Driver detection using dndscv

Adapted from Federico Abascal's practical

Detection of drivers in bladder carcinoma

We will be working with a dataset of bladder cancer from the TCGA consortium.

Generating the input file

dndscv works with an input file that consist of 5 columns. 1) Sample ID 2) Chromosome 3) Position 4) Reference 5) Mutation

We can extract this data from MAF files columns:

1. Tumor_Sample_Barcode
2. Chromosome
3. Start_Position
4. Reference_Allele
5. Tumor_Seq_Allele2

The input file can also be generated from vcf files using the following columns:

1. Tumor column (column number can differ based on the tool used to generate the vcf file)
2. Chrom
3. POS
4. REF
5. ALT

For this practical we have already prepared an input file for you `TCGA-BLCA.5col`. However, in your own work, you will have to typically generate this input file on your own. The easiest way to structure this input is by converting your variant calling output (.vcf formatted file) to .maf format. (See earlier discussion on the topic)

Data loading and exploration

Once we have an appropriately formatted 5 column input file with a suitable header we should load the data and explore it to ensure that the data are what we expect them to be and to check for any unexpected errors.

Load the input file

```
library(dndscv)
muts = read.table("TCGA-BLCA.5col", header=T, sep="\t", stringsAsFactors=F)
head(muts)
```

```
##      sampleID chr       pos ref mut
## 1 TCGA-2F-A9KO  10 101715548   C   T
## 2 TCGA-2F-A9KO  10 102822569   G   A
## 3 TCGA-2F-A9KO  10 103826020   C   T
## 4 TCGA-2F-A9KO  10 104160055   G   C
## 5 TCGA-2F-A9KO  10 118666167   C   T
## 6 TCGA-2F-A9KO  10  12043694   C   G
```

To count the number of samples we can run the following command:

```
length(unique(muts$sampleID))
```

```
## [1] 370
```

And to get the number of mutations

```
nrow(muts)
```

```
## [1] 53518
```

There are 370 donors and a total of 5318 mutations.

To see the mutational burden per sample we can do the following:

```
barplot(sort(table(muts$sampleID)), ylab="Number of mutations",
xlab="Donors", las=2, names.arg="")
```

> **Are there any hypermutators in the cohort?**

**t is relevant to explore our data in this way because hypermutators can have a negative impact on the statistical power to detect drivers and because some hypermutators (e.g. POLE mutant tumors) are under mutational processes not properly modeled by a trinucleotide-substitution model. Although there is no exact definition of a hypermutator, usually having more than 500 mutations in the exome (approximately 5.6810^{4} mutations in the whole genome) can be considered a hypermutator. In general, it is good to exclude these samples from this analysis.*

```

```

Driver detection

We will run dndscv to detect drivers in bladder cancer removing hypermutators (n>500)

- samples that have more than 3 mutations in a given gene (to protect against loss of sensitivity from clustered artefacts)

Gene level signals of selection

```
# analyses of selection using the dNdScv and dNdSloc models
dout = dndscv(muts,
max_muts_per_gene_per_sample=3,max_coding_muts_per_sample=50,
outmats=T)
```

```
## [1] Loading the environment...
## [2] Annotating the mutations...
##     Note: 1 samples excluded for exceeding the limit of mutations per
sample (see the max_coding_muts_per_sample argument in dndscv). 369 samples
left after filtering.
##     Note: 229 mutations removed for exceeding the limit of mutations per
gene per sample (see the max_muts_per_gene_per_sample argument in dndscv)
## 22% ...
##     43% ...
##     65% ...
##     86% ...
## [3] Estimating global rates...
## [4] Running dNdSloc...
## [5] Running dNdScv...
##     Regression model for substitutions (theta = 6.65).
##     Regression model for indels (theta = 0.422)
## Warning messages:
## 1: In dndscv(muts, max_muts_per_gene_per_sample = 3,
max_coding_muts_per_sample = 500,  :
##   Same mutations observed in different sampleIDs. Please verify that
these are independent events and remove duplicates otherwise.
## 2: In dndscv(muts, max_muts_per_gene_per_sample = 3,
max_coding_muts_per_sample = 500,  :
##   43 (0.093%) mutations have a wrong reference base (see the affected
mutations in dndsout$wrongmuts). Please identify the causes and rerun
dNdScv.
```

While running dndscv you will see some warnings. One of them reads "Same mutations observed in different sampleIDs. Please verify that these are independent events and remove duplicates otherwise." This warning relates to that only unique mutations are listed in the input file. For example if your input contains the same mutation in several related samples (samples that come from the same tumor) they should only be listed once in the file.

You will also see a warning indicating that some mutations have a wrong reference. This is because of a error in the original TCGA file. We can ignore this as the number of affected bases is very small.

Looking at the output

dndscv generates a list of objects as output. You can look at the contents of the list like this.

```
names(dout)
```

```
## [1] "globaldnds"   "sel_cv"       "sel_loc"      "annotmuts"
"genemuts"
## [6] "mle_submodel" "exclsamples"  "exclmuts"     "nbreg"
"nbregind"
## [11] "poissmodel"   "wrongmuts"    "N"            "L"
```

The most relevant output often is `sel_cv` as it contains the results of the neutrality tests at gene level. The `globaldnds` output has a table with global MLEs for the dN/dS ratios across all genes and their confidence intervals. The `annotmuts` output contains a table with annotated coding mutations. `genemuts` has a table with observed and expected number of mutations per gene.

Table of significant genes

`dout$sel_cv` contains the results for all the analyzed genes.

We can look at the significant genes in the table by filtering for genes with a `qglobal_cv < 0.1`. `qglobal` is the multiple hypothesis correction q-value for the `pglobal_cv` (the combined p-value for the different p-values calculated.

```
dout$sel_cv[which(dout$sel_cv$qglobal_cv<0.1),]
```

```
##             gene_name n_syn n_mis n_non n_spl n_ind   wmis_cv    wnon_cv
## 18057            TP53     3    82    14     1    12 53.092352  85.566153
## 12977          PIK3CA     1    44     0     0     0 18.717174   0.000000
## 1465           ARID1A     2    21    29     2    16  4.482524  62.492704
## 9207            KMT2D     6    19    24     5    16  1.465238  20.858366
## 14249             RB1     0     3    21     9    11  1.635651 110.255813
## 8939            KDM6A     3    12    13     5    12  3.368083  34.351581
## 16808           STAG2     1     8    11     2     8  3.286180  41.225276
## 5641             ELF3     0    18     1     0    12 18.535061   8.812791
## 3519           CDKN1A     1     3     3     0    13  6.602851  51.166925
## 6523            FGFR3     3    26     1     0     0 13.997261   5.943960
## 3523  CDKN2A.p16INK4a     0     6     1     1     3 16.344982  84.538504
## 3522    CDKN2A.p14arf     0     6     0     1     3 18.718600  63.957183
## 14506            RHOB     0    14     0     0     0 26.026691   0.000000
## 14505            RHOA     1    12     0     0     0 26.045683   0.000000
```

```
## 13874            PTEN     0     7     4     1     1  7.777147  50.520840
## 1635             ASXL2     0    17     8     0     0  6.352242  26.791178
## 6426             FBXW7     0     9     3     1     2  7.057201  25.595899
## 19448         ZFP36L1     0     1     1     0     8  1.098817  18.630778
## 6656             FOXA1     0     4     0     0     5  4.274081   0.000000
## 4293            CREBBP     1    11     7     1     2  2.752218  17.224390
## 5752             EP300     1    19     5     0     5  4.229062   8.420204
## 14268            RBM10     1     4     4     0     3  2.331943  20.529854
## 9225              KRAS     0     9     0     0     0 23.374342   0.000000
## 6343              FAT1     2    13    10     0     2  1.481619  13.306738
## 6701             FOXQ1     2     2     1     0     3  3.096953  37.951801
## 18313             TSC1     2     4     4     1     3  1.252455  15.111819
## 8107              HRAS     0     7     1     0     0 21.676870  47.063935
## 5815             ERBB2     3    21     0     0     0  7.039553   0.000000
## 9204             KMT2A     3    10     8     1     3  1.108220   8.810449
## 1467             ARID2     2     6     7     1     1  1.404154  12.193498
## 19023              WAC     2     5     5     0     1  2.581614  19.269769
## 5817             ERBB3     1    18     0     0     1  5.696714   0.000000
## 15211             RXRA     3    11     0     0     0  9.567561   0.000000
##          wspl_cv    wind_cv      pmis_cv    ptrunc_cv  pallsubs_cv
pind_cv
## 18057  85.566153 390.951687 0.000000e+00 0.000000e+00 0.000000e+00
1.107250e-15
## 12977   0.000000   0.000000 0.000000e+00 4.164436e-01 0.000000e+00
1.000000e+00
## 1465   62.492704  85.270823 1.244612e-04 0.000000e+00 0.000000e+00
7.582493e-10
## 9207   20.858366  41.443226 2.985509e-01 0.000000e+00 0.000000e+00
9.971435e-07
## 14249 110.255813 197.681548 5.034079e-01 0.000000e+00 0.000000e+00
6.788964e-12
## 8939   34.351581 142.634878 6.330144e-03 0.000000e+00 4.440892e-16
5.530143e-11
## 16808  41.225276 103.717907 2.484068e-02 7.460699e-14 6.666889e-13
5.015379e-08
## 5641    8.812791 204.090325 2.071909e-11 1.175565e-01 1.314769e-10
1.353943e-12
## 3519   51.166925 527.416382 1.996656e-02 4.628656e-05 3.451268e-05
4.903854e-18
## 6523    5.943960   0.000000 1.522116e-13 1.766996e-01 1.245115e-12
1.000000e+00
## 3523   84.538504 259.398648 2.057808e-05 2.834791e-04 4.982340e-07
4.657565e-06
## 3522   63.957183 305.699732 9.640805e-06 1.287712e-02 4.680562e-06
2.875384e-06
## 14506   0.000000   0.000000 1.274147e-11 8.430716e-01 1.037418e-10
1.000000e+00
## 14505   0.000000   0.000000 7.926348e-11 7.795986e-01 5.960913e-10
1.000000e+00
## 13874  50.520840  31.507846 5.132174e-04 3.252610e-07 5.357811e-08
3.013691e-02
## 1635   26.791178   0.000000 2.825636e-05 4.460292e-08 6.342425e-09
1.000000e+00
## 6426   25.595899  41.439451 2.765791e-04 5.895216e-05 3.729346e-06
```

```
3.295084e-03
## 19448  18.630778  81.996171 9.315351e-01 5.100517e-02 1.487340e-01
2.432690e-07
## 6656    0.000000 138.809332 3.929494e-02 7.620844e-01 1.117816e-01
5.748984e-07
## 4293   17.224390  11.237798 3.529453e-02 7.906942e-07 3.915134e-06
3.009566e-02
## 5752    8.420204  24.663421 5.242048e-04 1.377050e-03 1.783346e-04
7.940419e-04
## 14268  20.529854  58.486753 2.016139e-01 1.220962e-04 4.988068e-04
3.242623e-04
## 9225    0.000000   0.000000 2.936258e-08 7.661320e-01 1.889170e-07
1.000000e+00
## 6343   13.306738   3.961671 3.815497e-01 4.278626e-07 1.771352e-06
1.176079e-01
## 6701   37.951801 165.321479 2.054158e-01 2.249853e-02 3.798430e-02
1.730825e-05
## 18313  15.111819  30.963425 7.186328e-01 8.240048e-05 3.941722e-04
1.723682e-03
## 8107   47.063935   0.000000 9.217760e-07 1.810804e-02 7.129282e-07
1.000000e+00
## 5815    0.000000   0.000000 2.650661e-07 4.455630e-01 8.407223e-07
1.000000e+00
## 9204    8.810449  10.226782 8.246637e-01 2.286913e-05 5.600933e-05
2.078576e-02
## 1467   12.193498   8.824722 5.373321e-01 7.016978e-06 3.399276e-05
9.549333e-02
## 19023  19.269769  17.388350 1.160744e-01 2.578513e-05 9.763099e-05
5.248399e-02
## 5817    0.000000   7.032822 3.336318e-05 3.809935e-01 6.903687e-05
1.153199e-01
## 15211   0.000000   0.000000 2.278009e-06 6.643021e-01 1.163581e-05
1.000000e+00
##            qmis_cv    qtrunc_cv  qallsubs_cv   pglobal_cv   qglobal_cv
## 18057 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## 12977 0.000000e+00 9.023868e-01 0.000000e+00 0.000000e+00 0.000000e+00
## 1465  1.470912e-01 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## 9207  8.077001e-01 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## 14249 8.150777e-01 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## 8939  8.077001e-01 0.000000e+00 1.487033e-12 0.000000e+00 0.000000e+00
## 16808 8.077001e-01 2.498215e-10 1.913492e-09 0.000000e+00 0.000000e+00
## 5641  8.325346e-08 9.023868e-01 2.641503e-07 0.000000e+00 0.000000e+00
## 3519  8.077001e-01 6.199622e-02 2.889143e-02 0.000000e+00 0.000000e+00
## 6523  1.019361e-09 9.023868e-01 3.126951e-09 3.537592e-11 7.107377e-08
## 3523  3.445286e-02 2.432097e-01 6.673346e-04 6.448642e-11 1.177815e-07
## 3522  1.760849e-02 9.023868e-01 4.477960e-03 3.503414e-10 5.865591e-07
## 14506 6.399724e-08 9.023868e-01 2.315863e-07 2.488674e-09 3.846150e-06
## 14505 2.654138e-07 9.023868e-01 1.088734e-06 1.325744e-08 1.902538e-05
## 13874 4.050691e-01 8.168524e-04 8.280291e-05 3.430244e-08 4.594469e-05
## 1635  4.366911e-02 1.280168e-04 1.061881e-05 1.260621e-07 1.582946e-04
## 6426  2.778376e-01 7.402550e-02 3.932947e-03 2.361187e-07 2.790507e-04
## 19448 9.790988e-01 9.023868e-01 9.503354e-01 6.561564e-07 7.323799e-04
## 6656  8.077001e-01 9.023868e-01 9.503354e-01 1.128477e-06 1.193275e-03
## 4293  8.077001e-01 1.588584e-03 3.932947e-03 1.997669e-06 2.006758e-03
```

```
## 5752  4.050691e-01 7.373400e-01 1.193016e-01 2.374750e-06 2.271957e-03
## 14268 8.077001e-01 1.291071e-01 2.708521e-01 2.690979e-06 2.457475e-03
## 9225  8.427481e-05 9.023868e-01 2.711094e-04 3.113722e-06 2.719904e-03
## 6343  8.077001e-01 9.551320e-04 1.977124e-03 3.413231e-06 2.857301e-03
## 6701  8.077001e-01 9.023868e-01 9.503354e-01 1.001607e-05 7.981278e-03
## 18313 8.957163e-01 9.738283e-02 2.262661e-01 1.032867e-05 7.981278e-03
## 8107  2.057711e-03 9.023868e-01 8.952150e-04 1.080363e-05 8.039102e-03
## 5815  6.656804e-04 9.023868e-01 9.935854e-04 1.260159e-05 9.042091e-03
## 9204  9.399582e-01 3.534336e-02 4.328013e-02 1.707117e-05 1.182679e-02
## 1467  8.207039e-01 1.196232e-02 2.889143e-02 4.427026e-05 2.964780e-02
## 19023 8.077001e-01 3.700351e-02 7.005372e-02 6.754316e-05 4.377451e-02
## 5817  4.468664e-02 9.023868e-01 5.137110e-02 1.014346e-04 6.368507e-02
## 15211 4.576747e-03 9.023868e-01 1.062614e-02 1.438352e-04 8.756946e-02
```

The `sel_cv` table contains `3` types of columns:

- the data: the number of mutations of each class for each gene
- the coefficients of selection for mutations of each mutation class *(w)*
- the associated statistical significance values each mutation class *(p and q values) for each class*

> **How many significant genes do you find?**

```

```

> **Is there any gene under negative selection?**

```

```

> **Which genes are oncogenes? Which genes are tumor suppressors?**

- *Tip: look at the number and types of mutations n_syn, n_mis.....*

```

```

> **Considering the coefficient of selection for missense mutations in *ARID1A* how many missense mutations had been selected for in this cohort?**

- *Tip: the coefficient wmis_cvis 4.4825244 and there are 21 missense mutations in*ARID1A.*
- *Tip 2: (w-1)/w gives the proportion under positive selection.*
- *Tip 3: 95% confidence intervals for the selection coefficients can be obtained with `geneci(dout,gene_list="ARID1A")`.*
- *Tip 4: Have a look at `genemuts` to see how many mutations were expected**

```
dout$genemuts[which(dout$genemuts$gene_name=="ARID1A"),]
```

```
##      gene_name n_syn n_mis n_non n_spl  exp_syn  exp_mis   exp_non
exp_spl
## 1465    ARID1A     2    21    29     2 2.100349 5.592629 0.5095613
0.082616
##      exp_syn_cv
## 1465   1.986097
```

> **Are all those missense mutations under selection?**

Let's take a look at the mutations in *PIK3CA*:

```
dout$annotmuts[which(dout$annotmuts$gene=="PIK3CA"),]
```

```
          sampleID chr       pos ref mut   gene strand ref_cod mut_cod
ref3_cod
255   TCGA-2F-A9KO   3 178938934   G   A PIK3CA      1       G       A
TGA
1028  TCGA-2F-A9KW   3 178936091   G   A PIK3CA      1       G       A
TGA
3289  TCGA-4Z-AA84   3 178941935   C   G PIK3CA      1       C       G
TCT
3618  TCGA-4Z-AA87   3 178937518   G   C PIK3CA      1       G       C
AGT
3717  TCGA-4Z-AA89   3 178916891   G   A PIK3CA      1       G       A
CGG
3718  TCGA-4Z-AA89   3 178921553   T   A PIK3CA      1       T       A
ATG
3864  TCGA-5N-A9KI   3 178921339   G   A PIK3CA      1       G       A
AGA
3865  TCGA-5N-A9KI   3 178936091   G   A PIK3CA      1       G       A
TGA
5451  TCGA-BT-A20O   3 178936082   G   A PIK3CA      1       G       A
TGA
5759  TCGA-BT-A20R   3 178936091   G   A PIK3CA      1       G       A
TGA
9814  TCGA-CF-A5UA   3 178916836   C   G PIK3CA      1       C       G
TCA
10685 TCGA-CU-A5W6   3 178936094   C   A PIK3CA      1       C       A
GCA
11346 TCGA-DK-A1A5   3 178942564   G   C PIK3CA      1       G       C
AGA
11884 TCGA-DK-A1AB   3 178952074   G   T PIK3CA      1       G       T
TGA
```

```
14751 TCGA-DK-A6B2   3 178948096   G   C PIK3CA      1       G       C
TGA
14863 TCGA-DK-A6B5   3 178936082   G   A PIK3CA      1       G       A
TGA
15379 TCGA-DK-AA6Q   3 178936091   G   A PIK3CA      1       G       A
TGA
17541 TCGA-DK-AA77   3 178936091   G   A PIK3CA      1       G       A
TGA
18710 TCGA-E7-A4IJ   3 178936082   G   A PIK3CA      1       G       A
TGA
21312 TCGA-FD-A3B5   3 178936091   G   A PIK3CA      1       G       A
TGA
21520 TCGA-FD-A3B6   3 178936091   G   C PIK3CA      1       G       C
TGA
22072 TCGA-FD-A3NA   3 178928225   C   G PIK3CA      1       C       G
TCC
22762 TCGA-FD-A3SN   3 178936082   G   A PIK3CA      1       G       A
TGA
24545 TCGA-FD-A5BX   3 178916810   C   G PIK3CA      1       C       G
TCT
24546 TCGA-FD-A5BX   3 178922324   G   A PIK3CA      1       G       A
AGA
24740 TCGA-FD-A5C0   3 178937838   A   G PIK3CA      1       A       G
TAA
24836 TCGA-FD-A5C1   3 178952085   A   G PIK3CA      1       A       G
CAT
27941 TCGA-G2-A2EJ   3 178928074   G   T PIK3CA      1       G       T
GGA
29777 TCGA-G2-AA3B   3 178936091   G   A PIK3CA      1       G       A
TGA
31238 TCGA-GC-A3WC   3 178936095   A   G PIK3CA      1       A       G
CAG
32008 TCGA-GD-A3OP   3 178927486   G   A PIK3CA      1       G       A
AGA
33711 TCGA-GU-AATQ   3 178936091   G   A PIK3CA      1       G       A
TGA
36422 TCGA-HQ-A5NE   3 178936091   G   C PIK3CA      1       G       C
TGA
37942 TCGA-K4-A83P   3 178928079   G   A PIK3CA      1       G       A
AGA
42362 TCGA-XF-A8HI   3 178936082   G   A PIK3CA      1       G       A
TGA
46250 TCGA-XF-AAME   3 178952085   A   G PIK3CA      1       A       G
CAT
47645 TCGA-XF-AAN0   3 178936091   G   A PIK3CA      1       G       A
TGA
47958 TCGA-XF-AAN2   3 178936082   G   A PIK3CA      1       G       A
TGA
50145 TCGA-ZF-A9RE   3 178952090   G   C PIK3CA      1       G       C
TGG
50478 TCGA-ZF-A9RG   3 178928079   G   A PIK3CA      1       G       A
AGA
50479 TCGA-ZF-A9RG   3 178936091   G   A PIK3CA      1       G       A
TGA
```

```
51098 TCGA-ZF-AA4U   3 178951955   A   G PIK3CA      1       A       G
AAT
52414 TCGA-ZF-AA4X   3 178936082   G   A PIK3CA      1       G       A
TGA
53285 TCGA-ZF-AA56   3 178936082   G   A PIK3CA      1       G       A
TGA
53506 TCGA-ZF-AA5P   3 178916876   G   A PIK3CA      1       G       A
CGA
      mut3_cod aachange ntchange codonsub     impact             pid
255        TAA    E726K   G2176A  GAA>AAA   Missense ENSP00000263967
1028       TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
3289       TGT    L752V   C2254G  CTG>GTG   Missense ENSP00000263967
3618       ACT    V636L   G1906C  GTA>CTA   Missense ENSP00000263967
3717       CAG     R93Q    G278A  CGG>CAG   Missense ENSP00000263967
3718       AAG    N345K   T1035A  AAT>AAA   Missense ENSP00000263967
3864       AAA    R274K    G821A  AGA>AAA   Missense ENSP00000263967
3865       TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
5451       TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
5759       TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
9814       TGA     Q75E    C223G  CAA>GAA   Missense ENSP00000263967
10685      GAA    Q546K   C1636A  CAG>AAG   Missense ENSP00000263967
11346      ACA    E791Q   G2371C  GAG>CAG   Missense ENSP00000263967
11884      TTA   M1043I   G3129T  ATG>ATT   Missense ENSP00000263967
14751      TCA    L956F   G2868C  TTG>TTC   Missense ENSP00000263967
14863      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
15379      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
17541      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
18710      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
21312      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
21520      TCA    E545Q   G1633C  GAG>CAG   Missense ENSP00000263967
22072      TGC    P471A   C1411G  CCA>GCA   Missense ENSP00000263967
22762      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
24545      TGT     S66C    C197G  TCT>TGT   Missense ENSP00000263967
24546      AAA    E365K   G1093A  GAA>AAA   Missense ENSP00000263967
24740      TGA    L671L   A2013G  TTA>TTG Synonymous ENSP00000263967
24836      CGT   H1047R   A3140G  CAT>CGT   Missense ENSP00000263967
27941      GTA    G451V   G1352T  GGA>GTA   Missense ENSP00000263967
29777      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
31238      CGG    Q546R   A1637G  CAG>CGG   Missense ENSP00000263967
32008      AAA    E417K   G1249A  GAG>AAG   Missense ENSP00000263967
33711      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
36422      TCA    E545Q   G1633C  GAG>CAG   Missense ENSP00000263967
37942      AAA    E453K   G1357A  GAA>AAA   Missense ENSP00000263967
42362      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
46250      CGT   H1047R   A3140G  CAT>CGT   Missense ENSP00000263967
47645      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
47958      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
50145      TCG   G1049R   G3145C  GGT>CGT   Missense ENSP00000263967
50478      AAA    E453K   G1357A  GAA>AAA   Missense ENSP00000263967
50479      TAA    E545K   G1633A  GAG>AAG   Missense ENSP00000263967
51098      AGT   M1004V   A3010G  ATG>GTG   Missense ENSP00000263967
52414      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
53285      TAA    E542K   G1624A  GAA>AAA   Missense ENSP00000263967
53506      CAA     R88Q    G263A  CGA>CAA   Missense ENSP00000263967
```

Look at the aachange column to see the amino acid changes generated by the mutations.

> **Is there any recurrent mutation (hotspot)?**

```
PIK3CA = dout$annotmuts[which(dout$annotmuts$gene=="PIK3CA"),]
table(PIK3CA$aachange)
```

```
 E365K  E417K  E453K  E542K  E545K  E545Q  E726K  E791Q G1049R  G451V
H1047R
     1      1      2      8     10      2      1      1      1      1
2
 L671L  L752V  L956F M1004V M1043I  N345K  P471A  Q546K  Q546R   Q75E
R274K
     1      1      1      1      1      1      1      1      1      1
1
  R88Q   R93Q   S66C  V636L
     1      1      1      1
```

Global signals of selection

dndscv also estimates global dN/dS ratios in aggregate for all the genes. You can access this data in the dndscv output:

```
print(dout$globaldnds)
```

```
     name       mle     cilow    cihigh
wmis wmis 1.0524348 1.0290546 1.0763461
wnon wnon 1.2269231 1.1735965 1.2826728
wspl wspl 0.7915522 0.7297038 0.8586429
wtru wtru 1.1045772 1.0612096 1.1497171
wall wall 1.0629959 1.0398517 1.0866553
```

> **Is there evidence of positive or negative selection?**

`wspl` is lower than 1. That could mean negative selection but this result is often obtained with exome data because of the poorer sequencing coverage at splice sites. `dndscv` interprets the depletion of mutations at splice sites as negative selection.

However all the other coefficients are > 1 and their 95% confidence intervals too.

We can use the `globaldnds` information to estimate the number of missense driver mutations per sample.

- There are 30610 missense mutations in the cohort, and the coefficient of selection `wmis` is 1.0524348.
- Calculate the proportion of missense mutations under positive selection using the formula* `(w-1)/w`
- Find out the actual number of missense mutations under positive selection: `n_mis * (w-1)/w`
- Calculate the average per sample: `( n_mis * (w-1)/w ) / num_samples`

You can obtain all the info with:

```
w = dout$globaldnds[1,2]

n_mis = length(which(dout$annotmuts$impact=="Missense"))

num_samples = length(table(unique(muts$sampleID)))
```

Site/Codon level signals of selection

Analysis of hotspots

We will now look for signals of positive selection at specific DNA or protein sites.

Firstly, have a look at the `annotmuts` output and try to determine by eye if there are hotspots. A couple lines of code which may help with the task:

```
dout$annotmuts$gene_and_aachange = paste(dout$annotmuts$gene,
dout$annotmuts$aachange, dout$annotmuts$ntchange, dout$annotmuts$pos,
dout$annotmuts$impact, sep=":")

sort(table(dout$annotmuts$gene_and_aachange),decreasing=T)[1:10]
```

```
    FGFR3:S249C:C746G:1803568:Missense
TP53:R248Q:G743A:7577538:Missense
                                    13
11
PIK3CA:E545K:G1633A:178936091:Missense
PIK3CA:E542K:G1624A:178936082:Missense
                                    10
8
  RXRA:S427F:C1280T:137328351:Missense
TP53:E285K:G853A:7577085:Missense
                                     7
6
   ERBB2:S310F:C929T:37868208:Missense
FGFR3:Y375C:A1124G:1806099:Missense
```

```
                                    5
5
     TP53:R280T:G839C:7577099:Missense
C3orf70:S6L:C17T:184870595:Missense
                                    5
4
```

Go to the COSMIC database to gather further information about these hotspots. For example:

FGFR3 https://cancer.sanger.ac.uk/cosmic/gene/analysis?ln=FGFR3

Look at other hotspots, the domain structure, the 3D structure

The Hallmarks of Cancer has also valuable information on drivers:
https://cancer.sanger.ac.uk/cosmic/census-page/FGFR3

> **Do you think hotspots are more frequent in oncogenes or in tumour suppressors?**

Using sitednds: selection at specific sites

Running sitednds

One of the limitations of `sitednds` is that artefacts and contamination are common in cancer datasets and can generate false positive mutation calls. To reduce the risk of false positives and increase the signal to noise ratio, we will only consider mutations in Cancer Gene Census genes (v81).

The `sitednds` function takes the output of `dndscv` as input. In order for the `dndsout` object to be compatible with sitednds we must use the `"outmats=T"` argument in `dndscv`.

```
# Load the the Cancer Gene Census (v81) genes
data("cancergenes_cgc81", package="dndscv")
dout_cancergenes = dndscv(muts, outmats=T, gene_list=known_cancergenes)
```

```
[1] Loading the environment...
[2] Annotating the mutations...
    Note: 43 mutations removed for exceeding the limit of mutations per
gene per sample (see the max_muts_per_gene_per_sample argument in dndscv)
[3] Estimating global rates...
[4] Running dNdSloc...
[5] Running dNdScv...
    Regression model for substitutions (theta = 8.21).
    Regression model for indels (theta = 0.385)
Warning messages:
1: In dndscv(muts, outmats = T, gene_list = known_cancergenes) :
  Same mutations observed in different sampleIDs. Please verify that these
are independent events and remove duplicates otherwise.
```

```
2: In dndscv(muts, outmats = T, gene_list = known_cancergenes) :
  2 (0.067%) mutations have a wrong reference base (see the affected
mutations in dndsout$wrongmuts). Please identify the causes and rerun
dNdScv.
3: In dndscv(muts, outmats = T, gene_list = known_cancergenes) :
      Genes were excluded from the indel background model based on the
substitution data: TP53, PIK3CA, ARID1A, KMT2D, RB1, KDM6A, FGFR3, STAG2,
RHOA, PTEN, CDKN2A.p16INK4a, KRAS, ERBB2, CDKN2A.p14arf, FAT1, CREBBP,
HRAS, FBXW7, ARID2, KMT2A, ERBB3, EP300.
```

```
sout = sitednds(dout_cancergenes)
```

The output list contains the following objects:

```
names(sout)
```

`recursites` has information on the significant sites

```
[1] "recursites"     "overdisp"       "fpr_nonsyn_q05" "LL"
```

```
sout$recursites[which(sout$recursites$qval<0.1),]
```

```
   chr       pos ref mut   gene aachange   impact ref3_cod mut3_cod freq
1    4   1803568   C   G  FGFR3    S249C Missense      TCC      TGC   13
2   17   7577538   C   T   TP53    R248Q Missense      CGG      CAG   11
3    3 178936091   G   A PIK3CA    E545K Missense      TGA      TAA   10
4    3 178936082   G   A PIK3CA    E542K Missense      TGA      TAA    8
5    4   1806099   A   G  FGFR3    Y375C Missense      TAT      TGT    5
6   19  45867687   T   C  ERCC2    N238S Missense      AAC      AGC    4
7    4 153247289   G   C  FBXW7    R505G Missense      CCG      CGG    4
8   17  37868208   C   T  ERBB2    S310F Missense      TCC      TTC    5
9   17   7577085   C   T   TP53    E285K Missense      AGA      AAA    6
10  17   7578454   G   A   TP53    A159V Missense      GCC      GTC    3
11  17   7577099   C   G   TP53    R280T Missense      AGA      ACA    5
12  17   7577539   G   A   TP53    R248W Missense      CCG      CTG    4
13  12  56478854   G   T  ERBB3    V104L Missense      CGT      CTT    3
             mu       dnds         pval         qval
1  0.0008079132 16090.8379 7.376318e-29 3.573132e-22
2  0.0038932950  2825.3703 1.939040e-17 4.696412e-11
3  0.0046875620  2133.3051 2.692634e-15 4.347760e-09
4  0.0046875620  1706.6441 1.088835e-12 1.318595e-06
5  0.0007836589  6380.3270 1.592950e-12 1.543269e-06
6  0.0003810892 10496.2291 1.149977e-11 9.284260e-06
```

```
7  0.0005224576  7656.1236 4.039331e-11 2.795251e-05
8  0.0016794405  2977.1820 6.874221e-11 4.162392e-05
9  0.0051158831  1172.8181 7.681553e-10 4.134433e-04
10 0.0007043649  4259.1560 1.981545e-08 9.598719e-03
11 0.0056141654   890.6043 2.353526e-08 1.036420e-02
12 0.0033636803  1189.1736 6.200800e-08 2.503087e-02
13 0.0011948375  2510.8016 9.538711e-08 3.554314e-02
```

This output shows the hotspots studied, their position, the gene affected, amino acid change, the number of times the mutation was observed in the data, the number of expected mutations at the site by chance, the dN/dS ratio and significance values.

Using codondnds: selection at specific codons

Running codondnds

We will not run it because it requires creating a new database, which can take about 20', but this is how one would do it.

```
data("refcds_hg19", package = "dndscv")
RefCDS_codon = buildcodon(RefCDS)
codon_dnds = codondnds(dout_cancergenes, RefCDS_codon,
theta_option="conservative", min_recurr=2)
codon_dnds$recurcodons[which(codon_dnds$recurcodons$qval<0.1),]
```

The output should look something like this:

```
codon_dnds$recurcodons[which(codon_dnds$recurcodons$qval<0.1),]
```

```
  chr          gene codon freq           mu      dnds         pval
qval
1    4         FGFR3  S249   13 0.0026714453 4866.2797 2.621846e-25
1.376535e-19
2   17          TP53  R248   16 0.0099400778 1609.6453 7.833417e-22
2.056370e-16
3    3        PIK3CA  E545   12 0.0081868351 1465.7679 7.394593e-18
1.294115e-12
4   12          KRAS   G12    6 0.0014473316 4145.5601 2.651640e-14
3.480443e-09
5   17         ERBB2  S310    7 0.0032911176 2126.9371 6.485301e-14
6.809891e-09
6    3        PIK3CA  E542    8 0.0081206853  985.1385 1.365277e-12
1.194674e-07
7    4         FGFR3  Y375    5 0.0013764724 3632.4740 2.521340e-12
1.891095e-07
8   17          TP53  R280    8 0.0118571608  674.6978 2.381470e-11
1.562914e-06
```

```
9   19         ERCC2  N238    4 0.0011025375 3627.9944 1.384008e-10
8.073762e-06
10  17          TP53  E285    7 0.0124847138  560.6857 5.092478e-10
2.673678e-05
11   4         FBXW7  R505    4 0.0042853070  933.4220 2.947422e-08
1.406791e-03
12   2         SF3B1  E902    4 0.0050917631  785.5825 5.773892e-08
2.526198e-03
13  17          TP53  A159    4 0.0054963967  727.7495 7.772309e-08
3.138967e-03
14  12         ERBB3  V104    3 0.0042258838  709.9107 1.292705e-06
4.847876e-02
15   9 CDKN2A.p14arf   A97    2 0.0008437944 2370.2457 2.705783e-06
9.470692e-02
```

> `sitednds` looks for selection (mutation recurrence over random expectations) at specific DNA positions, while `codondnds` looks for selection at codons. Each method may be more sensitive for different kinds of hotspots, hence we recommend trying both.

Predicting drivers in a given donor using the Cancer Genome Interpreter

We will use the Cancer Genome Interpreter to predict drivers in one of our donors.

To make it more interesting, each one can select one donor randomly:

```
random_donor = sample(unique(muts$sampleID),1)
muts_in_random_donor = muts[which(muts$sampleID == random_donor),
c("chr","pos","ref","mut")]
cat(random_donor, " donor has ",nrow(muts_in_random_donor), "
mutations\n",sep="")
```

```
write.table(muts_in_random_donor, file=paste(random_donor,".tsv",sep=""),
col.names=F, row.names=F, quote=F )
```

Copy those mutations and paste them here: https://www.cancergenomeinterpreter.org/analysis

Select hg19 as "Reference genome" and click "Run". The analysis will take a few minutes.

You can also explore bladder cancer at Intogen: https://www.intogen.org/search There you would find 78 drivers defined for bladder cancer

Other useful tips

Reference Genomes

By default dndscv uses the GRCh37/hg19 reference genome assembly. If you need to use a different assembly or a different species you can create a new reference database (RefCDS object). A tutorial to do so can be found here: http://htmlpreview.github.io/?http://github.com/im3sanger/dndscv/blob/master/vignettes/buildref.html

Pre-made reference databases for other popular assemblies such as the GRCh38 are also available here: https://github.com/im3sanger/dndscv_data/tree/master/data

Troubleshooting

You can identify if your data is noisy (variant calling problems) by inspecting your dndscv output. If you see a very large excess of synonymous mutations (compare observed number of synonymous mutations against the expected) it can be a sign or presence of artefacts or contamination in your data.