UNIVERSITY OF COPENHAGEN
FACULTY OF HEALTH AND MEDICAL SCIENCES

PhD Thesis

Simone Montalbano

From Fixed Loci To Genome-Wide

Advancements In The Detection And Analysis Of Copy Number Variants In Large-Scale Datasets

Supervisor: Thomas Mears Werge

This thesis has been submitted to the Graduate School of Health and Medical Sciences,
University of Copenhagen February 1st 2025

Summary

Copy Number Variations (CNVs) have been a focus of human genetic research for decades. Particular attention has been given to large recurrent CNVs, meaning deletions or duplications occurring in the same locus across different individuals in a population. Several recurrent CNVs have been associated to human disorders and syndromes, including mental and developmental disorders such as Schizophrenia Spectrum Disorder (SSD), Autism Spectrum Disorder (ASD), and Attention Deficit Hyperactivity Disorder (ADHD), often with very high risk estimates. However, such associations often originate from highly selected syndromic case studies and collections, and have been shown to not hold, at least not to the same degree, in more population representative samples such as iPSYCH2015. Moreover, recurrent CNVs represent only a minor fraction of the overall genomic variation represented by structural variants, and studies assessing the impact of CNVs on a Genome-Wide scale are still rare and unrefined, underlining how detecting and analysing CNVs across the entire genome remains a challenging task.

In this thesis, with the four main manuscripts included, and the other papers from my colleagues that I co-authored during my PhD, I present my effort to push forward the research on the biology of CNVs and their association to human traits in large SNPs-genotyped cohorts. The first manuscript describes our pipeline to reliably call and analyse recurrent CNVs. This has been the base for several other research projects. The second manuscript studies the association of *NRXN1* deletions and mental disorders in the iPSYCH2015 case-cohort study. It also served as an exploratory ground for novel analytical strategies. The third and fourth manuscripts are tied together as they are centered around the genome-wide validation of CNV calls using machine vision instead of human analysts. They describe the method and its application in two large cohorts, respectively.

Resumé på dansk

Kromosomforandringer (e. Copy Number Variations; CNVs) har været i fokus for human genetisk forskning i flere årtier. Særlig opmærksomhed er blevet givet til store tilbagevendende CNVs, hvilket betyder deletioner eller duplikationer, der forekommer på samme locus uafhængigt på tværs af forskellige individer i en population. Flere tilbagevendende CNVs er blevet associeret med menneskelige lidelser og syndromer, herunder mentale og udviklingsmæssige forstyrrelser såsom Skizofrenispektrumforstyrrelse (SSD), Autismespektrumforstyrrelse (ASD), og Opmærksomhedsforstyrrelse med hyperaktivitet (ADHD), ofte med meget høje risikoestimater. Sådanne associationer stammer dog ofte fra højt selekterede syndromiske casestudier og -samlinger, og det er blevet påvist, at de ikke holder stik, i hvert fald ikke i samme grad, i mere populationsrepræsentative kohorter såsom iPSYCH2015. Desuden repræsenterer tilbagevendende CNVs kun en mindre del af den samlede genomiske variation, der kendetegnes af strukturelle varianter, og studier, der vurderer CNVs indvirkning på genomet som helhed, er stadig sjældne og uraffinerede, hvilket understreger, hvordan detektion og analyse af CNVs på tværs af hele genomet forbliver en udfordrende opgave.

I denne afhandling, med de fire hovedmanuskripter inkluderet, og de andre artikler fra mine kolleger, som jeg var medforfatter på under min ph.d., præsenterer jeg min indsats for at fremme forskningen i CNVs biologi og deres association til menneskelige træk i store SNP-genotyperede kohorter. Det første manuskript beskriver vores pipeline til at identificere, validere og analysere tilbagevendende CNVs. Dette har været grundlaget for flere andre forskningsprojekter. Det andet manuskript undersøger associationen mellem NRXN1-deletioner og psykisk sygdom i iPSYCH2015 case-kohortestudiet. Det tjente også som et undersøgende grundlag for nye analytiske strategier. Det tredje og fjerde manuskript er knyttet sammen, da de er centreret omkring validering af CNV-fund på tværs af hele genomet ved hjælp af maskinelt syn i stedet for visuelle analytikere. De beskriver henholdsvis metoden og dens anvendelse i to store kohorter.

Contents

List of Manuscripts Included in the Thesis

1. Montalbano, S. et al. Accurate and Effective Detection of Recurrent Copy Number Variants in Large SNP Genotype Datasets. Current Protocols 2, e621 (2022).
2. Montalbano, S. *et al.* Analysis of exonic deletions in a large population study provides novel insights into NRXN1 pathology. *npj Genom. Med.* **9**, 1–10 (2024).
3. Montalbano, S. et al. CNValidatron, automated validation of CNV calls using computer vision. 2024.09.09.612035 Preprint at https://doi.org/10.1101/2024.09.09.612035 (2024).
4. Montalbano, S. et al. A Characterisation of Genome Wide CNVs in two population-scale datasets and their impact on the human genome. Manuscript included in this thesis, not yet submitted either as preprint or to a peer-reviewed journal.

Research Papers Co-authored During the PhD

Published

- Calle Sánchez, X. et al. Comparing Copy Number Variations in a Danish Case Cohort of Individuals With Psychiatric Disorders. JAMA Psychiatry 79, 59–69 (2022).
- Sánchez, X. C. et al. Associations of psychiatric disorders with sex chromosome aneuploidies in the Danish iPSYCH2015 dataset: a case-cohort study. The Lancet Psychiatry 10, 129–138 (2023).
- Vaez, M. et al. Population-Based Risk of Psychiatric Disorders Associated With Recurrent Copy Number Variants. JAMA Psychiatry (2024) doi:10.1001/jamapsychiatry.2024.1453.

Preprints

- Vaez, M. *et al.* Evaluating the Joint Effects of Recurrent Copy Number Variants and Polygenic Scores on the Risk of Psychiatric Disorders in the iPSYCH2015 Case-Cohort Sample. 2024.09.23.24314234 Preprint at https://doi.org/10.1101/2024.09.23.24314234 (2024).

List of Abbreviations

ADHD:	Attention Deficit Hyperactivity Disorder
ASD:	Autism Spectrum Disorder
BAF:	B Allele Frequency
CN:	Copy Number
CNN:	Convolutional Neural Network
CNV:	Copy Number Variation
GCWF:	GC waviness factor
HMM:	Hidden Markov Model
IOU:	Intersection Over the Union
LLM:	Large Language Model
LRR:	Log R Ratio
NN:	Neural Network
SCA:	Sex Chromosome Aneuploidy
SNP:	Single Nucleotide Polymorphism
SNV:	Single Nucleotide Variant
SSD:	Schizophrenia Spectrum Disorder
SV:	Structural Variant
UKB:	UK Biobank

Research Objectives

This section briefly presents the main objectives for each of the included manuscripts, as well as some broader goals of the projects behind the actual research paper.

- **Manuscript I: CNV calling protocol in large genotype collections**.

  CNV calling from SNPs data is a relatively straightforward bioinformatic task when performed on a small number of individuals, however it can become complex when scaled to very large collections of samples. Moreover, expert knowledge is often required to fine tune small but potentially very important details of the analysis. With this manuscript and the attached software we wanted to design a robust protocol to perform CNV calling on large collections of human genetic data, condense the field general knowledge and the expertise of our institution into easy-to-follow guidelines and provide a framework to efficiently study recurrent CNVs. Finally, several design choices serve as the foundations of a larger framework to work with CNVs from SNPs array in a modern and fast way also for future software.

- **Manuscript II: *NRXN1* deletions and mental disorders**.

  The *NRXN1* locus is a known hotspot for non-recurrent CNVs, meaning CNVs are abundant in the region but do not follow specific patterns in their position. Accordingly, the main objective for this project was to design a study that could detail such intricacies. We intended to provide precise estimates of the population prevalence of deletions in the locus, to assess the risk conferred by such deletions to mental disorders in the Danish population and, finally, to test whether the risk can be fine mapped. The project also served as a fundamental step in moving from fixed loci CNVs to a more complex genomic landscape. We experimented with clustering, as well as with gene structures (exons and introns), to differentiate an heterogeneous set of CNVs into a smaller set of biologically-meaningful and internally-homogeneous groups.

- **Manuscript III: Accurate and automated validation of CNVs calls, genome wide.**

  Since its original publication in 2007, PennCNV has been the *de facto* standard for CNV calling from SNPs array data. However,, PennCNV calls are prone to a very high number of false positives hits. This is especially problematic considering CNVs are generally rare or very rare, thus even a small proportion of false positives might strongly bias an estimate. In this manuscript we present a novel software that is able to drastically reduce this number, with little effect on the true positives. We also use the opportunity of working with trios data to provide estimates of false negative and *de novo* rates. This project was the culmination of the technical part of my PhD project, and addressed what was possibly our strongest limitation in CNVs studies, i.e. the extreme effort required to get good sets of CNV calls.

- **Manuscript IV: Genome wide rare CNVs in two large biobanks.**

  We applied the newly developed method to the entirety of UK Biobank and the Icelandic biobank at deCODE genetics. The main objective for this study was to perform an extensive characterization of this type of variation, in many ways for the first time on such a scale, being both genome wide and in a very large sample size. The goal was to understand if, how and, possibly, why deletions and duplications occur at different frequencies across the human genome, what this can tell us about specific classes of genomic annotations, and what consequences this might have. More technical objectives were to establish how stable the distribution of detected CNVs is across different cohorts and how differences in population structure and, more importantly, genotyping chip can impact the final dataset, and thus understand how different samples can be effectively combined in large studies..

Introduction

Human Genetic Variation

Genetic variation, especially in humans, is a key topic of this thesis. In this section I will briefly cover the main concepts that are relevant for the work presented. The focus will thus be on structural variants.

Small Vs Large, Common Vs Rare; Types Of Genetic Variation

Genetic variation is an extremely heterogeneous concept and it is categorised in different classes, however, very often there are no clear distinctions between groups. In general, we can define two main axes to differentiate the types of variation: small-large and common-rare. The small variants consist of single nucleotide variants (SNVs), i.e. changes of one single basepair with respect to the reference genome, and indels, i.e. small insertions and deletions typically up to ~50bp.[1,2] SNVs that are found in at least 1% of the population are known as SNPs (single nucleotide polymorphisms), even though the threshold is not always clear,[3] and in practice the two terms are used also depending on the study type, with researchers in the GWAS (genome wide associations studies) field[4] using mostly “SNP” and researchers in the rare-variants/WES (whole exome sequencing) field[5] using mostly “SNV”. Larger variants affecting segments of DNA, from ~50bp to full chromosomes, are collectively known as structural variants (SVs).[6] These are an extremely heterogeneous class and can be further subgrouped, even though there is no clear consensus in the literature. For the purpose of the studies included in this thesis we propose the following working classification in three groups: large chromosomal abnormalities, CNVs (copy number variants), and “other structural variants” (small and/or complex SVs). This definition is partially based on the actual size of the variant (very large, medium-to-large, small-to-medium) but also, crucially, on the technology typically used for the detection of the variant (see also “CNVs Detection” section). Moreover, research fields have different “traditions” and thus are typically approached from different angles. Chromosomal aneuploidies and syndromic recurrent CNVs are usually detected using microscopy or microarrays techniques and are most often studied as single rare exposures[7,8], while smaller structural variants are usually detected using genome sequencing and studied as aggregate variation, e.g. using burden tests.[9,10] In this thesis, we define CNVs as deletions and duplications from ~50kbp to 10Mbp, i.e. the ideal size to be detected using genotyping chips. These can be grouped in two classes, recurrent and non-recurrent, as detailed in the following subsections.

Recurrent CNVs

Recurrent CNVs (rCNVs) consist of deletions and duplications that are found in a population in predictable locations (hence the term “recurrent”), even though they are not necessarily segregating. rCNVs typically occur due to non allelic homologous recombination (NAHR) mediated by low copy repeat (LCR) regions (figure 1), mostly during meiosis.[11] For this reason, even though such variants are often actively selected against, they are found at a somewhat stable frequency in the population.[6,12] rCNVs are typically large in size from ~0.5Mbp to ~4Mbp.[13] Some recurrent CNVs are associated with very specific syndromes, such as the 22q11.2 microdeletion and the DiGeorge syndrome.[14] Given the large size, these were often discovered well before the advent of modern molecular biology techniques (see also the section “Chromosomal Structural Abnormalities and Mental Disorders”).

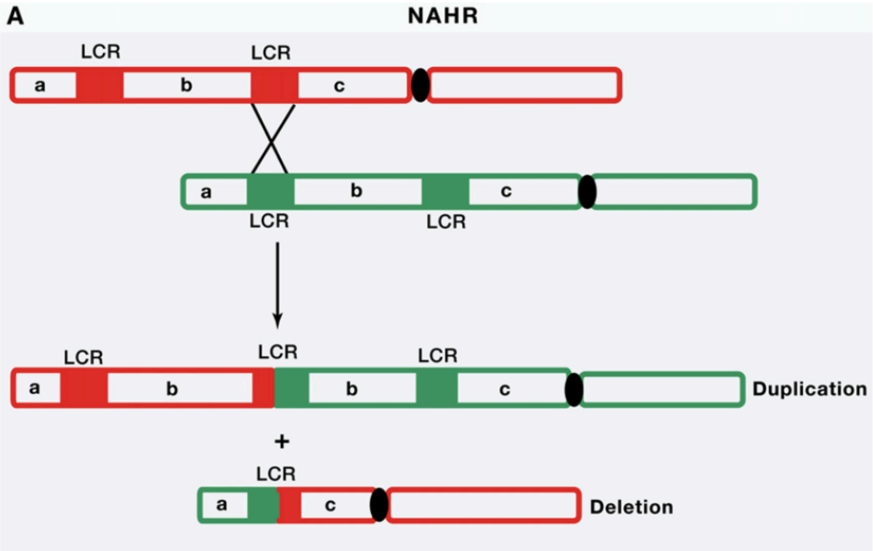


Figure 1.

Nonallelic homologous recombination (an unbalanced crossover) gives rise to a deletion on one chromosome and a duplication on the other. Adapted from Malotra and Sebat, 2012.[6]

Non-recurrent CVNs

As the name suggests, non-recurrent CNVs are grouped together more because of their difference from rCNVs, rather than some precise characteristic, and for this reason, they can be quite heterogeneous. Broadly speaking, we define non-recurrent all those CNVs that do not arise from NAHR and thus do not have predictable boundary regions. These CNVs can occur from different mechanisms both sequence homology based, such as non-homologous end joining (NHEJ)[15] and microhomology-mediated end joining (MMEJ)[16], and replication based, such as fork stalling and template switching (FoSTeS)[17] and microhomology-mediated break-induced replication (MMBIR)[18]. Briefly, homology based mechanisms are based on breaks in the DNA that are then erroneously repaired due to sequence homology with a different part of the chromosome, usually leading to deletions or insertions. In contrast, replication based mechanisms involve problems in the replication fork and often lead to more complex rearrangements. Of note, also retrotransposons can be defined as non recurrent CNVs[6], however they tend to be small in size and not tagged by SNPs, so they are usually not accessible by microarray experiments. CNVs are not randomly distributed across the genome but are known to accumulate in specific loci more than others, even when non-recurrent. It is reported that centromeric and telomeric regions[19], regions of low mappability (such as regions with high contents of repetitive DNA)[20], and also late replicating regions in general[21] are enriched in CNVs.

Consequences of Genetic Variation

The current paradigm in population genetics, especially for researchers focused on common variants such as SNPs, is the so-called threshold model.[22] Briefly, in this model the liability toward a (complex) trait, meaning the combination of all genetic and environmental exposures, is a normally distributed variable, and individuals with a liability higher than a certain value, the threshold, will express the phenotype. In this paradigm, rare and high risk exposures such as recurrent CNVs and protein truncating variants are somewhat treated as a separate source of genetic variation compared to SNPs. The interpretation is that if an individual carries a rare variant conferring high risk for a given phenotype (e.g. a disease) less liability from the common variants will be "needed" to express the phenotype or, in other words, the threshold for that individual would be moved towards the middle of the distribution.[23–25] In general, rare variants have been understudied by classic population genetics. This is both because of systemic limitations, such as the popularity of the paradigm 'common disease-common variants' (CDCV)[26], but also for instrumental limitations.[27] Rare variants are not commonly tagged by genotyping markers, moreover disease-associated variants often occur *de novo*.[28,29]

Mental Disorders

Mental disorders are the main focus of my research institute. In this section I present a brief description of the five main diagnoses included in the iPSYCH study (see "Dataset Used" section). These are ADHD (attention deficit hyperactivity disorder), ASD (autism spectrum disorder), BPD (bipolar disorder), MDD (major depressive disorder), and SSD (schizophrenia spectrum disorder). I also summarise the ties between the histories of mental health research and chromosomal rearrangements.

Core iPSYCH Disorders

ADHD is a neurodevelopmental disorder characterised a pattern of behaviours and symptoms impacting many parts of an individual's life, including impaired attention, impulsivity, fidgeting and hyperactivity.[30] The disorder is very heterogeneous both in terms of clinical manifestation, age of diagnosis and life trajectories.[31] ADHD has high heritability estimates, consistently among studies.[32] Moreover, according to evidence from adoption studies, genetic factors seem to confer higher risk than shared environmental factors.[32] Protein truncating variants[33] and some specific CNVs[34,35] are enriched in cases.

ASD is a heterogeneous group of neurodevelopmental disorders, both in terms of symptoms and their severity.[30] It is considered an early-onset disorder, as patients tend to receive a diagnosis before the age of three.[6,36] ASD has strong genetic components with association from common variants, rare variants and also CNVs.[37,38] Of note, *de novo* variation has been a key focus of ASD research.[6,29] The prevalence for ASD has increased rapidly in the last decades, and it is known that males are almost twice as likely to receive an ASD diagnosis than females.[39] Finally, ASD frequently co-occur with different secondary phenotypes: Intellectual Disability (ID) is present in ~35-60% of cases, and motor deficits, sleep abnormalities, gastrointestinal disturbances and epilepsy are also frequently observed.[38,40]

BPD is defined as a group of brain disorders characterized by extreme mood fluctuations, ranging from manic to depressive episodes.[30] It is fairly well established that the BPD heritability is high, however, also due to the extreme phenotypic heterogeneity, the genetics of BPD has proven to be elusive.[41] Despite the failures of early linkage and candidate genes studies, multiple GWAS hits have been reported, as well association with some recurrent CNV loci.[41]

MDD is a mood disorder characterised by one or more long periods (at least two weeks) of decreased energy, mood and interest.[30] Both genetics and environmental exposures have been associated with MDD. In contrast with the other disorders described here, despite multiple reports from single studies over the years, no robust association with CNVs has been replicated across studies.[42]

SSD is defined as a continuum of schizophrenia-related phenotypes characterised by a collection of so-called positive (such as hallucinations and delusions) and negative (such as social withdrawal and poverty of speech) symptoms.[30,43] Genetics play an important role in SDD, as heritability is estimated from twin studies to be at around 80%.[44] SSD is generally not considered an early onset disorder, with the first diagnosis typically being in early adult life or late adolescence, although this view is constantly evolving.[45] Altaugh the consensus is constantly evolving, males are more likely to receive a diagnosis than females.[46]

Chromosomal Structural Abnormalities and Mental Disorders

The research in genetics and mental health are historically tied in many ways. The discovery of the link between chromosome 21 trisomy and Down's Syndrome in 1958[47] is the first reported disorder caused by a chromosomal aberration and marks the beginning of the research field of cytogenetics.[48] Schizophrenia is one of the most studied phenotypes when it comes to association to both sex chromosome aneuploidies (SCA)[49] and chromosomal aberrations[50]. The first reports on small clinical collections date back to the 1960s for SCA[51] and the 1980s for chromosomal abnormalities[52]. The associations between "Bipolar Affective Disorder" and chromosomal aberrations has also been explored thoroughly, with the first reports dating to the 1970s.[53] Finally, the association between SCZ and velo-cardio-facial syndrome (22q11.2 microdeletion) in 1999[54], can be used to mark the beginning of the interest into smaller rearrangements, what we define as CNVs, in mental health research.[12]

Methods

CNVs Detection

More than half of my PhD was spent developing methods to analyse CNVs and deal with the limitations of CNV calling from genotype data. Thus, in this section I will delve into the technical details of calling CNVs, starting with a brief historical excursus.

History of Structural Variants Detection

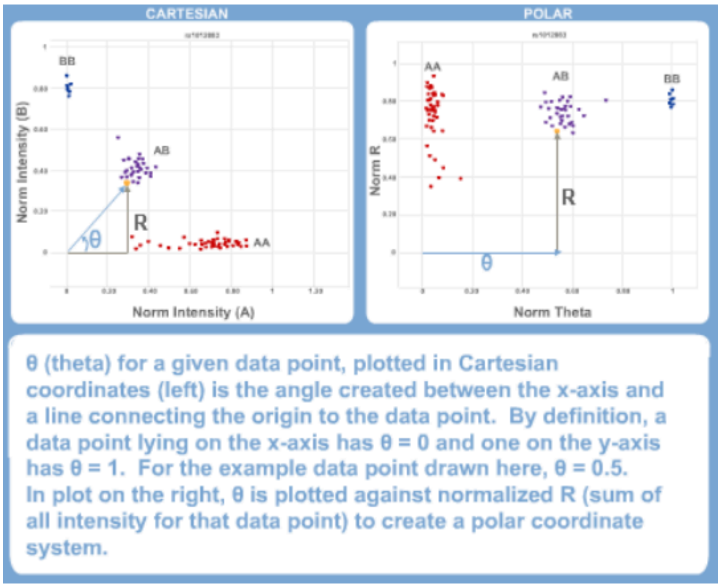
The history of CNV detection methods is reviewed thoroughly in Gordeeva et al.[55], here I report the main technological leaps. Researchers first encountered structural variants in the form of chromosomal aberrations, well before the completion of the human sequencing project. This was done using cytogenetics analysis, i.e. visualisation of stained chromosomes during metaphase. In fact, both discovery of chromosome 21 trisomy causing Down's syndrome[47] and the discovery of the extra chromosome X in males with Klinefelter's syndrome[56] date back to 1959. The development of DNA hybridization techniques, and later the development of fluorescent labels, instead of tritium-based radioactive ones, led to the development of FISH (fluorescence in situ hybridization) in 1977, a technique still in use today.[57] Arguably the greatest advancement that enabled the genome wide detection of what we today call CNVs was the comparative genome hybridization (CGH) technique[58], in 1992. In 1998, it was possible to detect genome wide sub-microscopic CNVs using array based CGH experiments[59], marking the beginning of modern-age array-based technologies. The sequencing of the human genome and the start of the GWAS-era led to the development of the oligonucleotide-based genotyping chip we know today, first by Affymetrix[60] and then by Illumina[61]. Finally, whole genome sequencing (WGS), first short and more recently long-reads, has been the latest leap in genomic analysis, including SVs and CNVs. WGS allows for an unprecedented precision in SVs boundary estimation, as well as the study of small and/or complex structural events that were previously inaccessible without the design of precise PCR experiments, or inaccessible to study at all. Very large collections of sequenced individuals are more and more common nowadays[62,63], however this technology still cannot be considered the golden standard for the detection of all SVs. CNV calling methods that rely on read depth (RD) changes (those used to detect "standard" CNVs[64]) seem to be differentially sensitive to deletions and duplications. Moreover for medium-to-large CNVs especially (> 50 kbp) read depth signal can be less stable than LRR.[65,66] Finally, microarray data is clearly more cost efficient (especially in terms of storage space) and, perhaps more importantly, is almost always available in very large collections of human samples as the first genetic data source.

CNV Calling from Genotype Arrays

While being based on different hybridization and reading technologies, both Affymetrix and Illumina arrays use the same starting measures for almost all CNV calling methods, log R ratio (LRR) and B allele frequency (BAF). In an Illumina genotyping experiment, these values are based on the light intensities for the two probes, a and b, measuring the two alleles, A and B (figure 2). LRR is a normalised measure of the signal intensity, thus it reflects the total amount of DNA. It is computed as

$$LRR = log_2\left(\frac{R_{observed}}{R_{expected}}\right)$$

where $R_{observed} = a + b$ for a given sample and $R_{expected}$ is computed from a panel of reference samples. SNPs in a region of normal copy number will have an LRR value around 1 ($R_{observed} \approx R_{expected}$), while it would be higher in the case of a duplication ($R_{observed} > R_{expected}$) and lower in the case of a deletion ($R_{observed} < R_{expected}$). The precise magnitude of the change (and thus the LRR value for the marker) depends on the actual copy number but also sample quality, genomic location and technical fluctuations. The BAF is the normalised measure of relative intensity between the two alleles (A and B). It is based on the theta values (θ, figure 3) of the three canonical clusters, AA, BB and AB. The formula (figure 3) is less obvious than the one for LRR, however, in practice it will be ~0 for the genotype AA, ~0.5 for AB and ~1 for BB. Similarly to LRR, the value can deviate from the theoretical value due to sample or probe quality, genomic location and technical imprecisions.



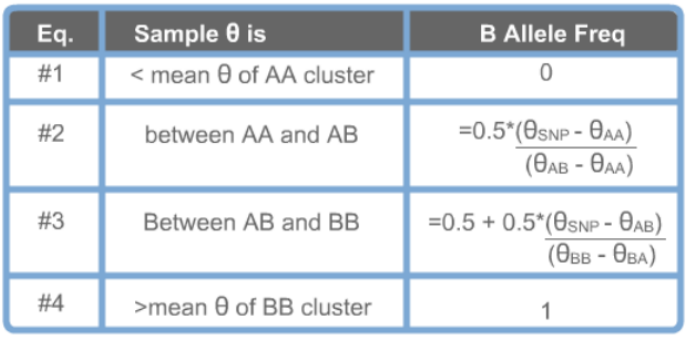
θ (theta) for a given data point, plotted in Cartesian coordinates (left) is the angle created between the x-axis and a line connecting the origin to the data point. By definition, a data point lying on the x-axis has θ = 0 and one on the y-axis has θ = 1. For the example data point drawn here, θ = 0.5. In plot on the right, θ is plotted against normalized R (sum of all intensity for that data point) to create a polar coordinate system.

Figure 2.

Schematic representation of the theta and R values for a biallelic SNPs in an Illumina genotyping array.

From https://support.illumina.com/training.html

The first CNV calling methods for array data (aCGH before genotyping chips) were mostly tailored to cancer studies. Researchers were more interested in performing a relatively rough segmentation of the genome into sections with the same copy number, with the expectation that this could be very high compared to normal tissue and “normal” CNVs. For this reason, as well as easier implementation, the first approaches relied only on the LRR, i.e. the DNA intensity, to call CNVs.[67,68] Shortly after, PennCNV was the first to include also the BAF into the models[69] and quickly became the golden standard in CNV calling from SNPs array data. Although comparisons studies have never been completely conclusive, it seems that HMM (Hidden Markov Model) based detection algorithms, such as PennCNV, performs better than others.[70,71] Of note, the research group that created PennCNV was also the first to provide a method to evaluate, and potentially mitigate, one source of noise in intensity data, the so called “genomic waviness” linked to GC content in the genome.[72] This measure is referred to as GC waviness factor (GCWF) and it is discussed in more details in the following section. At its core, PennCNV consists of a HMM (hidden Markov model) tasked with the challenge to predict the (hidden) copy number (CN) state of a given SNP marker based on the (observed) state of the previous one in the form of its LRR and BAF values. Consecutive groups of markers with the same predicted CN are then combined into CNV calls. In total, six possible states are defined: deletion of two copies (state 1, CN = 0), deletion of one copy (state 2, CN = 1), normal (state 3, CN = 2), copy-neutral loss of heterozygosity (LOH, state 4, CN = 2), single duplication (state 5, CN = 3), double duplication (triplication, state 6, CN = 4). Note that the CN reported are valid for autosomal chromosomes. This was an evolution from the simpler three states model used in the program QuantiSNP.[68] The definitions of the model emission probabilities for LRR and BAF and transition probabilities between states are beyond the scope of this thesis. However, it is important to notice that, while HMMs have been extremely successful models in computational biology, the implementation in PennCNV has one key limitation that is to essentially consider only a pair of SNPs at any given time. This makes it very sensitive to local noise, which can be quite high in genotype data.



Eq.	Sample θ is	B Allele Freq
#1	< mean θ of AA cluster	0
#2	between AA and AB	$=0.5^*\frac{(\theta_{SNP} - \theta_{AA})}{(\theta_{AB} - \theta_{AA})}$
#3	Between AB and BB	$=0.5 + 0.5^*\frac{(\theta_{SNP} - \theta_{AB})}{(\theta_{BB} - \theta_{BA})}$
#4	>mean θ of BB cluster	1

Figure 3.

MAF definition based on the theta values.

From https://support.illumina.com/training.html

The Signal-to-Noise Problem

Signal-to-noise ratio is pivotal in CNV calling, especially when the noise is non-random. With some simplifications, the noise in intensity data from genotyping arrays can be separated into two groups: random (affect each marker is independent independently of its position) and non-random (marker position is correlated with noise level). The random component is mostly caused by the fluctuation in light intensity for each probe; it can be due to processing (before reading the array) or by measurement (when actually reading the light intensity). In contrast, non-random noise has a relation with the SNP position in the genome. The main example is the so-called “genomic waviness” that is partly related to the GC content of a given section of DNA, but also to DNA concentration when preparing the genotyping experiment.[72] In the context of a PennCNV calling pipeline, noise is measured globally for a given sample using two measures: LRRSD (LRR standard deviation), and BAF drift. The GCWF (GC waviness factor) is also often considered when filtering samples in large cohorts as it should express the “global waviness” of a given sample. Interestingly, in the experience of our research group, an individual sample can have a relatively good quality measure (i.e. pass the QC), but still present an extremely wavy and noisy chromosome, suggesting that the noise magnitude can vary across different chromosomes (and/or that PennCNV QC measures are not as sensitive as we assume). In practice we noticed that, even after standard QC procedures (i.e. filtering bad samples, stitching CNV calls, and discarding excessively small or large ones), up to 50% of the CNV calls can be false positive in normal quality cohorts.[73] The proportion can be even higher in collections of bad quality samples (datasets relying on DNA samples of overall low concentration or otherwise compromised quality).

Visual Validation and its Limitations

As discussed above, CNVs calling from genotype arrays has some limitations. These include variable, and possibly poor, signal-to-noise ratio with noise patterns that can mimic, at least partially, true signal (such as increase or decrease of LRR due to waviness) and the intrinsic characteristics of the genotyping experiment and of the algorithm designed to detect CNVs. Regarding the genotype experiment these limitations include: a) the use of rare SNPs, this can lead to stretches of DNA where heterozygous markers are missing, strongly reducing the contrast of any CNVs in the regions; b) variable coverage and markers density, this limits the resolution of CNVs calling but can also make some regions completely inaccessible to CNV calling if too few SNPs are present. Some limitations of CNV calling algorithms have already been described in the previous sections. These can give rise to several problems, including a substantial false positives rate, as well as to partial calling (meaning that the true CNVs is larger than the CNV call) and fragmentation (meaning that the true CNV is splitted into multiple CNV calls). For this reason, software-based CNV calling used to be considered akin to a "discovery" step, and CNV calls were further validated using different methods. In the first publications[74–76] (and also in the Illumina manuals[77]), the manual inspection of the raw data trends (LRR and BAF) around the locus of interest or in the whole chromosome was considered routine. This is because true deletions and duplications give rise to very specific patterns in the raw data (figure 4) that are somewhat independent of the specific sample and locus characteristics, in a way that numerical measures (e.g. mean LRR inside the CNV call) cannot be. We refer to this process as visual inspection or visual validation. In the seminal studies a PCR experiment was also sometimes performed to validate the copy number and detect more precisely the CNV boundaries. This is still current practice in applications where precision is extremely important (e.g. diagnostic) and the signal from the chip is not clear enough.

Visual validation performed by a trained human analyst is an extremely powerful tool, especially when paired with the cleaning of the SNPs map by removing bad SNPs (in the context of CNV calling) such as very rare or non-bialleic markers. Such markers often don't behave as expected with respect to the CN, especially in the BAF, and add more noise than actual signal. However, the same factor that makes visual validation so precise and resistant to noise, i.e. the human analyst, is also the limiting one. In our experience, it takes between 5 and 30 seconds to inspect each putative CNV call. The time depends on aspects such as the noise level and type, the CNV size (larger CNVs are much easier to call reliably), the experience of the analyst, the locus, the type and consistency of CNVs (inspecting one specific locus per session is easier than a heterogeneous group of CNVs), the proportion of true CNVs and the interface used, only to name a few. Moreover, it is an extremely tiring process, especially if the noise is high, the true CNVs are a minority and the CNV calls are very heterogeneous. These limitations make visual validation feasible only for a small selection of loci, or samples. For this reason, most large studies have been focused on recurrent CNV loci, or specific genes, rather than the whole genome.

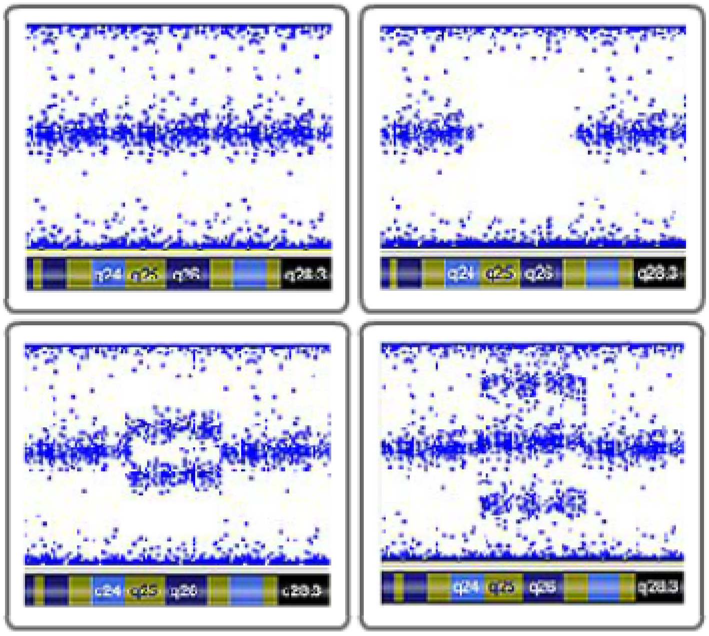


Figure 4.

BAF pattern for a normal CN (top left), a deletion (top right), a duplication (bottom left) and a triplication (bottom right).

From https://support.illumina.com/

training.html

Machine Learning, Neural Networks and Network Analysis Foundations

In the third manuscript we showcase a machine vision algorithm capable of performing the visual validation of CNVs, while in the fourth manuscript we describe its application in two large cohorts. In this section I introduce the key concepts behind the machine learning approach used. I also cover community detection in network analysis, the method we use to cluster CNVs into cohesive groups.

Brief historical perspective

Machine learning is a broad term that encompasses all those algorithms that rely on data to "train" their understanding of a given problem or task, rather than using fixed parameters set by the human programmer.[78] One of the simplest examples of machine learning algorithms is the logistic regression, while one of the more complex ones is the transformer architecture[79] behind the large language models (LLMs) like chatGPT and Google PaLM or text-to-image models such as Midjourney and Stable Diffusion. In an extremely simplified version, the history of deep learning starts with the first neural network, introduced by Frank Rosenblatt in his book Perceptron in 1958. The first proper deep learning algorithm came in 1965 and shortly after, in 1967, stochastic descent algorithm (a classic method to search for optimal weights value) was introduced. Then, 1970 saw the development of the back propagation algorithm (the methods that enable a neural network to "learn" in a computationally feasible manner), and 1982 its modern implementation. Finally in the early 2000s, deep learning as we know it was born, with ever increasing dataset and model size. See Schmidhuber (2022) for a more in depth "Annotated History of Modern AI and Deep Learning"[80], and the book "Deep learning"[78] as reference on the key topics and algorithms as well as the maths behind them.

Inference vs Classification

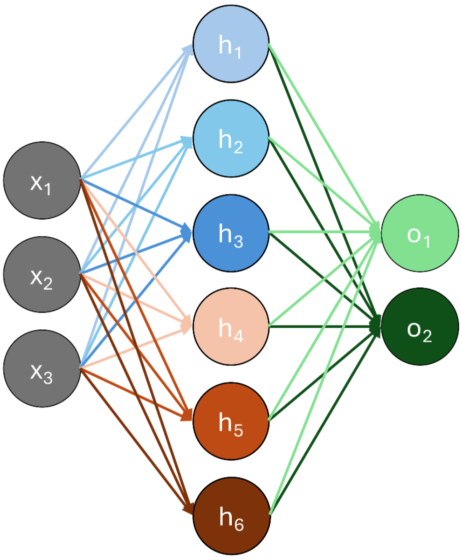
Most uses of statistical/machine learning methods fall into one of two large categories: solving an inference or a classification problem.[78,81,82] While parsimony, simplicity and interpretability are very important in an inferential model, in predictive models the only goal tends to be maximum accuracy. As an example, in line with this general consensus, in the second paper included in this thesis we use logistic regression to analyse the risk conferred by some specific CNVs to mental disorders where the beta of the models are interpreted as risk. In contrast, in the third paper we used a Convolutional Neural Network to categorise CNV calls good as either true or false. In this application the interpretability of each weight inside the model was not relevant, while maximum accuracy was the main goal.

Neural Networks Nomenclature

Figure 5 showcases the simplest implementation of a neural network, a three layer fully connected NN, consisting of three, six and two nodes respectively. The three layers consist of the input layer (where data from the world is fed to the model), a hidden layer, and an output layer. If we imagine it as a classifier (for example to recognise a specific object in an image), the two nodes of the output layers represent the two classes the model can discriminate (such as "there IS a cat in the picture" and "there IS NOT a cat in the picture"). From one layer to the next, all nodes are connected to each other (fully connected). Each connection has a weight (w_{ii}^{i}) and each node has a bias (b_{ii}), and the value for a node is given by:

$$h_1 = f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} w_1^1 \\ w_2^1 \\ w_3^1 \end{bmatrix} + b_1\right)$$

where f is some non-linear function (in modern implementations). The formula for the full layer reported in figure 5 is often seen in simplified form $f(W \bullet X + B) = H$ where the capital letters represent the full matrices for the weights, first layer, biases and second layer.



$$f\left(\begin{vmatrix} w_1^1 & w_1^2 & w_1^3 \\ w_2^1 & w_2^2 & w_2^3 \\ w_3^1 & w_3^2 & w_3^3 \\ w_4^1 & w_4^2 & w_4^3 \\ w_5^1 & w_5^2 & w_5^3 \\ w_6^1 & w_6^2 & w_6^3 \end{vmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}\right) = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix}$$

Figure 5. Schematic representation of a simple fully connected three layer neural network on the left and algebraic representation of the computation to get the values for the hidden layer from the input values. Montalbano S., own work 2025.

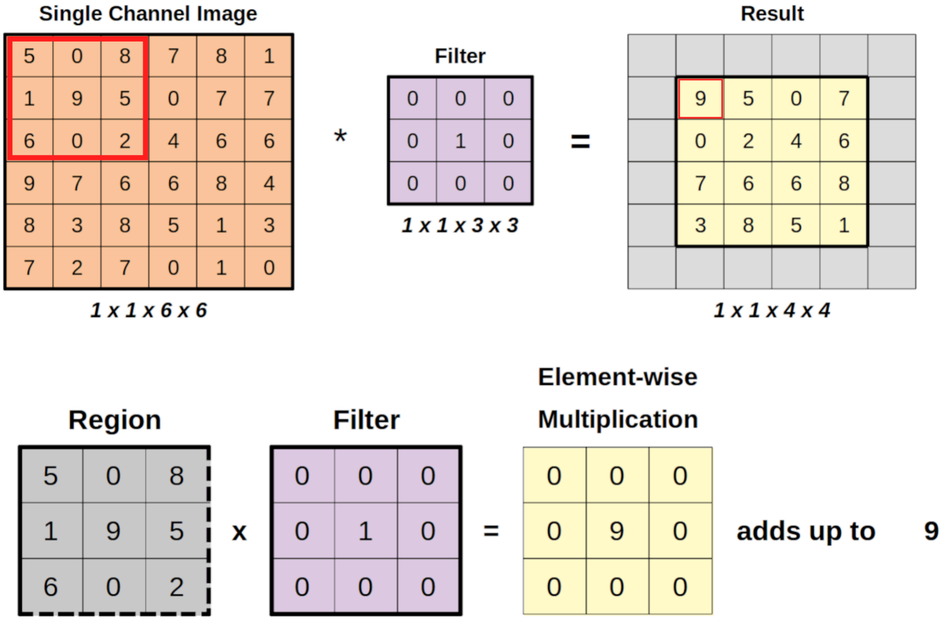
Neural Networks and Deep Learning

The concept of a neural (or neuronal) network (NN) is surprisingly old.[78] Earlier approaches were partially inspired by how actual biological neurons were thought to function (a neuron integrates signals from different sources and "decides" to fire or not), thus the name. However, the emphasis on this similarity has lost popularity as the field developed, and it is basically absent from modern day implementations. The first neural networks were simple, and they did not *learn* in the proper sense. An example is the "Mark I Perceptron", an analogical machine built in the 1950s designed for image recognition, consisting of three layers of interconnected units or nodes, already resembling in structure a modern neural network: an input layer of 20x20 (400) photocells (pixels), a hidden layer of 512 units and an output layer of 8 "response" units. The connections between units from one layer to the next had weights, i.e. parameters that were multiplied with the input to compute the output. In stark contrast with modern implementations, the weights from the first to the second layer were set randomly, while the one from the second to the last could be manually tuned. Arguably, the main innovation that makes modern NNs capable of "learning", is not the "network" design, but rather the backpropagation algorithm, i.e. a computationally feasible implementation of the chain rule for differentiating a function, and is at the heart of how a modern NN is tuned to solve a specific task.[83]

The main limitation of NNs that slowed their applicability before recent times is that they require large scale to reach good performance, and thus large amounts of computation power and time to be trained and run. Conveniently, in the last 10-15 years, faster and faster methods have been developed, also taking advantage of the similarity between the operations required to train and run a neural network with those required by modern graphical applications such as 3D videogames, that is linear algebra (vector matrix multiplications in particular). For this reason it has been possible to use the already extremely specialised architectures of GPU to train and run deep learning algorithms. The architecture of modern neural networks has also evolved from the early perceptrons, most notably non-linear activation functions have been introduced, such as the ReLU (rectified linear unit).[84] Interestingly, the ReLU function is actually similar to how a neuron works, somewhat reviving the biological analogies of the pioneers.[85]

Deep learning algorithms are known to be prone to overfitting, that is, learning patterns specific to the training data but failing to generalise out of sample. There are multiple methods to counter this, including early stop (interrupting training when the accuracy improvement starts to flatten, ideally before overfitting occurs), dropout (excluding some nodes at random from a given training iteration, forcing the model to not rely on very specific internal patterns), and data augmentation and noise robustness (input data is manipulated in such a way to create more and more varied examples).[78]

Regarding the implementation of deep learning algorithms, well-established libraries exist, two great examples being TensorFlow[86] and Torch[87]. Moreover, human friendly wrappers such as Keras[88] and R-torch[89] provide high level functions to build modern and quite advanced deep learning algorithms without the need to understand all the maths behind them.



5	0	8	7	8	1
1	9	5	0	7	7
6	0	2	4	6	6
9	7	6	6	8	4
8	3	8	5	1	3
7	2	7	0	1	0

0	0	0
0	1	0
0	0	0

	9	5	0	7	
	0	2	4	6	
	7	6	6	8	
	3	8	5	1	

5	0	8
1	9	5
6	0	2

0	0	0
0	1	0
0	0	0

0	0	0
0	9	0
0	0	0

Figure 6. Schematic representation of one convolution operation on a single channel image. The orange matrix is the input image with each square being a pixel, the purple matrix is the filter or kernel, and the yellow matrix is the output. The bottom row exemplifies the actual computation for the first pixel of the output image, highlighted in red. From https://github.com/dvgodoy/dl-visuals , CC-BY.

Machine vision and the convolution operation

Different families of NN architectures have been developed to tackle families of problems. Two classic examples are recurrent neural networks for handling sequential data[90], and convolutional neural networks (CNN) for machine vision[91] and in general any problem that can be represented as an image (e.g. sound processing[92]). A typical CNN consists of two main parts, a feature detector (performing the actual convolutions) and a classifier, usually in the form of a fully interconnected NN of varying depth depending on the task. Such NNs are called "convolutional" because they rely on an operation that is conceptually analogous to one interpretation of the mathematical operation called convolution. In machine learning the convolution operation is extremely powerful in detecting simple features in an image (e.g. edges) that are then combined in order to detect more complex shapes, such as a cat. Figure 6 shows a simple practical example; a convolution occurs between the input matrix (in the first layer, the original image) and a smaller matrix, called filter or kernel. The result is a smaller matrix, where each pixel is the sum of the element-wise multiplication of the input with the kernel. In practice the filter defines what the model is "looking for", for example the following kernel will screen the image for vertical edges (the results will be high when combined with a picture of a one pixel vertical line).

$$\begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

As shown in figure 7, a typical CNN input is an image, usually single-channel (gray scale) or three channels (RGB color image). In each layer all channels are processed with different filters, thus as the image progresses into the CNN it loses resolution but grows in number of channels. At the end, each channel should represent one or more features the model defined as important for the given task. As with weights and biases, the filters are learned features that are adjusted during training. The final layer of a CNN is usually then used as input for a "regular" fully connected NN that acts as a classifier.

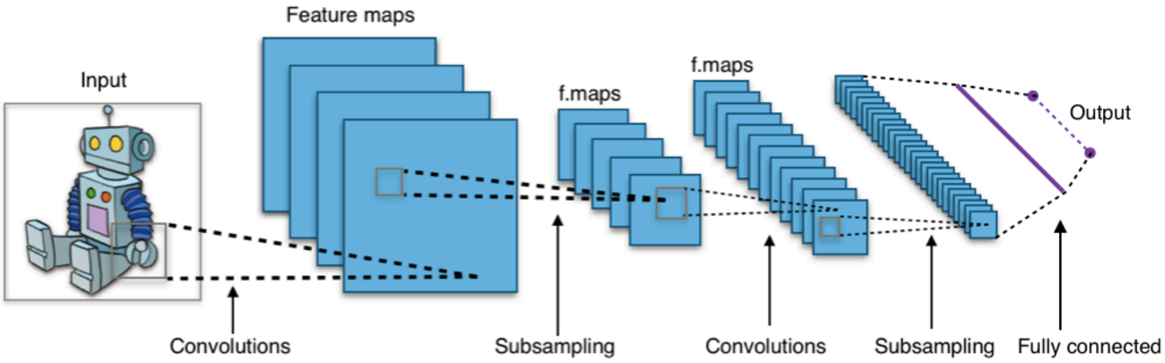


Figure 7. CNN architecture example. In each layer the size of the "image" gets smaller but the number of channels grows. By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374

Network analysis foundations

Network analysis is a field of studies and methods that focuses on the relationships between different entities. Classic applications include social relationships between humans[93] and animals, biological relationships between proteins[94] and groups of interconnected machines such as the modern-day internet[95]. The key concepts are the node, the unit or individual entity in a network, and the edges, the connections between two nodes. The connections can be simple (they can be represented as a boolean, true/false, 0/1) or weighted (each edge has a value representing the strength, usually scaled to the interval [0,1]). Usually the main focus in a network regards the nodes, in particular their degree (meaning the number of connections a node has) as well as their position and importance in the network, described with a measure defined as "centrality".[96] However, measures to characterize the importance of an edge have also been proposed (such as its "gravity"[97]). A classic interest of the field is to detect communities, i.e. distinct subgroups of nodes within a larger network. Following the examples above, these can be social groups, proteins in the same pathway, or computers of a company office. In the work included in this thesis, community detection algorithms are somewhat naively used to solve a clustering problem that is computing CNV regions (CNVRs). Given a set of CNVs, we define the CNVR as the smallest set of internally homogeneous groups that can describe the whole set. To solve this problem we decided to use network analysis and community detection algorithms to make sure that all CNVs are treated in the same way. In practice, visualising a set of mutually overlapping CNVs as a network defined only by their connections and the strength of those is an approach with far less assumptions than screening pairs of CNVs for similarity.

Datasets Used

iPSYCH2015

The iPSYCH (Lundbeck Foundation Initiative for Integrative Psychiatric Research) 2015 case-cohort collection[98] refers to the union of the original iPSYCH 2012 sample[99] and its subsequent expansion, known as iPSYCH2015i. The numbers here refer to the full iPSYCH2015 sample. The collection has a unique composition. The base population consists of all people born in Denmark between the 1st of May 1981 and 31st of December 2008 from a mother with a Danish CPR (civil registration number). From this population, all persons that received a major psychiatric diagnosis from the public healthcare system have been included in the study, for a total of 92,531 individuals. Then, a population-representative cohort of 50,615 samples have been randomly selected from the entire population. In total 3,030 individuals from the case group have been also selected in the cohort group. The case counts and ICD-10 definitions for the main diagnosis considered in this thesis are as follows: SSD (ICD-10 F20–F29; n=16,008), MDD (ICD-10 F32–F33 and ICD-8 296.09, 296.29, 298.09, and 300.49; n=37,555), ASD (ICD-10 F84; n=24,975), or ADHD (ICD-10 F90; n=29,668). By construction one sample might have multiple diagnoses. Biological samples were obtained through the Danish national neonatal screening biobank, where samples from virtually all children born in Denmark since 1 May 1981 are stored at − 20 °C in the form of dried blood spots. Genotyping was performed on amplified DNA using Illumina PsychArray for the original 2012 cohort, and on non-amplified DNA using Illumina GSA arrays for the 2015i expansion. Aside from possible ethical concerns regarding the uncommon "implicit consent model" used for the creation of the study, the main limitation of iPSYCH is the high level of noise in the genetic data, especially in the measures used for CNV calling, LRR and BAF. This is most likely due to the nature of the sample used, i.e. dried blood spots. Moreover, for the 2012 portion the genotyping array is also suboptimal for CNV calling. The PsychArray from Illumina was designed to be enriched in very rare markers discovered in the first generation of large GWAS, however, similar to the CNV markers, these proved to be of little utility and they mostly added noise to the data.

UK Biobank

The UK Biobank (UKB) is a large prospective study based in the United Kingdom, consisting of over 500,000 participants.[100] The study includes very rich phenotypic and genetic data, including genotype, brain scans, blood markers, linked medical records, among others.[101] Genotype was performed using the UKB Axiom genotype array for the majority of UKB samples. Recent additions include WGS[63], blood proteins levels[102] and a mental health questionnaire.[103] UKB partitecipants are not representative of the general population, in contrast a clear "healthy volunteer bias" was found, with participants being less likely to, among others, smoke, be obese, consume alcohol daily.[104] Despite this, it has been shown that findings in the UKB can be generalised to the general population.[104] However, the intrinsic bias in the population included remains perhaps its strongest limitation. Thanks to its open design, UKB has become a central resource in biomedical and genetic research since its original launch in 2012, and today it is used by thousands of research groups around the world. In the context of the studies included in this thesis, UKB was mostly used as a large reference dataset that is potentially open to the entire research community, and thus perfect for training and validating a new method. Concerning CNV calling, UKB data quality is overall very good. The noise level is low for most samples, the array coverage is quite uniform and the majority of SNPs can be used for effective CNV calling. However, when PennCNV is applied on Affymetrix-type genotype arrays it is known to produce less CNV calls compared to Illumina chips.[105] It is not clear if the missing calls consist mostly of false positives, if the CNV detection power of Affymetrix arrays is actually lower, if some part of the PennCNV implementation is too strictly tied to Illumina design since they were the intended target of the original implementation, or a mixture of these issues.

Icelandic biobank at deCODE genetics

Originally founded in 1996, deCODE genetics (subsidiary of Amgen from 2012) has been pioneering in many aspects of population genetics and genomics. During its almost three decades of operations the company collected a large sample of Icelanders, now just over 190,000, that has been genotyped across multiple generations of Illumina arrays[106]. Notably, researchers at deCODE recognised the advantage of a small population with a clear genealogy available such as the one of Iceland, and were able to implement an extremely accurate imputation method based on the sequencing on a small subset of samples.[106] With respect to the main overarching focus of this thesis, the Icelandic biobank at deCODE possesses two extremely useful characteristics. The first is the variety in terms of genotyping chips used. Almost 20 different SNP arrays from Illumina (which also co-owned the company deCODE for a period between 2009 and 2012) have been used on at least 500 samples each, from virtually all generations of Illumina arrays. Moreover, some individuals were genotyped on multiple different chips and can be used as technical replicates (with some caveats, as some were re-genotyped for quality issues). The second useful feature that this biobank offers for our studies, is the already mentioned extensive genealogy that is available for each sample. Genotyped individuals span multiple generations, increasing the number of available trios that allows for orthogonal validation methods that are based on inheritance patterns.

Summary and discussion of the four presented manuscripts

I: Accurate and Effective Detection of Recurrent Copy Number Variants in Large SNP Genotype Datasets

Background

As already introduced, recurrent CNV loci have been a key area of interest of genetic research, especially for syndromic disorders, psychiatric, and neurodevelopmental diseases, for a long time. From microscope based karyotyping to modern large scale biobanks, the technologies to detect CNVs, and structural variants in general, have vastly evolved, together with the size of the datasets. SNPs genotyping, usually on Illumina or Affymetrix arrays, still remains the standard for very large biobanks, even though large collections of WGS data are becoming more common. Even though CNV calling using PennCNV in a handful of samples is not a complex bioinformatics task, this is not the same on the scale of hundreds of thousands of samples. Here, automated QC steps and good practices become a necessity. Most research groups with a history in CNV calling have their own internal pipelines, scripts, and helper programs, however these are rarely made public in easy-to-use packages. Ultimately, a simple but complete protocol for calling, and more importantly, processing CNVs calls, did not exist. Moreover, several steps (such as SNPs filtering, minimum number of markers, CNV/locus overlap, quality checks after visual validation etc.) require expert knowledge and are often not extensively reported in large publications. For this reason, in the first year of my PhD study I developed a standardised protocol to perform CNVs calling on modern, large-scale collections of SNP genotyped samples, condensing the knowledge and common practice of the field, of which our research institute and close collaborator such as deCODE genetics, are main contributors. This is the technical backbone of several published and ongoing projects from our group.

Aims

The main steps of the protocol reflect the major limitations of detecting recurrent CNVs in a large collection of human samples:

- **SNPs filtering**. Not all SNPs are equally informative, some (such as the very rare ones) tend to add more noise than signal. As already mentioned, PennCNV is very sensitive to local noise, so there is a balance in trying to use as many markers, and introduce as little noise as possible.
- **HPC-ready PennCNV pipeline**. Not all researchers have expertise in handling large datasets and effectively using job schedulers. We provide a slurm and PBS compatible, start-to-finish PennCNV pipeline.
- **CNVs processing and filtering**. Raw CNVs from the PennCVN pipeline need to be processed, imported into R and filtered before visual inspection. The protocol gives the users a strong base to follow, but also allows them to skip certain steps and change filtering parameters. We also implement a refined filtering tree, especially useful in loci with low signal to noise ratio (i.e. a lot of putative CNVs, but few true carriers). The

objective is to minimise the number of CNVs a human analyst has to inspect, without losing any possible true carrier.

- **Visual inspection graphical interface**. Once a list of good putative carriers for the desired loci have been obtained, these need to be manually validated by a human analyst via visual inspection of the raw data trends (LLR and BAF) within and around the locus of interest. Depending on the amount of calls to process, different groups use simple methods (such as saving the plots manually and annotating the results in a worksheet) or more refined custom graphical interfaces. These more refined programs are often not publicly available, or compatible only with very specific workflows. For this reason we publicly released our internal solution with the protocol, and then updated it to a more modern solution (https://github.com/SinomeM/shinyCNV) easier to use and to maintain.
- **Quality assessment of the results**. With the list of validated CNV carriers in the loci of interest, we suggest some plots that can help diagnose potential issues in the analysis. The main focus of this step is to ensure that no true carriers are excluded from the putative list due to excessive filtering.

Results and Discussion

CNV calling from SNPs array data pipeline design, limitations, and challenges, have remained substantially unchanged since the introduction of PennCNV in 2007. Despite this, it can still feel more of a craft than a science. This is partially due to the fact that a plethora of different genotype chips from multiple manufacturers exists. Chips from a certain generation will often share a big portion of the “backbone” markers set, but across generations this is not always the case. Moreover, the number of markers might vary substantially across different chips, and even when two chips have a similar number of SNPs, the coverage in a given region can still differ heavily. Unpredictable and potentially very strong noise patterns do not help with standardisation either. In this protocol we try to standardise our own internal pipelines and programs, as well as to extensively discuss the reasoning behind each step and parameter. Since the pipeline was originally developed for the iPSYCH dataset (see relevant section in “Datasets”), two key goals of the process are to increase the signal to noise ratio and to integrate data coming from different chip types.

Admittedly, this is a kind of protocol that is very hard to standardise across research groups. A group routinely processing large quantities of data will already have an internal pipeline, and on very small scales a modern pipeline might not be needed in the first place. Nonetheless, it was pivotal for the development of our institute research projects, and the standardisation in file format and R functions served me as a base on which to build all the rest of the CNV infrastructure, such as the automated CNV validation tool presented in the third manuscript. We hope at least some portions of it (e.g. the visual inspection R-shiny app, which is completely independent of the protocol), will be of use to the larger field in the future. Finally, while the primary focus of the pipeline are CNVs in recurrent loci, the pipeline is not limited to this and can be used, e.g. by researchers interested in multiple CNVs from a small set of samples.

II: Analysis of exonic deletions in a large population study provides novel insights into NRXN1 pathology

Background

Neurexins are a highly conserved gene family that encodes for transmembrane proteins, and are involved in the development and function of the synapse.[107,108] Deletions in this gene have been associated with multiple mental disorders including SSD[109], ASD[37] and ADHD[35]. The *NRXN1* gene is a well-known hotspot for non-recurrent CNVs. Meaning that despite CNVs are often observed in the locus, they are not due to LCR regions, and thus the breakpoints do not follow a predictable pattern. CNVs in the region are considered to be most often sporadic. However, inherited events have been observed, and at least one CNV is known to segregate in the population. CVNs do not affect the gene randomly either: the 5' region of the gene (where the promoter of one the two main transcripts, alpha, is located) has been reported to suffer more CNVs than the rest of the gene. It is not clear whether it is because CNVs in the other parts of the genes are under stronger selection, less likely to occur for some biological reason, or both.[110] Notably, initial association studies were based on single case studies or case cohorts[74,111–113], while subsequent larger efforts were based on highly selected samples (very severe cases, and very healthy controls)[37,112,114,115] and, moreover, intronic deletions as well as all duplications were always discarded from the analysis.

Aims

This project has two sets of objectives: provide prevalence and risk estimates of *NRXN1* deletions, as well as experiment with the analysis of non-recurrent CNVs in a large cohort.

- **Precise, population based estimates**. In line with previous publications from our group and others, our first goal for the project was to provide population-based estimates of the prevalence of CNVs in the neurexin locus, across multiple categories. Due to low carriers count, we had to exclude duplications from the study, however we were able to describe both exonic and intronic deletions. Despite its simplicity, this step is of extreme importance as it sets the baseline for any association with phenotypes. Similarly to most recurrent CNVs, deletions in the NRXN1 locus are thought to be extremely rare in the population, but it was estimated from small and selected control groups.
- **Grouping of heterogeneous CNVs**. Given the heterogeneity of deletions in the locus, we experimented with methods of grouping similar variants together, while retaining high internal homogeneity within groups. We used two orthogonal approaches. A correlation matrix between deleted exons created good groups for use in association, but could not be used on all deletions. In contrast, a similarity matrix using the IOU (intersection over the union) between every pair of deletions was capable of examining all deletions together but produced less clear clusters. This approach would eventually be expanded and refined in the genome wide version of the CNVRs computation algorithm (in the following project, see Manuscript III).
- **Risk of mental disorders**. Finally, we combined the unique design of iPSYCH 2015 with the characterization of the subgroups in the locus to investigate the association between

deletions of the *NRXN1* gene and mental disorders. Our original goal was to properly analyse all CNVs in the locus, both deletions and duplications and regardless if they would hit an exon or not, and also try to pinpoint the precise location of the risk association as much as possible. Unfortunately, we eventually had to drop duplications from the analysis due to low carrier count.

Results and Discussion

In this paper we were able to characterise the deletions affecting the *NRXN1* gene locus, their prevalence in the Danish population, and their association to the five core iPSYCH mental disorders (ASD, ADHD, SSD, MDD, BPD). The major limitation of the study was unfortunately the sample size. Despite being considered a hotspot for non-recurrent deletions, CNVs in this locus are still overall a rare event in the population. For this reason we could not reliably include duplications in the analysis, as well as define smaller subgroups of exonic deletions, despite some indications from the data. Nevertheless, we find exonic deletions to increase the risk for ASD and ADHD, consistent with previous reports. Our risk estimates are lower compared to past research, however this difference is not statistically significant. We also find that the risk is mostly confined to one portion of the gene (the alpha promoter region), where deletions are more prevalent. Earlier studies already reported deletions in the other portion of the gene (beta promoter region) to be rarer, and suggested a possible embryonic/fetal lethality as the cause.[110] However, we do find some occurrences in both cases and controls, making this hypothesis unlikely. Notably, we do not find exon-disrupting deletions of the *NRXN1* gene to be associated with an increased risk of SSD, which appear to be in strong contrast with previously reported odd ratios in the range of almost 10. However, as discussed in the paper, we believe this to be a compelling example of multiple problems in association studies based on SNP genotyped cohorts. First, most of the studies after the original report (i.e. Rujescu et al.[112]) consisted of a simple Fisher exact test on the samples pooled with all previously available ones, both cases and controls. Critically, this led to 40-80% of the controls being from the original sample. This approach creates a large opportunity for batch effect as the sample from Rusjecu was genotyped on HumamHap300 arrays (a rather sparse chip), while new cases were genotyped on denser arrays. Thus not only the genotyping chip is an unaccounted covariate, but also cases were more likely to have a CNV detected by design. This is because the deletions in the locus are relatively small and thus the CNV calling precision is very sensitive to increases in the markers coverage. When looking at more recent studies where: 1) the original control sample from Rujescu et al. was not used, and 2) a proper meta-analysis across genotyping chips is performed, the risk estimate is much lower than the original report. In this new light, our results on SSD are not as surprising, while still being in contrast with previous reports. Finally, we also explored the impact of intronic deletions. We find a small segregating deletion, with a frequency of ~0.08% in the general population (as previously reported[112]) to be potentially associated with an increased risk of SSD, however the association did not survive multiple testing correction. Nevertheless, we believe this should be used as a suggestion for future similar studies to not discard non exonic variants by default.

III: CNValidatron, automated validation of CNV calls using computer vision

Background

Despite the growth in the last decade in larger WGS biobanks and collections, SNPs genotyping still remains a cornerstone of genetic data for large human cohorts. Intensity data has been used to detect structural variations such as CNVs since before the advent of modern genotyping chips, and genotype data remains, arguably, the most reliable data sources to detect CNVs in the medium-to-large range (from ~50kbp to several Mbp). Since its publication in 2007, PennCNV[69] has been the most popular program to detect CNVs from SNP data. The field had to deal with noise since the early days, only one year after the first publication Wang group published a methods that attempt to attenuate the so-called GC waves, i.e. wave-like noise patterns in the intensity values of the markers in parts of the genome that are thought to be partially due to GC content.[72] Moreover, when the number of samples analysed grew substantially researcher soon realised that raw CNV calls were often unreliable, and the field moved to manually validating every call, visually inspecting the intensity and allele frequency trends of the markers in the locus of interest. Understandably, this approach restricts studies to either relatively small sample sizes, or to a selection of few specific genomic loci. Other approaches include using more than one method to detect CNVs and joining the call sets, or applying a set of quality filters on the samples and/or CNVs calls included in the study. However, these strategies often assume (more or less implicitly) that false positives are equally distributed both across the genome and among cases and controls, that is not necessarily a fair assumption. In this manuscript we present our solution to the false positive problem in CNV calling from SNP array data. We believe the visual inspection performed by a human analyst is not trivial to translate in a pure analytical/mathematical approach, mainly because what is the same overall pattern for a human (a true deletion, a false call due to noise, etc) can be very different in terms of, e.g., mean LRR or proportion of heterozygous SNPs, even when considering ratios between the CNV call and the surrounding region. For this reason we use a CNN, a neural network specifically designed for image recognition, to automatically perform the same visual validation a human would. We train the model on a large set of human validated CNV data, based on two large cohorts, UKB and deCODE, and we validate its accuracy across different arrays from both Affymetix and Illumina.

Aims

The project behind this paper had the following major aims:

- **Characterise the magnitude of the false positive in CNV calling**. It is fairly well established in previous literature that false positives can be a serious problem in CNV calling from SNP arrays. Moreover, it is also accepted that the extent of the problem is strongly correlated to the noisyness of the samples analysed. However, a thorough characterisation in a large and modern cohort is still missing, thus we made it the first objective of the project, also given that it can be a byproduct of creating a large set of human validated CNV calls.
- **Characterise the extent of false negatives in CNV calling**. In contrast with false positives, false negatives are far less considered and described in literature, as expected. Having access to possibly one of the largest collections of genotyped trios (deCODE), we could

estimate the lower bound for the false positive rate, using putative de novo CNVs in an offspring and checking for a missing PennCNV call in either one of the parents.

- **Create a machine vision software capable of validating CNV calls**:
  - **Create the best image representation**. Image features and formats that are best suited for human analysts are not necessarily the best choice for a machine. We experimented with different views and resolutions to enhance the more important information and attempt to mimic the visual process done by the human analyst.
  - **Define the model**. Given the relative simplicity of the task (small number of classes, low resolution) the model choice was not particularly critical. We used the CNN architecture as it is designed for image recognition. We tested different configurations and, given equal performance, we chose the simplest model.
  - **Validate the trained model in and out of sample**. Since deep learning models are known to be prone to overfitting, we wanted to have a proper validation set, possibly on a third dataset. For simplicity, we relied mostly on deCODE data (same cohort but different samples from the training data), however it is worth noting that deCODE samples are genotyped on more than 20 different SNP arrays.
- **Provide the software as an R package to easily use the model on new datasets**. The model was built in R and is made available as a functional R package.

Results and Discussion

In this paper we describe the in depth characterisation on PennCNV calling results in two large collections (UKB and deCODE) genotyped on multiple different genotyping chips from the two main manufacturers, including a total of 17,000 unique samples between training and testing sets. Across training and testing sets, 22,138 CNVs were validated by a human analyst or a combination of genealogy, prediction and human analyst. We find that, depending on the sample, from 30% to 60% of the PennCNV calls are false positives. Moreover, by using the full trios in the deCODE set (1200 trios), we were also able to report a preliminary false negative and de novo rate, at 1.5% and 3.5% respectively. Since the original submission of the pre-print, we have expanded the set of trios to 6000 and these estimates and the estimates remained stable. We were able to train a model that is capable of predicting the state of a CNV call (true deletion, true duplication, false positive) with an accuracy of ~95%. Using this large set of manually validated CNV calls we were also able to study how different QC metrics affect the false positive rate, and how true and false positives differ in their genomic distribution. The main limitation of the study is the lack of proper out-of-sample testing. Our main test set is in deCODE, where a portion of the training data comes from, even though it consists of the smaller fraction. This limitation is partially offset by the fact that samples in deCODE are genotyped on a lot of different chips (see also the fourth manuscript) and, even though included trios were restricted to a subset of all possible arrays, the study still include samples from more than 10 different chips. The second limitation is the small sample size used to estimate false negatives and de novo rates, however we already increased the number of trios where the disagreements between the model and the genealogy were manually validated by a human analyst to 6000 and will update the results in the published version of the manuscript. Finally, two intrinsic limitations to this whole approach is that the model relies on PennCNV to define “where to look” and that the training is based on validation data produced by a human. We believe that the second point is not a strong limitation, especially considering that we are also using a completely independent validation in the form of the Icelandic genealogy, to estimate the accuracy. The first point is more complex. While on one hand the limitations of PennCNV are the very reason this project was conceived, on the other hand we felt that designing a novel CNV calling algorithm from scratch was beyond the scope of the project.

IV: A genome-wide characterisation of large Copy Number Variations in two population-scale datasets genotyped on different SNP arrays

Background

Both larger CNVs (such as recurrent ones) and, more recently, smaller structural variants have been thoroughly explored in increasingly larger cohorts. However, rare genome wide CNVs (defined as deletions and duplication in the size range of 50kbp to 10Mpb) have been understudied. This is mostly because of the imprecision of the available calling methods, that make aggregate burden studies the only feasible approach (not without its own limitations). We previously described our approach to scale visual validation of CNVs to millions of calls using machine vision. Here we apply this approach to two large cohorts, the UK Biobank and the Icelinc biobank at deCODE genetics.

Aims

The overarching goal of this project is to set a baseline of what we can and cannot do with automatically validated genome wide CNV calls. The main aims can be summarised as follows:

- **Apply the automated CNV validation at scale**. We showed the accuracy of the CNValidatron software in a relatively small subset of samples. However, applying it to the two entire cohorts is not trivial. Consequently, the first goal was to set up a small pipeline to effectively run CNValidatron on large samples. This will be available as a separate repository.
- **Characterise the genomic distribution of CNVs**. With a set of validated CNVs, we wanted to explore their genomic distribution. Some of the research questions included: Are CNVs equally spread across the chromosomes or some regions are more enriched/deprived? Is any feature associated with CNV density (e.g. distance from telomeres)? Do deletions and duplications behave in the same way?
- **Apply the CNV grouping and analysis at scale**. We showcased the CNVR algorithm in the previous paper on a small set of CNVs, here we applied it to the full dataset. We also showcase other types of CNV markers that can be used for different analysis with the objective of describing advantages and disadvantages of each.
- **Measure if a locus has more or less CNVs than expected**. With multiple effective markers to analyse CNVs the natural question was "Are there more CNVs than expected by chance in this marker (e.g. a gene)? How can it be measured?". We developed an analysis that can measure the fold change with respect to a background value and two orthogonal simulation approaches to set this background.
- **Compare two large cohorts**. One of the main objectives of the project was to test whether and how different samples can be combined into a single analysis. We extensively compared the two cohorts on multiple measures, including number and type of CNVs, CNV distribution and frequency. We also attempt to disentangle differences due to population structure and genotyping chips.
- **How are genes affected by CNVs?** Finally, given the nature of CNVs, we placed special

attention to how CNVs affect genes. We looked at CNV enrichment across different gene groups, including gene constraint and conservation. Finally we also check whether any CNV polymorphisms are present in the two populations, i.e. genes that are duplicated or deleted with high frequencies (>1%).

Results and Discussion

We applied our method for automated CNV validation to two large human cohorts, UKB and deCODE, in what is, to our knowledge, the first study of such scale. We explore different ways to analyse CNVs effectively depending on the goal, using regular windows and genes as CNV markers, as well as computing CNV regions (CNVRs). We investigate the distribution of deletions and duplication in the genome, showing how it behaves unevenly with a clear hotspots pattern, meaning some specific loci in the genome show a high accumulation of CNVs compared to the rest. We also replicate previous findings of higher rates of CNVs towards the telomeres and centromeres. Notably, we find the CNV distribution to be different also across the two datasets. Although we could not exclude differences in population structure, it seems the different genotyping chips are the main cause. This finding is of critical importance in the context of large studies that combine data from multiple sources. Finally we investigate how CNV affects genes in multiple ways. We integrate constraint and conservation scores and we define a “genomic enrichment score” capable of measuring if a given gene is enriched or deprived of CNVs compared to a random background.

At the moment, this study has some limitations. First of all our selection of transcripts limited our gene set to essentially only protein coding ones. This in turn made the analysis on gene sets quite limited in scope and results. Moreover, the differences in CNV distribution across cohorts prevent us from effectively analysing the combined dataset, limiting our power, especially on the rare side (such as for detecting genes that are significantly deprived of CNVs). These limitations (and more) will hopefully be addressed in the published version of the manuscript.

Bibliography

1. Mullaney, J. M., Mills, R. E., Pittard, W. S. & Devine, S. E. Small insertions and deletions (INDELs) in human genomes. *Hum. Mol. Genet.* **19**, R131–R136 (2010).

2. Mahmoud, M. *et al.* Structural variant calling: the long and the short of it. *Genome Biol.* **20**, 246 (2019).

3. Auton, A. *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).

4. Uffelmann, E. *et al.* Genome-wide association studies. *Nat. Rev. Methods Primer* **1**, 1–21 (2021).

5. Auer, P. L. & Lettre, G. Rare variant association studies: considerations, challenges and opportunities. *Genome Med.* **7**, 16 (2015).

6. Malhotra, D. & Sebat, J. CNVs: Harbingers of a Rare Variant Revolution in Psychiatric Genetics. *Cell* **148**, 1223–1241 (2012).

7. Sánchez, X. C. *et al.* Associations of psychiatric disorders with sex chromosome aneuploidies in the Danish iPSYCH2015 dataset: a case-cohort study. *Lancet Psychiatry* **10**, 129–138 (2023).

8. Vaez, M. *et al.* Population-Based Risk of Psychiatric Disorders Associated With Recurrent Copy Number Variants. *JAMA Psychiatry* (2024) doi:10.1001/jamapsychiatry.2024.1453.

9. Halvorsen, M. W. *et al.* A burden of rare copy number variants in obsessive-compulsive disorder. *Mol. Psychiatry* (2024) doi:10.1038/s41380-024-02763-7.

10. Pinto, D. *et al.* Functional impact of global rare copy number variation in autism spectrum disorders. *Nature* **466**, 368–372 (2010).

11. Gu, W., Zhang, F. & Lupski, J. R. Mechanisms for human genomic rearrangements. *PathoGenetics* **1**, 4 (2008).

12. Rees, E. & Kirov, G. Copy number variation and neuropsychiatric illness. *Curr. Opin. Genet. Dev.* **68**, 57–63 (2021).

13. Crawford, K. *et al.* Medical consequences of pathogenic CNVs in adults: analysis of the UK

Biobank. *J. Med. Genet.* **56**, 131–138 (2019).

14. Shprintzen, R. J. Velo-Cardio-Facial Syndrome: 30 Years of Study. *Dev. Disabil. Res. Rev.* **14**, 3–10 (2008).

15. Lieber, M. R. The mechanism of human nonhomologous DNA end joining. *J. Biol. Chem.* **283**, 1–5 (2008).

16. Lieber, M. R. The mechanism of double-strand DNA break repair by the nonhomologous DNA end-joining pathway. *Annu. Rev. Biochem.* **79**, 181–211 (2010).

17. Lee, J. A., Carvalho, C. M. B. & Lupski, J. R. A DNA replication mechanism for generating nonrecurrent rearrangements associated with genomic disorders. *Cell* **131**, 1235–1247 (2007).

18. Hastings, P. J., Ira, G. & Lupski, J. R. A Microhomology-Mediated Break-Induced Replication Model for the Origin of Human Copy Number Variation. *PLOS Genet.* **5**, e1000327 (2009).

19. Zarrei, M., MacDonald, J. R., Merico, D. & Scherer, S. W. A copy number variation map of the human genome. *Nat. Rev. Genet.* **16**, 172–183 (2015).

20. Monlong, J. *et al.* Human copy number variants are enriched in regions of low mappability. *Nucleic Acids Res.* **46**, 7236–7249 (2018).

21. Juan, D., Rico, D., Marques-Bonet, T., Fernández-Capetillo, Ó. & Valencia, A. Late-replicating CNVs as a source of new genes. *Biol. Open* **2**, 1402–1411 (2013).

22. Falconer, D. S. The inheritance of liability to certain diseases, estimated from the incidence among relatives. *Ann. Hum. Genet.* **29**, 51–76 (1965).

23. Vaez, M. *et al.* Evaluating the Joint Effects of Recurrent Copy Number Variants and Polygenic Scores on the Risk of Psychiatric Disorders in the iPSYCH2015 Case-Cohort Sample. 2024.09.23.24314234 Preprint at https://doi.org/10.1101/2024.09.23.24314234 (2024).

24. Bergen, S. E. *et al.* Joint Contributions of Rare Copy Number Variants and Common SNPs to Risk for Schizophrenia. *Am. J. Psychiatry* **176**, 29–35 (2019).

25. Martin, J., O'Donovan, M. C., Thapar, A., Langley, K. & Williams, N. The relative

contribution of common and rare genetic variants to ADHD. *Transl. Psychiatry* **5**, e506–e506 (2015).

26. Reich, D. E. & Lander, E. S. On the allelic spectrum of human disease. *Trends Genet. TIG* **17**, 502–510 (2001).

27. Saint Pierre, A. & Génin, E. How important are rare variants in common disease? *Brief. Funct. Genomics* **13**, 353–361 (2014).

28. Chaste, P., Roeder, K. & Devlin, B. The Yin and Yang of Autism Genetics: How Rare De Novo and Common Variations Affect Liability. *Annu. Rev. Genomics Hum. Genet.* **18**, 167–187 (2017).

29. An, J.-Y. *et al.* Genome-wide de novo risk score implicates promoter variation in autism spectrum disorder. *Science* **362**, (2018).

30. *Diagnostic and Statistical Manual of Mental Disorders: DSM-5™, 5th Ed.* xliv, 947 (American Psychiatric Publishing, Inc., Arlington, VA, US, 2013). doi:10.1176/appi.books.9780890425596.

31. Franke, B. *et al.* Live fast, die young? A review on the developmental trajectories of ADHD across the lifespan. *Eur. Neuropsychopharmacol. J. Eur. Coll. Neuropsychopharmacol.* **28**, 1059–1088 (2018).

32. Faraone, S. V. & Larsson, H. Genetics of attention deficit hyperactivity disorder. *Mol. Psychiatry* **24**, 562–575 (2019).

33. Satterstrom, F. K. *et al.* Autism spectrum disorder and attention deficit hyperactivity disorder have a similar burden of rare protein-truncating variants. *Nat. Neurosci.* **22**, 1961–1965 (2019).

34. Williams, N. M. *et al.* Rare chromosomal deletions and duplications in attention-deficit hyperactivity disorder: a genome-wide analysis. *Lancet Lond. Engl.* **376**, 1401–1408 (2010).

35. Gudmundsson, O. O. *et al.* Attention-deficit hyperactivity disorder shares copy number variant risk with schizophrenia and autism spectrum disorder. *Transl. Psychiatry* **9**, 258 (2019).

36. Hallmayer, J. *et al.* Genetic heritability and shared environmental factors among twin pairs with autism. *Arch. Gen. Psychiatry* **68**, 1095–1102 (2011).

37. Girirajan, S. *et al.* Refinement and discovery of new hotspots of copy-number variation associated with autism spectrum disorder. *Am. J. Hum. Genet.* **92**, 221–237 (2013).

38. de la Torre-Ubieta, L., Won, H., Stein, J. L. & Geschwind, D. H. Advancing the understanding of autism disease mechanisms through genetics. *Nat. Med.* **22**, 345–361 (2016).

39. Santomauro, D. F. *et al.* The global epidemiology and health burden of the autism spectrum: findings from the Global Burden of Disease Study 2021. *Lancet Psychiatry* **0**, (2024).

40. De Rubeis, S. & Buxbaum, J. D. Genetics and genomics of autism spectrum disorder: embracing complexity. *Hum. Mol. Genet.* **24**, R24–R31 (2015).

41. Gordovez, F. J. A. & McMahon, F. J. The genetics of bipolar disorder. *Mol. Psychiatry* **25**, 544–559 (2020).

42. Otte, C. *et al.* Major depressive disorder. *Nat. Rev. Dis. Primer* **2**, 16065 (2016).

43. Jablensky, A. The diagnostic concept of schizophrenia: its history, evolution, and future prospects. *Dialogues Clin. Neurosci.* **12**, 271–287 (2010).

44. Owen, M. J., Legge, S. E., Rees, E., Walters, J. T. R. & O'Donovan, M. C. Genomic findings in schizophrenia and their implications. *Mol. Psychiatry* **28**, 3638–3647 (2023).

45. Insel, T. R. Rethinking schizophrenia. *Nature* **468**, 187–193 (2010).

46. Ferrara, M. *et al.* Sex differences in schizophrenia-spectrum diagnoses: results from a 30-year health record registry. *Arch. Womens Ment. Health* **27**, 11–20 (2024).

47. Lejeune, J., Gauthier, M. & Turpin, R. [Human chromosomes in tissue cultures]. *Comptes Rendus Hebd. Seances Acad. Sci.* **248**, 602–603 (1959).

48. Ataman, A. D., Vatanoğlu-Lutz, E. E. & Yıldırım, G. Medicine in stamps: history of Down syndrome through philately. *J. Turk. Ger. Gynecol. Assoc.* **13**, 267–269 (2012).

49. DeLisi, L. E. *et al.* Schizophrenia and Sex Chromosome Anomalies. *Schizophr. Bull.* **20**,

495–505 (1994).

50. Bassett, A. S. Chromosomal Aberrations and Schizophrenia: Autosomes. *Br. J. Psychiatry* **161**, 323–334 (1992).

51. Raphael, T. & Shaw, M. W. Chromosome studies in schizophrenia. *JAMA* **183**, 1022–1028 (1963).

52. Sherrington, R. *et al.* Localization of a susceptibility locus for schizophrenia on chromosome 5. *Nature* **336**, 164–167 (1988).

53. Craddock, N. & Owen, M. Chromosomal Aberrations and Bipolar Affective Disorder. *Br. J. Psychiatry* **164**, 507–512 (1994).

54. Murphy, K. C., Jones, L. A. & Owen, M. J. High rates of schizophrenia in adults with velo-cardio-facial syndrome. *Arch. Gen. Psychiatry* **56**, 940–945 (1999).

55. Gordeeva, V., Sharova, E. & Arapidi, G. Progress in Methods for Copy Number Variation Profiling. *Int. J. Mol. Sci.* **23**, 2143 (2022).

56. Jacobs, P. A. & Strong, J. A. A Case of Human Intersexuality Having a Possible XXY Sex-Determining Mechanism. *Nature* **183**, 302–303 (1959).

57. Rudkin, G. T. & Stollar, B. D. High resolution detection of DNA-RNA hybrids in situ by indirect immunofluorescence. *Nature* **265**, 472–473 (1977).

58. Kallioniemi, A. *et al.* Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science* **258**, 818–821 (1992).

59. Pinkel, D. *et al.* High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.* **20**, 207–211 (1998).

60. Bignell, G. R. *et al.* High-Resolution Analysis of DNA Copy Number Using Oligonucleotide Microarrays. *Genome Res.* **14**, 287–295 (2004).

61. Peiffer, D. A. *et al.* High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping. *Genome Res.* **16**, 1136–1148 (2006).

62. null, null. The "All of Us" Research Program. *N. Engl. J. Med.* **381**, 668–676 (2019).

63. Halldorsson, B. V. *et al.* The sequences of 150,119 genomes in the UK Biobank. *Nature*

607, 732–740 (2022).

64. Zhao, M., Wang, Q., Wang, Q., Jia, P. & Zhao, Z. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. *BMC Bioinformatics* **14**, S1 (2013).

65. Zhou, B. *et al.* Whole-genome sequencing analysis of CNV using low-coverage and paired-end strategies is efficient and outperforms array-based CNV analysis. *J. Med. Genet.* **55**, 735–743 (2018).

66. Gabrielaite, M. *et al.* A Comparison of Tools for Copy-Number Variation Detection in Germline Whole Exome and Whole Genome Sequencing Data. *Cancers* **13**, 6283 (2021).

67. Venkatraman, E. S. & Olshen, A. B. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinforma. Oxf. Engl.* **23**, 657–663 (2007).

68. Colella, S. *et al.* QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Res.* **35**, 2013–2025 (2007).

69. Wang, K. *et al.* PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res.* **17**, 1665–1674 (2007).

70. Koike, A., Nishida, N., Yamashita, D. & Tokunaga, K. Comparative analysis of copy number variation detection methods and database construction. *BMC Genet.* **12**, 29 (2011).

71. Seiser, E. L. & Innocenti, F. Hidden Markov Model-Based CNV Detection Algorithms for Illumina Genotyping Microarrays. *Cancer Inform.* **13s7**, CIN.S16345 (2014).

72. Diskin, S. J. *et al.* Adjustment of genomic waves in signal intensities from whole-genome SNP genotyping platforms. *Nucleic Acids Res.* **36**, e126–e126 (2008).

73. Montalbano, S. *et al.* CNValidatron, automated validation of CNV calls using computer vision. 2024.09.09.612035 Preprint at https://doi.org/10.1101/2024.09.09.612035 (2024).

74. Kirov, G. *et al.* Comparative genome hybridization suggests a role for NRXN1 and APBA2 in schizophrenia. *Hum. Mol. Genet.* **17**, 458–465 (2008).

75. Sebat, J. *et al.* Strong Association of De Novo Copy Number Mutations with Autism. *Science* **316**, 445–449 (2007).

76. Stefansson, H. *et al.* Large recurrent microdeletions associated with schizophrenia. *Nature* **455**, 232–236 (2008).

77. Training. https://support.illumina.com/training.html?filters=web%253Asoftware%252Fgenomestudio.

78. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).

79. Vaswani, A. *et al.* Attention Is All You Need. Preprint at https://doi.org/10.48550/arXiv.1706.03762 (2023).

80. Schmidhuber, J. Annotated History of Modern AI and Deep Learning. Preprint at https://doi.org/10.48550/arXiv.2212.11279 (2022).

81. Max Khun & Kjell Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. (Taylor & Francis Group, 2019).

82. James, G., Witten, D., Hastie, T., Tibshirani, R. & Taylor, J. *An Introduction to Statistical Learning: With Applications in Python*. (Springer International Publishing, Cham, 2023). doi:10.1007/978-3-031-38747-0.

83. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

84. Glorot, X., Bordes, A. & Bengio, Y. Deep Sparse Rectifier Neural Networks.

85. Householder, A. S. A theory of steady-state activity in nerve-fiber networks: I. Definitions and preliminary lemmas. *Bull. Math. Biophys.* **3**, 63–69 (1941).

86. Abadi, M. *et al.* TensorFlow: A System for Large-Scale Machine Learning.

87. Collobert, R., Kavukcuoglu, K. & Farabet, C. Torch7: A Matlab-like Environment for Machine Learning.

88. Chollet, F. Keras: The Python Deep Learning library. in (2018).

89. Falbel, D. & Luraschi, J. torch: Tensors and Neural Networks with ‘GPU’ Acceleration.

(2024).

90. Elman, J. L. Finding Structure in Time. *Cogn. Sci.* **14**, 179–211 (1990).

91. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun ACM* **60**, 84–90 (2017).

92. Maccagno, A. *et al.* A CNN Approach for Audio Classification in Construction Sites. in *Progresses in Artificial Intelligence and Neural Systems* (eds. Esposito, A., Faundez-Zanuy, M., Morabito, F. C. & Pasero, E.) 371–381 (Springer, Singapore, 2021). doi:10.1007/978-981-15-5093-5_33.

93. Demongeot, J. & Taramasco, C. Evolution of social networks: the example of obesity. *Biogerontology* **15**, 611–626 (2014).

94. Maslov, S. & Sneppen, K. Specificity and Stability in Topology of Protein Networks. *Science* **296**, 910–913 (2002).

95. On power-law relationships of the Internet topology | ACM SIGCOMM Computer Communication Review. https://dl.acm.org/doi/10.1145/316194.316229.

96. Vignery, K. & Laurier, W. A methodology and theoretical taxonomy for centrality measures: What are the best centrality indicators for student networks? *PLOS ONE* **15**, e0244377 (2020).

97. Helander, M. E. & McAllister, S. The gravity of an edge. *Appl. Netw. Sci.* **3**, 7 (2018).

98. Bybjerg-Grauholm, J. *et al.* The iPSYCH2015 Case-Cohort sample: updated directions for unravelling genetic and environmental architectures of severe mental disorders. 2020.11.30.20237768 Preprint at https://doi.org/10.1101/2020.11.30.20237768 (2020).

99. Pedersen, C. B. *et al.* The iPSYCH2012 case–cohort sample: new directions for unravelling genetic and environmental architectures of severe mental disorders. *Mol. Psychiatry* **23**, 6–14 (2018).

100. Sudlow, C. *et al.* UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Med.* **12**, e1001779 (2015).

101. Bycroft, C. *et al.* The UK Biobank resource with deep phenotyping and genomic data.

Nature **562**, 203–209 (2018).

102. Eldjarn, G. H. *et al.* Large-scale plasma proteomics comparisons through genetics and disease associations. *Nature* **622**, 348–358 (2023).

103. Davis, K. A. S. *et al.* Mental health in UK Biobank – development, implementation and results from an online questionnaire completed by 157 366 participants: a reanalysis. *BJPsych Open* **6**, e18 (2020).

104. Fry, A. *et al.* Comparison of Sociodemographic and Health-Related Characteristics of UK Biobank Participants With Those of the General Population. *Am. J. Epidemiol.* **186**, 1026–1034 (2017).

105. PennCNV-Affy - PennCNV. https://penncnv.openbioinformatics.org/en/latest/user-guide/affy/.

106. Gudbjartsson, D. F. *et al.* Sequence variants from whole genome sequencing a large group of Icelanders. *Sci. Data* **2**, 150011 (2015).

107. Reissner, C., Runkel, F. & Missler, M. Neurexins. *Genome Biol.* **14**, 213 (2013).

108. Südhof, T. C. Synaptic Neurexin Complexes: A Molecular Code for the Logic of Neural Circuits. *Cell* **171**, 745–769 (2017).

109. Rees, E. *et al.* Analysis of copy number variations at 15 schizophrenia-associated loci. *Br. J. Psychiatry* **204**, 108–114 (2014).

110. Castronovo, P. *et al.* Phenotypic spectrum of *NRXN1* mono- and bi-allelic deficiency: A systematic review. *Clin. Genet.* **97**, 125–137 (2020).

111. Zahir, F. R. *et al.* A patient with vertebral, cognitive and behavioural abnormalities and a de novo deletion of NRXN1alpha. *J. Med. Genet.* **45**, 239–243 (2008).

112. Rujescu, D. *et al.* Disruption of the neurexin 1 gene is associated with schizophrenia. *Hum. Mol. Genet.* **18**, 988–996 (2009).

113. Marshall, C. R. *et al.* Structural variation of chromosomes in autism spectrum disorder. *Am. J. Hum. Genet.* **82**, 477–488 (2008).

114. Kirov, G. *et al.* Neurexin 1 (NRXN1) deletions in schizophrenia. *Schizophr. Bull.* **35**,

851–854 (2009).

115. Lowther, C. *et al.* Molecular characterization of NRXN1 deletions from 19,263 clinical microarray cases identifies exons important for neurodevelopmental disease expression. *Genet. Med. Off. J. Am. Coll. Med. Genet.* **19**, 53–61 (2017).

Accurate and Effective Detection of Recurrent Copy Number Variants in Large SNP Genotype Datasets

Simone Montalbano,[1,2,4] Xabier Calle Sánchez,[1,2,4] Morteza Vaez,[1,2] Dorte Helenius,[1,2] Thomas Werge,[1,2,3,5] and Andrés Ingason[1,2,3,5]

[1] Institute of Biological Psychiatry, Mental Health Services, Copenhagen University Hospital, Roskilde, Denmark
[2] The Lundbeck Foundation Initiative for Integrative Psychiatric Research (iPSYCH), Copenhagen and Aarhus, Denmark
[3] Lundbeck Foundation Center for GeoGenetics, GLOBE Institute, University of Copenhagen, Copenhagen, Denmark
[4] These authors contributed equally to this work.
[5] Corresponding author: *thomas.werge@regionh.dk, andres.ingason@regionh.dk*

Published in the Bioinformatics section

Structural variations, including recurrent Copy Number Variants (CNVs) at specific genomic loci, have been found to be associated with increased risk of several diseases and syndromes. CNV carrier status can be determined in large collections of samples using SNP arrays and, more recently, sequencing data. Although there is some consensus among researchers about the essential steps required in such analysis (i.e., CNV calling, filtering of putative carriers, and visual validation using intensity data plots of the genomic region), standard methodologies and processes to control the quality and consistency of the results are lacking. Here, we present a comprehensive and user-friendly protocol that we have refined from our extensive research experience in the field. We cover every aspect of the analysis, from input data curation to final results. For each step, we highlight which parameters affect the analysis the most and how different settings may lead to different results. We provide a pipeline to run the complete analysis with effective (but customizable) pre-sets. We present software that we developed to better handle and filter putative CNV carriers and perform visual inspection to validate selected candidates. Finally, we describe methods to evaluate the critical sections and actions to counterbalance potential problems. The current implementation is focused on Illumina SNP array data. All the presented software is freely available and provided in a ready-to-use docker container. © 2022 The Authors. Current Protocols published by Wiley Periodicals LLC.

Basic Protocol 1: From raw intensity data files to CNV calls
Basic Protocol 2: From CNV calls to validated CNV carriers.
Basic Protocol 3: Quality control and quality assessment
Basic Protocol 4: Install the necessary software

Keywords: bioinformatics pipeline • CNVs • structural variation • SNPs

CURRENT PROTOCOLS
A Wiley Brand

Current Protocols e621, Volume 2
Published in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpz1.621

How to cite this article:
Montalbano, S., Sánchez, X. C., Vaez, M., Helenius, D., Werge, T., & Ingason, A. (2022). Accurate and effective detection of recurrent copy number variants in large SNP genotype datasets. *Current Protocols*, *2,* e621. doi: 10.1002/cpz1.621

INTRODUCTION

Structural variants are genomic rearrangements involving a sequence of base pairs. They can be categorized into multiple subgroups including indels, tandem repeats, Copy Number Variants (deletions and duplications; CNVs), and large chromosomal abnormalities (Sudmant et al., 2015). Recurrent CNVs constitute a structural variant class of particular interest, as many of them have been shown to be associated with disease traits (Malhotra & Sebat, 2012; Weischenfeldt, Symmons, Spitz, & Korbel, 2013). Recurrent CNVs usually consist of duplications and deletions occurring in specific loci of the genome interspersed by so-called low-copy repeats (LCR). The LCR are typically relatively large (>10 kb) sequence elements with high (>95%) sequence homology, and during cell replication their proximity can facilitate a non-allelic homologous recombination (NaHR) between chromosomes/chromatids, resulting in the deletion and duplication of the genetic material in between the two LCR copies.(Sharp et al., 2005; Turner et al., 2008) Although such events occur sporadically, the breakpoints of the resulting CNVs will always be dictated by the genomic position of the LCRs, hence the term "recurrent." Most recurrent disease-associated CNVs are rare (<0.1%) and rather large (>300 kb) (Crawford et al., 2019; Stankiewicz & Lupski, 2002). They often occur *de novo* in the carrier and tend to not segregate over many generations (Stefansson et al., 2008), although this varies widely across loci and CNV types depending on the severity and frequency of the pathogenic symptoms associated with them. Some examples include 1q21.1, 15q11.2, 15q13.3, and 22q11.2 (Driscoll et al., 1992; Stefansson et al., 2008).

Large chromosomal rearrangements have been traditionally detected via microscopic karyotype inspection, while most recurrent CNVs are too small for such detection (hence sometimes referred to as "submicroscopic"), and must be detected through inferential molecular genetics methods. Currently, most recurrent CNVs are routinely identified ("called") from SNP-arrays and/or sequencing technology. Of these, SNP-arrays are more commonly applied in very large studies, as SNP-array genotyping is less expensive than whole-genome sequencing, whereas the latter typically returns more accurate and precise results but is more complex to analyze. The most only applied method to "call" CNVs from SNP-array data is PennCNV (Wang et al., 2007), a hidden Markov model (HMM)–based approach which evaluates deviations from expected patterns in two key derived measures of the raw A and B allele intensities from which SNP-array genotypes are determined. These derived measures are the log-R-ratio (LRR), which measures the overall relative intensity of each probe compared to all other probes, and the B-allele-frequency (BAF), which measures the relative intensity of the B-allele to the total intensity for each probe. PennCNV is considered the current standard for CNV calling; however, it has some shortcomings. PennCNV internal HMM only considers successive SNPs at any step, making it extremely sensitive to local noise. Due to this intrinsic limitation of the model, false positives (calls only due to noise), over-segmentation (a true CNV erroneously split into smaller calls), and imprecise boundaries in general are common problems affecting PennCNV calls. These issues especially affect results obtained from noisy datasets (the main sources of noise are discussed in the main text). Of note, complete false negatives are not generally considered a concern.

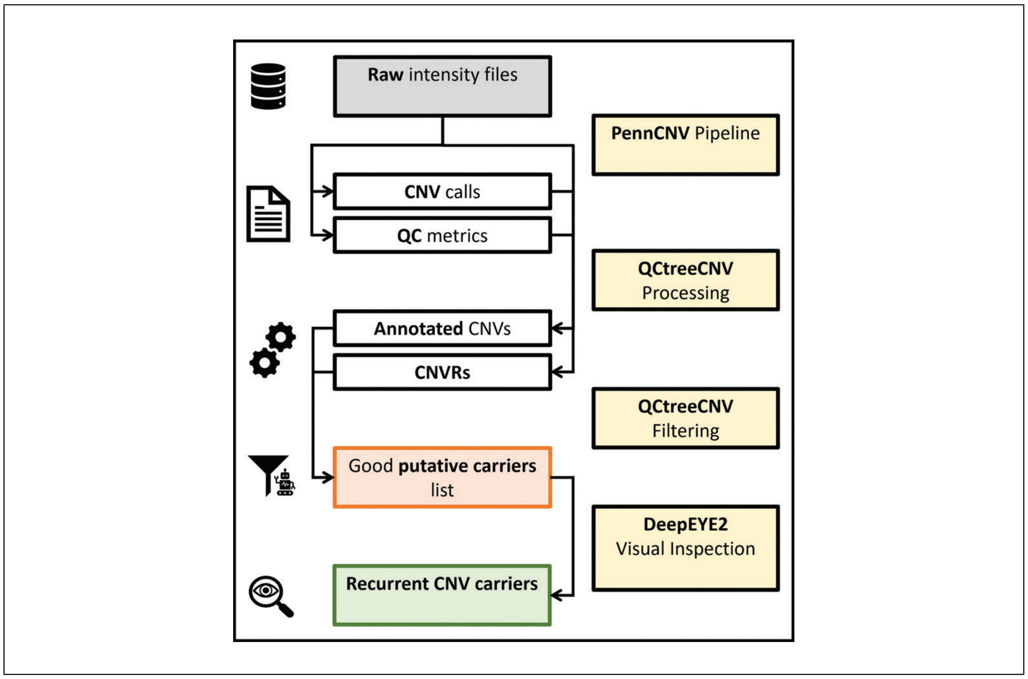


Figure 1 Schematic representation of the whole protocol. The four major processing blocks are shown in yellow, the main inputs (intensity files) in gray, intermediary outputs in red, and the final results in green.

Prompted by our own analyses of recurrent CNVs in several large SNP-array genotype datasets, and a perceived lack of standardized procedures to deal with the most commonly encountered issues in such analyses, we have developed a set of protocols that in our opinion will facilitate accurate and efficient calling of recurrent CNVs (or CNVs at other fixed loci) in any large SNP-array dataset. In Basic Protocol 1, we describe how to filter uninformative SNPs from intensity data in order to improve accuracy of CNV calls (for simplicity, here and in following protocols we consider the Illumina "final report," containing the key LRR and BAF measures, as raw data —see the following sections, Necessary Resources, and Basic Protocol 1 for a formal data description). Moreover, we show how to use the provided PennCNV pipeline (including some light processing), as well as the tabix-indexing of the raw data for the following steps. In Basic Protocol 2, we illustrate how to effectively filter putative carriers in selected loci as well as how to visually validate the CNVs calls. This section is implemented using the software packages we provide. Figure 1 illustrates the overall protocol schematics. In Support Protocol 1, we illustrate how to run and interpret an extensive battery of tests in order to assess the quality of the results. Finally, we describe which parameters in which steps should be modified in order to improve the overall quality of the results, if any. In Support Protocol 2, we detail the install process for all necessary software.

As discussed in the Commentary, the protocol is primarily designed for processing large datasets (e.g., >50,000 samples). However, we also tried to code and describe most steps so that the whole protocol, or at least sections of it, can be as useful also in smaller studies and in ones not strictly focused on recurrent CNVs.

NECESSARY RESOURCES

Hardware

The protocol is designed for large collections of genotyping data. For this reason, the user will need at least a modern workstation, but a high-performance computing cluster is preferred. While the RAM and disk space requirements are not high compared to, e.g., programs that handle next-generation sequencing data, several steps of the pipeline need to read raw data; thus a fast storage solution is advised.

Software

We provide a docker/singularity image containing all necessary software to run the protocols. Refer to Support Protocol 2 for instructions on how to use the image and how to install our software separately. The initial PennCNV calling pipeline is designed to take advantage of the job scheduler SLURM and run all commands via the singularity container. Only Linux environments are supported.

Files

The article is focused on Illumina genotyping data; raw data for each sample of the cohort is basically the only necessary prerequisite. There must be one file per sample. Additionally, a key file linking each sample to its raw data file is needed. The raw data format is Illumina "intensity file"/"final report" format, which is a text file containing at least the following columns: "Name," "Log R Ratio," "B Allele Freq." The expected format is detailed in step 1 of this protocol. Please note that while intensity files are considered raw data, they are not the most primitive form in which Illumina SNP-array data can exist. If the raw data is in IDAT format, it needs to be processed into intensity files before proceeding with the protocol. IDAT files can be converted to GTC files using the Illumina software IAAP (available at *https://support.illumina.com/array/array_software/illumina-array-analysis-platform.html*), then GTCs can be converted to intensity file using the python library IlluminaBeadArrayFiles (available here *https://github.com/Illumina/BeadArrayFiles*). All required inputs are discussed in more detail at the beginning of Basic Protocols 1 and 2.

BASIC PROTOCOL 1

FROM RAW INTENSITY FILES TO CNV CALLS

PennCNV (Wang et al., 2007) has been the de facto standard for CNV calling since its initial release 15 years ago. While it is fairly easy to use, implementing a pipeline to process tens or hundreds of thousands of samples can be challenging. Moreover, there are many checks the users need to perform in order to ensure all commands run as intended, as well as some files that are not straightforward to generate or obtain. The pipeline we refined takes care of most of these issues. This protocol details how to run the pipeline and obtain the raw CNV calls and sample QC files to use for further processing.

Required Files

To run the protocol three main files plus the intensity files are required, and they must be in the correct format. We provide GC content file (requirement 4); however, the user needs to generate the samples list (requirement 3) and the SNPs position file (requirement 2).

1. Intensity files. These files store the raw information regarding each Illumina SNP probe. Each file should contain at least the following three columns: "Name," "Log R Ratio," "B Allele Freq." Column names must be exact; other columns can be present as long as these are the first three. These columns contain, namely: the name of the SNP markers, the value of Log2 R ratio (a relative measure of the light intensity), and B Allele Frequency (a measure of the allelic composition). Intensity files are also called "Final Reports" and may contain a multi-line header. Moreover, it is possible that multiple samples are stored in the same file (e.g., an entire genotyping batch) and that the files are compressed. In such cases, PennCNV will not work. All these problems can be very different from case to case, but they can be solved using standard GNU utilities tools (included in every major linux distribution) such as sed or the program awk. To solve the second problem, the PennCNV script `split_illumina_report.pl` can also be used. Note that in this case

the user may need to regenerate a correct `samples_list.txt` file. Note that it is important to know in which build of the human genome the intensity data is (e.g., "hg19"), and that all samples in the same project must be on the same version.

2. `snppos.txt`. This is a tab-separated text file that contains at least the following three columns: "Name," "Chr," and "Position," which are the name and genomic location of each SNP marker. The first column must use the same names as in the intensity files. In projects where each intensity file contains this information, any sample can be used to generate the SNP position file. Otherwise, there should be a file called "PFB" (Population Frequency of the B allele) that contains all the columns required by the `snppos.txt` format.
3. `samples_list.txt`. This must be a two-column tab-separated text file with a header. The two columns must be "sample_ID" and "file_path." "sample_ID" needs to be the identifier for the specific sample, and "file_path" needs to be the complete path (thus starting from root, "`/`"; therefore avoid using links and the home directory, "∼/", in the path) to the intensity file for that sample. Again, this protocol assumes there is one intensity file per sample. If this is not the case, see step 4. In the case where samples were genotyped in waves or batches, these can be used; otherwise, batches of approximately 2000 samples will be created. In the first instance, an additional column called "batch" must be present, and the content must be integers indicating the specific batch.
4. GC content file. This file is used by the PennCNV script `cal_gc_snp.pl`. This file is not straightforward to generate, but we provide the hg38, hg19 and hg18 version as part of the IBPcnv repository. The hg38 version is available from the PennCNV repository: *https://github.com/WGLab/PennCNV/blob/master/gc_file/*. It is also included in the IBPcnv repository for user convenience. Finally, the original hg18 version can also be obtained directly from UCSC Genome Browser at *http://hgdownload.cse.ucsc.edu/goldenPath/hg18/database/gc5Base.txt.gz*.

Protocol steps

1. Setup. We provide all required software and scripts in a docker/singularity container and a GitHub repository. Excluding the raw data, all files and subdirectories need to be in the same directory (`$workingdir` hereafter). Support Protocol 2 details the installation process, as well as some suggestions on how to set up directories and the environment for the analysis.

2. Initial checks and required files generation:

   a. This will do the following. First, it will run a series of tests to check that all intensity files are present and in the correct format. Then, it will check if samples were already divided into batches and that this is in the correct format. If not, it will create the batches and separate the samples into groups of approximately 2000 each; the actual number may vary in order to not have the last batch significantly smaller than the rest.
   b. To complete this step, move to the main working directory (`cd $workingdir`) and run:

```
singularity exec ibpcnv.simg Rscript \
IBPcnv/penncnv_pipeline/01_preprocess.R \
$workingdir 1 2000 $tabix_folder
```

where `$tabix_folder` is the complete path to the directory where the tabix-indexed files will be created in step 5. The approximate batch size can be changed via the third parameter; however it is not recommended to use a small batch size, as the population B allele frequency (PFB) files will be created per batch.

   c. If the script fails checking the batches but no errors are found in the intensity files, the second parameter can be set to 0 to skip the initial checks that constitute the slower part of the testing.
   d. If all checks are successfully completed, the script will write the per-batch sample list files needed by PennCNV in `$workingdir/listfile/`. The script will also print the number of batches that will be used.

3. Select the SNP markers:

   a. This step will do the following. First it will extract the marker names and positions from an intensity file. It will then download the HRC SNP list, selecting only SNPs that are strictly biallelic, known (with a name in dbSNP142, no “.”) and with a minor allele frequency of at least $minMAF (MAF values for each SNP are obtained from the HRC); we suggest 0.001 (0.1%). Then, it will merge the two tables, remove markers with duplicated “SNP_ID,” remove markers that map to the same position, and finally it will create the `snppos.txt` file needed by PennCNV.
   b. To complete this step, move to the main working directory and run:

```
singularity exec ibpcnv.simg Rscript \
   IBPcnv/penncnv_pipeline/02_select_SNPs.R \
   $workingdir $minMAF $hgversion TRUE
```

The removal of duplicated markers can be avoided by setting the last parameter to FALSE. At the moment of this writing, the HRC SNP list is available only in hg19 coordinates; thus, the last argument can take only “hg19” as value. This may change in the near future. Any other list of SNPs that follows the same format will work. See the Sanger Institute documentation for details: *http://ngs.sanger.ac.uk/README* and *ftp://ngs.sanger.ac.uk/production/hrc/HRC.r1/README*. When using a different SNP list, it is suggested to first use the default one in order to be able to easily check the format.

4. PennCNV calling pipeline:

   a. First, run the calling pipeline in parallel on each wave. To complete this step, move to the main working directory and run the command:

```
bash IBPcnv/penncnv_pipeline/03_penncnv_pipeline.sh \
   $workingdir $ibpcnvdir $n_batches hg19
```

where `$n_batches` is the number of batches from step 2.

   b. PennCNV calling parameters (minimum number of SNPs and minimum length of each call) can be changed in the script `$ibpcnvdir/penncnv_pipeline/03_2_cnv_calling.sh`, before launching the previous step. By default, these are set to 5 and 1000 bp respectively.
   c. Check that all batches are completed successfully. Catching any PennCNV error is quite straightforward—the following command can be used

```
cd $workingdir && grep ’ERROR’ pennlogs/*
```

However, SLURM and PBS problems can be more complex to find. To check that all samples have been processed, run:

```
if [$(wc -l samples_list.txt | cut -d ’ ’ -f1) == \
   $(wc -l results/autosome.qc | cut -d ’ ’ -f1)]; then
   echo "All good"; fi
```

If this check fails, the following command can be used to list the samples that have not been processed by PennCNV:

```
join -v2 -1 1 -2 2 <(LANG=C tail -n+2 1 results/autosome.qc | sort -k) \
   <(LANG=C tail -n+2 samples_list.txt | sort -k 2)
```

One common problem is for a full batch or a batch chunk to fail, usually due to the job scheduler. If that is the case, the full batch can be relaunched running (for PBS systems, use qsub instead of sbatch):

```
sbatch IBPcnv/penncnv_pipeline/03_1_per_wave.sh $workingdir \
$ibpcnvdir $n_wave
```

where $n_batch is the number of the failed batch. We suggest redoing the whole batch also in the case of a partial failure.

d. After the calling pipeline is completed, the batches can be combined into two single files (CNV calls and QC) running (for PBS systems, use qsub instead of sbatch)

```
sbatch IBPcnv/penncnv_pipeline/04_combine_results.sh \
   $workingdir $ibpcnvdir $maxgap
```

e. This will also perform a "soft stitching" step, meaning that for each sample, close calls with the same copy number will be stitched together. This is controlled via the $maxgap parameter; we recommend a value of 0.2 and not higher than 0.4, as higher values could alter the raw call-set. A stronger stitching will be performed in Basic Protocol 2 when selecting putative CNVs in a specific locus.

5. Tabix indexing the intensity files. In Basic Protocol 2, we take advantage of the speed of tabix-indexed files as well as of the GC waviness-adjusted LRR values.

a. The GC model file for the specific array in use should have already been generated. If the user is interested only in the second half of the protocol, the following command can be used:

```
bash IBPcnv/misc/create_gcmodel.sh $workingdir $ibpcnvdir
```

b. To perform the indexing run:

```
bash IBPcnv/penncnv_pipeline/05_launch_tabix.sh $workingdir $ibpcnvdir
```

This will also compute the GC waviness-adjusted LRR values for each marker. This can be used in the visual inspection and is also discussed in the Commentary. This step concludes Basic Protocol 1.

BASIC PROTOCOL 2

FROM CNV CALLS TO VALIDATED CNVs CARRIERS

CNV calling using SNP-array data is an intrinsically imprecise process, and strongly depends on the quality of the initial SNP-array raw data. In particular, it is prone to pick up noise as signal, as well as to "over-segment," that is, incorrectly splitting a (often large) CNV call into several smaller ones. This ultimately may lead to unreliable results. Moreover, precise CNV boundaries (at the level of 1-2 SNP probes) are very difficult to obtain. Different research groups have developed different strategies to overcome these problems. To counter over-segmentation, it is common practice to "stitch" close, consecutive calls with the same copy number (CN). To reduce noise, some kind of filtering is almost always included. At the CNV call level, the filtering can act on the call length, the number of SNP probes, or, more rarely, on the confidence score. Some filtering is usually also performed at sample level; this may include removing samples with too high an LRR standard deviation (LRRSD), extreme BAF drift, or too many calls. A few studies, such as this protocol, also perform an initial filtering on the SNPs in order to reduce the general noise of the raw data. Notably, these approaches tend to lead to false positive calls being a larger issue than false negative ones. Changing the type and strength of the CNV call level filtering enables researchers to balance between false negatives and the number of calls to validate (i.e., false positives to manually screen). In order to reduce the number of false positives, two main strategies have typically been used. The first is to

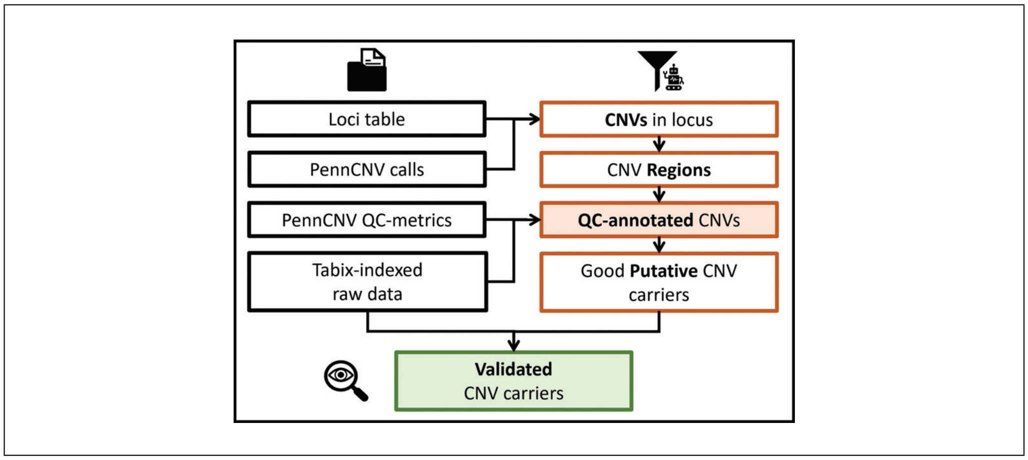


Figure 2 More detailed schematics of Basic Protocol 2. Colors and icons are kept consistent with Figure 1, indicating starting data files, QCtreeCNV, and DeepEYE respectively.

visually validate all putative CNV calls. This approach is time consuming and prone to human error; however, when done correctly, it is the overall best approach, and it is commonly used when calling CNVs in a limited set of specific genomic loci, such as recurrent CNV loci (Calle Sánchez et al., 2021; Stefansson et al., 2014). This approach should always be preferred when the CNVs of interest are rare, and thus a few false positives or negatives could significantly affect estimated prevalence and downstream analysis. The second approach is to use multiple algorithms to perform the CNV calling and intersect the resulting callsets (often using a form of reciprocal overlap in terms of base pairs or probes) and filter the results. An example is using more than one program [such as PennCNV (Wang et al., 2007), QuantiSNP (Colella et al., 2007), and others) and considering "validated" only calls where two or more programs made the call. In this case the reasoning is that even if some false positives are introduced, with a large sample size, these errors will be equally distributed among, e.g., cases and controls. Such studies usually also perform some kind of grouping before doing any analysis (e.g., deletions affecting at least one gene) and rarely consider individual CNVs on their own.

In this protocol, we follow the first approach (i.e., always relying on visual inspection as the final validation step), and we integrate it with newly developed filters that do not depend on the PennCNV output but directly on the raw data. By doing so we are able to reduce the number of putative carriers to inspect, while also minimizing the number of false negatives as much as possible. This approach is particularly suited for large cohorts where the number of false positives can be high. In smaller datasets, it may not be equally useful to perform the filtering step, and the user can simply use our package to standardize the files and then use the graphical interface to validate all putative carriers. It is important to notice that, as stated in the main introduction, this protocol has been designed with a strong focus on recurrent CNV, meaning CNV with relatively fixed and precise start and end positions. As an example, a researcher interested in all recurrent CNVs in the larger 15q region should specify each specific locus individually in the `loci.txt` file (see required files), while a researcher interested in all CNVs overlapping the *NRXN1* gene should use a very low minoverlap value (see step 3c) and avoid the advanced filtering (see step 4d).

Figure 2 shows a schematic representation of the Basic Protocol 2.

Required Files

The only additional file needed to run the second part of the protocol is a list of the loci of interest. The file `loci.txt` must be a four-column tab-separated text file with a header. The columns must be called "locus," "chr," "start," and "end." The chromosome must be in integer format. This format is used in all the results and intermediate R objects, as well as the tabix-indexed intensity files. It simply consists of integers from 1 to 22 that are used for autosomes, plus 23 for X, 24 for Y, 25 for XY, and 26 for MT.

Protocol steps

1. Setup. All software and scripts required are provided in a docker/singularity container and a GitHub repository. Support Protocol 2 details the installation process. It is assumed the user has successfully completed Basic Protocol 1. Throughout this protocol, all files are loaded in R as `data.table` objects. Please note that they behave slightly differently than `data.frame` in certain situations. If the user prefers using data.frame to explore the results, after the protocol completion the main objects can be converted by running:

   ```
   objectA_df <- as.data.frame(objectA_dt)
   ```

   This will ensure full consistency when using rbase commands and the tidyverse framework.

2. QCtreeCNV filtering pipeline. This step is meant to be run interactively in an R session. The user can use the provided commands in an R script; however, we suggest exploring at least a couple of loci interactively before setting the final parameters.

3. Preparation. All code lines are shown in code block 1.

   a. Load data into R. To launch an interactive R session using the provided singularity image, run:

   ```
   singularity exec ibpcnv.simg R
   ```

   At this point, the four main files can be loaded into R. Assuming the user followed the suggested naming in Basic Protocol 1, this can be done by running lines 1 to 5.

   b. Check formats and select the calls in the loci of interest. These steps can be performed with a single function, `qctree_pre()`. It takes the four main objects from the previous step and the parameters for stitching close calls. Line 7 shows the code for default values. If there is any problem with the inputs, the function will fail with an error message explaining the specific problem. By default, the function will also take care of multiple calls in a locus from a single locus, keeping only the largest call, regardless of the copy number. The user can avoid this by setting `rm_dup = F`. Note however that the downstream steps do not support multiple calls per sample in a single locus and will throw an error if any is found.

   c. The stitching function takes three main parameters, minimum number of SNPs for calls to be considered, maximum gap between two consecutive calls with same CN in order to stitch them together, and minimum overlap between a call and a locus for the call to be selected as a putative CNV. Default values are respectively 20, 0.5 and 0.2. These values can be changed with the following parameters: `minsnp, maxgap, minoverlap`, e.g., `pre <- qctree_pre(loci, cnvs, qc, samples, minsnp = 15, maxgap = 0.4, minoverlap = 0.5)`.

   d. Compute CNV Regions. We provide two different functions to compute CNVRs (CNV Regions), `cnvr_fast()` and `cnvrs_create()`. Additionally, the user is free to use a different method as long as the results are in the correct format. CNVRs are used here to separate groups of largely overlapping calls within a certain locus, in particular groups with different lengths. CNVRs and their computation are further discussed in the Commentary and in the package manuals and vignette. The suggested function can be run as shown in line 9:

```
1> library(data.table); setDTthreads(2); library(QCtreeCNV)
2> cnvs <- fread("results/autosome.cnv")
```

```
qc <- fread("results/autosome.qc")
loci <- fread("loci.txt")
samples <- fread("samples_list.txt")

put_cnvs <- qctree_pre(loci, cnvs, qc, samples)

cnvrs <- cnvr_fast(put_cnvs)
```

Code block 1.

4. `qctree()` filtering:

   a. The filtering function is structured as a decision-making tree consisting of five main steps, and each step has multiple parameters the user can change. More technical details on each step and how the main parameters are connected with the outputs of Support Protocol 1 (quality control) are further discussed in the Commentary. To run the function with default values type:

   ```
   cnvs_out <- qctree(cnvrs[[1]], cnvrs[[2]], loci)
   ```

   If no filtering is deemed necessary (for example when the number of putative calls is small), the user can proceed directly to step 4d.

   b. Step 1 of the filtering tree is removing QC outliers sample-wise, and this is the most accessible step to customize. By default, samples will be removed if they have LRRSD > 0.35, BAF drift > 0.01, or GCWF outside of the window –0.02 to 0.02. These values can be changed with the following parameters: `maxLRRSD`, `maxBAFdrift`, `maxGCWF`, `minGCWF`.

   c. The resulting table will contain the column “excl” with the value 0, meaning the line is a good putative CNV call, or 1, meaning the line is a bad putative CNV and can be skipped in the visual inspection step. This table can be exported in the correct format for the visual inspection interface with:

   ```
   export_cnvs(cnvs_out[excl == 0,], "putative_cnvs.txt")
   ```

   This will write the file `putative_cnvs.txt` in the `$workingdir` containing only the good putative CNV calls in the format expected by DeepEYE. To export all calls, type:

   ```
   export_cnvs(cnvs_out, "putative_cnvs.txt")
   ```

   d. If no advanced filtering is needed, the user can simply apply the standard filters (LRRSD, BAFdrift, GCWF) when exporting the table. Using the `data.table` syntax, run:

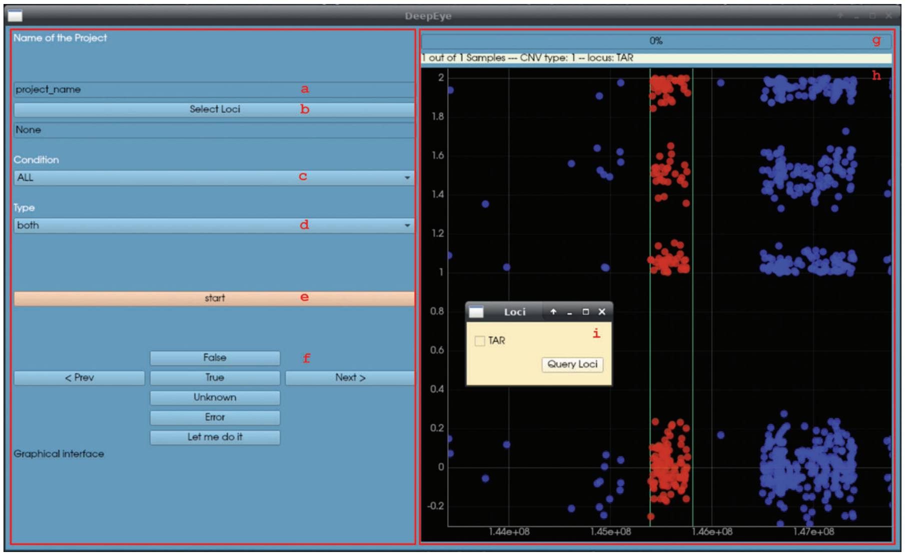
   ```
   export_cnvs(put_cnvs[LRRSD <= 0.35 & BAFdrift <= 0.01 &
         between(GCWF, -0.02, 0.02, incbounds=T),],
      "putative_cnvs.txt")
   ```

5. Visual inspection.

   a. Visual inspection of the putative CNVs is necessary to validate the true CNV carriers with precision. The in-house program DeepEYE (included in the provided container) provides the user with a graphical interface to assign a label (true, false, unknown) to each CNV candidate. The results are stored in an extra column in the putative CNVs file.

   b. DeepEYE requires three main inputs—the samples and loci lists, plus the putative CNVs table. It can be run from the singularity image, assuming all files have the standard names described in the protocol:

   ```
   singularity exec ibpcnv.simg python3 /opt/eyeCNV/visualizer.py \
   $workingdir putative_cnvs.txt loci.txt samples_list.txt GC_YES
   ```



DeepEye

Name of the Project: project_name

None

Condition: ALL

Type: both

0%

1 out of 1 Samples --- CNV type: 1 -- locus: TAR

Figure 3 DeepEYE graphical interface (no CNV is present in the region, simulated data). Refer to step 5b of Basic Protocol 2 for the usage instruction.

This will open a graphical window as shown in Figure 3.

i. Initially, the window will display only a series of buttons and boxes (Fig. 3 left side). In order to start the actual visual inspection (Fig. 3 right side), the user needs to follow these steps:
ii. Name the project (box a); the name will dictate the name of the output file, i.e., `visual_output_projectname.txt`. The project name should be meaningful, e.g., when user "abc" is evaluating deletions in the TAR locus, a good project name could be `TAR_del_abc`.
iii. Select the loci to inspect. Left clicking on button b will open a secondary window (i) where it is possible to select the loci to inspect in the current run.
iv. Select the condition. Left clicking on button c will open a menu with the following options: true, false, unknown, all. In a new project "all" should be selected; while re-evaluating a previous project the user can select only a portion of the calls (e.g., unknown).
v. Select the type. Left clicking on button d will open a menu with the following options: duplications, deletions, any. This lets the user select a specific type of CNV to inspect.
vi. Once this is set, the user can click button e, "start."

c. If at least one CNV call was selected, the right panel will appear. The panel consists of two plots (h); the top is for LRR and the bottom is for BAF. Each dot represents a marker. The red dots are within the locus of interest, the blue dots are outside.
d. For each plot, the user can evaluate if a CNV is present within the red region and record the decision using the dedicated buttons (f):
    i. True: there is a CNV in the red region (with the correct CN, as shown in g).
    ii. False: there is not a CNV in the red region (or the CN is not correct).
    iii. Unknown: it is not possible to tell whether a CNV is present or not, likely due to excessive noise in the region.
    iv. Error: useful to mark samples with problematic intensity data.
e. The progress bar and text in g mark the session's progress. Once all selected calls have been inspected, the result file will be written in `$workingdir`. The last column, "visual_output," contains the record of the visual inspection as integers:

1 (true), 2 (false), 3 (unknown), –7 (error). This file can be used again as a putative CNV file, for example to re-evaluate unknown calls only.

6. Visual evaluation concludes Basic Protocol 2. The file `visual_output_projectname.txt` will contain the results. The visual output codes (step 5e) can be used to filter the relevant CNVs. See also the final section Understanding Results.

SUPPORT PROTOCOL 1

QUALITY CONTROL AND QUALITY ASSESSMENT

This protocol, similar to any other CNV calling pipeline, implements a certain set of filters. We propose valid default values based on the scientific literature and our research experience (Calle Sánchez et al., 2021; Stefansson et al., 2014). However, since this protocol is mostly aimed at CNV calling in recurrent loci where the actual CNVs are quite rare, we need to be extremely careful that our processing does not introduce any false negatives that may severely bias our estimates (false positives are controlled via visual inspection). In large cohorts, this ultimately means balancing the strength of the filters and the amount of manual validation required. Finally, some filters are applied sample-wise on values that directly reflect the noise in the data, for the specific sample (LRRSD and BAF drift in particular). We found that it is often quite possible to use relaxed filters, thus eliminating fewer samples. However, in doing so, one must be able to assess that the ability to detect CNVs is not significantly different in "noisier" samples, otherwise biases may be introduced in the results. Here, we show how to produce a series of plots that help identify such potential issues in the results of the protocol, and briefly discuss the interpretation of each one. Specific problems and solutions are then highlighted in the troubleshooting section.

Protocol steps

1. Setup. This support protocol is meant to help evaluate the performance of the CNV calling protocol. First, start an R session. From `$workingdir`:

```
singularity exec ibpcnv.simg R
```

   Load the visual inspection results, e.g., assuming the visual inspection results were saved as `visual_output_ALL.txt` in `$workingdir`:

```
library(data.table); vi_res <- fread("visual_output_ALL.txt")
```

2. Create the plots. The function `qc_plots_cnvs()` will create three plots for the main filtering arguments and more supporting ones. Here we will discuss briefly only the main one, for the complete discussion refer to the Commentary. As an example, to create the QC plots for the CNVs in all loci, simply run:

```
library(QCtreeCNV); qc_plots_cnvs(vi_res, "all_loci")
```

   This will create the folder `all_loci` in the working directory and save all QC plots in it. Note that the plots show the results regarding only the CNVs marked as true (`visual_ouput == 1`).

3. Interpret the results. Figure 4 shows a good and a bad example of the two plot types. Also seeUnderstanding Results for more discussion.

   a. Plot 1 (example in Figure 4A) shows the CNV prevalence in different LRRSD chunks (low, medium, high), separated for deletions and duplications. It illustrates the ability to detect true CNVs in different groups of samples from the noise perspective (high LRRSD can be considered the main indication of a noisy sample). Ideally, the prevalence should not strongly differ across the three groups, especially the one with higher LRRSD compared to the rest. A significant change means that the ability to detect CNVs is significantly affected in noisy samples, usually becoming lower. This means the LRRSD filter threshold may need to be increased.

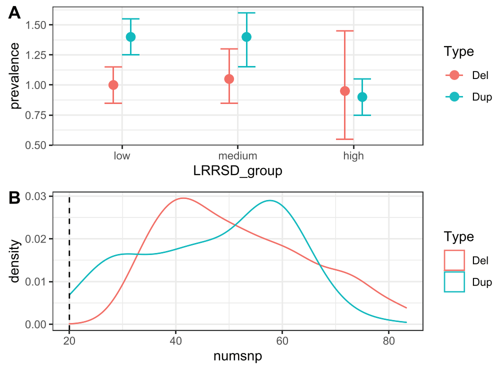


Figure 4 A good and a problematic example of QC plots 1, 4, and 5 (panel A), and 2 and 3 (panel B). In both plots, the deletion in light red represent the ideal situation, and the duplication in light blue represent the problematic situation. (**A**) the group of samples with high LRRSD have fewer true duplications than the other two groups, and that the confidence interval is relatively small. (**B**) The threshold value we selected for numsnp appears to be cutting the left tail of the distribution for the true duplications.

On the other hand, if a strong LRRSD filter was used and plot 1 shows very high consistency, the user may want to explore lowering it to possibly exclude fewer samples from the analysis.

b. Plots 2 and 3 (example in Figure 4B) show the distribution of the number of SNPs and overlap proportion with the locus per call for true CNVs, numsnp, and overlap, respectively. Both these measures are used as filters when selecting putative CNV calls. Ideally, the distribution should not seem to be "cut" at the threshold values for numsnp and overlap. If that is the case, it might indicate that some potential true CNVs are being filtered out, and it may be worth trying to relax the filters. This is especially important for very rare CNV loci, where a small increase in carriers can have an impact on power. Note that plot 2 may be more meaningful when used on individual loci or a group of loci with very similar marker coverage, since it is the absolute number of markers of each call. If different thresholds were used for different loci, they must be treated separately to obtain meaningful results.

c. Plots 4 and 5 can be interpreted in the same way as plot 1 (Fig. 4A) but regarding two other noise measures, BAF drift and GCWF, respectively. They can be considered secondary since these dimensions are less prone to affect the CNV detection accuracy.

4. Deal with possible detection bias. If the dataset includes sample groups selected differently (e.g., cases and controls), it may be a good idea to analyze them separately. The CNV prevalence is often expected to differ in different groups (e.g., some recurrent CNVs are more frequent in neuropsychiatric case groups than in population controls); thus, combining them may lead to confusion in the interpretation of the QC plots, especially if LRRSD distribution also differs across those groups. Possible problems in these plots are described in the last two points of the troubleshooting section. Assuming the `sample_IDs` for one group are in vector `groupA QC` analysis can be run separately with:

```
qc_plots_cnvs(vi_res[sample_ID %in% groupA,], "groupA")
```

5. Explore an individual locus or groups of loci. The process described in previous steps 2 and 3 can be applied to groups of loci as well as to individual loci. This is useful especially when there seems to be some problems, but only in a fraction of the loci or in a particular one. For example, to look only at the results for loci "A" and "B" run:

```
qc_plots_cnvs(vi_res[locus %in% c("A", "B"),], "loci_A_B")
```

As a rule, it can be helpful to divide the loci of interest into two or three groups based on length (e.g., small/large) and inspect the QC plots for the groups in addition to the ones for all loci. Groups can also be based on prevalence (very rare/not very rare) or other measures. The general idea is that, while looking at all loci at the same time can give an overall impression of the results quality, it can also mask problems linked to one or very few loci. Smaller groups can help identify those, if any, and the QC plot for individual loci can pinpoint the actual problem.

SUPPORT PROTOCOL 2

INSTALL THE NECESSARY SOFTWARE

We provide a docker image containing all software required to run the protocol on Docker Hub at *https://hub.docker.com/r/sinomem/docker_cnv_protocol*. This container has the following software installed: htslib (Bonfield et al., 2021; Danecek et al., 2021) (1.14), PennCNV (Wang et al., 2007) (1.0.5), R (R Core Team, 2018) (4.1.2), and DeepEYE2, as well as several R packages, including `data.table` (Dowle et al., 2020), `fpc` (Hennig, 2020), and `VariantAnnotation` (Obenchain et al., 2014).

Regarding the in-house software used in this protocol, the QCtreeCNV R package is available on GitHub at *https://github.com/SinomeM/QCtreeCNV*. Instructions to install the package are given in the README. DeepEYE2 is available at *https://github.com/XabierCS/eyeCNV*. All other scripts are collected in a GitHub repository at *https://github.com/SinomeM/IBPcnv*.

The statistical programming language R is available at *https://www.r-project.org/*. Tabix is part of the HTSlib suite, together with SAMtools and BCFtools. It can be obtained at *https://www.htslib.org/download/*. PennCNV is the *de facto* standard in CNV calling from array data, in particular Illumina. It is available at *http://penncnv.openbioinformatics.org/en/latest/user-guide/download/*. In the following section, we detail how to install the docker/singularity image and use it to run the protocol.

Protocol steps

1. Setup. Throughout the protocol, `$workingdir` is used to indicate the main project folder. This folder will contain all the input and output files. For simplicity, the user can define an environmental variable:

```
export workingdir=/path/to/workingdir
```

2. Install singularity. The software should be already installed on most modern HPCs. If not, users should ask the system administrator to install it for them. To install it on a Linux workstation, one should follow the official instructions available at *https://sylabs.io/guides/3.0/user-guide/installation.html*. A precompiled RPM package is available at *https://dl.fedoraproject.org/pub/epel/8/Everything/x86_64/Packages/s/* and the program alien can be used to convert it to DEB (*https://github.com/apptainer/singularity/issues/5390*). Finally, in systems where the use of conda environments is encouraged or enforced, it should be possible to use the conda package at *https://anaconda.org/conda-forge/singularity*.

3. Download the provided container image. We provide the container on DockerHub, and it can be pulled directly by singularity. To do so, first move in the desired folder (`cd $workingdir`) and then type:

```
singularity pull ibpcnv.simg docker://sinomem/docker_cnv_protocol:latest
```

Note that the protocol expects the image to have the suggested name (`ibpcnv.simg`) and be stored (or linked) in the main working directory (`$workingdir`). If singularity fails with an error regarding the `/tmp` folder, it may help to set the environmental variables `SINGULARITY_TMPDIR` and `SINGULARITY_CACHEDIR` to some non-protected location (such as `~/tmp` or `$workingdir/tmp`).

4. Download the IBPcnv repository. We provide two versions, one that uses SLURM (srun/sbatch) and one that uses PBS (qsub). They can be obtained running:

```
wget https://github.com/SinomeM/IBPcnv/archive/refs/heads/master.zip&&\
    unzip master.zip && mv IBPcnv-masterIBPcnv && rm master.zip
# or
wget https://github.com/SinomeM/IBPcnv/archive/refs/heads/pbs.zip&&\
unzip pbs.zip && mv IBPcnv-pbs IBPcnv && rm pbs.zip
```

   for the SLURM and PBS versions respectively. For convenience, we can define the environmental variable `$ibpcnvdir:`

```
export ibpcnvdir=${workingdir}/IBPcnv
```

5. Add the SLURM/PBS account if needed. The four scripts that use the job scheduler (03, 03.1, 03.2, and 04 in `$ibpcnvdir/penncnv_pipeline/`) are designed to be easily edited if the system requires the use of a specific account name. Throughout the text we provide both versions of the commands when run interactively.

6. Run the protocol. All scripts assume that the singularity image is used. The pipeline in Basic Protocol 1 is designed to take advantage of the SLURM or PBS job scheduler, depending on which branch of the IBPcnv repository was chosen.

7. Docker versus singularity. To download the container using docker we run:

```
docker pull sinomem/docker_cnv_protocol:latest
```

   Then, to print the tabix help page using singularity image or docker we type respectively:

```
singularity exec /path/to/ibpcnv.simg tabix --help
# or
docker run sinomem/docker_cnv_protocol:latest tabix --help
```

   One of the main differences is that, conveniently, singularity automatically mounts several file storage locations to the container while Docker does not. Moreover, in order to use docker a user needs to be added to the docker group and this process may require sudo permissions. Refer to the docker documentation for further details, *https://docs.docker.com/*.

COMMENTARY

Background Information

The main scope of this protocol is to provide a framework to enable the creation of high-quality datasets of recurrent CNVs from large-scale SNP-genotyped collections. We condense years of experience in the field into easy-to-use pipelines and an extensive set of best practices, providing strong default values and great customizability for all major parameters. The framework is composed of four main elements: a docker/singularity container, a standardized PennCNV pipeline, an R package for data handling and cleaning, and a graphical interface to perform visual validation of the CNVs. The singularity image (Support Protocol 1) contains all necessary software in the correct version, and the whole protocol is designed to use it. Using a container, software installation is not a variable in the process, and instructions are always ensured to work as intended. Distributing it in both docker and singularity formats, we have made it accessible to most systems. The CNV calling pipeline (Basic Protocol 1) has multiple qualities. It is meant to work on most HPC systems, freeing researchers from the task of designing *ad hoc* pipelines, thus providing standardization and user friendliness. Moreover, it integrates several seemingly simple steps that required years of expertise to be refined, and can have a huge impact on the final results. The R package (Basic Protocol 2) serves multiple functions. It standardizes putative CNV selection and filtering. It also implements a novel, more advanced filtering algorithm designed to reduce the number of putative CNV to inspect, but with a strong focus on minimizing false negatives.

Quality control after visual validation (Support Protocol 2) is also handled by this package. Finally, the graphical interface is a very powerful program that can be used not only to validate CNVs in fixed loci, but to manually refine the boundaries of a CNV call (feature not showcased in this manuscript). Together with the possibility of using GCWF-adjusted values for LRR, in our opinion, this makes it superior to any similar solution.

While CNVs can be detected and studied with different approaches, we designed the protocol with a clear focus, namely effective and precise detection of rare recurrent CNVs in specific and characterized genomic loci using large collections of SNP-array data. Nonetheless, we believe the graphical interface, as well as the PennCNV pipeline, also provide a high value when used on their own. Examples include when analyzing small collections or for studies not focused on recurrent CNVs. To our knowledge, no comparable software packages are available. In contrast, the novel filtering algorithm we developed (step 4 of Basic Protocol 2) is narrower in its approach. While the use of the R package to manage and select the CNV call from PennCNV is advised for all users, `qctree()` has been specifically constructed in order to deal with a high number of calls, where the user knows (or suspects from exploratory analysis) that a good portion of the putative calls are false. This can be due to noise, if the collection has for example a high LRRSD or GCWF, or due to the presence of smaller irrelevant CNVs within the locus of interest. The algorithm can deal with both of these problems and, by default, will behave quite conservatively, meaning that for a call to be filtered out, two or more measures need to decisively point towards the exclusion. In other words, in a dubious situation, the decision should always be made by the analyst. In accord with this model, visual validation always needs to be performed. If the number of putative calls to inspect is manageable, and/or if it is suspected that no large groups of false positives are present, then it is advised to skip the step and perform visual validation directly.

Design Choices

We made several design choices during the creation of the protocol; we discuss the major ones in this section. In Basic Protocol 1, the pipeline is structured to be very easy to use even for inexperienced analysts while including several advanced steps that would require some level of expertise. However, all design choices we made can be considered fairly standard and coherent with current standard practice. In contrast, Basic Protocol 2 makes use of tabix-indexed intensity files (introduced in Basic Protocol 1), which is a novel idea in CNV studies. Tabix-indexed files provide an enormous speed advantage when accessing specific sections of the file (such as a genomic region in a BED file), by avoiding the need to load the entire file into the computer RAM or to screen the file from the beginning. This advantage is used throughout all functions that need access to raw data in QCtreeCNV, as well as to create the CNV plots on-the-fly in DeepEYE. Of note, when creating the indexed file, we also add an additional column containing the GCWF-adjusted values for LRR. The GCWF-adjusted LRR value can be used in DeepEYE, meaning that, in contrast to any other methods to our knowledge, the user can actually “see” the raw data trends in the same way as PennCNV did. This is because GCWF correction is performed by PennCNV when calling the CNVs; however the adjusted values are usually not stored and are thus not accessible by the user. This whole process comes with a cost as well, as basically a full copy of each intensity file needs to be generated (CPU time and high I/O on the system) and stored (disk space). We strongly believe that the advantages exceed the costs in disk space and propose this format as the new standard for all future programs that deal with SNP-array data in the form of intensity files. Another important choice was to use the R data.table structures and grammar in QCtreeCNV internal functions as well as in Basic Protocol 2. This package provides several advantages, including implicit parallelization of each operation if multiple cores are available, as well as a simple and extremely powerful grammar. We believe that in small and very specific R packages like ours, there is no need to avoid relying on external dependencies. Moreover, the aforementioned advantages (implicit parallelization in particular) vastly outweigh the small added complexity in the protocol commands. We also state clearly that all objects can be reverted back to simple data.frame after completion if the user wishes to continue with downstream analysis. Moreover, we believe that exploratory analysis is a necessary part of this kind of study, and that no protocol can replace them. For this reason and considering the small number of commands and the large number of tuneable parameters, we show how to run the QCtreeCNV pipeline in an interactive R session, and we do not provide an R script. Testing multiple settings and exploring the results is strongly encouraged. An RStudio session may be ideal for some users, but we choose to not include RStudio in the container, as it would make it more complex and heavier. We note that all R packages required to run the Basic Protocol 2 are listed in Support Protocol 2. Finally, we show how to run the protocol as a whole in all loci of interest, but this may be limiting in some cases. One example is when the user is using a list of loci where one or a few are very different from the others (e.g., very small or covered by very few markers). In this case, it may be beneficial to treat a particular locus or group of loci differently from the others with regard to some critical parameters such as minimum number of SNPs, length, or overlap. This applies to Basic Protocol 2 and Support Protocol 1, as all samples must be treated the same in Basic Protocol 1.

Technical Details

In this section, we provide more technical details, in particular about CNVRs and the advanced filtering function `qctree()`. CNV regions can be defined in different ways depending on the scope, but in general they are regions of the genome where very similar CNVs are present, across samples. Recurrent CNV loci can be considered CNVRs, but they also can be smaller. In this protocol, we use CNVRs only to extract false positives from the putative CNVs. In practice, the function `cnvr_fast()` performs 2D clustering on the normalized center position and length of all putative CNVs in each locus of interest (separately). If multiple subgroups are found, they are then categorized and potentially treated differently in the `qctree()` function, meaning that CNVs belonging to large CNVRs will be harder to mark as false positives, CNVs belonging to small CNVRs will be easier to exclude, and those from medium CNVRs will receive an in-between treatment. Supplementary Figure 1 in Supporting Information provides a schematic representation of the filtering pipeline. Step 1 is applied sample-wise and will exclude any QC outlier, based on LRR standard deviation, BAF drift, and GC waviness factor. It is the more canonical step and the one more accessible to customization. Step 2 separates CNV calls based on the CNVR they belong to. The reasoning is that CNV calls that are very similar to the locus of interest should almost always be passed to visual inspection, while CNVs from smaller CNVRs may be filtered more aggressively. Step 3 further separates CNVs based on CNVRs frequency, sending the putative calls either directly to step 5 or first to step 4. All the dimensions on which step 4 and 5 are applied are derived directly from the intensity files, and thus do not depend on PennCNV processing. Broadly speaking, they measure how LRR and BAF trends (see Supplementary Figure 2 in Supporting Information for a visual interpretation on how BAF is used in this context) behave and are compared to threshold values that we derived from a broad collection of human-validated true and false CNVs calls (~15,000) in thirty different recurrent loci. In short, from step 3, if the CNVR is small as well as frequent, it may indicate the presence of a smaller true recurrent CNVs locus within the main one. Thus, CNVs with these characteristics proceed to step 4, where an aggressive filter is applied to test whether the raw data is consistent with the assumption. In contrast, if the CNVR is small but infrequent, it is likely that the CNVs are either noise or true CNVs called only partially by PennCNV. Thus, a lighter filter should be sufficient to separate the two groups.

Critical Parameters

In this section, we briefly discuss the most important parameters of the entire protocol. In Basic Protocol 1 the user can mainly change the parameters of step 3b and 4d, namely the SNPs filtering and the first round of CNV call stitching. In 3b, it is suggested to keep the default value of `minMAF` (the minimum value of Minor Allele Frequency); however, users can set the last parameter to FALSE if they feel too many SNPs are excluded because of duplicated SNP IDs or positions. In Basic Protocol 2 the critical parameters are in steps 3c and 4a. Step 3c parameters control how the putative CNVs are selected; `minsnp, maxgap`, and `minoverlap` control the required minimum number of markers per call (applied after stitching), the maximum gap allowed before stitching two calls, and the minimum amount of overlap between CNV and locus of interest. Step 4a consists of the advanced filtering function, as already stated the most accessible parameters are the sample-wise filters `maxLRRSD`, `maxBAFdrift`, `maxGCWF`, `minGCWF`. These control, respectively, the maximum LRR standard deviation and B allele frequency drift, and the GCWF range allowed. The first two can be considered the major knobs the user can turn when managing the noise-to-signal ratio, other than the `qctree()` function itself.

Table 1 Troubleshooting

Problem	(Possible) cause	Suggested solution
Errors in Support Protocol 2	No formal software installation is required thus the most likely problems are: required programs (singularity, wget, unzip) are missing and no internet connection.	Refer to the official website for singularity installation. If wget is not installed on the system, one may use curl. Another alternative is to use git clone. When using git, remember to switch to the PBS branch if necessary!
The computing system does not provide a job scheduler (PBS or SLURM)	While common practice in large HPCs, smaller systems may not use a job scheduler to manage the computing load.	Only a section of Basic Protocol 1 requires a job scheduler (step 4), the actual CNV calling pipeline. In a small dataset, a quick solution is to manually run each batch in a separate session ($ibpcnvdir/penncnv_pipeline/03_1_per_wave.sh), avoiding intra-batch parallelization (with small modifications of $ibpcnvdir/penncnv_pipeline/03_2_cnv_calling.sh).
Bash and PBS/SLURM errors in Basic Protocol 1	The main cause of errors are erroneous paths or typos in naming the required files.	Check multiple times that all required files are in the expected format: in particular the file name, field separator (TAB), and columns header. Check also that all paths are complete (starting from ‘/’) and do not include any links (as an example going through the user home directory).
PBS/SLURM job scheduler throws errors regarding incorrect “account” or “partition”	To interact with the job scheduler, one may be required to use a specific account or queue.	As described in step 5 of Support Protocol 2, all scripts that use PBS/SLURM are designed to be easily modified in this case. Also, all commands in the protocol that used srun or qsub must be modified accordingly. For srun, adding the flag –account=account_name.
Step 4c of Basic Protocol 1 shows some samples were not processed by PennCNV	Samples are called in chunks (within each batch) of ∼200 per job. Some jobs may fail for multiple reasons.	A simple solution is to rerun the chunk or the whole batch manually, as shown in step 4c.
Step 4c of Basic Protocol 1 shows some samples were not processed by PennCNV	A second reason some samples may fail processing is that the intensity file is in the wrong format or corrupted. For convenience reasons, step 2b will check that around 25% of all intensity files exist, but only 100 will be opened to check for the correct format. It is possible that some files have errors or are missing.	Check the format of all intensity files. Lines 19-37 of $ibpcnvdir/penncnv_pipeline/01_preprocess.R can be used as template.
Errors in Basic Protocol 2 R session, such as “file not found”	The commands shown in the protocol require precise file naming and formats.	Check multiple times to make sure that the paths and file names used correspond to the commands run (e.g., some output files may not have standard names), and change the suggested commands when necessary.

(*Continued*)

Table 1 Troubleshooting, *continued*

Problem	(Possible) cause	Suggested solution
QC plot 1, 4 or 5 show problems in Support Protocol 1	As described in the protocol, these plots are meant to show if the ability to detect CNVs is affected by the noise. The high noise category is the most likely to show problems, i.e., a different prevalence than the other groups. The relevant filtering happens in step 4b of Basic Protocol 2.	This must be interpreted with some care, as often the high-noise group is also the smaller one. If the numbers are too low and the estimate is unstable (especially for a single locus), it is quite hard to recommend a specific action. In general, here it is a matter of balancing losing samples with a stronger filter and getting potentially biased results from too-noisy data. Key parameters to remove the noisy samples are explained in step 4b of Basic Protocol 2.
QC plots 2 or 3 show problems in Support Protocol 1	As described in the protocol, these plots are meant to show if the minsnp and overlap filters are in step. The relevant filtering happens in step 3c of Basic Protocol 2.	The distribution can take various shapes but, with high numbers, it should be somewhat Gaussian. The main problem is if the distribution appears to be "cut" at the threshold value for the measure of interest. This is better observed with a single locus or a group of similar loci. If such a problem presents, it means that some true CNVs were "tagged" by very small calls or calls only very partially overlapping the locus of interest. The solution is to try reducing the relevant threshold value. However, this could lead to a higher number of false positives, even though qctree() in the following step should remove most of them.

Troubleshooting

Table 1 lists problems that may arise with the protocols in this article along with their possible causes and solutions.

Understanding Results

The ultimate results of this protocol are a file containing visually inspected CNVs (Basic Protocol 2, step 5e and 6) and a collection of QC plots describing the prevalence of those CNVs that were deemed true across bins of increasing values of sample-quality-relevant metrics (LRRSD, BAFDRIFT, GCWF) as well as the distribution of those true calls with respect to measures relevant to CNV detection sensitivity (number of SNPs per call, and the level of CNV call overlap with the test locus, see Support Protocol 1). In this section, we will briefly describe both. The output of Basic Protocol 2 can be considered the result of the entire methodology; it will contain all visually inspected CNVs. Depending on the specific application the user is working on, it may be more or less important to obtain the highest possible precision. In situations where maximum precision is required, it can be helpful for more than one analyst to inspect the same set of CNVs (Basic Protocol 2, steps 5-6) and then to merge the results, re-analyzing together any CNVs where there was no consensus among the analysts. In any case, at the end of the process, most users will want to have a final table containing only the true validated CNVs. This can be achieved, in R, as shown in Support Protocol 2, step 1, to load the table, followed by `vi_res_true <- vi_res[visual_output == 1,]` to select only the true ones. The object `vi_res_true` can then be saved in the preferred disk location. Regarding the QC plots, we will describe the provided example, Figure 4. The last two points of the troubleshooting section provide some guidance on how to interpret such plots. Following them we can see in both Figure 4A and 4B that the prevalence and call size distribution of the verified deletions look unproblematic, while those of the verified duplications do not. In Figure 4A, the deletion prevalence is consistent with respect to LRR SD. The wider prevalence error bars observed for the group with the highest LRR SD reflects the fact that relatively few samples have LRR SD values in this range. In contrast, in Figure 4A we observe a lower duplication prevalence in the high LRR SD group than in the other two LRR SD groups. This indicates that the duplication detection sensitivity in this LRR-SD group may be reduced, as prevalence should not vary depending on LRR SD. In this example, the duplication prevalence error bars are narrow, indicating that the lower prevalence in this group may not be a chance finding (i.e., that we are not detecting all true duplications in samples within this LRR SD range). The solution depends on the application. In a discovery study, no adjustment may be needed; however, in a prevalence study, the filtering cut-off for LRR SD should be lowered, to ensure that we are able to detect true deletions and true duplications with the same efficiency. In Figure 4B, the call size distribution of the deletions is unproblematic; with a sample large enough the density curve should approach a Gaussian distribution. In contrast, the density curve for duplications appears to be cut at the threshold value of `numsnp`, indicating that some true duplications may have been filtered out in step 3c of Basic Protocol 2. The filtering value should in this case be lowered to ensure optimal capture of true CNVs at the locus in question.

Time Considerations

The time required to run the protocol depends on the amount of sample the user needs to process, as well as the coverage of the specific SNP-array used. A collection of approximately 500,000 samples should take less than 72 hr to complete Basic Protocol 1. However, given the fact that most steps make strong use of parallel computation, this estimate is strongly dependent on the actual computing capabilities available (i.e., the number of nodes and amount of CPU cores and RAM per node) and thus the number of concurrent jobs possible. Basic Protocol 2 (except step 5) and Support Protocol 1 are exploratory in nature; however, they should not require more than 1 day of analyst work each. Step 5 of Basic Protocol 2 (i.e., the visual confirmation) is the most human-labor-intensive step of the entire protocol. Based on our experience, an expert analyst needs between 2 and 10 s on average to evaluate each putative call.

Acknowledgments

Our research was supported by the NIH (R01 MH124789-01), but the funding source had no involvement in the design of the protocol, nor in collection, analysis, or interpretation of data. While protocol is designed as a general software, it was initially conceptualized in conjunction with CNV analysis of data from The Lundbeck Foundation Initiative for Integrative Psychiatric Research (iPSYCH).

Author Contributions

Simone Montalbano: conceptualization, data curation, methodology, resources, software, validation, visualization, writing original draft, writing review and editing; **Xabier Sánchez**: conceptualization, data curation, methodology, software, writing original draft, writing review and editing; **Morteza Vaez**: data curation, resources; **Dorte Helenius**: supervision; **Andres Ingason**: conceptualization, data curation, funding acquisition, resources, supervision, writing review and editing; **Thomas Werge**: conceptualization, funding acquisition, supervision.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

We provide a docker image containing all software required to run the protocol on Docker Hub at *https://hub.docker.com/r/sinomem/docker_cnv_protocol*. The protocol is designed to handle large collections of human genetic data. For privacy reasons, all plots shown were created using simulated data. The QCtreeCNV R package is available on GitHub at *https://github.com/SinomeM/QCtreeCNV*, and the code to generate Figure 4 is located in `tmp/dev.R`. DeepEYE is available at *https://github.com/XabierCS/eyeCNV*; toy data (used in Figure 3) is available in `toy-data/`.

Literature Cited

Bonfield, J. K., Marshall, J., Danecek, P., Li, H., Ohan, V., Whitwham, A., … Davies, R. M. (2021). HTSlib: C library for reading/writing high-throughput sequencing data. *GigaScience*, *10*(2), giab007. doi: 10.1093/gigascience/giab007

Calle Sánchez, X., Helenius, D., Bybjerg-Grauholm, J., Pedersen, C., Hougaard, D. M., Børglum, A. D., … Werge, T. (2021). Comparing copy number variations in a Danish case cohort of individuals with psychiatric disorders. *JAMA Psychiatry*, *79*(1), 59–69. doi: 10.1001/jamapsychiatry.2021.3392

Colella, S., Yau, C., Taylor, J. M., Mirza, G., Butler, H., Clouston, P., … Ragoussis, J. (2007). QuantiSNP: An Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Research*, *35*(6), 2013–2025. doi: 10.1093/nar/gkm076

Crawford, K., Bracher-Smith, M., Owen, D., Kendall, K. M., Rees, E., Pardiñas, A. F., … Kirov, G. (2019). Medical consequences of pathogenic CNVs in adults: Analysis of the UK Biobank. *Journal of Medical Genetics*, *56*(3), 131–138. doi: 10.1136/jmedgenet-2018-105477

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., … Li, H. (2021). Twelve years of SAMtools and BCFtools. *GigaScience*, *10*(2), giab008. doi: 10.1093/gigascience/giab008

Dowle, M., Srinivasan, A., Gorecki, J., Chirico, M., Stetsenko, P., Short, T., … Eddelbuettel, D. (2020). data.table: Extension of 'data.frame' (Version 1.13.2). *Retrieved from*, https://CRAN.R-project.org/package=data.table

Driscoll, D. A., Spinner, N. B., Budarf, M. L., McDonald-McGinn, D. M., Zackai, E. H., Goldberg, R. B., … Jones, M. C. (1992). Deletions and microdeletions of 22q11.2 in velo-cardio-facial syndrome. *American Journal of Medical Genetics*, *44*(2), 261–268. doi: 10.1002/ajmg.1320440237

Hennig, C. (2020). fpc: Flexible Procedures for Clustering (Version 2.2-9). *Retrieved from*, https://CRAN.R-project.org/package=fpc

Malhotra, D., & Sebat, J. (2012). CNVs: Harbingers of a rare variant revolution in psychiatric genetics. *Cell*, *148*(6), 1223–1241. doi: 10.1016/j.cell.2012.02.039

Obenchain, V., Lawrence, M., Carey, V., Gogarten, S., Shannon, P., & Morgan, M. (2014). VariantAnnotation: A Bioconductor package for exploration and annotation of genetic variants. *Bioinformatics*, *30*(14), 2076–2078. doi: 10.1093/bioinformatics/btu168

R Core Team. (2018). *R: A Language and Environment for Statistical Computing*.

Sharp, A. J., Locke, D. P., McGrath, S. D., Cheng, Z., Bailey, J. A., Vallente, R. U., … Eichler, E. E. (2005). Segmental duplications and copy-number variation in the human genome. *American Journal of Human Genetics*, *77*(1), 78–88. doi: 10.1086/431652

Stankiewicz, P., & Lupski, J. R. (2002). Genome architecture, rearrangements and genomic disorders. *Trends in Genetics*, *18*(2), 74–82. doi: 10.1016/s0168-9525(02)02592-1

Stefansson, H., Rujescu, D., Cichon, S., Pietiläinen, O. P. H., Ingason, A., Steinberg, S., … Stefansson, K. (2008). Large recurrent microdeletions associated with schizophrenia. *Nature*, *455*(7210), 232–236. doi: 10.1038/nature07229

Stefansson, H., Meyer-Lindenberg, A., Steinberg, S., Magnusdottir, B., Morgen, K., Arnarsdottir, S., … Stefansson, K. (2014). CNVs conferring risk of autism or schizophrenia affect cognition in controls. *Nature*, *505*(7483), 361–366. doi: 10.1038/nature12818

Sudmant, P. H., Rausch, T., Gardner, E. J., Handsaker, R. E., Abyzov, A., Huddleston, J., … Korbel, J. O. (2015). An integrated map of structural variation in 2,504 human genomes. *Nature*, *526*(7571), 75–81. doi: 10.1038/nature15394

Turner, D. J., Miretti, M., Rajan, D., Fiegler, H., Carter, N. P., Blayney, M. L., … Hurles, M. E. (2008). Germline rates of de novo meiotic deletions and duplications causing several genomic disorders. *Nature Genetics*, *40*(1), 90–95. doi: 10.1038/ng.2007.40

Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J., Grant, S. F. A., … Bucan, M. (2007). PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research*, *17*(11), 1665–1674. doi: 10.1101/gr.6861907

Weischenfeldt, J., Symmons, O., Spitz, F., & Korbel, J. O. (2013). Phenotypic impact of genomic structural variation: Insights from and for human disease. *Nature Reviews Genetics*, *14*(2), 125–138. doi: 10.1038/nrg3373

Published in partnership with CEGMR, King Abdulaziz University

https://doi.org/10.1038/s41525-024-00450-8

Analysis of exonic deletions in a large population study provides novel insights into NRXN1 pathology

Simone Montalbano[1,2], Morten Dybdahl Krebs[1,2], Anders Rosengren[1,2], Morteza Vaez[1,2], Kajsa-Lotta Georgii Hellberg[1,2], Preben B. Mortensen[2,3], Anders D. Børglum[2,4,5], Daniel H. Geschwind[6,7,8,9], iPSYCH Investigators*, Armin Raznahan[10], Wesley K. Thompson[2,11], Dorte Helenius[1,2], Thomas Werge[1,2,12] & Andrés Ingason[1,2]

The *NRXN1* locus is a hotspot for non-recurrent copy number variants and exon-disrupting *NRXN1* deletions have been associated with increased risk of neurodevelopmental disorders in case-control studies. However, corresponding population-based estimates of prevalence and disease-associated risk are currently lacking. Also, most studies have not differentiated between deletions affecting exons of different *NRXN1* splice variants nor considered intronic deletions. We used the iPSYCH2015 case-cohort sample to obtain unbiased estimates of the prevalence of *NRXN1* deletions and their associated risk of autism, schizophrenia, depression, and ADHD. Most exon-disrupting deletions affected exons specific to the alpha isoform, and almost half of the non-exonic deletions represented a previously reported segregating founder deletion. Carriage of exon-disrupting *NRXN1* deletions was associated with a threefold and twofold increased risk of autism and ADHD, respectively, whereas no significantly increased risk of depression or schizophrenia was observed. Our results highlight the importance of using population-based samples in genetic association studies.

Larger genomic deletions in the *NRXN1* locus have been associated with a highly increased risk of mental disorders and, in particular, schizophrenia. However, the locus is known to harbour highly heterogeneous CNVs (Copy Number Variations, deletions and duplications) and, moreover, no population-based estimates of risk are available. Here, we use the iPSYCH2015 case-cohort sample to investigate the population prevalence and phenotypic consequences of specific types of deletions within the locus.

Neurexins are a family of highly conserved transmembrane proteins strongly involved in the development and function of neuronal synapses[1]. Like all mammals, humans possess three genes encoding different neurexin proteins (NRXN1-3)[2]. All three genes encode two main protein isoforms, alpha and beta[1], and are almost exclusively expressed in neuronal tissue[3,4]. Notably, hundreds of splicing isoforms are expressed in humans and mice, many of which are specific to certain neuronal cell types[1,5,6]. Neurexin proteins are expressed by neurons at the presynaptic nerve terminal and their expression peaks around birth[1]. Among other ligands, neurexins bind to the calcium/calmodulin-dependent serine protein kinase (CASK) scaffolding molecules, contributing to the coupling of Ca^{2+} channels to synaptic release machinery[1,7].

NRXN1 is a 1.3 Mbp gene located on the short arm of chromosome 2 (GRCh38:49,918,503–51,225,575)[8]. Among the three neurexin genes, *NRXN1* is the most studied with respect to association with disease[9–11]. Multiple case-control studies have associated exonic deletions with increased risk of neurodevelopmental disorders, including schizophrenia

[1] Institute of Biological Psychiatry, Mental Health Services, Copenhagen University Hospital, Roskilde, Denmark. [2] The Lundbeck Foundation Initiative for Integrative Psychiatric Research (iPSYCH), Copenhagen and Aarhus, Denmark. [3] National Centre for Register-based Research, Aarhus University, Aarhus, Denmark. [4] Department of Biomedicine – Human Genetics and the iSEQ Center, Aarhus University, Aarhus, Denmark. [5] Center for Genomics and Personalized Medicine, Aarhus, Denmark. [6] Department of Neurology, University of California, Los Angeles, CA, USA. [7] Department of Human Genetics, University of California, Los Angeles, CA, USA. [8] Center for Autism Research and Treatment, Semel Institute, David Geffen School of Medicine, University of California, Los Angeles, CA, USA. [9] Program in Neurobehavioral Genetics, Semel Institute, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, USA. [10] Section on Developmental Neurogenomics, Human Genetics Branch, National Institute of Mental Health Intramural Research Program, Bethesda, MD, USA. [11] Laureate Institute for Brain Research, Tulsa, OK, USA. [12] Department of Clinical Medicine, University of Copenhagen, Copenhagen, Denmark. *A list of authors and their affiliations appears at the end of the paper. e-mail: andres.ingason@regionh.dk

(odds ratio (OR): 4.5; 95% CI: 2.0–10.9)[12], autism spectrum disorder (OR: 7.2; 95% CI: 0.9–326)[13], attention-deficit/hyperactivity disorder (OR: 4.68; CI95%: 1.82–10.64)[14], depression (OR: 2.01; CI95%: 1.18–3.19)[15], intellectual disability and/or developmental delay (OR: 8.14; 95% CI: 2.91–22.7)[16], epilepsy (OR: 9.91; 95% CI: 1.92–51.1)[17], and Tourette Syndrome (OR: 20.3; 95% CI: 2.6–156)[18]. Deletions in the 5′ end of the gene are more commonly observed compared to the rest of the gene[9]. To our knowledge, duplications and intronic deletions have not been strongly associated with disease risk in previous studies and, in general, appear to be far less studied than exonic deletions of *NRXN1*.

CNVs in the *NRXN1* locus are non-recurrent, meaning that CNVs result from unrelated de novo mutations which do not share fixed breakpoints, and their mutational mechanism is different from that observed in non-allelic homologous recombination (NaHR) mediated by low-copy repeat (LCR) sequence elements[19]. One possible explanation for this genomic instability is that the *NRXN1* locus, similarly to other large genes, is a late replicating region and therefore more prone to mutations resulting from stress-induced replication errors[20].

As was also the case for rare recurrent CNVs (such as 22q11.2 deletions and 16p11.2 duplication) *NRXN1* deletions were originally associated with high risk of disease from single case studies or small collections of cases[21–24], followed by larger case-control studies also based on highly selected samples (e.g., cases with severe or long-term illness and controls screened for any family history of mental illness)[11,13,16–18,25]. However, recent research on recurrent CNV loci in larger and more population-representative study samples suggests that associations obtained using selected case-control samples tend to be biased toward an overestimation of the disease risk, owing largely to an underestimation of the prevalence of recurrent CNVs in the general population[26–28].

In this study, we use the unique design of the iPSYCH2015[29] case-cohort study to provide population-representative estimates of the prevalence of *NRXN1* deletions, and the associated risk of attention-deficit/hyperactivity disorder (ADHD), major depressive disorder (MDD), schizophrenia spectrum disorder (SSD), autism spectrum disorder (ASD), and bipolar disorder (BPD). We assess the risk of any deletion in the *NRXN1* locus as well as that of different subgroups (including non-exonic ones). Moreover, we show that a significant proportion of intronic deletions in the locus is segregating in the population and may be associated with an increased risk of some psychiatric disorders.

Methods

Study design, phenotypes, and genotyping

This study is based on the iPSYCH2015 case-cohort sample[29], an expanded version of iPSYCH2012, which has been previously described in detail[30]. In brief, the base population is defined as all 1,657,449 singleton births that occurred in Denmark between May 1, 1981, and Dec 31, 2008, who were alive and residing in Denmark on their first birthday and had a mother registered in the Danish Civil Registration System[31]. From the base population all persons who received a diagnosis of a major psychiatric disorder (as specified below) no later than Dec 31, 2015, were included in the case sample, $N = 92{,}531$ individuals. Then, a randomly selected population-representative cohort of $N = 50{,}615$ individuals was drawn from the base population, including 3030 who overlapped with the case sample. Individual diagnosis sample counts are as follow: SSD (ICD10 F20–F29; $n = 16{,}008$), MDD (ICD10 F32–F33 and ICD 8 296.09, 296.29, 298.09, and 300.49; $n = 37{,}555$), ASD (ICD10 F84; $n = 24{,}975$), or ADHD (ICD10 F90; $n = 29{,}668$).

We also assessed three other brain disorders; intellectual disability (ID), epilepsy, and Tourette syndrome (TS), with prior evidence of association with NRXN1 deletions[16–18], using information on hospital diagnoses that had been obtained through the Danish Psychiatric Central Research Register[32] and the Danish National Patient Registry[33] for other iPSYCH2015 studies. The diagnostic codes used to identify individuals with these disorders were as follows: ID (ICD10: F70-F79; ICD8: 311-315), epilepsy (ICD10: G40; ICD8: 345 (excluding 345.29)), TS (ICD10: F95.2).

Supplementary Table 2 provides carrier count for each diagnosis, as well as a subset by subcohort (iPSYCH2012 or iPSYCH2015i) and gender.

Genotyping was performed using Illumina microarrays and has been described elsewhere[30]. Notably, the genotyping was performed on dried blood spot samples taken at birth. iPSYCH2012 and the additional extension (iPSYCH2015i) were genotyped using two different arrays, PsychArray version 1.0 and Global Screening Array version 2 (GSA), respectively. B allele frequency (BAF) and logR ratio (LRR) values were extracted using GenomeStudio and samples with a genotyping call rate below 95% were excluded.

CNV calling and pre-processing

CNVs were called using PennCNV[34] as described in our previously published CNV calling and processing protocol[35]. All steps of the calling pipeline were run using the Singularity container provided in the protocol. In brief, the intensity files were filtered to include only biallelic autosomal SNPs mapping uniquely to the Haplotype Reference Consortium (HRC) hg19 reference map[36], with a minor allele frequency of at least 0.1%, which yielded 280,700 and 509,754 probes for the PsychArray and GSA, respectively. Next, PennCNV calls were obtained with the script *"detect_cnv.pl"* setting a minimum number of probes (*--minsnp*) at 5, and the minimum length (*--minlength*) at 1000 bp. We then merged adjacent calls, with the *PennCNV* script *"clean_cnv.pl"* using the settings *"--fraction 0.2 --bp"* whereby two calls are merged if the gap between them corresponds to less than 20% of the combined length (in base pairs) of the calls. After CNV calling, we excluded samples with high levels of noise from the analysis. Thus, samples were excluded if they had either a LRR standard deviation value ≥ 0.35, BAF drift ≥ 0.005 or |GCWF | ≥ 0.02.

The locus of interest was defined as the NRXN1 gene in Ensembl[8] GRCh37 (https://grch37.ensembl.org/Homo_sapiens/Info/Index) plus 0.5 Mbp upstream and downstream of the gene boundaries (chr2:49 645 643-51 759 674). Any CNV call overlapping the region by at least 0.1% of its length was selected for visual validation using the function "*select_stich_calls()*" from the R package QCtreeCNV[35]; this step also removed CNV smaller than 10 SNPs. Visual inspection was performed independently by two analysts as already described[35]. The boundaries of true CNVs were manually adjusted if necessary and any discordant call between the analysts was re-evaluated in a final joint session.

CNV analysis

The genomic coordinates of *NRXN1* exons and transcripts were extracted using Ensembl[8] GRCh37 (https://grch37.ensembl.org/Homo_sapiens/Info/Index). We decided to focus on protein-coding transcripts only and thus selected all 9 transcripts with a protein match in UniProt[37] (https://www.uniprot.org/), yielding a total of 41 unique exons.

Under the assumption that exons mapping close to each other are likely to be deleted by the same CNVs, we investigated if any larger pattern was present at the level of the whole gene. We computed a genomically ordered correlation matrix across all exons, defined as an $N \times N$ matrix where N is the number of exons and the cell xy is the number of times a CNV affecting exon x also affects exon y.

CNVs are not equally distributed across the locus. We explored this topic using an IOU matrix, defined as an NxN matrix where N is the number of CNVs (381) and the cell xy is the IOU (Intersection Over the Union) score for CNVs x and y. IOU is 1 for two identical segments and ranges between 0 and 1 for any two overlapping segments, while non-overlapping segment pairs have an IOU range from 0 to approaching an asymptote at −1 the farther apart the two segments are. We then subgrouped exons in "alpha" and "beta" regions, based on Fig. 1d and previous literature[38], corresponding to exons ENSE00001682911 to ENSE00002460080 (beta), and exons ENSE00002453754 to ENSE00001547151 (alpha). For the purpose of the secondary analysis (Table 1), deletions affecting exons from both groups were assigned to "alpha".

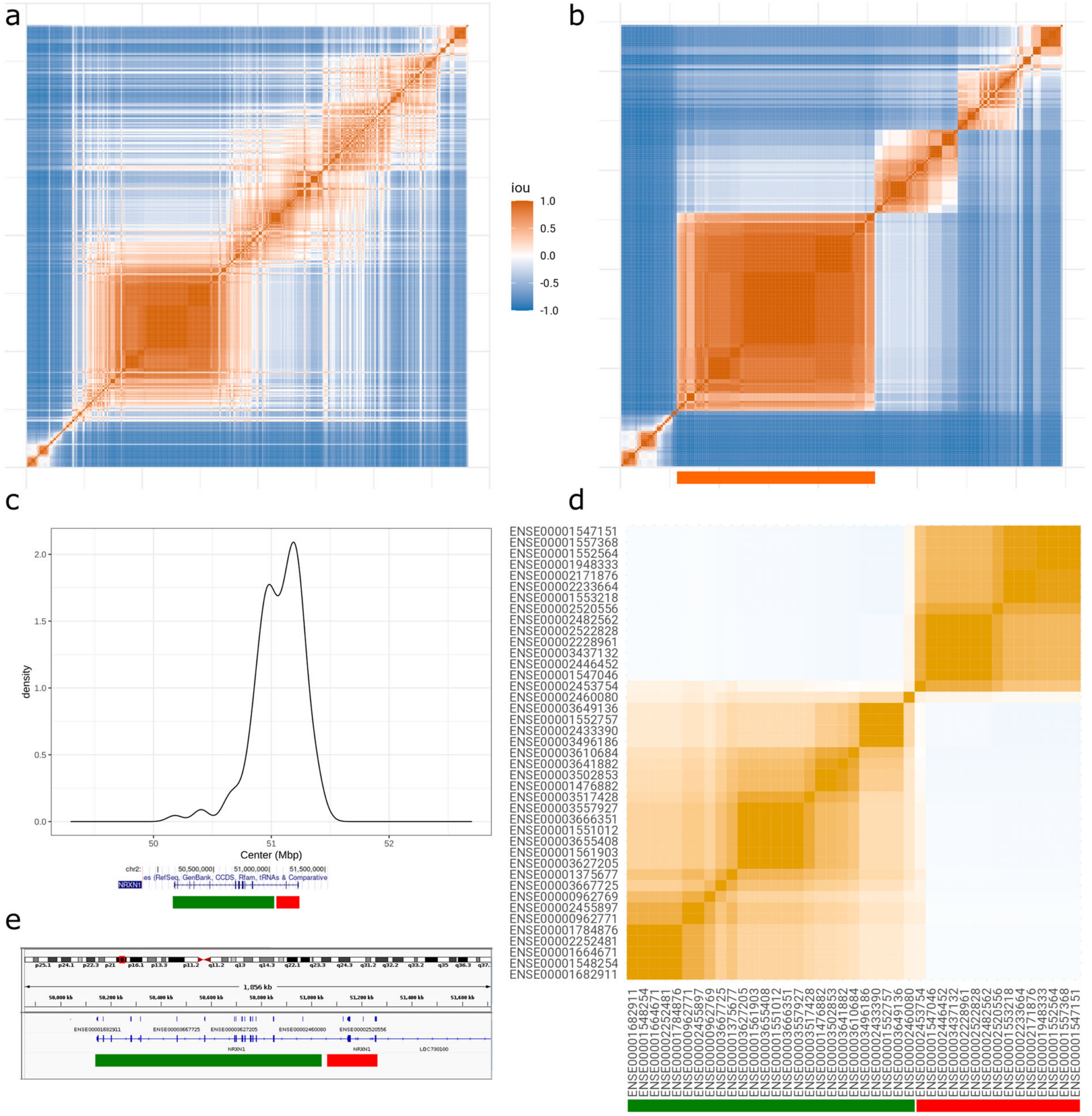


Fig. 1 | *NRXN1* deletions similarity matrices, and NRXN1 correlation matrix. Note that the *NRXN1* gene is encoded on the reverse strand, meaning the alpha promoter region (5′ of the gene) is shown on the right in this figure (see panel c for a breakdown of the gene structure). a Similarity heatmap for all deletions in the neurexin locus. Similarity is measured as IOU (intersection over the union), as described in the methods. Each row represents a deletion. Deletions are ordered on the x-axis based on the genomic position of the respective centre. Note that the scale is not linear as CNVs are not distributed equally across the locus. **b** Positional similarity for intronic deletions. This makes more evident the large group of very homogeneous deletions (marked with the orange bar on the x-axis). This group is referenced as segregating in the main text. **c** Distribution of the centre position for all exonic deletions in the NRXN1 gene locus. A schematic of the main gene isoform is aligned below the x-axis. The green and red bars mark the two exonic groups described in (**d**). **d** Exon correlation matrix. Exons are ordered based on genomic location. Note that the scale is not linear as exons are not distributed equally across the locus (see **c**). The red bar marks the exons in the "alpha" group and the green in the "beta" group. **e** A different view on the NRXN1 gene, the top blue graph shows all exons used in the study, while the bottom shows the top isoform.

Segregating deletion analysis

The coordinates of the segregating *NRXN1* deletion found in Rujescu et al.[25] were lifted over from hg18 to hg19 using the online tool LiftOver (https://genome.ucsc.edu/cgi-bin/hgLiftOver).

To identify SNPs in high linkage disequilibrium (LD) with the segregating deletion, we performed an association analysis (using the "*--assoc*" command in PLINK[39,40] with default settings, Supplementary Fig. 2) where we compared the 100 identified carriers with 5000 randomly drawn non-carriers, across all SNPs with MAF > 0.01 and info >0.95 mapping on the entire chromosome 2, using an imputed genotype dataset of the iPSYCH2015[41]. We then pruned the resulting SNPs with the following settings *--clump-p1 0.00001 --clump-r2 0.8 --clump-kb 1000000*.

The phased genotypes of the top 10 SNPs (shown in Supplementary Table 1) were imported in R. Here we constructed all possible haplotypes of length between two and five SNPs and tested their association with the deletion carriers using the R function *fisher.test()*. The haplotypes with an OR ≥ 2 and a p-value ≤ 0.0001 were further tested using the function *roc()* from the R package pROC[42] to get the AUC (Area Under the Curve) value.

Table 1 | *NRXN1* deletions and associated risk of psychiatric disorders

Exposure	Outcome[a]	N_{aff}[b]	OR[c]	CI95%[c]	P[d]	P_{FDR}[d]
Main analysis (all exonic vs all non-exonic deletions)						
Exonic	Any	108	2.13	1.39–3.26	0.00048	0.0067
Exonic	ADHD	41	2.01	1.23–3.31	0.0057	0.040
Exonic	ASD	52	3.05	1.87–4.97	7.4×10^{-6}	0.00031
Exonic	MDD	31	1.46	0.83–2.56	0.19	0.53
Exonic	SSD	12	1.41	0.69–2.90	0.35	0.80
Exonic	SCZ	8	1.88	0.81–4.33	0.14	0.41
Non-exonic	Any	147	1.05	0.79–1.39	0.74	0.86
Non-exonic	ADHD	51	1.04	0.72–1.51	0.82	0.86
Non-exonic	ASD	41	1.06	0.71–1.58	0.79	0.86
Non-exonic	MDD	49	0.91	0.61–1.35	0.64	0.86
Non-exonic	SSD	37	1.48	0.96–2.29	0.078	0.27
Non-exonic	SCZ	23	1.90	1.13–3.1907	0.0200	0.090
Secondary analysis (alpha vs beta exonic, and segregating vs non-segregating non-exonic deletions)						
Exonic alpha	Any	68	2.83	1.56–5.13	0.0006	0.0067
Exonic alpha	ADHD	28	3.02	1.54–5.94	0.0013	0.011
Exonic alpha	ASD	29	3.75	1.88–7.47	0.00020	0.0036
Exonic alpha	MDD	20	2.30	1.02–5.18	0.045	0.19
Exonic alpha	SSD	>7	2.29	0.88–5.97	0.091	0.30
Exonic beta	Any	40	1.49	0.81–2.75	0.20	0.53
Exonic beta	ADHD	13	1.15	0.53–2.47	0.72	0.86
Exonic beta	ASD	23	2.45	1.22–4.90	0.011	0.068
Exonic beta	MDD	11	0.89	0.39–2.03	0.78	0.86
Exonic beta	SSD	<5	0.78	0.24–2.46	0.67	0.86
Non-exonic segregating	Any	68	1.19	0.77–1.82	0.43	0.80
Non-exonic segregating	ADHD	23	1.22	0.70–2.13	0.49	0.80
Non-exonic segregating	ASD	18	1.23	0.66–2.28	0.51	0.80
Non-exonic segregating	MDD	24	1.06	0.57–1.96	0.86	0.86
Non-exonic segregating	SSD	20	2.19	1.15–4.16	0.017	0.080
Non-exonic other	Any	79	0.95	0.66–1.38	0.79	0.86
Non-exonic other	ADHD	28	0.93	0.57–1.51	0.77	0.86
Non-exonic other	ASD	23	0.95	0.56–1.61	0.84	0.86
Non-exonic other	MDD	25	0.81	0.47–1.38	0.44	0.80
Non-exonic other	SSD	17	1.06	0.58–1.95	0.85	0.86
Tertiary analysis (segregating non-exonic deletion vs other carriers of the underlying haplotype)						
Segregating deletion carriers	Any	68	1.19	0.77–1.82	0.43	0.80
Segregating deletion carriers	ADHD	23	1.22	0.70–2.13	0.49	0.80
Segregating deletion carriers	ASD	18	1.23	0.66–2.28	0.52	0.80
Segregating deletion carriers	MDD	24	1.06	0.57–1.96	0.86	0.86
Segregating deletion carriers	SSD	20	2.19	1.15–4.16	0.017	0.080
Other top haplotype carriers	Any	1512	0.96	0.88–1.05	0.37	0.80
Other top haplotype carriers	ADHD	530	0.99	0.88–1.10	0.80	0.86
Other top haplotype carriers	ASD	442	0.97	0.86–1.10	0.69	0.86
Other top haplotype carriers	MDD	592	0.94	0.84–1.06	0.35	0.80
Other top haplotype carriers	SSD	250	0.95	0.81–1.10	0.49	0.80
Quaternary analysis (same as above for SSD, but only in unrelated subjects of European ancestry)						
Segregating deletion carriers	SSD	14	1.93	0.94–3.97	0.074	0.27
Other top haplotype carriers	SSD	190	0.89	0.75–1.06	0.20	0.53

[a] The risk associated with different classes of deletions (and for carriers of a haplotype underlying a segregating founder deletion) was assessed separately for; attention-deficit/hyperactivity disorder (ADHD), autism spectrum disorder (ASD), major depressive disorder (MDD), and schizophrenia spectrum disorder (SSD), as well as combined (i.e., being affected with any of those disorders; Any). In the main analysis (top) we also assessed the risk associated with schizophrenia, narrowly defined (ICD10:F20), and in the quaternary analysis (bottom) we assessed the risk associated with SSD for the founder deletion and the underlying haplotype in a subset of unrelated European-ancestry samples only.

[b] For each test we provide the number of affected carriers (Naff); for the alpha and beta subclasses of exonic deletions, the exact number of carriers with SSD cannot be disclosed due to legislation regarding the protection of personal-level data in the research of nationwide registers and biobanks in Denmark.

[c] The odds ratios (OR) and 95% confidence intervals (CI95%) were in all instances derived from a logistic regression model including sex (as assigned at birth), age (at end of follow-up) and genotyping array (PsychArray or GSA) as covariates.

[d] The associated *p*-values (P) were subsequently corrected for multiple testing using false discovery rate adjustment (P_{FDR}).

Statistical analysis

We derived population-based prevalence (with CI95%) for the different subgroups of *NRXN1* deletions using the svydesign() and svyciprop() functions from the R package survey[43], with finite population correction (FPC) to account for oversampling of cases in iPSYCH2015.

Briefly, we divided the post-QC number of cases (77,655) and individuals from the random population subcohort (43,311) with the total number of corresponding individuals in the source population (90,218 and 1,657,449) to derive the sampled population fractions; 0.85068 (100% of cases minus the ones failing genotype or excluded in QC) and 0.02613, respectively. Samples from overlapping individuals (cases-in-subcohort) were assigned the case population fraction (0.85068).

We calculated the corresponding prevalence of exonic *NRXN1* deletions in the UKB directly from carrier counts provided by Crawford et al.[44] and derived CI95% as follows (R pseudocode): *CI95% = qbeta(c(0.05/2,1-0.05/2), nCarrier + 0.5, nTotal-nCarrier + 0.5)*, where nCarrier and nTotal indicate the number of carriers and the total number of assessed samples (421,268), respectively.

We compared the prevalence of exonic deletions in iPSYCH2015 and UKB with Welch's test of the difference between two means assuming unequal variance. Briefly, we defined the difference; $d = (|log(p_{iPSYCH}/p_{UKB})|)$, the standard error of the difference; $SEd = \sqrt{(SE_{iPSYCH}{}^2 + SE_{UKB}{}^2)}$, and the p-value; $P = 2*(1\text{-}pnorm(d/SEd))$, where p_{iPSYCH} and SE_{iPSYCH}, and p_{UKB} and SE_{UKB}, indicate the prevalence and standard error of prevalence in iPSYCH2015 and UKB, respectively.

To estimate the risk of index psychiatric disorders associated with *NRXN1* deletions we ran a logistic regression analysis using *gam()* from the R package mgcv[45]. We used age, sex (at birth) and SNP array type as covariates, with a smoothed function to model the effect of age using the mgcv function s(). In each association, we included all cases for the phenotype of interest and all controls, defined as individuals not having any of the index diagnoses. For the later-onset disorders SSD, MDD and SCZ, we only included those controls who were at least as old as the youngest case. Multiple testing correction was applied to the table containing the results of all three analyses (Table 1) using the R function *p.adjust(method = "fdr")*. We then compared risk estimates with those reported in published case-control studies (in each case the study applying the largest case-control sample size for the respective disorder; only considering studies that controlled for genotyping array, when including samples genotyped on different arrays) using a Welch's test in a similar way as described above for prevalence estimate comparison. We performed two additional sensitivity analyses, we ran the first model on the phenotype schizophrenia (ICD10, F20) instead of SSD, and we ran the last model on the European unrelated subset of iPSYCH2015[41].

To estimate the risk of the three other brain disorders associated with *NRXN1* deletions we fitted a logistic regression model using case status for each of the four iPSYCH disorders (ADHD, ASD, MDD and SSD) as covariates in addition to age, sex (at birth) and SNP array type.

Software

All analyses were performed on HPC running CentOS Linux 7. PLINK[39,40] version 190b6.21, R[46] version 4.0.5 and VCFtools[47] 0.1.17 were installed via the conda package manager (https://anaconda.org/). PennCNV[34] version 1.0.5, bcftools[48] version 1.14, htslib[49] 1.14 are a part of the container we used for the CNV calling described in the previous section, available on Docker Hub (https://hub.docker.com/r/sinomem/docker_cnv_protocol). For the analysis and the figures, we used the following R packages: data.table[50], pROC[42], survey[43], mgcv[45] and ggplot2[51].

Ethics statement

This study is in full compliance with all relevant ethical regulations including the Declaration of Helsinki. Access to the data and its use for research purposes was granted by The Danish Scientific Ethics Committee, the Danish Health Data Authority, the Danish Data Protection Agency, and the DNSB Steering Committee. For this study, the Danish Scientific Ethics Committee has, in accordance with the Act on Research Ethics Review of Health Research Projects (in Danish: *Komitéloven*), waived the need for informed consent in biomedical research based on existing biobanks.

Results

Descriptive statistics and prevalences

After quality control, our sample consisted of 77,655 cases of the four disorders ascertained in iPSYCH2015 (22,167 ASD, 26,186 ADHD, 31,622 MDD, 13,126 SSD) and a population-representative random cohort of 43,311 samples, for a total of 118,427 unique samples. Given the structure of the sample, there is a small overlap between the two groups. Moreover, a given case can be diagnosed with more than one of the index disorders. We called CNVs in the larger *NRXN1* locus (*NRXN1* gene plus 0.5 Mbp upstream and downstream) and performed visual validation as described in the methods. In total 1387 calls were evaluated, of those 378 were deemed as true CNVs, 573 as false calls, and 436 as unknown (meaning no definitive judgement was possible, most often due to the small number of markers available). Given the small proportion of duplications (21 out of 378) and the low reliability of validating small duplications, we discarded duplications from all subsequent analyses and focused on deletions only. This resulted in a total of 357 carriers (255 cases, 102 controls) of which 135 (108 cases, 27 controls) were exonic, i.e., overlapping at least one exon.

The prevalence of *NRXN1* deletions in the general Danish population is 2.55 (95% CI: 2.13–3.04) per 1000 individuals and 0.70 (95% CI: 0.50–0.98) when restricting to exonic deletions. This is almost two times higher than what was previously reported in UKB[44], 0.70 vs 0.39 per 1000 individuals (*p*-value 0.0014, Welch's test). Subgrouping by subcohort (iPSYCH2012 and the extension iPSYCH2015i respectively) the prevalence estimates are 2.20 (95% CI: 1.72–2.81) and 3.07 (95% CI: 2.37–3.98) for any deletion, and 0.78 (95% CI: 0.52–1.17) and 0.58 (95% CI: 0.32–1.05) for exonic deletions only. Supplementary Table 3 provides a prevalence breakdown per gender.

NRXN1 deletions subgrouping

Neither exonic nor non-exonic deletions are distributed uniformly across the locus (Supplementary Fig. 1). In order to disentangle the risk signal in *NRXN1* CNVs further than exonic/non-exonic deletions, we created a set of subgroups. We used a similarity matrix of all CNV pairs (Fig. 1a, b) and a correlation matrix of the deleted exons (Fig. 1c) as described in the methods. Regarding non-exonic CNVs, we identified a clear subgroup of 100 very similar CNVs (IOU > 80%) corresponding to those between exons ENSE00003649136 and ENSE00002460080 (Fig. 1a, b, Supplementary Fig. 1d). The average boundaries of this group of deletions correspond to a deletion previously found segregating in several European populations (Chr2:50,882,153–50,945,699 in Rujescu et al. and Chr2:50,882,111–50,947,645 in this study)[25]. The prevalence of this segregating intronic deletion is 0.77 (95% CI: 0.55–1.06) per 1000 individuals.

Regarding exonic CNVs, the correlation plot (Fig. 1d) shows that exons are affected by deletions essentially in two blocks, exons ENSE00001682911 to ENSE00002460080 (roughly corresponding to the 3′ end of the gene to the group of exons where the promoter of the beta isoform is located, referred to as beta region from now on), and exons ENSE00002453754 to ENSE00001547151 (roughly corresponding to said group of exons to the 5′ end of the gene, referred to as alpha promoter region from now on). See also Supplementary Table 4 and Supplementary Fig. 3 for more details on exonic deletions. The number of carriers in each group was 81 and 54, for the alpha and beta promoter regions, respectively. While smaller clusters are observed within both large groups, further subgrouping of these two main clusters resulted in limited study power, thus we only used these two main clusters for further analysis.

NRXN1 deletions and associated risk of psychiatric disorders

To estimate the association between *NRXN1* deletions and the risk of the four index psychiatric disorders (ADHD, ASD, MDD, SSD) we conducted three separate analyses based on the deletion subgroups described above. As

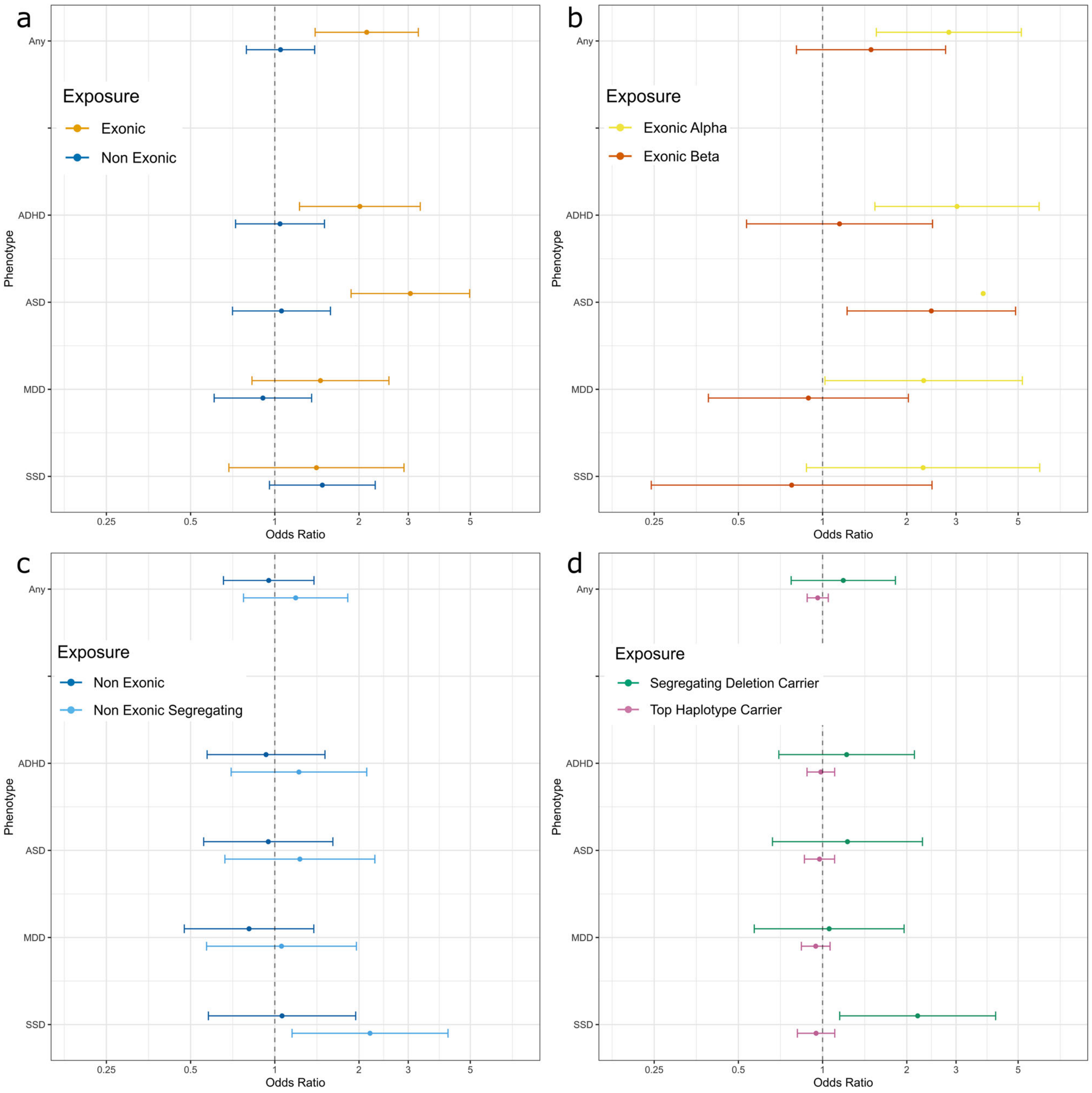


Fig. 2 | Forest plots showing the ORs resulting from three logistic regression analyses on four neurodevelopmental disorders. a First model, ORs for exonic and non-exonic deletions in the *NRXN1* locus. **b**, **c** Second model, exonic deletion is divided into three subgroups based on the exons they overlap (alpha promoter region, beta promoter region, at least one of both) and non-exonic are divided into two subgroups (those belonging to the segregating deletion and all the rest). Note that the scale of (**b**) differs from the rest. **d** Third model, ORs for being a carrier of the segregating deletion or of the haplotype associated with the deletion but without such deletion.

described in the methods, we used a logistic model adjusting for age, SNP array type and sex. The resulting OR estimates and carrier counts are summarised in Fig. 2 and Table 1. Overall, we see an increased risk of ADHD and ASD associated with carriage of exonic deletions, but not of SSD (also when running the analysis on the stricter schizophrenia phenotype, OR: 1.87, 95% CI: 0.81–4.33) or MDD.

We also attempted to replicate findings of previous studies linking exonic NRXN1 deletions to increased risk of ID[16], epilepsy[17] and TS[18], although these disorders had not been specifically targeted by the iPSYCH case-cohort design and as a consequence our estimates are not as well powered (or population-representative) as for the four index psychiatric disorders (Supplementary Table 5). As shown in Table 2, we replicate the previous reports for ID and epilepsy, but not for TS. In all instances, (both for the four index psychiatric disorders and the three other brain disorders) our risk estimates are lower than reported in the case-control studies that we draw comparisons with, although not significantly so except for SSD and TS (Table 2). When we used the stricter SCZ diagnosis (ICD:F20) the difference with the comparison study[12] was not significant ($P = 0.15$; Table 2).

When subgrouping CNVs, deletions in the alpha promoter region of the gene appear to carry the majority of the signal. This is in accordance with previous literature both based on case-control studies as well as in vitro studies[5,38].

While we observed no association between exonic deletions and risk of SSD (OR = 1.40; 95% CI: 0.68–2.89), this diagnosis group was the only one where we observed a significant increase in risk associated with intronic deletions. As shown in Fig. 2c, this association seems to be driven by the

Table 2 | Comparison of effect sizes for exon-disrupting NRXN1 deletions between iPSYCH2015 and published case-control studies

Psychiatric outcome	iPSYCH2015			Comparison study[a]			Welch's test[b]	
	OR (CI95%)	P	N[c]	OR (CI95%)	P	N[c]	d (se)	P_d
ADHD	2.01 (1.22–3.32)	0.0057	26,186 (0.15%) 40,626 (0.066%)	4.68 (1.82–10.6)	0.00093	8883 (0.1%) 180,809 (0.021%)	0.84 (0.52)	0.10
ASD	3.06 (1.88–4.95)	7.4×10^{-6}	22,167 (0.23%) 40,626 (0.066%)	7.24 (0.93–326)	0.036	2558 (0.27%) 2670 (0.037%)	0.81 (1.52)	0.57
MDD	1.46 (0.83–2.56)	0.19	31,622 (0.10%) 40,626 (0.066%)	2.01 (1.18–3.19)	0.0057	23,979 (0.079%) 383,095 (0.039%)	0.32 (0.38)	0.41
SSD	1.41 (0.69–2.90)	0.35	13,126 (0.091%) 40,626 (0.066%)	4.50 (2.03–10.9)	2.8×10^{-5}	20,403 (0.15%) 26,628 (0.034%)	1.16 (0.56)	0.040
ID	2.68 (1.65–4.34)	6.7×10^{-5}	5975 (0.38%) 40,626 (0.066%)	8.14 (2.91–22.7)	<0.0001	19,263 (0.21%) 15,264 (0.026%)	1.11 (0.58)	0.055
Epilepsy	1.94 (1.01–3.73)	0.046	3957 (0.25%) 40,626 (0.066%)	9.91 (1.92–51.1)	0.0049	1569 (0.32%) 6201 (0.032%)	1.63 (0.90)	0.070
TS	1.53 (0.65–3.56)	0.33	2222 (0.27%) 40,626 (0.066%)	20.3 (2.6–156)	5.9×10^{-5}	2434 (0.49%) 4093 (0.033%)	2.59 (1.13)	0.022

Comparison of effect sizes between iPSYCH2015 and published case-control studies

[a] Risk estimates for exonic deletions in iPSYCH2015 were compared with estimates from the largest available published case-control studies for attention-deficit/hyperactivity disorder (ADHD)[14], autism spectrum disorder (ASD)[13], major depressive disorder (MDD)[15], schizophrenia spectrum disorder (SSD)[12], intellectual disability (ID)[16], epilepsy[17], and Tourette syndrome (TS)[18].

[b] The comparison was done through a Welch's test, with d (se) denoting the absolute difference in estimates ($|\log(OR_1/OR_2)|$) and standard error thereof ($\sqrt{(SE_1^2 + SE_2^2)}$), and P_d indicating the significance of the difference (2*(1-pnorm(d/(se)))). The difference in risk estimates between iPSYCH2015 and Rees et al. (fourth row from top) was not significant when using iPSYCH2015 estimates for narrowly defined (ICD10;F20) schizophrenia (OR (CI95%) = 1.87 (0.81–4.33), d (se) = 0.88 (0.61), P_d = 0.15).

[c] Above; number of affected (% of affected with an exonic deletion in NRXN1 gene) – Below; number of unaffected (% of unaffected with an exonic deletion in NRXN1 gene).

segregating intronic deletion described above (*OR* = 2.20; 95% CI: 1.15–4.18). Since intronic deletions are usually not considered pathogenic, we hypothesised that the risk associated with the segregating deletion could be explained by another variation co-segregating with it. As described in the methods, we ran a simple association test between all SNPs in chromosome 2 and the recurrent deletion. Using the 10 most associated SNPs we constructed all two-to-five SNPs haplotypes, and we identified the most characteristic haplotype with an AUC of 0.94 (rs10205006-T, rs7608415-G, rs62140665-C, rs17041353-G). We then ran a final analysis grouping samples based on whether they were carriers of this haplotype or not. The results, shown in Fig. 2d, confirm that this deletion is only associated with an increased risk of SSD and, notably, that the associated risk is confined to the deletion (*n* = 100) and not observed among carriers of the underlying haplotype without the deletion (*n* = 2341). However, we do not observe a significantly increased risk of SSD associated with this deletion when we restrict the sample to the European unrelated subset (OR: 1.8, 95% CI: 0.8–3.8). Finally, given the high number of analyses we performed multiple testing corrections (FDR, adjusted *p*-values are provided in Table 1). As expected, the strongest association reported in this study, namely ASD and ADHD with exonic deletions in the NRXN1 locus, remains significant after the correction. However, the SSD association with the segregating intronic deletion did not remain significant after correction.

Discussion

Deletions affecting the *NRXN1* gene have been investigated for associations with psychiatric and developmental disorders for almost twenty years. CNVs in the *NRXN1* locus can be very heterogeneous, affecting one or more exons, besides occurring between two exons. Exonic deletions in particular have been associated with SDD[12,25], ADHD[14], MDD[15] and ASD[13]. However, most of the published studies have been limited to smaller case-control samples or meta-analyses of case-control samples. Moreover, intronic deletions are usually discarded from the analysis[11,25,44]. In this study, we attempt to disentangle the risk profile of exonic as well as intronic deletions defining subgroups of similar deletions. Using the population-representative case-cohort design of iPSYCH2015, we report unbiased estimates of the population prevalence and association of such subtypes of deletions with four core psychiatric disorders.

As in previous studies on the same cohort[26–28], we find the prevalence in the general population to be higher and the risk associations to be lower than previously reported. We observe exonic deletions to be associated with ASD and ADHD. When subgrouping deletions based on location in the gene, the association is driven by deletions in the alpha promoter region of the gene, while deletions in the other half of the gene are rarer and possibly associated with less increased risk of psychiatric disorders. Notably, CNVs in the alpha promoter region are known to be more frequent and indeed are in our sample as well. The association appears robust, suggesting a biological reason for the excess risk in one proportion of the gene. However, it may also be exacerbated by the difference in number of carriers. We also confirm the presence of a small segregating deletion that does not affect any exon and find it to be potentially linked to SSD. While this signal did not survive multiple testing corrections, we believe it can be taken as an indication that intronic CNVs should not be discarded a priori in this kind of analysis.

Notably, we do not find exonic deletions in the *NRXN1* locus to be associated with an increased risk of SSD, which at first glance seems in strong contrast with previous reports[11,12,25,52–55]. However, when we examine the methodology and timeline of these previous reports, a more conciliatory picture emerges. The first large-scale study of schizophrenia-associated risk with exon-disrupting NRXN1 deletions was that of Rujescu et al.[25], who reported an OR of 9 in a meta-analysis of European samples including ~3000 cases and >30,000 controls. Most subsequent studies derived their risk estimates either fully[11,52,55] or in part[53] by merging all schizophrenia cases and controls from previously published studies and performing a simple Fisher's exact test on the pooled sample. As a consequence, in all these studies a large fraction of the control individuals (40%–80%) are those from the original report by Rujescu et al.[25], whereas most case individuals are from other studies, most often applying denser arrays than the HumanHap300 array used in Rujescu et al.[25]. As NRXN1 deletions vary widely in size and breakpoints, the approach taken in these studies is very vulnerable to batch effects owing to differing resolution to detect exon-disrupting deletions across different genotyping platforms.

Since the initial report of Rujescu et al. only two other large-scale studies (Rees et al.[12], and Marshall et al.[54]) have been published that do not include the large control sample of Rujescu et al. Both these studies report slightly lower carrier rates in cases (0.15% and 0.11%) and higher carrier rates in controls (0.034% and 0.020%) than Rujescu et al. (0.24% in cases and

0.015% in controls), and when meta-analysing across genotyping platforms, both studies correspondingly report lower odds ratios (4.5 and 5.8, respectively). These estimates are still higher than we find in iPSYCH2015, as is also the case for the other three core iPSYCH2015 disorders. This could in part be due to case ascertainment; iPSYCH2015 relies on hospital-based diagnoses from national registers, without any further confirmation of case status. However, the carrier frequency among iPSYCH2015 cases is very similar to those reported by the largest previously published studies for each disorder. In contrast, the population-based prevalence of exon-disrupting NRXN1 deletions in iPSYCH2015 is twice as high as reported in UKB[15,44] and the control samples used in Rees et al.[12] and Girirajan et al.[13], and more than three times higher than among the controls of Gudmundsson et al.[14] This is in line with results of our previous CNV studies involving iPSYCH2015 and suggests that the overall tendency for lower CNV-associated risk estimates in iPSYCH2015 is in large part explained by the higher CNV prevalence in the general population compared to individuals used as controls in other studies.

The sample size is the major limitation of this study. Although *NRXN1* is a hotspot for non-recurrent CNVs, such events are rare. For this reason, we lacked the power to include duplications in the study or subgroup deletions beyond the two major groups. Also, both the relatively young age of participants and the specific focus on a limited number of psychiatric disorders in the iPSYCH case-cohort design limits our study power for the later-onset iPSYCH disorders (MDD and SSD) as well as other brain disorders not targeted by the study design (such as ID, epilepsy and TS). Some of the individuals from the random subcohort will later go on to develop MDD or SSD, which in the case of MDD, with its high lifetime prevalence of 10–15%, could have had an attenuating effect on the estimated OR, while it is unlikely to have had affected the risk estimate for SSD, with its much lower lifetime prevalence (1.0–1.5%). As for the brain disorders not targeted by the case-cohort design, the case sample sizes are relatively small and enriched with individuals with comorbid ADHD, ASD, MDD and/or SSD. To account for this enrichment, while also retaining the maximum case sample size, we fitted a logistic model that included each of the four iPSYCH disorders as covariates. While maximising study power, this approach probably leads to an overestimate of case carrier frequency but at the same time an underestimate of the associated OR for these disorders.

Notwithstanding these limitations, our results add important insight into the association between *NRXN1* deletions and the risk of psychiatric illness. Most importantly, we show that the risk is mainly driven by deletions disrupting exons specific to the alpha isoform of Neurexin 1. Also, we show that as with recurrent CNVs, previous case-control studies of *NRXN1* deletions have likely underestimated their population prevalence and consequently overestimated their associated risk. Finally, we characterise the haplotype background of a previously reported intronic deletion segregating at ~0.1% carrier frequency in the Danish population, and while inconclusive, our results warrant further study into its possible association with psychiatric and/or other cognitive/behavioural traits.

Data availability

Regarding access to study data (other than sensitive person-level data, which by requirement of the data custodian and Danish legislation cannot be shared) please contact the corresponding author.

Received: 4 June 2024; Accepted: 26 November 2024;
Published online: 19 December 2024

References

1. Südhof, T. C. Synaptic neurexin complexes: a molecular code for the logic of neural circuits. *Cell* **171**, 745–769 (2017).
2. Reissner, C., Runkel, F. & Missler, M. Neurexins. *Genome Biol.* **14**, 213 (2013).
3. GTEx Portal. https://gtexportal.org/home/.
4. Lonsdale, J. et al. The Genotype-Tissue Expression (GTEx) project. *Nat. Genet.* **45**, 580–585 (2013).
5. Flaherty, E. et al. Neuronal impact of patient-specific aberrant NRXN1α splicing. *Nat. Genet.* **51**, 1679–1690 (2019).
6. Jenkins, A. K. et al. Neurexin 1 (NRXN1) splice isoform expression during human neocortical development and aging. *Mol. Psychiatry* **21**, 701–706 (2016).
7. Fuccillo, M. V. & Pak, C. Copy number variants in neurexin genes: phenotypes and mechanisms. *Curr. Opin. Genet. Dev.* **68**, 64–70 (2021).
8. Cunningham, F. et al. Ensembl 2022. *Nucleic Acids Res.* **50**, D988–D995 (2022).
9. Castronovo, P. et al. Phenotypic spectrum of *NRXN1* mono- and bi-allelic deficiency: a systematic review. *Clin. Genet.* **97**, 125–137 (2020).
10. Béna, F. et al. Molecular and clinical characterization of 25 individuals with exonic deletions of *NRXN1* and comprehensive review of the literature. *Am. J. Med. Genet. B Neuropsychiatr. Genet.* **162**, 388–403 (2013).
11. Kirov, G. et al. Neurexin 1 (NRXN1) deletions in schizophrenia. *Schizophr. Bull.* **35**, 851–854 (2009).
12. Rees, E. et al. Analysis of intellectual disability copy number variants for association with schizophrenia. *JAMA Psychiatry* **73**, 963–969 (2016).
13. Girirajan, S. et al. Refinement and discovery of new hotspots of copy-number variation associated with autism spectrum disorder. *Am. J. Hum. Genet.* **92**, 221–237 (2013).
14. Gudmundsson, O. O. et al. Attention-deficit hyperactivity disorder shares copy number variant risk with schizophrenia and autism spectrum disorder. *Transl. Psychiatry* **9**, 258 (2019).
15. Kendall, K. M. et al. Association of rare copy number variants with risk of depression. *JAMA Psychiatry* **76**, 818–825 (2019).
16. Lowther, C. et al. Molecular characterization of NRXN1 deletions from 19,263 clinical microarray cases identifies exons important for neurodevelopmental disease expression. *Genet. Med. J. Am. Coll. Med. Genet.* **19**, 53–61 (2017).
17. Møller, R. S. et al. Exon-disrupting deletions of NRXN1 in idiopathic generalized epilepsy. *Epilepsia* **54**, 256–264 (2013).
18. Huang, A. Y. et al. Rare copy number variants in NRXN1 and CNTN6 increase risk for tourette syndrome. *Neuron* **94**, 1101–1111 (2017).
19. Enggaard Hoeffding, L. K. et al. Sequence analysis of 17 NRXN1 deletions. *Am. J. Med. Genet. B Neuropsychiatr. Genet.* **165B**, 52–61 (2014).
20. Wilson, T. E. et al. Large transcription units unify copy number variants and common fragile sites arising under replication stress. *Genome Res.* **25**, 189–200 (2015).
21. Kirov, G. et al. Comparative genome hybridization suggests a role for NRXN1 and APBA2 in schizophrenia. *Hum. Mol. Genet.* **17**, 458–465 (2008).
22. Zahir, F. R. et al. A patient with vertebral, cognitive and behavioural abnormalities and a de novo deletion of NRXN1alpha. *J. Med. Genet.* **45**, 239–243 (2008).
23. Kim, H.-G. et al. Disruption of neurexin 1 associated with autism spectrum disorder. *Am. J. Hum. Genet.* **82**, 199–207 (2008).
24. Marshall, C. R. et al. Structural variation of chromosomes in autism spectrum disorder. *Am. J. Hum. Genet.* **82**, 477–488 (2008).
25. Rujescu, D. et al. Disruption of the neurexin 1 gene is associated with schizophrenia. *Hum. Mol. Genet.* **18**, 988–996 (2009).
26. Olsen, L. et al. Prevalence of rearrangements in the 22q11.2 region and population-based risk of neuropsychiatric and developmental disorders in a Danish population: a case-cohort study. *Lancet Psychiatry* **5**, 573–580 (2018).
27. Calle Sánchez, X. et al. Comparing copy number variations in a Danish Case Cohort of Individuals With Psychiatric Disorders. *JAMA Psychiatry* **79**, 59–69 (2022).
28. Vaez, M. et al. Population-Based Risk of Psychiatric Disorders Associated With Recurrent Copy Number Variants. *JAMA Psychiatry* **81**, 957–966 (2024).

29. Bybjerg-Grauholm, J. et al. The iPSYCH2015 Case-Cohort sample: updated directions for unravelling genetic and environmental architectures of severe mental disorders. Preprint at *medRxiv* https://doi.org/10.1101/2020.11.30.20237768 (2020).
30. Pedersen, C. B. et al. The iPSYCH2012 case–cohort sample: new directions for unravelling genetic and environmental architectures of severe mental disorders. *Mol. Psychiatry* **23**, 6–14 (2018).
31. Pedersen, C. B. The Danish Civil Registration System. *Scand. J. Public Health* **39**, 22–25 (2011).
32. Mors, O., Perto, G. P. & Mortensen, P. B. The Danish Psychiatric Central Research Register. *Scand. J. Public Health* **39**, 54–57 (2011).
33. Schmidt, M. et al. The Danish National Patient Registry: a review of content, data quality, and research potential. *Clin. Epidemiol.* **7**, 449–490 (2015).
34. Wang, K. et al. PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res.* **17**, 1665–1674 (2007).
35. Montalbano, S. et al. Accurate and effective detection of recurrent copy number variants in large SNP genotype datasets. *Curr. Protoc.* **2**, e621 (2022).
36. McCarthy, S. et al. A reference panel of 64,976 haplotypes for genotype imputation. *Nat. Genet.* **48**, 1279–1283 (2016).
37. UniProt Consortium, The UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **51**, D523–D531 (2023).
38. Cosemans, N. et al. The clinical relevance of intragenic NRXN1 deletions. *J. Med. Genet.* **57**, 347–355 (2020).
39. Chang, C. C. et al. Second-generation PLINK: rising to the challenge of larger and richer datasets. *GigaScience* **4**, 7 (2015).
40. Purcell, S. et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* **81**, 559–575 (2007).
41. Appadurai, V. et al. Accuracy of haplotype estimation and whole genome imputation affects complex trait analyses in complex biobanks. *Commun. Biol.* **6**, 1–12 (2023).
42. Robin, X. et al. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinform.* **12**, 77 (2011).
43. Lumley, T. Analysis of complex survey samples. *J. Stat. Softw.* **9**, 1–19 (2004).
44. Crawford, K. et al. Medical consequences of pathogenic CNVs in adults: analysis of the UK Biobank. *J. Med. Genet.* **56**, 131–138 (2019).
45. Wood, S. N., Pya, N. & Saefken, B. Smoothing parameter and model selection for general smooth models (with discussion). *J. Am. Stat. Assoc.* **111**, 1548–1563 (2016).
46. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2021). https://www.R-project.org/.
47. Danecek, P. et al. The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).
48. Danecek, P. et al. Twelve years of SAMtools and BCFtools. *GigaScience* **10**, giab008 (2021).
49. Bonfield, J. K. et al. HTSlib: C library for reading/writing high-throughput sequencing data. *GigaScience* **10**, giab007 (2021).
50. Barrett, T. et al. data.table: Extension of 'data.frame'. https://CRAN.R-project.org/package=data.table (2024).
51. Wickham, H. *Ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag, New York, 2016).
52. Dabell, M. P. et al. Investigation of NRXN1 deletions: clinical and molecular characterization. *Am. J. Med. Genet. A* **161A**, 717–731 (2013).
53. Rees, E. et al. Analysis of copy number variations at 15 schizophrenia-associated loci. *Br. J. Psychiatry* **204**, 108–114 (2014).
54. Marshall, C. R. et al. Contribution of copy number variants to schizophrenia from a genome-wide study of 41,321 subjects. *Nat. Genet.* **49**, 27–35 (2017).
55. Hu, Z. et al. Genetic insights and neurobiological implications from NRXN1 in neuropsychiatric disorders. *Mol. Psychiatry* **24**, 1400–1414 (2019).

Acknowledgements

This work was supported by grants from the Lundbeck Foundation (R165–2013–15320, R155–2014–1724, and R248–2017–2003) and the NIH (R01 MH124789-01).

Author Contributions

Concept and design: Montalbano, Krebs, Rosengren, Vaez, Helenius, Ingason. Acquisition, analysis or interpretation of data: Montalbano, Ingason. Drafting of the manuscript: Montalbano, Ingason. Crtitical review of the manuscript for important intellectual content: Montalbano, Krebs, Rosengren, Vaez, Hellberg, Mortensen, Børglum, Geschwind, Raznahan, Thompson, Helenius, Werge, Ingason. Statistical analysis: Montalbano, Vaez, Thompson Helenius, Ingason. Obtained funding: Mortensen, Børglum, Raznahan, Werge, Ingason. Administrative, technical or material support: Montalbano, Rosengren, Werge, Ingason. Supervision: Werge, Ingason.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41525-024-00450-8.

Correspondence and requests for materials should be addressed to Andrés. Ingason.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

iPSYCH Investigators

Anders D. Børglum[2,4,5], David M. Hougaard[13], Merete Nordentoft[14], Ole Mors[15], Preben B. Mortensen[2,3], Thomas Werge[1,2,12], Jakob Grove[2,4,5,16], Thomas D. Als[2,4,5], Alfonso Buil[1,2], Anders Rosengren[1,2], Andrés Ingason[1,2] ✉, Andrew J. Schork[1,2], Dorte Helenius[1,2], Jesper Gådin[1], Richard Zetterberg[1], Vivek Appadurai[1], Joeri Meijsen[1], Kajsa-Lotta Georgii Hellberg[1,2], Bjarni J. Vilhjálmsson[3,16], Carsten B. Pedersen[3], Esben Agerbo[3], Jakob Christensen[3], Liselotte V. Petersen[3], Marianne Giørtz Pedersen[3], Jonas Bybjerg-Grauholm[13] & Marie Bækvad-Hansen[13]

[13]Department for Congenital Disorders, Statens Serum Institute, Copenhagen, Denmark. [14]Mental Health Centre Copenhagen, Capital Region of Denmark, Copenhagen University Hospital, Copenhagen, Denmark. [15]Psychosis Research Unit, Aarhus University Hospital-Psychiatry, Aarhus, Denmark. [16]BiRC, Bioinformatics Research Centre, Aarhus University, Aarhus, Denmark.

CNValidatron, automated validation of CNV calls using computer vision

Simone Montalbano[1], G. Bragi Walters[2], Gudbjorn F. Jonsson[2], Jesper R. Gådin[1], Thomas Werge[1,3], Hreinn Stefansson[2], Kári Stefánsson[2] & Andrés Ingason[1]

[1]Institute of Biological Psychiatry, Mental Health Services, Copenhagen University Hospital, Roskilde, Denmark.

[1]deCODE genetics/Amgen, Inc., Reykjavik, Iceland.

[3]Department of Clinical Medicine, University of Copenhagen, Copenhagen, Denmark.

Abstract

Motivation

For more than a decade, running PennCNV on SNP array data has been the gold standard for detecting Copy Number Variants (CNVs, deletions and duplications). It is generally assumed that PennCNV has high sensitivity but poor specificity, leading to a large portion of CNV calls being false positives. Researchers often rely on manual inspection of the raw data to validate CNV calls. However, this approach is not feasible for more than a handful of loci in large collections.

Results

Here we present an R package implementing a convolutional neural network capable of automating CNV validation with an accuracy comparable to a trained human analyst. We also present an in-depth analysis into PennCNV false positive and false negative rates. Finally, we propose an algorithm to simplify the analysis of genome-wide CNV calls computing CNV regions.

Availability and implementation

The code is available on GitHub https://github.com/SinomeM/CNValidatron_fl.

Introduction

Copy Number Variants (CNVs) are a class of structural genetic variation, representing a segment of chromosomal DNA sequence that has been deleted, duplicated, or otherwise misplaced compared to the reference genomic sequence.[1] For the purposes of this manuscript, we define CNVs as all deletions and duplications ranging between small structural variants and large chromosomal rearrangements, roughly corresponding to ~50kb to ~10Mb.

For more than a decade, running PennCNV[2] on SNP array data has been the gold standard for detecting CNVs, especially in large collections of samples. While collections of whole-genome sequenced (WGS) samples are growing rapidly[3,4] and WGS (especially from long read technologies) is the ideal data source for detecting small and complex structural variants (SV), WGS is not necessarily better than SNPs arrays in detecting larger CNVs with good accuracy.

All CNV calling programs rely on some measure of relative DNA intensity (log-R-ratio (LRR) for SNP arrays and read depth for WGS data), and some also include a measure of allelic composition (such as B-allele frequency (BAF) for Illumina arrays)[5,6]. PennCNV uses a Hidden Markov Model to predict the state (copy number) of each marker, based on the state of the previous marker, integrating both LRR and BAF. Sets of consecutive markers with the same state are then stitched together to create the actual calls. CNV calling from SNP array data using PennCNV is believed to have a generally high sensitivity, however it tends to have poor specificity, with more than 50% of calls typically being false positives[7]. Depending on sample quality, array type and genomic location, the false positive rate can be much higher[7]. For this reason, researchers have used different approaches to control the false positive rate, including filtering whole samples as well as individual CNVs based on quality metrics, or using the intersection of calls across different CNV calling algorithms.[8] Moreover, especially when precision is extremely important (e.g. in very rare exposures such as recurrent CNVs)[9,10], visual validation, performed by a human analyst through inspection of LRR and BAF plots around the suspected CNV call, remains the best approach and is possibly second only to a wet-lab validation using a quantitative polymerase chain reaction (qPCR) experiment. Clearly, such an approach is overly time consuming, and its application is therefore limited to studies with a very specific focus, i.e., a handful of genomic loci.

In this study, we present a novel solution to this problem based on machine vision. We showcase an algorithm that is capable of automating the visual inspection of CNVs with an accuracy and precision comparable to (if not better than) that of a human analyst and distribute it as an R package.

This solution provides us with an unprecedented quality of a genome-wide CNV dataset, which in turn creates new research opportunities as well as new analytical challenges. One of the most impelling challenges being that CNVs are very heterogeneous, for example two CNVs that can be considered identical in practice while not having the exact same boundaries. In this case it is not trivial to decide when they can be treated as the same variant using a single simple rule. For this reason, we also developed a method to group CNVs into biologically plausible CNV regions (CNVRs) based on network analysis, and we demonstrate its function in a selected set of well characterised loci.

Methods

CNV calling and processing

In this study we use CNV calls from two different sample cohorts; a subsample of full trios (offspring, father, mother) from genotyped Icelandic samples at deCODE genetics (deCODE trios), and singleton samples from the UK Biobank (UKB). Samples were genotyped on different chips in different cohorts, Affymetrix UK Biobank Axiom Array for most UKB samples and multiple generations of Illumina arrays for the deCODE trios.

All CNV calls were generated using PennCNV[2]. The CNV calling in the Icelandic trios and UKB was performed at deCODE genetics. In brief, a PennCNV pipeline was used, main settings differing from default were minimum number of SNPs set to 5 and using a filtered SNP list as described below. After calling, all calls were processed using our CNV protocol.[11] In the deCODE data, samples had already been filtered for basic quality control (QC) measures (LRR_SD < 0.35, BAF_DRIFT < 0.01, GCWF between -0.02 and 0.02) while no QC filter was applied in the UKB. Then, individual samples were processed to stitch together consecutive calls if they had the same copy number and were close enough (gap not larger than 20% of the combined length). SNP markers were also filtered as previously described,[11] i.e. only biallelic autosomal SNPs mapping uniquely to the Haplotype Reference Consortium (HRC) hg19 reference map[12], with a minor allele frequency of at least 0.1%. Finally, samples with full or partial chromosomal abnormalities (loosely defined as any deletion or duplication larger than 25Mbp) were excluded. These were defined as chromosomes where CNV calls covered more than 10% of the total markers and were manually validated by a human analyst.

Training data

A schematic of the datasets used in the study is shown in figure 1. The model was trained on 13,078 CNV call examples from a collection of samples from the UKB and the Icelandic biobank at deCODE genetics obtained as follows: 15,000 samples were selected at random from the UKB cohort as well as 800 offspring from the deCODE trios. In the UKB data, a large portion of CNVs belonged to a small set of groups of CNVs (CNV regions, CNVRs). To avoid introducing bias towards specific genomic regions in the model, such regions were down-sampled as follows. CNVRs with a frequency above 20 were divided in three groups based on the frequency within the 15,000 samples: group A, CNVRs with a frequency up to 50, group B, CNVRs with a frequency between 50 and 100, and group C, CNVRs with a frequency above 100. Up to 20 CNVs were sampled from CNVRs in frequency group A, up to 35 from CNVRs in group B and up to 40 from CNVRs in group C.

All CNVs above 25 SNPs and 15 SNPs for UKB and deCODE examples respectively, were manually validated using in-house software[11] (https://github.com/SinomeM/shinyCNV) by at least one human analyst. Each CNV was classified as either true or false, leading to three possible categories: true deletion, true duplication, false call. Very ambiguous calls (e.g., a missed chromosomal aneuploidy) were allowed to be excluded from the training set, however, we tried to limit this as much as possible.

Prediction model design and training

The data representation was designed to be a simplified and condensed version of what the human analyst sees when performing the visual inspection of CNVs (i.e., tracks of LRR and BAF around a putative CNV in each sample). All the processing described here has two main objectives, first to fit all necessary information in the smallest amount of data possible (here, the smallest image) to minimise computation, and second to make the CNVs representation as similar as possible across different genomic loci, different CNVs sizes, different markers numbers and distributions (to minimise any influence of indirect factors on the evaluation of each call). We loosely tested different resolutions before setting it to 96 by 96 pixels. As shown in supplementary figure 1, each image is composed of three rows, containing two views of the LRR values around the CNVs and one view of the BAF. Horizontally the image is composed of three sectors: markers on the left, within and to the right of the CNV call. Depending on the size of the CNV call (below 100kbp, between 100kbp and 1Mbp, above 1Mbp) the two bottom rows will cover 19, 15 or 11 CNVs lengths in total, with equal spacing on each two side, while the top row is always zoomed out by a further factor of three. This is done so that the number of markers in the image is more consistent across CNV length ranges.

The pixel image is constructed as follows. For each of the three rows, all points are converted to the coordinate system, i.e., from genomic position in x and LRR/BAF in y to 0-96 in x and [a, b] in y (where a and b values depend on the specific row). In this smaller space, multiple markers will fall onto the same pixel and the counts for each individual pixel, n, are collected. Then, n is first moved to the log scale and then to [0, 1]. This is done in windows to minimise the effect of differential SNPs density across the image, that could otherwise drown the signal of normal regions, due to small segments of very high SNPs coverage (such as exons). Window size is 4, 8 or 32 pixels, depending on the CNVs length class, as described above. LRR values are processed before being converted to the pixel values system to reduce the effect of outliers. This process is done independently for the three sectors of the image: left side of the CNV, inside the CNV or right side of the CNV. In brief, the farther a point is from the mean value of its sector the more it is 'pulled' towards it, as in the following R pseudo-code (for values above the mean): `new_lrr = lrr - abs(lrr-mean_lrr) * shrink_lrr_factor`, with `shrink_lrr_factor` set to 0.2 by default. In practice, this has very little effect on most points, but should mitigate the effect that outliers might have on the final image. This is done to simulate what the human eye does automatically during visual inspection, i.e., give more "weight" to the centre of the distribution and ignore outliers, if rare. This process does not alter the overall shape of the distribution, especially any waviness pattern that might be present. After this step, any remaining outlier, defined as a point outside three standard deviations of the sector distribution, is removed.

We increased the number of examples using data augmentation in the form of image flipping on the x axis with a probability of 0.5 per example. Other types of data augmentation, such as adding random noise to each pixel, were considered but eventually discarded. Training was performed on the whole dataset in a 75-25% configuration for training and validation set.

The model was built using torch[13] with the R packages torch[14], torchvision[15] and luz[16]. It is composed of 5 convolutional layers (nn_conv2d) followed by a 10-layer linear classifier (nn_linear). Each layer has a 5% dropout rate (nn_dropout2d(p=0.05)). Training was performed using luz::fit() with the following settings: max learning rate of 0.2, max epochs of 50, and early stopping with a patience value of 4.

Test data

Model accuracy was tested in a different set of deCODE samples, while precision was tested in both UKB and deCODE data. The UKB set was designed to also test the calibration of the model probability output as follows. 2800 predicted true CNVs were randomly selected, sampling 400

CNVs from each of the following prediction probability intervals [0.75, 0.85), [0.85, 0.90), [0.90, 0.92), [0.92, 0.94), [0.94, 0.96), [0.96, 0.98), and [0.98, 1]. In the deCODE test set, all CNVs from 1200 randomly sampled individuals were selected. For the UKB calibration set all selected CNVs were manually validated by a human analyst and tagged as either true, false or undetermined/unknown (i.e., calls that could not be ruled out as being true while lacking strong indication of being so). For the purpose of computing accuracy and precision, unknown calls were considered as false. For the deCODE test set we partially relied on the genealogy for validation, and we considered as true any CNVs predicted true by the program and present in at least one of the carrier's parents with at least 50% overlap without the need of manual visual inspection. All discordant CNVs (i.e., those predicted as true by the program but where no concordant call was present in a parent or vice-versa) were manually validated. Accuracy is defined as (TP + TN) / (TP + FP + TN + FN), where TP, TN, FP, and FN are true positives, true negatives, false positives and false negatives respectively, with the truth being the human analyst evaluation for UKB, and both model-genealogy agreement and human evaluation in the deCODE sets.

CNVR detection algorithm

The CNVR detection algorithm is composed of three separate steps. First, CNVs are divided into networks, defined as the smallest set of mutually overlapping CNVs. By definition, the largest possible network is thus a chromosomal arm (in our pipeline CNVs cannot span across the centromere), and the smallest network is a single CNV that does not overlap with any neighbour. The second step consists in creation of the CNVRs. For each network, an NxN matrix is constructed, where N is the number of CNVs in the network and each entry is the *intersection over the union* (IOU) between a pair of CNVs. Two identical CNVs will have an IOU of 1, two overlapping CNVs will have an IOU between 0 and 1, depending on the similarity, and two non-overlapping CNVs will have an IOU closer to -1 the more distant they are. This matrix is then translated into a weighted igraph[17] network object where all pairs with an IOU above a certain minimum value will be connected, with the IOU values being the strength of the connection. Default minimum IOU value is set to 0.75. Higher minimum values tend to create smaller, tighter networks, while lower minimum values tend to create larger and potentially more shallow networks. We do not recommend setting the minimum IOU value below 0.5. This network is then processed with the community detection algorithm Leiden[18] (igraph::cluster_leiden(), default resolution value is set to 1), and the resulting networks are converted back into CNVRs with the boundaries being the median values of the start and end of the constitutive CNVs respectively. CNVs with no CNVR (singletons) are given their own group. The third and final step is an optional forced merge of CNVRs. For each chromosome, all CNVRs are processed in order of start position. For any given CNVR x, all regions with a reciprocal overlap above a certain threshold with x (default value is 0.75) are merged, the new start and end are computed again as the median from all the CNVs in the new group. This process continues until all CNVRs have been either processed or merged with a previous one. This is done for a fixed number of iterations, 5 by default. Each CNVR can be modified only once per iteration, thus the higher the number of iterations, the more CNVRs will be merged. The idea is to combine very similar CNVRs, especially spurious results from the community detection algorithm, without moving the boundaries of the region too far away from the corresponding CNVs.

When computing CNVRs on genome wide CNVs, we defined three groups of settings to test, very strict, loose, and in between, defined as follows. Very strict (min_iou=0.9, max_force_merge_rounds=1, force_merge_min_overlap=0.9) maximise intra-CNVR similarity, loose (min_iou=0.5, max_force_merge_rounds=5, force_merge_min_overlap=0.75) minimise the number of regions created, while the settings for the in-between group (min_iou=0.75, max_force_merge_rounds=3, force_merge_min_overlap=0.75) should be a balance between the two extremes.

Software availability

The program is available as an R package at https://github.com/SinomeM/CNValidatron_fl/. The visual validation interface is available at https://github.com/SinomeM/shinyCNV. Most of the data processing was done using the R package QCtreeCNV available at https://github.com/SinomeM/QCtreeCNV and previously described.[11]

Results

PennCNV false positive calls

In this study we used CNV calls from two different cohorts, genotyped on different arrays from two manufacturers to investigate and mitigate the burden of false positive CNV calls in large datasets. As described in the methods, these consist of UK Biobank (UKB)[19] and deCODE trios (deCODE genetics / Amgen, Reykjavík, Iceland).

We assessed the PennCNV false positive rate in three different CNV sets from two cohorts. These consist of 1) UKB training set, 2) deCODE training set, 3) deCODE test set. 15,000 samples were randomly selected in the UKB and 2,000 offspring from full trios in the deCODE sets (800 train and 1,200 test respectively), for a total of 17,000 unique samples. All CNVs from these samples were processed and filtered for 25 minimum number of SNPs in the UKB and 15 in deCODE data (table 1). All CNVs were manually validated for the deCODE portion of the training set, while common CNVRs were down sampled in the UKB training set (see methods). This was done to reduce the model bias towards specific common loci and was deemed necessary only in the UKB portion of the training set. Finally, CNVs in the deCODE test set were validated only in the case of discordance with the genealogy, as described in the methods. In total almost 22,168 CNVs were validated (12,153 UKB train, 3,760 deCODE train, 6,255 deCODE test). CNVs were evaluated by at least one human analyst and tagged as either true or false. Results are shown in table 1. In summary, ~30% to ~60% of the CNV calls produced by PennCNV are false positives depending on the specific cohort and the minimum number of SNPs.

Prediction model training and testing

The model training was performed on a combined set of human-validated CNV calls from UKB and deCODE, for a total of 13,078 unique examples. Training stopped after 12 epochs with an accuracy of 0,96.

Accuracy and precision are 95% and 92% in the deCODE test set respectively (table 2), while precision is 95% in the UKB calibrations set. We also tested the calibration of the probability outputted by the model in the UKB. We randomly selected a total of 2,800 predicted true CNVs from intervals of probabilities. As shown in supplementary figure 4, the output probability is not properly calibrated, but can be used as a lower limit, meaning that selecting CNVs predicted true with at least 0.75 probability will lead to a dataset where at least 75% of CNVs would be evaluated as true by a human analyst. In practice this percentage will likely be much higher (95% in this within-sample set).

PennCNV false negatives estimation and *de novo* CNV rate

In contrast with false positives, false negatives are essentially impossible to quantify without other means of detecting CNVs. We attempted to get an estimate of the false negative rate using the deCODE trios from our analysis of the calls where the model prediction was not supported by parent-offspring segregation consisting of 637 CNVs, excluding the MHC region. Of these, 99 were found to be true positives in the offspring and false negatives in the parent, meaning that the CNV was present (based on visual inspection) but had been missed by PennCNV in the parent. The majority of these (96 of 99) are duplications and the median length is ~90kbp, with 11 being larger than 200kbp. This would set the minimum false negative CNV call rate at ~1.5% (99 out of 6522 CNVs) and ~4.0% (96 out of 2413) for duplications only.

Moreover, 42 CNVs (30 deletions, and 12 duplications) were found to be true in the offspring but not present in the parents with a median length of ~161kbp, setting the minimum *de novo* CNV mutation rate at ~3.5% (42 events in 1,200 samples).

False positives calls and QC measures

We investigated the link between false positive PennCNV calls and quality measures, LRR standard deviation (LLR_SD), BAF drift, and GC waviness factor (GCWF). As shown in supplementary figure 3, false CNVs (especially deletions) tend to be from samples with somewhat higher LRR SD and BAF drift, but not GCWF. As expected, undetermined CNVs lie in between true and false ones. Moreover, true CNVs (especially duplications) tend to be larger than false and undetermined ones (supplementary figure 2). True duplications also tend to have a higher number of markers than any other categories. This is consistent with our experience, as duplications tend to need more points to be reliably considered true evaluated by the analyst compared to deletions.

We also compared our program accuracy against standard QC filtering from previous literature in the UKB train set, as Kendall et al.[20]. Excluding samples with a $|WF| > 0.03$, more than 30 CNVs or $LRRSD > 0.35$. In the UKB train set, this excluded 144 (2%) true, 1,163 (54%) false and 122 (6%) undetermined CNVs, removing 24% of the CNVs evaluated as not true (false + undetermined) by a human analyst while sacrificing only 2% of the true positives.

Differences in distribution between true and false positive calls

To understand the effect that insufficient filtering of false positive CNVs can have on a genome wide analysis, we plotted the genomic distribution of the UKB train set, marking human validated true and not-true CNVs (supplementary figure 5). Similarly, to the overall CNVs distribution, differences in true and not-true distribution vary widely both within and across chromosomes. In the UKB train set (panel a), the distribution of true and false CNV calls is very similar for chromosomes 16 and 18, and 5 and 9 for deletions and duplications respectively. In contrast, they are very different in chromosomes 9 and 20 for deletions, and 11 and 20 for duplications. In the deCODE test set (panel b), we did not identify a chromosome where the distribution of true and false CNVs followed each other as closely as in the UKB train set. In contrast the two distributions are very different in chromosomes 8, 13 and 17 for both deletions and duplications.

CNV regions algorithm

CNV regions (CNVRs) do not have a unanimous definition in the literature. In this study we defined them as the results of grouping CNVs based on similarity, maximising the within-group similarity while minimising the number of different groups. We tested multiple algorithms based on incrementally growing CNVRs, that is add a new CNV to a group if it is similar enough to its members (e.g., in terms of reciprocal overlap or intersection over the union, IOU), or create a new group if no match is found. We found that such approaches tend to be slow and more importantly, give potentially very different results based on the order in which CNVs are evaluated. For this reason, we developed an algorithm to compute CNVRs in a whole locus at the same time, based on the community detection method Leiden, from network analysis.

We applied our CNVR detection algorithm to all CNVs predicted as true deletions or duplication with a probability above 75% in the UKB (~525,000). We tested three sets of settings (very strict, in-between, and loose) and evaluated the results at four well characterised genomic loci: the *NRXN1* gene locus on the 2p arm, and the recurrent CNV loci 15q11.2-13.3, 16p13.1-p12.3, and the 22q11.2 broader region (figure 2 and supplementary figure 6). The most stringent setting (green in supplementary figure 6) creates too many groups, given the high internal similarity requirements, and it is not shown in figure 2. Between the two, neither is preferable in all possible situations. As shown in figure 2, some CNVRs seem to be very specific, meaning that all CNVs within them are extremely similar, and are detected with the same number of carriers in both methods. Two examples demonstrating this are the 127 carriers CNVR in 15q13.2 marked with '*' (figure 2 panel a) and the 30 carriers one in 16p13.11 marked with '~' (figure 2 panel b). However, in most cases the two settings will produce slightly different results. The looser setting tends to create fewer CNVRs that are less specific, meaning CNVs in them can have less closely aligned boundaries. In figure 2 we annotated two such cases in known CNV loci. In 15q13.3, with loose settings there is a CNVR including 2,815 carriers overlapping the known CHRNA7 copy-number-variable region marked with '+' that is separated into three different regions with 2,281, 484 and 168 carriers respectively when applying the strict/intermediate setting. Conversely, in 16p13.11 there are 756 carriers of a CNVR marked with '#' covering the known recurrent locus that the medium setting correctly separates in two groups, the most frequent including an additional group of markers on the right (Mb 15.0-15.2), whereas the loose setting groups the two CNV subtypes together in one CNVR. Depending on the application, researchers can balance the number and carrier frequency of CNVRs with their specificity in terms of internal similarity.

Discussion

In this study we provide the first in-depth analysis of the extent of false positive CNV calls from PennCNV across two cohorts genotyped on different SNP arrays from two different manufacturers. We demonstrate that false positives constitute a large fraction of calls across cohorts, while depending somewhat on factors such as array type (Illumina or Affymetrix), sample quality, and the number of probes in each CNV call. Moreover, we show that standard QC filtering practice only remove a minor portion of the false positives in UKB data. Finally, we show that false and true CNV calls do not follow the same genomic distribution. Studies that do not use a strong method to exclude false positive calls (such as visual inspection) will therefore likely produce highly biased results.

Of note, PennCNV produced fewer CNV calls on Affymetrix arrays compared to Illumina, both before and after filtering. Possible reasons for this include differences in markers density between the two manufacturers, lower signal to noise ratio in Affymetrix data, and lower sensitivity for Affymetrix data from the PennCNV model. While using only one algorithm (PennCNV) to call CNVs, we do not think this is a strong limitation as, 1) PennCNV is by far the most widely used software to call CNV in SNP array data, 2) PennCNV has been shown to be the most reliable software in a previous comparison[23] and 3) most programs (with the notable exception of ipattern[21]) uses the same raw data (LRR and BAF) and some implementation of the same HMM architecture as PennCNV.[5,22,23]

We also provide, to our knowledge, the first estimate of false negative rate in CNV calls. Given the high number of false positives, it is normally assumed that PennCNV is over-sensitive, and that false negatives are not an issue. Indeed, we find them to be most likely less than the false positives, with an estimate from the deCODE trios' data setting the minimum to 1.6%, and mostly confined to smaller duplications. Considering most CNVs are rare or ultra-rare exposures the false negative rate might still be a source of bias in some contexts, and particularly difficult to mitigate *in-silico*, especially without access to full pedigrees, as is the case in most modern biobanks.

Visual inspection remains the best solution to the false positives in PennCNV calls, and arguably the most reliable. However, it is extremely time consuming, and remains virtually unfeasible for genome wide applications in large collections. Here, we provide software to automate CNVs validation using computer vision, with an accuracy comparable to a trained human analyst, across multiple datasets.

In our view, the major limitation of this work is the absence of an "absolute truth" to compare with. Both the training examples and the test sets are based on visual inspection by a human analyst, and this leads to possible biases. Even though a wet-lab validation using qPCR experiment would be the *ideal* gold standard, visual inspection remains the *practical* gold standard in the field, and the best option available on such a scale and given the sensitivity of the data. Moreover, having access to a full pedigree, allowing to compare the same region across closely related samples, reduces uncertainty in human evaluation.

We also propose an algorithm to compute CNV regions. Also in this case, there is no clear "gold standard" nor unanimous definition of what such regions should look like or how they should behave genome-wide. For our purposes we define CNVRs as "the smallest possible number of sets that can group CNVs that are as similar as possible together". Clearly, low number of groups and internal similarity cannot be maximised at the same time. We show how different settings of our algorithm can produce different "flavours" of CNV regions. Strict setting will create few but more heterogenous groups, which might be ideal to limit the number of variants to test in an associations study, with loose setting will produce several but more homogenous regions, which might be ideal when the focus is on precise genomic structures such as a specific gene. We showcase the results of our algorithm at different well-characterised loci, and we believe some degree of manual scrutiny is required when computing CNVRs, however it's impossible to manually check the entire genome for coherence.

In conclusion, despite some important but mostly out-of-our-control limitations, we believe the software described is capable of removing the vast majority of false positives from a CNV calls set genome wide, and not introduce strong false negative issues. This will open the field to new and interesting research opportunities.

Bibliography

1. Freeman, J. L. *et al.* Copy number variation: new insights in genome diversity. *Genome Res.* **16**, 949–961 (2006).
2. Wang, K. *et al.* PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res.* **17**, 1665–1674 (2007).
3. Taliun, D. *et al.* Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. *Nature* **590**, 290–299 (2021).
4. Halldorsson, B. V. *et al.* The sequences of 150,119 genomes in the UK Biobank. *Nature* **607**, 732–740 (2022).
5. Seiser, E. L. & Innocenti, F. Hidden Markov Model-Based CNV Detection Algorithms for Illumina Genotyping Microarrays. *Cancer Inform.* **13s7**, CIN.S16345 (2014).
6. Panda, A. *et al.* Genome-wide analysis and visualization of copy number with CNVpytor in igv.js. *Bioinforma. Oxf. Engl.* **40**, btae453 (2024).
7. Winchester, L., Yau, C. & Ragoussis, J. Comparing CNV detection methods for SNP arrays. *Brief. Funct. Genomic. Proteomic.* **8**, 353–366 (2009).
8. Marshall, C. R. *et al.* Contribution of copy number variants to schizophrenia from a genome-wide study of 41,321 subjects. *Nat. Genet.* **49**, 27–35 (2017).
9. Vaez, M. *et al.* Population-based Risk of Psychiatric Disorders Associated with Recurrent CNVs. 2023.09.04.23294975 Preprint at https://doi.org/10.1101/2023.09.04.23294975 (2023).
10. Crawford, K. *et al.* Medical consequences of pathogenic CNVs in adults: analysis of the UK Biobank. *J. Med. Genet.* **56**, 131–138 (2019).
11. Montalbano, S. *et al.* Accurate and Effective Detection of Recurrent Copy Number Variants in Large SNP Genotype Datasets. *Curr. Protoc.* **2**, e621 (2022).

12. McCarthy, S. *et al.* A reference panel of 64,976 haplotypes for genotype imputation. *Nat. Genet.* **48**, 1279–1283 (2016).

13. Ansel, J. *et al.* PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)* (ACM, 2024). doi:10.1145/3620665.3640366.

14. Falbel, D. & Luraschi, J. torch: Tensors and Neural Networks with 'GPU' Acceleration. (2024).

15. Falbel, D. torchvision: Models, Datasets and Transformations for Images. (2024).

16. Falbel, D. luz: Higher Level 'API' for 'torch'. (2024).

17. Csárdi, G. *et al.* igraph for R: R interface of the igraph library for graph theory and network analysis. Zenodo https://doi.org/10.5281/zenodo.10681749 (2024).

18. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).

19. Sudlow, C. *et al.* UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Med.* **12**, e1001779 (2015).

20. Kendall, K. M. *et al.* Cognitive performance and functional outcomes of carriers of pathogenic copy number variants: analysis of the UK Biobank. *Br. J. Psychiatry* **214**, 297–304 (2019).

21. Pinto, D. *et al.* Functional impact of global rare copy number variation in autism spectrum disorders. *Nature* **466**, 368–372 (2010).

22. Colella, S. *et al.* QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Res.* **35**, 2013–2025 (2007).

23. Seiser, E. L. & Innocenti, F. Hidden Markov Model-Based CNV Detection Algorithms for Illumina Genotyping Microarrays. *Cancer Inform* **13s7**, CIN.S16345 (2014).

Figures

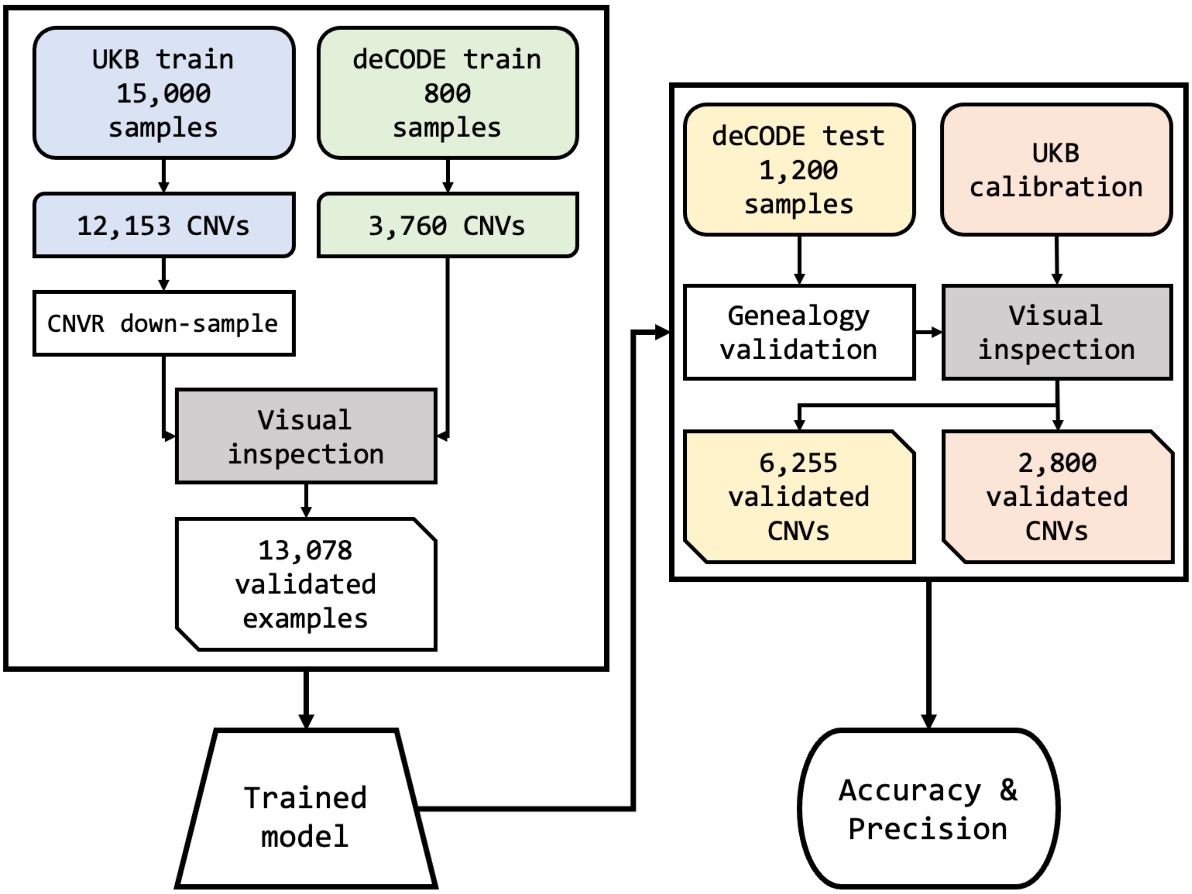


Figure 1: Schematic representation of the training and test sets across the two cohorts.

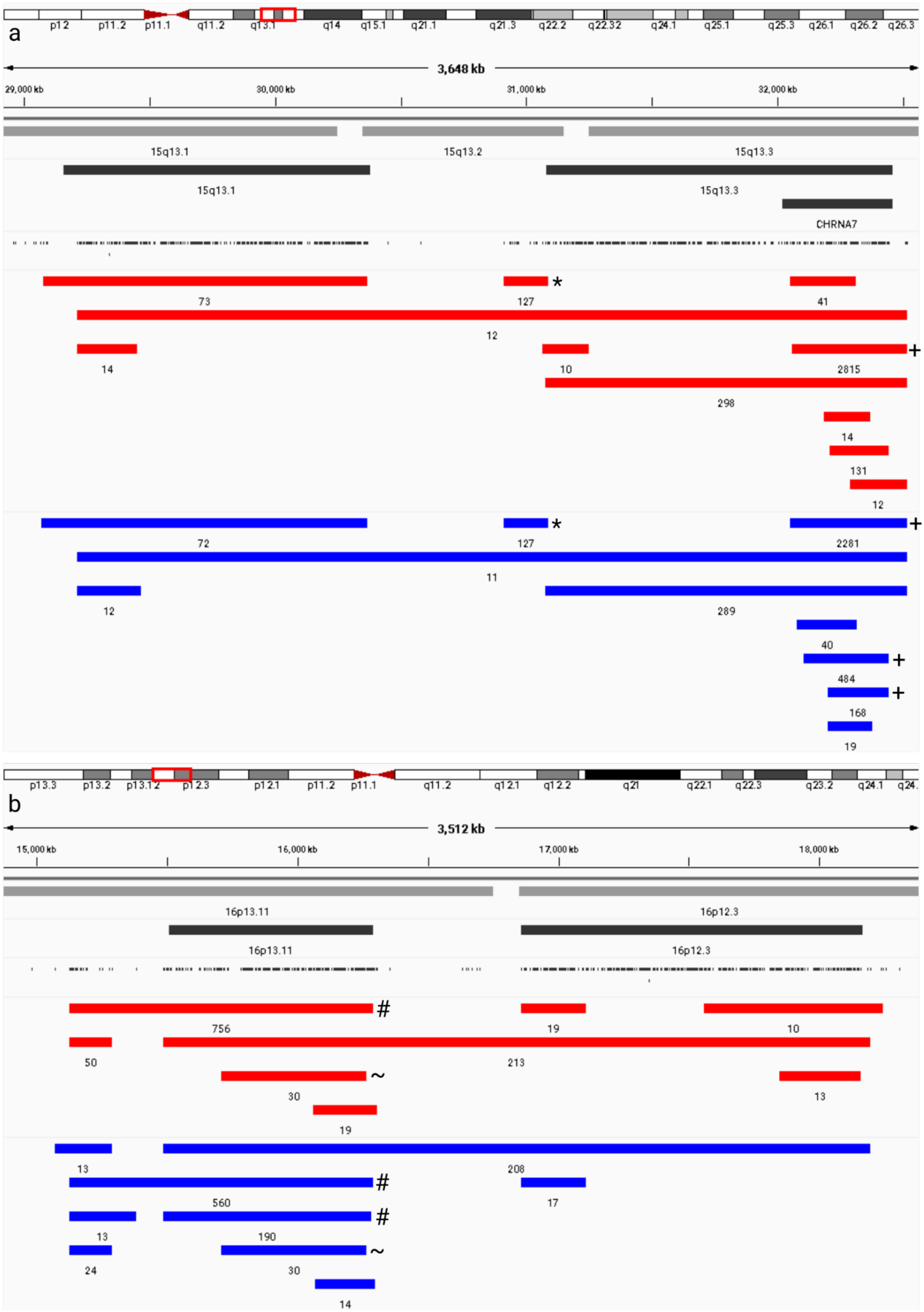

a
p12
p11.2
p11.1
q11.2
q13.1
q14
q15.1
q21.1
q21.3
q22.2
q22.32
q24.1
q25.1
q25.3
q26.1
q26.2
q26.3
3,648 kb
29,000 kb
30,000 kb
31,000 kb
32,000 kb
15q13.1
15q13.2
15q13.3
15q13.1
15q13.3
CHRNA7
73
127
*
41
12
14
10
2815
+
298
14
131
12
72
127
*
2281
+
11
12
289
40
484
+
168
+
19
b
p13.3
p13.2
p13.12
p12.3
p12.1
p11.2
p11.1
q11.2
q12.1
q12.2
q21
q22.1
q22.3
q23.2
q24.1
q24.
3,512 kb
15,000 kb
16,000 kb
17,000 kb
18,000 kb
16p13.11
16p12.3
16p13.11
16p12.3
756
#
19
10
50
213
30
~
13
19
13
208
560
#
17
13
190
#
24
30
~
14

Figure 2: IGV plot showing CNVRs computed with different settings. From the top, the light grey track shows the cytobands, the dark grey show known recurrent CNV loci, and the second dark grey shows the SNP markers coverage. The red track shows CNVRs computed with the looser setting, while the blue one shows the medium settings. The strict setting track is shown only in supplementary figure VII in green. The number below each CNVR is the number of CNV belonging to it. a) 15q13.3 b) 16p13.1-12.3. CNVRs with a frequency below 10 are not shown.

Tables

Dataset	Chip	Min length	Min SNPs	Tot samples	Tot CNVs	CNV/ Sample	True (%)	False (%)	Unknown (%)
deCODE train[1]	Multiple	NA	15	800	3760	4.7	1547 (41%)	2213 (59%)	0 (0%)
deCODE test[1]	Multiple	NA	15	1200	6522	5.4	2790 (43%)	3732 (57%)	0 (0%)
UKB train[2]	Affymetrix	NA	25	15,000	12153	0.8	6829 (56%)	3296 (27%)	2028 (17%)
UBK calibration	Affymetrix	NA	15	NA	2800	NA	2665 (95%)	5 (0%)	130 (5%)

[1]: for deCODE data, multiple full trios were inspected together, this essentially eliminated the need for the undetermined/unknown tag.
[2]: common CNVRs were down sampled in this set.

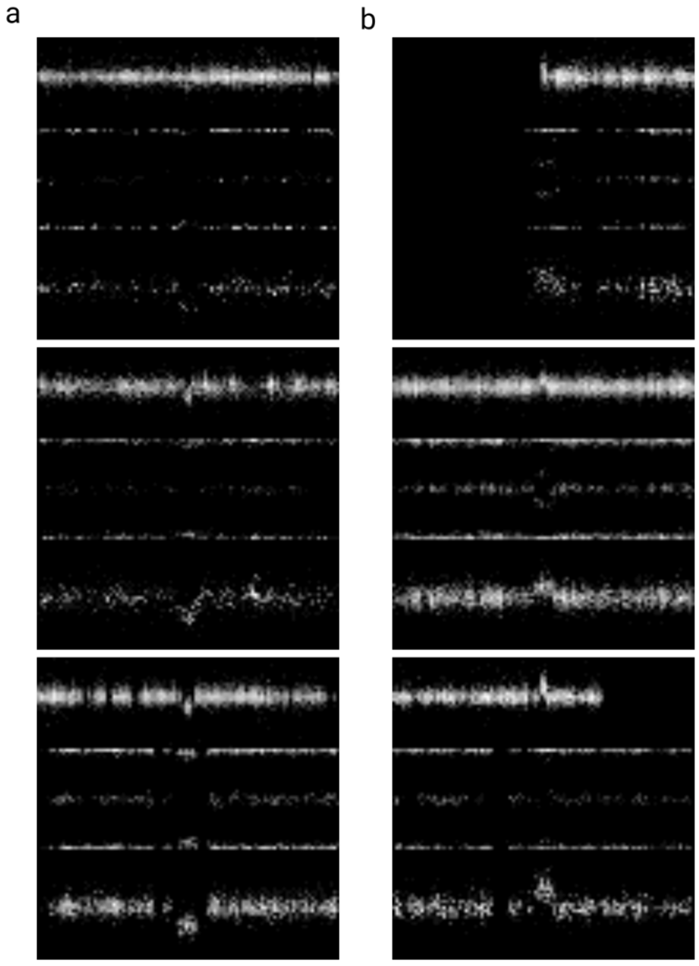
Table 1: Results from human visual validation in the different CNV sets. Training and test sets consist of all CNVs from a randomly selected set of samples, with some filters applied.

Dataset	**TP**	**TN**	**FP**	**FN**	**Accuracy (TP+TN)/All**	**Precision TP/(TP+FP)**
deCODE test	2578	3639	93	212	95%	92%
UBK calibration	2666	NA	134	NA	NA	95%

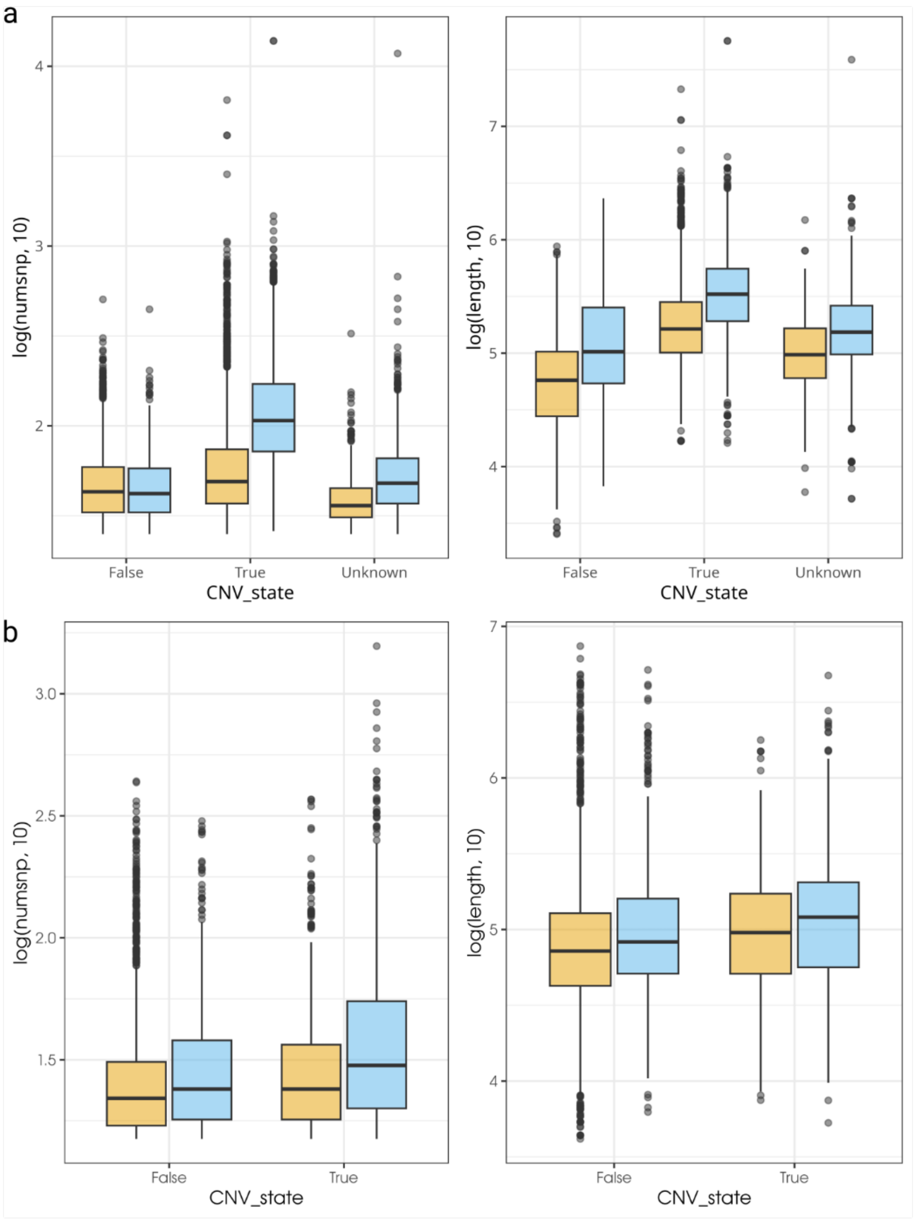
Table 2: Accuracy and precision in the different CNV sets.

Supplementary material

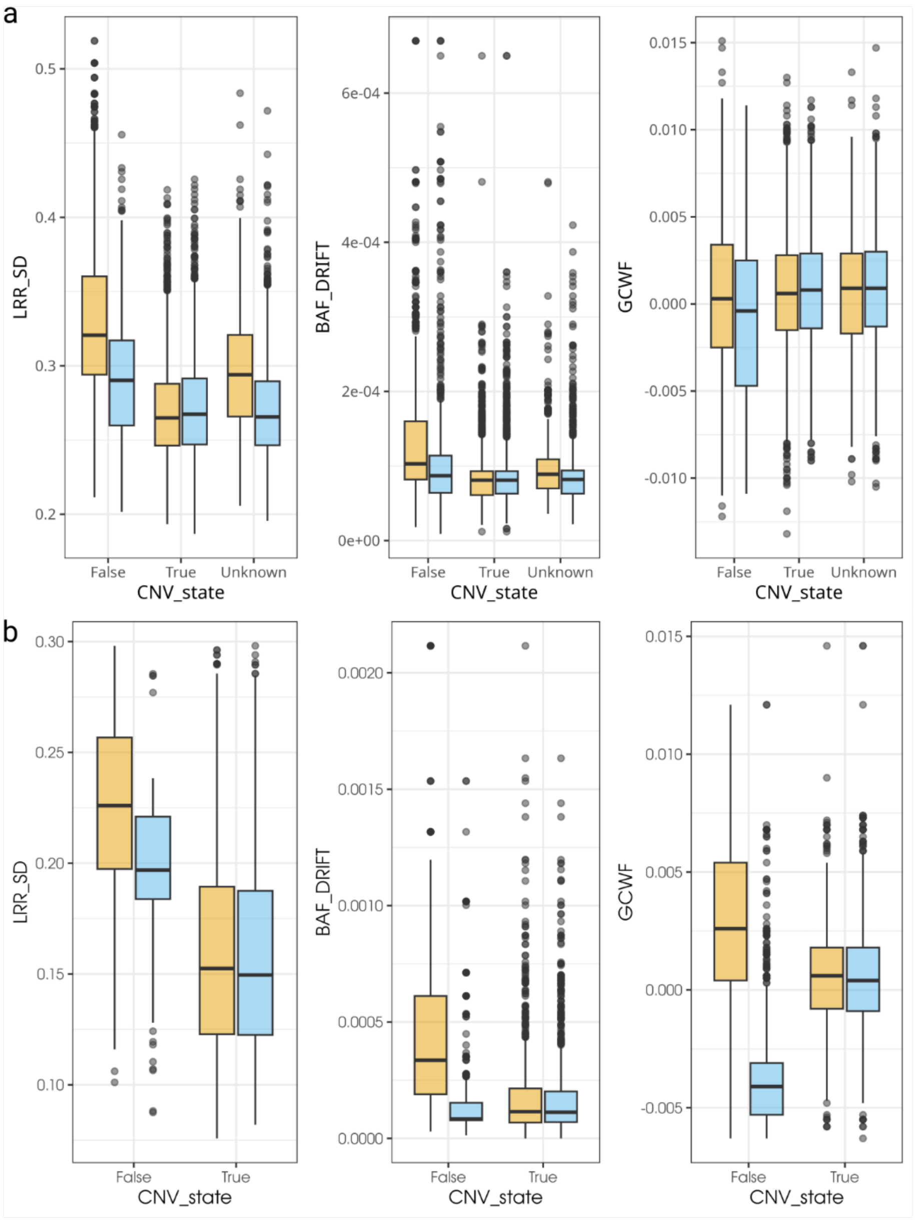
Supplementary figure 1. Examples of the 96x96 pixels image for the CNN. Three examples of true deletions are shown in a, and three examples of true duplication in b. In each image it can be recognised the three data rows described in the methods, form the top, a zoomed-out view of the LRR values around the CNV call, the BAF in the middle, and the LRR values on the bottom. The deletions showcase the characteristic "dip" in LRR value and a loss of heterozygosity in the BAF (especially clear in the bottom example), while the duplications showcase the increase in LRR values and the split of central BAF band (especially clear in the first and second from the top).



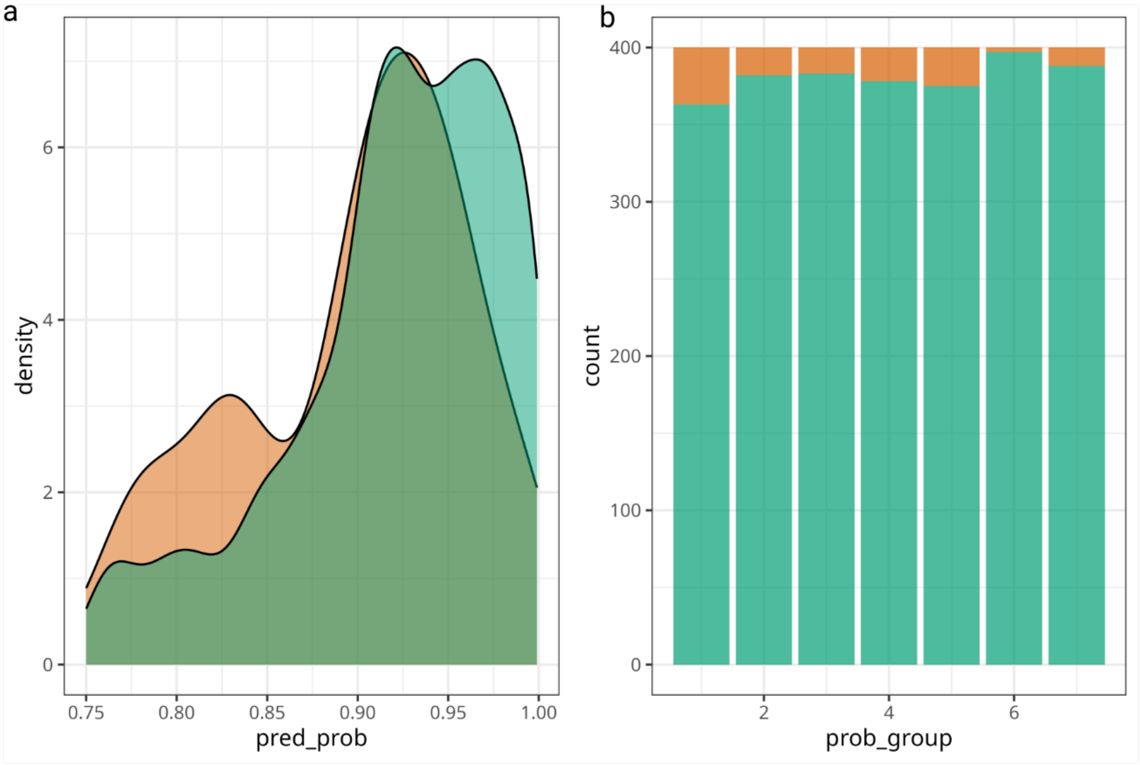
Supplementary figure 2. Boxplots showing the distributions of number of markers and length of CNVs across the different categories: deletion/duplication, true/false/unknown. Log 10 number of SNP markers (left) and length (right) in the true, false and undetermined/unknown CNVs. Deletions in yellow, duplications in blue. a) UKB train, b) deCODE test. Duplications tends to be larger and have more markers than deletions, especially in UKB data. True CNVs also tend to be larger and have more markers than false/unknow ones, again, especially in UKB.



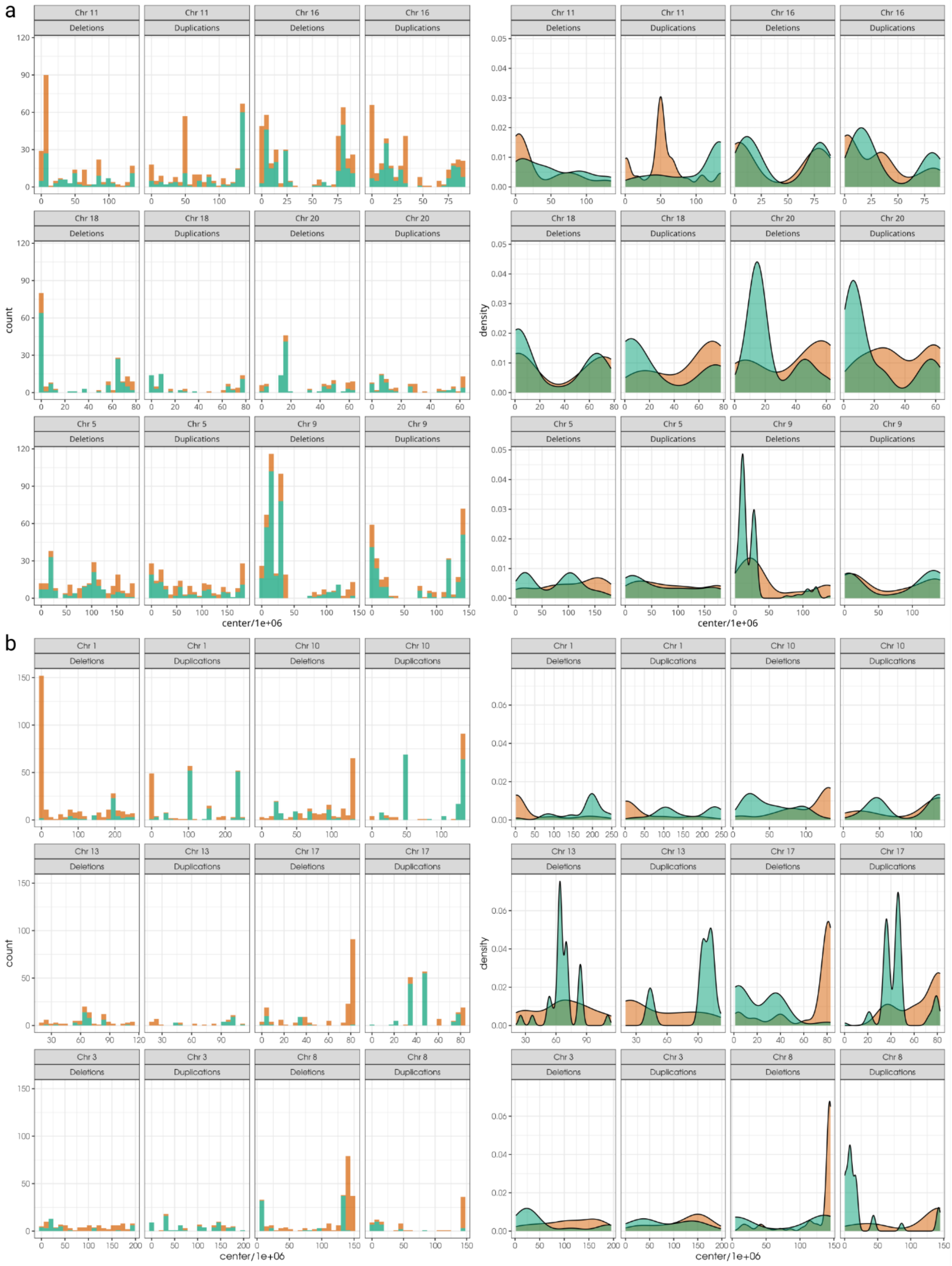
Supplementary figure 3. Boxplots of the three main QC measures across the different categories: deletion/duplication, true/false/unknown. LRR SD (left), BAF drift (middle), and GCWF (right) distributions in the true, false and undetermined/unknown CNVs. Deletions in yellow, duplications in blue. Each CNV inherits the QC measure of the sample. a) UKB train, b) deCODE test. False CNVs tend to belong to samples with high LRR SD in both UKB and deCODE data, the same is true for false deletions and BAF drift. GCWF seems to be different only for false deletions in deCODE data.



Supplementary figure 4. Results of Visual validation in the UKB calibration set. All CNVs have been predicted as true by the model and have been validated by a human analyst. 400 CNVs were sampled for each probability group, as follows. 1: [0.75, 0.85), 2: [0.85, 0.90), 3: [0.90, 0.92), 4: [0.92, 0.94), 5: [0.94, 0.96), 6: [0.96, 0.98), and 7: [0.98, 1]. a) overall distribution of the prediction probability in the true (green) and false (orange) CNVs. b) bar plot of the visual validation status in each prediction probability group. Lower groups seem to have a higher proportion of false CNVs, but there is no clear overall pattern.

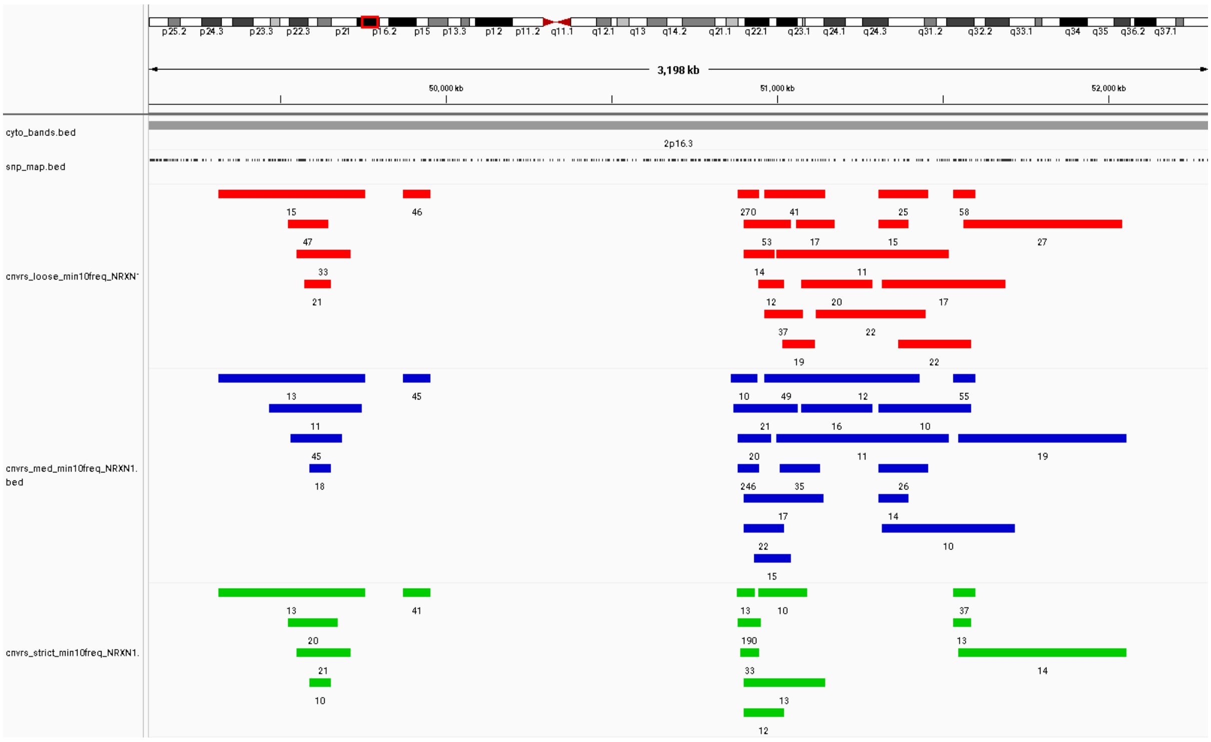


Supplementary figure 5. Genomic distribution of the True and False CNV calls in the a) UKB train set, and b) deCODE test. True CNVs in green, false in orange. Deletion and duplications are plotted separately for a selection of chromosomes, both stacked histograms, on the left, and density plots, on the right, are provided. In both plots type, the x axes show the genomic coordinate of the centre position of CNVs, while on the y axes show the count for each bin and the density for the staked histogram and density plot respectively. In the density plots, the density for true and false CNVs is computed separately, so the histogram shows the overall CNVs distribution better, while the in density plots it is easier to see the individual distribution of the two groups. The plots show how in some chromosomes true and false CNVs seem to reflect the same distribution, while in some other chromosomes they follow completely different patterns.

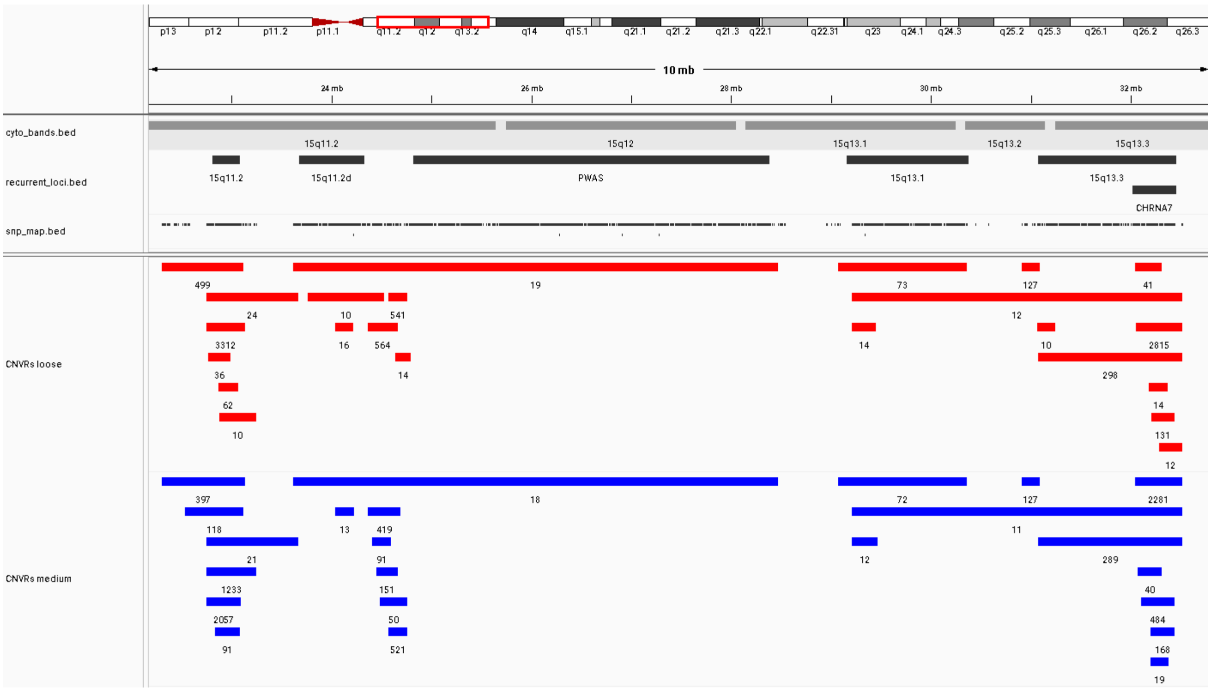

a
Chr 11 Deletions
Chr 11 Duplications
Chr 16 Deletions
Chr 16 Duplications
Chr 18 Deletions
Chr 18 Duplications
Chr 20 Deletions
Chr 20 Duplications
Chr 5 Deletions
Chr 5 Duplications
Chr 9 Deletions
Chr 9 Duplications
count
density
center/1e+06
b
Chr 1 Deletions
Chr 1 Duplications
Chr 10 Deletions
Chr 10 Duplications
Chr 13 Deletions
Chr 13 Duplications
Chr 17 Deletions
Chr 17 Duplications
Chr 3 Deletions
Chr 3 Duplications
Chr 8 Deletions
Chr 8 Duplications
count
density
center/1e+06

Supplementary figure 6. Showcase of CNVRs algorithm results. This is an extended version of figure 2. CNVRs computed with different settings. Red: loose, blue: medium, green: strict. Four different genomic loci are shown: a) *NRXN1* gene locus (chr2:49,645,643-51,759,674), b) 15q11.2-13.3 (chr15:22,600,000-32,600,000), c) 16p13.1-12.3 (chr16:15,000,000-18,300,000), d) 22q11.1-11.2 (chr22:17,200,000-23,800,000). For the sake of space, the green track is shown only in panels a and d. This is due to the high number of CNVRs produced by the strict setting. Only regions with a minimum frequency of 10 are plotted, so this effect is not present in panel a and only slightly present in panel d, where the number of total CNVs is lower than panels b and c and thus most small CNVRs are not plotted.

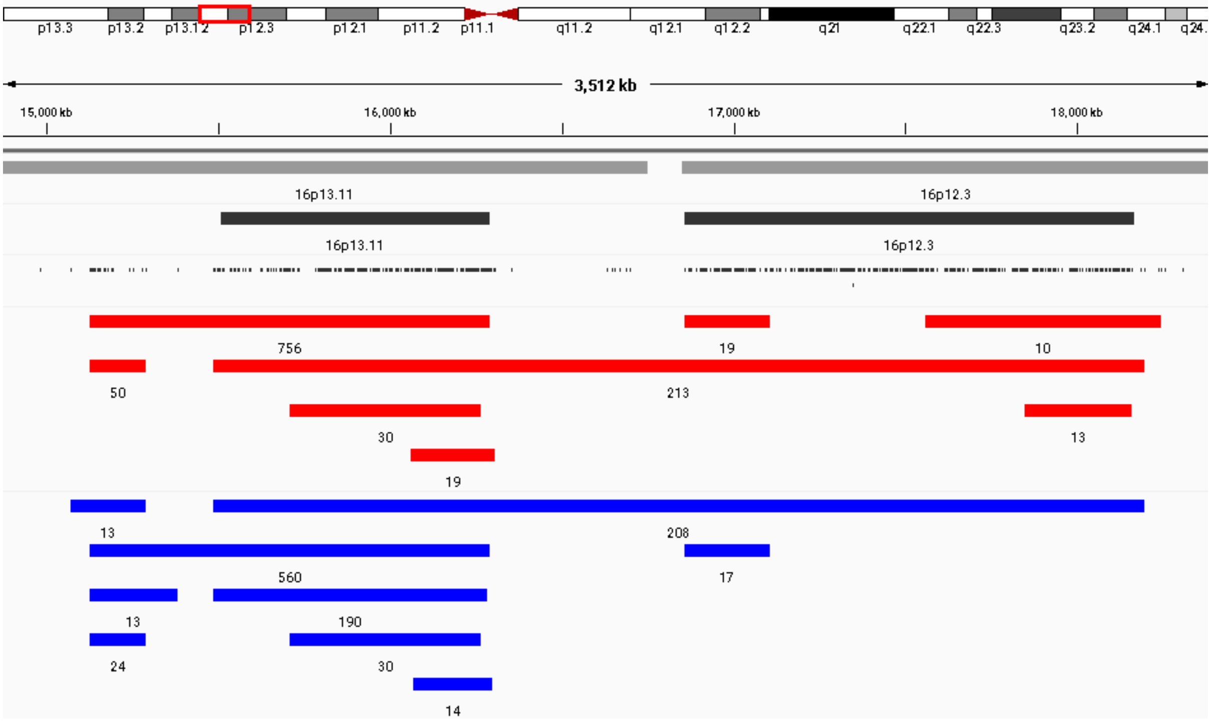
a)


p25.2 p24.3 p23.3 p22.3 p21 p16.2 p15 p13.3 p12 p11.2 q11.1 q12.1 q13 q14.2 q21.1 q22.1 q23.1 q24.1 q24.3 q31.2 q32.2 q33.1 q34 q35 q36.2 q37.1
3,198 kb
50,000 kb
51,000 kb
52,000 kb
cyto_bands.bed
2p16.3
snp_map.bed
cnvrs_loose_min10freq_NRXN'
15 46 270 41 25 58
47 53 17 15 27
33 14 11
21 12 20 17
37 22
19 22
cnvrs_med_min10freq_NRXN1.bed
13 45 10 49 12 55
11 21 16 10
45 20 11 19
18 246 35 26
17 14
22 10
15
cnvrs_strict_min10freq_NRXN1.
13 41 13 10 37
20 190 13
21 33 14
10 13
12

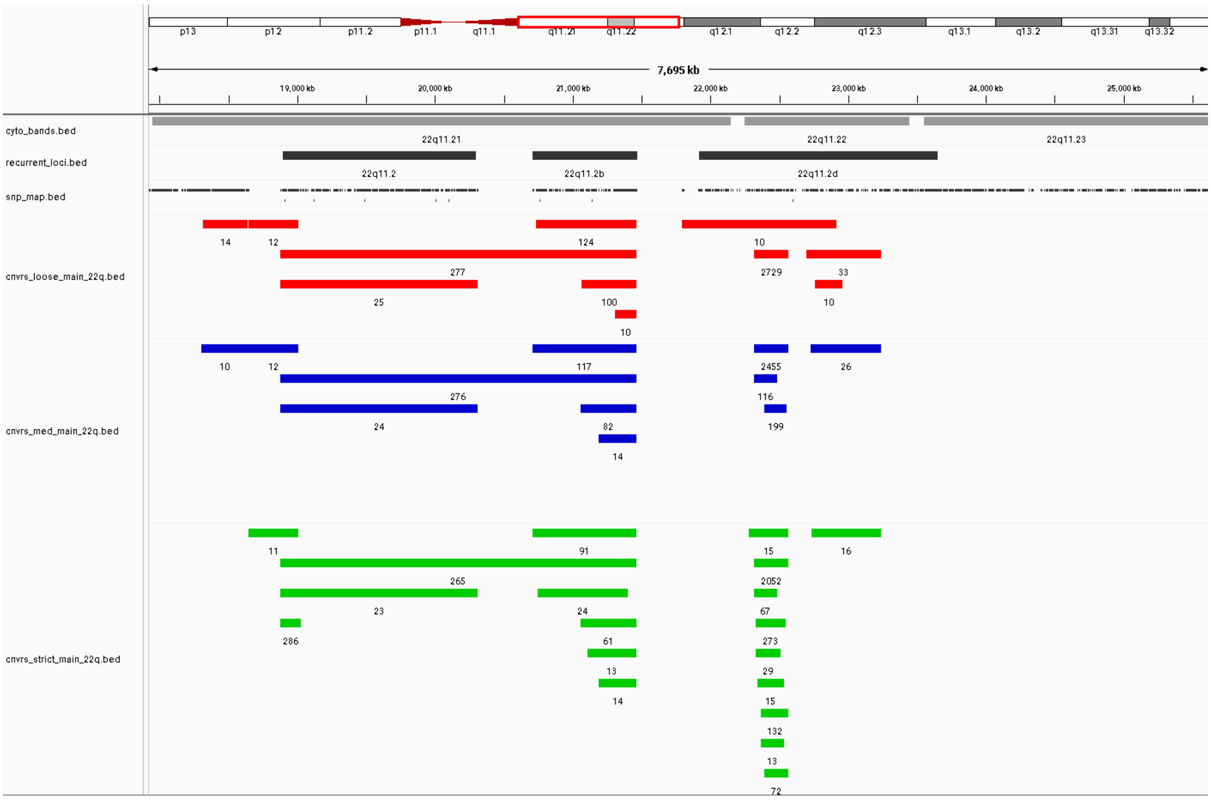
b)


p13 p12 p11.2 p11.1 q11.2 q12 q13.2 q14 q15.1 q21.1 q21.2 q21.3 q22.1 q22.31 q23 q24.1 q24.3 q25.2 q25.3 q26.1 q26.2 q26.3
10 mb
24 mb
26 mb
28 mb
30 mb
32 mb
cyto_bands.bed
15q11.2 15q12 15q13.1 15q13.2 15q13.3
recurrent_loci.bed
15q11.2 15q11.2d PWAS 15q13.1 15q13.3
CHRNA7
snp_map.bed
CNVRs loose
499 19 73 127 41
24 10 541 12
3312 16 564 14 10 2815
36 14 298
62 14
10 131
12
CNVRs medium
397 18 72 127 2281
118 13 419 11
21 91 12 289
1233 151 40
2057 50 484
91 521 168
19

c)


p13.3
p13.2
p13.12
p12.3
p12.1
p11.2
p11.1
q11.2
q12.1
q12.2
q21
q22.1
q22.3
q23.2
q24.1
q24.
3,512 kb
15,000 kb
16,000 kb
17,000 kb
18,000 kb
16p13.11
16p12.3
16p13.11
16p12.3
756
19
10
50
213
30
13
19
13
208
560
17
13
190
24
30
14

d)


p13
p12
p11.2
p11.1
q11.1
q11.21
q11.22
q12.1
q12.2
q12.3
q13.1
q13.2
q13.31
q13.32
7,695 kb
19,000 kb
20,000 kb
21,000 kb
22,000 kb
23,000 kb
24,000 kb
25,000 kb
cyto_bands.bed
recurrent_loci.bed
snp_map.bed
cnvrs_loose_main_22q.bed
cnvrs_med_main_22q.bed
cnvrs_strict_main_22q.bed
22q11.21
22q11.22
22q11.23
22q11.2
22q11.2b
22q11.2d
14
12
124
10
277
2729
33
25
100
10
10
10
12
117
2455
26
276
116
24
82
199
14
11
91
15
16
265
2052
23
24
67
286
61
273
13
29
14
15
132
13
72

A genome-wide characterisation of large Copy Number Variations in two population-scale datasets genotyped on different SNP arrays

Simone Montalbano[1], G. Bragi Walters[2], Guðfjörn F. Jónsson[2], Hreinn Stefánsson[2], Thomas Werge[1,3], Kári Stefánsson[2] & Andrés Ingason[1]

[1]*Institute of Biological Psychiatry, Mental Health Services, Copenhagen University Hospital, Roskilde, Denmark.*

[2]*deCODE genetics/Amgen, Inc., Reykjavik, Iceland.*

[3]*Department of Clinical Medicine, University of Copenhagen, Copenhagen, Denmark.*

Abstract

Structural variants are a major source of variation in the human genome. In particular, copy number variants (CNVs) have been associated with multiple diseases and syndromes. CNVs are typically defined as deletions or duplications spanning ~50kbp to ~10Mbp. Genotyping arrays still remain the most widely used platform to detect CNVs from, especially in large biobanks. However, CNV calling algorithms are prone to produce a high number of false positives (from 10% up to more than 50% depending on the level of sample quality), thus requiring analysts to manually "validate" calls. This has largely limited CNV research to the so-called recurrent loci. We have developed a machine learning algorithm based on the convolutional neural network architecture that is capable of automating the visual validation of CNVs across the whole human genome with an accuracy above 90%. Here we apply it to two large cohorts, UKB and deCODE, with the goal of providing an in depth characterization of genome wide CNVs. We describe how CNVs are distributed across the genome and how regions and genes are differentially permissive or intolerant to the presence of CNVs. Finally, we explore how the distribution of deletions and duplications differs depending on the genotype chip.

Introduction

Copy Number Variations (CNVs) are a class of structural variants (SVs) consisting of deletions and duplications, meaning a net loss or gain of a DNA segment. The study of structural variants is a variegated research field, in terms of technologies, sample sizes and research goals. From a historical view point, we typically separate two main study areas; on one side those studies that can be traced back to the discovery of chromosomal aneuploidies (such as studies of trisomy 21[1] and sex chromosome aneuploidies[2]) and other large abnormalities (such as Cri-du-chat[3] and Prader Willi/Angelman syndrome deletions[4] at 5p and 15q11-q13, respectively) and then to the recurrent sub-microscopic CNVs, typically flanked by non-allelic homologous repeat (NaHR) regions (such as at 1q21.1, 15q13.3 and 22q11.2)[5]; and on the other side a more recent but no less prolific study area that emerged from the evolution of DNA sequencing technologies and the discovery of smaller and potentially very complex, structural variants.[6,7]

In many aspects, most CNVs fall in between these two large fields and are often defined as SVs larger than 50-1000bp in studies involving sequencing data, and larger than 10-50kbp in studies involving microarray data. For these reasons, the terminology and, perhaps more importantly, the methodology are not very standardised across CNV studies. Most CNV studies that involve large study cohorts are based on SNP microarray data, and here we will refer to CNVs as deletions and duplications in the size range of ~50kb-10Mb, with smaller events being "other structural variants" (or indels if less than 50bp), and larger events being "large chromosomal rearrangements".

Detection (or "calling") of CNVs from SNP arrays is known to be afflicted by a high number of false positives.[8,9] We have shown previously that the most relevant factors behind spurious CNV calls from SNP array data, log-R-ratio (LRR), B-allele frequency (BAF) and the GC waviness factor (GCWF)[10], are not equally distributed across the genome.[11] Another common problem is the erroneous fragmentation of a true CNV into several smaller calls. Given the intrinsic limitations of CNV calling, large studies have generally followed one of two approaches; visual validation of the putative CNV carriers, or strong general filtering strategies.

The former approach is typically followed in a study focused on a limited number of preselected genomic loci (e.g., recurrent CNV loci or CNVs overlapping a specific gene), where a human analyst evaluates the LRR and BAF tracks underlying each CNV call. In such a study, each locus is the unit, and all carriers are treated the same if they have a CNV that fulfils given requirements (e.g. spanning more than half of the locus, covering a predefined critical region, or disrupting exons of a given gene).[12–14] Putative carriers are usually screened if they have any CNV call in the locus, even if much smaller than the locus itself, to account for fragmentation or partial calling of true CNVs. While very rigorous (if performed adequately), this approach is very time consuming, and essentially limited by the time required by the human analyst(s) to inspect the data behind each putative CNV call.

The second approach is typically applied in a study where CNV calls are filtered in some way (e.g. by QC metrics, minimum number of markers and/or minimum size, or algorithmic combinations[15]), multiple calling methods are used and then the intersection is considered, or a combination of both.[16–18] While this kind of study provides a more genome-wide approach to CNVs (akin to the one that revolutionised the discovery of common SNP variants associated to human conditions following the introduction of array-based SNP mapping) its major limitation is that it does not adequately control for false positive calls, in fact the false positives rate is usually not even computed, but errors are assumed to be randomly distributed across samples and between groups, e.g. cases and controls. When cases and controls are genotyped in exactly the same way (same chip, same facility etc), this assumption might hold. However, bias might still be introduced by the fact that false positive calls are not randomly distributed across the genome. In contrast, when cases and controls are genotyped separately this assumption fails entirely, since as we show here, different arrays will produce highly different CNV call sets, depending both on their SNP coverage (affecting the sensitivity to detect true CNVs) and array-specific profile of relevant QC measures (e.g., LRR, BAF and GCWF, affecting their underlying propensity to generate spurious calls).

We have developed an algorithm based on machine vision, capable of validating CNV calls from PennCNV[19], effectively reducing the number of false positives of CNVs as efficiently as when inspected by a human analyst, with minimal loss of overall sensitivity, thereby retaining the best qualities of both above described approaches; a genome-wide CNV assessment with the accuracy of a locus-specific visual evaluation of calls. In this study, we use this dataset of thoroughly validated genome wide CNV calls to characterize the frequency and distribution of CNVs across all autosomes in two large cohorts genotyped on different arrays. We explore how CNVs are distributed in the genome within and across the two populations. We showcase multiple methods to analyse CNVs, and present their strengths and limitations. Finally, we derive an expected distribution for CNVs based on simulation, and highlight regions and genes that are enriched or deprived of deletions and duplications, and how those genes and regions correlate with annotations such as gene constraint and cross-species conservation.

Methods

Samples, genotyping and CNV calling

This study is based on the UK Biobank (UKB hereafter)[20] and on the collection of Icelandic samples from deCODE genetics (deCODE hereafter). Genotyping has been described previously.[21,22] Briefly, most samples in UKB were genotyped within a relatively short timeframe using the UK Biobank Axiom array, while deCODE samples have been genotyped over a period spanning close to two decades across multiple versions of Illumina arrays. Unless explicitly otherwise stated, data from UKB and deCODE was processed separately but in the exact same manner at all steps. CNVs were called at deCODE[23] genetics using PennCNV[19] with standard setting. After CNV calling, samples were QC filtered following a standard procedure.[24] Samples were excluded if they had a LRR_SD (log R ratio standard deviation) > 0.35, BAF_drift (B allele frequency drift) > 0.01, $|GCWF|$ (absolute value of GC waviness factor) > 0.02, or more than 100 raw CNV calls. Samples were also screened for very large chromosomal abnormalities, defined as deletions or duplications spanning the majority of a chromosome or chromosomal arm. Visually validated carriers of such variants were excluded from the study. All CNV calls including at least 5 SNP markers were taken forward to the next step. A schematic representation of the two samples and the overall pipeline is shown in figure 1.

CNV processing

CNV calls were processed using the function `select_stich_cnvs()` from the R package QCtreeCNV[24] with the following values: `loci = hg38_chr_amrs`, `max_gap = 0.5`, `min_overlap = 0`, `minsnp = 10`. This will stitch CNV calls that are close and with the same copy number. Finally, CNVs with less than 10 markers or larger than 10 Mpb, after the stitching attempt, were excluded. The resulting call-sets were processed using our automated CNV validation R package CNValidatron[11], to produce the final set of validated CNVs. The software is described in detail elsewhere.[11] Briefly, we trained a convolutional neural network to perform visual validation of CNV calls using a simplified image. The model was trained on 13,000 human-labelled CNV calls and was validated in both UKB (2,800 examples) and deCODE (6,255 examples), resulting in an accuracy of 0.95 in deCODE and a precision of 0.92 in deCODE 0.95 in UKB.[11] All validation steps were performed using default values. Validated CNVs were defined as those with a prediction category 2 or 3 (true deletion and true duplication respectively) with a prediction probability higher or equal 0.75, a matching genotype from PennCNV and at least 10 markers from the filtered SNP maps (see next section). The prediction probability distribution is shown in supplementary figures 1 and 2.

SNP map filtering

The number and distribution of SNPs on the genotyping array play a central role when it comes to the validation of a CNV call. Moreover, simulations to determine the genome-wide CNV baseline, and thus the analysis of the genome-wide distribution of CNVs (see subsection below), are based on the global map of the selected SNPs. While virtually all individuals in UKB are genotyped on the same Affymetrix array, the deCODE sample is genotyped on multiple Illumina arrays from different generations. As shown in supplementary table 1, there is a large variation in the number of samples typed on each array subtype, with 12 subtypes used for less than 1000 samples each. Moreover, several arrays share the same "backbone" of SNPs and consequently have an IOU (intersection over the union) > 0.9 with each other (supplementary figure 3). For this reason, the 26 different chips can be grouped in a smaller set of chip families. Genotyping chips were grouped following the IOU clusters from supplementary figure 3 and less used chips (less than 750 samples) were discarded if they did not belong to a selected group. This resulted in a total of five chip families: Omni2.5 (HumanOmni2.5-4v1_H, HumanOmni2.5-8v1_A, HumanOmni2.5-4v1-Multi_H), Hap (HumanHap300_(v1.0.0), HumanHap300v2_A, HumanCNV370v1_C, HumanCNV370-Quadv3_C), Quad (Human610-Quadv1_B, Human660W-Quad_v1_H), OmniExpress (HumanOmniExpress-12v1_H, HumanOmniExpress-12v1-Multi_H, HumanOmniExpress-12v1-1_B, HumanOmniExpress-24v1-0_A, InfiniumOmniExpress-24v1-3_A1, InfiniumOmniExpress-24v1-2_A1, HumanOmniExpress-24v1-1_A, DECODE_OEx-8_A), 1M (Human1Mv1_C, Human1M-Duov3_B). Finally, both the UKB and the deCODE SNP maps were filtered to include only SNPs that are single nucleotide variants and with a minor allele frequency of at least 0.1 % in dbSNP 155. Compared to our previous approaches[24], we had to drop the "strictly biallelic" requirement for the filtered SNPs as it was excluding more markers than expected. This is most likely due to the fact that in hg38 most SNPs have at least one extremely rare variant reported and thus are not biallelic even if only two alleles are in fact observed in the population.

CNV grouping

CNVs can be extremely heterogeneous. We performed all analyses on different types of groups. These consist of: regular windows (250kbp and 50kbp), genes, and CNV regions (CNVRs). Regular windows and genes were used as bins in the function `binned_cnvs()` from the R package CNValidatron[11] while CNVR were computed using the function `cnvrs_iou()` from the same package. For all groupings, the result is a set of variants or markers that is smaller than the original CNV set. For all new variants each sample is assigned a genotype: 0 for normal copy number, 1 for deletions, and 2 for duplications. For exonic CNVs, we used the exons as bins but the genes as final variants, i.e., any sample with a CNV affecting at least one exon of a gene is assigned a genotype of 1 or 2 (for deletion and duplication, respectively) for that gene's designated variant. The CNVRs algorithm is described in detail elsewhere.[11] Very briefly, CNVs are first divided in groups of mutually overlapping segments. This is converted into a weighted `igraph`[25] object where each CNVs is a node and the two nodes are connected if the intersection over the union (IOU) value between them is above 0.75, with the IOU value being the strength of the connection. CNVRs are created using the community detection algorithm Leiden[26] on each network. After the process is completed a merging step is performed, where if very similar CNVRs (IOU > 0.75) are present in the results, they are combined. This final step ensures that the groups resulting from the network space remain meaningful in the genomic space.

Genomic annotations

We obtained information and genomic coordinates of genes and exons through the Ensembl (version 113) BioMart web interface (https://www.ensembl.org/biomart/martview/).[27,28] We focused our analysis on CNVs affecting at least one exon of a given gene. For each gene, we selected the transcript with a RefSeq match (indicated by the "RefSeq" field in the table downloaded from Ensembl 113). We excluded genes in the MHC (major histocompatibility) region (chr6:28,510,120 - 33,480,577, https://www.ncbi.nlm.nih.gov/grc/human/regions/MHC), which left a total of 13597 unique Ensembl identifiers. Finally, we excluded genes with inadequate SNP coverage as follows: We defined any 250 kbp window with at least 10 markers as having adequate SNP coverage, and then discarded any gene not fully covered by 250 kbp windows with adequate SNP coverage. This was done separately for UKB and deCODE; for deCODE we used the combined OmniExpress chip family map, as >70% of the deCODE sample had been genotyped on an array belonging to this chip family. For the combined analysis, we defined as good bins all 250kbp windows with at least 10 SNPs both in UKB and deCODE (OmniExpress chip). The final number of genes was thus 12,993 in UKB, 12,956 in deCODE, and 12,956 in the combined analysis.

We downloaded gene constraint values (loss-of-function observed/expected upper fraction (LOEUF) scores) from the gnomAD website (https://gnomad.broadinstitute.org/),[29,30] and obtained recombination rate scores and conservation scores from the "recomb. deCODE avg." track[31] and "phyloP100way" and "phastCons100way" tracks[32,33] in the UCSC genome browser (https://genome.ucsc.edu/),[34,35] respectively. For both recombination and conservation score we used the utility "bigWigAverageOverBed" to annotate our CNV gene/window-based markers using the appropriate track in "bigWig" format (e.g., to add the recombination score on the 50 kbp regular windows, "bigWigAverageOverBed deCODE_recomb_avg.bw 50kbp.bed 50kb_recomb.bed").

Genome-wide CNV baseline

To assess CNV enrichment across the genome, we first set a baseline, i.e. the expected number of CNV carriers per marker based on a random distribution. This was done using simulations. Briefly, CNVs in our datasets are grouped into groups using the CNVR algorithm as described above. Then each CNVR is shuffled in the genome; this is done following the SNP map and keeping the new length similar to the original one but not restricted to the chromosome of origin. Finally, CNVs are moved to the new location and the start and end positions of each are allowed to "wiggle" of +/- 10%. This process is repeated 1000 times. Finally, for each CNV marker set used (regular windows and genes), the baseline for a given marker is computed as the median N from the 1000 simulations. Supplementary figure 4 provides a schematic representation of the simulation algorithm. The simulation was run separately for UKB and the five different genotype chip families in deCODE.

To set a baseline value for genes we also used a second, orthogonal, simulation method, whereby we shuffled the genes rather than the CNVs. Genes were allowed to take a new start position anywhere between the first and last SNP marker of all chromosomal arms, followed by a corresponding updating of the start and end coordinates of all exons belonging to the gene. In the unlikely situation where a gene would end up "falling out" of adequately covered regions, it would be "pulled in" by its length value. This was also done 1000 times for each gene and the median N value across simulations used as the baseline value for the genomic enrichment tests. As the previous, also this simulation was run separately for UKB and the five different genotype chip families in deCODE.

CNVs enrichment scores

We computed the enrichment score for a given CNV marker (e.g. a gene) as the base two log of the number of observed carriers plus one, divided by the baseline value (i.e. the median from 1000 simulation) plus 1, and it is referred to as log2FC from now on, $log2FC = log_2\left(\left(N_{observed} + 1\right) \div \left(N_{expected} + 1\right)\right)$. For each marker we also computed a p-value using the fisher exact test function in R on the two by two table, in R syntax, `matrix(c($N_{observed}$, $N_{total} - N_{observed}$, $N_{expected}$, $N_{total} - N_{expected}$), ncols = 2, byrow = TRUE)`, where $N_{observed}$ represent the number of CNV in the marker, N_{total} the total number of CNVs in the dataset, and $N_{expected}$ the expected number of CNVs from the simulations. This was done separately for the two genotypes in the three marker sets (50 and 250 kbp regular windows, and genes) in the two sample subsets (UKB and deCODE) as well as in the combined sample (UKB+deCODE). In the analysis of gene CNV markers (i.e. grouping together all CNVs intersecting at least one exon of the same gene), we additionally subgrouped the CNVs into complete (CNVs fully encompassing all exons of the gene) and partial events.

Statistical analysis

All statistical analyses and plots were performed using R version 4 or above[36], and the R packages `data.table`[37], `ggplot`[38], `ggpubr`[39], `ggsignif`[40] and `patchwork`[41]. Plots were assembled using the free and open source software Inkscape (https://inkscape.org/about/).

To test for differences in the number of CNVs per sample in the two cohorts we use the Poisson model as follows. In R syntax, `glm(N ~ group, data, family = "poisson")`, where "N" is the number of CNV per sample and "group" is UKB or deCODE. We run three models for each filtering/validation step, one for deletions, one for duplication and one for any CNV.

To test for differences in the proportion of deletions and duplications between the two cohorts we used a Fisher exact test as follows. In R syntax `fisher.test(matrix(c($N_{dels}^{UKB}$, $N_{dups}^{UKB}$, $N_{dels}^{deCODE}$, $N_{dups}^{deCODE}$), nrow = 2))`. We run one test for each filtering/validation step.

All plotted correlations were obtained in the plotting call as Pearson coefficients using the function `stat_cor(cor.coef.name = "R", p.accuracy = 0.001)` from the R package `ggpubr`. All correlations not tied to a plot were computed using the base R function `cor()`.

All plotted significance bars and p-values between groups in the box plots were obtained in the plotting call with a Wilcoxon test using the function `geom_signif()` from the R package `ggsignif`.

To test for differences in the proportion of genes enriched or deprived of CNVs compared to the reference (protein coding) we used a Fisher exact test as follows. For a given subgroup (e.g. constrained genes) and a given genotype (deletions or duplications) the R syntax is `fisher.test(matrix(c($N_{deprived}^{subgroup}$, $N_{enriched}^{subgroup}$, $N_{deprived}^{all}$ - $N_{deprived}^{subgroup}$, $N_{enriched}^{all} - N_{enriched}^{subgroup}$)))`, where $N_{deprived}^{subgroup}$ and $N_{enriched}^{subgroup}$ are the number of genes significantly enriched or deprived in CNVs belonging to the specific subgroup, and $N_{deprived}^{all}$ and $N_{enriched}^{all}$ are the number of genes significantly enriched or deprived in CNVs in the reference group (protein coding).

Results

CNV calling and validation

After filtering and processing of PennCNV calls in the two datasets separately, 1,472,028 (3.2 per sample, 853,990 deletions (1.9) 618,038 duplications (1.3)) and 1,217,936 (6.9, per sample, 770,244 deletions (4.4), and 447,692 duplications (2.5)) CNV calls remained from 459,218 and 176,189 samples for UKB and deCODE respectively. We performed automated visual validation using the R package CNValidatron[42] on this dataset, which yielded 583,339 (1.3 per sample, 399,117 deletions (0.9), and 184,282 duplications (0.4)) and 488,264 (2.7 per sample, 256,286 deletions (1.5), and 231,987 duplications (1.2)) CNVs predicted to be true with a likelihood of >0.75 in UKB and deCODE, respectively. Figure 1 provides a schematic representation of the calling and validation pipeline.

The mean number of CNVs per sample was significantly lower in UKB compared to deCODE at all steps (raw, filtered, and validated) for both deletions and duplications separately, as well as any CNVs (GLM poisson test ORs between 0.15 and 0.27 for raw, 0.42 and 0.53 for filtered, 0.3 and 0.6 for validated CNVs, all p-values < 1e-15). The proportion of deletions and duplications also differed between the two datasets at all steps (Fisher test ORs 0.55, 0.80 and 1.96 for raw filtered and validated respectively, all p-values < 1e-15). Finally, also the proportion of validated CNVs (i.e., predicted true calls with a probability of at least 0.75) out of all evaluated CNV calls was different between the two datasets (Fisher test OR=0.98, p-value 3.3e-13).

Supplementary figures 5 and 6 show the distribution of the CNV length and their number of markers, respectively, in the raw and filtered sets of CNV calls in the two sample subsets (UKB and deCODE), while supplementary figures 7 and 8 show the same distributions in the validated and discarded CNVs after evaluation by the CNValidatron. Both in the UKB and deCODE, the length distribution for duplications is shifted to the right (towards larger events) compared to deletions, at all stages of the CNV calling, and this effect seems to be stronger in UKB. The distributions of the number of SNP markers per CNVs in UKB also seem to follow the same pattern, where duplications tend to include more markers than deletions do. This, however, does not seem to be the case in deCODE, where the distributions of the number of markers for deletions and duplication seem to follow each other quite closely at all steps.

Distribution of CNVs in the human genome

To enable the analysis of such a varied class of structural variation, we defined multiple CNV markers to base most of our analyses on. We used regular windows (50 and 250 kbp), genes and CNV regions (CNVRs) as markers sets. We used CNVs markers to analyse the genomic distribution in the genome, to detect differences in two populations, and to explore the positive or negative enrichment of CNVs across the genome.

CNVs are not distributed equally in terms of CNVs per chromosome and CNVs per Mbp across chromosomes (supplementary figure 9). Moreover, telomeric and centromeric regions have been previously reported to be enriched in CNVs compared to the rest of the genome.[43] We also observe the same pattern, supplementary figures 10 and 11, however, it does not appear to be constant neither across CNV genotype (deletions/duplications) nor across sample

(UKB/deCODE). In general CNVs appear to occur at very disparate frequencies at different genomic locations. The distribution for the joint sample is shown in figure 2 while supplementary figures 12 and 13 show UKB and deCODE separately. Often the pattern of deletions and duplications follow each other, but not always, and rarely to the same magnitude. Across all chromosomes, we observe a strong pattern of CNVs accumulating in what we define as “hotspots”, i.e. genomic locations where CNVs accumulate, while not necessarily being all of the same kind.

CNVRs can effectively reduce the number of variants in a CNV set

We previously described an algorithm to effectively group CNVs into highly internally homogeneous groups, referred as CNV regions (CNVRs).[42] Here we highlight how CNVRs can be used to reduce the complexity of a CNV dataset while retaining a resolution very similar to the individual CNV calls. Of note, CNVRs were computed and analysed for all CNVs together (deletions and duplications) and this is in contrast to all other markers in the study. In total we detected 49,130 CNVRs in UKB and 19,661 in deCODE. Given the sensitive nature of human research, many studies require a minimum number of individuals per variant in order to be studied. In our study, this minimum is set at 5 carriers for each dataset. Only 8,780 (18%) and 4,400 (22%) of CNVRs had at least 5 CNVs assigned to them, in UKB and deCODE respectively. However, the “ultra-rare” groups that comprise the majority of the CNVRs accounted for a minor portion of the total CNVs, 62,534 in UKB (11%) and 24,292 in deCODE (5%). This means that a small set of groups (CNVRs with at least 5 carriers) can be effectively used to describe the vast majority of CNVs in the two samples, while retaining approximate information of their actual size and boundaries, which would be lost when using fixed markers such as the regular windows. The reduction in number of variants is 60 and 105 fold in UKB and deCODE, respectively. As shown in supplementary figure 14, most CNVRs are rare, even when excluding those with less than 5 carriers. Despite this, a sizable portion has a frequency above 1% (128 in UKB and 446 in deCODE) and these account for the majority of the total number of CNVs in the sets, 322,069 (55%) and 365,175 (75%) respectively. Finally, CNVR carrier frequency is not correlated to CNVR length in either sample (supplementary figure 15, R=-0.04 in UKB and R=0.005 in deCODE).

Investigating the stability of CNVs frequency across the two populations

The carrier frequency of CNV markers across the two samples are moderately correlated when using either 50 or 250 kbp fixed windows (supplementary figure 16, average correlation 0.47, p-value < 0.001) however, a sizable proportion of markers seems to be unique to one population. The correlation does not change substantially between 50 and 250 kbp windows, which suggests that the differences we observe in CNV carrier frequency between UKB and deCODE at many loci are more likely due to population-specific differences and/or site-specific CNV detection differences between arrays, than to array-related differences in resolution of the boundaries of detected CNVs (supplementary figure 16, a vs b and c vs d), since a larger unit of observation should to a large extent mask such differences.

To more easily compare markers across the two cohorts we defined some frequency groups: 1, markers with CNVs detected in less than 1 in 40,000 samples (corresponding to less than 5 carriers in deCODE); 2, between 1/40,000 and 1/4,000; 3, between 1/4,000 and 1/400; 4, between 1/400 and 1/40; 5, more than 1/40 (corresponding to 2.5% prevalence of the CNV). As reported in supplementary table 2 and 3, there are consistently more CNV markers in frequency group 1 (and, conversely, less in group 2) in deCODE than UKB, while from group 3 onwards (frequency $> 1/4,000$) the numbers seem to stabilize across the two samples.

Given the limitations in comparing the two samples due to the differences in genotyping array used, as well as an alternative to using frequency, we computed CNV enrichments scores for each marker, as log2 fold change (log2FC) values, comparing the number of CNVs in each marker against the random expectation from 1000 simulations. As shown in supplementary figure 17, the correlation across the two populations is very similar to the one using simple frequency, however, using log2FC we are able to gain resolution in the lower end of the frequency spectrum, where markers tend to often fall below the minimum 5 carriers threshold.

Finally, we also use CNVRs to compare the two samples. To do so, we combined the CNVRs computed in the two cohorts into a single set. A shared region was defined as a pair of CNVR in the two sets with an IOU value of at least 0.9, while the rest were considered regions unique to one cohort. Interestingly, only 432 CNVRs were found to be shared across the two cohorts and having at least 5 carriers in both. These common regions accounted for 178,066 CNVs in UKB and 138,417 in deCODE and their frequency in the two cohorts shows a positive correlation of 0.52 (p-value < 0.001, supplementary figure 18). However, it can vary substantially, with the log2 frequency ratio (log in base two of the ratio between the frequency in deCODE and the frequency in UKB, log2FR) going from -5 to above 5 (supplementary figure 19). This set of shared regions more often has a higher carrier frequency in deCODE than in UKB (log2FR distribution mean=0.73 and standard deviation=2.52).

Genes are differentially affected by CNVs across the genome and the two populations

We analysed how CNVs affect human genes and focused our analysis on well characterised transcripts, defined as those with a reported RefSeq ID in Ensembl 114. This essentially restricted the analysis to only protein coding genes. First, we applied the same frequency grouping described above on the genes as well. As shown in supplementary table 4 the number of genes in each frequency group is quite stable across the two cohorts. However, the proportion of genes that are common between the two cohorts in each group is less than 50% in all groups except the first (for events rarer than 1/40,000), both for deletions and duplications. As expected by the difference in sample size, more genes are affected exactly zero times by a deletion or a duplication in deCODE than in UKB (7,989 vs 4,334 for deletions and 6,607 vs 2,296 for duplications). To further investigate this, we extracted genes that are never affected by either a deletion or a duplication in one cohort, and checked to which frequency group they belong in the other. As reported in supplementary table 5 and 6, most of these belong to the first group (which include zeros) also in the second sample, however, a sizable portion was found having higher frequencies, in groups 2 and 3 (thus in frequencies up to 1/400). Considering that the two samples are genotyped on different platforms, we investigated if genes were differentially covered by SNPs markers on average. Supplementary figure 20, confirms that overall genes are equally covered in all frequency groups and across the two cohorts. This is also true for the genes affected zero times in one sample but not in the other, supplementary figure 21.

CNVs differentially affect genes based on their tolerance to loss-of function variants

We explored how deletions and duplications affect different groups of genes. Having focused our analysis on protein coding genes and exons overlapping CNVs, we defined groups of genes based on tolerance to loss of function variants (LOEUF score)[29,30] and conservation across species (phastCons and phyloP scores).[32,33] We first explored how these scores are distributed across the frequency groups, with the expectation that genes where we observe fewer CNVs would be less tolerant to variation, supplementary figure 22. Overall, the results follow our expectation both in UKB and deCODE, genes in frequency group 1 have lower values of LOEUF score and higher values of phastCons and phyloP scores, consistent with being less tolerant to variation. As larger genes will - all other things equal - be overlapped by more CNVs than smaller ones, we used two separate orthogonal simulation approaches to set a size-adjusted background value (or random expectation) for each gene, i.e. the median number of deletions or duplications that a given gene is affected by, across 1000 iterations. Comparing this to the actual number from our results we can compute a “genomic enrichment score” in terms of log2FC, essentially describing if a gene is overlapped more or less often by a CNV than randomly expected. The log2FC values obtained from the two different simulations are highly correlated for both deletions and duplication in the two samples ($R > 0.9$, p-value < 0.001, supplementary figure 23). The log2FC value for a gene partially follows its CNV frequency by definition, however, it is worth noting that we are capable of detecting genes that are “protected” but still affected by CNVs in moderate frequencies (group 2), supplementary figure 24. As shown in supplementary figure 25 the log2FC across the two samples are correlated but not as strongly as we would have expected if the two populations were truly comparable (0.5 on average between the two simulations for deletions and 0.49 for duplications). The correlation coefficient grows slightly when restricting to only those genes with at least 5 CNVs, to 0.61 for deletions and 0.5 for duplication on average across the two simulations.

The relationship between gene constraint and CNV genomic enrichment score

Having observed a pattern between frequency groups and genes constrain and conservation scores, we plotted the log2FC distribution (with a p-value threshold of 0.01) in different gene groups and tested for differences, using the protein coding genes as reference. As shown in Figure 3, in UKB (panel a, first row) a similar number of protein coding genes have a negative log2FC value (meaning they are affected by a CNVs less often than expected) for deletions and duplications (755 and 737), while the difference is higher for positive log2FC values (915 genes for deletions and 1247 for duplications). In contrast to UKB, in deCODE (figure 3 panel b, first row) we do not observe many genes with a negative log2FC at the p-value threshold we used. We reasoned this could be due to (a) the lower sample size, (b) the fact that the deCODE sample is a more population representative sample (and thus potentially deleterious variants are observed more often than in UKB, a sample healthier than the general population), (c) true population differences, or (d) a combination of those reasons. A random subsampling of the UKB sample to the same size as the deCODE sample suggests that the difference may be in large part driven by the first reason, i.e. a lack of power to identify genes significantly deprived of CNVs in the deCODE sample (supplementary figure 26). Moreover, moving the p-value threshold to 0.1, does not drastically increase the number of genes with negative log2FC (supplementary figure 27, from 18 and 36 to to 26 and 48 for deletions and duplications respectively). The log2FC values for deletions and duplications for a given gene are moderately correlated (supplementary figure 28) and the effect is more clear in UKB, likely due to power.

We define four gene subgroups: constrained and not constrained (gnomAD LOEUF score lower than 0.6 and higher or equal to 0.6 respectively), positive and negative mean phyloP score (mean phyloP across the whole gene, excluding zeros). We then tested if the proportion of genes with a significant log2FC value changes across the subgroups (text in each panel of figure 3) with an ORs above one indicating the weight of the distribution moved towards more genes being deprived of CNVs and ORs below one indicating that the distribution moved towards more genes being enriched in CNVs. Starting from UKB (figure 3, panel a), genes than are not constrained are more likely to be enriched in CNVs (both deletions and duplications) than deprived, compared to the whole protein coding set (OR=0.08; p-value=2.8e-98 for deletions and OR=0.61; p-value=2.8e-5 for duplications). In contrast, constrained genes are more likely to be deprived in both deletions and duplications than enriched (OR=12.48; p-value=2.8e-98 for deletions and OR=1.63; p-value=2.8e-5 for duplications). In both groups (constrained and non constrained genes) the effect is substantially stronger for deletions than duplications, suggesting genes are in general more sensitive to being deleted than duplicated. A similar effect is observed in more and less conserved genes (positive and negative mean phyloP value), however the effect on deletions is less pronounced. In deCODE (figure 3, panel b), despite the lower power for negative log2FC values compared to UKB, we observe a similar pattern (all ORs on the same side compared to UKB). Finally, when analysing all samples together (supplementary figure 29), the results seem to follow the ones for UKB.

CNVs polymorphisms in human genes

As reported in the previous section, a small group of genes are affected by a deletion or a duplication (considered separately) in 1% or more of the population (the cutoff usually used for SNPs). In total, we find 34 such genes in either UKB or deCODE (supplementary table 7). Of those, 8 are shared between the two cohorts, while 6 and 20 are exclusive to UKB and deCODE respectively. Out of the 8 shared ones, only two were deletions. After manual inspection of the genomic location and of the raw data trends of a random set of samples, we were able to group the 34 genes in 17 distinct loci (meaning in some cases one CNV was overlapping multiple genes). All but one (*OR4K5*) were deemed true CNV polymorphism loci. Two loci were rather heterogeneous: duplication covering the genes *SLC2A3* and *SLC2A14*, and deletions covering the genes *PSG1*, *PSG2*, *PSG4*, *PSG5*, *PSG9* and *PSG11*. The rest of the loci were characterised by fairly stable CNV boundaries. Of note, in one locus (the genes *KIR2DL1* and *KIR2DL4*) both deletions and duplication were in high frequency. Differential SNP coverage both in the genes themself (supplementary table 7) and in the locus (not shown) partially explains the differences in frequencies between the two cohorts, especially the more extreme ones. The more stark example is the gene *RHD*, responsible for the blood type Rh-negative (D-negative), where deletion spanning the entire gene is already reported with a high frequency in the literature.[44] This is indeed found in UKB although at a much lower frequency than expected, however in deCODE there are no markers covering the gene, and thus it is not detected. Similarly, on chromosome 1 we observe duplications affecting a cluster of amylase-encoding genes in 1%-2.5% of samples in deCODE, where the different frequency across the genes is explained by differential coverage across Illumina array families, while no carriers are observed in UKB, seemingly because of very low SNP coverage in the locus.

Discussion

In this study we have performed a genome-wide assessment of medium-to-large CNVs using SNP array data from more than 600,000 human samples from the UKB and deCODE. In doing so, we have sought to combine a genome-wide focus at large scale with a highly controlled false positive rate, by automated visual validation using a computer vision approach which effectively removes ~90% of false positive PennCNV calls but retains nearly all true calls. To the best of our knowledge, this is the first study that successfully combines such visual validation of each CNV call with a large-scale genome-wide approach.

Also, we have explored different ways to account for the extreme heterogeneity of CNVs, owing both to real differences in boundaries as well as inaccuracy in determining boundaries (by the CNV calling algorithm) and/or different SNP coverage across array types, by grouping deletions and duplications by overall location (regular windows of 50kb and 250kb), similarity (grouping highly similar CNV calls into CNV regions), and functionality (grouping together CNVs that disrupt the same gene). Applying these different classifications of CNVs, we then sought to describe the main trends and characteristics of the genome-wide distribution of both deletions and duplications, with respect to both chromosomal regions (e.g. telomeric and centromeric regions) and genome functionality (e.g. protein-coding genes and other conserved regions).

We replicate previous findings such as an accumulation of CNVs towards the telomeres and the centromeres. Moreover, we detect a clear tendency of CNVs to concentrate in specific genomic loci (high frequency windows) and clusters (high frequency CNVRs). Due to this, while almost the entire genome is covered by at least one CNV in our combined dataset, the majority of CNVs we detect belong to a CNV "hotspots". Somewhat unexpectedly, most of these CNV hotspots were not shared between the UKB and deCODE cohorts. Upon inspection of the most disparate of these (i.e., in terms of difference in carrier frequency between the two cohorts) we found that in most cases the SNP coverage differed markedly between the different microarrays used in this study.

To investigate how deletions and duplications are distributed across gene-encoding and other conserved regions of the genome, we had to first derive a baseline, or null distribution. This was by no means a trivial task as we had to account for various factors, such as; (a) that CNVs with shared breakpoints more often than not are identical-by-descent (IBD) and therefore do not represent independent mutational events, (b) that the CNVs detected in our dataset are bound by the unequal SNP coverage on the microarrays on which samples were genotyped, and (c) that genes of different sizes and shape have differing likelihoods of being overlapped by a CNV by chance. We solved this by applying two approaches; (1) by assigning each CNV to a CNVR and then moving the CNVRs randomly within the SNP map of the corresponding microarray, and (2) by moving each gene in a corresponding manner within the range of the SNP map.

Through these approaches we derived a CNV enrichment score (log2FC) describing whether CNVs were observed more or less often overlapping a gene than expected by chance. To our reassurement, the enrichment scores correlated very well between the two approaches (i.e. obtained shuffling the CNVs or the genes) and showed the same trend in both UKB and deCODE, that constrained genes for which the number of CNV carriers deviated significantly from the expected, were more likely to be deprived of carriers (i.e., having a negative log2FC), than the corresponding group of non-constrained genes. This was true for both gene-overlapping deletions and duplications.

Taken together, our results show that even when the false positive rate of array-based CNV calls can be adequately controlled on a genome-wide basis, the limited sensitivity in CNV detection imposed by the SNP coverage of the genotyping array still presents an important limitation in the capacity to derive unbiased genome-wide inferences in high resolution. Here, for example, we identify the same overall trends in two large datasets typed on different types of microarrays, but even if we can on this basis conclude that CNVs aggregate around telomeres and centromeres, and that CNVs are less often found within constrained than non-constrained genes, we can not unambiguously identify which are the most important genes, or compare CNV frequencies at each specific locus between the two cohorts.

Bibliography

1. Lejeune, J., Gauthier, M. & Turpin, R. [Human chromosomes in tissue cultures]. *Comptes Rendus Hebd. Seances Acad. Sci.* **248**, 602–603 (1959).

2. Robinson, A., Goad, W. B., Puck, T. T. & Harris, J. S. Studies on chromosomal nondisjunction in man. 3. *Am. J. Hum. Genet.* **21**, 466–485 (1969).

3. Overhauser, J., Beaudet, A. L. & Wasmuth, J. J. A fine structure physical map of the short arm of chromosome 5. *Am. J. Hum. Genet.* **39**, 562–572 (1986).

4. Butler, M. G. & Greenstein, M. A. Molecular cytogenetics of Prader-Willi and Angelman syndromes. *Lancet Lond. Engl.* **338**, 1276 (1991).

5. Stefansson, H. *et al.* Large recurrent microdeletions associated with schizophrenia. *Nature* **455**, 232–236 (2008).

6. Collins, R. L. *et al.* A structural variation reference for medical and population genetics. *Nature* **581**, 444–451 (2020).

7. Schmitz, D. *et al.* Copy number variations and their effect on the plasma proteome. *Genetics* **225**, iyad179 (2023).

8. Winchester, L., Yau, C. & Ragoussis, J. Comparing CNV detection methods for SNP arrays. *Brief. Funct. Genomic. Proteomic.* **8**, 353–366 (2009).

9. Seiser, E. L. & Innocenti, F. Hidden Markov Model-Based CNV Detection Algorithms for Illumina Genotyping Microarrays. *Cancer Inform.* **13s7**, CIN.S16345 (2014).

10. Diskin, S. J. *et al.* Adjustment of genomic waves in signal intensities from whole-genome SNP genotyping platforms. *Nucleic Acids Res.* **36**, e126–e126 (2008).

11. Montalbano, S. *et al.* CNValidatron, automated validation of CNV calls using computer vision. 2024.09.09.612035 Preprint at https://doi.org/10.1101/2024.09.09.612035 (2024).

12. Crawford, K. *et al.* Medical consequences of pathogenic CNVs in adults: analysis of the UK Biobank. *J. Med. Genet.* **56**, 131–138 (2019).

13. Vaez, M. *et al.* Population-Based Risk of Psychiatric Disorders Associated With Recurrent Copy Number Variants. *JAMA Psychiatry* (2024) doi:10.1001/jamapsychiatry.2024.1453.

14. Montalbano, S. *et al.* Analysis of exonic deletions in a large population study provides novel insights into NRXN1 pathology. *Npj Genomic Med.* **9**, 1–10 (2024).

15. Macé, A. *et al.* New quality measure for SNP array based CNV detection. *Bioinformatics* **32**, 3298–3305 (2016).

16. Huguet, G. *et al.* Effects of gene dosage on cognitive ability: A function-based association study across brain and non-brain processes. *Cell Genomics* **4**, 100721 (2024).

17. Stylianou, C. E. *et al.* Germline copy number variants and endometrial cancer risk. *Hum. Genet.* **143**, 1481–1498 (2024).

18. Halvorsen, M. W. *et al.* A burden of rare copy number variants in obsessive-compulsive disorder. *Mol. Psychiatry* (2024) doi:10.1038/s41380-024-02763-7.

19. Wang, K. *et al.* PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res.* **17**, 1665–1674 (2007).

20. Sudlow, C. *et al.* UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Med.* **12**, e1001779 (2015).

21. Bycroft, C. *et al.* The UK Biobank resource with deep phenotyping and genomic data. *Nature* **562**, 203–209 (2018).

22. Gudbjartsson, D. F. *et al.* Sequence variants from whole genome sequencing a large group of Icelanders. *Sci. Data* **2**, 150011 (2015).

23. Gudmundsson, O. O. *et al.* Attention-deficit hyperactivity disorder shares copy number variant risk with schizophrenia and autism spectrum disorder. *Transl. Psychiatry* **9**, 258 (2019).

24. Montalbano, S. *et al.* Accurate and Effective Detection of Recurrent Copy Number Variants in Large SNP Genotype Datasets. *Curr. Protoc.* **2**, e621 (2022).

25. Csárdi, G. *et al.* igraph for R: R interface of the igraph library for graph theory and network analysis. Zenodo https://doi.org/10.5281/zenodo.10681749 (2024).

26. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).

27. Cunningham, F. *et al.* Ensembl 2022. *Nucleic Acids Res.* **50**, D988–D995 (2022).

28. Haider, S. *et al.* BioMart Central Portal—unified access to biological data. *Nucleic Acids Res.* **37**, W23–W27 (2009).

29. Karczewski, K. J. *et al. Variation across 141,456 Human Exomes and Genomes Reveals the Spectrum of Loss-of-Function Intolerance across Human Protein-Coding Genes*. http://biorxiv.org/lookup/doi/10.1101/531210 (2019) doi:10.1101/531210.

30. Chen, S. *et al.* A genomic mutational constraint map using variation in 76,156 human genomes. *Nature* **625**, 92–100 (2024).

31. Halldorsson, B. V. *et al.* Characterizing mutagenic effects of recombination through a sequence-level genetic map. *Science* **363**, eaau1043 (2019).

32. Pollard, K. S., Hubisz, M. J., Rosenbloom, K. R. & Siepel, A. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.* **20**, 110–121 (2010).

33. Siepel, A. *et al.* Evolutionarily conserved elements in vertebrate, insect, worm, and yeast

genomes. *Genome Res.* **15**, 1034–1050 (2005).

34. Karolchik, D. *et al.* The UCSC Table Browser data retrieval tool. *Nucleic Acids Res.* **32**, D493-496 (2004).

35. Fujita, P. A. *et al.* The UCSC Genome Browser database: update 2011. *Nucleic Acids Res.* **39**, D876–D882 (2011).

36. R Core Team. *R: A Language and Environment for Statistical Computing*. (R Foundation for Statistical Computing, Vienna, Austria, 2024).

37. Barrett, T. *et al. Data.Table: Extension of `data.Frame`*. (2024).

38. Wickham, H. *Ggplot2: Elegant Graphics for Data Analysis*. (Springer-Verlag New York, 2016).

39. Kassambara, A. *Ggpubr: 'ggplot2' Based Publication Ready Plots*. (2023).

40. Constantin, A.-E. & Patil, I. ggsignif: R Package for Displaying Significance Brackets for 'ggplot2'. *PsyArxiv* (2021) doi:10.31234/osf.io/7awm6.

41. Pedersen, T. L. *Patchwork: The Composer of Plots*. (2024).

42. Montalbano, S. *et al.* CNValidatron, automated validation of CNV calls using computer vision. 2024.09.09.612035 Preprint at https://doi.org/10.1101/2024.09.09.612035 (2024).

43. Zarrei, M., MacDonald, J. R., Merico, D. & Scherer, S. W. A copy number variation map of the human genome. *Nat. Rev. Genet.* **16**, 172–183 (2015).

44. Wagner, F. F. & Flegel, W. A. *RHD* gene deletion occurred in the *Rhesus box*. *Blood* **95**, 3662–3668 (2000).

Figures

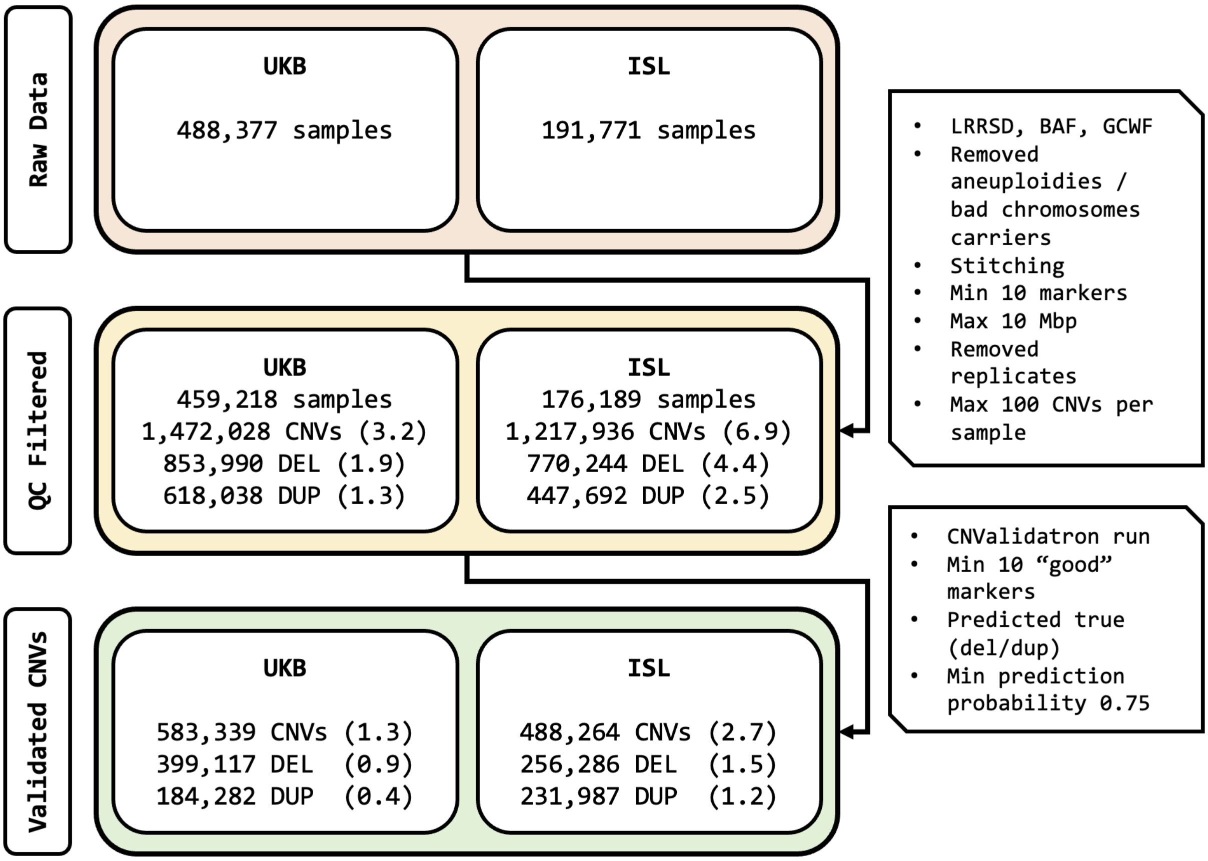


Figure 1. Schematic representation of the calling, filtering, and validation pipeline. The boxes on the right shows a summary of the main filtering and processing steps for each stage (from raw data to filtered and from filtered to validated CNVs). The numbers in parenthesis are the mean CNVs per sample.

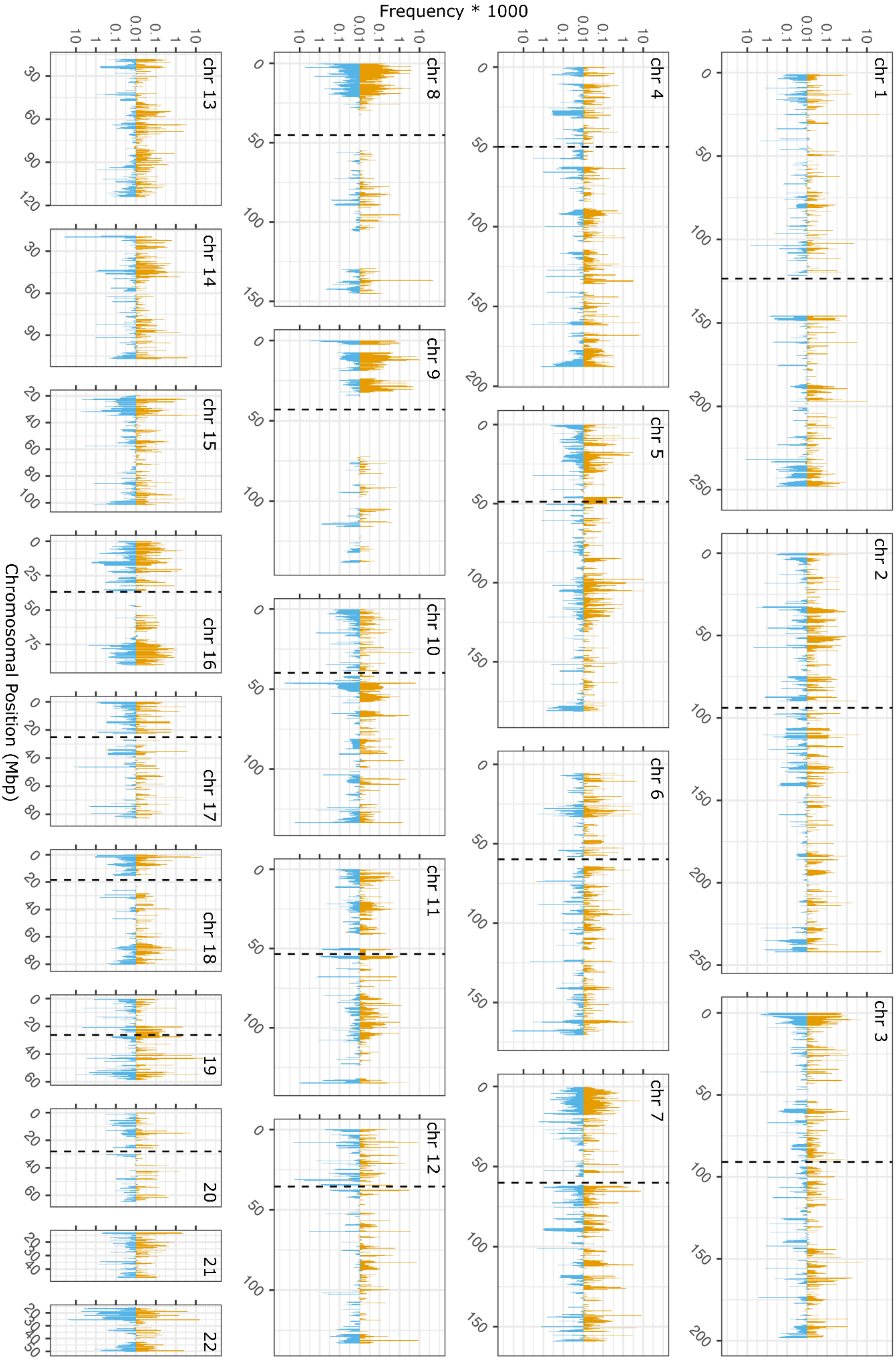

Frequency * 1000
10
1
0.1
0.01
0.1
1
10
chr 1
chr 2
chr 3
chr 4
chr 5
chr 6
chr 7
chr 8
chr 9
chr 10
chr 11
chr 12
chr 13
chr 14
chr 15
chr 16
chr 17
chr 18
19
20
21
22
0
50
100
150
200
250
30
60
90
120
20
40
60
80
25
75
Chromosomal Position (Mbp)

Figure 2. Genomic distribution of CNVs in UKB and deCODE considered as a joint sample. Each bar represents the frequency of either deletions (in yellow) and duplications (in blue) in a 250 kbp bin. Note that the y axis scale starts at 0.01, thus all bins with a CNV frequency for the given genotype below 1/100,000 are effectively shown as zeros.

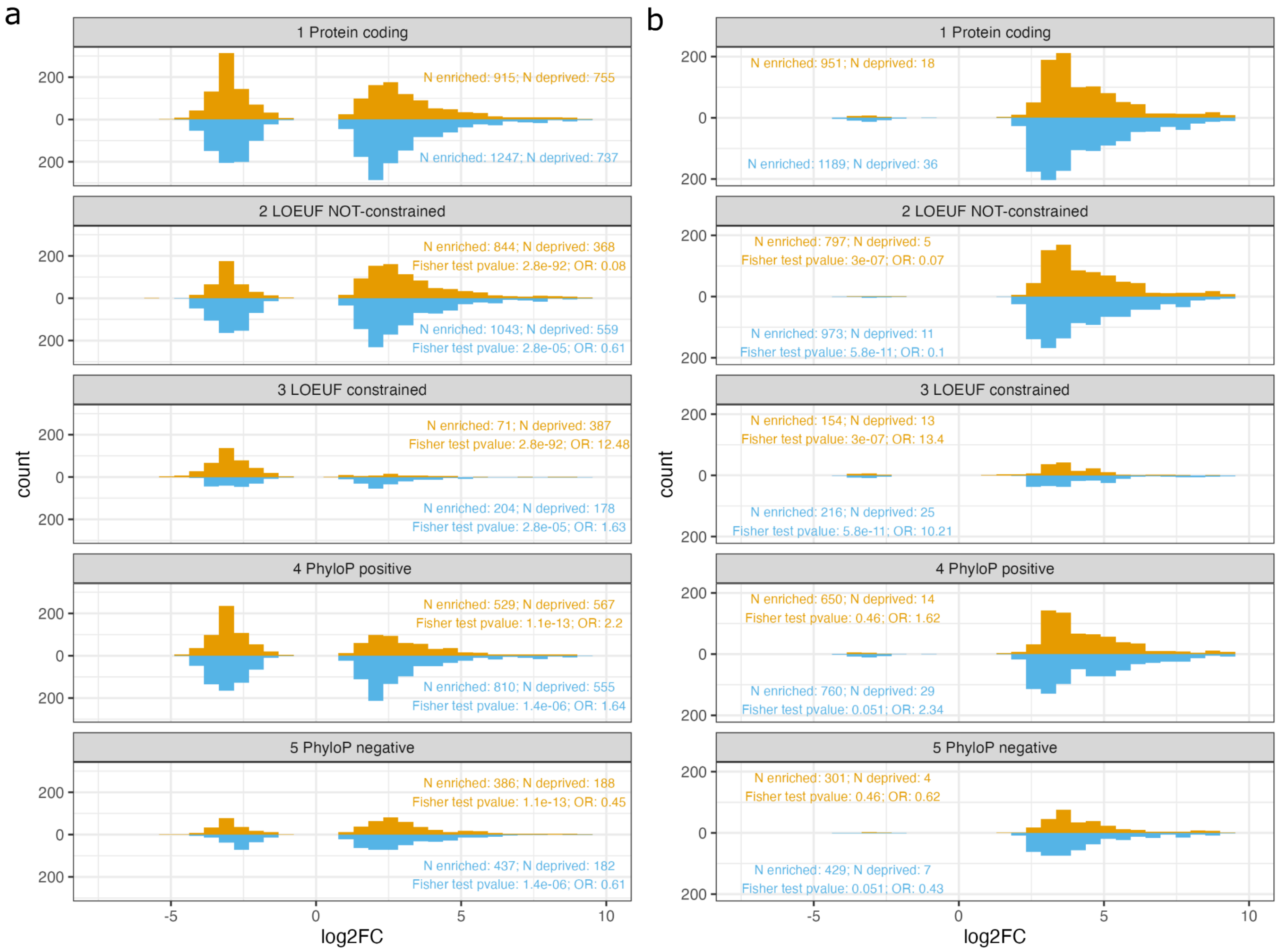
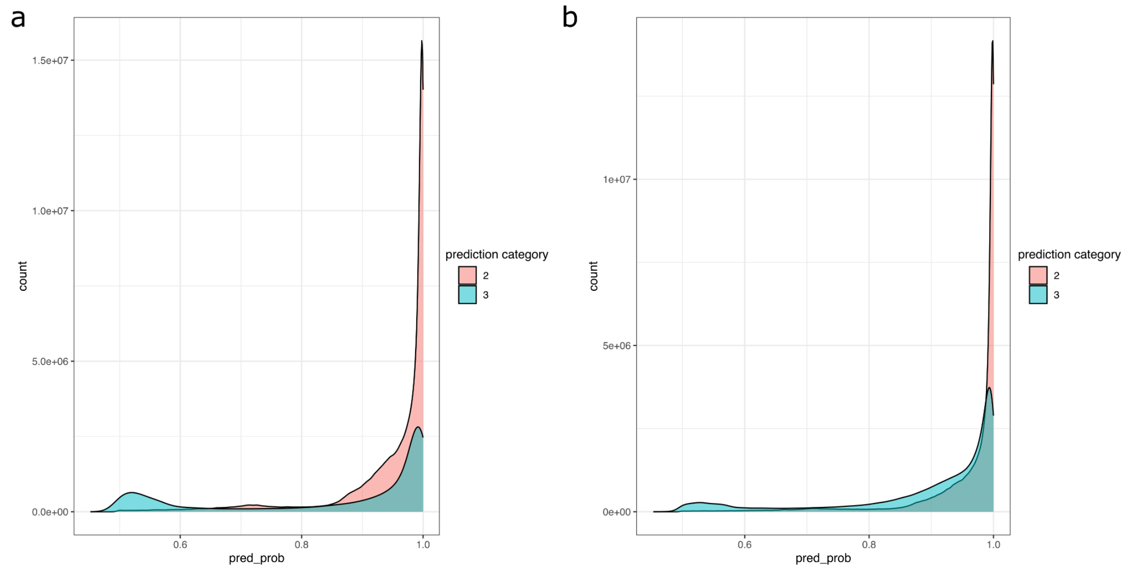
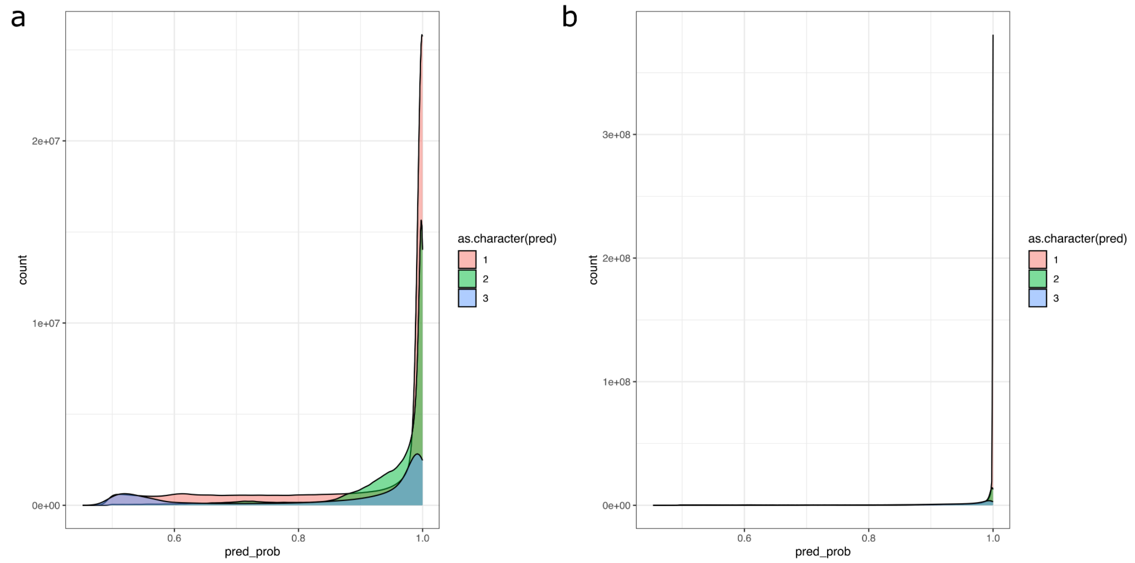


Figure 3. Histograms for the log2FC value for each gene at the 0.01 p-value threshold (meaning only significant genes are shown) in UKB (a) and deCODE (b). The log2FC shown are obtained using the CNVRs simulation to obtain the $N_{expected}$. Values for the deletions are shown in yellow and duplications in blue. The set of protein coding genes (first row in both panels) is used as reference, and a fisher test is used to detect differences in the proportion of significantly enriched/deprived genes in different subgroups.

Supplementary material



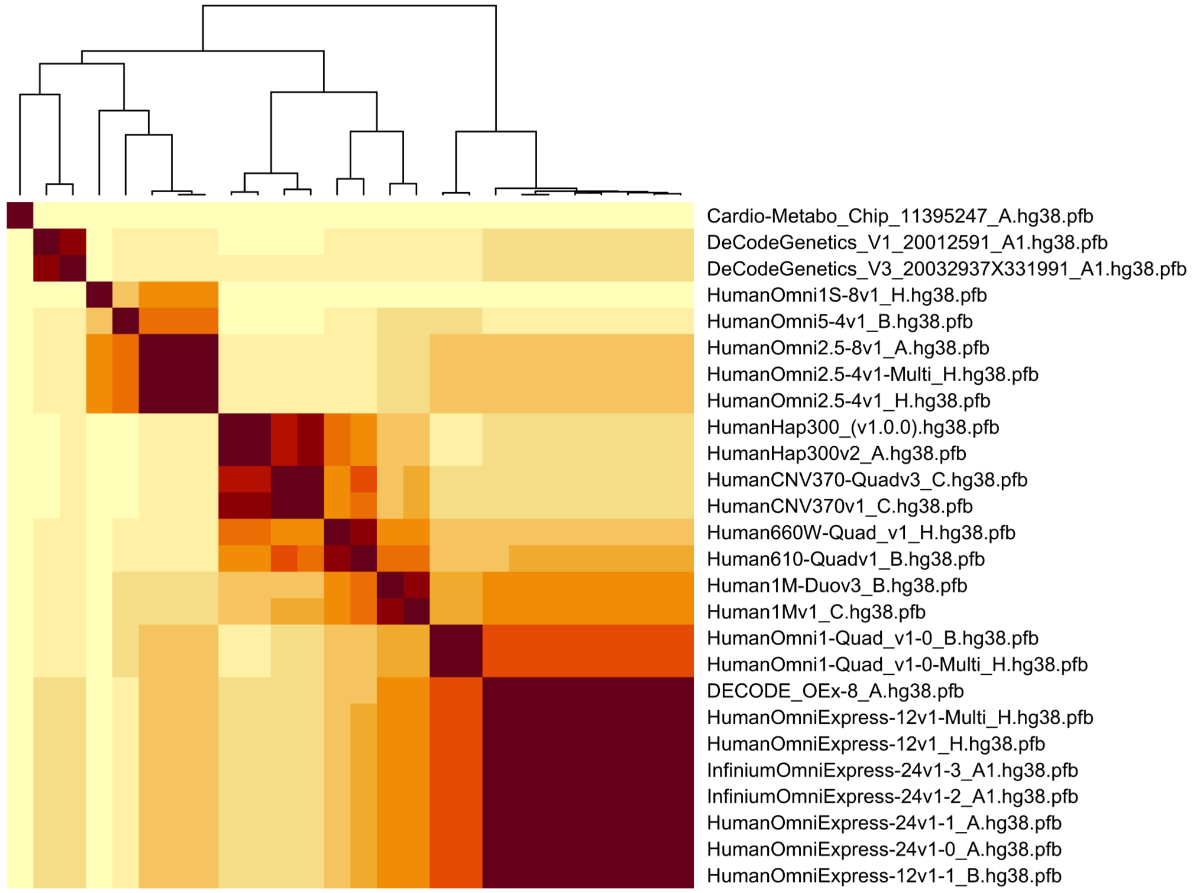
Supplementary figure 1. CNValidatron output prediction probability distribution in a) UKB and b) deCODE. Prediction categories are 1: False call, 2: True deletion, and 3: True duplication. The category “false” is omitted for readability.



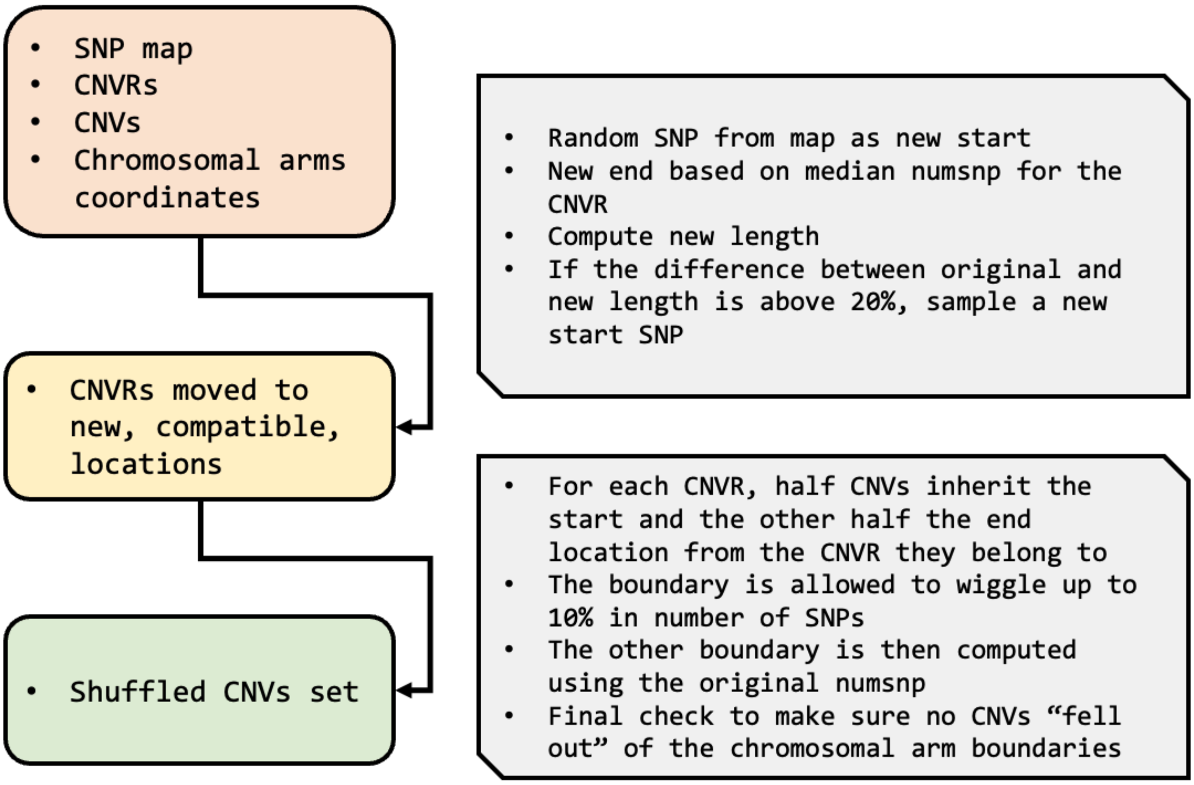
Supplementary figure 2. CNValidatron output prediction probability distribution in a) UKB and b) deCODE. Prediction categories are 1: False call, 2: True deletion, and 3: True duplication.

Chip name	N samples
HumanHap300_(v1.0.0)	16460
Human1Mv1_C	757
HumanHap300v2_A	7005
HumanCNV370v1_C	14537
HumanCNV370-Quadv3_C	305
Human610-Quadv1_B	671
Human1M-Duov3_B	573
HumanOmni1-Quad_v1-0_B	11237
Cardio-Metabo_Chip_11395247_A	998
DECODE_OEx-8_A	13019
HumanOmniExpress-12v1_H	32565
Human660W-Quad_v1_H	31
HumanOmni1-Quad_v1-0-Multi_H	76
HumanOmni2.5-4v1_H	2727
HumanOmniExpress-12v1-Multi_H	2907
HumanOmniExpress-12v1-1_B	20302
HumanOmni2.5-8v1_A	4173
HumanOmni2.5-4v1-Multi_H	427
HumanOmni5-4v1_B	708
HumanOmni1S-8v1_H	235
HumanOmniExpress-24v1-0_A	35656
InfiniumOmniExpress-24v1-3_A1	26109
InfiniumOmniExpress-24v1-2_A1	6774
HumanOmniExpress-24v1-1_A	1365
DeCodeGenetics_V1_20012591_A1	54
DeCodeGenetics_V3_20032937X331991_A1	128

Supplementary table 1. Samples count per genotyping chip in deCODE raw data.



Supplementary figure 3. IOU heatmap for all Illumina genotyping chips SNP maps used in deCODE. Higher values correspond to greater proportions of shared markers between two different arrays.



- SNP map
- CNVRs
- CNVs
- Chromosomal arms coordinates

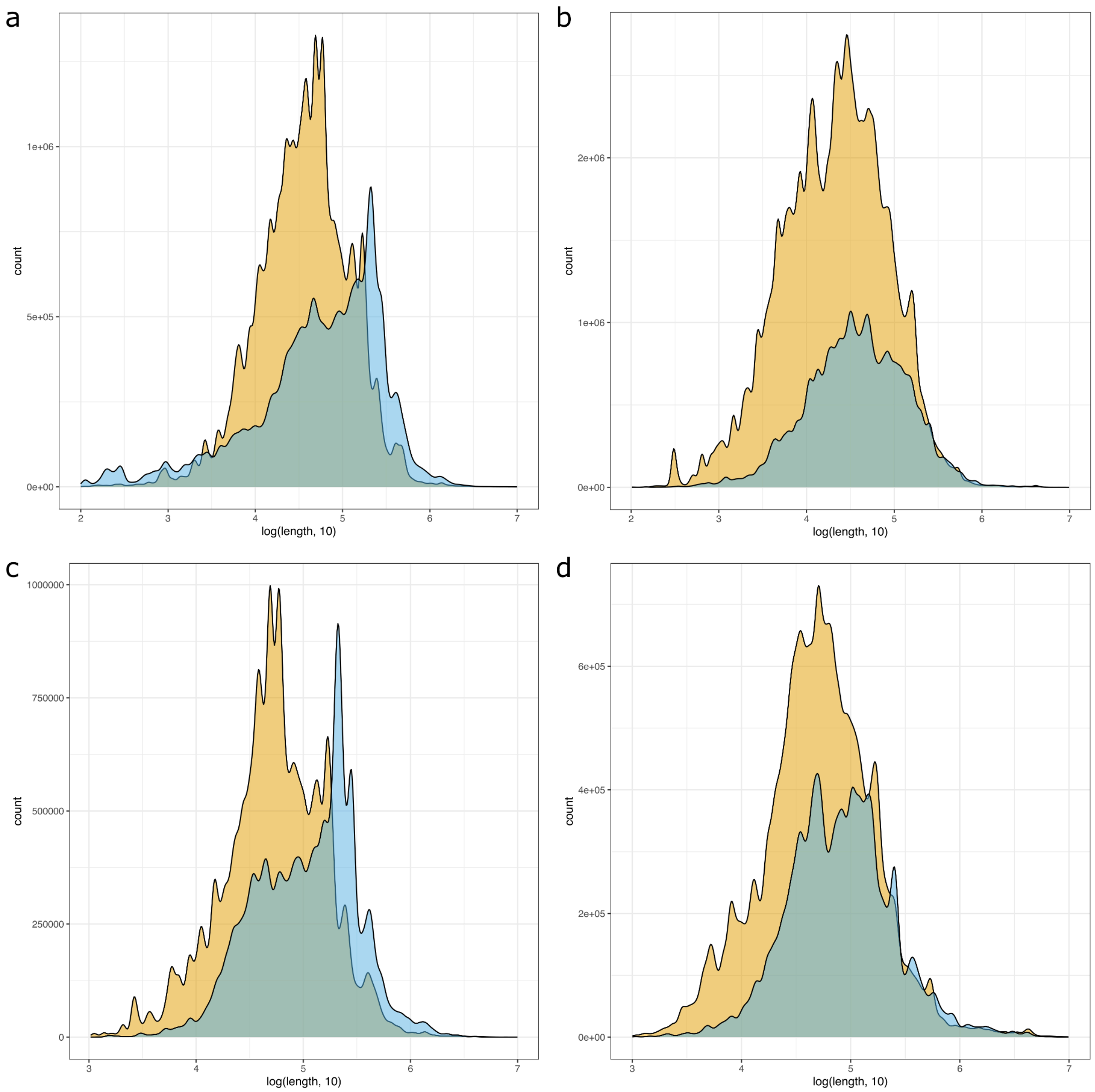
- Random SNP from map as new start
- New end based on median numsnp for the CNVR
- Compute new length
- If the difference between original and new length is above 20%, sample a new start SNP

- CNVRs moved to new, compatible, locations

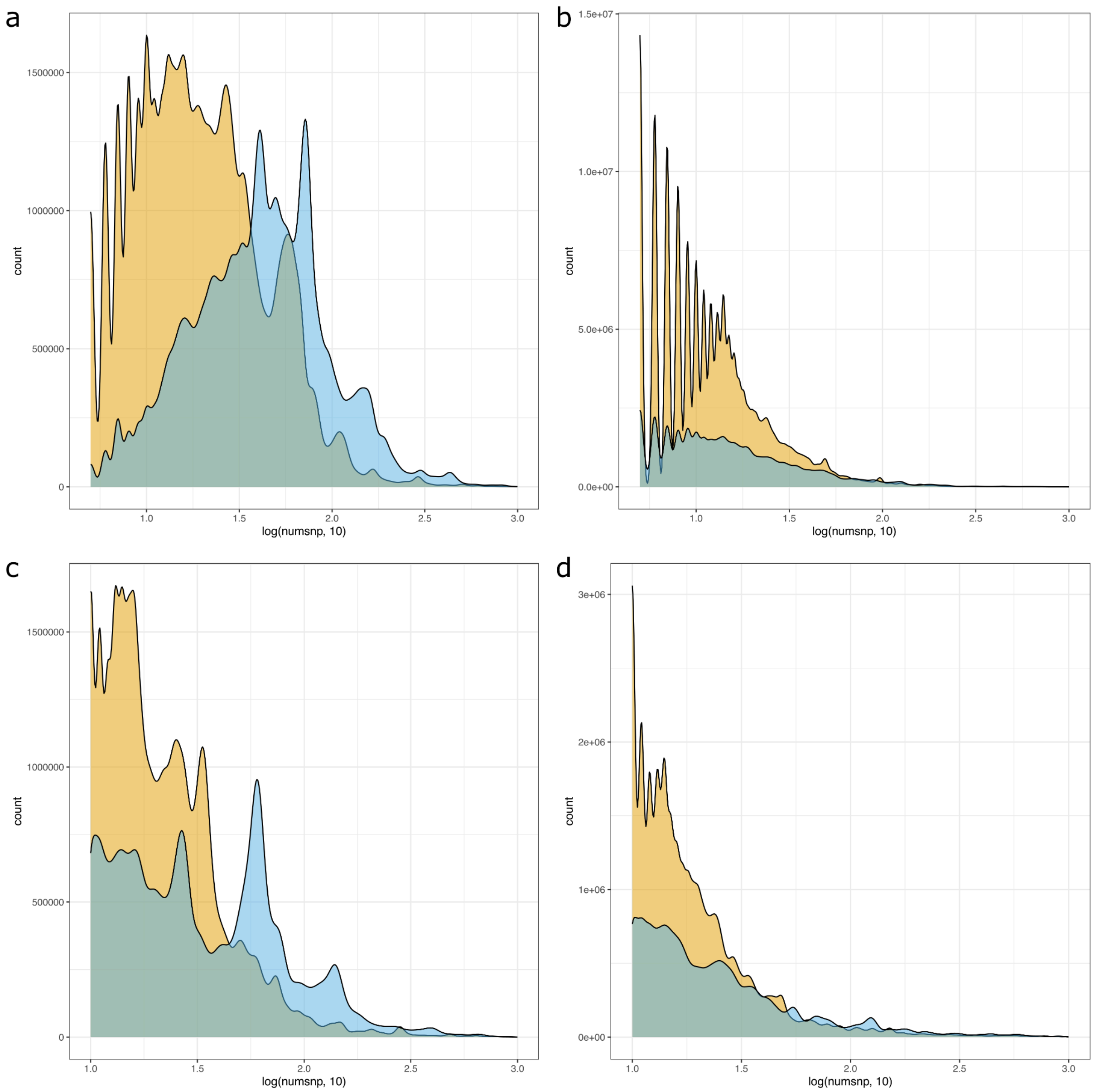
- For each CNVR, half CNVs inherit the start and the other half the end location from the CNVR they belong to
- The boundary is allowed to wiggle up to 10% in number of SNPs
- The other boundary is then computed using the original numsnp
- Final check to make sure no CNVs “fell out” of the chromosomal arm boundaries

- Shuffled CNVs set

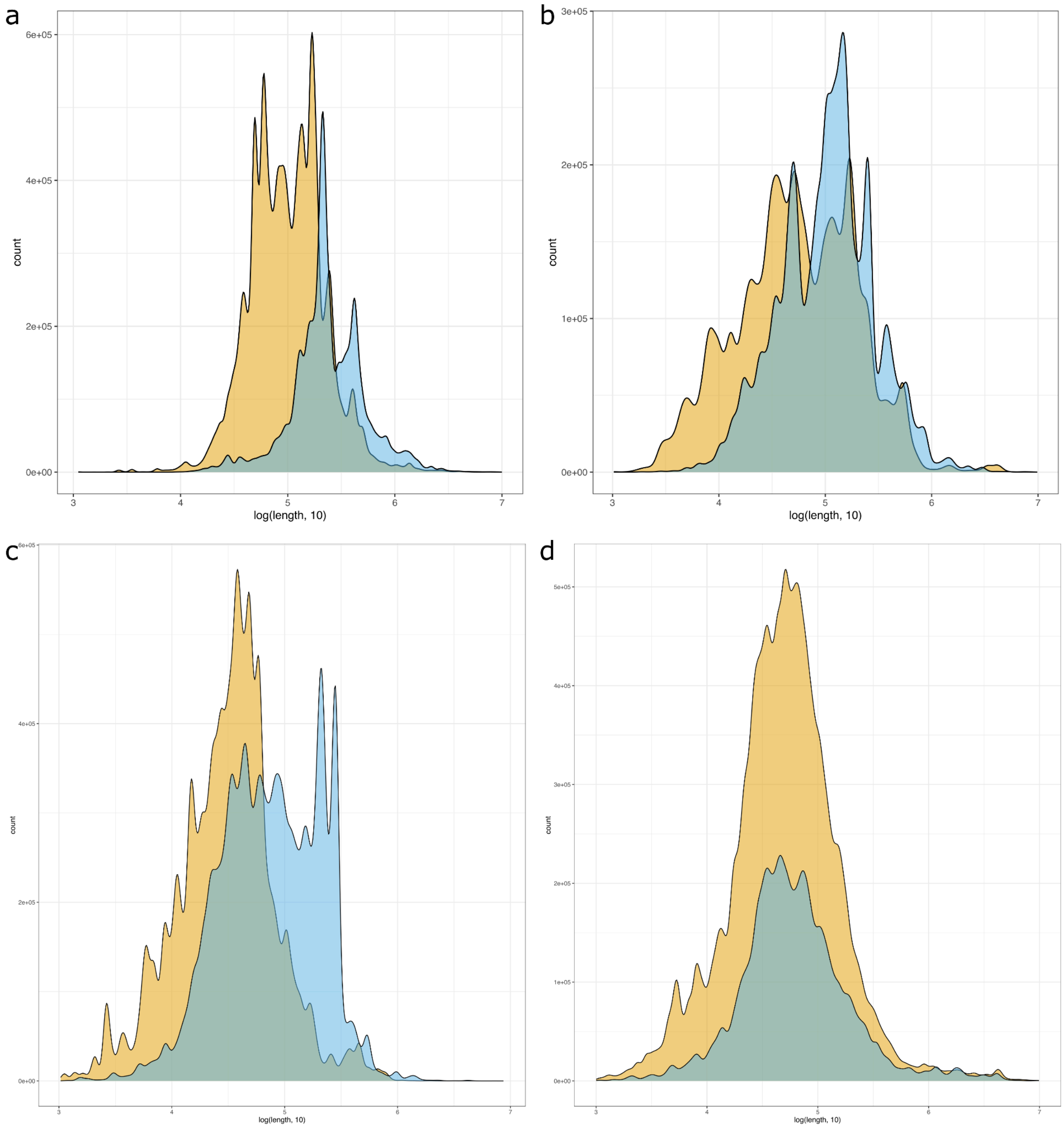
Supplementary figure 4. Schematics representation of the shuffling algorithm used to simulate the “CNV background”.



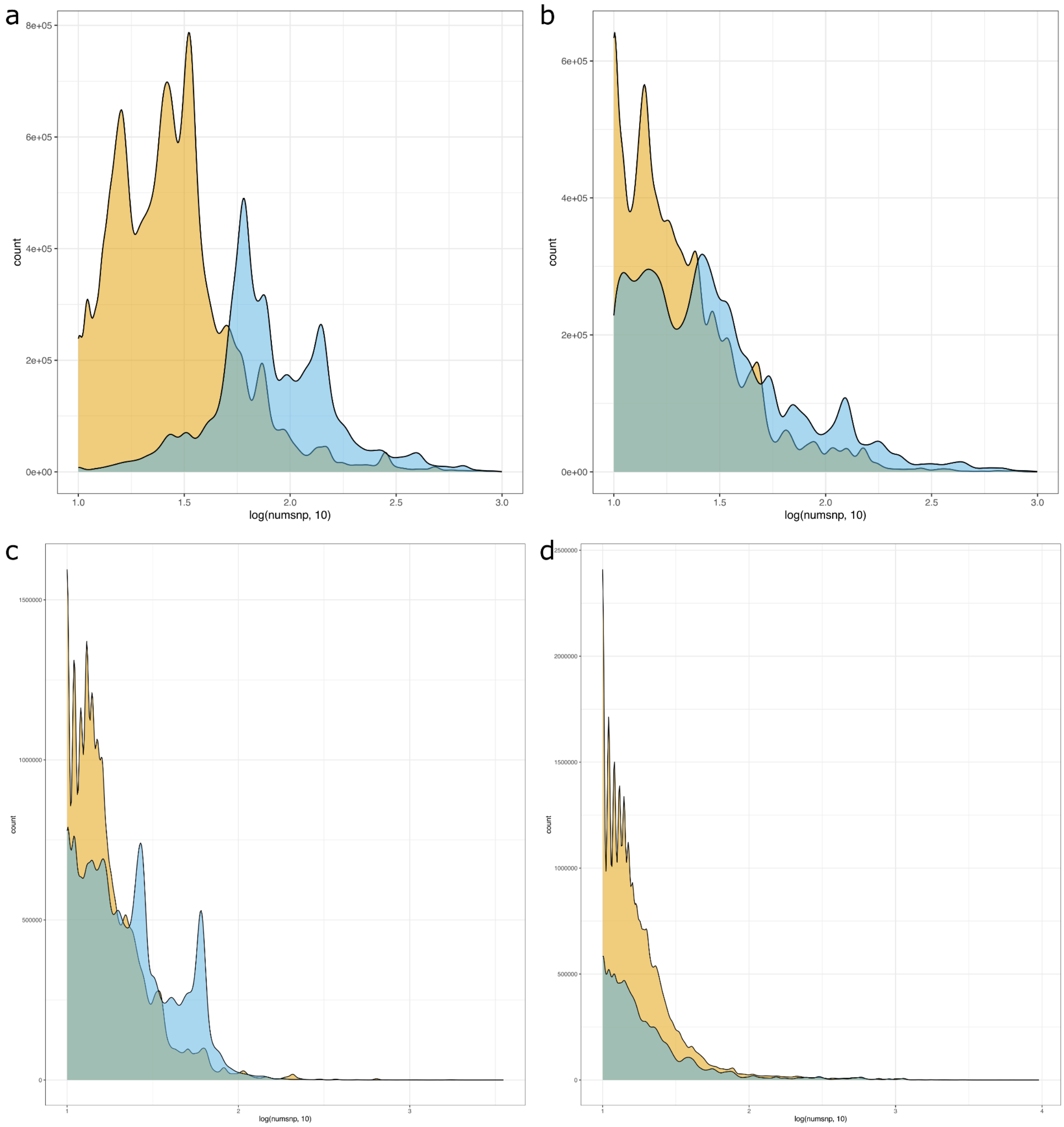
Supplementary figure 5. CNV length distribution for raw (top, a and b) and filtered (bottom, c and d) CNVs calls in UKB (left, a and c) and deCODE (right, b and d). For each group, deletions are in yellow and duplications in blue. Note the change of scale on the x-axes between the top and bottom row.



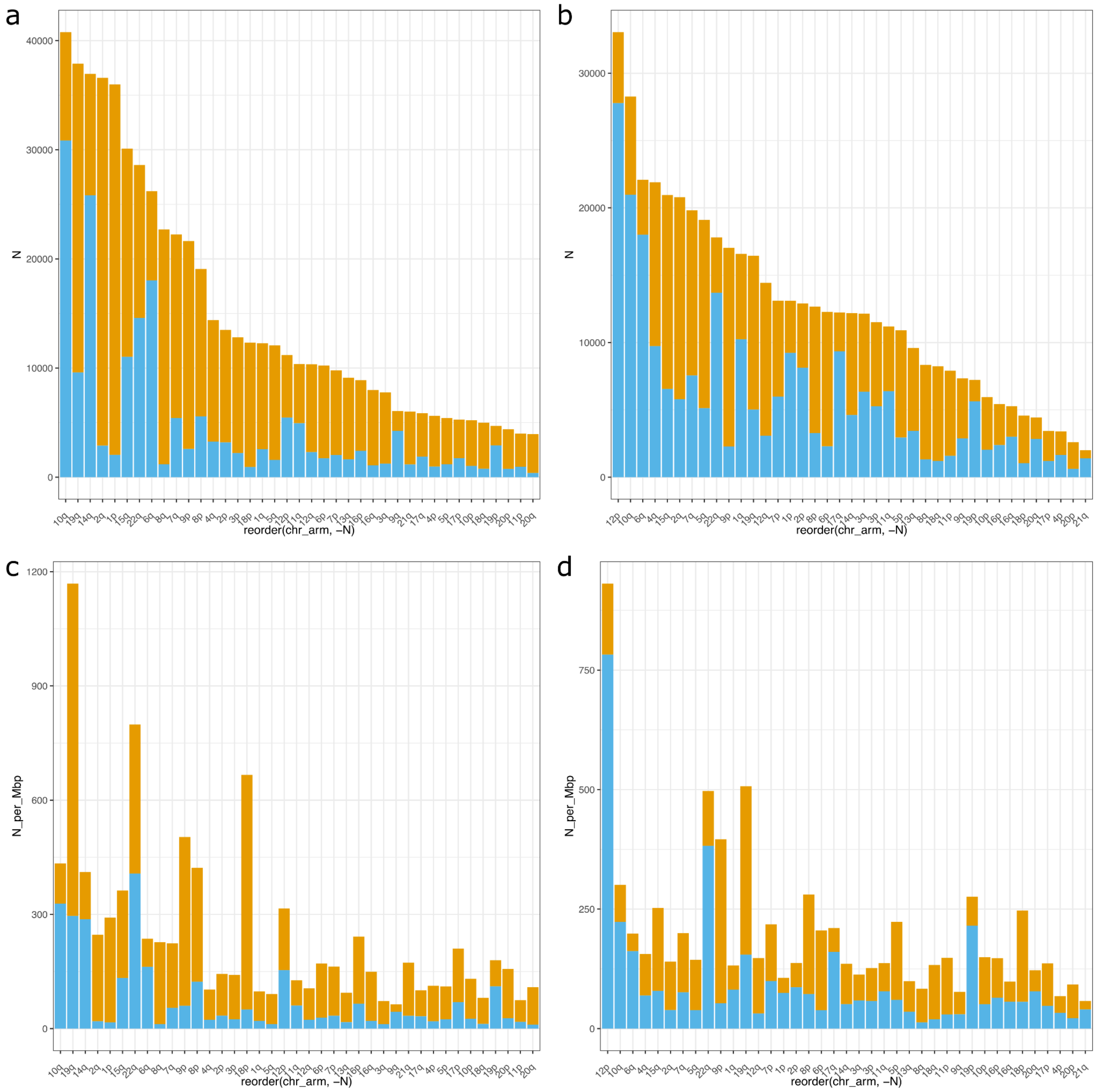
Supplementary figure 6. Number of SNP markers per CNV distribution for raw (top, a and b) and filtered (bottom, c and d) CNVs calls in UKB (left, a and c) and deCODE (right, b and d). For each group, deletions are in yellow and duplications in blue. Note the change of scale on the x-axes between the top and bottom row.



Supplementary figure 7. CNV length distribution for validated CNVs, i.e. predicted true with a probability above 0.75, (top, a and b) and discarded (bottom, c and d) CNVs calls in UKB (left, a and c) and deCODE (right, b and d). For each group, deletions are in yellow and duplications in blue.



Supplementary figure 8. Number of SNP markers per CNV distribution for validated CNVs, i.e. predicted true with a probability above 0.75, (top, a and b) and discarded (bottom, c and d) CNVs calls in UKB (left, a and c) and deCODE (right, b and d). For each group, deletions are in yellow and duplications in blue.



Supplementary figure 9. Number of CNVs per chromosomal arm in UKB (left, a and c) and deCODE (right, b and d). For each group, deletions are in yellow and duplications in blue. Chromosomal arms are sorted on the x axes based on the absolute number of CNVs in two samples separately (top row, a and b). The same order is kept when showing the number of CNV per Mbp in the bottom row (c and d)

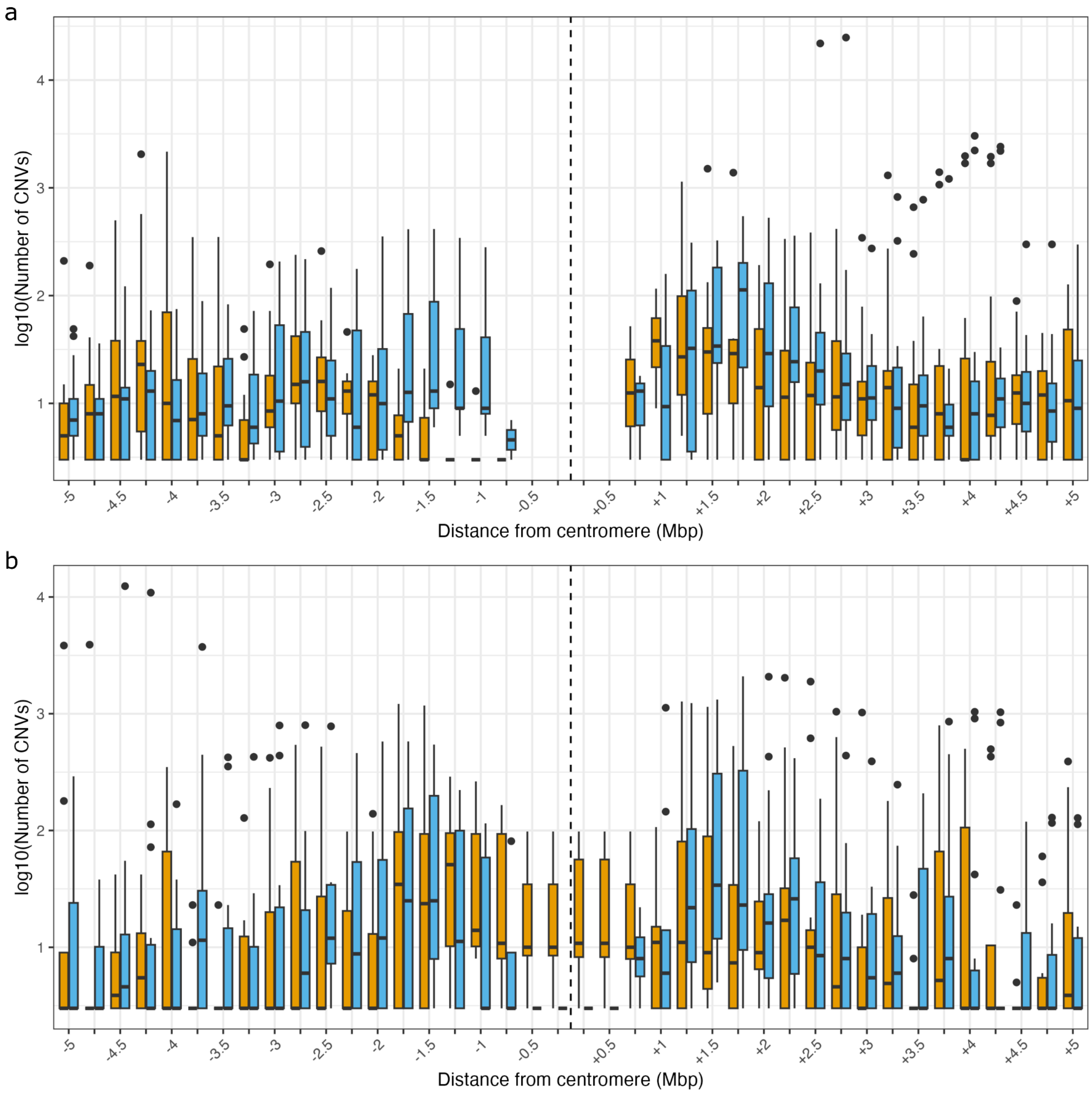


Figure 10. CNVs frequency in the centromeric region of chromosomes 1 to 22 across the two samples, a) UKB, b) deCODE. For each group, deletions are in yellow and duplications in blue. Dashed line marks the center of the chromosome. Each pair of box plots represent a 250 kbp window. The plot shows 5 Mbp on each side of the centromere.

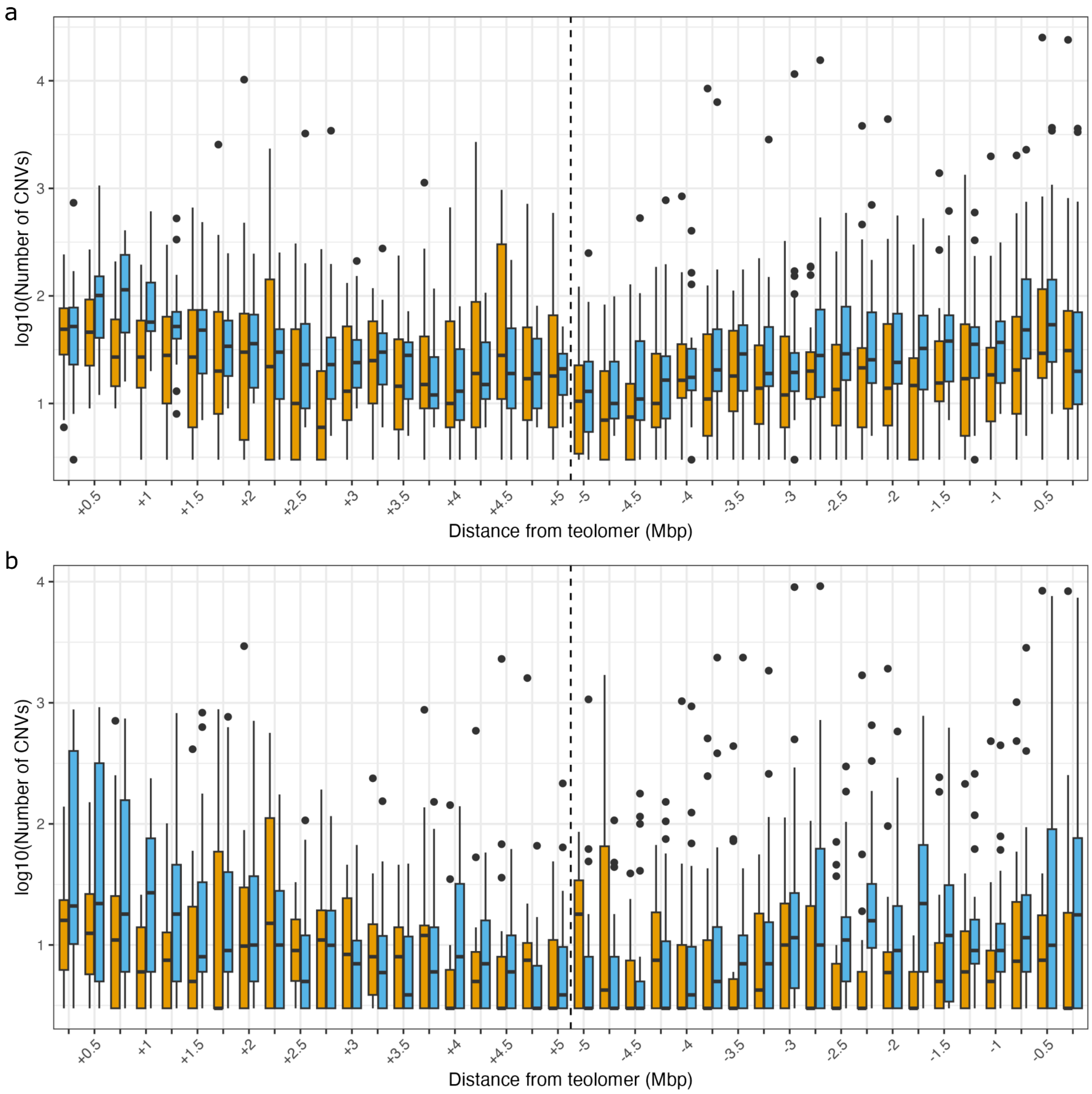


Figure 11. CNVs frequency in the telomeric region of chromosomes 1 to 22 across the two samples, a) UKB, b) deCODE. For each group, deletions are in yellow and duplications in blue. Dashed line marks the rest of the chromosomes (omitted from the plot). Each pair of box plots represent a 250 kbp window. The plot shows 5 Mbp from either telomeres at the respective end of each chromosome.

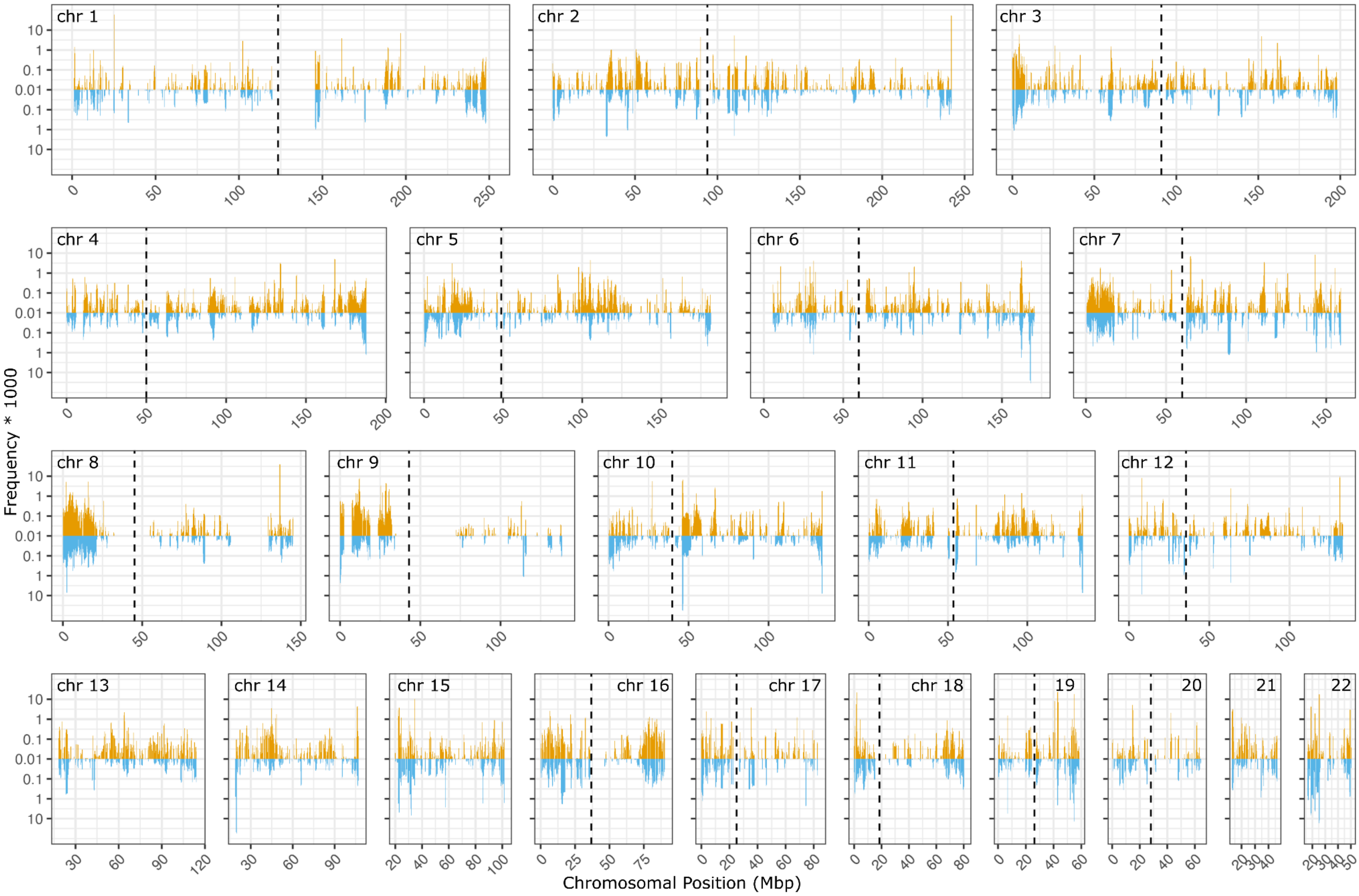


Figure 12. Genomic distribution of CNVs in UKB. Refer to figure 2 for the full legend.

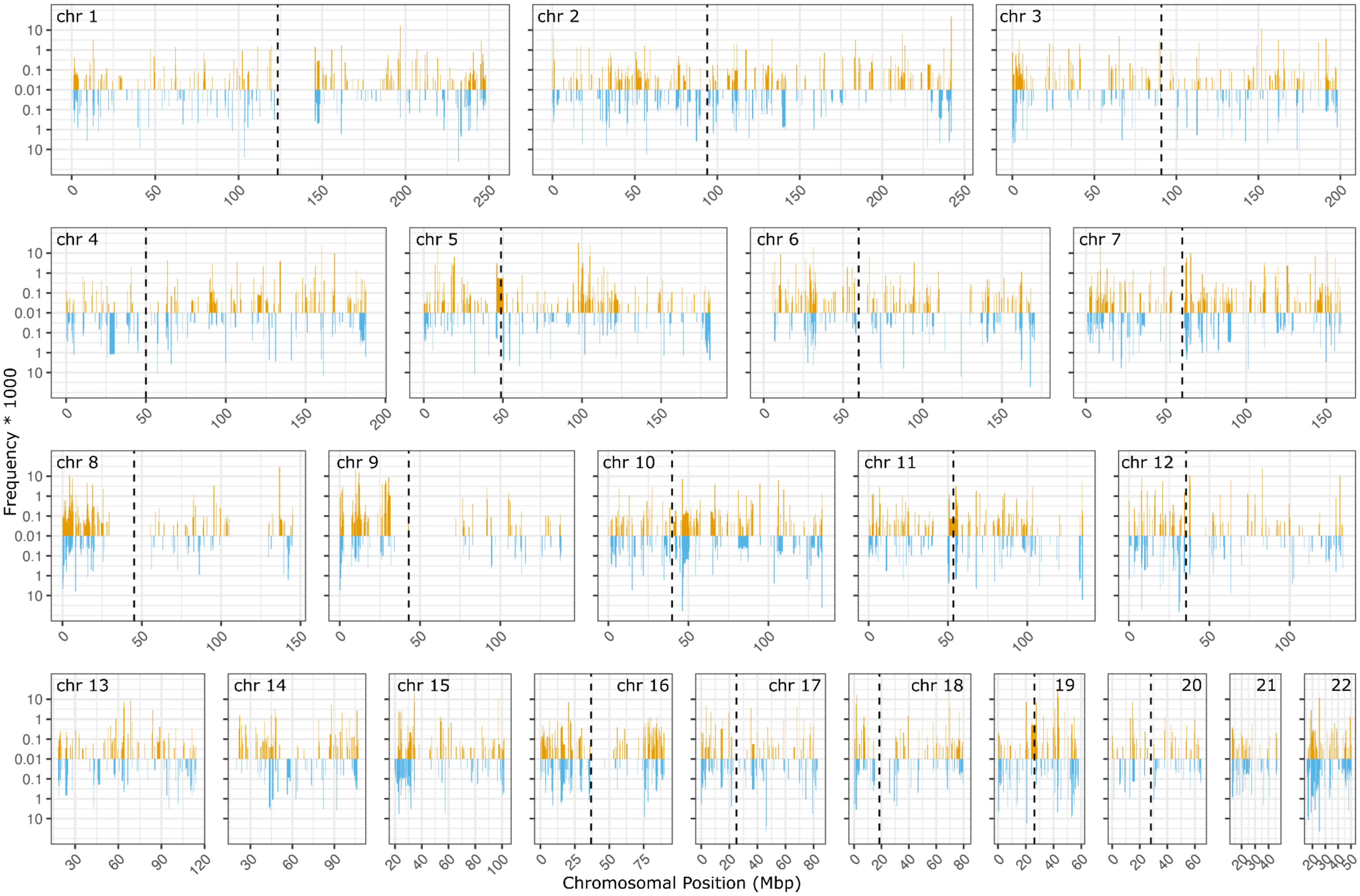
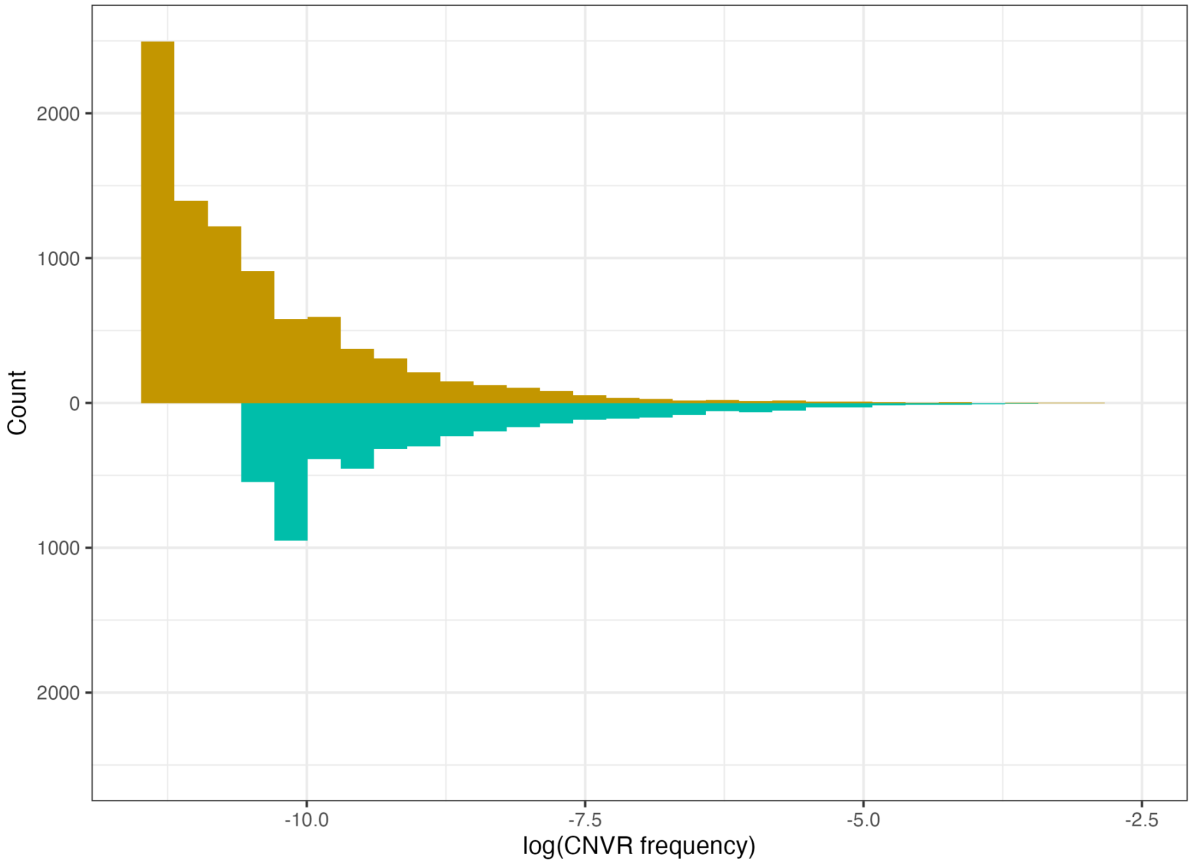
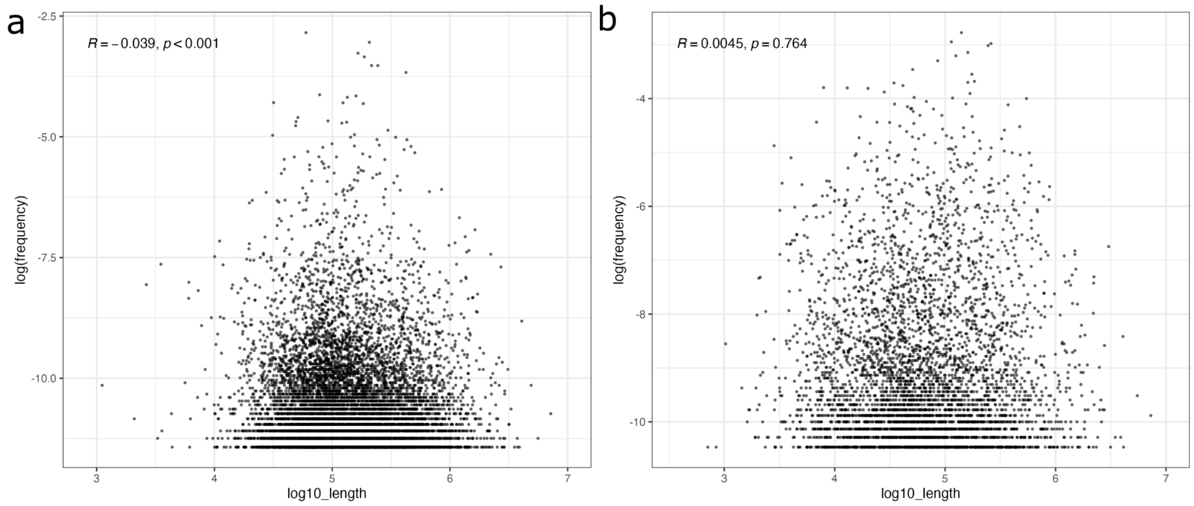


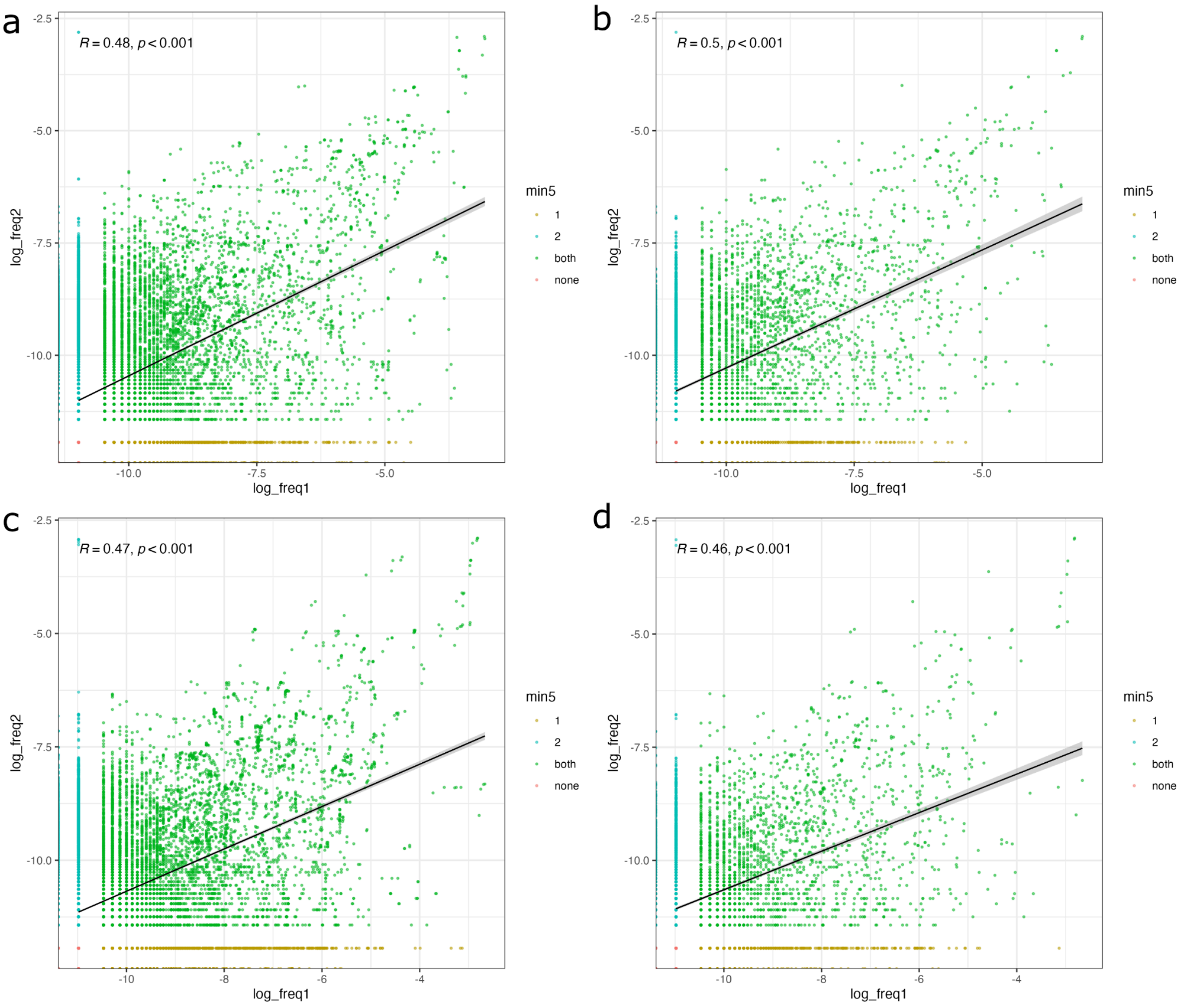
Figure 13. Genomic distribution of CNVs in deCODE. Refer to figure 2 for the full legend.



Supplementary figure 14. Histogram of the frequency distribution for CNVRs in UKB (positive values, yellow), and deCODE (negative values, green). CNVRs with less than 5 carriers (in each sample separately) are not shown. The difference in sample size is apparent, as CNVRs in UKB can have lower frequency but still at least 5 carriers compared to the deCODE set.



Supplementary figure 15. CNVRs frequency (natural log) over length (log 10) scatter plot for a) UKB and b) deCODE. The length is distributed around ~100 kbp in both samples (slightly higher in UKB and slightly lower in deCODE) but no clear correlation with the frequency is present ($R < 0.05$).



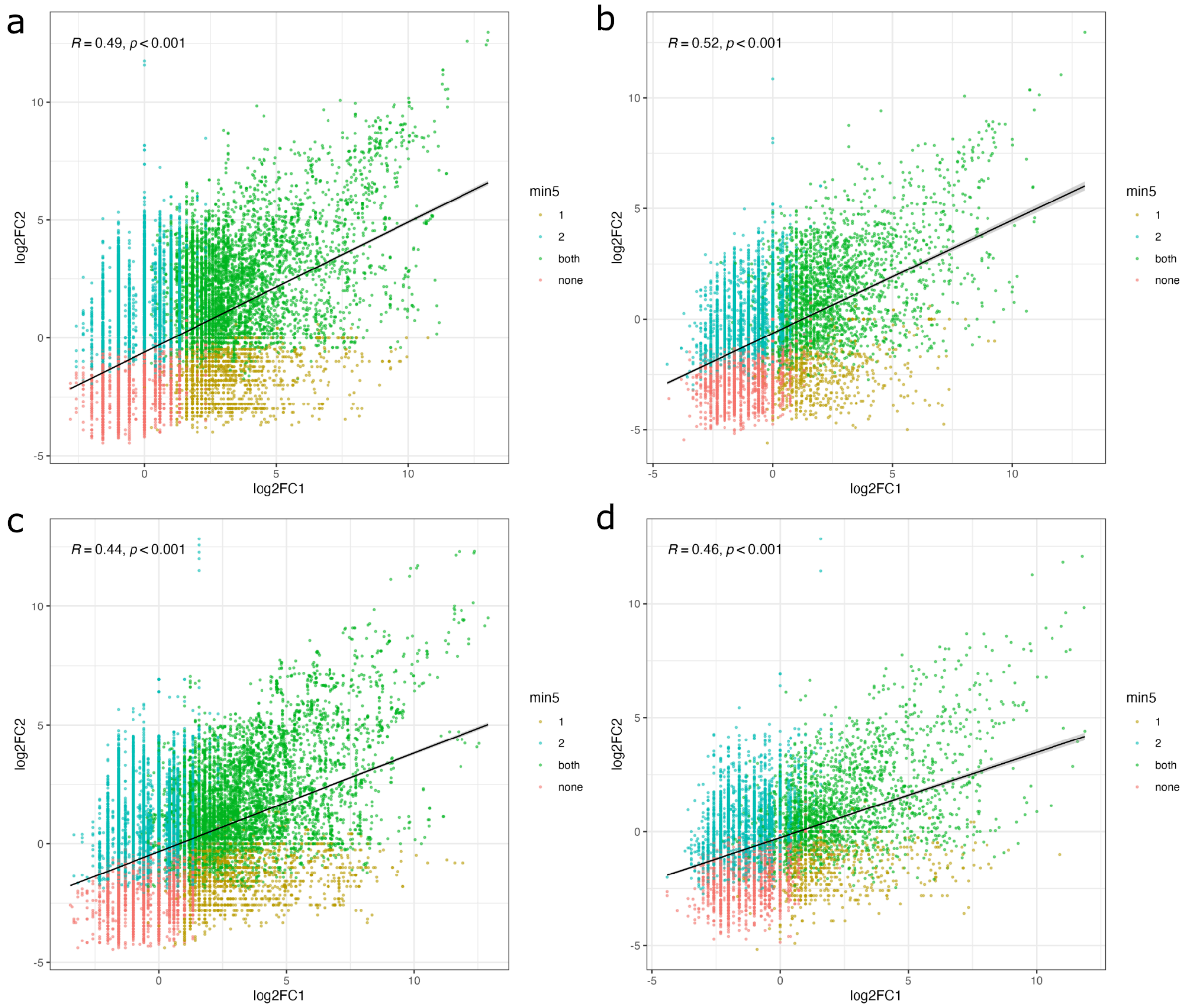
Supplementary figure 16. Correlations of the CNVs carrier frequency in each 50 kbp (left a and c) and 250 kbp (right, b and d) regular windows across the two samples. Deletions on top (a and b), duplication on the bottom (c and d). Markers with an N between 1 and 4 (below 5 but not 0) have been all assigned to 3 to comply with limitation in export of individualised data from the secure compute environment.

Genotype	Frequency group	N UKB	N deCODE	N shared
1	1	45755	53301	44180
1	2	10256	3283	1423
1	3	1298	772	253
1	4	181	144	61
1	5	10	0	0
2	1	46414	52836	44564
2	2	9865	3369	1557
2	3	1120	1164	368
2	4	84	129	33
2	5	17	2	0

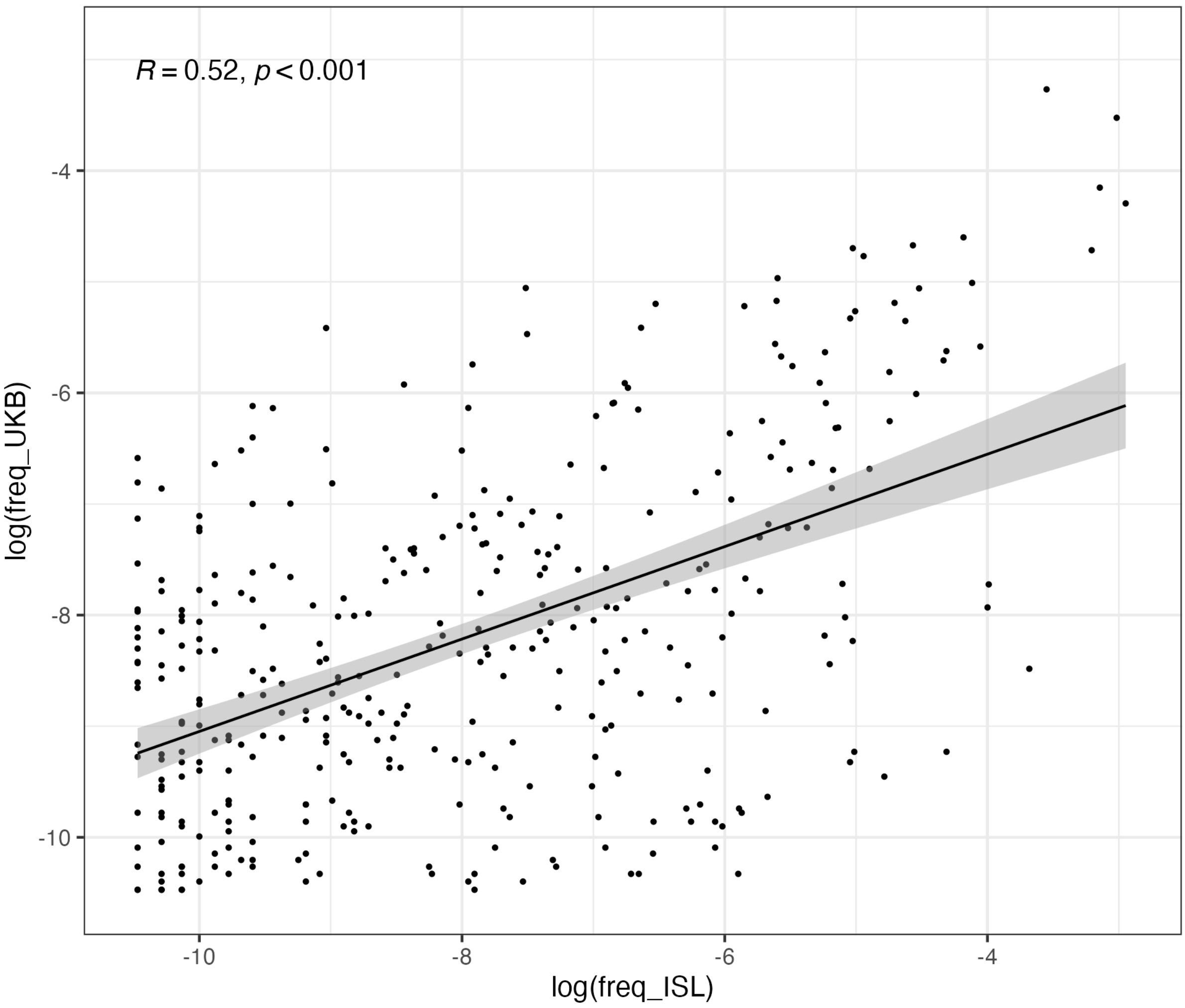
Supplementary table 2. Number of 50kbp windows in each frequency group, defined as: 1, CNV carriers frequency in the marker is less than 1 times in 40,000 samples (less than 5 total events in deCODE); 2, between 1/40,000 and 1/4,000; 3, between 1/4,000 and 1/400; 4, between 1/400 and 1/40; 5, above 1/40 (2.5% prevalence).

Genotype	Frequency group	N UKB	N deCODE	N shared
1	1	7568	9764	7027
1	2	3275	1316	601
1	3	578	351	124
1	4	72	67	25
1	5	5	0	0
2	1	8549	9964	7952
2	2	2593	1093	521
2	3	322	389	100
2	4	27	51	11
2	5	7	1	0

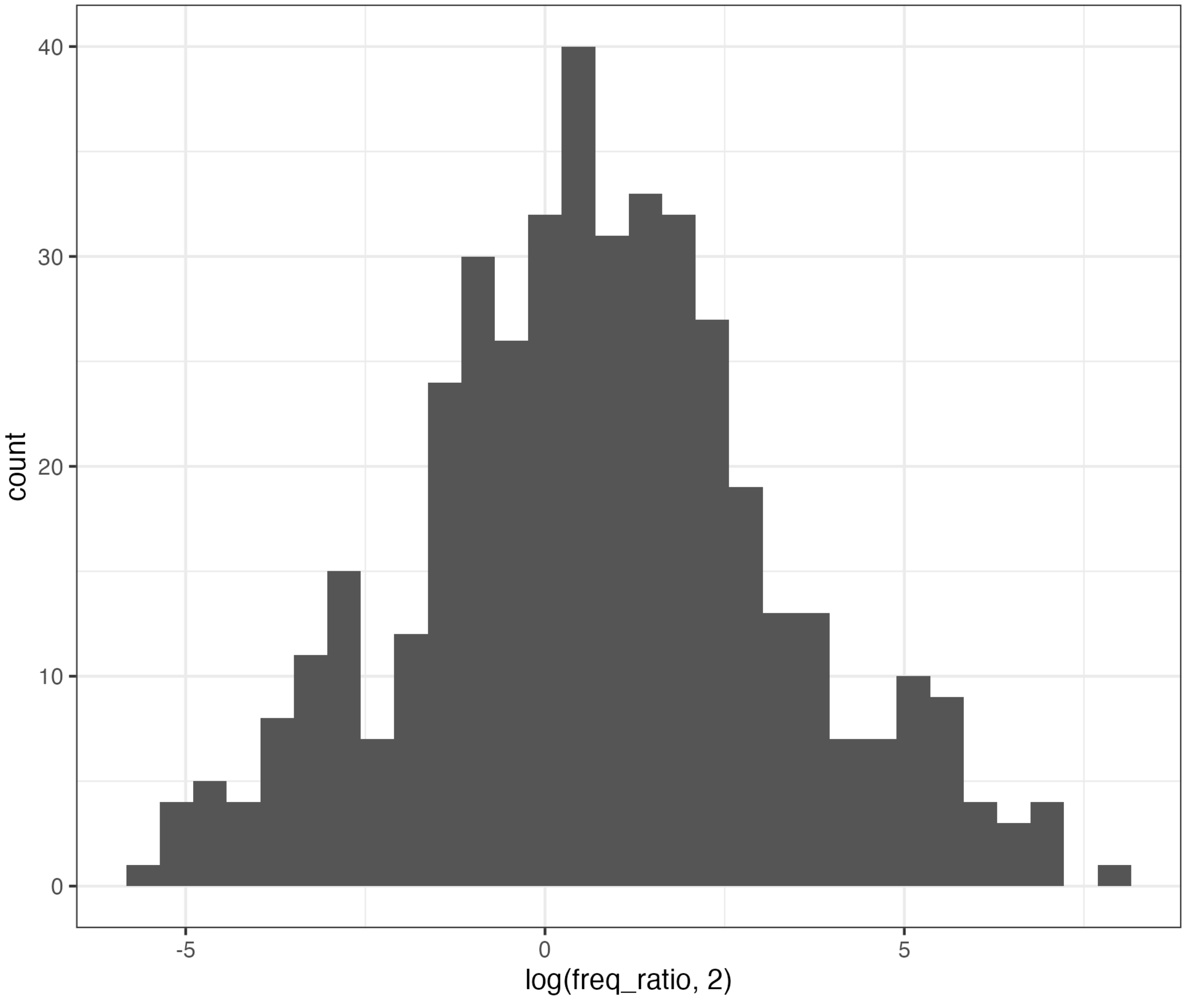
Supplementary table 3. Number of 250kbp windows in each frequency group, defined as: 1, CNV carriers frequency in the marker is less than 1 times in 40,000 samples (less than 5 total events in deCODE); 2, between 1/40,000 and 1/4,000; 3, between 1/4,000 and 1/400; 4, between 1/400 and 1/40; 5, above 1/40 (2.5% prevalence).



Supplementary figure 17. Correlations of the log2FC values in each 50 kbp (left a and c) and 250 kbp (right, b and d) regular windows across the two samples. Deletions on top (a and b), duplication on the bottom (c and d).



Supplementary figure 18. CNVRs frequency correlation between UKB (y axis) and deCODE (x axis) in the set of regions shared across the two cohorts and with at least 5 markers in each.



Supplementary figure 19. deCODE vs UKB log2FR distribution in the set of regions shared across the two cohorts and with at least 5 markers in each.

Genotype	Frequency group	N UKB	N deCODE	N shared
1	1	11486	11415	10612
1	2	1294	1274	440
1	3	177	209	66
1	4	35	57	18
1	5	1	1	0
2	1	10620	10878	9610
2	2	2103	1610	696
2	3	244	377	91
2	4	20	81	9
2	5	6	10	3

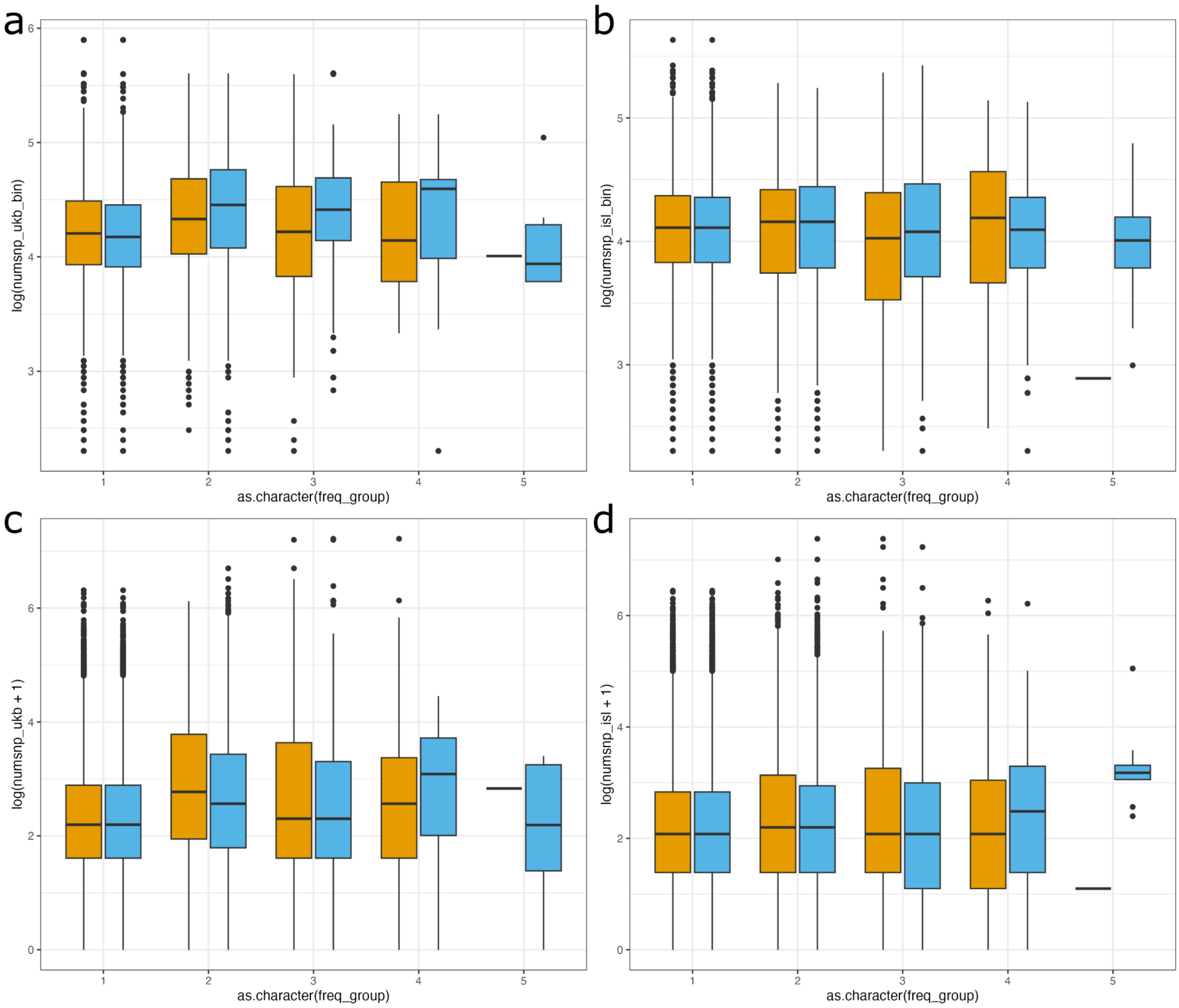
Supplementary table 4. Number of genes in each frequency group, defined as: 1, CNV carriers frequency in the marker is less than 1 times in 40,000 samples (less than 5 total events in deCODE); 2, between 1/40,000 and 1/4,000; 3, between 1/4,000 and 1/400; 4, between 1/400 and 1/40; 5, above 1/40 (2.5% prevalence).

Genotype	Frequency group UKB	N
1	1	7694
1	2	291
1	3	4
2	1	6142
2	2	460
2	3	5

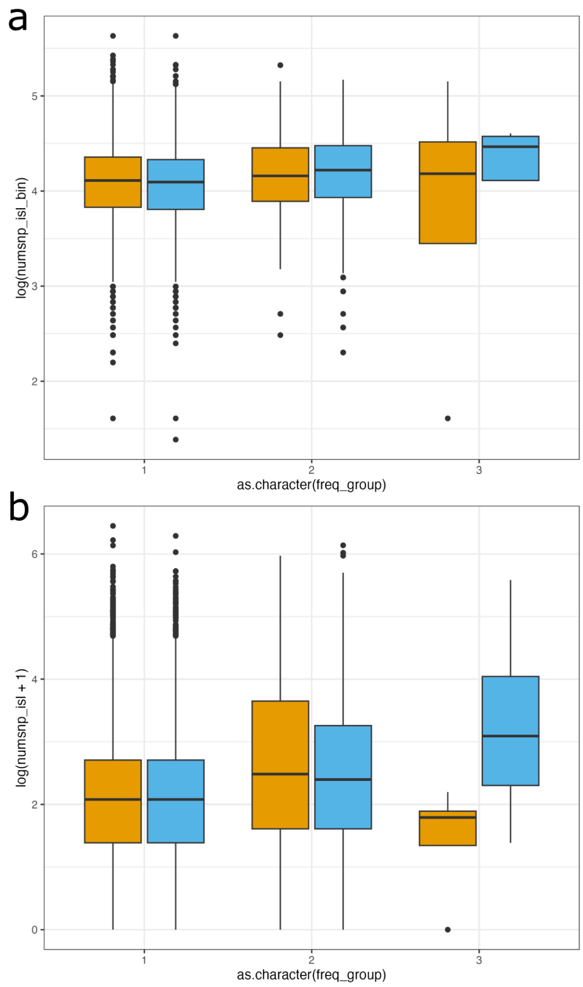
Supplementary table 5. Frequency groups counts in UKB for genes that have a CNV carrier frequency of zero in deCODE. Deletions and duplications are considered separately.

Genotype	Frequency group deCODE	N ISL
1	1	4152
1	2	152
1	3	24
2	1	2183
2	2	99
2	3	8
2	4	3

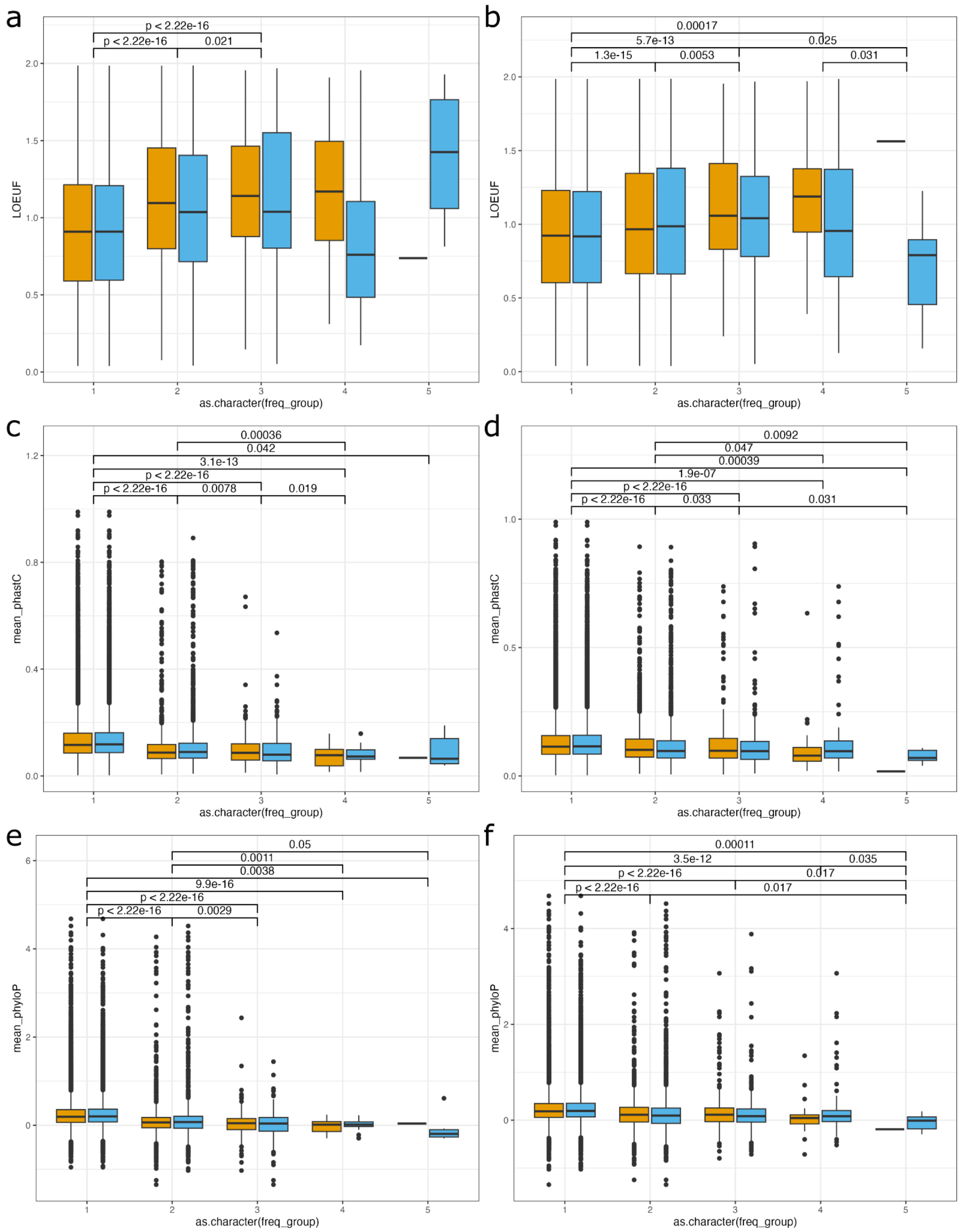
Supplementary table 6. Frequency groups counts in deCODE for genes that have a CNV carrier frequency of zero time in UKB. Deletions and duplications are considered separately.



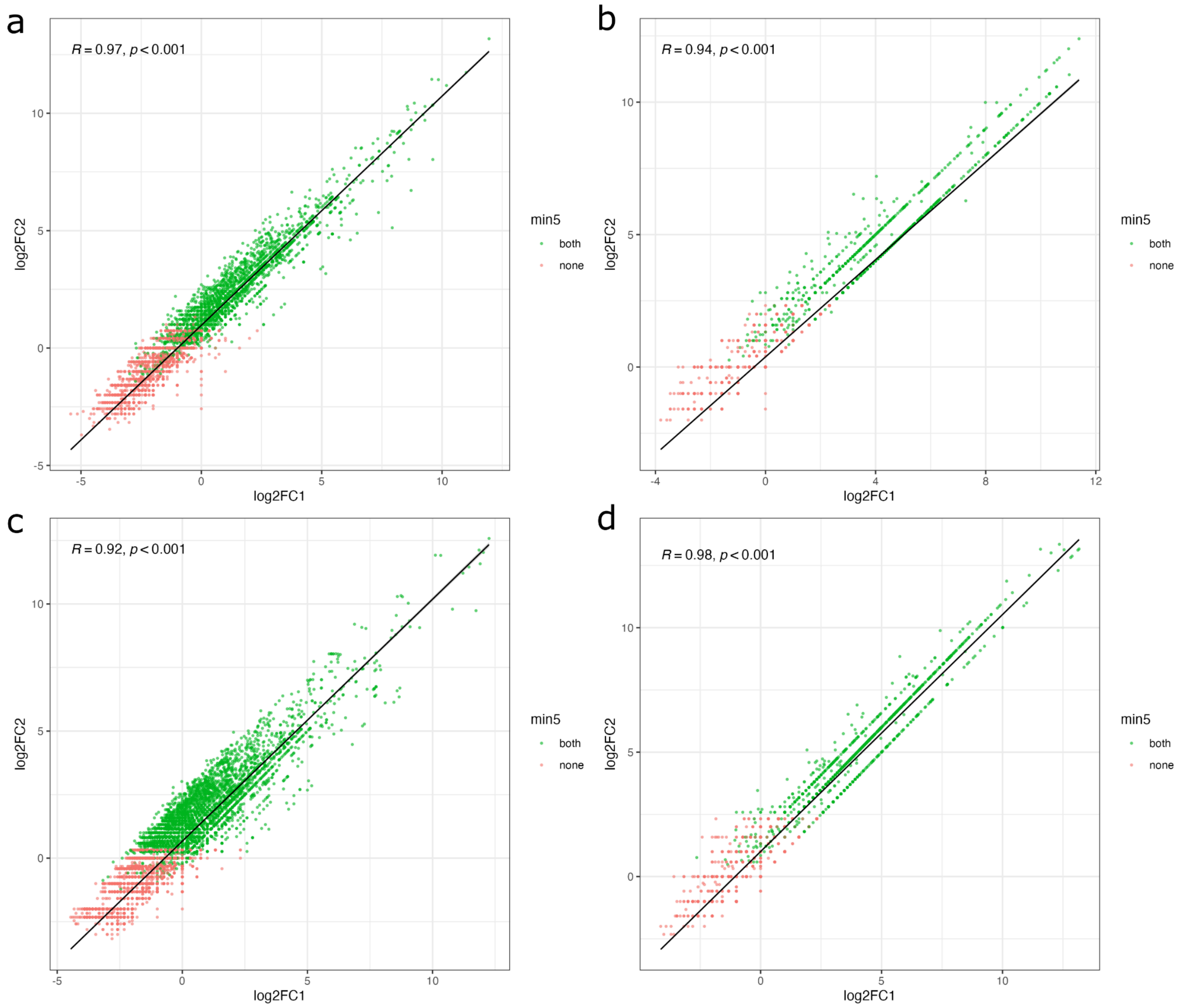
Supplementary figure 20. SNP markers coverage for included genes in UKB (left, a and c) and deCODE (right, b and d) in the different frequency groups. For each group, deletions in yellow, duplications in blue. Top row (a and b) consider the 250 kbp window to which the gene belongs to, while the bottom row (c and d) consider the gene itself. Note that when looking at the gene the differences in length are not corrected for.



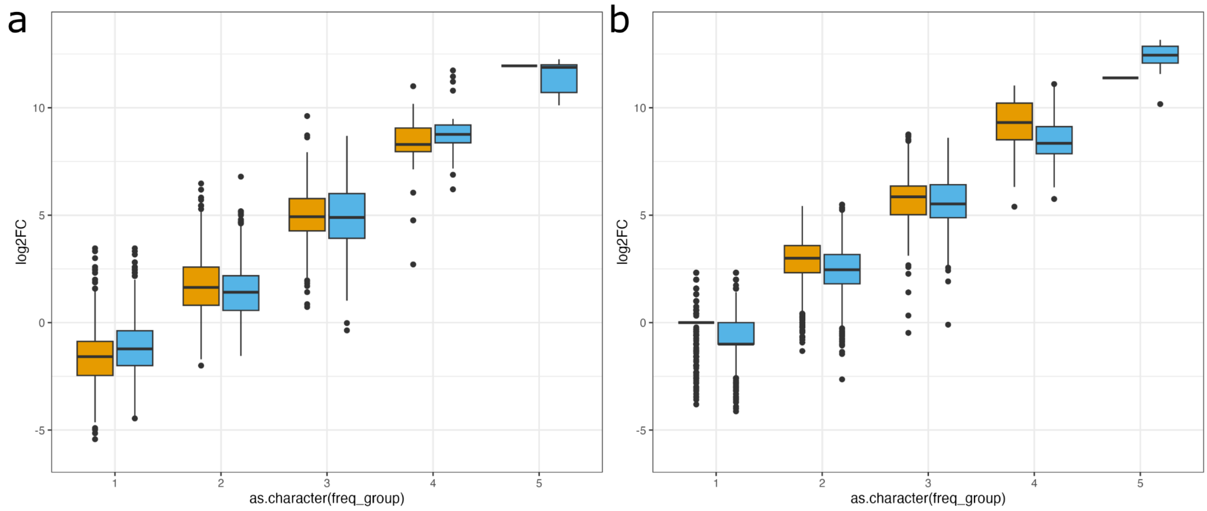
Supplementary figure 21. SNP markers coverage of deCODE "0s" over UKB frequency groups for the same genes. For each group, deletions in yellow, duplications in blue. Even though the frequency is substantially different for some genes (x in group 2 across dels and dups), the coverage in deCODE is not substantially different from the rest of the genes (supplementary figure g1), especially for the windows "behind" the gene (panel a), that are not sensitive to the gene size.



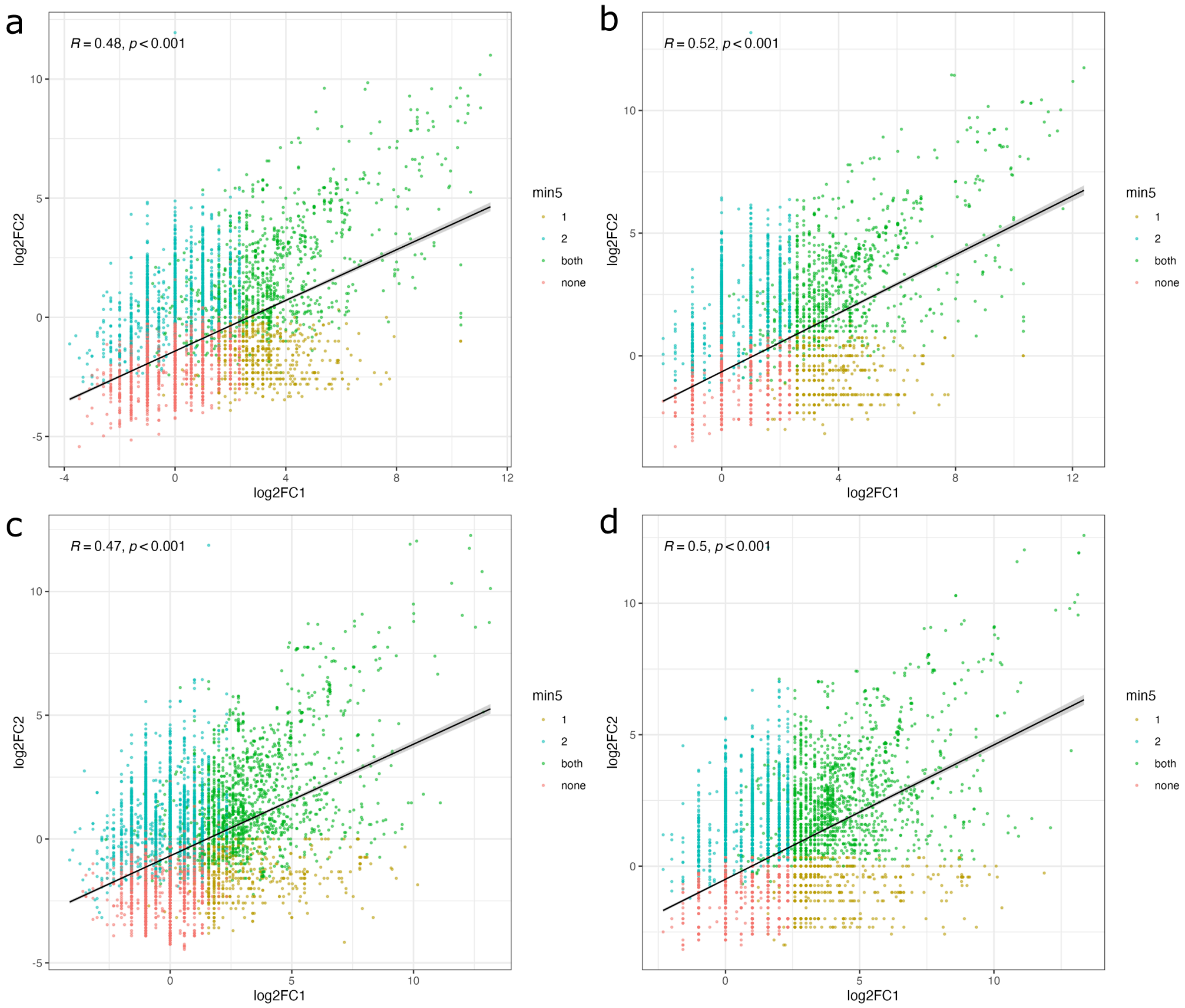
Supplementary figure 22. Genes metrics box-plots across the different frequency groups in UKB (left, a, c, e) and deCODE (b, d, f). For each group, deletions in yellow, duplications in blue. Top row (a and b) LOEUF score, middle (c and d) phastC, bottom (d and e) phyloP. Statistically significant differences between any pair of frequency groups are marked
with the corresponding p-value from a Wilcoxon test.



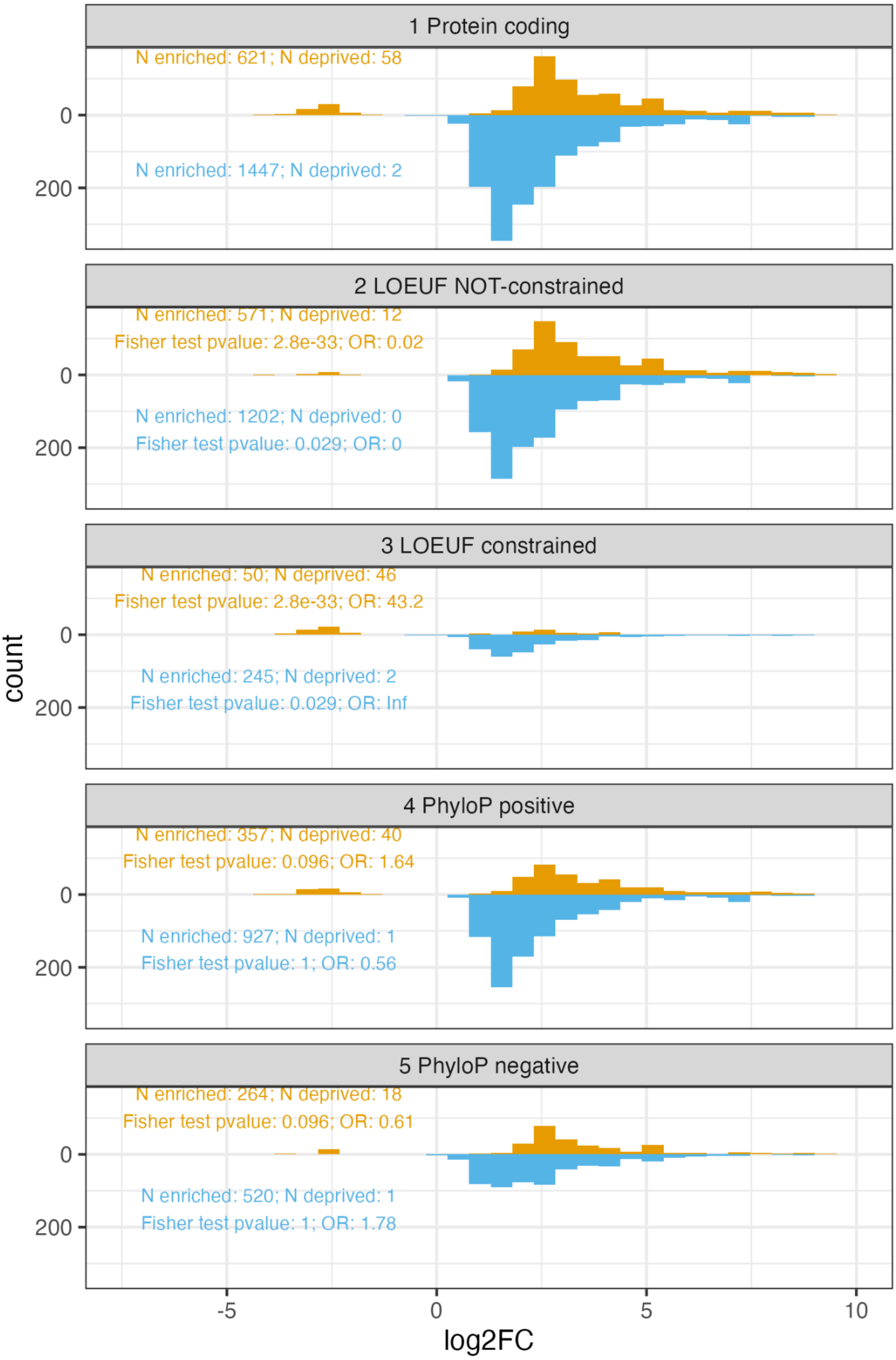
Supplementary figure 23. Log2FC correlation across simulations in UKB (left, a and c) and deCODE (right, b and d). Top row (a and b) deletions, bottom row (c and d) duplications



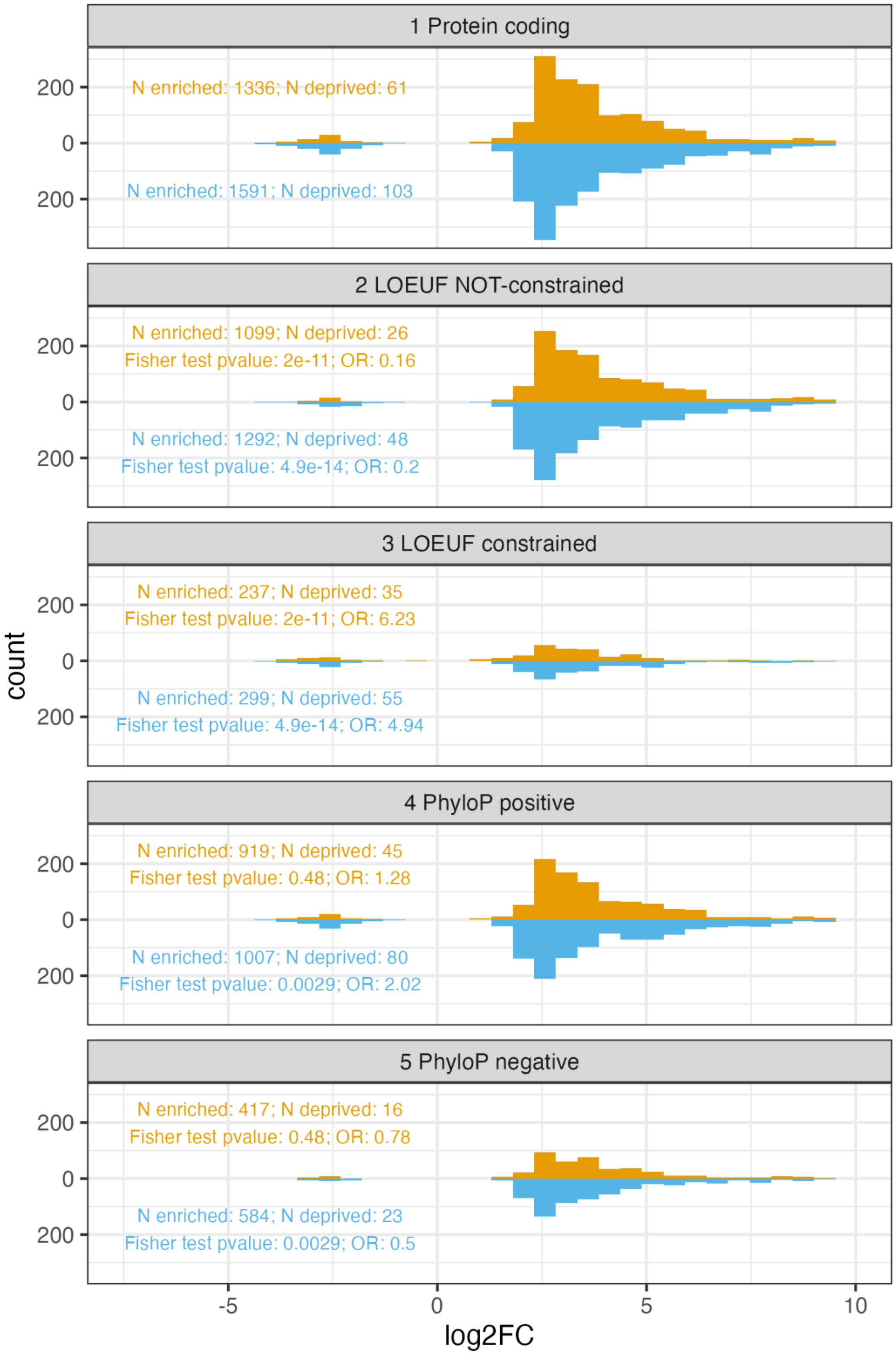
Supplementary figure 24. Log2FC box-plots in the different frequency groups in a) UKB and b) deCODE. For each group, deletions in yellow, duplications in blue.



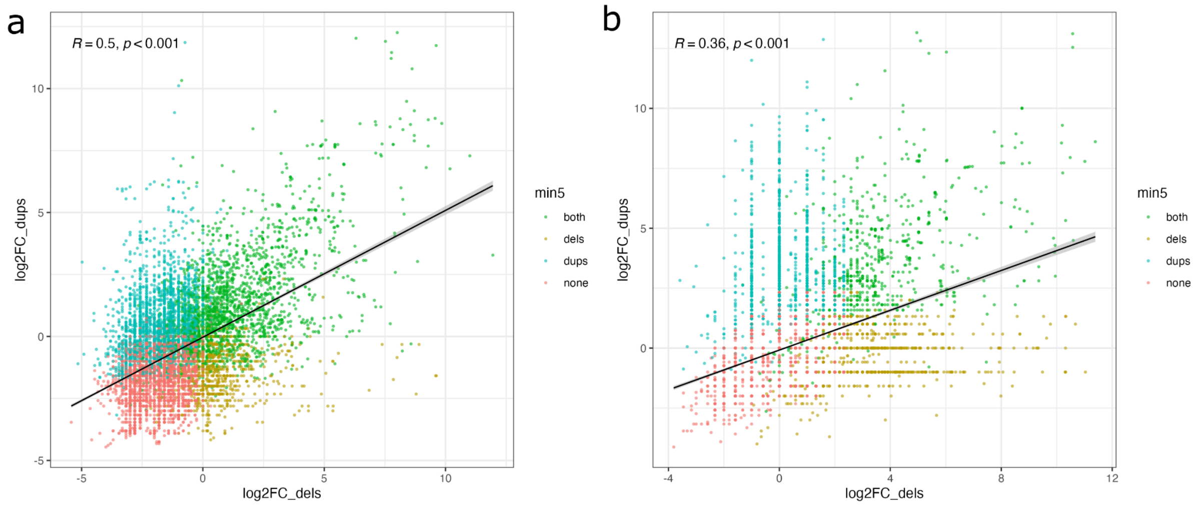
Supplementary figure 25. Log2FC correlation across samples using the two different simulations, CNVRs shuffling on the left (a and c) genes shuffling on the right (b and d). Top row (a and b) deletions, bottom row (c and d) duplications.



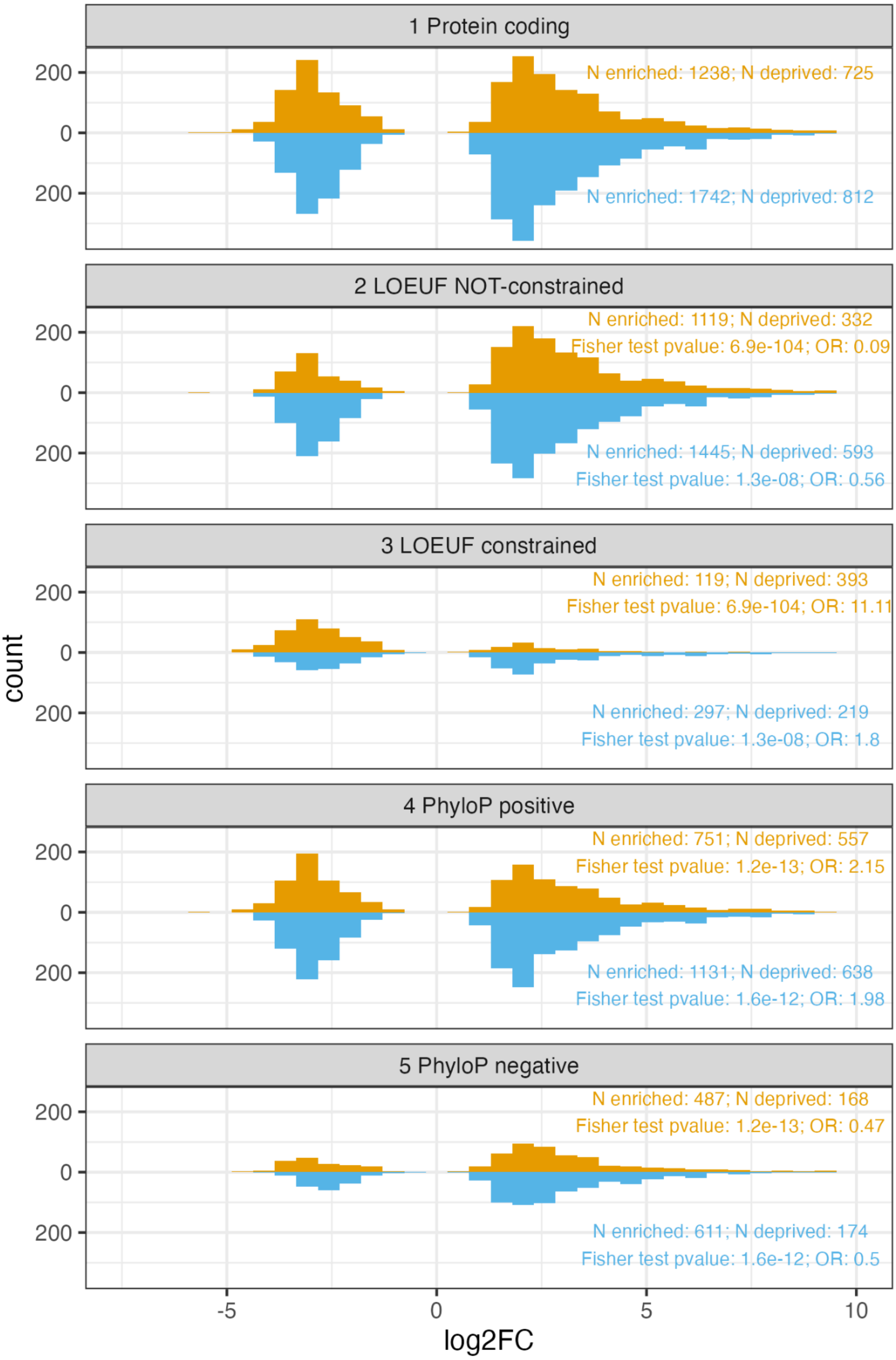
Supplementary figure 26. Log2FC distribution in a UKB subset, the N and median N (background value from the simulation) for all genes were divided by the UKB/deCODe sample size factor of 2.6, before recomputing log2FC score and fisher exact p-values. Refer to the legend of figure 3 for a full description.



Supplementary figure 27. Log2FC distribution in deCODE with a milder p-value threshold, 0.1. Refer to the legend of figure 2 for a full description.



Supplementary figure 28. Correlation between the log2FC for deletions and duplications in a given gene in UKB (a) and deCODE (b).



Supplementary figure 29. Log2FC distribution in all samples combined (UKB + deCODE). Refer to the legend of figure 3 for a full description.

HGNC			Freq		log2FC	log2FC	log2FC	numsnp	numsnp
symbol	GT	Freq UKB	deCODE	Freq ALL	UKB	deCODE	ALL	UKB	deCODE

ANTXRL	2	**0.053**	**0.059**	**0.055**	12.3	12.4	12.5	19	10
RHD	1	**0.060**	0.000	**0.044**	12.0	0.0	11.8	16	0
FRMD1	2	**0.034**	**0.052**	**0.039**	10.1	13.2	10.8	29	27
KIF25	2	**0.034**	**0.052**	**0.039**	10.3	11.6	10.8	27	21
OR4K5	2	**0.049**	0.000	**0.035**	11.9	1.6	11.9	3	0
ANXA8L1	2	**0.036**	**0.013**	**0.030**	12.0	10.1	11.9	0	0
NPY4R	2	**0.033**	**0.011**	**0.027**	11.9	9.9	11.7	3	0
PSG11	1	**0.022**	**0.030**	**0.025**	11.0	11.4	11.3	4	2
AFDN	2	**0.011**	**0.047**	**0.021**	9.0	12.0	10.3	26	23
SLC2A3	2	0.008	**0.050**	**0.020**	8.7	13.1	10.5	40	20
SLC2A14	2	0.008	**0.051**	**0.020**	8.6	12.5	10.2	33	23
PSG1	1	**0.015**	**0.023**	**0.017**	10.2	11.0	10.6	4	2
CYP2E1	2	0.008	**0.041**	**0.017**	10.8	12.8	12.4	23	35
KIR2DL4	1	**0.018**	0.001	**0.014**	9.6	6.9	9.5	15	2
KIR3DL1	1	**0.018**	0.001	**0.013**	9.8	7.0	9.7	11	1
SYCE1	2	0.007	**0.029**	**0.013**	11.7	12.3	13.0	5	12
KANSL1	2	0.000	**0.043**	**0.012**	5.1	12.9	11.3	73	25
DISC1	2	0.000	**0.043**	**0.012**	-1.8	10.2	8.3	146	155
KIR2DL4	2	**0.014**	0.002	**0.010**	8.8	7.6	8.8	15	2
CFHR4	1	0.007	**0.018**	0.010	8.9	10.6	9.7	5	4
KIR3DL1	2	**0.011**	0.002	0.009	8.6	7.6	8.6	11	1
PSG2	1	0.006	**0.013**	0.008	9.2	10.2	9.8	4	2
PSG9	1	0.006	**0.011**	0.007	8.8	9.9	9.4	12	3
PSG5	1	0.004	**0.016**	0.007	8.2	10.5	9.4	4	1
AMY2B	2	0.000	**0.025**	0.007	1.5	11.1	9.8	4	7
AADAC	1	0.005	**0.012**	0.007	8.8	11.0	9.8	4	7
PSG4	1	0.003	**0.013**	0.006	8.2	10.2	9.3	4	1
OR4A47	1	0.000	**0.019**	0.005	5.3	10.7	9.2	4	0
ZNF626	2	0.000	**0.015**	0.004	1.8	10.4	9.1	12	10
XKR3	2	0.001	**0.012**	0.004	6.7	11.0	9.0	13	16
MALRD1	1	0.001	**0.012**	0.004	2.7	7.5	5.5	205	264
CCT8L2	2	0.001	**0.011**	0.004	7.4	10.9	9.6	3	1
AMY2A	2	0.000	**0.011**	0.003	1.5	9.9	8.6	0	0

AMY1A	2	0.000	**0.010**	0.003	1.5	9.8	8.5	0	0

Supplementary table 7. Genes with a CNV polymorphism in UKB or deCODE. CNV polymorphisms are defined as genes for which more than 1% of the population have a deletion (GT=1) or a duplication (GT=2) affecting it (considered separately). Entries above 0.01 in are highlighted in the frequency columns. Genes that are CNV polymorphisms in both populations (n=8) are highlighted as well.