bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Bamgineer: Introduction of simulated allele-specific copy number variants into exome and targeted sequence data sets

Authors

Soroush Samadian[1], Jeff P. Bruce[1], Trevor J. Pugh [1,2]

1. Princess Margaret Cancer Centre, University Health Network, Toronto, Ontario, Canada
2. Department of Medical Biophysics, University of Toronto, Toronto, Ontario, Canada

Address correspondence to:
Trevor Pugh, PhD, FACMG
MaRS Centre, 101 College Street
Princess Margaret Cancer Research Tower, Room 9-305
Toronto, Ontario, Canada M5G 1L7
trevor.pugh@utoronto.ca
416-581-7689

Abstract

Somatic copy number variations (CNVs) play a crucial role in development of many human cancers. The broad availability of next-generation sequencing (NGS) data has enabled the development of algorithms to computationally infer CNV profiles from a variety of data types including exome and targeted sequence data; currently the most prevalent types of cancer genomics data. However, systemic evaluation and comparison of these tools remains challenging due to a lack of ground truth reference sets. To address this need, we have developed Bamgineer, a tool written in Python to introduce user-defined haplotype-phased allele-specific copy number events into an existing Binary Alignment Mapping (BAM) file, with a focus on targeted and exome sequencing experiments. As input, this tool requires a read alignment file (BAM format), lists of non-overlapping genome coordinates for introduction of gains and losses (bed file), and an optional file defining known haplotypes (vcf format). To improve runtime performance, Bamgineer introduces the desired CNVs in parallel using queuing and parallel processing on a local machine or on a high-performance computing cluster. As proof-of-principle, we applied Bamgineer to a single high-coverage (mean: 220X) exome sequence file from a blood sample to simulate copy number profiles of 3 exemplar tumours from each of 10 tumour types at 5 tumour cellularity levels (20-100%, 150 BAM files in total). In addition to these reference sets, we expect Bamgineer to be of use for systematic benchmarking of CNV calling algorithms using their own data and expected tumour content for a variety of applications. The source code and reference datasets are freely available at http://github.org/pughlab/bamgineer.

Introduction

The emergence and maturation of next-generation sequencing technologies, including whole genome sequencing, whole exome sequencing, and targeted sequencing approaches, have enabled researchers to perform increasingly more

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

complex analysis of copy number variants (CNVs)[1]. While genome sequencing-based methods have long been used for CNV detection, these methods can be confounded when appled to exome and targeted sequencing data due to non-contiguous and highly-variable nature of coverage and other biases introduced during enrichment of target regions[1–5]. In cancer, this analysis is further challenged by bulk tumour samples that often yield nucleic acids of variable quality and are composed of a mixture of cell-types, including normal stromal cells, infiltrating immune cells, and subclonal cancer cell populations. Therefore, development of CNV calling methods on arbitrary sets of tumour-derived data from public repositories may not reflect the type of tumour specimens encountered at an individual centre, particularly formalin-fixed-paraffin embedded tissues routinely profiled for diagnostic testing.

Due to lack of a ground truth for validating CNV callers, many studies have used simulation studies to model tumour data[6]. Most often, simulation studies are used in an *ad-hoc* manner using customized formats to validate specific tools and settings with limited adaptability to other tools. More generalizable approaches aim at the *de novo* generation of sequencing reads according to a reference genome (e.g. Art-illumina[7], wessim[1] and dwgsim[8]). However, *de novo* simulated reads do not necessarily capture subtle features of empiric data, such as read coverage distribution, read insert size, quality scores, error rates, strand bias and GC content[6]; factors that can be more variable for exome and targeted sequencing data particularly when derived from clinical specimens. Recently, Ewing et al. developed a tool, BAMSurgeon, to introduce synthetic mutations into existing reads in a Binary alignment Mapping (BAM) file[9,10]. BAMSurgeon provides support for adjusting variant allele fractions (VAF) of engineered mutations based on prior knowledge of overlapping CNVs but does not currently support direct simulation of CNVs themselves.

Here we introduce Bamgineer, a tool to modify existing BAM files to precisely model allele-specific and haplotype-phased CNVs. This is done by introducing new read pairs sampled from existing reads, thereby retaining biases of the original data such as local coverage and strand bias. As input, Bamgineer requires a BAM file and two lists of non-overlapping genomic coordinates to introduce allele-specific gains and losses. The user may explicitly provide known haplotypes or chose to use the BEAGLE[11] phasing module that we have incorporated within Bamgineer. We implemented parallelization of the Bamgineer algorithm for both standalone and high performance computing cluster environments, significantly improving the scalability of the algorithm.

As proof of principle, we generated, from a single high coverage (mean : 220X) BAM file derived from a human blood sample, a series of 30 new BAM files containing a total of 1,693 simulated copy number variants (on average, 56 CNVs comprising 1800Mb i.e. ~55% of the genome per tumour) corresponding to profiles from exemplar tumours for each of 10 cancer types. To demonstrate quantitative introduction of CNVs, we further simulated 4 levels of tumour cellularity (20, 40, 60, 80% purity) resulting in an additional 120 new tumour BAM files. We validated our

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

approach by comparing CNV calls and inferred purity values generated by an allele-specific CNV-caller (Sequenza[12]) as well as a focused comparison of allelic variant ratios, haplotype-phasing consistency, and tumour/normal log2 ratios for inferred CNV segments (Figure S5-S7). In every case, inferred purity values were within ±5% of the targeted purity; and majority of engineered CNV regions were correctly called by Sequenza (accuracy > 94%; Figure S5-S7). Allele variant ratios were also consistent with the expected values both for targeted and the other haplotypes (Median within ±3% of expected value). Median tumour/normal log2 ratios were within ±5% of the expected values.

Overall, Bamgineer gives investigators complete control to introduce CNVs of arbitrary size, magnitude, and haplotype into an existing reference BAM file. We have uploaded all software code to a public repository (https://github.org/pughlab/bamgineer).

Methods

Bamgineer uses several python packages (Supplemental Material) for parsing input files (*pyVCF*[13], *VCFtools*[14] , and *pybedtools*[15]), manipulating BAM files (*pysam*[16] , *Samtools*[17], *BamUtil*[18], *Sambamba*[19]), phasing haplotypes (BEAGLE[11]) , and distributing compute jobs in cluster environments (*ruffus*[20]). HaplotypeCaller[21] from the Genome Analysis Toolkit (GATK) is used to call germline heterozygous SNPs (*het.vcf*) if known haplotype SNP data is not provided. The analysis workflow is outlined in Figure 1.

Inputs

The user provides 2 mandatory inputs to Bamgineer as command-line arguments: 1) a BAM file containing aligned NGS reads ("*Normal.bam"*), 2) a BED file containing the genome coordinates and type of CNV (e.g. allele-specific gain) to introduce ("CNV regions.bed"). Bamgineer can be used to add four broad categories of CNVs: Balanced Copy Number Gain (BCNG), Allele-specific Copy Number Gain (ASCNG), Allele-specific Copy Number Loss (ACNL), and Homozygous Deletion (HD). For example consider a genotype AB at a genomic locus where A represents the major and B represents the minor allele. Bamgineer can be applied to convert that genomic locus to any of the following copy number states:

{ A, B, ABB, AAB, ABB, AABB, AAAB, ABBB}

An optional VCF file containing phased germline calls can be provided (*phased_het.vcf*). If this file is not provided, Bamgineer will call germline heterozygous single nucleotide polymorphisms (SNPs) using the GATK HaplotypeCaller[21] and then categorize alleles likely to be co-located on the same haplotypes using BEAGLE[11] and population reference data from the HapMap[22] project.

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Isolation of source reads to construct haplotype-specific CNVs

To obtain paired-reads in CNV regions of interest, we first intersect *Normal.bam* with the targeted regions overlapping user-defined CNV regions (*roi.bed*). This operation generates a new BAM file (*roi.bam*). Subsequently, depending on whether the CNV event is a gain or loss, the algorithms performs two separate steps as follows:

I. CNV Gain Implementation

Creation of new read pairs simulating copy gains

To introduce copy number gains, Bamgineer creates new read-pairs constructed from existing reads within each region of interest. This approach thereby avoids introducing pairs that many tools would flag as molecular duplicates due to read 1 and read 2 having start and end positions identical to an existing pair. For each new read-pair, we require that each the new insert size is within 3 standard deviation (SD) of the overall distribution for the original read pairs in the region of interest. The newly created read- pairs are provided unique read names to avoid confusion with the original input BAM file. To enable inspection of these reads, these newly created read pairs are stored in a new BAM file, *gain_re_paired_renamed.bam,* prior to merging into the final engineered BAM. Since we only consider high quality reads (i.e. properly paired reads, primary alignments and mapping quality > 30), the newly created BAM file contains fewer reads compared to the input file (~90-95% in our proof-of-principle experiment). As such, at every transition we log the ratio between number of reads between the input and output files.

Introduction of mutations according to haplotype state

To ensure newly constructed read-pairs match the desired haplotype, we alter the base at heterozygous SNP locations (*phased_het.vcf*) within each read according to haplotype provided by the user or inferred using the BEAGLE algorithm. To achieve this, we iterate through the set of re-paired reads used to increase coverage (*gain_re_paired_renamed.bam*) and modify bases overlapping SNPs corresponding to the target haplotype (*phased_het.vcf*). We then write these reads to a new BAM file (*gain_re_paired_renamed_mutated.bam*) prior to merging into the final engineered BAM. (Figure 1).

As an illustrative example consider two heterozygous SNPs , *AB* and *CD* both with allele frequencies of ~0.5 in the original BAM file (i.e. approximately half of the reads supporting reference bases and the other half supporting alternate bases. To introduce a 2-copy gain of a single haplotype, reads to be introduced must match the desired haplotype rather than the two haplotypes found in the original data. If heterozygous *AB* and *CD* are both located on a haplotype comprised of alternative alleles, at the end of this step, 100% of the newly repaired reads will support alternate base-pairs (e.g. BB and DD). Based on the haplotype structure provided, other haplotype combinations are possible including AA/DD, BB/CC, etc. Figure 2C-D summerize the steps involved in the process.

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Sampling of reads to reflect desired allele fraction:

Depending on the absolute copy number desired for the for CNV gain regions, we sample the BAM files according to the desired copy number state. We define conversion coefficient as the ratio of total reads in the created BAM from previous step (gain_re_paired_mutated.bam) to the total reads extracted from original input file (roi.bam):

$$\rho = \frac{\text{no. of reads in } gain\_re\_paired\_mutated.bam}{\text{no. of reads in } roi.bam}$$

According to the maximum number of absolute copy number (ACN) for simulated CNV gain regions (defined by the user), two scenarios are conceivable as follows:

Single copy gain (ACNG = 3)

To achieve the single copy gain (ACN =3, e.g. ABB copy state), the file in the previous step (*gain_re_paired_renamed_mutated.bam*), should be sub-sampled such that on average depth of coverage is half that of extracted reads from the target regions from the original input normal file(*roi.bam*). Thus, the final sampling rate is calculated by dividing 0.5 by ρ (adjusted sampling rate is usually 0.51-0.59 ; $0.85 < \rho < 1$) and the new reads are written to a new BAM file (*gain_re_paired_renamed_mutated_sampled.bam)* that we then merge with the original reads (*roi.bam*) to obtain *gain_final.bam* (Figure 2C).

Double copy gain (ACNG = 4)

To achieve a 2-copy gain (ACN =4, e.g. AABB copy state), the average depth of coverage for input file *roi.bam* and sampled BAM file, should be the equal. However since $\rho < 1$, we merge the engineered reads (*gain_re_paired_renamed_mutated.bam*) with a subsample of the original input *reads (roi.bam)* corresponding to ρ reads, to generate *gain_final.bam* (Figure 2D).

II. CNV Loss Implementation

To introduce CNV losses, Bamgineer removes reads from the original bam corresponding to a specific haplotype and does not create new read pairs from existing ones. To diminish coverage in regions of simulated copy number loss, we sub-sample the BAM files according to the desired copy number state and write these to a new file. The conversion cofficient is defined similarly as the number of reads in *loss_mutated.bam* divided by number of reads in *roi_loss.bam* (> ~0.98). Similar to CNV gains, the sampling rate is adjusted such that after the sampling, the average depth of coverage is half that of extracted reads from the target regions (calculated by dividing 0.5 by conversion ratio, as the absolute copy number is 1 for loss regions). Finally, we subtract the reads in CNV loss BAMs from the *input.bam* (or *input_sampled.bam*) and merge the results with CNV gain BAM (gain_final.bam) to obtain, the final output BAM file harbouring the desired copy number events(Figure 2A).

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Results

Proof-of-principle experiments using whole exome sequence data

For all proof-of-principle experiments we used whole exome sequencing data from a single normal (peripheral blood lymphocyte) DNA sample. DNA was captured using the Agilent SureSelect Exome v5+UTR kit and sequenced to 220X median coverage as part of a study of neuroendocrine tumours[23]. Reads were generated aligned to the hg19 build of the human genome reference sequence and processed using the Genome Analysis Toolkit (GATK) Best Practices pipeline.

Arm-level and chromosome-level copy number alteration

To verify the ability of our workflow to introduce copy gains, we first generated single-copy, allele-specific copy number amplification (ASCNA) of chromosomes 21 and 22, followed by calculation of hybrid-selection metrics using the Picard suite[24]. As expected, the number of read pairs increased by 50% with no statistically significant impact on insert size distribution metric including: median, mean, and standard deviation of insert size for paired-reads (Table S1). The distribution of paired reads distances was nearly indistinguishable before and after the addition of ASCNA (median 211.97 vs. 212.3, Two-sided KS-test p>0.99, Figure S2). We subsequently applied our whole exome analysis pipeline for detecting CNVs using Mpileup[25] , Varscan2[26] and Sequenza[12] to infer allele-specific copy number profiles (Figure 3). This analysis detected the desired arm- and chromosome-level gains at the expected depth ratio of 1.5 and we verified specific gain of desired haplotypes by confirming that 97-98% of the reads contained variants corresponding to the target haplotype (mean variant allele frequency of SNPs on amplified haplotypes = 0.66 ± 0.03 and 0.33 versus 0.50 ± 0.03 on non-target haplotypes; Figure 3A&B).

We next repeated this experiment to introduce a single copy loss of these two regions. Again, the number of read pairs was consistent with the desired copy state (decrease of 50% of reads in the target regions). We also verified that the single copy deletion was restricted to a single haplotype with > 99% of reads containing variants corresponding to the target haplotype.

Synthetic tumour-normal mixtures of exemplar tumours from The Cancer Genome Atlas

Following the validation of our tool for readily-detected chromosome- and arm-level events, we next used Bamgineer to simulate CNV profiles mimicking 3 exemplar tumours from each of 10 different cancer types profiled by The Cancer Genome Atlas using the Affymetrix SNP6 microarray platform: lung adenocarcinoma (LUAD); lung squamous cell (LUSC); head and neck squamous cell carcinoma (HNSC); glioblastoma multiformae (GBM); kidney renal cell carcinoma (KIRC); bladder (BLCA); colorectal (CRC); uterine cervix (UCEC); ovarian (OV), and breast (BRCA) cancers (Table 1).

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

To select 3 exemplar tumours for each cancer type, we chose profiles that best represented the copy number landscape for each cancer type. First, we addressed over-segmentation of the CNV calls from the microarray data by merging segments of <500 kb in size with the closest adjacent segment and removing the smaller event from the overlapping gain and loss regions. We then assigned a score to each tumour that reflects its similarity to other tumour of the same cancer type (Figure 4). This score integrates total number of CNV gain and losses (Table 1), median size of each gain and loss, and the overlap of CNV regions with GISTIC peaks for each cancer type as reported by The Cancer Genome Atlas (Supplementary Material). Subsequently, we selected three high ranking tumours for each cancer type such that, together, all significant GISTIC[26] peaks for that tumour type were represented.

For each of the 30 selected tumour profiles (3 for each of 10 cancer types), we introduced the corresponding CNVs at 5 levels of tumour cellularity (20, 40, 60, 80, and 100%) resulting in 150 BAM files in total. For each BAM file, we used Sequenza to generate allele-specific copy number calls as done previously (Figure 6). A representative profile from a single tumour is shown in Figure 3C. From this large set of tumours, we next set out to compare Picard metrics and CNV calls as we did for the arm- and chromosome-level pilot.

For each of the 30 selected tumour profiles (3 for each of 10 cancer types), we introduced the corresponding CNVs at 5 levels of tumour cellularity (20, 40, 60, 80, and 100%) resulting in 150 BAM files in total. For each BAM file, we used Sequenza to generate allele-specific copy number calls as done previously (Figure 6). A representative profile from a single tumour is shown in Figure 3C. From this large set of tumours, we next set out to compare Picard metrics and CNV calls as we did for the arm- and chromosome-level pilot.

Performance evaluation

We subsequently evaluated Bamgineer using several metrics: tumour allelic ratio, SNP phasing consistency, and tumour to normal log2 ratios (Figure 7). As expected, across all regions of a single copy gain, tumour allelic ratio was at ~0.66 (interquartile range: 0.62 – 0.7) for the targeted haplotype and 0.33 (interquartile range: 0.3-0.36) for the other haplotype. As purity was decreased, we observed a corresponding decrease in allelic ratios, from 0.66 down to 0.54 (interquartile range: 0.5- 0.57) for targeted and an increase (from 0.33) to 0.47 (interquartile range: 0.43-0.5) for the other haplotype for 20% purity (Figure 5A,B). This decrease (increase) correlated directly with decreasing purity (R^2 > 0.99) for both haplotypes.

Similarly, for single copy loss regions, as purity was decreased from 100% to 20% the allelic ratio linearly decreased (R^2 > 0.99) from ~0.99 (interquartile range: 0.98-1.0) for targeted haplotype to ~0.55 (interquartile range: 0.51-0.58) for targeted haplotype and increases from 0 to ~0.43 (interquartile range: 0.4-0.46) for the other haplotype (Figure 5B).

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

The results for log2 tumour to normal depth ratios of segments normalized for average ploidy are also consistent with the expected values (equation 2, Supplementary Materials). For CNV gain regions, log2 ratio decreased from ~0.58(log2 3) to ~0.13 as purity was decreased from 100% to 20%. For CNV loss regions, as purity was decreased from 100% to 20%, log2 ratio was increased from ~ -1 (log2 0.5) to ~-0.15, consistent with equation 2, Supplementary Materials (Figure 6C; S5-S8 for individual cancers)

Ultimately, we wanted to assess whether Bamgineer was introducing callable CNVs consistent with segments corresponding to the exemplar tumour set. To assess this, we calculated an accuracy metric (Figure 6D) as:

$$accuracy = \frac{TP+TF}{TP+TF+FP+FN}$$

where TP, TF, FP and FN represent number of calls from Sequenza corresponding to true positives (perfect matches to desired CNVs), true negatives (regions without CNVs introduced), false positives (CNV calls outside of target regions) and false negatives (target regions without CNVs called). TP, TF, TN, FN were calculated by comparing Sequenza absolute copy number (predicted) to the target regions for introduction of CNVs 1 Mb bins across the genome (Supplementary materials). As tumour content decreased, accuracy for both gains and losses decreased as false negatives became increasingly prevalent due to small shifts in log2 ratios. We note that (as expected), overall decreasing cancer purity from 100% to 20% generally decreases the segmentation accuracy. Additionally, we observe that segmentation accuracy, is on average, significantly higher for gain regions compared to the loss regions for tumour purity levels below 40%(Figure 6D). This is consistent with previous studies that show the sensitivity of CNV detection from sequencing data is slightly higher for CNV gains compared to CNV losses[28]. We also note that with decreasing cancer purity, the decline in segmentation accuracy follows a linear pattern of decline for gain regions and an abrupt stepwise decline for loss regions (Fig 6D; segmentation accuracies are approximately similar for 40% and 20% tumour purities).

Finally, we observed some degree of variation in terms of segmentation accuracy across individual cancer types (Figure S5-S8). For instance segmentation accuracy is considerably lower for LUAD, OV and UCEC in comparison with other simulated cancers for this study. The relative decline in performance is seen in cancer types where there are a large number of CNV gains and losses covering a sizeable porion of the genome; and hence, the original loss and gain events sampled from TCGA had a significant overlaps. As a result, after resolving overlapping gain and loss regions, on average, the final target regions constitute a larger number of loss regions immediately followed by gain regions and vice versa; making the accurate segmentation challenging for the CBS (circular binary segmentation) algorithm

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

implemented by. These are precise issues that may be addressed by further improvement of copy number caller using CNVs simulated by Bamgineer.

In summary, application of an allele-specific caller to BAMs generated by Bamgineer recapitulates CNV segments consistent with >95%(medians: 95.1 for losses and 97.2 for gains) of those input to the algorithm. However, we note some discrepancies between the expected and called events, primarily due to small size of CNV as well as large segments of unprobed genome in exome data.

Runtime and parallelization

Bamgineer is computationally intensive and the runtime of the program is dictated by the average coverage of the input BAM file and the genomic footprint of target regions (corresponding to the number of reads that must be processed). To ameliorate the computational intensiveness of the algorithm, we employed a parallelized computing framework to maximize use of a high performance compute cluster environment when available. We took advantage of two features in designing the parallelization module. First, we required that added CNVs are independent for each chromosome (although nested events can likely be engineered through serial application of Bamgineer). Second, since we did not model interchromosomal CNV events, each chromosome can be procecced independently. As such, CNV regions for each chromosomes can be processed in parralel and aggregated as a final step. Figure S3 shows the runtimes results for The Cancer Genome Atlas experiments. Using a single node with 12 cores and 256GB of RAM, each syntetic BAM took less than 3.5 hours to generate. We also developed a version of Bamgineer that can be launched from grid engine cluster environments. It uses python pipeline management package *ruffus* to parallelize tasks automatically and log runtime events. It is highly modular and easily updatable. If disrupted during a run, the pipeline can continue to completion without re-running previously completed intermediate steps.

Limitations of current implementation and potential extensions

The work presented herein can be extended in several directions. First, Bamgineer is not able to reliably perform interchromosomal operations such as chromosomal translocation, as our focus has been on discrete regions probed by exome and targeted panels. Additionally, in our current implementation, we limited the maximum total copy number to 4. Certainly, higher-level amplifications occur in cancer and interative application of Bamgineer may enable introduction of copy number states beyond four chromosomal copies as well as complex, nested events. While these are challenging events to model, we appreciate that this class of structural variant play an important driving role in cancer; for example, EGFR vIII variants observed in brain cancer[27]. We expect this approach to be applicable beyond exome data and to be of use to tune algorithms for detection of subtle CNV signals such as somatic mosaicism or circulating tumour DNA. As these subtle shifts are beyond the sensitivity of many CNV callers, we expect our tool to be of value for

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

the development of new methods capable of detecting such events trained on conventional DNA sequencing data. Finally, introduction of compound, serially acquired CNVs may be of interest to model subclonal phylogeny developed over time in bulk tumour tissue samples (e.g. ASCNA followed by ASCND can result in a different copy number state than ASCND followed by ASCNA). Using our framework iteratively and combining it with tools used to introduce SNVs (such as Bamsurgeon), we plan to create tumour phylogenies to model the clonal structure in tumour sub-populations (Figure S8).

Conclusion

Bamgineer is a bioinformatics tool that can be used to add or delete haplotype-phased and allele-specific copy number events to existing alignments of targeted next-generation sequencing data. We demonstrated the utility of our tool by generating, from a single bam file derived from non-cancerous cells, 150 CNV profiles corresponding to 3 exemplar tumours derived from 10 cancer types at 5 tumour purity levels. With this synthetic data set and the ability to create customized data-sets of their own, Bamgineer could be a valuable tool to aid in development and benchmarking of CNV calling and other sequence data analysis tools and pipelines.

References

1. Sathirapongsasuti, J. F. *et al.* Exome Sequencing-Based Copy-Number Variation and Loss of Heterozygosity Detection: ExomeCNV. *Bioinformatics* btr462 (2011). doi:10.1093/bioinformatics/btr462
2. Chiang, D. Y. *et al.* High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat. Methods* **6,** 99–103 (2009).
3. Kim, S., Jeong, K. & Bafna, V. Wessim: a whole-exome sequencing simulator based on in silico exome capture. *Bioinformatics* **29,** 1076–1077 (2013).
4. Stankiewicz, P. & Lupski, J. R. Structural variation in the human genome and its role in disease. *Annu. Rev. Med.* **61,** 437–455 (2010).
5. Tan, R. *et al.* An evaluation of copy number variation detection tools from whole-exome sequencing data. *Hum. Mutat.* **35,** 899–907 (2014).
6. Escalona, M., Rocha, S. & Posada, D. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nat Rev Genet* **advance online publication,** (2016).
7. Huang, W., Li, L., Myers, J. R. & Marth, G. T. ART: a next-generation sequencing read simulator. *Bioinformatics* **28,** 593–594 (2012).
8. dwgsim. Available at: https://github.com/nh13/DWGSIM.
9. Ewing, A. D. *et al.* Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat Meth* **12,** 623–630 (2015).
10. adamewing/bamsurgeon. *GitHub* Available at: https://github.com/adamewing/bamsurgeon. (Accessed: 9th July 2016)

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

11. Browning, S. R. & Browning, B. L. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am. J. Hum. Genet.* **81,** 1084–1097 (2007).
12. Favero, F. *et al.* Sequenza: allele-specific copy number and mutation profiles from tumor sequencing data. *Ann. Oncol.* **26,** 64–70 (2015).
13. *PyVCF* Available at: https://github.com/jamescasbon/PyVCF.
14. VCFtools. Available at: https://vcftools.github.io/index.html.
15. Dale, R. K., Pedersen, B. S. & Quinlan, A. R. Pybedtools: a flexible Python library for manipulating genomic datasets and annotations. *Bioinformatics* **27,** 3423–3424 (2011).
16. pysam. Available at: https://github.com/pysam-developers/pysam.
17. Samtools. Available at: http://www.htslib.org/.
18. BamUtil. Available at: https://github.com/statgen/bamUtil.
19. Available at: http://lomereiter.github.io/sambamba/.
20. Goodstadt, L. Ruffus: A Lightweight Python Library for Computational Pipelines. *Bioinformatics* btq524 (2010). doi:10.1093/bioinformatics/btq524
21. Available at: https://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_gatk_tools_walkers_haplotypecaller_HaplotypeCaller.php.
22. International HapMap Consortium. The International HapMap Project. *Nature* **426,** 789–796 (2003).
23. Quevedo, R. Near-global copy-neutral loss of heterozygosity is a defining feature of pancreatic neuroendocrine tumours. *Cancer Discov* (2017).
24. Picard. *Picard* Available at: http://broadinstitute.github.io/picard.
25. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).
26. Koboldt, D. C. *et al.* VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.* **22,** 568–576 (2012).
27. Mermel, C. H. *et al.* GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biology* **12,** R41 (2011).
28. Wang, H., Nettleton, D. & Ying, K. Copy number variation detection using next generation sequencing read counts. *BMC Bioinformatics* **15,** 109 (2014).
29. Montano, N. *et al.* Expression of EGFRvIII in Glioblastoma: Prognostic Significance Revisited. *Neoplasia* **13,** 1113–1121 (2011).

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Figures

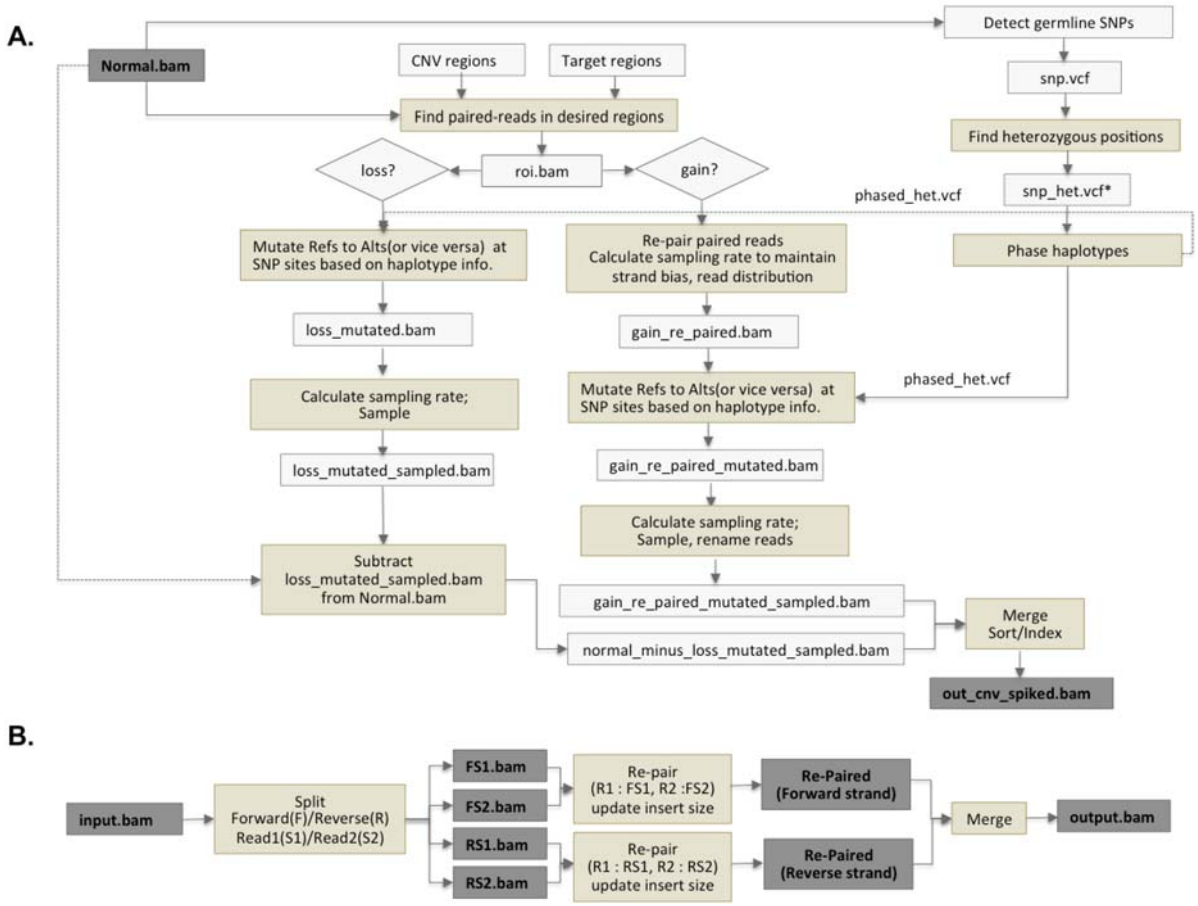

Figure 1: **A**. Overall architecture of Bamgineer for editing an existing BAM file to add and delete the user defined CNV event. The input and output files are shown in dark grey. The modules are shown darker relative to the files generated at each step. **B**. Creating new paired-reads from existing ones. The algorithms splits the input BAM file into four separate BAM files according to DNA strand and read information. Bamgineer will then iterate through split reads (read1 and read2) from each strand separately , pairing one read from read1 splits to another read from read2 split. The insert-size (tlen) in the newly paired read is then calculated and updated

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

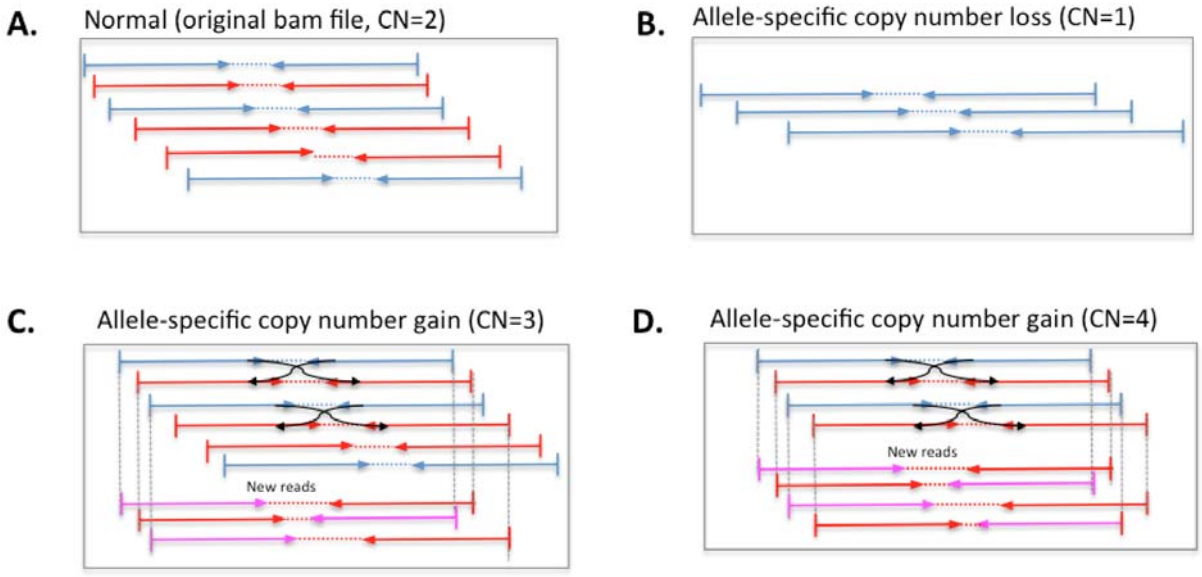

Figure 2: Haploype-specific CNVs simulated using repaired reads. Red and blue colors represent read-pairs corresponding to different haplotypes. **A**. Original BAM file used as the input. **B**. Allele-specific loss, reads matching a target haplotype are removed **C**. Allele-specific gain for CN=3; new read pairs are constructed from existing reads and reads are modified at SNP loci to match the desired haplotyope (magenta). **D.** Allele-specific gain for CN=4; in addition to introduction of new read pairs, the original reads from the original BAM file are downsampled to ensure the desired ABBB ratio.

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

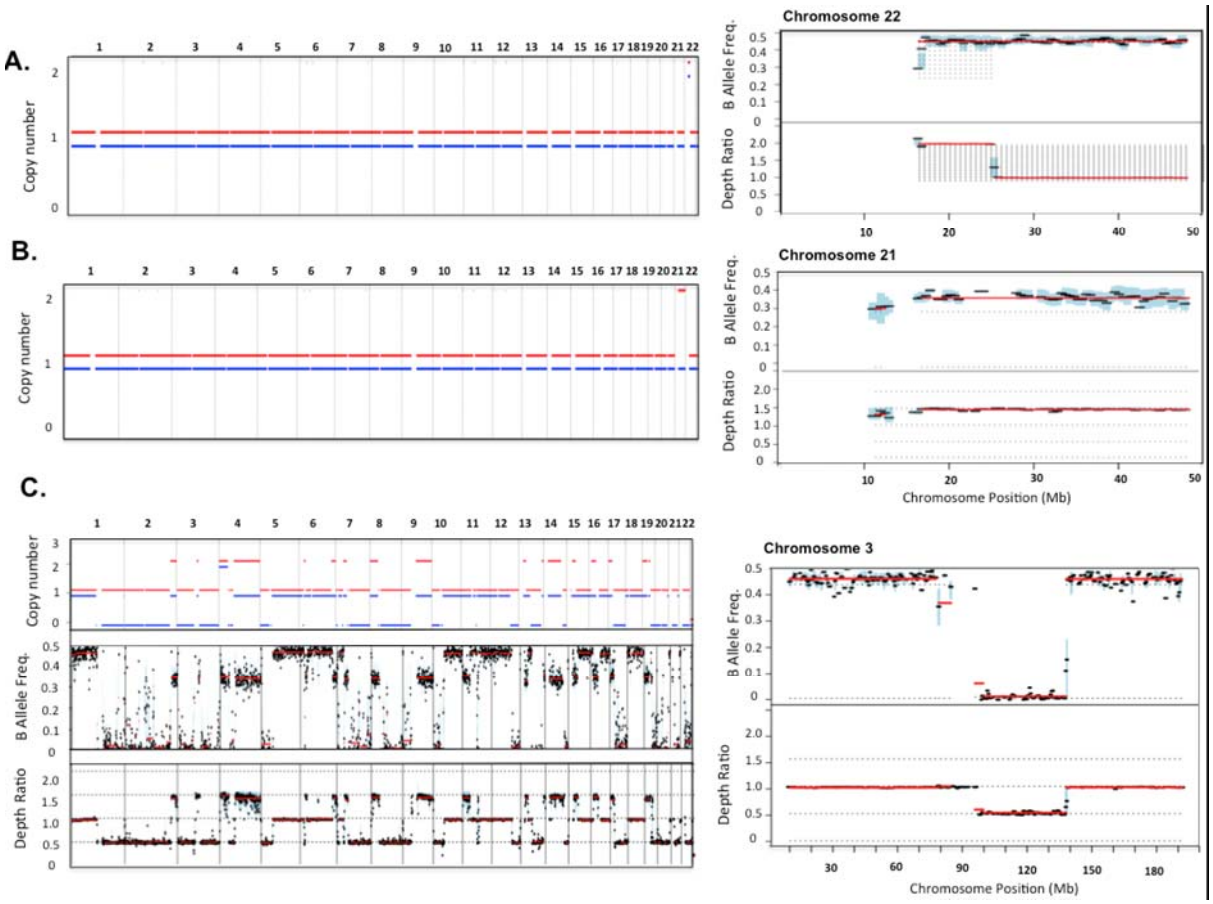

Figure 3. Panel A: Genome-wide (right) and chromosome-view (left) of allele specific copy number, BAF and depth ratios for balanced gain of p-arm in chromosome 22. Blue and red lines show allele specific copy number profiles for each chromosome (lines are offset from discrete copy number values by ± 0.1 for visual separation of the two alleles) .The small blue and red spots on the top figure show a balanced gain on p-arm of chromosome 22 (BAF is not affected as a result of balanced gain). Each black dot on the right figures represents a genomic locus and the red lines indicate the inferred value for consecutive segments. Panel B: Allele-specific gain of entire chromosome 21. As shown only one copy of the chromosome is gained and hence the allele frequency is reduced from the 0.5 to ~0.33. Panel C. Genome-wide (right) and chromosome-view (left) for 36 events (21 gains and 25 losses) sampled from Genome Atlas for Bladder Urothelial Carcinoma (BLCA) for 100% tumour content. As expected depth ratio and BAFs are approximately 0.5 and zero respectively.

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

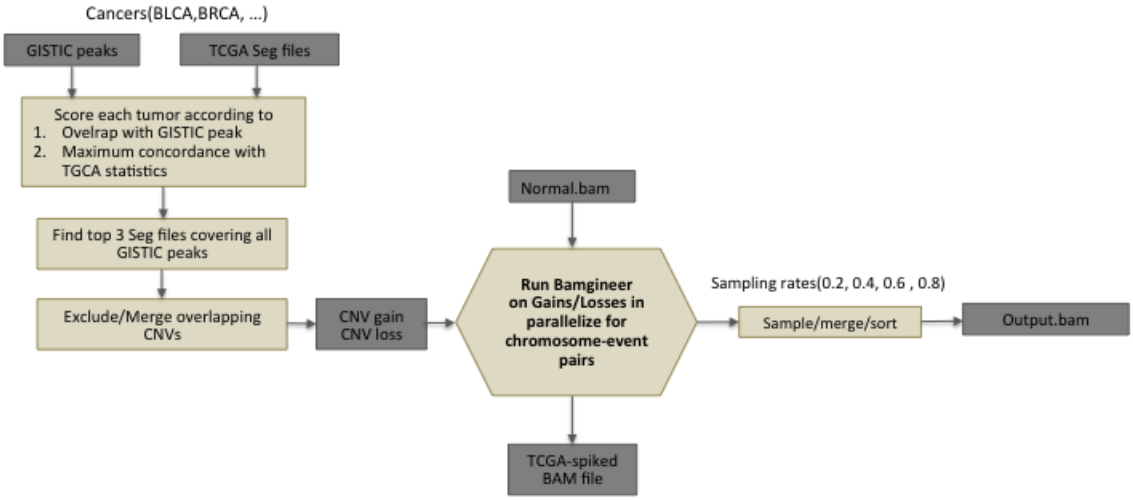

Figure 4. Overview of the design used to introduce cancer-specific CNV events. Parallelization module enables to simultaneously implement cancer-based, chromosome-based and event-based engineering of CNVs, significantly improving the performance (see “Runtime benchmarks and parallelization”)

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

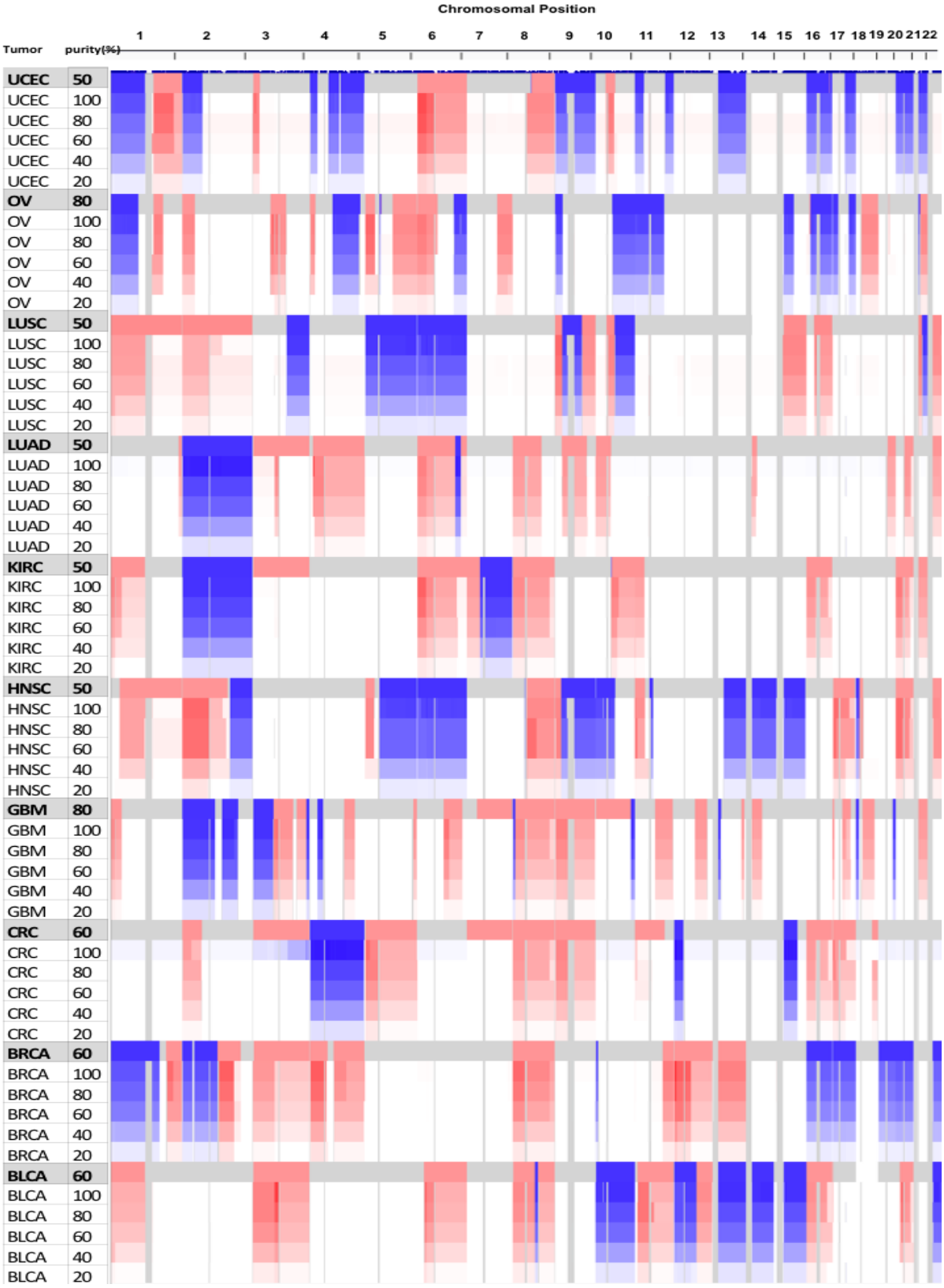

Figure 5. Log2 ratios of copy number segments inferred using the Sequenza algorithm, shown as a heatmap (blue: loss, red: gain; data range is -1.5 to 1.5) for different cancer types (separated by dashed lines) and different tumour cellularities.

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

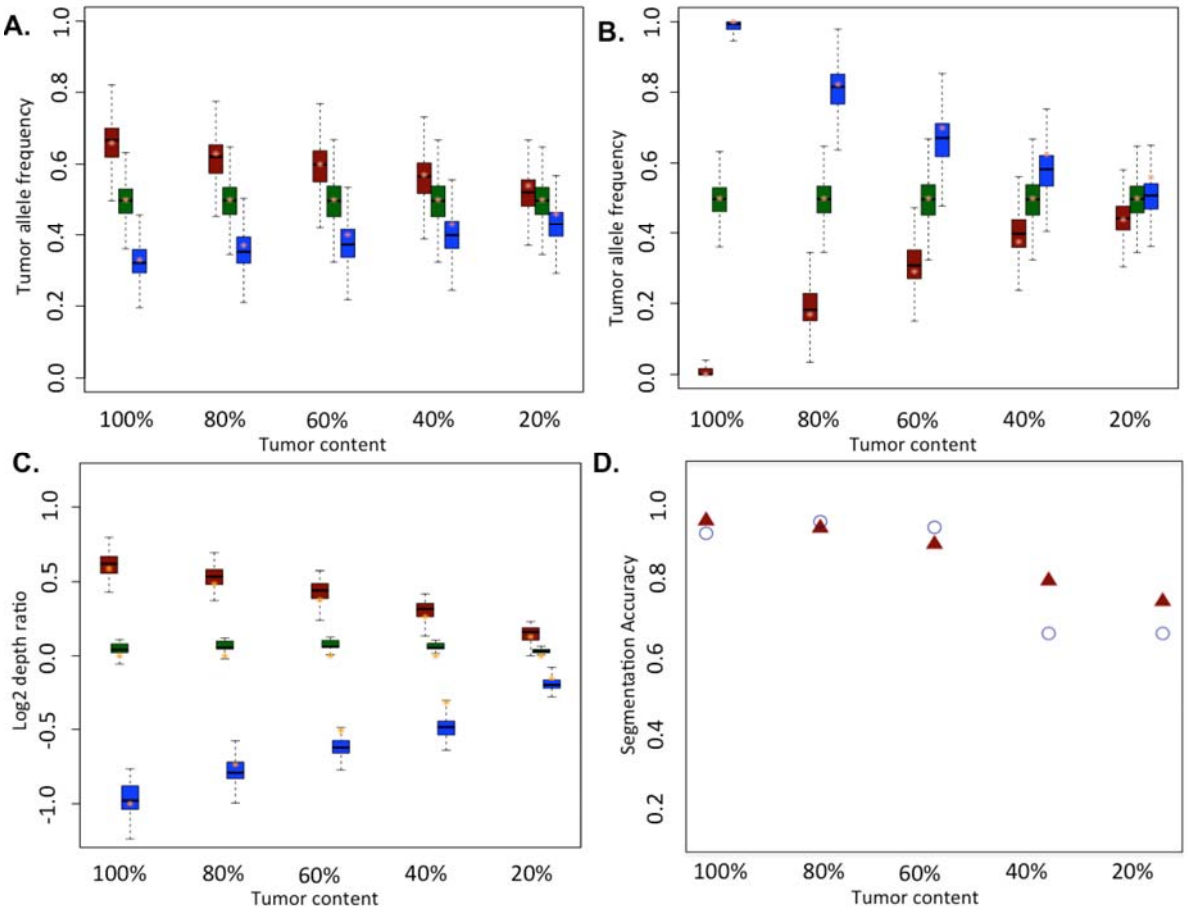

Figure 6: Allelic ratio for allele-specific copy number gain (Panel **A**) and loss (Panel **B**) events at heterozygous SNP loci for haplotypes affected (blue), haplotypes not affected (red), and SNPs not in engineered CNV regions (green) as negative controls at different tumour cellularity levels (x-axis) across all cancers. Panel **C**: Tumour to normal log2 depth ratio boxplots of copy number gain (red) and loss (blue) segments from Sequenza across all cancers (Table 1). Panel **D**: The segmentation accuracy for gain (red) and loss (blue) events across all cancers at different cellularity levels

bioRxiv preprint doi: https://doi.org/10.1101/119636; this version posted March 22, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC 4.0 International license.

Tables

Table 1. Patterns of different CNVs across multiple cancer types from TCGA (light grey), together with the number of CNV gains and losses and percentage of genome modified (GM%) by CNVs for each sample

TCGA			Sampled tumours								
Cancer	Avg. no. Amp	Avg. no. Del	Tumour1			Tumour2			Tumour3		
			Amp	Del	GM%	Amp	Del	GM%	Amp	Del	GM%
BLCA	23	30	23	30	70%	22	25	72%	22	25	71%
BRCA	26	36	24	34	59%	27	33	67%	21	30	66%
CRC	24	42	27	48	27%	23	29	70%	22	30	69%
GBM	26	36	23	34	51%	19	33	73%	30	43	57%
HNSC	24	40	28	40	61%	24	30	55%	24	28	62%
KIRC	6	16	6	16	75%	7	15	57%	8	14	66%
LUAD	26	34	26	25	55%	26	30	63%	22	32	67%
LUSC	25	43	34	47	26%	31	34	68%	24	31	73%
UCEC	39	47	39	43	62%	38	38	59%	37	37	66%
OV	29	37	26	35	68%	28	34	74%	28	31	65%