Supplementary Materials

1 Sample preparation and validation

Pooled-DNA sample sequencing from single individuals
We performed pooled-DNA sequencing on 974 individuals enrolled in the Family Heart Study (FHS) (Higgins M et al 1996) and 178 individuals enrolled in the Silent Cerebral Infarction Transfusion Trial (SIT) (Vendt BA et al. 2008). Each pool of human DNA was designed to contain 450 ng of DNA per individual. Pooling was performed robotically using the Eppendorf epMotion 5075 pipetting robot (Eppendorf, Hauppauge, NY, U.S.A) in order to minimize dilution errors. Patients from the FHS study were divided into 8 pools ranging from 94 to 150 individuals, whereas the remaining 178 patients were pooled into a single ninth pool. We computationally generated a list of 14 genomic loci selected on the basis that they contained at least one single or double base pair insertion or deletion reported in dbSNP129 at various frequency ranges. We defined a genomic region of 400 bp centered on the targeted IN/DEL and we designed primers in order to target the known variant in a final amplicon with length ranging from 150 to 200 bp. Primers were designed as described previously (Druley TE et al. 2009). Each PCR reaction for each pool was repeated multiple times in order to minimize the likelihood of PCR errors appearing as rare variants in the sequencing output. Each PCR reaction was performed as described above with the exception of undergoing 28 total cycles. Each PCR mix contained 2.5 l of 10X PfuUltra Buffer, 10 M of forward and reverse primers, 1M Betaine (Sigma-Aldrich/Fluka,St. Louis, MO, U.S.A.), 1.25 units PfuUltra DNA polymerase and between 30 and 50 ng of template DNA, representing 50 genome copies per individual (see Supplementary Table 4). For every analyzed pool a positive and a negative control were generated. The positive control consisted of a synthetic pool containing substitutions and IN/DELs at the lowest possible frequency for each of the analyzed pools (i.e. 1 divided by the total number of alleles in the pool). The positive control was prepared as described, with PCR reactions performed for 28 cycles in order to match the sample preparation. The negative control consisted of a DNA fragment for which the sequence is known in order to generate a run-specific second order error model to be used in the data analysis. In order to generate a negative control, we performed PCR amplification on the M13 plasmid (New England Biolabs, Ipswich, MA, U.S.A.) generating a 1934 bp product (see Supplementary Table 4). PCR reaction was performed as described above and repeated multiple times. We then sequenced our samples using a single lane per pool and mapped back all the sequencing reads.

Four candidate loci identified by a GWA study performed on the 974 FHS patients were targeted in a total of 36 PCR reactions spanning 20,729 base pairs per individual (data not shown). Sample preparation, sequencing and analysis was performed as described. We identified a total of genomic variants that were also represented by the Illumina 6.0 genotyping array performed on each individual. Pearsons correlation coefficient between GWAS and SPLINTER variant frequencies was calculated by using the frequency of the minor allele for each

variant (Figure 3e).

Independent Validation of Putative Variants
Independent validation of putative variants identified by SPLINTER was performed by Sequenom (Sequenom, San Diego, CA, U.S.A.) according to the manufacturers protocol (for probes see Supplementary Table 5). Sanger sequencing validation was performed using the same primer pairs used for initial PCR amplification prior to pooled-DNA sequencing.

2 Sequencing reads mapping and error model calculation

Semi-local gapped sequence alignment
In order to efficiently map sequencing reads with gaps without compromising mapping accuracy (i.e. deviating from the optimal mathematical solution of the alignment), we developed a new Smith-Waterman-like alignment strategy (ref 1). This allowed us to have a pure and controlled implementation of dynamic programming while being feasible in terms of speed. Quality scores are ignored for the alignment. We first build a hash-map of the reference sequence with hash key size equal to k, which is defined as

$$k = \lfloor \frac{l-c}{c+1} \rfloor$$

where l is the length of the sequencing read and c is the maximum edit distance cutoff between the read sequence r and the reference sequence s. k is guaranteed to be the largest possible stretch of perfect match nucleotides achieved in the case of maximum entropy, i.e. when the edits are distributed uniformly in the reads, minimizing the length of the shortest read fragment. While l is run-dependent, c is defined by the users at the time of the alignment, leading to a consequent user-defined value of k.

When r is aligned to s, the first step is to hash-map all the substrings of r of length k to s. Every mapped susbtring allows to define the boundaries of a dynamic programming (DP) matrix for sequence alignment. The value of c determines the dimensions of the DP matrix, which are equal to l and $l + 2 * c$, assuming that the read will contain all the allowed edits.

Once the boundaries of the DP matrix are defined, we perform DP programming in the following way: first the matrix is initialized by setting the values of the first column to 0 (s dimension) as in Smith-Waterman (Smith TF, Waterman MS 1981) whereas the first row is set to 0 at position 0 and at progressively adding a gap penalty for every increasing position (r dimension) as in Needleman-Wunsch (Needleman SB, Wunsch CD. 1970). This strategy allows every alignment to start at any position in the reference sequence but always at the first position of the read, therefore being semi-local. Gaps are inserted according to an affine-gap penalty model (Durbin R et al 1998), adopting a gap insertion penalty of 2 and gap extension penalty of 1.

Traceback is peformed starting from the highest scoring position in the last column (corresponding to the last position in r) until the first position of r is

reached. The final result is the optimal mathematical solution for the gapped semi-local alignment of r with respect to s. r and its reverse complement are both mapped to the positive strand of s. If r aligns to multiple loci of s with the same minimum edit distance, its alignment to s is discarded in order to minimize noise due to spurious mapping. Insertions and deletions present inside a nucleotide homopolymer (i.e. AAA,CCC,GGG ...) are aligned at the beginning of the homopolymer on the positive strand by default, as their true position in the sequence is arbitrarily established.

We previously reported that error rates change significantly for every sequencing run (ref 6) and therefore, for every run we calculated an independent error model. Quality scores have been discarded as they have been previously shown to be less informative than an empirically derived run-specific error model (Druley TE et al. 2009). This finding is further supported by the lower performance of every other approach that we compared to SPLINTER that also integrated quality scores in the analysis (see Figure 3 and Supplementary Figure 3).

Since this approach does not take into account quality score information, in order to save computing time while preserving the same amount of information, we compressed the original SCARF file by keeping only unique reads sequence in it and adding a weight to each read counting the number of times a read with the same sequence appeared in the original file. This strategy generated files at the best 10% of the original size of the SCARF output, linearly reducing the alignment run-time of the same factor.

Error model calculation

A 2^{nd}-order error model was parameterized from a negative control sequence included in every sample, i.e. a DNA fragment consisting of a PCR product from the M13 vector. The negative control allows to estimate the likelihood of a sequencing error defined as the rate of observed mutations in the sequencing reads without variants being present in the analyzed DNA fragment. Briefly, for every base n and its context defined as the two preceding bases m and l, we calculate the likelihood of observing a substitution s, an insertion i or a deletion d where $l, m, n, s \; \varepsilon \; \{A, C, G, T, N\}$, $i \; \varepsilon \; \{Insertion_{A,C,G,T}\}$, and $d \; \varepsilon \; \{D\}$. For substitutions, we calculate $Pr(s|n, m, l, j, r)$ for each read base j and run r as the ratio between the number of observed read bases with base equal to s and the total number of observed read bases. Deletion error rates are calculated the same way as substitutions, where j in this case is assumed to be the read base number preceeding the deletion. Insertions are analyzed by selecting only reads that overlap consecutive loci n and o. $Pr(i|o \sim n, m, l, j, r)$ is therefore computed as the ratio between the reads that contain one or more inserted bases between n and o and the total number of reads overlapping n and o.

3 Structure of the SPLINTER algorithm

SPLINTER: IN/DEL and substitution detection using Large Deviation Theory

Since previously designed algorithms are unable to precisely call and quantify short IN/DELs in large pools, we designed and implemented SPLINTER (Short IN/DEL Prediction by Large deviation Inference and Non-linear True frequency Estimation by Recursion), a new algorithm based on *Sanov's theorem*, which is part of the information theoretic treatement of Large Deviation Theory (Cover T. and Thomas J.A 1991).

SPLINTER takes in input aligned sequencing reads. For every position i of the reference sequence, SPLINTER stores the counts of each observed base character $b = \{A, C, G, T, N, D\}$ as well as the counts for each inserted base stretch g between i and its consecutive position $i + 1$ of length c (maximum number of accepted edits).

SPLINTER assumes that sequencing reads are generated independently from one another and that read bases within the same read are incorporated indepedently from one another.

Substitution variants can be detected at a particular position i by estimating the distance of the empirical distribution P of observed nucleotides A,C,G,T,N from the expected distribution Q representing the expected frequency of nucleotides assuming that i does not harbor any variant present in the pool. Q is computed as a linear product between the error model matrix A for each read base j and sequencing run r and the true frequency vector $\tau_{i,null}$ under the null hypothesis that only the reference base is present as

$$Q_{j,r,s} = A_{i,j,r} * \tau_{i,null}$$

The distance between P and Q is computed independently for each read base j, sequencing run r and strand s as

$$Q_{j,r,s}^{n_{j,r,s}}(E) = 2^{-n_{j,r,s} D(P_{j,r,s} \| Q_{j,r,s})}$$

where $D(P_{j,d,s} \parallel Q_{j,d,s})$ is the *Kullback–Leibler* distance between P and Q. $Q_{j,r,s}^{n_{j,r,s}}(E)$ is a p-value calculated by testing the hypothesis that P was sampled from Q. Since j and r are independent, according to the initial assumptions, a cumulative p-value for each strand s is computed as

$$Q_{s}^{n_s}(E) = 2^{-\sum_r \sum_j n_{j,r,s} D(P_{j,r,s} \| Q_{j,r,s})}$$

Deletions are detected by estimating the distance of the fraction of observed deletions P_D from the fraction of expected deletions Q_D. P_D and Q_D are defined for each read base j, run r and strand s as

$$P_{D_{j,r,s}} = \left(\frac{d_{j,r,s}}{C_{j,r,s}}, 1 - \frac{d_{j,r,s}}{C_{j,r,s}} \right)$$

$$Q_{D_{j,r,s}} = \left(A_{D_{i,j,r,s}}, 1 - A_{D_{i,j,r,s}} \right)$$

where $d_{j,r,s}$ and $C_{j,r,s}$ represent the number of observed deletions and the total observed coverage and $A_{D_{i,j,r,s}}$ corresponds to the expected likelihood of observing a deletion at i for j, r and s given the error model matrix A.

The distance between $P_{D_{j,r,s}}$ and $Q_{D_{j,r,s}}$ is again computed independently as

$$Q^{n_{D_{j,r,s}}}_{D_{j,r,s}}(E) = 2^{-n_{D_{j,r,s}} D(P_{D_{j,r,s}} \| Q_{D_{j,r,s}})}$$

and the cumulative p-value for s is computed as

$$Q^{n_{D_s}}_{D_s}(E) = 2^{-\sum_r \sum_j n_{D_{j,r,s}} D(P_{D_{j,r,s}} \| Q_{D_{j,r,s}})}$$

Insertions are analyzed as stretches of nucleotides (g) of maximum length c located between the adjacent and consecutive positions i and $i+1$ and are detected by comparing the observed insertion distribution $P_{I_{i\sim i+1}}$ and the expected insertion distribution $Q_{I_{i\sim i+1}}$, defined as

$$P_{I_{i\sim i+1,j,r,s}} = \left(\frac{g_{i\sim i+1,j,r,s}}{C_{i\sim i+1,j,r,s}}, 1 - \frac{g_{i\sim i+1,j,r,s}}{C_{i\sim i+1,j,r,s}} \right)$$

$$Q_{I_{i\sim i+1,j,r,s}} = \left(A_{I_{i\sim i+1,j,r,s}}, 1 - A_{I_{i\sim i+1,j,r,s}} \right)$$

where $A_{I_{i\sim i+1,j,r,s}}$ corresponds to the expected likelihood of observing a insertion at $i \sim i+1$ for j, r and s given the error model matrix A. Calculation of the p-value for each strand s is performed as described for deletions.

Presence or absence of any given variant at position i is assessed by asking the p-values for both strands to be equal or less than a user-defined empirical cutoff α (where $\alpha <= 0.05$). We find that requiring both strands to pass α greatly increases the accuracy of our method, as previously observed (ref 4)

Frequency estimation of identified pool variants

For every identified pool variant, SPLINTER performs estimation of the true variant frequency vector τ_i at position i and/or the true insertion frequency vector $\tau_{i\sim i+1}$ between i and $i+1$. τ is fit by maximum likelihood

$$\tau = \arg\max_{\tau} 2^{-\sum_s \sum_r \sum_j n_{j,r,s} D(P_{j,r,s} \| Q_{j,r,s,\tau})} * Pr(\tau)$$

where we implicitly assume that $Pr(\tau)$, the prior distribution for τ, is uniform, leading to

$$\tau \approx \arg\min_{\tau} \sum_s \sum_r \sum_j n_{j,r,s} D(P_{j,r,s} \parallel Q_{j,r,s,\tau})$$

SPLINTER is significantly different from our previous pooled DNA SNP caller algorithm, SNPseeker (Druley TE et al. 2009) at various levels. First, it is able to detect indels in large pools by using new models and new integrated data structures, whereas SNPseeker can only detect substitutions. Secondly, SPLINTER is more sensitive and specific than SNPseeker as it integrates information of a positive control to define the optimal cutoff for the values of $Q^{n_s}_s(E)$,

$Q_{D_s}^{n_{D_s}}(E)$, $Q_{I_s}^{n_{I_s}}(E)$ (p-value cutoff) at every position i. Thirdly, SNPseeker implemented a non-linear least-square fit for estimating the true frequency vector τ (Druley TE et al. 2009), whereas SPLINTER uses a maximum likelihood method. We found that this leads to more accurate frequency estimates (data not shown) but it also allows incorporation of prior information.

4 Data analysis

Evaluation of sensitivity and specificity of variant calling and accuracy of frequency estimation
In order to determine the discriminatory power of our method, we calculated sensitivity and specificity in a p-value cutoff-independent way by iterating over a range of p-value cutoff values from 0 to -3000 with increments of -0.001 at each round. The optimal cutoff was determined as the value that minimized the Euclidean distance between the corresponding specificity and sensitivity (ranging from 0 to 1) to perfect specificity and sensitivity (1,1). This strategy was repeated by analyzing the data incorporating 12, 18, 21, 24 bases per read (cycles) and comparing sensitivity and specificity of the analysis, resulting in the definition of the optimal cutoff and incorporated read bases. This was done because we have previously demonstrated that the likelihood of sequencing errors increases for later cycles (Druley TE et al. 2009), and different error rates will affect the accuracy of discrimination between signal and noise. The optimal combination between cycles and cutoff was then used for data analysis. Accuracy of the frequency estimation was measured by calculating Pearson's correlation coefficient between the observedand estimated frequencies.

Monte Carlo sampling and calculation of Receiver Operating Characteristics Curves
In order to determine the relationship between p-value and coverage per base per strand for any given variant, we performed Monte Carlo sampling on aligned reads for a selected synthetic pool. We randomly sampled fractions equivalent to 0.005, 0.010, 0.015, 0.020, 0.025, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.75, 0.90 of the total number of aligned reads 100 times each and performed an analysis with SPLINTER on every sample. This allowed us to generate a distribution of p-values for each coverage point. For each set of 100 samples we calculated a Receiver Operating Characteristics (R.O.C.) curve. ROC curves plot a methods sensitivity (here, the fraction of mutant positions correctly identified) versus the false positive rate (the fraction of the bases without mutation that were incorrectly reported) for different p-value cutoffs. For each ROC curve we computed the corresponding Area Under the Curve (A.U.C.), and we used it as a metric for assessing the lowest value of coverage per base at which 100% specificity and 100% sensitivity (AUC equal to 1) are reached.

Comparison between different variant callers
In order to compare the performance of SPLINTER with that of other approaches, we applied SNPseeker (Druley TE et al. 2009), MAQ v0.7.1 (Li H et

al 2008), SAMtools (Li H et al. 2009) and VarScan (Koboldt DC et al. 2009) to the synthetic pool datasets. We separately compared the performance of each approach for detection of substitutions and indels given the fact that indels are not supported by SNPseeker and MAQ. Performance was evaluated by determining sensitivity (fraction of true positives identified by the method over total true positives in the set) and positive predicted value (fraction of true positives identified by the method over total positions identified by the method) and values were plotted and compared for each approach. For substitutions, Pools 4 and 5 were used in their entirety (renamed sub 1 and sub 2 in Figure 3) whereas only substitutions were considered for the pools simulating 100, 250 and 500 samples. For indels, Pools 1, 2 and 3 were used in their entirety (renamed indel 1, indel 2 and indel 3) whereas only indels were considered for the pools simulating 100,250 and 500 samples. SNPseeker was run as previously described and performance was computed after determining the optimal p-value cutoff and that maximized its performance. MAQ was run as described in (Li H et al 2008) with snp filtering after its execution in order to reduce the number of false positives. For SAMtools and VarScan, files were previously aligned using Novoalign at its default settings (www.novocraft.com), and SAM files were then converted into BAM and then pileup files. For SAMtools, variants were called from the pileup file, variants are unfiltered because when filtering was applied no hits were returned in output for any of the tested libraries. VarScan was run on the SAMtools pileup files and results were filtered by finding the optimal p-value cutoff that maximized its performance. We compared also the performance of CRISP (Bansal V 2010) by running the approach applied to all the pools using the default settings. We additionally compared the performance of the Genome Analysis Toolkit (GATK) framework on our set (McKenna A et al. 2010) using the suggested default parameters but we could not detect any of the true positives in any of the synthetic sets, so we decided not to include this analysis in the comparison. We believe that this result was due to the Unified Genotyper being optimized for single individual genotyping rather than large pools sequencing. Additionally, GATK is not able to detect indels in its current iteration (v1.0.3864).

References

- Bansal Vikas A statistical method for the detection of variants from next-generation resequencing of DNA pools Bioinformatics 26:318-324 (2010)
- Cover T. and Thomas J.A. Elements of Information Theory. Wiley Interscience (1991)
- Druley TE et al. Quantification of rare allelic variants from pooled genomic DNA. Nature Methods Apr;6:263-5 (2009)
- Durbin R, Eddy S, Krogh A, Mitchison G Biological Sequence Analysis. Cambridge University Press (1998)

- Higgins M et al. NHLBI Family Heart Study: objectives and design. American Journal of Epidemiology 143(12):1219-28 (1996)
- Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, Weinstock GM, Wilson RK, Ding L VarScan: variant detection in massively parallel sequencing of individual and pooled samples Bioinformatics Sep 1;25(17):2283-5 (2009)
- Li H, Ruan J, Durbin R Mapping short DNA sequencing reads and calling variants using mapping quality scores Genome Research Nov;18(11):1851-8 (2008)
- Li H et al The Sequence Alignment/Map format and SAMtools Bioinformatics Aug15;25(16):2078-9 (2009)
- McKenna A et al The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Research July;20:1297-1301 (2010)
- Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48(3):443-53 (1970)
- Smith TF, Waterman MS Identification of common molecular subsequences. Journal of Molecular Biology 147(1):195-7 (1981)
- Vendt BA et al. Silent Cerebral Infarct Transfusion (SIT) trial imaging core: application of novel imaging information technology for rapid and central review of MRI of the brain. Journal of Digital Imaging 22(3):326-43 (2008)