfTaxons(seedEmergTaxon["taxon"])
```

[67]:
```
seedlEmergContingenceMatrix.shape
```

[67]:
```
(140, 6)
```

[68]:
```
traitsOccurences = pd.Series({key:␣
 ↪(allTraitsForAllTaxonWithSeedlEmerg[key]["somme"]/
 ↪allTraitsForAllTaxonWithSeedlEmerg["SeedlEmerg"]["somme"])*100 for key,␣
 ↪value in allTraitsForAllTaxonWithSeedlEmerg.items()})
```

```
print(len(traitsOccurences))
```

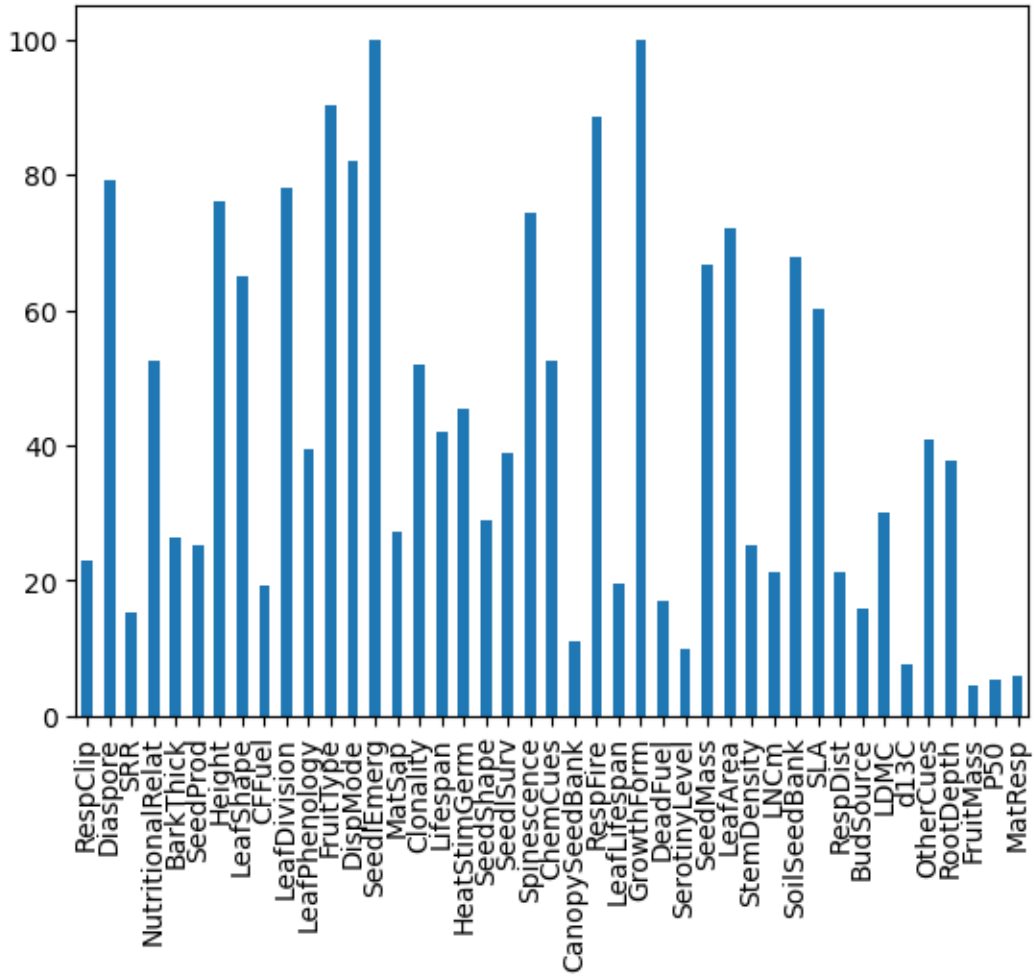
```
43
```

Nous obtenons au total 43 traits sur 44, c'est à dire que la quasi-totalité des traits sont étudiés pour l'ensemble des taxons possédant le trait **SeedlEmerg** et la modalité *high*

Regardons maintenant quels sont les traits les plus représentés parmis l'ensemble des taxons. Attention, si un trait n'apparaît pas pour un taxon donné, cela ne veut pas dire que celui-ci ne le possède pas dans l'absolu, cela signifie simplement que la littérature n'a pas donné d'information à ce sujet. Le graphe ci-dessous nous permet simplement d'extraitre les traits qui sont les plus intéressants à étudier, puisque présent pour la majeure partie des taxons possédant le trait **SeedEmerg** et la modalité *high*

[69]:
```
traitsOccurences.plot(kind="bar")
```

[69]:
```
<AxesSubplot:>
```



Tout à fait arbitrairement, j'ai décidé de m'intéresser aux traits dont on à l'information pour 80% des taxons possédant déjà le trait **SeedEmerg** et la modalité *high*. Ce qui me permet de pouvoir généraliser un minimum mes observations. Je retiens donc les traits :

- Diaspore
- FruitType
- DispMode
- RespFire
- GrowthForm

Bien entendu, dans une étude plus approfondie et avec de meilleure connaissance du domaine, d'autres choix de traits d'études pourraient être fait.

Nous pouvons interroger notre base de données afin de savoir à quoi correspondent ces différents traits.

[113]:
```
traitsOfInterest = ["Diaspore", "FruitType", "DispMode", "RespFire",␣
 ↪"GrowthForm"]
traitExplanation(traitsOfInterest)
```

```
######################
Trait : Diaspore
 Dispersal unit for sexual reproduction.


######################
Trait : FruitType
 Type of the fruit. Including single or aggregated fruits (dispersal unit).


######################
Trait : DispMode
 The vector(s) used for dispersal (of seeds or any other dispersal unit for
sexual reproduction). One or
several types, e.g.: GW means that both Gravity and Wind are important); if
several, typically the ï¬rst is
the most important.


######################
Trait : RespFire
 Resprouting capacity after one year when most of the plant has been scorched
(average proportion of
adult plants that resprout as percentage). Not reported for annual plants (which
can be assumed to be
RespFire = no).


######################
```

```
Trait : GrowthForm
 Morphology of the whole plant related with its size (for non-disturbed
individuals).
```

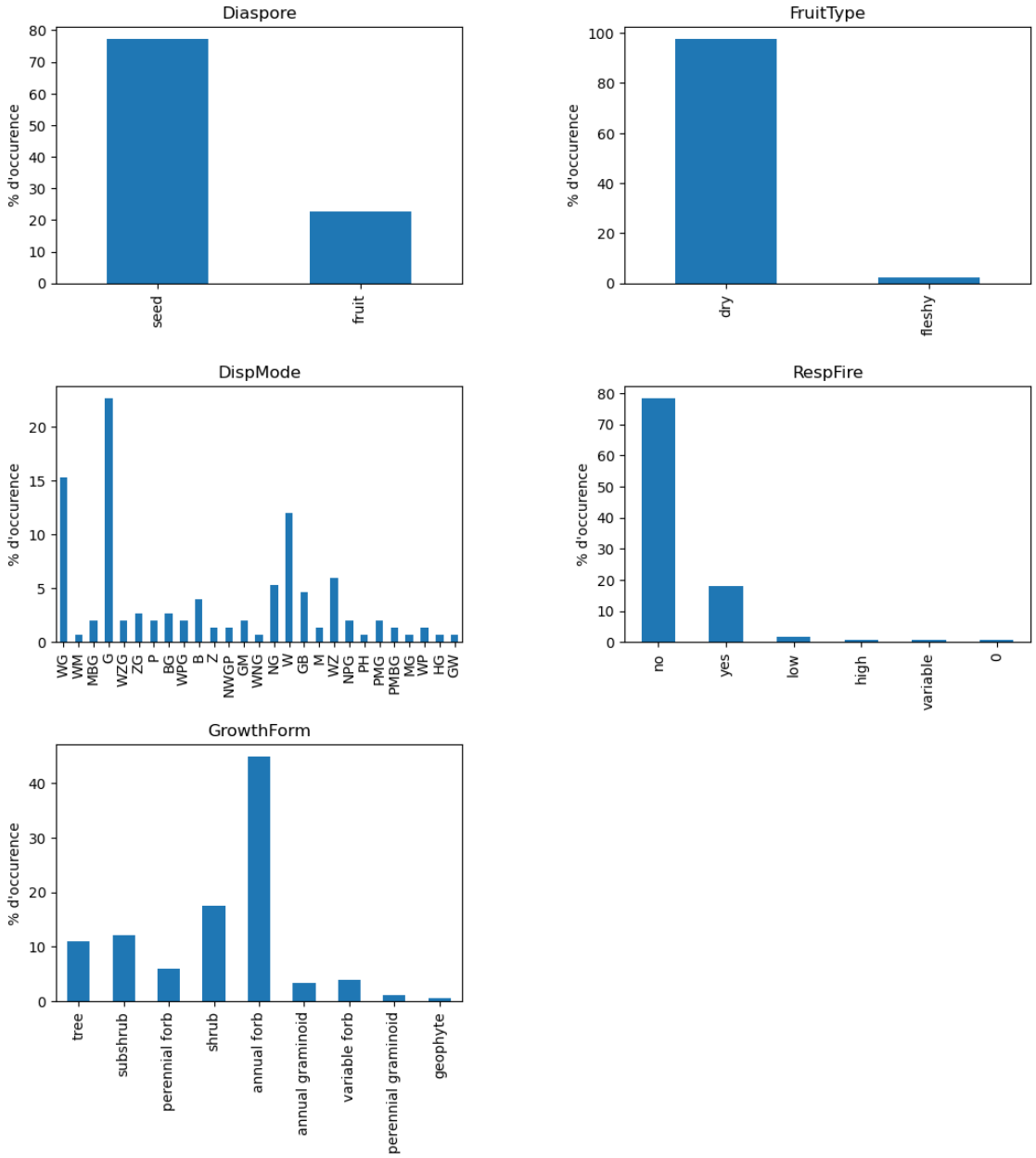
Avec la fonction ***allTraitsOfTaxons*** j'ai les modalités pour chacun des traits ainsi que le nombre de fois où ces modalités ont été prises.

[114]:
```
for trait in traitsOfInterest:
    print(f"{trait} : {allTraitsForAllTaxonWithSeedlEmerg[trait]} \n")
```

```
Diaspore : {'seed': 112, 'fruit': 33}

FruitType : {'dry': 161, 'fleshy': 4}

DispMode : {'WG': 23, 'WM': 1, 'MBG': 3, 'G': 34, 'WZG': 3, 'ZG': 4, 'P': 3,
'BG': 4, 'WPG': 3, 'B': 6, 'Z': 2, 'NWGP': 2, 'GM': 3, 'WNG': 1, 'NG': 8, 'W':
18, 'GB': 7, 'M': 2, 'WZ': 9, 'NPG': 3, 'PH': 1, 'PMG': 3, 'PMBG': 2, 'MG': 1,
'WP': 2, 'HG': 1, 'GW': 1}

RespFire : {'no': 127, 'yes': 29, 'low': 3, 'high': 1, 'variable': 1, '0': 1}

GrowthForm : {'tree': 20, 'subshrub': 22, 'perennial forb': 11, 'shrub': 32,
'annual forb': 82, 'annual graminoid': 6, 'variable forb': 7, 'perennial
graminoid': 2, 'geophyte': 1}
```

[111]:
```
figs=1
plt.figure(figsize=(12,12))
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=0.9,
                    wspace=0.4,
                    hspace=0.4)

for trait in traitsOfInterest:

    copy = allTraitsForAllTaxonWithSeedlEmerg[trait]
    if "somme" in copy:
        del copy["somme"]

    percentage = {key: (copy[key]/sum(copy.values()))*100 for key, value in␣
 ↪copy.items()}
    #Figure
    plt.subplot(3,2,figs)
    pd.Series(percentage).plot(kind="bar")
```

```
    plt.title(f"{trait}")
    plt.ylabel("% d'occurence")
    figs+=1

plt.show()
```



Si l'on regarde chacun des graphiques ci-dessus, on peut constater que :

- Pour le trait **Diaspore** : rappelons que la diaspore est une partie du végétal (graine ou

fruit) qui se détache du pied mère pour reproduire l'espèce après dissémination. Ici, on peut constater que les taxons possédant le trait **SeedlEmerg** et la modalité *high* possédent (pour ceux dont on a l'information) une diaspore sous forme de graine.

- Pour le trait **FruitType**, c'est la modalité *dry* qui est sur-représentée, ainsi
- Pour le trait **DispMode**, c'est la modalité *G* (autochorie par gravité) suivie des modalités *WG* (anémochorie et gravité) et *W*. Ainsi, les taxons étudié ici ont un système de dispersion plutôt localisé, car elle a lieu grâce à la gravité (les fruits chutent). Même si en regardant plus en détail, on peut constater que la modalité *W* est présente dans bon nombre de combinaison de modalité, par conséquent, il faudrait une étude plus approfondie pour comprendre quelle modalité sont réellement sur-représentées pour ce trait.
- Le résultat du trait **RespFire** est à mettre en comparaison avec le résultat du trait **GrowthForm**. En effet, d'après l'article, pour les plantes annuelles, le trait **RespFire** possède la modalité *no* puisqu'il renseigne sur la capacité d'une population adulte (donc plus d'un an) à rester en place après un incendie. Il est donc normal de retrouver une sur-représentation de la modalité *no* puisque la plupart des taxons étudiés ici possèdent la modalité *annual forb* pour le trait **GrowthForm**.

Par manque de temps, je n'ai pas pu approfondir l'analyse et j'en suis resté à du descriptif, cependant je pense que le potentiel de cette base de données est très important. Nous pouvons l'illustrer par un dernier exemple.

1.3.4 Focus sur le trait FireStimFlower

Comme nous avons pu le voir précédemment, le trait **FireStimFlower** était le moins représenté parmi les traits que j'ai identifié comme étant des traits d'adaptation au feu.

Cependant, je souhaitais savoir s'il était possible d'extraire de l'information à partir de la base de données.

Par exemple, nous pouvons extraire l'ensemble des taxons possédant ce trait et regarder quelles sont les familles qui le possédent.

[11]:
```
fireStimFlower = allTaxonWithTraitAndModality("FireStimFlower", "yes")
fireStimFlowerFamilies = pd.unique(fireStimFlower["famille"])
print(f"Nombre de plante possédant ce trait : {len(fireStimFlower)}")
print(f"Nombre de familles ayant au moins une espèce dont la floraison est␣
 ↪stimulée pour les incendies : {len(fireStimFlowerFamilies)}")
print(fireStimFlowerFamilies)
```

```
Nombre de plante possédant ce trait : 24
Nombre de familles ayant au moins une espèce dont la floraison est stimulée pour
les incendies : 8
['Asphodelaceae' 'Primulaceae' 'Asparagaceae' 'Amaryllidaceae' 'Liliaceae'
 'Dioscoreaceae' 'Araceae' 'Iridaceae']
```

[12]:
```
percentageOfFireStimFlowerFamily = round((len(fireStimFlowerFamilies)/
 ↪len(allFamilies))*100, 2)
```

```
print(f"{percentageOfFireStimFlowerFamily}% des familles possèdent au moins une␣
 ↪espèce ayant le trait FireStimFlower")

percentageOfFireStimFlowerTaxon = round((len(fireStimFlower) / len(allTaxons))␣
 ↪* 100, 2)
print(f"{percentageOfFireStimFlowerTaxon}% des taxons ont une floraison␣
 ↪stimulée par le feu")

print(f"{round((len(fireStimFlower) / allTaxonAdaptedToFire.shape[0]) * 100,␣
 ↪2)}% des taxons ayant au moins une adapatation au feu ont une floraison␣
 ↪stimulée par le feu")
```

```
6.72% des familles possèdent au moins une espèce ayant le trait FireStimFlower
0.98% des taxons ont une floraison stimulée par le feu
7.48% des taxons ayant au moins une adapatation au feu ont une floraison
stimulée par le feu
```

Seulement 6% des familles possédent au moins une espèce ayant le trait **FireStimFlower**. Et moins d'1% des plantes voient leur floraison stimulée par le feu.

D'après la documentation, ce trait indique que les plantes le possédant sont capables (voire nécessitent) de fleurir après des incendies. Si l'on regarde un peu plus en détail la documentation, il est indiqué que ce sont les plantes géophytes qui présentent le plus ce trait. Une plante géophyte est une plante qui possède un organe permettant de "passer la mauvaise saison" (wikipédia), tel qu'un bulbe ou bien un rhizome. On peut donc regarder la proportion de plantes géophytes dans celle présentant le trait **FireStimFlower**.

Il existe un trait : **GrowthForm** renseignant les plantes géophytes ou non.

[115]:
```
def modalityOfOneTraitForOneTaxon(taxon, trait):
    rq = '''match(tx:Taxon {taxon: $taxon})<-[:TRAIT_OF_TAXON]-(d:Data)-[:
 ↪TYPE_OF_TRAIT]->(t:Trait {name: $trait}) return tx.taxon as taxon, d.data as␣
 ↪trait'''
    res = graph.run(rq, taxon=taxon, trait=trait).to_data_frame()
    if(len(res) != 0):
        return pd.unique(res["trait"])
    return []
```

[116]:
```
for el in fireStimFlower["taxon"]:
    print(el, modalityOfOneTraitForOneTaxon(el, "GrowthForm"))
```

```
Asphodelus ramosus ['geophyte']
Cyclamen hederifolium ['geophyte']
Muscari racemosum ['geophyte']
Urginea maritima ['geophyte']
Narcissus triandrus ['geophyte']
Narcissus dubius ['geophyte']
Gagea peduncularis ['geophyte']
Dioscorea communis ['geophyte']
```

```
Acis nicaeensis ['geophyte']
Ornithogalum narbonense ['geophyte']
Arisarum vulgare ['geophyte']
Gladiolus illyricus ['geophyte']
Iris lutescens ['geophyte']
Lapiedra martinezii ['geophyte']
Gladiolus illyricus ['geophyte']
Romulea tempskyana ['geophyte']
Ornithogalum sphaerocarpum ['geophyte']
Allium hirtovaginatum ['geophyte']
Allium sandrasicum ['geophyte']
Allium amethystinum ['geophyte']
Gagea graeca ['geophyte']
Leopoldia comosa ['geophyte']
Asparagus stipularis ['liana']
Allium stamineum ['geophyte']
```

Nous pouvons constater qu'a l'exception du taxon "Asparagus stipularis", tous les taxons avec le trait **FireStimFlower** sont en fait des géophytes.

Nous pouvons regarder combien de plantes sont géophytes dans notre base de données avec de savoir si le fait d'être géophyte suffit pour voir sa floraison stimulé par le feu ou bien s'il faut prendre en compte d'autre caractéres.

[15]:
```
allGeophytes = allTaxonWithTraitAndModality("GrowthForm", "geophyte")
print(f"{round((len(fireStimFlower) / len(allGeophytes))*100)} % des geophytes␣
  ↪possédent le trait FireStimFlower")
```

```
13 % des geophytes possédent le trait FireStimFlower
```

D'après le résultat précédent, nous pouvons constater que le fait d'être géophyte n'est pas suffisant pour avoir une floraison stimulée par le feu.

1.4 Conclusion

Ce projet avait pour but de prendre en main la technologie neo4j et la représentation des données sous forme de graphe. Cette technologie me semble pertinente pour étudier les jeux de données tel que celui que j'ai utilisé durant ce projet même si je pense qu'avec plus de connaissance sur cette technologie, on pourrait aller encore plus loin et être plus efficace au niveau des requêtes et de la construction des noeuds et leur relations. En effet, je pense que certains traitements de données post-requêtes pourraient être évités avec des requêtes plus élaborées.

Pour ce qui est de la base de données en elle même, j'ai trouvé le sujet très intéressant et comme dit en introduction, les problématiques sont diverses et très nombreuses. Pour ma part, chaque nouvelle requête suscitait de multiples questions pour essayer de caractériser mes individus. Ma démarche a été très exploratoire et j'ai tenté de répondre aux questions que je me posais en posant des hypothèses parfois fortes. Ainsi je pense que pour approfondir cette étude et donner plus de valeurs aux résultats, il me faudrait travailler avec des experts en écologie comme les auteurs de l'article sur lequel j'ai basé mon travail.

Pour conclure, il pourrait être très intéressant de développer une interface graphique (comme une

application web) pour automatiser les différentes requêtes afin d'avoir un outil au service d'expert en écologie.

1.5 Bibliographie

(1) BiologyOnline. *Traits*. URL: https://www.biologyonline.com/dictionary/traits, date de consultation : 28.10.2022

(2) Tavşanoğlu, Ç. & Pausas, J. G. A functional trait database for Mediterranean Basin plants. Sci. Data 5:180135 doi: 10.1038/sdata.2018.135 (2018).