Transformación_de_datos

December 14, 2020

1 Transformación de datos

[1]:
```
import pse as PseAAC
from pse import AAIndex
import acc as ACC
import AAComposition as AAC

import sys
import os
import pickle
from math import pow
import time
import errno

import const
from util import frequency
from util import get_data
from util import check_args, read_k, write_to_file
from nac import make_kmer_list
from data import index_list
from pse import get_aaindex
import pandas as pd
import arff
```

2 Declaración de variables

[2]:
```
organismo ="nematoda"
dataset = 1
nombre = ("ds" + str(dataset) + "_" + str(organismo))

r1 = ("Datos/listas/"+ str(organismo) + "/" + str(nombre))
r2 = ("Datos/resultados/"+ str(organismo) + "/" + str(nombre))
r3 = (str(r2) + "/temp")

#Ruta a los archivos
efectores_csv = pd.read_csv(str (r1) + '/' + str(nombre) + '_efectores.csv',␣
↪sep = ',')
```

```
efectores_fasta = (str (r1) + '/' + str(nombre) + '_efectores.fasta')
no_efectores_csv = pd.read_csv(str (r1) + '/' + str(nombre) + '_no_efectores.
 ↪csv', sep = ',')
no_efectores_fasta = (str (r1) + '/' + str(nombre) + '_no_efectores.fasta')

#índices fisicoquímicos
ind_1 =('Hydrophobicity', 'Hydrophilicity', 'Mass')
ind_2 =('Hydrophobicity', 'Hydrophilicity')
ind_3 =('Mass')

tipo = [str(efectores_fasta), str(no_efectores_fasta)]
```

[3]:
```
efectores_csv
```

[3]:
```
      etiqueta  num_amino                                          encabezado  \
0    efectores        359  ETN87146.1 collagen triple helix repeat protei…
1    efectores        307  CDJ81163.1 unnamed protein product [Haemonchus…
2    efectores        321  PDM84681.1 hypothetical protein PRIPAC_33704 […
3    efectores        260  KHJ93799.1 hypothetical protein OESDEN_06282 […
4    efectores        119  XP_003368004.1 lysosomal acid phosphatase [Tri…
..         …        …                                                   …
495  efectores        858  RCN35896.1 valine--tRNA ligase [Ancylostoma ca…
496  efectores         56  KJH44985.1 hypothetical protein DICVIV_08969 […
497  efectores        544                CTP81510.1 Bm2278 [Brugia malayi]
498  efectores         74  KHJ82467.1 hypothetical protein OESDEN_17839 […
499  efectores         98  EPB76671.1 hypothetical protein ANCCEY_04233 […

                                              secuencia
0    MPAETLMYTRKGPILLKQASVHPPDFLSRGKPCCSLVIMRVVSDCT…
1    MIVCHENFLTSNFSDATMVVMFVLFGTYLFAYAHILVASSGGLKAA…
2    MKISAITVFLVAKAVVAVDASCAEQSNLCNLSAYDGLMNFYCKKTC…
3    MTKTDKTCVPKNRNKSFKQRSESPVTLHQWHKAEVWRTGKGILMKV…
4    MYRHGYRTPLGTFPTDEYQEWAYPNGFRQLTKLGCQQQYELGQYLR…
..                                                 …
495  MVTADQSRRKTEKELKKETEKAGKLAKFEEKQQKLQEKARQAKPKQ…
496  MRAYEFVTNTPIKLDEHRDYFVDDVALVRSSPPLSPPIKGNSAHNC…
497  MVRDSFDGGHTGDPERDLAYEREHVRRDTMSDEVVPDDVAQYLIYF…
498  MTKPIPKNFAYADTILLFKSGDPENLANYRPISFLSTLYKVLTKLI…
499  MSMKIPYHRHEYWLSCMMVVVVTDLIEYVRSWLRVVIVMHDDREGK…

[500 rows x 4 columns]
```

[4]:
```
no_efectores_csv
```

[4]:
```
         etiqueta  num_amino  \
0    no_efectores        438
1    no_efectores       1558
```

```
2    no_efectores        178
3    no_efectores         68
4    no_efectores        248
..            …        …
495  no_efectores         67
496  no_efectores        333
497  no_efectores        382
498  no_efectores        705
499  no_efectores        878

                                                encabezado  \
0    CAD5153398.1 unnamed protein product [Bursaphe…
1    KAF1771504.1 hypothetical protein GCK72_003331…
2    TKR72102.1 hypothetical protein L596_019612 [S…
3    CAD2199032.1 unnamed protein product [Meloidog…
4    CAB3411057.1 unnamed protein product [Caenorha…
..                                                 …
495  CAD2130146.1 unnamed protein product [Meloidog…
496  KAF1760690.1 hypothetical protein GCK72_008939…
497  TKR86373.1 hypothetical protein L596_010978 [S…
498  CAD5230165.1 unnamed protein product [Bursaphe…
499  NP_001364585.1 Chloride channel protein [Caeno…

                                                secuencia
0    MNSVRRLHTTAEVSRKWIRIHWWPKYERWQDLEKDTIFKKKGDEAV…
1    MKFIIFLALASATLNSTPPIEDRHGPDPLQSNRPNYYGVETYYHSP…
2    MNIELLQLEDGRVTKHKRISMAHDGTSSRHGNWHYEPKGSCAENED…
3    MIIDQNFYYFQVPRPSTRAIMDIKIGTRTFLESEVANKHKRVDLYK…
4    MLERTDLGMDASFGGKLSADEQPSKQAVILVHGITNKITRFAGTMN…
..                                                 …
495  MFNRSKTLMFPYVKMQPLLKRAATNKSFYSSYAAYGASLFVLTVYI…
496  MGQHGAIRLQNEVQEGVMPVHELTEEEQWAEEHRKMHEKHKGHEAM…
497  MPLCDDKAVFVACSGVKHQGSECHSYHQSPAPSRPRQKLLKTFEGS…
498  MRKQICQSPDAVVTSKLTDFLEVSAMQLDQDNSPQTSELKDLNGED…
499  MKQVGFSEHVTVFPAPPTSIAVIATTNKDIDNNVMSDVKRSLLDMS…

[500 rows x 4 columns]
```

3 Composición de aminoácidos (AAC)

[5]:
```python
#AAC
comp = "AAC"

matrix_efec = []
for idx,row in efectores_csv.iterrows(): #iterar por renglones
    sec = row['secuencia']
```

```
    seq_efectores = AAC.CalculateAAComposition(sec)

    list_1 = []
    for k in seq_efectores.keys():
        list_1.append(seq_efectores[k])

    list_1.append(row['etiqueta'])
    matrix_efec.append(list_1)

#nombre del txt
nom = ('ds' + str(dataset) + '_' + str(comp) + '_efectores_' + str(organismo) +␣
 ↪'.temp')

#ruta al directorio donde se almacenara
os.makedirs(str(r3), exist_ok=True)
resultado = ((r3) + '/' + str(nom))


with open(resultado, 'w') as f:
    print(matrix_efec, file=f)
```

[6]:
```
#AAC
comp = "AAC"

matrix_no_efec = []
for idx,row in no_efectores_csv.iterrows(): #iterar por renglones
    sec2 = row['secuencia']
    seq_no_efectores = AAC.CalculateAAComposition(sec2)

    list_2 = []
    for k in seq_no_efectores.keys():
        list_2.append(seq_no_efectores[k])

    list_2.append(row['etiqueta'])
    matrix_no_efec.append(list_2)


#nombre del txt
nom = ('ds' + str(dataset) + '_' + str(comp) + '_no_efectores_' +␣
 ↪str(organismo) + '.temp')

#ruta al directorio donde se almacenara
os.makedirs(str(r3), exist_ok=True)
resultado = ((r3) + '/' + str(nom))

with open(resultado, 'w') as f:
    print(matrix_no_efec, file=f)
```

[7]:
```
df_dataset=pd.concat ([pd.DataFrame(matrix_efec),pd.DataFrame(matrix_no_efec)])
```

[8]:
```
df_dataset
```

[8]:

	0	1	2	3	4	5	6	7	8	9	…	11	12	13	14	15	16	17	18	19	20
0	4.457	8.357	3.621	6.407	2.228	5.571	4.178	16.713	1.950	4.457	…	5.014	1.114	1.671	15.042	4.735	3.621	0.557	2.507	4.457	efectores
1	10.423	4.560	3.583	3.257	0.977	3.583	1.954	4.235	1.629	8.795	…	3.257	2.932	7.166	3.909	7.166	6.840	2.280	4.560	8.795	efectores
2	11.526	2.804	4.984	5.296	4.050	5.296	2.804	3.738	2.492	5.607	…	6.542	2.181	4.673	4.361	6.542	9.657	1.558	2.181	5.607	efectores
3	3.462	6.154	4.615	3.846	3.462	6.154	4.615	9.231	3.077	4.231	…	5.769	3.846	3.462	3.846	7.692	8.077	1.923	1.154	6.923	efectores
4	4.202	7.563	5.882	2.521	2.521	6.723	8.403	8.403	2.521	3.361	…	4.202	3.361	5.042	4.202	3.361	6.723	1.681	8.403	3.361	efectores
..	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
495	7.463	5.970	4.478	1.493	1.493	4.478	2.985	1.493	0.000	2.985	…	10.448	4.478	8.955	4.478	7.463	4.478	1.493	8.955	7.463	no_efectores
496	4.805	4.204	1.201	3.303	3.303	6.006	3.904	8.408	5.405	6.607	…	5.706	5.105	5.706	2.703	3.003	4.204	3.303	4.505	9.009	no_efectores
497	8.115	5.759	3.403	4.188	1.571	4.712	2.880	7.853	4.188	3.403	…	5.497	1.309	7.330	6.806	8.115	5.236	0.785	3.403	5.497	no_efectores
498	6.667	5.390	5.106	6.950	0.993	9.504	3.972	5.816	2.270	4.965	…	6.241	2.695	2.979	4.681	8.511	4.681	1.135	2.411	5.957	no_efectores
499	7.517	4.784	4.100	3.872	1.708	4.214	2.506	6.492	2.278	6.948	…	3.645	3.645	6.036	5.239	8.200	6.948	0.569	2.733	8.884	no_efectores

[1000 rows x 21 columns]

[9]:
```
print (len(matrix_efec[0]))
```

21

4 Covarianza de auto cruzamiento (ACC)

[10]:
```
comp = "ACC"

for z in 1, 2, 3:
    if z == 1:
        indice = ind_1
        res_ind = "hidro_mass"

    if z == 2:
        indice = ind_2
        res_ind = "hidro"

    if z == 3:
        indice = [ind_3]
        res_ind = "mass"

    for x in tipo:
        for ind in indice:
            out = ACC.acc(open(x), k=1, lag=13, theta_type=3,phyche_list=[ind],␣
↪alphabet=index_list.PROTEIN)


            #Establece la etiqueta
            if x == str(efectores_fasta):
                etiq = "efectores"

            if x == str(no_efectores_fasta):
                etiq = "no_efectores"

            #nombre del txt
            nom = ('ds' + str(dataset) + '_' + str(comp) + '_' + str(res_ind) +␣
↪'_' + str(etiq) + '_' + str(organismo) + '.temp')

            #ruta al directorio donde se almacenara
            os.makedirs(str(r3), exist_ok=True)
            resultado = ((r3) + '/' + str(nom))

            with open(resultado, 'w') as f:
                print(out, file=f)

            print (indice)
            print (len(out[0]))
```

```
('Hydrophobicity', 'Hydrophilicity', 'Mass')
```

```
13
('Hydrophobicity', 'Hydrophilicity', 'Mass')
13
('Hydrophobicity', 'Hydrophilicity', 'Mass')
13
('Hydrophobicity', 'Hydrophilicity', 'Mass')
13
('Hydrophobicity', 'Hydrophilicity', 'Mass')
13
('Hydrophobicity', 'Hydrophilicity', 'Mass')
13
('Hydrophobicity', 'Hydrophilicity')
13
('Hydrophobicity', 'Hydrophilicity')
13
('Hydrophobicity', 'Hydrophilicity')
13
('Hydrophobicity', 'Hydrophilicity')
13
['Mass']
13
['Mass']
13
```

5 Composición de pseudo aminoácidos (PseAAC)

[11]:
```
#PseAAC
comp = "PseAAC"
k = 1 # no varia k para este algoritmo
w = 0.5 # peso de los índices
lamada = 21 # número de niveles para el factores de correlación
alphabet_list = index_list.PROTEIN
e = None
aa = False
theta_type = 2 #parallel

for z in 1, 2, 3:
    if z == 1:
        indice = ind_1
        res_ind = "hidro_mass"

    if z == 2:
        indice = ind_2
        res_ind = "hidro"

    if z == 3:
        indice = [ind_3]
```

```
        res_ind = "mass"

    for x in tipo:
        for ind in indice:
            out = PseAAC.pseknc(open(x), k, w, lamada, indice,␣
↪alphabet_list,extra_index_file=e, all_prop=aa, theta_type=theta_type)


            #Establece la etiqueta
            if x == str(efectores_fasta):
                etiq = "efectores"

            if x == str(no_efectores_fasta):
                etiq = "no_efectores"


            #nombre del txt
            nom = ('ds' + str(dataset) + '_' + str(comp) + '_' + str(res_ind) +␣
↪'_' + str(etiq) + '_' + str(organismo) + '.temp')

            #ruta al directorio donde se almacenara
            os.makedirs(str(r3), exist_ok=True)
            resultado = ((r3) + '/' + str(nom))


            with open(resultado, 'w') as f:
                print(out, file=f)

            print (indice)
            print (len(out[0]))
```

```
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity', 'Mass')
83
('Hydrophobicity', 'Hydrophilicity')
62
('Hydrophobicity', 'Hydrophilicity')
62
```

```
('Hydrophobicity', 'Hydrophilicity')
62
('Hydrophobicity', 'Hydrophilicity')
62
['Mass']
41
['Mass']
41
```

[ ]: