# De Novo Damaging DNA Coding Mutations Are Associated With Obsessive-Compulsive Disorder and Overlap With Tourette's Disorder and Autism

## *Supplemental Information*

## SUPPLEMENTAL METHODS

### Sequence alignment and variant calling

Alignment and variant calling of the sequencing reads followed the latest Genome Analysis Toolkit (GATK) (1) Best Practices guidelines. Reads were aligned using BWA-mem (2) to the b37 human reference sequence with decoy sequences. Picard's MarkDuplicates tool was used to mark PCR duplicates (https://broadinstitute.github.io/picard/). In order to minimize downstream effects of differential coverage between two different capture platforms used in this study, a target bed file was created by taking the intersection of the EZExomeV2 and MedExome target regions. GATK was used to realign indels, recalibrate quality scores, and generate GVCF files for each sample using the HaplotypeCaller tool. All samples were called jointly using GATK's GenotypeGVCFs tool, variant score recalibration was applied to the called variants, and all variant call data was written to a VCF file. This pipeline uses GATK's Best Practices parameters and the default parameters for BWA and Picard. Only passing variants were used in downstream analyses. Variants were annotated against the RefSeq hg19 gene definitions and multiple external databases of variant population frequency, conservation scores, variation intolerance, mutation severity, and predicted functional effects using ANNOVAR (3).

### Quality control, de novo and inherited variant calling

Relatedness statistics were calculated based on the method of Manichaikul et al. (4), implemented in VCFtools v0.1.14.10 (5). Trios were omitted if expected family relationships were not confirmed or if there were unexpected relationships within or between families. Trios were omitted if > 5 de novo variants were observed. PLINK/SEQ (6) (i-stats; https://psychgen.u.hpc.mssm.edu/plinkseq/stats.shtml), PicardTools, and GATK DepthOfCoverage tools were used to generate quality metrics (Table S1). To identify outliers that might confound our case-control analysis, we performed principal components analysis (PCA) using this data. A scree plot determined the number of principal components accounting for the greatest proportion of variance, and we removed trios with family members falling more than three standard deviations from the mean in any of these principal components. See Figure S1, Table S1, and below.

We used stringent thresholds for identifying de novo mutations because DNA from control subjects (from the Simons Simplex Collection) was not available for confirmation by Sanger sequencing. De novo variants were called using an in-house script that required the following: child is heterozygous for a variant with alternate allele frequency between 0.3 and 0.7 in the child and < 0.05 in the parents, sequencing depth (DP) ≥ 20 in all family members at the variant position, alternate allele depth (AD) ≥ 5, observed allele frequency (AC) < 0.01 (1%) among all cases and controls, mapping quality (MQ) ≥ 30. False positive calls were removed by in silico visualization. We performed Sanger sequencing on the probands and parents for 149 putative de novo variants from this project; 147 were confirmed, resulting in 98.7% specificity. Confirmation status for each de novo variant is listed in Table S2.

Inherited variant calling required a variant alternate allele frequency of at least 0.3 in the child, sequencing depth (DP) ≥ 20 in all family members at the variant position, alternate allele depth (AD) ≥ 5, observed allele frequency (AC) < 0.01 (1%) among all cases and controls, and mapping quality (MQ) ≥ 30.

### Principal component analysis (PCA)

PCA was performed on all sequencing quality metrics (Table S1) in R using the following code:

```
library(xlsx)
library("FactoMineR")
library("factoextra")
library("corrplot")

## Load Data from Table S1, first tab
data1 <- read.delim("Table_S1.xlsx")
# select only certain columns
data1.temp <- data1[c(2,14:45)]
# make first column the row names
data1.active <- data.frame(data1.temp[,-1], row.names=data1.temp[,1])

## Load additional data for later use (non-numeric labels/groups)
# select only certain columns
data2.temp <- data1[c(4,3,1)]
# make first column the row names
data2.active <- data.frame(data2.temp[,-1], row.names=data2.temp[,1])

## Principal component analysis
pdf("PCA_factor_maps.pdf")
res.pca <- PCA(data1.active, scale.unit = TRUE, ncp = 10, graph = TRUE, axes = c(1,2))
dev.off()

print(res.pca)

## Export PCA coordinates to determine outliers (Table S1)
indcoord<-res.pca$ind$coord
```

```
write.xlsx(indcoord, "Table_S1.xlsx")

## Estimate the number of components in Principal Component Analysis (Factominer)
sink("EstmateNumberPCs.txt")
estim_ncp(data1.active, ncp.min=0, ncp.max=NULL, scale=TRUE, method="Smooth")
sink()

## Variances of the principal components
eigenvalues <- res.pca$eig

## Make scree plot using base graphics : A scree plot is a graph of the
eigenvalues/variances associated with components (Figure S3.A).
pdf("ScreePlot.pdf")
barplot(eigenvalues[, 2], names.arg=1:nrow(eigenvalues),
       main = "Variances",
       xlab = "Principal Components",
       ylab = "Percentage of Variance",
       col ="steelblue")
lines(x = 1:nrow(eigenvalues), eigenvalues[, 2],
      type="b", pch=19, col = "red")
dev.off()

## Make cumulative variance graph (Figure S3.B)
pdf("ScreePlot_cumulative.pdf")
barplot(eigenvalues[, 3], names.arg=1:nrow(eigenvalues),
        main = "Variances",
        xlab = "Principal Components",
        ylab = "Cumulative Percentage of Variance",
        col ="steelblue")
lines(x = 1:nrow(eigenvalues), eigenvalues[, 3],
      type="b", pch=19, col = "red")
dev.off()

## GRAPHS OF VARIABLES
pdf("PCA_factor_maps_variables.pdf")
fviz_pca_var(res.pca, col.var="contrib") + scale_color_gradient2(low="white", mid="blue",
high="red", midpoint=55)+theme_bw()
dev.off()

## GRAPHS OF INDIVIDUALS (Figure S3.C)
pca = prcomp(data1.active, scale = TRUE)
pdf("PCA_prcomp_factor_map_indiv.pdf")
plot(pca$x, pch = 20, col = c(rep("red", 366), rep("blue", 1200)))
dev.off()
```

Minimizing variant calling bias

Exome capture was performed at the Yale Center for Genomic Analysis (YCGA) using the NimbleGen SeqCap EZExomeV2 (109 trios) or MedExome (113 trios) capture libraries (Roche NimbleGen, Madison, WI, USA). The EZExomeV2 platform captures 44 Mb of DNA, covering 97.4% of RefSeq transcripts. The MedExome platform captures 47 Mb of DNA, covering 99.8% of RefSeq transcripts. Furthermore, MedExome has improved read depth distribution and increased coverage with less sequencing, For example, MedExome at 3.9 GB of sequence has similar sensitivity (98.1%) and specificity (99.6%) for heterozygous calls as

EZExomeV2 at 5.1 GB of sequence (http://cnpg.comparenetworks.com/179181-Evaluation-of-Roche-NimbleGen-exome-capture-products/). To minimize potential variant calling bias that may occur due to differential coverage between capture platforms, we aligned our sequencing to a customized target bed file that is the intersection between regions captured by both platforms (see Methods); furthermore, our primary comparison between OCD and controls uses mutation rates within the "callable" exome (see Methods and below).

Mutation rate analysis

Within each cohort, we calculated the rates of de novo and inherited mutations per base pair. For accurate rate calculation, we first determined the number of "callable" base pairs per family using the GATK DepthOfCoverage tool. We considered only bases covered at ≥ 20x in all family members, with base quality ≥ 20, and map quality ≥ 30; these thresholds match those required for GATK and de novo variant calling. We used mutation rates within the callable exome as our primary comparison to further minimize any potential bias for differential variant calling between the two cohorts. For each cohort, we summed the "callable" base pairs in every family and used this number as the denominator for de novo rate calculations. The resulting rate was divided by two to give haploid rates. Confidence intervals were calculated using the *pois.conf.int (pois.exact)* function from the epitools v0.5-9 package in R. We compared de novo mutation rates in cases versus controls (burden analysis) using a one-tailed rate ratio test in R (https://cran.r-project.org/package=rateratio.test), considering only those variants present with a frequency of <0.01 in the ExAC v0.3.1 database (7). We compared inherited mutation rates in a similar manner but considered only those variants seen once across all cases and controls, and not reported in ExAC.

Calculating "callable" base pairs

We calculated the rates of de novo and inherited mutations per base pair within each cohort. For accurate rate calculation, we first determined the number of "callable" base pairs per family using the GATK DepthOfCoverage tool. We considered only bases covered at ≥ 20x in all family members, with base quality ≥ 20, and map quality ≥ 30; these thresholds match those required for GATK and de novo variant calling. The following command was used to calculate the callable base pairs in each trio:

```
java -jar GenomeAnalysisTK.jar -T DepthOfCoverage -R human_g1k_v37.fasta -o FamilyID -I
FamilyID.list -L target_intersection.bed --minMappingQuality 30 --minBaseQuality 20 --
summaryCoverageThreshold 20
```

`FamilyID.list` contains names and locations of the three trio bam files. The .bed file contains the genomic intervals over which to calculate the callable base pairs. To calculate the coding callable base pairs (used for coding mutation rates, e.g. synonymous, nonsynonymous, missense, etc., see Table 1, Table S2), we used a bed file with intervals spanning the intersection of both capture array target intervals and the RefSeq coding intervals (32,027,823 bp total). To calculate all callable base pairs (used for the total coding + noncoding mutation rate, see “All” in Table 1), we used a bed file with intervals spanning the intersection of both capture array target intervals (33,973,867 bp total). The number of coding and total callable base pairs for every family passing QC is listed in Table S1.

Contribution of de novo mutations to OCD risk

For every proband and control subject passing QC (obtained from Table S1), we made a file containing the number of Mis-D, LGD, and damaging (Mis-D + LGD) variants (obtained from Table S2), then calculated the haploid mutation rate per subject for each variant type. Haploid mutation rates were calculated by dividing the number of mutations by the twice the number of callable coding bases (“CallableExomeCoding” in Table S1).

The following R code was then used to calculate the percentage of cases with a mutation mediating risk and the percentage of mutations carrying risk, along with 95% confidence intervals for each. The number of de novo mutations per individual was calculated by multiplying the mutation rate by the size of the RefSeq hg19 coding exome (33,828,798 bp).

```
library(magrittr)
library(dplyr)

#### Load files containing haploid mutation rates
caseVarFile="OCD_rates.txt"
caseVarData <- read.table(caseVarFile, sep="\t", header=T)

ctrlVarFile="SSC_rates.txt"
ctrlVarData <- read.table(ctrlVarFile, sep="\t", header=T)

#### Calculations for Mis-D variants
MisD_contrib <- t.test(caseVarData$haploidRatesMisD, ctrlVarData$haploidRatesMisD, paired
= F)
```

```
df_MisD_contrib <- data.frame(case.rate = MisD_contrib$estimate[1] * 33828798 * 2,
                   ctrl.rate = MisD_contrib$estimate[2] * 33828798 * 2,
                   diff.lower.ci = MisD_contrib$conf.int[1] * 33828798 * 2,
                   diff.upper.ci = MisD_contrib$conf.int[2] * 33828798 * 2)
df_MisD_contrib <- df_MisD_contrib %>% mutate(diff = case.rate - ctrl.rate, percent =
diff / case.rate,
                   percent.lower.ci = diff.lower.ci / case.rate,
                   percent.upper.ci = diff.upper.ci / case.rate)

#### Calculations for LGD variants
LGD_contrib <- t.test(caseVarData$haploidRatesLGD, ctrlVarData$haploidRatesLGD, paired =
F)
df_LGD_contrib <- data.frame(case.rate = LGD_contrib$estimate[1] * 33828798 * 2,
                   ctrl.rate = LGD_contrib$estimate[2] * 33828798 * 2,
                   diff.lower.ci = LGD_contrib$conf.int[1] * 33828798 * 2,
                   diff.upper.ci = LGD_contrib$conf.int[2] * 33828798 * 2)
df_LGD_contrib <- df_LGD_contrib %>% mutate(diff = case.rate - ctrl.rate, percent = diff
/ case.rate,
               percent.lower.ci = diff.lower.ci / case.rate,
               percent.upper.ci = diff.upper.ci / case.rate)

#### Calculations for Damaging variants
Damaging_contrib <- t.test(caseVarData$haploidRatesDamaging,
ctrlVarData$haploidRatesDamaging, paired = F)
df_Damaging_contrib <- data.frame(case.rate = Damaging_contrib$estimate[1] * 33828798 *
2,
               ctrl.rate = Damaging_contrib$estimate[2] * 33828798 * 2,
               diff.lower.ci = Damaging_contrib$conf.int[1] * 33828798 * 2,
               diff.upper.ci = Damaging_contrib$conf.int[2] * 33828798 * 2)
df_Damaging_contrib <- df_Damaging_contrib %>% mutate(diff = case.rate - ctrl.rate,
percent = diff / case.rate,
               percent.lower.ci = diff.lower.ci / case.rate,
               percent.upper.ci = diff.upper.ci / case.rate)
```

Mutation rates for variant simulations

To perform subsequent maximum likelihood estimation and TADA analyses, we used published per gene de novo mutation rates from unaffected parent-child trios (8). For the control samples in our dataset, we calculated the proportion of the overall coding mutation rate that comprised LGD and Mis-D mutations, and then used these proportions to calculate the expected LGD and Mis-D mutation rate per gene.

The following R code was used to generate the mutation rate tables:

```
library(denovolyzeR)
library(plyr)

########################
#### Mutation type fractions from controls in OCD project
########################

# fractions of overall coding mutation rate for each variant type in SSC controls (see
Table 1)
```

```
fracLGD <- 0.0611
fracMisD <- 0.2944

########################
#### Get published de novo mutation rtaes
########################

denovolyzer <- viewProbabilityTable()
mutationProbs <- denovolyzer[ , c("geneName", "all")]
mutationProbs <- rename(mutationProbs, c("geneName"="gene.name"))  #rename column
mutationProbs <- rename(mutationProbs, c("all"="mut.rate"))  #rename column
#save(mutationProbs, file = "denovolyzer_rates_all_unadjusted.RData")
#write.table(mutationProbs, "denovolyzer_rates_all_unadjusted.txt", sep="\t")

########################
#### Add LGD and Mis-D de novo mutation rates based on fractions seen in our study
########################

mutationProbs$lgd <- mutationProbs$mut.rate * fracLGD
mutationProbs$misD <- mutationProbs$mut.rate * fracMisD

save(mutationProbs, file = "de_novo_mutation_rates.RData")
write.table(mutationProbs, "de_novo_mutation_rates.RData", sep="\t")
```

TADA analysis

The enrichment of de novo LGD and Mis-D mutations in OCD raises the possibility that these classes of mutations target a set of genes that mediates OCD risk. We used the transmitted and de novo association (TADA) test to test this hypothesis. TADA is based on a Bayesian model that can combine data from families (de novo and inherited variants) with variants from case-control cohorts to significantly increase the power of gene discovery. In this study, we ran two versions of the TADA test, one that included both de novo and inherited variants (TADA), and one that included only de novo variants (TADA-Denovo). The code and documentation for this tool can be found here (TADA.v1.2.R; http://wpicr.wpic.pitt.edu/WPICCompGen/TADA/TADA_homepage.htm).

TADA-Denovo analyzes two types of de novo variants, LGD and severe missense (“Mis-D”; Polyphen2 HDIV score ≥0.957). Input for this test includes (a) the number of de novo LGD variants per gene, (b) the number of de novo Mis-D variants per gene, and (c) mutation rates for all human genes. We used mutation rates from Sanders et al. (9). TADA-Denovo analyzes LGD and Mis-D occurrences separately, then combines the evidence in a Bayesian fashion, weighing each type of mutation differently.

TADA-Denovo computes the Bayes factors and p-values of all input genes using the following parameters:

- n.family: the number of OCD parent-child trios passing quality control (184)

- The fold-enrichment (λ) for each mutational class is calculated as follows, where $X_{case}$ and $X_{ctrl}$ are the number of mutations in cases and controls, respectively; $S_{case}$ and $S_{ctrl}$ are the number of synonymous mutations in case and controls, respectively: $\lambda = (X_{case} / (X_{ctrl} \cdot (S_{case}/S_{ctrl})))$. As shown in the code below, we used the estimated coding variants per individual (Table 1) and multiplied by the number of cases or controls. Fold-enrichment for LGD was calculated as 2.09, and fold-enrichment for Mis-D was 1.46.

- The fraction of causal genes (π) is the estimated number of OCD risk genes (335, calculated by MLE as detailed below) divided by the number of RefSeq genes with mutational rates included in the TADA-Denovo algorithm. $\pi = 335 / 19618 = 0.0171$

- gamma.mean.dn: The average relative risk (γ) is related to the fold-enrichment (λ) and the fraction of causal genes (π) by the following equation: $\pi \cdot (\gamma\text{-}1) = \lambda\text{-}1$. Solving for γ gives LGD $\gamma = 64.74$, Mis-D $\gamma = 27.91$).

To calculate the p-value for each gene, TADA-Denovo uses each gene's specified mutation rate to simulate random mutation data and obtain a null distribution of Bayes factors. We used 1,000 samplings of de novo mutations in each gene to determine null distributions. Finally, an FDR q-value is calculated for each gene using a Bayesian "direct posterior approach." Lower q-values indicate stronger evidence for OCD association. Genes with FDR < 0.3 are considered probable risk genes, and those with FDR < 0.1 are high-confidence risk genes.

The following R code performed the TADA-Denovo analysis:

```r
source("TADA.v1.2.R")

# set.seed(100)

### read mutation rates and counts (see Table S3, second tab)
tada.file="Table_S5.txt"
tada.data=read.table(tada.file,header=T)

### Number of mutations and TADA parameters

numLgdMutations <- (0.15*184)
lgdRate <- numLgdMutations/184
```

```
numControlLgdMutations <- (0.074*777)
controlLgdRate <- numControlLgdMutations/777
lgdRiskFraction <- (lgdRate - controlLgdRate) / lgdRate

numMis3Mutations <- (0.51*184)
mis3Rate <- numMis3Mutations/184
numControlMis3Mutations <- (0.36*777)
controlMis3Rate <- numControlMis3Mutations/777
mis3RiskFraction <- (mis3Rate - controlMis3Rate) / mis3Rate

numGenes <- 335  # from MLE analysis below

nPerms <- 1000

pi <- numGenes / nrow(tada.data)
pi0 <- 1-pi

numSilentMutations <- (0.33*184)
numControlSilentMutations <- (0.34*777)

dn.lof.lambda <- (numLgdMutations) / (numControlLgdMutations *
(numSilentMutations/numControlSilentMutations))
dn.lof.relativeRisk <- 1 + ((dn.lof.lambda-1) / pi)
dn.mis3.lambda <- (numMis3Mutations) / (numControlMis3Mutations *
(numSilentMutations/numControlSilentMutations)) # num Mis3 in ctrls, num silent in cases,
num silent in ctrls
dn.mis3.relativeRisk <- 1 + ((dn.mis3.lambda-1) / pi)

n.family = 184
n = data.frame(dn=n.family, ca=NA, cn=NA)
sample.counts <- list(cls1=n, cls2=n)

### create the mutational data used by TADA-Denovo
cls1.counts=data.frame(dn=tada.data$dn.cls1, ca=NA, cn=NA)
rownames(cls1.counts)=tada.data$gene.id
cls2.counts=data.frame(dn=tada.data$dn.cls2, ca=NA, cn=NA)
rownames(cls2.counts)=tada.data$gene.id
tada.counts=list(cls1=cls1.counts,cls2=cls2.counts)

### set up mutation rates
mu=data.frame(cls1=tada.data$mut.cls1,cls2=tada.data$mut.cls2)

### specify de novo only analyses
denovo.only=data.frame(cls1=TRUE,cls2=TRUE)

### set up parameters -
cls1= data.frame(gamma.mean.dn.=dn.lof.relativeRisk,beta.dn=1,gamma.mean.CC=NA,beta.CC=NA
,rho1=NA,nu1=NA,rho0=NA,nu0=NA)
cls2=
data.frame(gamma.mean.dn=dn.mis3.relativeRisk,beta.dn=1,gamma.mean.CC=NA,beta.CC=NA,rho1=
NA,nu1=NA,rho0=NA,nu0=NA)
hyperpar=list(cls1=cls1,cls2=cls2)

### running TADA-Denovo
re.TADA <- do.call(cbind.data.frame, TADA(tada.counts=tada.counts,
sample.counts=sample.counts, mu=mu, hyperpar=hyperpar, denovo.only=denovo.only))

### Bayesian FDR control
re.TADA$qval=Bayesian.FDR(re.TADA$BF.total, pi0 = pi0)
```

```
### run permutation to get the null distributions to use for calculating p-values for
TADA
re.TADA.null=do.call(cbind.data.frame, TADAnull(tada.counts=tada.counts,
sample.counts=sample.counts, mu=mu, hyperpar=hyperpar, denovo.only=denovo.only,
nrep=nPerms))
re.TADA$pval=bayesFactor.pvalue(re.TADA$BF.total,re.TADA.null$BFnull.total)

### display top 10 genes based on BF.total
re.TADA[order(-re.TADA$BF.total)[1:10],]

### write all table to file - See Table S5
write.table(re.TADA, "TADA_denovo_Results_.txt", sep="\t")
save.image(file="TADA_denovo_Workspace.RData")
```

The following R code performed the TADA (Denovo + Inherited) analysis:

```
source("TADA.v.1.2.R")

#set.seed(100)

### read mutation rates and counts (see Table S5, fourth tab)
tada.file="TableS5.txt"
tada.data=read.table(tada.file,header=T)

### Number of mutations and TADA parameters

numLgdMutations <- (0.15*184)
lgdRate <- numLgdMutations/184
numControlLgdMutations <- (0.074*777)
controlLgdRate <- numControlLgdMutations/777
lgdRiskFraction <- (lgdRate - controlLgdRate) / lgdRate

numMis3Mutations <- (0.51*184)
mis3Rate <- numMis3Mutations/184
numControlMis3Mutations <- (0.36*777)
controlMis3Rate <- numControlMis3Mutations/777
mis3RiskFraction <- (mis3Rate - controlMis3Rate) / mis3Rate

numInherLgdMutations <- (3.79*184)
InherLgdRate <- numInherLgdMutations/184
numInherControlLgdMutations <- (3.59*777)
InherControlLgdRate <- numInherControlLgdMutations/777
InherLgdRiskFraction <- (InherLgdRate - InherControlLgdRate) / InherLgdRate

numInherMis3Mutations <- (15.49*184)
InherMis3Rate <- numInherMis3Mutations/184
numInherControlMis3Mutations <- (15.83*777)
InherControlMis3Rate <- numInherControlMis3Mutations/777
InherMis3RiskFraction <- (InherMis3Rate - InherControlMis3Rate) / InherMis3Rate

numGenes <- 335  # from MLE analysis below

nPerms <- 100

pi <- numGenes / nrow(tada.data)
pi0 <- 1-pi

numSilentMutations <- (0.33*184)
```

```
numControlSilentMutations <- (0.34*777)
numInherSilentMutations <- (16.44*184)
numInherControlSilentMutations <- (16.78*777)

dn.lof.lambda <- (numLgdMutations) / (numControlLgdMutations *
(numSilentMutations/numControlSilentMutations))
dn.lof.relativeRisk <- 1 + ((dn.lof.lambda-1) / pi)
dn.mis3.lambda <- (numMis3Mutations) / (numControlMis3Mutations *
(numSilentMutations/numControlSilentMutations)) # num Mis3 in ctrls, num silent in cases,
num silent in ctrls
dn.mis3.relativeRisk <- 1 + ((dn.mis3.lambda-1) / pi)

inher.lof.lambda <- (numInherLgdMutations) / (numInherControlLgdMutations *
(numInherSilentMutations/numInherControlSilentMutations))
inher.lof.relativeRisk <- 1 + ((inher.lof.lambda-1) / pi)
inher.mis3.lambda <- (numInherMis3Mutations) / (numInherControlMis3Mutations *
(numInherSilentMutations/numInherControlSilentMutations)) # num Mis3 in ctrls, num silent
in cases, num silent in ctrls
inher.mis3.relativeRisk <- 1 + ((inher.mis3.lambda-1) / pi)

n.family = 184
n.case = 0
n.ctrl = 0
n = data.frame(dn=n.family, ca=n.case+n.family, cn=n.ctrl+n.family)
sample.counts <- list(cls1=n, cls2=n)

### create the mutational data used by TADA
cls1.counts=data.frame(dn=tada.data$dn.cls1, ca=tada.data$trans.cls1+tada.data$case.cls1,
cn=tada.data$ntrans.cls1+tada.data$ctrl.cls1)
rownames(cls1.counts)=tada.data$gene.id
cls2.counts=data.frame(dn=tada.data$dn.cls2, ca=tada.data$trans.cls2+tada.data$case.cls2,
cn=tada.data$ntrans.cls2+tada.data$ctrl.cls2)
rownames(cls2.counts)=tada.data$gene.id
tada.counts=list(cls1=cls1.counts,cls2=cls2.counts)

### set up mutation rates
mu=data.frame(cls1=tada.data$mut.cls1,cls2=tada.data$mut.cls2)

### set up denovo only TRUE/FALSE, here we do not want to restrict ourselves to de novo
only analyses
denovo.only=data.frame(cls1=FALSE,cls2=FALSE)

### set up parameters -
***Seed – yes, perm 1000
cls1=
data.frame(gamma.mean.dn.=dn.lof.relativeRisk,beta.dn=1,gamma.mean.CC=inher.lof.relativeR
isk,beta.CC=10 ,rho1=0.1,nu1=10,rho0=0.1,nu0=10)
cls2=
data.frame(gamma.mean.dn=dn.mis3.relativeRisk,beta.dn=1,gamma.mean.CC=inher.mis3.relative
Risk,beta.CC=10,rho1=0.1,nu1=10,rho0=0.1,nu0=10)
hyperpar=list(cls1=cls1,cls2=cls2)

### running TADA-Denovo
re.TADA <- do.call(cbind.data.frame, TADA(tada.counts=tada.counts,
sample.counts=sample.counts, mu=mu, hyperpar=hyperpar, denovo.only=denovo.only))

### Bayesian FDR control
re.TADA$qval=Bayesian.FDR(re.TADA$BF.total, pi0 = pi0)

### run permutation to get the null distributions to use for calculating p-values for
TADA
```

```
re.TADA.null=do.call(cbind.data.frame, TADAnull(tada.counts=tada.counts,
sample.counts=sample.counts, mu=mu, hyperpar=hyperpar, denovo.only=denovo.only,
nrep=nPerms))
re.TADA$pval=bayesFactor.pvalue(re.TADA$BF.total,re.TADA.null$BFnull.total)

### display top 10 genes based on BF.total
re.TADA[order(-re.TADA$BF.total)[1:10],]

### write all table to file - See Table S5
write.table(re.TADA, "TADA_Results.txt", sep="\t")
save.image(file="TADA_Workspace.RData")
```

Applying this model to 184 OCD parent-child trios passing QC identifies two high confidence OCD risk genes (q<0.1). See Table 2 and Table S5.

## Maximum likelihood estimation (MLE) method for estimating the number of OCD risk genes

We first used a maximum likelihood estimation (MLE) method to estimate the number of genes contributing risk to OCD, based on vulnerability to de novo damaging variants (10). For every number of risk genes from 1 to 2,500, we simulated 95 variants (the number of damaging de novo variants observed in probands in our case-control burden analysis). Variant simulations were performed 100,000 times at each number of risk genes. Following each simulation, a percentage of variants was randomly assigned to the risk genes. The percentage of variants assigned to risk genes was determined by the fraction of de novo damaging variants estimated to carry OCD risk, and variant simulations were weighted by gene size and GC content (11). We then counted the number of risk and non-risk genes containing two variants and the number containing three or more variants. The frequency of concordance between our simulated and observed data was calculated. A curve was plotted to show the concordance frequency (y-axis) at each assumed number of risk genes (x-axis), and the peak was taken as the estimate of the most likely number of risk genes (10). See Figure S2.

The following R code was used to perform these calculations:

```
library(ggplot2)
library(parallel)
library(data.table)

plotDir <- getwd()
load("de_novo_mutation_rates.RData") # see Table S5, first tab
mutationProbs <- as.data.table(mutationProbs)
```

```
K <- 95  # total OCD de novo damaging mutations (Mis-D+LGD)
R2 <- 2 # number of above mutations hitting same gene twice
R3 <- 1 # number of above mutations hitting sane gene three times
M1 <- 95/184  # observed rate of de novo damaging mutations in OCD
M2 <- 236/777 # observed rate of de novo damaging mutations in controls
E <- (M1-M2)/M1 # estimating fraction of de novo damaging variants carrying risk
nPerms <- 100000 # number of permutations to perform at each assumed number of risk genes
maxGenes <- 2500 # perform permutations from 1 to this number

# get number of cores available
numCores <- max(1, detectCores() - 1)

##############################################################################
# FUNCTIONS
##############################################################################
getRecurrence <- function(G, mutationProbs, K, E, R2, R3, nPerms){
    permutationVector <- lapply(1:nPerms, function(x) {
        riskGeneIndex <- sample(1:nrow(mutationProbs), G, replace = F)
        riskGenes <- mutationProbs[riskGeneIndex,]
        nonRiskGenes <- mutationProbs[-riskGeneIndex,]

        C1 <- rbinom(1,K,E)
        C2 <- K-C1

        C1geneMutations <- sample(riskGenes$gene.name, C1, replace = T, prob =
riskGenes$damaging)
        C2geneMutations <- sample(nonRiskGenes$gene.name, C2, replace = T, prob =
nonRiskGenes$damaging)

        allGeneMutations <- c(C1geneMutations, C2geneMutations)
        length(which(table(allGeneMutations)==2))==R2 &
length(which(table(allGeneMutations)>=3))==R3
    })
    proportionMatchingObserved <- length(which(unlist(permutationVector)))/nPerms

}
##############################################################################
# RUN
##############################################################################
RNGkind("L'Ecuyer-CMRG")
set.seed(1)
mc.reset.stream()

permutationTest <- mclapply(1:maxGenes,
                   function(x) getRecurrence(x, mutationProbs, K, E, R2, R3, nPerms),
                   mc.cores = numCores, mc.set.seed = T )

save(permutationTest, file = paste("UpTo", maxGenes, "genes", nPerms, "perms",
"MaxLikelihoodPermutation.RData", sep="_"))

toPlot <- data.frame(likelihood = unlist(permutationTest), nGenes =
1:length(unlist(permutationTest)))

save(toPlot, file = paste("UpTo", maxGenes, "genes", nPerms, "perms",
"MaxLikelihoodPermutationToPlot.RData", sep="_"))

p <- ggplot(toPlot, aes(x=nGenes, y=likelihood))
p <- p + geom_line() + geom_smooth()
ggsave(file.path(plotDir, paste("UpTo", maxGenes, "genes", nPerms, "perms",
"MaxLikelihood.pdf", sep="_") ), p)
```

```
save.image(file = paste("Workspace", nPerms, "perms.RData", sep="_"))
```

## "Unseen species" method for estimating the number of OCD risk genes

Following a method used previously in estimating the number of risk genes for autism spectrum disorder (9), we used the following R code to obtain a second estimate of the number of risk genes (C) in OCD and 95% confidence intervals. This method uses the frequency and number of observed variant types (or species) to infer how many species are present in the population.

The following R code was used to perform these calculations:

```
library(epitools)
# total OCD de novo damaging mutations (Mis-D+LGD)
damaging <- 95
# scale the observed number of damaging mutations in controls to get expected number in
cases
expectedDamaging <- ceiling( (236) * 184/777 )
# calculate the number of risk associated mutations
d <- damaging - expectedDamaging
nRecurrent <-  3
expectedNumRecurrent <- 2 # there were 8 in 777 controls, so (8*184/777)
# total number of observed risk genes
c <- d - (nRecurrent + 2)
# number of genes mutated once
c1 <- c - nRecurrent
# probability that newly added mutation hits a previously mutated gene
u = 1 - c1/d
#Estimate the number of risk genes
( C <-  c/u + 1*d*(1-u)/u )
# 316.875

## calculating 95% confidence interval for number of risk genes
# import tab delimited file with number of damaging mutations per control sample
ctrlDamagingFile="SSC_damaging.txt"
ctrlDamagingData <- read.table(ctrlDamagingFile, sep="\t", header=T)
# calculate upper and lower number of damaging mutations per control sample
ci_pois <- pois.exact(sum(ctrlDamagingData$NumDamaging), pt = nrow(ctrlDamagingData),
conf.level = 0.95)

# repeat above risk gene calculations using lower value - gives upper estimate
expectedDamaging_low <- ceiling((ci_pois$lower*777) * 184/777 )
d_low <- damaging - expectedDamaging_low
c_low <- d_low - (nRecurrent + 2)
c1_low <- c_low - nRecurrent
u_low = 1 - c1_low/d_low
( C_low <-  c_low/u_low + 1*d_low*(1-u_low)/u_low )
# 454.25

# repeat above risk gene calculations using upper value - gives lower estimate
expectedDamaging_hi <- ceiling((ci_pois$upper*777) * 184/777 )
d_hi <- damaging - expectedDamaging_hi
c_hi <- d_hi - (nRecurrent + 2)
```

```
c1_hi <- c_hi - nRecurrent
u_hi = 1 - c1_hi/d_hi
( C_hi <-  c_hi/u_hi + 1*d_hi*(1-u_hi)/u_hi )
# 189.875
```

Predicting the number of risk genes identified by cohort size

Fixing the gene number at 335 (from MLE estimate above), we varied the cohort size (from 25 to 3000, in increments of 25). At each cohort size, we simulated a number of variants matching the observed mutation rate in OCD probands. Simulated variants were randomly assigned to the risk genes. The percentage of variants assigned to risk genes was determined by the fraction of de novo damaging variants estimated to carry OCD risk, and variant simulations were weighted by gene size and GC content (11). At each cohort size, 10,000 simulations were performed. LGD and Mis-D variants were generated separately. Simulated variants were then combined and given as input to the TADA-Denovo algorithm, using the same parameters described above for the observed data. The number of high confidence (q<0.1) and probable (q<0.3) risk genes were recorded and plotted using polynomial regression fitting; this regression model allows prediction of the number of genes identified at a specified cohort size. See Figure S3.

The following R code was used to perform these calculations:

```
library(ggplot2)
library(parallel)
library(reshape2)

plotDir <- getwd()
source(file = "TADA.v1.1.R")

load("de_novo_mutation_rates.RData")

numLgdMutations <- (0.15*184)
lgdRate <- numLgdMutations/184
numControlLgdMutations <- (0.074*777)
controlLgdRate <- numControlLgdMutations/777
lgdRiskFraction <- (lgdRate - controlLgdRate) / lgdRate

numMis3Mutations <- (0.51*184)
mis3Rate <- numMis3Mutations/184
numControlMis3Mutations <- (0.36*777)
controlMis3Rate <- numControlMis3Mutations/777
mis3RiskFraction <- (mis3Rate - controlMis3Rate) / mis3Rate

numGenes <- 335
nPerms <- 10000

# get number of cores available
```

```
numCores <- max(1, detectCores() - 1)

pi <- 0.01707615
pi0 <- 1-pi

numSilentMutations <- (0.33*184)
numControlSilentMutations <- (0.34*777)

dn.lof.lambda <- (numLgdMutations) / (numControlLgdMutations *
(numSilentMutations/numControlSilentMutations))
dn.lof.relativeRisk <- 1 + ((dn.lof.lambda-1) / pi)
dn.mis3.lambda <- (numMis3Mutations) / (numControlMis3Mutations *
(numSilentMutations/numControlSilentMutations)) # num Mis3 in ctrls, num silent in cases,
num silent in ctrls
dn.mis3.relativeRisk <- 1 + ((dn.mis3.lambda-1) / pi)


############################################################################
# FUNCTIONS
############################################################################

getGenes <- function(numGenes_f=numGenes, mutationProbs_f, cohortSize_f, mutationRate_f,
riskFraction_f, probability_f=c("lgd", "mis3")[1]){
        numMutations <- ceiling(cohortSize_f * mutationRate_f)
        riskGeneIndex <- sample(1:nrow(mutationProbs_f), numGenes_f, replace = F)
        riskGenes <- mutationProbs_f[riskGeneIndex,]
        nonRiskGenes <- mutationProbs_f[-riskGeneIndex,]

        C1 <- rbinom(1, numMutations, riskFraction_f)
        C2 <- numMutations - C1

        C1geneMutations <- sample(riskGenes$gene.name, C1, replace = T, prob =
riskGenes$probability)
        C2geneMutations <- sample(nonRiskGenes$gene.name, C2, replace = T, prob =
nonRiskGenes$probability)

        allGeneMutations <- c(C1geneMutations, C2geneMutations)
}

runIteration <- function(numGenes_f=numGenes, mutationProbs_f, cohortSize_f,
lgdMutationRate_f, mis3MutationRate_f, lgdRiskFraction_f, mis3RiskFraction_f, nTadaRep_f
= 100){
    lgdMutations <- getGenes(numGenes, mutationProbs_f, cohortSize_f, lgdMutationRate_f,
lgdRiskFraction_f, "lgd")
    lgdMutations_df <- data.frame(gene=lgdMutations, lof=1, mis3=0, stringsAsFactors = F)
    mis3Mutations <- getGenes(numGenes, mutationProbs_f, cohortSize_f,
mis3MutationRate_f, mis3RiskFraction_f, "lgd")
    mis3Mutations_df <- data.frame(gene=mis3Mutations, lof=0, mis3=1, stringsAsFactors =
F)
    combinedMutations <- rbind(lgdMutations_df, mis3Mutations_df)
    combinedMutations <- aggregate(combinedMutations[,c("lof", "mis3")],
by=list(combinedMutations$gene), sum)
    colnames(combinedMutations) <- c("gene.id", "dn.lof", "dn.mis3")
    tadaResults <- runTada(cohortSize_f = cohortSize_f, mutationTable_f =
combinedMutations, mutationProbs_f = mutationProbs_f, nTadaRep_f = nTadaRep_f)
    return(tadaResults)
}

runTada <- function(cohortSize_f, mutationTable_f, mutationProbs_f, nTadaRep_f = 100){
    tada.data <- merge(mutationTable_f, mutationProbs_f[,c("gene.name", "lgd", "mis3")],
by.x="gene.id", by.y="gene.name")
```

```r
    names(tada.data)[which(names(tada.data)=="lgd")] <- "mut.lof"
    names(tada.data)[which(names(tada.data)=="mis3")] <- "mut.mis3"

    n.family = cohortSize_f
    n = data.frame(dn=n.family, ca=NA, cn=NA)
    sample.counts <- list(cls1=n, cls2=n)

    cls1.counts=data.frame(dn=tada.data$dn.lof, ca=NA, cn=NA)
    rownames(cls1.counts)=tada.data$gene.id
    cls2.counts=data.frame(dn=tada.data$dn.mis3, ca=NA, cn=NA)
    rownames(cls2.counts)=tada.data$gene.id
    tada.counts=list(cls1=cls1.counts,cls2=cls2.counts)

    mu=data.frame(cls1=tada.data$mut.lof,cls2=tada.data$mut.mis3)

    denovo.only=data.frame(cls1=TRUE,cls2=TRUE)

    cls1=
data.frame(gamma.mean.dn=dn.lof.relativeRisk,beta.dn=1,gamma.mean.CC=NA,beta.CC=NA
,rho1=NA,nu1=NA,rho0=NA,nu0=NA)
    cls2= data.frame(gamma.mean.dn=
dn.mis3.relativeRisk,beta.dn=1,gamma.mean.CC=NA,beta.CC=NA,rho1=NA,nu1=NA,rho0=NA,nu0=NA)
    hyperpar=list(cls1=cls1,cls2=cls2)


    re.TADA <- do.call(cbind.data.frame, TADA(tada.counts=tada.counts,
sample.counts=sample.counts, mu=mu, hyperpar=hyperpar, denovo.only=denovo.only))
    re.TADA$qval=Bayesian.FDR(re.TADA$BF.total, pi0 = pi0)

    tadaResults <- re.TADA[order(re.TADA$qval, decreasing = F), ]

    probableGenes <- length(which(tadaResults$qval<0.3))
    highConfidenceGenes <- length(which(tadaResults$qval<0.1))
    return(data.frame(probable = probableGenes, highConfidence = highConfidenceGenes))
}

############################################################################
# RUN
############################################################################

RNGkind("L'Ecuyer-CMRG")
set.seed(1)
mc.reset.stream()

tadaSimulations <- mclapply(seq(from=25, to=3000, by=25), function(x)
    lapply(1:nPerms, function(y) runIteration(numGenes_f = numGenes, mutationProbs_f =
mutationProbs, cohortSize_f = x, lgdMutationRate_f = lgdRate, mis3MutationRate_f =
mis3Rate, lgdRiskFraction_f = lgdRiskFraction, mis3RiskFraction_f = mis3RiskFraction)),
    mc.cores = numCores, mc.set.seed = T)

save(tadaSimulations, file = paste("tadaSimulations", nPerms, "perms",
"forGeneDiscoveryEstimate_noPval.RData", sep="_"))

resultsByCohortSize <- lapply(tadaSimulations, function(x) do.call(rbind, x))

averageGeneDiscoveryByCohortSize <- lapply(resultsByCohortSize, function(x) apply(x, 2,
mean))

averageGeneDiscoveryByCohortSize_DF <- as.data.frame(do.call("rbind",
averageGeneDiscoveryByCohortSize))
averageGeneDiscoveryByCohortSize_DF$cohortSize <- seq(from=25, to=3000, by=25)
```

```
save(averageGeneDiscoveryByCohortSize_DF, file =
paste("averageGeneDiscoveryByCohortSize", nPerms, "perms", ".RData", sep="_"))

toPlot <- melt(averageGeneDiscoveryByCohortSize_DF, measure.vars=c("probable",
"highConfidence"),
               variable.name = "confidenceThreshold", value.name = "numGenes")

save(toPlot, file = paste("averageGeneDiscoveryByCohortSizetoPlot", nPerms, "perms",
".RData", sep="_"))

p <- ggplot(toPlot, aes(x=cohortSize, y=numGenes, col=confidenceThreshold))
p <- p + geom_line()

ggsave(p, file=file.path(plotDir, paste("averageGeneDiscoveryByCohortSize", nPerms,
"perms.pdf", sep="_")))
```

### Gene set overlap using DNENRICH

We used DNENRICH (6) (https://psychgen.u.hpc.mssm.edu/dnenrich/) to test whether OCD genes harboring de novo damaging mutations (89 genes; excluding two genes, *TTN* and *CACNA1E*, found to harbor de novo damaging variants in control subjects) were significantly enriched among previously reported genes identified in autism (ASD), schizophrenia (SCZ), developmental disorders (DD), Tourette’s disorder (TD), and intellectual disability (ID). Gene lists were obtained from a recent cross-disorder study (12) that included de novo single nucleotide and indel variants from multiple exome sequencing studies in ASD (13-17), SCZ (6, 18-21), and ID (22-25). Three of these studies also included de novo variants present in unaffected siblings (13, 14, 19). DD (26) and TD (27) genes were obtained from recently published WES studies. Because our study included 42 trios that were also included in the TD study by Wang et al. (27), we performed enrichment analysis using a TD gene list of de novo mutations from probands in Wang et al. with TD only (no OCD, which is comorbid in 37% of the cohort), to avoid confounding our overlap analysis.  DNENRICH simulates random mutations while accounting for gene size, trinucleotide context, and mutational effect. We performed 100,000 permutations, comparing the observed and expected overlap with each gene set. Empirical p-values were generated, based on a one-sided enrichment analysis under a binomial model of greater than expected hits per gene set. We tested for overlap between our OCD genes and those in ASD, SCZ, DD, TD, ID, and unaffected siblings that harbored de novo (LGD, nonsynonymous, synonymous) mutations. We also tested for overlap with ASD genes achieving q<0.01 in a recent meta-analysis (28) and DD genes achieving genome-wide statistical significance (26). Given that we identified *CHD8* as an OCD risk gene in our study, we tested for overlap with lists of genes that are targets of CHD8 in human brain (29). Finally, to determine whether the observed overlap between OCD and TD is more likely due to pleiotropy than known comorbidity, we performed a secondary DNENRICH analysis, using only mutations from OCD subjects without known tics or Tourette's Disorder. See Table 3, Table S6.

The following Linux commands were used to run the DNENRICH analysis:

```
dnenrich . 100000 alias.txt refseq_gene_sizes.txt gene_lists.set
mutations_ocd_damaging.mut > results;

csh extractDnenrichResults.csh results > results.txt
```

### Exploratory pathway and network analyses

To determine whether all genes harboring de novo damaging variants in OCD are enriched for specific biological pathways, we used the same gene list from our gene set overlap analysis (n=89) to identify the most significant canonical pathways, biological processes, networks, and diseases suggested by MetaCore (Clarivate Analytics, version 6.37, build 69500, https://portal.genego.com/) and Ingenuity Pathway Analysis (IPA, build version 4866170M, content version 46901286, release date 2018-11-21; Qiagen Bioinformatics, http://www.ingenuity.com/). The following default settings were used for MetaCore: Analyze Single Experiment Tool, species Homo sapiens, threshold 0, p-value 1, signals both. The following default settings were used for IPA: Reference set: Ingenuity Knowledge Base (Genes Only); direct and indirect relationships; does not include endogenous chemicals; consider only relationships where species = human and confidence = experimentally observed. See Table S8.

Using the GeNets algorithm (https://apps.broadinstitute.org/genets), we mapped all 89 genes harboring de novo damaging mutations in OCD onto the GeNets Metanetwork v1.0 to determine whether they are functionally connected. The GeNets Metanetwork contains integrated protein-protein interactions from InWeb3 (30), ConcensusPathDB (http://consensuspathdb.org) (31), and 5,057 drug-target interactions (32); the total network size is 530,532 interactions. The GeNets algorithm determines the density of the mapped network (density = number of edges / number of possible edges) and compares this to computed densities for randomly sampled gene sets. An empirically determined p-value is generated. The network is determined to be significantly more connected than random if the density is greater than 95% of the randomly sampled gene sets. Additionally, in the process of mapping our genes onto the Metanetwork, additional candidate genes are predicted, based on their connectivity to our input genes. An overall network connectivity p-value is generated, both with and without these additional predicted candidates. Also, as part of the GeNets analysis, gene "communities" were determined, defined as genes that are more connected to one another than they are to other groups of genes. See Figure S6 and Table S7. All GeNets results for this analysis are also available in interactive form here: https://www.broadinstitute.org/genets#/visualize/58d9425ea4e00291af652379.

## SUPPLEMENTAL TABLES

### Table S1 – Phenotype, exome sequencing metrics, and principal components analysis.

*(see "TableS1.xlsx")*

First tab contains individual-level sample information (columns A-K), including family ID, individual ID, phenotype, cohort, collection site, gender, capture platform, size of "callable exome", and parental age (years) at birth, where available. Column L lists reasons for any sample exclusions by quality control methods; "0" indicates that the sample was not excluded and was included in subsequent analyses. Columns M-AH list individual sample sequencing metrics generated using PicardTools, and GATK DepthOfCoverage tools. Columns AI-AS list individual sample sequencing metrics generated using PLINK/SEQ (i-stats; https://psychgen.u.hpc.mssm.edu/plinkseq/stats.shtml). Columns B, M-AS were included in Principal Components Analysis (PCA). Third tab contains cohort-level metrics calculated using samples passing quality control. ±95% confidence intervals are given, when applicable. Fourth tab contains coordinates generated for each sample for the top 10 principal components following PCA. The code used to generate this data is included in Supplementary Methods. Using these coordinates, we removed trios with family members falling more than three standard deviations from the mean in any of the first five principal components; this information is contained in the fifth tab.

### Table S2 – Annotated de novo variants in OCD and controls.

*(see "TableS2.xlsx")*

Detailed information on all high confidence de novo variants in cases and controls. These variants were annotated using Annovar, based on RefSeq hg19 gene definitions. Column descriptions are provided in a separate tab of this file. A third tab provides the number of each de novo variant type per sample.

### Table S3 – Annotated inherited variants in OCD and controls.

*(see “TableS3.xlsx”)*

Detailed information on all high confidence inherited variants in cases and controls. These variants were annotated using Annovar, based on RefSeq hg19 gene definitions. Column descriptions are provided in a separate tab of this file. A third tab provides the number of each inherited variant type per sample.

**Table S4 – Distribution of inherited coding variants in OCD cases and controls**

| Inherited variant type[a] | Variant counts | | Mutation rate (x10$^{-7}$) per bp (95% CI)[i] | | Estimated coding variants per individual (95% CI)[j] | | Rate ratio (95% CI) | p-value[k] |
|---|---|---|---|---|---|---|---|---|
| | OCD (N=184) | Control (N=777) | OCD (N=184) | Control (N=777) | OCD (N=184) | Control (N=777) | | |
| **Coding[b]** | 8320 | 31422 | 8.55<br>(8.37-8.74) | 8.57<br>(8.47-8.66) | 57.85<br>(56.63-59.13) | 57.98<br>(57.31-58.59) | 1.00<br>(0.97-1.02) | 0.92 |
| **Synonymous SNV** | 2362 | 9087 | 2.43<br>(2.33-2.53) | 2.48<br>(2.43-2.53) | 16.44<br>(15.76-17.12) | 16.78<br>(16.44-17.12) | 0.98<br>(0.75-1.31) | 0.34 |
| **Nonsynonymous[c]** | 5774 | 21752 | 5.94<br>(5.78-6.09) | 5.93<br>(5.85-6.01) | 40.19<br>(39.11-41.2) | 40.12<br>(39.58-40.66) | 1.00<br>(1.02-1.40) | 0.97 |
| **All Missense (Mis)** | 5230 | 19816 | 5.38<br>(5.23-5.53) | 5.40<br>(5.33-5.48) | 36.40<br>(35.38-37.41) | 36.54<br>(36.06-37.08) | 1.00<br>(0.97-1.03) | 0.78 |
| **Mis-D[d]** | 2228 | 8568 | 2.29<br>(2.20-2.39) | 2.34<br>(2.29-2.39) | 15.49<br>(14.88-16.17) | 15.83<br>(15.49-16.17) | 0.98<br>(0.94-1.03) | 0.42 |
| **MIs-P[e]** | 890 | 3495 | 0.92<br>(0.86-0.98) | 0.95<br>(0.92-0.98) | 6.22<br>(5.82-6.63) | 6.43<br>(6.22-6.63) | 0.96<br>(0.89-1.0) | 0.29 |
| **Mis-B[f]** | 1962 | 7424 | 2.02<br>(1.93-2.11) | 2.02<br>(1.98-2.07) | 13.67<br>(13.06-14.28) | 13.67<br>(13.4-14.01) | 1.00<br>(0.95-1.05) | 0.91 |
| **Likely Gene Disrupting (LGD)[g]** | 544 | 1936 | 0.56<br>(0.51-0.61) | 0.53<br>(0.50-0.55) | 3.79<br>(3.45-4.13) | 3.59<br>(3.38-3.72) | 1.06<br>(0.96-1.17) | 0.24 |
| **Damaging (LGD + Mis-D)** | 2772 | 10504 | 2.85<br>(2.75-2.96) | 2.86<br>(2.81-2.92) | 19.28<br>(18.61-20.03) | 19.35<br>(19.01-19.76) | 1.00<br>(0.95-1.04) | 0.84 |
| **LGD SNV** | 296 | 1075 | 0.30<br>(0.27-0.34) | 0.29<br>(0.28-0.31) | 2.03<br>(1.83-2.3) | 1.96<br>(1.89-2.1) | 1.04<br>(0.91-1.18) | 0.58 |
| **LGD frameshift indel** | 248 | 861 | 0.25<br>(0.22-0.29) | 0.23<br>(0.22-0.25) | 1.69<br>(1.49-1.96) | 1.56<br>(1.49-1.69) | 1.09<br>(0.94-1.25) | 0.27 |
| **Nonframeshift indel** | 94 | 298 | 0.097<br>(0.078-0.12) | 0.081<br>(0.072-0.091) | 0.66<br>(0.53-0.81) | 0.55<br>(0.49-0.62) | 1.19<br>(0.93-1.51) | 0.16 |
| **Unknown[h]** | 90 | 285 | 0.093<br>(0.074-0.11) | 0.078<br>(0.069-0.087) | 0.63<br>(0.50-0.74) | 0.53<br>(0.47-0.59) | 1.19<br>(0.93-1.51) | 0.17 |

[a] Variants were annotated with Annovar, using RefSeq hg19 gene definitions. [b] "Coding" variants include synonymous, nonsynonymous, nonframeshift, and those annotated as "unknown" by Annovar. [c] "Nonsynonymous" variants include all missense and LGD variants. [d] "Mis-D" are "probably damaging" missense variants with a Polyphen2 (HDIV) score ≥0.957. [e] Mis-P are "possibly damaging" missense variants with a Polyphen2

(HDIV) score <0.957 and ≥0.453. [f]Mis-B are “benign” missense variants with a Polyphen2 (HDIV) score <0.453. Two OCD missense variants and five control missense variants had no prediction by Polyphen2, but were included in the "All Missense (Mis)" variant type. [g]LGD variants are those altering a stop codon, canonical splice site, and frameshift indels. [h]“Unknown” variants are not included in the synonymous or nonsynonymous counts. [i]Inherited mutation rates were calculated as the number of variants divided by the number of haploid “callable” bases. [j]The estimated number of inherited mutations per individual was calculated by multiplying the mutation rate by the size of the RefSeq hg19 coding exome (33,828,798 bp). [k]Rates were compared using a two-sided rate ratio test.

**Table S5 – Gene-level de novo mutation rates, variant counts, and TADA results.**

*(see "TableS5.xlsx")*

First tab contains de novo mutation rates used to perform subsequent maximum likelihood estimation (MLE) and TADA analyses. The following mutation rates are listed for each gene: overall, likely gene disrupting (lgd), predicted damaging missense (misD), and all damaging (lgd + misD). These mutation rates were previously published (Ware et al., 2015) from unaffected parent-child trios. The code used to generate the mutation rate table is provided in Supplementary Methods. Second tab contains the input file for the TADA-Denovo code. Gene-level expected mutation rates for LGD ("mut.cls1" column) and Mis-D variants ("mut.cls2" column) are listed, along with their respective observed de novo mutation counts in our OCD data ("dn.cls1" and "dn.cls2", respectively). Code for running TADA-Denovo is given in Supplementary Methods. Third tab contains the final output results from TADA-Denovo code provided in Supplementary Methods. Genes harboring more than one damaging de novo (LGD or Mis-D) variant in OCD probands are highlighted in yellow (*SCUBE1, CHD8, TTN*). Two of these genes (*SCUBE1* and *CHD8*) exceeded thresholds for being considered a probable (qval < 0.3) or high confidence (qval < 0.1) risk gene. Fourth tab contains the input file for the TADA (Denovo + Inherited) code. Gene-level expected mutation rates for LGD ("mut.cls1" column) and Mis-D variants ("mut.cls2" column) are listed, along with their respective observed de novo mutation counts in our OCD data ("dn.cls1" and "dn.cls2", respectively), observed transmitted mutation counts (trans.cls1, trans.cls2), and observed non-transmitted mutation counts (present in either parent but not the child; ntrans.cls1, ntrans.cls2). Code for running TADA (Denovo + Inherited) is given in Supplementary Methods.

**Table S6 – DNENRICH gene lists and results.**

*(see "TableS6.xlsx")*

See Supplementary Methods for details of DNENRICH analysis. First tab contains the input gene lists to determine enrichment for our OCD damaging de novo mutations. Second tab contains input for DNENRICH analysis. Each row represents a de novo damaging mutation in an OCD proband. Third tab contains final results output from the DNENRICH primary analysis, also shown in Table 3. Fourth tab contains input for

DNENRICH secondary analysis; each row represents a de novo damaging mutation in an OCD proband without known tics or Tourette's disorder. Fifth tab contains the results of this  secondary analysis. Significantly enriched gene sets are highlighted.

**Table S7 – GeNets network connectivity analysis results.**

*(see "TableS7.xlsx")*

Complete results from GeNets network analysis of de novo damaging variants found in OCD probands. First tab contains summary statistics of the resulting network, considered both with and without nearby predicted "tier 1" candidate genes. Second tab contains the input gene list and the candidate genes predicted by the network analysis. Third tab groups genes (without predicted candidates) into nearby "communities" that are more connected with each other than their neighbors. Fourth tab contains network edges without the predicted candidates. Fifth tab contains gene community groupings, including predicted candidates. Sixth tab contains network edges including predicted candidates. See Methods for further details of this analysis.

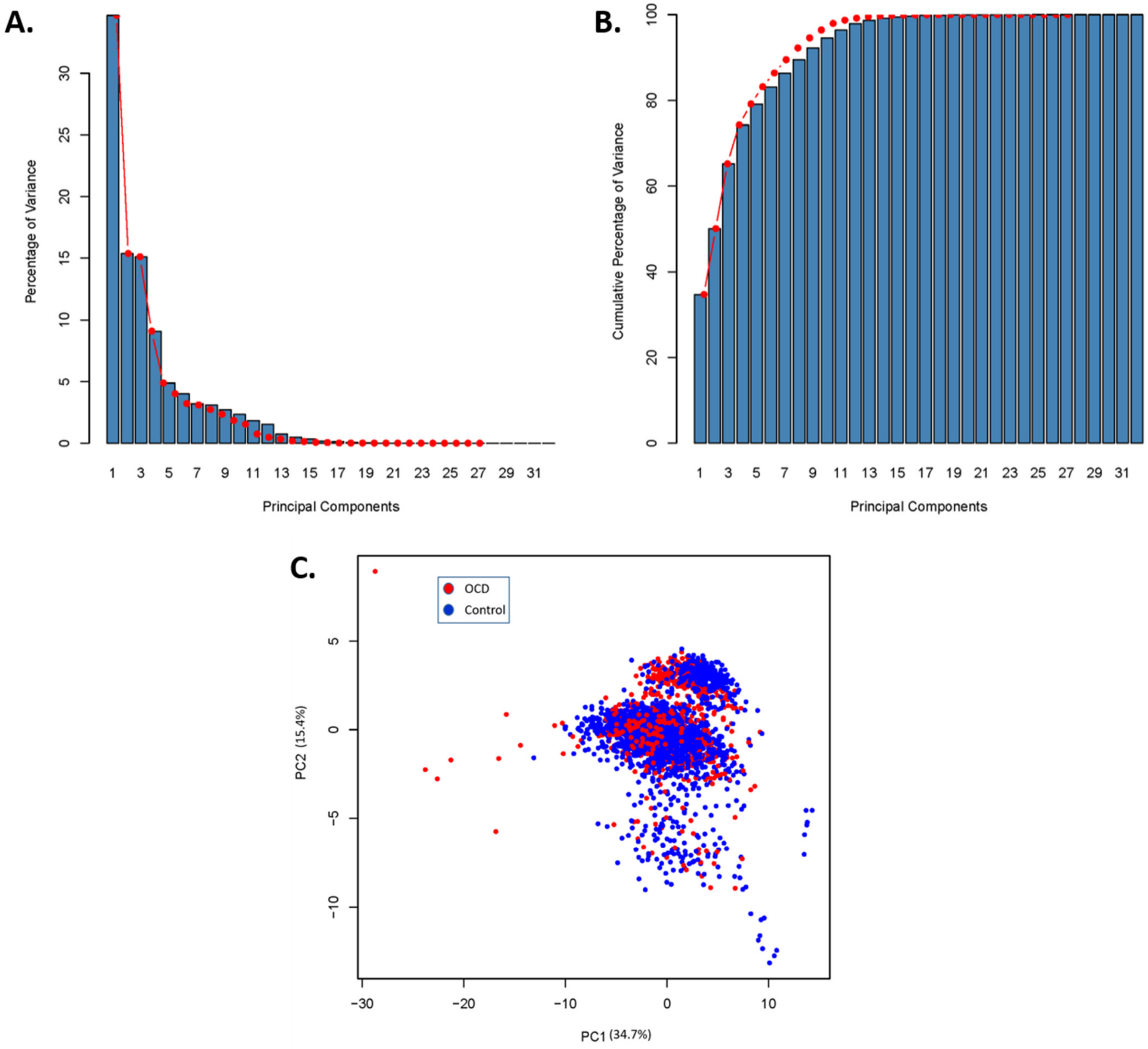
**TableS8 – MetaCore and Ingenuity Pathway Analysis (IPA) gene enrichment analysis results.**

*(see "TableS8.xlsx")*

Complete results from Metacore (first tab) and IPA (second tab) gene enrichment analyses, with p-values calculated by each analysis algorithm. See Methods for details of these analyses.

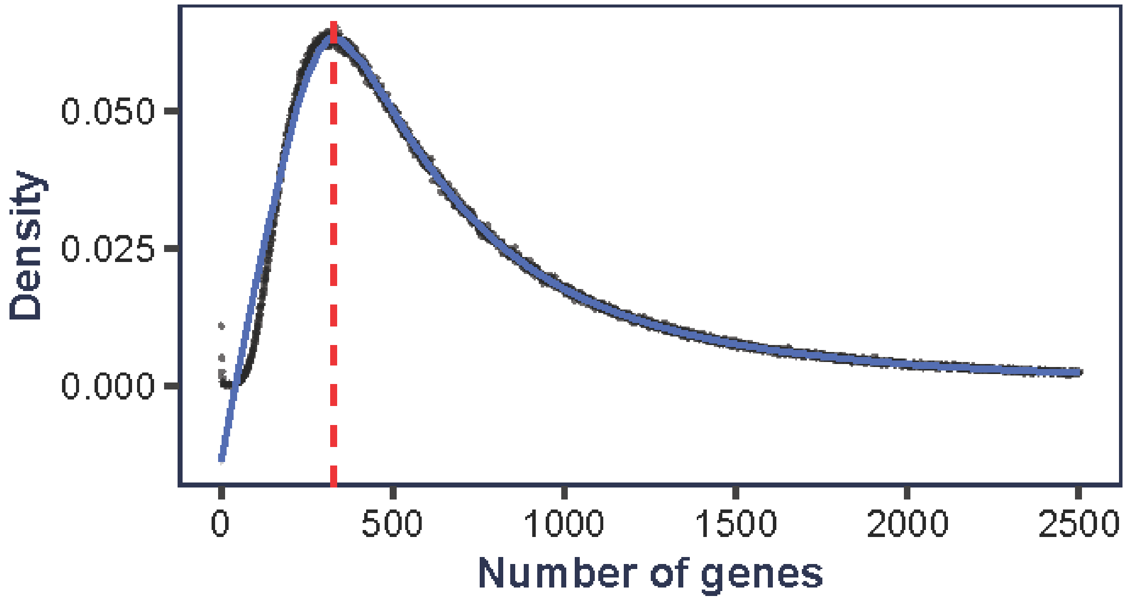
## SUPPLEMENTAL FIGURES

### Figure S1 – PCA scree and individual plots.



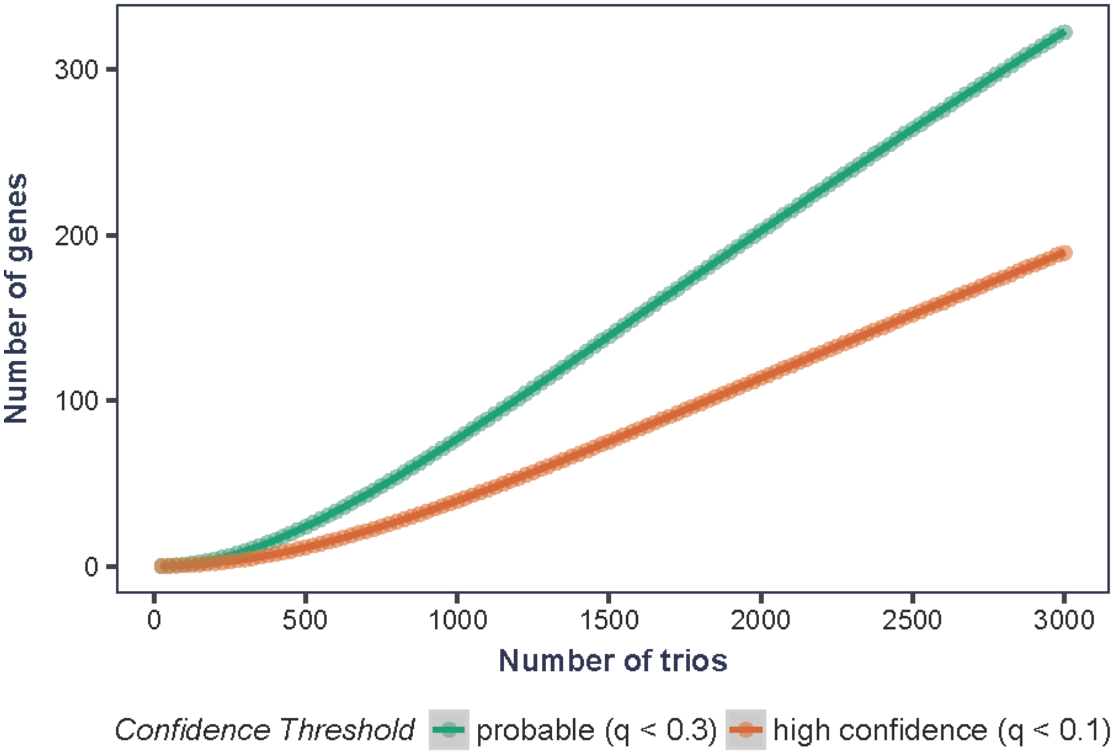
Scree plots following Principal Components Analysis (PCA), showing (A) the percentage of variance captured by each of the first 32 principal components, and (B) the cumulative percentage of variance captured by these same components in the exome metrics data from cases and controls. The “elbow” of the scree plot is visualized to be around the 5th principal component. This was confirmed by the Factominer R code function “estim_ncp()”. The first 5 PCs capture almost 80% of the variance, and this number of PCs was used to determine PCA outliers during quality control (see Table S1 and Supplementary Methods). (C) Individual plots for the first two principal components, based on PCA of exome sequencing quality metrics. OCD cases are plotted in red, and controls in blue. The first two PCs together capture 50.1% of the variance. R code to generate this data and figure are in Supplementary Methods, and individual PC factor values are in Table S1. This figure includes PCA outliers (>3 standard deviations from the mean in PCs 1-5), which were removed during quality control, prior to further analysis of case-control data.

**Figure S2 – Maximum Likelihood Estimate (MLE) of number of OCD risk genes.**



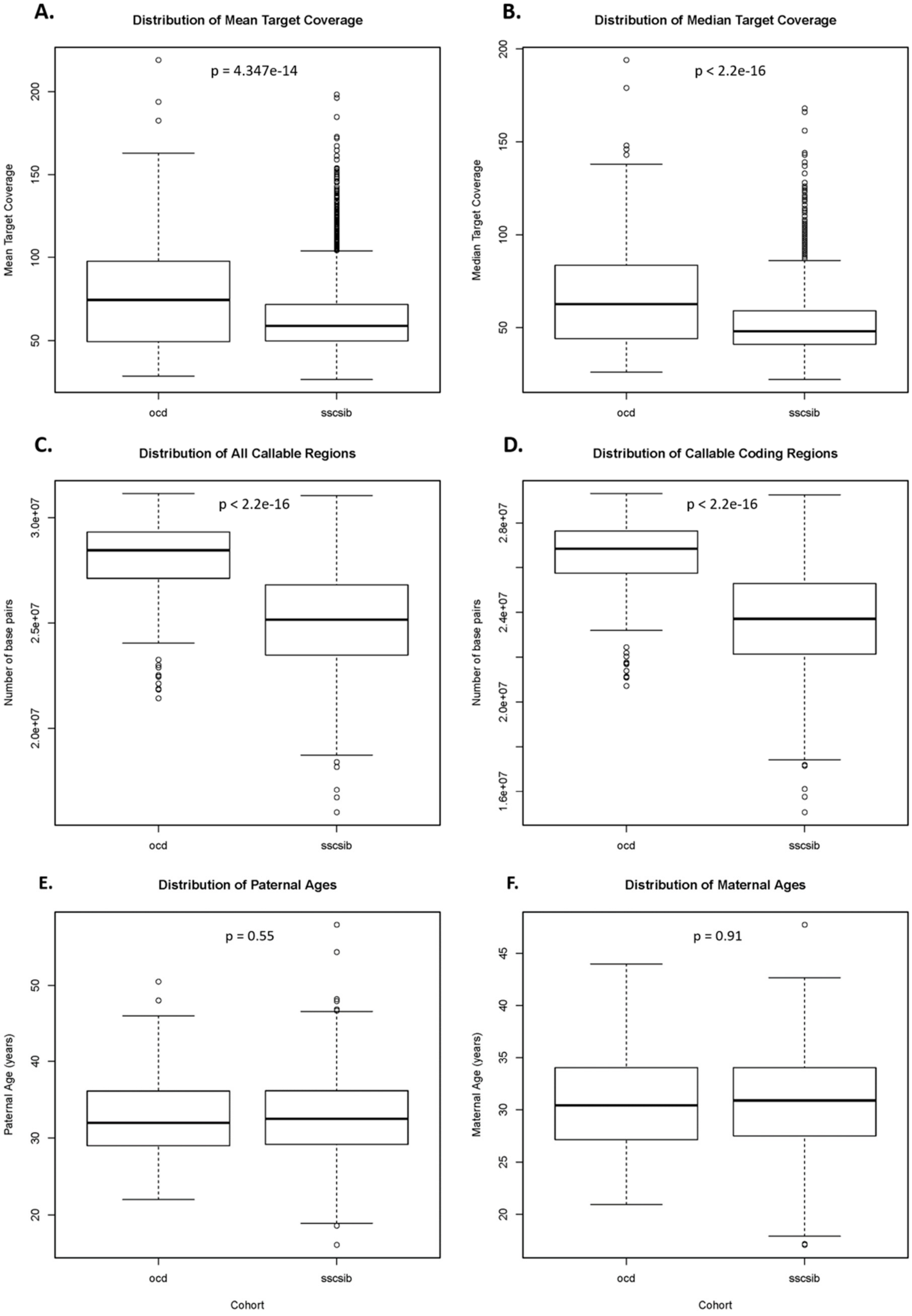
Assuming each number of possible risk genes between 1-2,500, 100,000 simulations were conducted to determine the number of risk genes that yielded the closest agreement between our observed and simulated data. In each simulation, we generated 95 variants (the number of de novo damaging variants observed in our OCD sample), then randomly assigned a percentage of variants (determined by the fraction of de novo damaging variants estimated to carry OCD risk) to the risk genes, recording the frequency at which the number of genes with two and three recurrent variants matched the number observed in our study (2 and 1, respectively). This MLE method yields an estimate of 335 OCD risk genes (red vertical line), a number that is in close agreement with that from an alternative "unseen species" method (317 genes, 95% CI: 190-454).

**Figure S3 – Gene discovery by number of trios sequenced.**



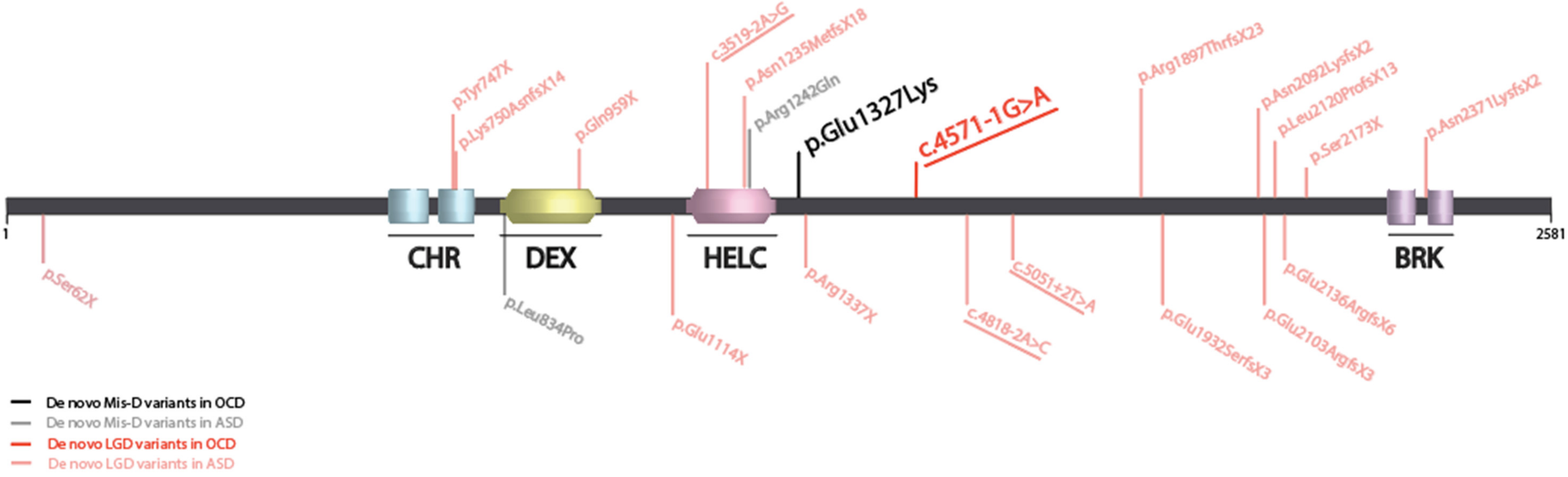
Using the MLE estimate of 335 risk genes, we estimated the number of probable (q<0.3) and high-confidence (q<0.1) risk genes that will be discovered as more OCD trios are sequenced. We performed 10,000 simulations at each cohort size from 25-3,000 trios, randomly generating variants and assigning to risk genes in agreement with the proportions seen in our data, then applying the TADA-Denovo algorithm.

**Figure S4 – Sequencing coverage and parental age distributions.**



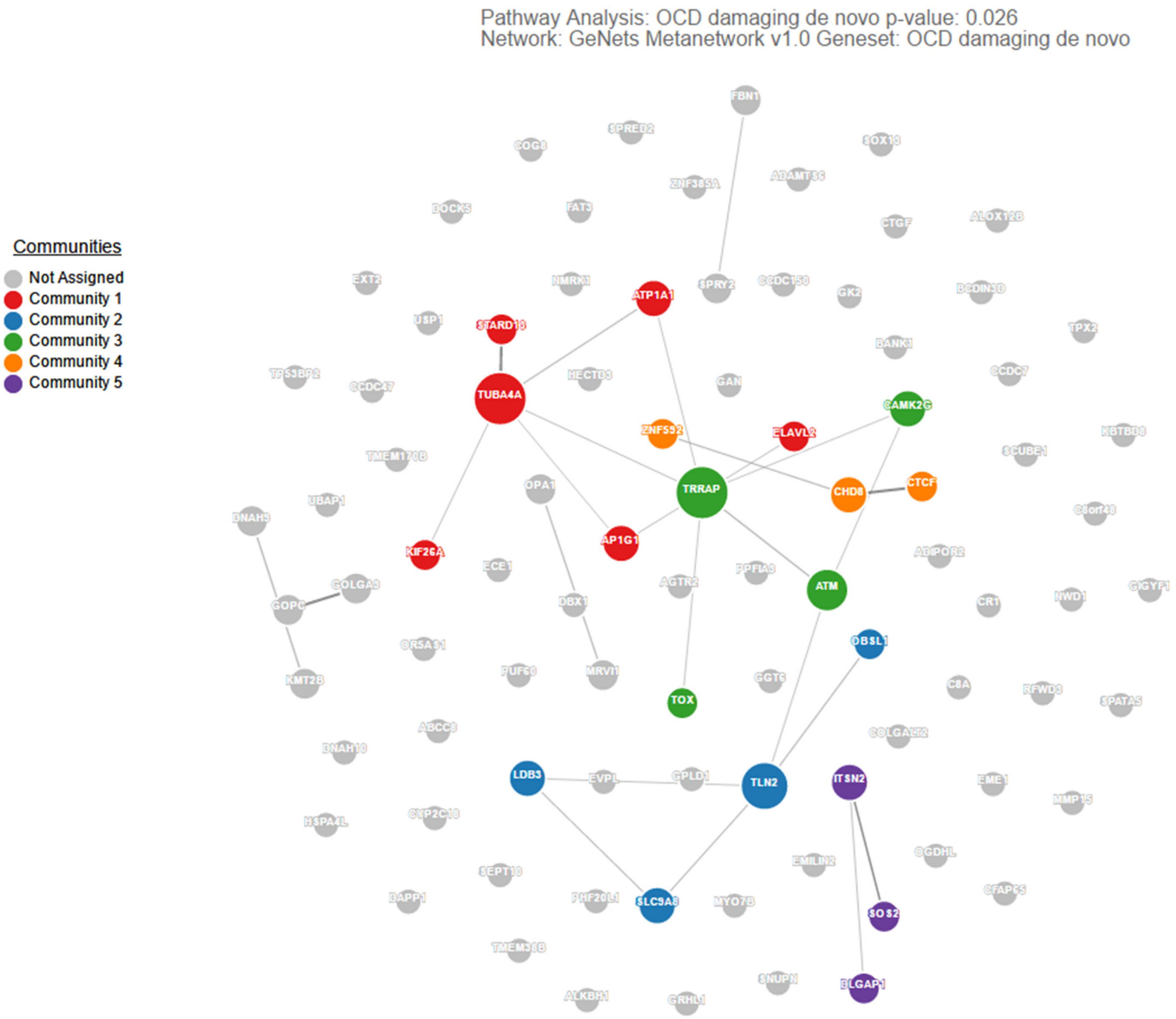
Distribution boxplots of values for (A) mean target coverage, (B) median target coverage, (C) number of base pairs in all “callable” regions, (D) number of base pairs in coding “callable” regions, (E) paternal age, and (F) maternal age for both OCD and control cohorts. For each cohort, the box extends from the first through third quartiles, and the horizontal line is at the second quartile (median) of the data.  Whiskers extend to the largest non-outliers, and outlier data points are plotted individually. For each comparison, a p-value was calculated using a two-sided Wilcoxon rank sum test with continuity correction. Panels A-D show increased opportunity for variant calling in the OCD cohort, necessitating the use of de novo mutation rate comparisons within the callable exome, as explained in the main text and methods. Panels E-F show no significant difference in parental ages between case and control cohorts. Also see Table S1.

**Figure S5 – CHD8 variants in OCD and ASD.**



Two de novo likely gene disrupting (LGD, red) and damaging missense (Mis-D, black) variants identified in CHD8 among OCD probands are indicated. ASD-associated de novo LGD and Mis-D mutations reported in the Simons Foundation Autism Research Initiative (SFARI) database (accessed April 12, 2017) are also shown in muted colors. Only variants with identifiable allele or residue sequence positions in the SFARI database were included in the above protein diagram, and splice site variants across cohorts are indicated by the respective allele change and underlined. Annotated protein domains predicted with confidence by the Simple Modular Architecture Research Tool (SMART) are shown as follows: CHR, chromatin organization modifier domain (blue), DEX, DEAD-like helicases superfamily (yellow), HELC, helicase superfamily c-terminal domain (pink), and BRK, domain of unknown function associated with CHROMO domain helicases (purple).

**Figure S6 – GeNets network analysis without candidates.**



Using the GeNets algorithm (https://apps.broadinstitute.org/genets), we mapped all 89 genes harboring de novo damaging mutations in OCD (excluding two genes, *TTN* and *CACNA1E*, which harbored de novo damaging variants in control subjects) onto the GeNets Metanetwork v1.0 to determine whether they are functionally connected. The density of the mapped network (density = number of edges / number of possible edges) was greater than 95% of randomly sampled gene sets, indicating that the network is significantly more connected than random (p=0.026). In the figure, node (gene) size is proportional to the number of connections.

Node color indicates “community” assignment. A community is a set of genes that are more connected to one another than to another group of genes. Interactive results are available here: https://www.broadinstitute.org/genets#/visualize/58d9425ea4e00291af652379.

## SUPPLEMENTAL REFERENCES

1. McKenna A, *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* 20(9):1297-1303.

2. Li H & Durbin R (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* 26(5):589-595.

3. Wang K, Li M, & Hakonarson H (2010) ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res* 38(16):e164.

4. Manichaikul A, *et al.* (2010) Robust relationship inference in genome-wide association studies. *Bioinformatics (Oxford, England)* 26(22):2867-2873.

5. Danecek P, *et al.* (2011) The variant call format and VCFtools. *Bioinformatics (Oxford, England)* 27(15):2156-2158.

6. Fromer M, *et al.* (2014) De novo mutations in schizophrenia implicate synaptic networks. *Nature* 506(7487):179-184.

7. Lek M, *et al.* (2016) Analysis of protein-coding genetic variation in 60,706 humans. *Nature* 536(7616):285-291.

8. Ware JS, Samocha KE, Homsy J, & Daly MJ (2015) Interpreting de novo Variation in Human Disease Using denovolyzeR. *Current protocols in human genetics / editorial board, Jonathan L. Haines ... [et al.]* 87:7.25.21-15.

9. Sanders SJ, *et al.* (2012) De novo mutations revealed by whole-exome sequencing are strongly associated with autism. *Nature* 485(7397):237-241.

10. Homsy J, *et al.* (2015) De novo mutations in congenital heart disease with neurodevelopmental and other congenital anomalies. *Science* 350(6265):1262-1266.

11. He X, *et al.* (2013) Integrated model of de novo and inherited genetic variants yields greater power to identify risk genes. *PLoS Genet* 9(8):e1003671.

12. Shohat S, Ben-David E, & Shifman S (2017) Varying Intolerance of Gene Pathways to Mutational Classes Explain Genetic Convergence across Neuropsychiatric Disorders. *Cell reports* 18(9):2217-2227.

13. Iossifov I, *et al.* (2014) The contribution of de novo coding mutations to autism spectrum disorder. *Nature* 515(7526):216-221.

14. Iossifov I, *et al.* (2012) De novo gene disruptions in children on the autistic spectrum. *Neuron* 74(2):285-299.

15. De Rubeis S, *et al.* (2014) Synaptic, transcriptional and chromatin genes disrupted in autism. *Nature* 515(7526):209-215.

16. Neale BM, *et al.* (2012) Patterns and rates of exonic de novo mutations in autism spectrum disorders. *Nature* 485(7397):242-245.

17. O'Roak BJ, *et al.* (2012) Sporadic autism exomes reveal a highly interconnected protein network of de novo mutations. *Nature* 485(7397):246-250.

18. Girard SL, *et al.* (2011) Increased exonic de novo mutation rate in individuals with schizophrenia. *Nat Genet* 43(9):860-863.

19. Gulsuner S, *et al.* (2013) Spatial and temporal mapping of de novo mutations in schizophrenia to a fetal prefrontal cortical network. *Cell* 154(3):518-529.

20. McCarthy SE, *et al.* (2014) De novo mutations in schizophrenia implicate chromatin remodeling and support a genetic overlap with autism and intellectual disability. *Mol Psychiatry* 19(6):652-658.

21. Xu B, *et al.* (2011) Exome sequencing supports a de novo mutational paradigm for schizophrenia. *Nat Genet* 43(9):864-868.

22. de Ligt J, *et al.* (2012) Diagnostic exome sequencing in persons with severe intellectual disability. *The New England journal of medicine* 367(20):1921-1929.

23. Gilissen C, *et al.* (2014) Genome sequencing identifies major causes of severe intellectual disability. *Nature* 511(7509):344-347.

24. Hamdan FF, *et al.* (2014) De novo mutations in moderate or severe intellectual disability. *PLoS Genet* 10(10):e1004772.

25. Rauch A, *et al.* (2012) Range of genetic mutations associated with severe non-syndromic sporadic intellectual disability: an exome sequencing study. *Lancet* 380(9854):1674-1682.

26. Deciphering Developmental Disorders Study (2017) Prevalence and architecture of de novo mutations in developmental disorders. *Nature*.

27. Wang S, *et al.* (2018) De Novo Sequence and Copy Number Variants Are Strongly Associated with Tourette Disorder and Implicate Cell Polarity in Pathogenesis. *Cell reports* 24(13):3441-3454.e3412.

28. Sanders SJ, *et al.* (2015) Insights into Autism Spectrum Disorder Genomic Architecture and Biology from 71 Risk Loci. *Neuron* 87(6):1215-1233.

29. Cotney J, *et al.* (2015) The autism-associated chromatin modifier CHD8 regulates other autism risk genes during human neurodevelopment. *Nature communications* 6:6404.

30. Lage K, *et al.* (2007) A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat Biotechnol* 25(3):309-316.

31. Kamburov A, Stelzl U, Lehrach H, & Herwig R (2013) The ConsensusPathDB interaction database: 2013 update. *Nucleic Acids Res* 41(Database issue):D793-800.

32. Rask-Andersen M, Masuram S, & Schioth HB (2014) The druggable genome: Evaluation of drug targets in clinical trials suggests major shifts in molecular class and indication. *Annual review of pharmacology and toxicology* 54:9-26.