[Tutorial] SCN2A mutations in neurodevelopmental disorders

Contents

1 1. Introduction

1.1 1.1. Background

SCN2A is a voltage-gated sodium channel gene that encodes the neuronal sodium channel NaV1.2 and plays a critical role in action potential initiation during early neurodevelopment. The latest study demonstrated that it is loss of function mutations that in SCN2A that lead to autism spectrum disorders (ASD), in contrast to gain of function, which leads to infantile seizures (Ben-Shalom 2018).

In this tutorial, we will handle genetic data for SCN2A mutations identified in latest genomic studies, and then explore the data format to describe genetic mutations using R basic functions. Our tutorial will utilize the summary data from Sanders et al. (2018).

1.2 1.2. Aims

What we will do with this dataset,

Understand the dataset from a scientific journal

Apply some functions you have learnt from the Chapter 2 and 3

2 2. Explore your data

2.1 2.1. Unboxing your dataset

Here we obtain the list of mutations in the Supplementary Table 1 from Sanders et al. (2018).

Using the rio package, reading the excel file from the file link into your workspace. If you don't have the rio package in your system, please install as following:

```
# install.packages('rio')
```

Now you can read the file from the website. This will create the d object.

```
d = rio::import('https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6015533/bin/NIHMS957592-supplement-1.xlsx
```

Let's explore the object you just loaded. How would you check the class of the object `d`?

```
class(d)
```

```
## [1] "data.frame"
```

It shows that the `d` object is `data.frame`.

Then, let's overview the data frame. We will use head function to print out first few lines.

```
head(d)
```

```
##      PatientID PatientSex PatientAgeAtAssessment Chr  Pos_hg19       Ref
## 1            .          M                     7y   2 154370122 166384278
## 2            .          F                     3y   2 163706754 166506754
## 3     Patient2          F                    18m   2 163875903 166478766
## 4 Case1,280269          F                     4y   2 165798270 166304847
## 5            .          M                     3y   2 166019786 166249879
## 6            .          F                    25y   2 166060054 166153823
##                 Alt Type         Effect c.DNA p.Protein  Inheritence
## 1  12Mb Duplication  CNV DuplicationCNV     .         . DeNovoMosaic
## 2    2.8Mb Deletion  CNV    DeletionCNV     .         .       DeNovo
## 3 2.6Mb Duplication  CNV DuplicationCNV     .         .    Inherited
## 4 507kb Duplication  CNV DuplicationCNV     .         .    Inherited
## 5    230kb Deletion  CNV    DeletionCNV     .         .      Unknown
## 6     94kb Deletion  CNV    DeletionCNV     .         .       DeNovo
##                 Source SourcePMID Ben-Shalom2017 Wolff2017 AnyRecurrence
## 1    Vecchi et al 2011   21893419              Y         N             1
## 2      Chen et al 2010   20346423              Y         N             1
## 3 Yoshitomi et al 2015   25843248              Y         N             1
## 4 Thuresson et al 2016   27153334              Y         N             1
## 5     Celle et al 2013   24080482              Y         N             1
## 6   Bartnik et al 2011   20807223              Y         N             1
##   UniqueSample/Family TrueRecurrence Seizures SeizureOnsetDays SeizureType
## 1                   Y              1        Y               90     Unknown
## 2                   Y              1        N                .        None
## 3                   Y              1        Y                3     Unknown
```

```
## 4                   Y             1        Y                3     Unknown
## 5                   Y             1        N                .        None
## 6                   Y             1        Y              365     Unknown
##       ASD DD/ID DD/ID severity                                    OtherFeatures
## 1       N     Y           Mild                                       clumsiness
## 2       Y     Y         Severe                  dysmorphia, immature myelination
## 3 Unknown     Y         Severe                                 cerebral atrophy
## 4       N     N              .                                                .
## 5       Y     Y         Severe                                      microcephaly
## 6       N     Y       Moderate hypotonia, bipolar disorder, behavioral problems
##    Classification
## 1 IEE_Mild/Ataxia
## 2          ASD/DD
## 3             IEE
## 4             BIS
## 5          ASD/DD
## 6          ASD/DD
```

When you execute code in a notebook chunk, an output will be visible immediately beneath the input. From this, you can see several rows and columns in the data frame.

Let's look at the first column PatientID‘ and check which class it is.

```
head(d$PatientID)
```

```
## [1] "."            "."            "Patient2"     "Case1,280269" "."
## [6] "."
```

Cool. Now you can see the `TrueRecurrence` column. What is the class of the column `TrueRecurrence`?

```
class(d$TrueRecurrence)
```

```
## [1] "numeric"
```

To check the class of columns, you don't need to type an individual column. We can overview the summary of the dataset using `summary` function. Which column has the `character` class?

```
summary(d)
```

```
##   PatientID          PatientSex        PatientAgeAtAssessment      Chr
##  Length:293         Length:293         Length:293             Min.   :2
##  Class :character   Class :character   Class :character       1st Qu.:2
##  Mode  :character   Mode  :character   Mode  :character       Median :2
##                                                               Mean   :2
##                                                               3rd Qu.:2
##                                                               Max.   :2
##    Pos_hg19             Ref                Alt                Type
##  Length:293         Length:293         Length:293         Length:293
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##    Effect             c.DNA            p.Protein         Inheritence
##  Length:293         Length:293         Length:293         Length:293
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Source            SourcePMID        Ben-Shalom2017      Wolff2017
##  Length:293         Length:293         Length:293         Length:293
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  AnyRecurrence    UniqueSample/Family TrueRecurrence    Seizures
##  Min.   : 1.000   Length:293          Min.   :1.000   Length:293
##  1st Qu.: 1.000   Class :character    1st Qu.:1.000   Class :character
##  Median : 1.000   Mode  :character    Median :1.000   Mode  :character
##  Mean   : 2.119                       Mean   :1.768
##  3rd Qu.: 2.000                       3rd Qu.:1.000
##  Max.   :10.000                       Max.   :7.000
##  SeizureOnsetDays   SeizureType            ASD               DD/ID
##  Length:293         Length:293         Length:293         Length:293
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  DD/ID severity     OtherFeatures      Classification
##  Length:293         Length:293         Length:293
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

2.2 2.2. Difference between data frame and matrix

Here we will convert the data frame into a matrix, and compare which part will be different in this. To convert a data frame into a matrix, you can use the command called `as.matrix`.

```
m = as.matrix(d)
```

Let's overview the matrix object. Can you tell difference with data frame?

```
head(m[,'TrueRecurrence'])
```

```
##   1   2   3   4   5   6
## "1" "1" "1" "1" "1" "1"
```

2.3 2.3. Subset and Sort

Some patients who have the SCN2A mutation (hereafter called "SCN2A patient") often have seizures. So we want to know when the seizure occurs in development.

Let's check the class first.

```
class(d$SeizureOnsetDays)
```

```
## [1] "character"
```

Why this column contains character? Let's head the first few lines.

```
head(d$SeizureOnsetDays)
```

```
## [1] "90"  "."   "3"   "3"   "."   "365"
```

It seems that some rows contain samples who do not have seizure or unknown information. It's represented by `"."`, and also recorded in another column called `Seizures`.

```
head(d$Seizures)
```

```
## [1] "Y" "N" "Y" "Y" "N" "Y"
```

So we want to subset rows where the seizure phenotype is available.

```
d1 = d[d$Seizures=='Y',]
```

Let's see how many samples have seizure phenotypes? Then, you can ask when is the earliest days for the representation of seizure phenotype? How can we check this? The fisrt, as seen previously the SeizureOnsetDays column is character so we cannot apply functions for numeric.

```
head(d1$SeizureOnsetDays)
```

```
## [1] "90"   "3"    "3"    "365"  "30"   "1825"
```

So we have to convert this into numeric first.

```
d1$SeizureOnsetDays2 <- as.numeric(d1$SeizureOnsetDays)
```

```
## Warning:          NA
```

Hmm. There's an warning for `NA` introduction. This is because some rows do not have character that we can properly convert from character to numeric. So possible solutions are either you can bear with this in your downstream analyses or 2) convert character into an appropriate form of numeric conversion.

Then, the question is how can we find the rows with `NA`? We will ask whether the rows contains `NA` or not using `is.na` function. This will return boolean as to `NA` presence.

```r
# I couldn't find any  rows with NA
head(is.na(d1$SeizureOnsetDays2))
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
# There are rows with NA in these
is.na(d1$SeizureOnsetDays2)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE
```

See we can find some rows with `NA`. One of them is the 13th row. Let's see how it looks like.

```r
d1$SeizureOnsetDays[13]
```

```
## [1] "<365"
```

Here you have `<` (angle bracket) in the character so it won't properly converted to numeric information. Did you find more of these cases?

```r
d1[is.na(d1$SeizureOnsetDays2),]$SeizureOnsetDays
```

```
##  [1] "<365" "<365" "<365" "<28"  "<30"  "<365" "<365" "<365" "<365" "<365"
```

Our `NAs` all contains `<`, which prevent converting a character into a numeric.

We would fix for downstream analyses. For example, we can convert `<365` into `365`. One function we can try is `gsub`. This replace your string into a format that you may not get `NA`. For example,

```r
# gsub('pattern in your character', 'new character you want to replace', vectors for your character)
d1$SeizureOnsetDays3 <- gsub('<', '', d1$SeizureOnsetDays)
head(d1$SeizureOnsetDays3)
```

```
## [1] "90"   "3"    "3"    "365"  "30"   "1825"
```

Let's convert them into numerics.

```
d1$SeizureOnsetDays3 <- as.numeric(d1$SeizureOnsetDays3)
```

Did you get warning for this? Now we can ask our initial question. When is the earliest day for having seizure?

```
min(d1$SeizureOnsetDays3)
```

```
## [1] 0
```

3 3. Exercise

The dataset contains more details for genetic mutations in SCN2A patients. From this information, what can we analyze further? Here I list up few questions you can examine further. - Finding the position of the genetic mutations within SCN2A. Which information you would use? If you are not familiar with positional information on genetic variants (or mutations), please find the Figure 1 or the slides for Mutation (BSMS205 Session 3-1). - Counting the recurrent mutations at the same protein position (in other words, the same mutations seen across different patients), and examine whether the patients have similar phenotype. - Finding the position where different consequences mutations occur. Please note that "consequences" are loss-of-function (Nonsense, Frameshift) or missense. - Sketch a plot to visualize your analysis.

Figure 1. Standard mutation nomenclature (Ogino et al. 2007). According to this guideline, genetic mutations can be represented for coding DNA reference sequences (`c.` prefix) and protein-level amino acid sequences (`p.` prefix).

3.1 3.1. For-loops and Vectorization

Here we examine more details of genetic mutations as to their functional consequence and position of SCN2A mutations. In the dataset includes, there are two columns called `c.DNA` and `p.Protein`, containing the cDNA or protein position for the genetic mutations. During these exercises, we will look at the concept of `for-loop` and `vectorization`, which you learn from the Chapter 3.4. Let's look at the column `p.Protein`. It contains protein positions from each patient. What would you check at the first place? - Task 1 - Task 2 - Task 3 - …. - Task N

Let's write down your code to explore this column.

```
head(d$p.Protein)
```

```
## [1] "." "." "." "." "." "."
```

If you need more information on SCN2A, please visit the Uniprot description for SCN2A. The Uniprot database contains description for protein domains. Then, I would remove the characters from the string so we can have only numerics for positions. Here I use `gsub` function to extract numbers from string. Let's remove non-numeric characters from the string.

```
gsub('[^0-9]', '', 'p.R102X')
```

```
## [1] "102"
```

In the dataset, we have many rows for protein positions. One way we might try is to set up `for-loop` to process each row.

```
# for (i in 1:293) {
#  a[i] <- gsub('[^0-9]', '', d[i, 'p.Protein'] )
# }
```

Do you think this is an effective approach? As we have done in your assignment, `for-loop` is not a good choice to process vectors because R can do vectorization for this process with a shorter and clearer code. So this mean you can apply `gsub` on vector and return your output to another column (could be new assign).

```
help(gsub)
```

```
## starting httpd help server ... done
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
a <- d %>%
  mutate(d, protein_num = gsub('[^0-9]', '', p.Protein)) %>%
  select(p.Protein, protein_num)

a
```

```
##          p.Protein protein_num
## 1                .
## 2                .
## 3                .
## 4                .
## 5                .
## 6                .
## 7                .
## 8                .
## 9           p.D12N          12
## 10          p.R28C          28
## 11          p.R36G          36
## 12          p.R36G          36
## 13          p.R36G          36
## 14         p.V76fs          76
## 15          p.D82G          82
## 16         p.R102X         102
## 17         p.R102X         102
## 18      SpliceSite
## 19      SpliceSite
## 20         p.N132K         132
```

```
## 21         p.M136I         136
## 22         p.M136I         136
## 23      SpliceSite
## 24      SpliceSite
## 25         p.E169X         169
## 26         p.E169G         169
## 27         p.I172V         172
## 28         p.R188W         188
## 29         p.R188W         188
## 30         p.A202V         202
## 31      SpliceSite
## 32      SpliceSite
## 33         p.F207S         207
## 34         p.V208E         208
## 35         p.G211D         211
## 36         p.G211D         211
## 37         p.N212D         212
## 38         p.V213D         213
## 39         p.V213D         213
## 40         p.T218N         218
## 41         p.R220G         220
## 42         p.R223Q         223
## 43         p.T227I         227
## 44      SpliceSite
## 45         p.T236S         236
## 46         p.I237N         237
## 47         p.I237N         237
## 48         p.A240S         240
## 49         p.M252V         252
## 50         p.C258S         258
## 51         p.V261M         261
## 52         p.V261M         261
## 53         p.A263T         263
## 54         p.A263V         263
## 55         p.A263V         263
## 56         p.A263V         263
## 57         p.A263V         263
## 58         p.A263V         263
## 59         p.A263V         263
## 60         p.A263V         263
## 61         p.A263V         263
## 62         p.G266X         266
## 63         p.W281X         281
## 64         p.F328V         328
## 65         p.F328V         328
## 66         p.D343G         343
## 67         p.T365M         365
## 68         p.R379H         379
## 69         p.R379H         379
## 70         p.L389F         389
## 71         p.V423L         423
## 72         p.V423L         423
## 73         p.V424L         424
## 74         p.E430Q         430
```

```
## 75         p.E430A         430
## 76         p.E430G         430
## 77        p.A439fs         439
## 78        p.A439fs         439
## 79        p.E440fs         440
## 80        p.G478fs         478
## 81        p.K502fs         502
## 82         p.G564V         564
## 83        p.S565fs         565
## 84         p.R583X         583
## 85         p.R607X         607
## 86        p.D609fs         609
## 87         p.F612S         612
## 88         p.P614S         614
## 89         p.D649N         649
## 90         p.T674K         674
## 91        p.S686fs         686
## 92         p.S700X         700
## 93      SpliceSite
## 94       p.E717Gfs         717
## 95         p.A733T         733
## 96         p.N772S         772
## 97         p.T784M         784
## 98      SpliceSite
## 99         p.A812V         812
## 100        p.G828V         828
## 101        p.L849I         849
## 102        p.R850P         850
## 103        p.R853Q         853
## 104        p.R853Q         853
## 105        p.R853Q         853
## 106        p.R853Q         853
## 107        p.R853Q         853
## 108        p.R853Q         853
## 109        p.R853Q         853
## 110        p.R853Q         853
## 111        p.R853Q         853
## 112        p.R853Q         853
## 113        p.R856L         856
## 114        p.S863F         863
## 115        p.I873M         873
## 116        p.N876T         876
## 117        p.G879R         879
## 118        p.A880S         880
## 119        p.L881P         881
## 120        p.G882R         882
## 121        p.G882E         882
## 122        p.G882E         882
## 123        p.T885I         885
## 124        p.V887A         887
## 125        p.I891T         891
## 126        p.V892I         892
## 127        p.F895S         895
## 128        p.G899S         899
```

```
## 129        p.G899S         899
## 130        p.K905N         905
## 131        p.K905N         905
## 132        p.K908E         908
## 133        p.F928C         928
## 134        p.F928C         928
## 135        p.H930Q         930
## 136        p.R937C         937
## 137        p.R937C         937
## 138        p.R937H         937
## 139        p.C959X         959
## 140        p.M965R         965
## 141        p.N976K         976
## 142        p.F978L         978
## 143        p.F978L         978
## 144        p.L983W         983
## 145        p.S987I         987
## 146        p.E999K         999
## 147        p.E999K         999
## 148        p.E999K         999
## 149        p.E999K         999
## 150        p.E999K         999
## 151        p.E999K         999
## 152        p.E999V         999
## 153       p.N1001K        1001
## 154       p.L1003I        1003
## 155       p.G1013X        1013
## 156      p.I1021fs        1021
## 157      p.I1021fs        1021
## 158       p.D1050V        1050
## 159       p.F1114L        1114
## 160       p.E1115K        1115
## 161      p.E1125fs        1125
## 162       p.M1128T        1128
## 163      p.E1155fs        1155
## 164     p.C1170Vfs        1170
## 165       p.E1211K        1211
## 166       p.E1211K        1211
## 167       p.E1211K        1211
## 168       p.G1223R        1223
## 169       p.R1235X        1235
## 170       p.K1260E        1260
## 171       p.K1260Q        1260
## 172       p.I1281F        1281
## 173       p.V1282F        1282
## 174       p.V1282F        1282
## 175       p.A1310S        1310
## 176       p.R1312T        1312
## 177       p.R1319W        1319
## 178       p.R1319Q        1319
## 179       p.R1319Q        1319
## 180       p.R1319Q        1319
## 181       p.R1319Q        1319
## 182       p.R1319Q        1319
```

```
## 183       p.R1319Q        1319
## 184       p.R1319P        1319
## 185       p.R1319L        1319
## 186       p.E1321K        1321
## 187       p.M1323V        1323
## 188 SpliceSite3-10         310
## 189 SpliceSite3-10         310
## 190       p.V1326L        1326
## 191       p.V1326D        1326
## 192       p.L1330F        1330
## 193       p.S1336Y        1336
## 194       p.S1336Y        1336
## 195       p.M1338T        1338
## 196       p.L1341R        1341
## 197       p.L1342P        1342
## 198       p.L1342P        1342
## 199       p.L1342P        1342
## 200       p.L1342P        1342
## 201       p.C1386R        1386
## 202     p.K1387Sfs        1387
## 203       p.W1398X        1398
## 204       p.V1408A        1408
## 205 SpliceSite3-10         310
## 206       p.T1420M        1420
## 207       p.K1422E        1422
## 208       p.R1435X        1435
## 209       p.R1435X        1435
## 210       p.R1435X        1435
## 211     SpliceSite
## 212     SpliceSite
## 213       p.I1473M        1473
## 214       p.Q1479P        1479
## 215       p.Q1479P        1479
## 216       p.A1500T        1500
## 217      p.M1501fs        1501
## 218       p.R1515X        1515
## 219     SpliceSite
## 220       p.Q1521X        1521
## 221       p.G1522A        1522
## 222       p.G1522A        1522
## 223     p.V1528Cfs        1528
## 224       p.Q1531K        1531
## 225       p.Q1531K        1531
## 226       p.S1536R        1536
## 227      p.I1537fs        1537
## 228       p.M1545V        1545
## 229       p.M1548V        1548
## 230       p.L1563V        1563
## 231       p.G1576R        1576
## 232       p.Y1589C        1589
## 233       p.G1593R        1593
## 234       p.G1593R        1593
## 235       p.W1594R        1594
## 236       p.I1596S        1596
```

```
## 237       p.F1597L       1597
## 238       p.D1598G       1598
## 239       p.D1598V       1598
## 240    SpliceSite
## 241       p.P1622S       1622
## 242       p.T1623N       1623
## 243       p.R1626X       1626
## 244       p.R1626Q       1626
## 245       p.V1627M       1627
## 246       p.R1629H       1629
## 247       p.R1629H       1629
## 248       p.R1629L       1629
## 249      p.R1632fs       1632
## 250       p.G1634D       1634
## 251       p.G1634V       1634
## 252       p.R1635Q       1635
## 253       p.I1640S       1640
## 254       p.K1641N       1641
## 255       p.L1650P       1650
## 256       p.F1651C       1651
## 257       p.A1652P       1652
## 258       p.S1656F       1656
## 259       p.L1660W       1660
## 260       p.L1665F       1665
## 261    p.T1711Lfs       1711
## 262       p.W1716X       1716
## 263       p.C1731Y       1731
## 264       p.G1744R       1744
## 265       p.G1744E       1744
## 266       p.S1758R       1758
## 267       p.A1773T       1773
## 268       p.A1773V       1773
## 269       p.A1773V       1773
## 270       p.A1773V       1773
## 271       p.N1778I       1778
## 272       p.S1780I       1780
## 273       p.Q1811E       1811
## 274       p.L1829F       1829
## 275       p.H1853R       1853
## 276       p.L1875F       1875
## 277       p.E1880K       1880
## 278       p.R1882G       1882
## 279       p.R1882G       1882
## 280       p.R1882Q       1882
## 281       p.R1882Q       1882
## 282       p.R1882Q       1882
## 283       p.R1882Q       1882
## 284       p.R1882Q       1882
## 285       p.R1882Q       1882
## 286       p.R1882P       1882
## 287       p.R1882L       1882
## 288       p.R1902C       1902
## 289       p.R1902C       1902
## 290       p.Q1904E       1904
```

```
## 291       p.R1918H        1918
## 292       p.K1933M        1933
## 293       p.S1974L        1974
```

3.2 3.2. Counting the recurrent mutations

Recurrent mutations are the ones that the same genetic mutations occur in multiple individuals. Recurrent mutations can be common 1) when the mutation does not affect on natural selection, 2) when the mutation is beneficial, 3) in the hotspot for a disease or strongly associated with trait. However, given we are dealing with the genetic mutations from rare disorders, the mutations in the dataset are supposed to be uniquely present in general population. Otherwise, the recurrent mutations can indicate strong association with the phenotype.

To assess the recurrent mutations, the first thing we can try is to examine whether the same mutations occur in multiple individuals. Since the dataset contains individual patients for each row, we can simply check the frequency using:

```
c <- as.data.frame(table(d$p.Protein))
c
```

```
##               Var1 Freq
## 1                .    8
## 2         p.A1310S    1
## 3         p.A1500T    1
## 4         p.A1652P    1
## 5         p.A1773T    1
## 6         p.A1773V    3
## 7          p.A202V    1
## 8          p.A240S    1
## 9          p.A263T    1
## 10         p.A263V    8
## 11        p.A439fs    2
## 12         p.A733T    1
## 13         p.A812V    1
## 14         p.A880S    1
## 15      p.C1170Vfs    1
## 16        p.C1386R    1
## 17        p.C1731Y    1
## 18         p.C258S    1
## 19         p.C959X    1
## 20        p.D1050V    1
## 21          p.D12N    1
## 22        p.D1598G    1
## 23        p.D1598V    1
## 24         p.D343G    1
## 25        p.D609fs    1
## 26         p.D649N    1
## 27          p.D82G    1
## 28        p.E1115K    1
## 29       p.E1125fs    1
## 30       p.E1155fs    1
## 31        p.E1211K    3
## 32        p.E1321K    1
```

```
## 33         p.E169G    1
## 34         p.E169X    1
## 35        p.E1880K    1
## 36         p.E430A    1
## 37         p.E430G    1
## 38         p.E430Q    1
## 39        p.E440fs    1
## 40       p.E717Gfs    1
## 41         p.E999K    6
## 42         p.E999V    1
## 43        p.F1114L    1
## 44        p.F1597L    1
## 45        p.F1651C    1
## 46         p.F207S    1
## 47         p.F328V    2
## 48         p.F612S    1
## 49         p.F895S    1
## 50         p.F928C    2
## 51         p.F978L    2
## 52        p.G1013X    1
## 53        p.G1223R    1
## 54        p.G1522A    2
## 55        p.G1576R    1
## 56        p.G1593R    2
## 57        p.G1634D    1
## 58        p.G1634V    1
## 59        p.G1744E    1
## 60        p.G1744R    1
## 61         p.G211D    2
## 62         p.G266X    1
## 63        p.G478fs    1
## 64         p.G564V    1
## 65         p.G828V    1
## 66         p.G879R    1
## 67         p.G882E    2
## 68         p.G882R    1
## 69         p.G899S    2
## 70        p.H1853R    1
## 71         p.H930Q    1
## 72       p.I1021fs    2
## 73        p.I1281F    1
## 74        p.I1473M    1
## 75       p.I1537fs    1
## 76        p.I1596S    1
## 77        p.I1640S    1
## 78         p.I172V    1
## 79         p.I237N    2
## 80         p.I873M    1
## 81         p.I891T    1
## 82        p.K1260E    1
## 83        p.K1260Q    1
## 84      p.K1387Sfs    1
## 85        p.K1422E    1
## 86        p.K1641N    1
```

```
## 87        p.K1933M    1
## 88        p.K502fs    1
## 89         p.K905N    2
## 90         p.K908E    1
## 91        p.L1003I    1
## 92        p.L1330F    1
## 93        p.L1341R    1
## 94        p.L1342P    4
## 95        p.L1563V    1
## 96        p.L1650P    1
## 97        p.L1660W    1
## 98        p.L1665F    1
## 99        p.L1829F    1
## 100       p.L1875F    1
## 101        p.L389F    1
## 102        p.L849I    1
## 103        p.L881P    1
## 104        p.L983W    1
## 105       p.M1128T    1
## 106       p.M1323V    1
## 107       p.M1338T    1
## 108        p.M136I    2
## 109      p.M1501fs    1
## 110       p.M1545V    1
## 111       p.M1548V    1
## 112        p.M252V    1
## 113        p.M965R    1
## 114       p.N1001K    1
## 115        p.N132K    1
## 116       p.N1778I    1
## 117        p.N212D    1
## 118        p.N772S    1
## 119        p.N876T    1
## 120        p.N976K    1
## 121       p.P1622S    1
## 122        p.P614S    1
## 123       p.Q1479P    2
## 124       p.Q1521X    1
## 125       p.Q1531K    2
## 126       p.Q1811E    1
## 127       p.Q1904E    1
## 128        p.R102X    2
## 129       p.R1235X    1
## 130       p.R1312T    1
## 131       p.R1319L    1
## 132       p.R1319P    1
## 133       p.R1319Q    6
## 134       p.R1319W    1
## 135       p.R1435X    3
## 136       p.R1515X    1
## 137       p.R1626Q    1
## 138       p.R1626X    1
## 139       p.R1629H    2
## 140       p.R1629L    1
```

```
## 141      p.R1632fs    1
## 142       p.R1635Q    1
## 143       p.R1882G    2
## 144       p.R1882L    1
## 145       p.R1882P    1
## 146       p.R1882Q    6
## 147        p.R188W    2
## 148       p.R1902C    2
## 149       p.R1918H    1
## 150        p.R220G    1
## 151        p.R223Q    1
## 152         p.R28C    1
## 153         p.R36G    3
## 154        p.R379H    2
## 155        p.R583X    1
## 156        p.R607X    1
## 157        p.R850P    1
## 158        p.R853Q   10
## 159        p.R856L    1
## 160        p.R937C    2
## 161        p.R937H    1
## 162       p.S1336Y    2
## 163       p.S1536R    1
## 164       p.S1656F    1
## 165       p.S1758R    1
## 166       p.S1780I    1
## 167       p.S1974L    1
## 168       p.S565fs    1
## 169       p.S686fs    1
## 170        p.S700X    1
## 171        p.S863F    1
## 172        p.S987I    1
## 173       p.T1420M    1
## 174       p.T1623N    1
## 175     p.T1711Lfs    1
## 176        p.T218N    1
## 177        p.T227I    1
## 178        p.T236S    1
## 179        p.T365M    1
## 180        p.T674K    1
## 181        p.T784M    1
## 182        p.T885I    1
## 183       p.V1282F    2
## 184       p.V1326D    1
## 185       p.V1326L    1
## 186       p.V1408A    1
## 187     p.V1528Cfs    1
## 188       p.V1627M    1
## 189        p.V208E    1
## 190        p.V213D    2
## 191        p.V261M    2
## 192        p.V423L    2
## 193        p.V424L    1
## 194        p.V76fs    1
```

```
## 195        p.V887A    1
## 196        p.V892I    1
## 197       p.W1398X    1
## 198       p.W1594R    1
## 199       p.W1716X    1
## 200        p.W281X    1
## 201       p.Y1589C    1
## 202     SpliceSite   13
## 203 SpliceSite3-10    3
```

or we can check the number of unique variants in the dataset by:

```
b <- d %>%
  group_by(Ref, Alt) %>%
  count()

b
```

```
## # A tibble: 41 x 3
## # Groups:   Ref, Alt [41]
##    Ref       Alt                   n
##    <chr>     <chr>             <int>
##  1 .         .                     1
##  2 166153823 94kb Deletion         1
##  3 166198692 26kb Duplication      1
##  4 166249879 230kb Deletion        1
##  5 166304847 507kb Duplication     1
##  6 166384278 12Mb Duplication      1
##  7 166478766 2.6Mb Duplication     1
##  8 166506754 2.8Mb Deletion        1
##  9 166755313 671kb Duplication     1
## 10 A         ??                    1
## # ... with 31 more rows
```

How many unique variants you can find? and which variants are occurred in multiple times? Then, you can use other columns to check frequency for different groups. Which columns you would use for more grouping? If you find that, please check the recurrent mutations for each group.

```
data_1 <- d %>%
    group_by(PatientSex, Pos_hg19) %>%
  filter(PatientSex %in% c('M', 'F')) %>%
  count() %>%
  arrange(desc(n))

data_1
```

```
## # A tibble: 185 x 3
## # Groups:   PatientSex, Pos_hg19 [185]
##    PatientSex Pos_hg19      n
##    <chr>      <chr>     <int>
##  1 F          166198975     6
##  2 M          166229841     5
##  3 F          166231247     4
```

```
##  4 M          166166923     4
##  5 M          166210777     4
##  6 M          166245961     4
##  7 F          166245634     3
##  8 F          166245961     3
##  9 M          166198975     3
## 10 F          166153563     2
## # ... with 175 more rows
```

3.3 3.3. What is the proportion of diagnosis for SCN2A patient?

SCN2A mutation can have multiple different consequences for disease phenotypes. It can cause ASD but also other neurodevelopmental conditions. In total cases, how many phenotypes occur in SCN2A patients. Then, calculate the proportion of the phenotypes among total cases.

```
d %>% group_by(Classification) %>% count() %>%
  mutate(proportion = n / 293 *100)
```

```
## # A tibble: 7 x 3
## # Groups:   Classification [7]
##   Classification      n proportion
##   <chr>           <int>      <dbl>
## 1 ASD/DD             92      31.4
## 2 BIS                36      12.3
## 3 IEE               111      37.9
## 4 IEE_Mild/Ataxia     7       2.39
## 5 Other               3       1.02
## 6 Schizophrenia       5       1.71
## 7 Unclear            39      13.3
```

Then, you might be intrigued to whether females and males have different occurrence in each disorder. Let's check it.

```
d %>% filter(PatientSex %in% c("M", "F")) %>% group_by(PatientSex, Classification) %>% count() %>%
  mutate(proportion = n / 293 *100)
```

```
## # A tibble: 13 x 4
## # Groups:   PatientSex, Classification [13]
##    PatientSex Classification      n proportion
##    <chr>      <chr>           <int>      <dbl>
##  1 F          ASD/DD             26      8.87
##  2 F          BIS                11      3.75
##  3 F          IEE                57     19.5
##  4 F          IEE_Mild/Ataxia     3      1.02
##  5 F          Other               2      0.683
##  6 F          Schizophrenia       1      0.341
##  7 F          Unclear            14      4.78
##  8 M          ASD/DD             44     15.0
##  9 M          BIS                12      4.10
## 10 M          IEE                43     14.7
## 11 M          IEE_Mild/Ataxia     4      1.37
## 12 M          Other               1      0.341
## 13 M          Unclear            15      5.12
```

Another question you can ask is whether different mutation consequences occur in each phenotype. Let's find out how many mutation consequences are observed in each phenotype.

```
d %>% group_by(Classification, Effect) %>% count()
```

```
## # A tibble: 25 x 3
## # Groups:   Classification, Effect [25]
##    Classification Effect                      n
##    <chr>          <chr>                   <int>
##  1 ASD/DD         DeletionCNV                 3
##  2 ASD/DD         DuplicationCNV              1
##  3 ASD/DD         Frameshift                 17
##  4 ASD/DD         Missense                   40
##  5 ASD/DD         Nonsense                   16
##  6 ASD/DD         PopulationVariantInExAC     3
##  7 ASD/DD         SpliceSite                 12
##  8 BIS            DuplicationCNV              1
##  9 BIS            Frameshift                  1
## 10 BIS            Missense                   34
## # ... with 15 more rows
```

3.4 3.4. Find the position where different consequences of mutations occur

If you checked the recurrent mutations, you might want to find a locus where two or more variants occur. Such loci might indicate functionally important position of the gene and you might find some insight as to a cause of disease.

```
data <- d %>%
  group_by(Pos_hg19) %>%
  count() %>%
  arrange(desc(n)) %>%
  filter(n > 1)

data
```

```
## # A tibble: 44 x 2
## # Groups:   Pos_hg19 [44]
##    Pos_hg19      n
##    <chr>     <int>
##  1 166198975     9
##  2 166166923     8
##  3 166229841     8
##  4 166245961     8
##  5 166210777     6
##  6 166231247     4
##  7 166152439     3
##  8 166223837     3
##  9 166234155     3
## 10 166245202     3
## # ... with 34 more rows
```

3.5 3.5. Sketch a plot to visualize your analysis

When you examine the dataset, you would draw something to show your output. Though we haven't learnt how to plot data yet, we can have a quick sketch for the dataset. There's no restriction on your suggestion. Please submit your hand-drawing for the plot you would like to show from this dataset.

```
data_graph <- d %>%
  filter(PatientSex %in% c('F', 'M') & Inheritence == 'DeNovo') %>%
  group_by(Classification, PatientSex) %>%
  count()%>%
  group_by(PatientSex) %>%
  mutate(proportion = n / sum(n) * 100)

total <- d %>%
  filter(PatientSex %in% c('F', 'M') & Inheritence == 'DeNovo') %>%
  group_by(PatientSex) %>%
  count %>%
  mutate(total = sum(n))

total
```

```
## # A tibble: 2 x 3
## # Groups:   PatientSex [2]
##   PatientSex     n total
##   <chr>      <int> <int>
## 1 F             92    92
## 2 M             92    92
```

```
data_graph
```

```
## # A tibble: 11 x 4
## # Groups:   PatientSex [2]
##    Classification  PatientSex     n proportion
##    <chr>           <chr>      <int>      <dbl>
##  1 ASD/DD          F             25      27.2
##  2 ASD/DD          M             38      41.3
##  3 BIS             F              7       7.61
##  4 BIS             M              3       3.26
##  5 IEE             F             46      50
##  6 IEE             M             37      40.2
##  7 IEE_Mild/Ataxia F              2       2.17
##  8 IEE_Mild/Ataxia M              3       3.26
##  9 Schizophrenia   F              1       1.09
## 10 Unclear         F             11      12.0
## 11 Unclear         M             11      12.0
```

the reason that I thought that proportion was something meaningful, is because as seen as the object `total`, number of `Male`and `Female` patients with DeNovo mutation are equal (92)