Interpreting *de novo* variation in human disease using **denovolyzeR**

Authors: James S. Ware1-4, Kaitlin E. Samocha1-3, Mark J. Daly1-3

Title ideas:  
A statistical aproach to assess the frequency of *de novo* variation  
Interpreting *de novo* variation in human disease using **denovolyzeR**  
Interpreting *de novo* variation in human disease: a model-based approach

Contact information  
1 Department of Genetics, Harvard Medical School, Boston MA  
2 Broad Institute of MIT and Harvard, Cambridge MA  
3 Analytical and Translational Genetics Unit, Massachusetts General Hospital and Harvard Medical School, Boston MA  
4 NIHR Cardiovascular Biomedical Research Unit at Royal Brompton Hospital and Imperial College London, London UK

### This will be removed from final manuscript, but prevents permutations from re-running every time manuscript is updated during draft phase...

library(knitr)

## Warning: package 'knitr' was built under R version 3.1.2

opts\_chunk$set(cache=T)

## ABSTRACT

### Keywords:

## INTRODUCTION

Spontaneously arising *de novo* genetic variants are important in human disease{ref}. Each of us carries approximately 100 variants that have not been inherited from our parents{ref}, but rather have arisen via mutational events in the parental germline or early embryo, with a median of 1 *de novo* variant affecting the protein-coding genome{ref}.

Exome sequencing and analysis of *de novo* mutations has successfully identified genes underlying rare and genetically homogeneous Mendelian diseases. In Kabuki sybdrome, *de novo* mutations were identified in *KMT2D* (*MLL2*) in 9 out of 10 unrelated individuals{ref}, an accumulation that would be extremely improbable in the absence of a causal role in the disease, given the *rarity* and independence of *de novo* mutations.

By contrast, it is more challenging to dissect the role of *de novo* mutations in conditions with high levels of locus heterogeneity, including heritable complex traits and some Mendelian conditions, where *de novo* mutations are spread across multiple genes, and may play a smaller role in pathogenesis. Here it may be possible to assess the global contribution of *de novo* coding variants to disease by comparing their frequency in cases and controls, given sufficiently large sample sizes. However, at the level of individual genes, the interpretation of *de novo* mutations is complicated by the background mutation rate, which varies greatly between genes. As more individuals are sequenced, it is inevitable that multiple *de novo* mutations will be observed in some genes by chance.

A statistical framework has recently been developed to address these challenges{ref}, that assesses *de novo* single nucleotide variants in coding sequence. Briefly, the mutability of each gene is individually assessed based on local sequence context, and the probability that a *de novo* event will arise in a single copy of the gene in one generation is calculated. The consequence of each possible *de novo* SNP is computed, and *de novo* probabilities are tabulated for each variant class. For a given study population, *de novo* variants can be evaluated by comparing the observed numbers of variants with the number expected based on this model and the population size, using a poisson framework. This permits the robust evaluation of *de novo* variation in individual genes, and increases the power of genome-wide analyses.

In this unit, we describe the application of this statistical framework to analyze *de novo* variants using denovolyzeR {url?}, an open-source software package written for the R statistical software environment{ref}. We present protocols for four analyses, to assess (i) whether there is a genome-wide excess of *de novo* variation for different functional classes of variant, (ii) whether there is a genome-wide excess of genes with multiple *de novo* mutations (iii) whether individual genes carry an excess of *de novo* variants (iv) whether a pre-specified set of genes collectively shows an enrichment of *de novo* variants.

# BASIC PROTOCOL 1

## Assessing the genome-wide burden of *de novo* variants

This protocol will assess whether there is a genome-wide excess of *de novo* variation for different functional classes of variant

### Materials

A computer running the R software environment, available for UNIX platforms, Windows and MacOS from <http://www.r-project.org>.  
The denovolyzeR package - download and installation options are described at <http://denovolyzer.org>  
dplyr and reshape packages. These dependencies may be installed automatically when denovolyzeR is installed (depending on your installation route). Otherwise they can be installed by running

install.packages("dplyr","reshape")

A table of de novo variants. The minimum input comprises two columns of data: gene names, and variant classes (functional consequence of each variant).  
Example data is included in the denovolyzeR package, and will be used in this protocol. The dataset comprises a data.frame of *de novo* variants identified in 1078 individuals with autism{ref}, named autismDeNovos. It is assumed that readers are able to import their own data into the R environment, using the read.table function or equivalent (in R, ?read.table will provide help).

1. In R, load the denovolyzeR package

library(denovolyzeR)

1. (optional) view the demonstration data provided with the denovolyzeR package. Alternatively, users may import their own data in an equivalent format.

dim(autismDeNovos); head(autismDeNovos)

## [1] 1096 2

## gene dnmClass  
## 1 NOC2L mis  
## 2 ADPRHL2 mis  
## 3 SGIP1 mis  
## 4 HMCN1 syn  
## 5 KCNT2 mis  
## 6 MARK1 syn

#### [ insert list of acceptable classes here]

1. Evaluate the burden

The denovolyzeByClass function will perform the required analysis. The function has three required arguments:

* dnm.genes: a vector of names of genes that contain *de novo* mutations
* dnm.classes: a vector of variant consequences (corresponding to the gene list)
* nsamples: the total number of samples analysed (including samples without *de novo* variants). For the example data, 1078 individuals were sequenced.

denovolyzeByClass(dnm.genes=autismDeNovos$gene,  
 dnm.class=autismDeNovos$dnmClass,  
 nsamples=1078)

## Warning: joining factors with different levels, coercing to character  
## vector

## class observed expected enrichment p.value  
## 1 syn 251 295.8 0.848 9.97e-01  
## 2 mis 652 668.6 0.975 7.45e-01  
## 3 lof 140 92.4 1.520 2.45e-06  
## 4 prot 792 761.0 1.040 1.35e-01  
## 5 all 1043 1056.9 0.987 6.70e-01

The output can be modified to display only a subset of variant classes of interest

denovolyzeByClass(dnm.genes=autismDeNovos$gene,  
 dnm.class=autismDeNovos$dnmClass,  
 nsamples=1078,  
 include.class=c("mis","lof","all"))

## Warning: joining factors with different levels, coercing to character  
## vector

## class observed expected enrichment p.value  
## 1 mis 652 668.6 0.975 7.45e-01  
## 2 lof 140 92.4 1.520 2.45e-06  
## 3 all 1043 1056.9 0.987 6.70e-01

There are a number of additional optional arguments. Information on these, and help generally, is available using the help function

help(denovolyzeByClass)

# BASIC PROTOCOL 2

## Assessing the number of genes with multiple *de novo* mutations

The occurence of multiple *de novo* events in a single gene, in a cohort of individuals with a common phenotype, may implicate that gene in the pathogenesis of the condition under study. Before evaluating single genes, it is instructive to assess the total number of genes harboring multiple *de novo* mutations. Here, the number of genes containing multiple de novos is compared with an empirical distribution derived by permutation.

### Materials

As for protocol 1

1. Ensure the denovolyzeR library and data for analysis are loaded, as before

library(denovolyzeR)

1. The denovolyzeMultiHits function will perform the required analysis.  
   The same three arguments are required as for BASIC PROTOCOL 1: dnm.genes (vector of names of genes containing *de novo* mutations), dnm.classes (a vector of variant consequences) and nsamples (number of samples). In addition, nperms determinines the number of permutations run (defaults to 100)

denovolyzeMultiHits(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078,  
 nperms=100)

## ObsGenes AvgExpGenes MaxExpGenes Empirical.P TotalObsDNM  
## syn 4 3.91 11 0.58 269  
## mis 33 22.43 35 0.01 684  
## lof 6 1.13 7 0.01 143  
## prot 48 30.80 43 0.00 827  
## all 74 52.95 71 0.00 1096

For each variant class, the function returns the observed number of variants containing multiple *de novos* in the user data provided, the average number of genes containing multiple hits across nperms permutations, the maximum number of genes containing multiple hits in any permutation, and an empirical p value. In this case the empirical p value is returned as 0, indicating (in this case <0.01). We can obtain a better estimate by increasing the number of permutations.

denovolyzeMultiHits(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078,  
 nperms=5000,  
 include.class="prot")

## ObsGenes AvgExpGenes MaxExpGenes Empirical.P TotalObsDNM  
## prot 48 31.7118 55 0.002 827

Finally it reports that total number of *de novo* of a given class, which is the number used as input to the permutation.

1. This highlights another important option. The expected number of genes containing >1 hit is obtained by permutation: given n de novo variants, how many genes contain >1 de novo? There are two options for selecting n: by default it is derived from your data: e.g. in the example above autismDeNovos contains 143 lof variants, so this is the number used in the permutation:

sum(autismDeNovos$dnmClass %in% c("frameshift","non","splice"))

## [1] 143

denovolyzeMultiHits(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078,  
 include.class="lof")

## ObsGenes AvgExpGenes MaxExpGenes Empirical.P TotalObsDNM  
## lof 6 0.95 5 0 143

This behaviour is controlled by the expectedDNMs arguement, whose default value is "actual". This is a conservative approach, addressing the question: “given the number of variants in our dataset, do we see more genes with >1 variant than expected?” An alternative approach simply asks whether there are more genes with >1 variant than our de novo model predicts. This is accessed by setting expectedDNMs="expected".

denovolyzeMultiHits(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078,  
 include.class="lof",  
 expectedDNMs="expected")

## ObsGenes AvgExpGenes MaxExpGenes Empirical.P TotalObsDNM  
## lof 6 0.48 2 0 92.40006

# BASIC PROTOCOL 3

## Assessing the frequency of *de novo* mutations in individual genes

In the previous protocol, we assessed whether there were more genes containing multiple *de novo* variants than expected by chance. In the example data we noted 6 genes with multiple loss-of-function hits. In this next protocol, we will determine whether any individual genes carry an excess of *de novo* variants, using the denovolyzeByGene function.

### Materials

As for protocol 1

1. Ensure the denovolyzeR library and data for analysis are loaded, as before

library(denovolyzeR)

1. Call the denovolyzeByGene function.The same three arguments are required as for the previous protocols: dnm.genes (vector of names of genes containing *de novo* mutations), dnm.classes (a vector of variant consequences) and nsamples (number of samples). This function will return one row per gene, ordered according the significance of any enrichment in *de novos*. Given the size of the data, we will only view the first few lines here, using the head function

### [NB - change function output to only return results for genes with at least 1 de novo?]

head(  
denovolyzeByGene(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078)  
)

## lof.observed lof.expected lof.p.value prot.observed prot.expected  
## DYRK1A 3 0 6.15e-08 3 0.1  
## SCN2A 3 0 9.20e-07 5 0.1  
## CHD8 3 0 1.76e-06 4 0.2  
## KATNAL2 2 0 1.19e-05 2 0.0  
## SUV420H1 1 0 7.97e-03 3 0.1  
## POGZ 2 0 8.93e-05 2 0.1  
## prot.p.value  
## DYRK1A 2.87e-05  
## SCN2A 3.15e-07  
## CHD8 3.20e-05  
## KATNAL2 6.08e-04  
## SUV420H1 3.48e-05  
## POGZ 4.82e-03

The p-values returned are not corrected for multiple testing. These default options apply two tests across 18,271 genes, so a Bonferroni corrected p-value threshold at α = 0.05 would be 1.4 × 10-6.

By default this function compares the number of LoF variants against expectation for each gene, and then the total number of protein-altering variants (LoF + missense). It can also be configured to return other classes if relevant, using the include.class argument.

head(  
denovolyzeByGene(dnm.genes=autismDeNovos$gene,  
 dnm.classes=autismDeNovos$dnmClass,  
 nsamples=1078,  
 include.class="syn")  
)

## syn.observed syn.expected syn.p.value  
## PBLD 2 0 3.01e-05  
## TRAPPC8 2 0 4.46e-04  
## ADNP2 2 0 4.80e-04  
## SPRR2D 1 0 1.67e-03  
## PTMS 1 0 2.35e-03  
## C1ORF146 1 0 2.82e-03

# BASIC PROTOCOL 4

## Assessing a pre-specified geneset

This protocol assesses whether a pre-specified set of genes collectively shows an enrichment of *de novo* variants. Note that any of the previous analyses can be restricted to a pre-specified geneset in the same way, using the include.gene argument. This may be appropriate if a smaller panel of genes have been sequenced (rather than whole exome sequencing), or to explore biologically relevant genesets.

### Materials

As for protocol 1

1. In R, load the denovolyzeR package

library(denovolyzeR)

1. Define a geneset. This should be a vector of genes, which may be entered by hand, or read from file using read.table or equivalent. In this example, we use an example geneset included with the denovolyzeR package, a list of 842 genes that interact with the fragile X mental retardation protein (FMRP).

length(FMRPgenes);head(FMRPgenes)

## [1] 842

## [1] "BSN" "KIF1A" "MAP1A" "APC" "MAP1B" "AGAP2"

1. Evaluate the frequency of *de novo* events in our pre-specified genelist, using the denovolyzeByClass function. Specify the genelist using the include.gene argument, which defaults to "all", but accepts a vector of genes.

denovolyzeByClass(dnm.genes=autismDeNovos$gene,  
 dnm.class=autismDeNovos$dnmClass,  
 nsamples=1078,  
 include.gene=FMRPgenes)

## Warning: joining factors with different levels, coercing to character  
## vector

## class observed expected enrichment p.value  
## 1 syn 30 33.0 0.909 7.22e-01  
## 2 mis 80 73.7 1.090 2.46e-01  
## 3 lof 34 9.5 3.560 6.80e-10  
## 4 prot 114 83.2 1.370 7.90e-04  
## 5 all 144 116.2 1.240 7.04e-03

In this example we see a highly significant enrichment of *de novo* lof variants in genes that interact with FMRP in our cohort of autism cases.

# SUPPORT PROTOCOL 1

## Getting help

Help on any of the functions described is available using the standard R help functions, e.g. help(denovolyze) or ?denovolyze. Additional details are also available in the package vignette, accessed using vignette("denovolyzeR\_intro").

## COMMENTARY

#### Background Information

We need to put something here. Could put some more info on the model here, with maybe a figure? Or the underlying poisson stats?

#### Critical Parameters

expectedDNMs choice of probability table

#### Troubleshooting

Leave blank

#### Anticipated Results

Leave blank

#### Time Considerations

These analyses are not especially computationally intensive, and will run on a desktop of laptop computer in seconds. The denovolyzeMultiHits function uses permutation, and computation time increases linearly with the number of permutations. Elapsed times (in seconds) to run the three principal functions on *de novo* variants from 1078 samples, using defaut settings, on a MacBook Air (1.7GHz i7, 8Gb RAM) are as follows:

system.time(denovolyzeByClass(dnm.genes=autismDeNovos$gene,dnm.class=autismDeNovos$dnmClass,nsamples=1078))["elapsed"]

## Warning: joining factors with different levels, coercing to character  
## vector

## elapsed   
## 0.486

system.time(denovolyzeMultiHits(dnm.genes=autismDeNovos$gene,dnm.class=autismDeNovos$dnmClass,nsamples=1078,nperms=1000))["elapsed"]

## elapsed   
## 5.248

system.time(denovolyzeByGene(dnm.genes=autismDeNovos$gene,dnm.class=autismDeNovos$dnmClass,nsamples=1078))["elapsed"]

## elapsed   
## 3.449

## ACKNOWLEDGEMENT

JSW is supported by the NIHR Biomedical Research Unit in Cardiovascular Disease at Royal Brompton & Harefield NHS Foundation Trust and Imperial College London, Academy of Medical Sciences, Wellcome Trust, British Heart Foundation Arthritis Research UK, Fondation Leducq, and an HICF award from Wellcome Trust and Department of Health.  
**Acknowledge DM / CES/ JGS?**  
**Any developers / contributors?**

## LITERATURE CITED

## KEY REFERENCE (optional)

Samocha 2014?

## INTERNET RESOURCES (optional)

<http://www.r-project.org/>  
<http://denovolyzer.org/>  
<http://jamesware.github.io/denovolyzeR/>

## FIGURE LEGENDS

## TABLES

na

Additional instructions: The following should be submitted as individual files, NOT as part of the main document: • Figures • COPYRIGHT PERMISSION (if required) • VIDEOS (optional)

If you have any questions about your manuscript, or formatting, or submitting it, consult the Contributor’s Style Guide (For Authors page) or contact the Developmental Editor listed for your title. For questions about our submissions site (ScholarOne Manuscripts), your Contributor Agreement, or copyright permissions, please contact our Editorial Program Coordinator at [cpsubmissions@wiley.com](mailto:cpsubmissions@wiley.com).

# NOTES TO SELF

Will need to define abbreviations (lof etc) somewhere

Should I describe other aspects:

* viewProbabilityTables()
* custom probability tables (need to put some on github if so)

list all arguments as an appendix or support protocol?

* gene.id
* signif.p
* round.expected

Get rid of warning (a bug) "Warning: joining factors with different levels, coercing to character"