# Chapter 1: Introduction

## 1.1 Preliminaries

For most of the history of DNA sequencing, expense and difficulty rendered whole-genome sequencing and assembly attainable only for large, highly funded organisations and projects. However, in the last 15 to 20 years, massive advances in automation and speed, and reductions in cost of DNA sequencing have sparked a burst of mutually reinforcing growth in the accessibility of whole-genome sequencing and assembly, application domains, and bioinformatics tools for assembly and downstream analyses[1–3]. Whole-genome assembly (WGA), in particular *de novo* WGA i.e. reconstruction of an individual genome up to chromosome length without consulting previously resolved sequence[4], is a maturing and active field of research with well-established paradigms and a variety of tools addressing a range of needs. Nevertheless, genome assembly is an unsolved problem; due to the sheer difficulty of the task as well as the difficulty of finding realistic mathematical representations for it, assemblers still rely on heuristics and other ad hoc techniques instead of rigorous algorithms with provable performance guarantees[1]. Thus, there remains ample room at the time of writing for new approaches to continuing challenges in WGA. One of those, namely a statistical method for copy number estimation of contigs assembled from high-throughput sequencing data, shall be the subject of this thesis.

To set the stage for a more technical treatment of the subject matter, we shall start with a brief history of genome sequencing from the perspective of its usage in *de novo* WGA.

### 1.1.1 A brief history of genome assembly

Modern genome sequencing can be said to have begun with the introduction in 1977, by Frederick Sanger and colleagues, of a method for DNA sequencing with chain-terminating inhibitors[5], now commonly known as Sanger sequencing. This method became the most commonly used DNA sequencing technology for several years, spurring increasing automation and forming the basis of the first commercial DNA sequencing machines[6,7]; the data produced contains relatively long reads, typically between 300 and 1000 bps in length[8], at low read depth (note that  “coverage” is often used instead of “depth”), i.e. each genomic locus is likely to be sampled in a relatively small number of reads.

During the "first generation" of genome sequencing represented by Sanger's method, the shotgun process, still prevalent today, was first developed and used to assemble long contiguous genomic sequences from shorter reads. The process randomly breaks a target molecule; the resulting fragments are sampled and sequenced to obtain reads[9,10]. In the case of whole-genome shotgun (WGS) sequencing, the target molecules consist of the chromosomes making up a genome. When a target is oversampled, the resulting reads overlap; they can then be computationally ordered and assembled[4]. The broad applicability of WGS was demonstrated by the 1995 completion of the 1.8-Mbp (million base pair) *Haemophilus influenzae* genome by WGS sequencing[11] and a number of subsequent projects[12]. Milestones were reached when almost all of the 120-Mbp euchromatic portion of the *Drosophila melanogaster* genome[13], and (in the Human Genome Project or HGP) a 2.91-Gbp consensus sequence of the euchromatic portion of the human genome[12], were determined through WGS sequencing supported by other techniques.

Concurrent with the spread and progress of first-generation sequencing technology, a luminescent method for measuring pyrophosphate synthesis was introduced[14]; its application to DNA sequencing via a technique called pyrosequencing was pioneered over the next decade[15,16] and licensed to the company 454 Life Sciences, where it was deployed in the first major commercially successful "next-generation sequencing" (NGS) machines[7]. These machines coupled pyrosequencing with massively parallelised sequencing reactions, producing up to a million reads ~200-500 bps long in each run, which represented an orders of magnitude increase in sequencing yield[7,17,18], and higher read depth relative to Sanger data. Massively parallel platforms from Solexa (now Illumina) output even shorter reads of 20 to 40 bps[19].

The massively parallel output of relatively short shotgun-generated reads is a shared, defining characteristic of several sequencing techniques that emerged over the next decade, which have come to be seen as second-generation sequencing (SGS). These techniques are also known by various other names, i.e. the aforementioned NGS, massively parallel sequencing (MPS), and high-throughput sequencing (HTS)[7,18,20]. Since the release of Illumina's Genome Analyzer II sequencer in 2006, competition between SGS technology manufacturers drove tremendous gains in output and reductions in cost, with raw per-base cost plummeting by four orders of magnitude between 2007 and 2012[21,22]. As a result, SGS had almost completely supplanted Sanger sequencing a decade after the HGP was completed in 2001, and Illumina has achieved a near-monopoly on the SGS market. At present, SGS reads are typically a few hundred bps long with low error rates (below 1% across Illumina platforms, consisting mostly of substitutions[18,20]), and sequencers produce an output of up to 1Tbps, or billions of reads, per run[7,20,22].

Alongside the rise to ubiquity of NGS, yet another class of sequencing technologies, sometimes called the third generation, has been advancing rapidly[7,22]. Characterised by long-range single-molecule resolution, these technologies do not require amplification in library preparation, and now routinely produce reads of average length around 10kb, in a range that can exceed 1Mb[23]. They consist of two main approaches, single-molecule real-time (SMRT) sequencing from PacBio[24] and nanopore-based sequencing from Oxford Nanopore Technologies[25]. Distinct from these true long-read platforms, synthetic long reads are created through library preparation protocols that associate NGS short reads derived from a single larger molecule with the same barcode; these are currently available on platforms from two vendors, Illumina and 10X Genomics[26]. Another NGS-compatible advance has been the creation of very long-range mate pair-like data from Hi-C and related chromatin crosslinking protocols[23,27]. Lastly, new optical mapping technology from BioNano Genomics uses nicking enzymes to create high-resolution sequence motif physical maps that can be *de novo* assembled into scaffolds to complement assembled genomic sequence[23,28].

In principle, if assembled using effectively tailored approaches, single-molecule data offer alternatives or solutions to the drawbacks of NGS technology for *de novo* WGA, including amplification artefacts from library preparation, and difficulty in characterising long repeats and large structural variation[23]. For example, high-quality long-read-only assemblies were created of microbial[29], maize[28], and human[30,31] genomes. However, in practice at the time of writing, the relatively high error rates (e.g. around 10% for PacBio reads[22]), right-skewed length distribution[23], and per-base cost of single-molecule reads make them impractical for many purposes unless used in conjunction with a short-read base assembly[32]: they require costly error-correction procedures[33], can be very computationally expensive to assemble[23,34,35], or require prohibitively high depth to overcome the error rates and increase availability of the longest, most useful reads[23]—the aforementioned example assemblies used 65x to 103x SMRT read depth or, in one case, 22x and 24x SMRT with 80x optical mapping depth. Moreover, short reads, particularly Illumina reads, are widely used and likely to remain so for some time due to their high accuracy and low cost, especially for large-cohort studies and price-sensitive personalised medicine applications[32,36]. Not least, a NGS workflow can be leveraged to include linked reads and produce a high-quality assembly at modest extra cost[23,37,38]. Thus, NGS short-read *de novo* WGA is likely to remain highly relevant for some time.

### 1.1.2 An overview of *de novo* whole-genome shotgun sequence assembly

Of course, raw genomic sequence reads do us no good without tools, such as assemblers, that turn them into useful information. For much of the history of DNA sequencing—i.e. before the revolution in accessibility and applicability of the SGS era—genome sequencing and assembly was its primary purpose[22]; at the same time, its inaccessibility translated into a low level of activity in the field of assembly. The proliferation of SGS technologies and applications renewed interest in assembly, driving the rapid development of assemblers able to adapt to and leverage the characteristics of data produced by new sequencers.

This history is visible in the taxonomy of *de novo* WGS assemblers. At the most basic level, they belong to two paradigms, each corresponding to a distinct underlying search strategy: greedy and graph-based[39]. The earliest assemblers used greedy algorithms in response to the assumed computational intractability of the earliest formalisation of shotgun assembly as equivalent to the shortest common superstring (SCS) problem for a set of sequences[40]; some well-known examples areTIGR[41] and phrap[42]. However, the difficulty of incorporating longer-range information into the inherently local assembly process of greedy assemblers is a major handicap. Thus, they have been superseded by graph-based assemblers[1], which in turn consist of two major subtypes: overlap-layout-consensus (OLC), and de Bruijn graph (DBG)[1,39].

Regardless of paradigm, all sequence assemblers operate on the general assumption that a sufficiently high degree of similarity between two sequences indicates that they are contiguous (or even coincident) in the genome used to generate the assembly dataset[1]. A graph-based assembler constructs a graph representing sequenced data to provide a basis for assembly decisions, with each node representing a sequence, and directed edges, each representing a suffix-to-prefix overlap between the nodes it connects.

The simplest definition of an OLC graph contains a node for every read, and an edge between every pair of reads with an overlap exceeding minimum thresholds in length and sequence identity[43]. This allows it to fully exploit information present in reads and supports error tolerance—characteristics well-suited for assembling longer, more inaccurate reads in lower-depth datasets, e.g. those from Sanger sequencing[1,4]. Unsurprisingly, the OLC paradigm was popularised by the Celera assembler, which helped produce the *Drosophila* genome[13,43] from Sanger reads. However, concerns about the inefficiency of computing overlaps between large numbers of reads have limited its application to SGS data, especially for larger genomes, though it is regaining ground with the emergence of long-read third-generation sequencing. Two well-known exceptions are the assembler Edena[44], which was published with results from bacterial genomes, and SGA[45], which uses efficient string indexing data structures to enable assembly by a variant of OLC, the string graph, that simplifies the overlap graph by removing redundant information (transitive edges)[1].

In contrast, the DBG paradigm took off with the success of SGS. In the context of sequence assembly, a de Bruijn graph consists of all distinct substrings of some specified length, k, extracted from reads in a sequencing dataset, each uniquely represented by a node, and a directed edge between every pair of nodes with an exact k-1 bp suffix-to-prefix overlap, e.g. from ACGT to CGTA. First proposed as the "Eulerian approach" (i.e. representation as a search for an Eulerian path, a path traversing each edge once, in a graph) in relation to sequencing by hybridisation in 1989[46], it was shown to be applicable to Sanger read assembly in 1995[47]. The first DBG assembler, EULER, was introduced in 2001, with pre-assembly error correction to mitigate the drastic effects of base-calling error on its performance[48]. With the time and space required to build a DBG being linear in the size of the sequence underlying a dataset, as opposed to the quadratic increase with input size for building an OLC graph, DBGs scale well with the high depth of SGS datasets. As SGS became widely adopted over the next decade, several DBG-based assemblers were developed and improved to keep up with the characteristics of new sequencer outputs. Examples include EULER-USR (one of a few adaptations of EULER)[19], Velvet[49,50], ALLPATHS and ALLPATHS-LG[51,52], ABySS[53], SOAPdenovo[54,55], SPAdes (using variants on the standard DBG)[56], and DISCOVAR[57]. Moreover, the generally very low error rates of SGS support DBG reliance on exact overlaps. With advances in long-read sequencing, DBG assembly has also been leveraged as the basis of hybrid workflows incorporating long-read data, e.g. in ABySS 2.0[36] and Supernova[38] (using linked reads), as well as Unicycler[32] and a recent unnamed diploid assembler[58] utilising SPAdes (using single-molecule reads).

The active research on genome assembly reflects the fundamental difficulty of the problem (at least with the technologies available to date), as revealed by the pioneering work of Esko Ukkonen et al. in the 1980s[59]. He demonstrated that sequence assembly, including genome assembly, ranges from trivial, to computationally intractable, to impossible[60] (i.e. information in sequencing data is insufficient to identify the correct genome reconstruction from an exponential number of equally good alternatives). This is certainly true of shotgun short-read assembly, even in hybrid workflows; of special interest here are some continuing challenges in the context of de Bruijn graph assembly of SGS data and how they can be addressed by the application of a statistical method for copy number estimation of partially assembled sequences. The rest of this thesis presents a technical treatment of its subject, starting in the next section with an introduction to DBG *de novo* WGS assembly of SGS data, followed by a survey of the issues of interest and existing approaches to addressing them.

## 1.2 Introduction to de Bruijn graph assembly

So far, DBG assembly has been presented as a monolithic, relatively unvarying entity; the reality is, of course, more complicated. That said, familiarity with a single, representative workflow shall suffice for an appreciation of the main subject matter of this thesis. Thus, while the exact terminology, graph structures, and workflow used vary considerably across assemblers, just one set of definitions (underlined, in bold) and descriptions, summarised and abstracted from the literature, shall serve as the conceptual basis for the rest of this material.

The k-mer de Bruijn graph, the foundational data structure of DBG assembly, is as defined earlier. The main stages of assembly are as follows:

### 1.2.1 k-mer de Bruijn graph construction

The canonical first step in the DBG assembly workflow. In practice, it is often preceded by an error correction step intended to fix or discard k-mers or reads containing sequencing errors, especially substitution errors[19,51,52,54,55,57,61]; alternatively, it can be accompanied by filtering out low-depth k-mers, i.e. those appearing in a number of reads below a specified threshold[36,38,53]. The construction is done as one might expect: given (as is often the case) a uniform input read length of some length $L$, extract all $L - k + 1$ k-mers present in each read, creating nodes to represent those newly encountered, and directed edges as implied by the definition given earlier. Thus, each node and each edge is unique; a k-mer is represented once regardless of how much it occurs in the genome, a key advantage in space complexity. Note that each node can have at most four incoming edges (corresponding to the letters of the DNA alphabet, A, C, G, and T), and similarly, at most four outgoing edges. Figure 1 gives a toy example of a k-mer DBG.

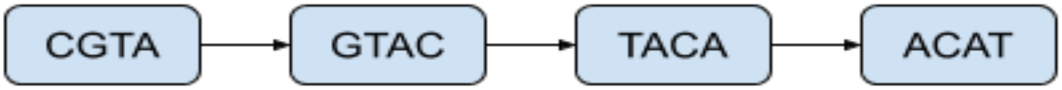

**Figure 1 k-mer graph of the sequence "CGTACAT", with *k* = 4.**

### 1.2.2 Contig building

Intuitively, nodes are merged along linear paths in the graph; the resulting string for each path is called a **<u>contig</u>**. For example, if a path consists of nodes representing "ACG", "CGT", and "GTA", they would be merged to give a contig representing "ACGTA". More formally, each linear path has a source and a sink node, with all intervening nodes having both in-degree and out-degree 1; each node can belong to exactly one path. Thus, each path ends at a dead end, or at a branching point (at or adjacent to a node with total degree over 2). This step results in the contig graph, in which the k-mer nodes underlying each contig have been merged into a single node representing the contig, and each edge represents an overlap between two contigs. Figure 2 illustrates with an example.

Note that the contig graph contains no non-branching paths longer than a single node.

### 1.2.3 Scaffolding

This step is enabled by the ubiquity of paired-end sequencing in SGS (as well as the availability of other forms of long-range information), in which each read is located on one end of a sequence fragment, with a partner on the other end sequenced in the opposite orientation. The distance between the paired reads is drawn from a distribution and only approximately known[1].

In this step, long-range information, e.g. paired-end reads, mate pairs (paired reads providing the same type of information as paired-end reads at larger distances), linked reads, or single-molecule reads, are aligned or associated to the contig or k-mer graph to assess the degree of long-range support for graph connection and path candidates. Each contig is then merged or linked with other contigs judged to be connected, possibly with intervening gaps. This merging/linking process continues until a dead end is encountered, or a point is reached where no sufficiently supported path extension alternative is available, or an end-to-end path in the contig graph is found between two contigs classified as single-copy (usually by heuristic criteria). The resulting sequence (which may contain gaps) is called a **scaffold**. This process is carried out on all contigs possible, taking care to avoid redundancy. Multiple types of long-range information may be used, often in successive stages in increasing order of distance.

Scaffolding encompasses repeat resolution, gap filling, and gap size estimation. The set of scaffolds obtained at the end of this stage represent the most contiguous reconstruction possible of a genome from a particular dataset, with a particular configuration of the assembler used. This can be considered the final output of assembly.

k-mer graph construction (1) and contig building (2) are often separated by graph simplification steps, in lieu of [36,49,50,53,56] or in addition to[54,55] pre-assembly error correction. Those steps typically consist of spur erosion (also called tip removal), bubble collapsing, and sometimes removal of low-coverage sequences; the next section shall cover them in more detail.

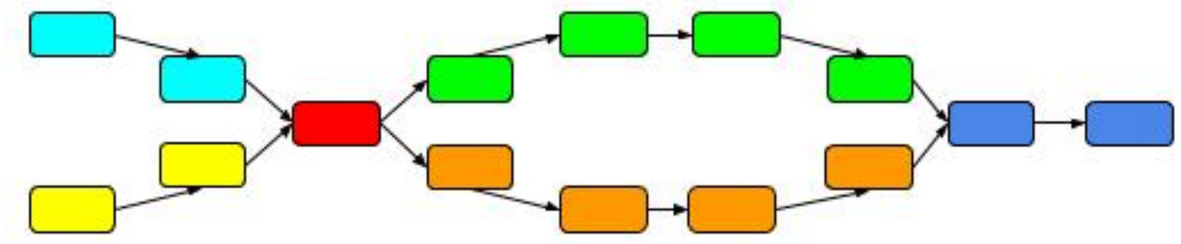

**Figure 2a k-mer graph (k-mers omitted): Linear paths, each corresponding to a contig, colour coded.**

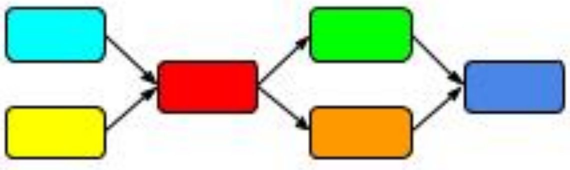

**Figure 2b Contig graph: Linear path nodes merged into contigs; each node represents a contig.**

## 1.3 Challenges in de Bruijn graph assembly of short reads

The goal of *de novo* genome assembly is to reconstruct a target genome to the highest possible degree of correctness, contiguity, and coverage (i.e. completeness). In particular, correctness comprises both base-level and structural accuracy, contiguity the lengths of reconstructed segments, and coverage the fraction of the genome reconstructed. Structural correctness and coverage additionally encompass assembly size accuracy, which in turn affects genome size estimation insofar as it is used for that purpose.

With respect to many organisms of interest, this task is fundamentally limited by the fact that the chromosomes constituting their genomes are longer than any sequencing read length possible to date. This is true of short-read assembly to an even greater degree, and for even more genomes. Furthermore, sequencing reads are subject to random error, i.e. mainly base call errors in the case of Illumina platforms, which effectively monopolise the present SGS market. Thus, the target genome is over-sampled to maximise genome coverage (with the degree of oversampling referred to as **sequencing depth**), and to minimise the impact of error through redundancy of correct base calls, though this strategy is complicated somewhat by chance and by the compositional bias of sequencing technologies against regions with more extreme levels (both low and high) of GC content[62,63].

### 1.3.1 Definition: Repeats

Crucially, given any particular level of sequencing depth, many genomes (i.e. those of nontrivial size and complexity) also possess characteristics that further confound sequence assembly, namely heterozygosity and repeats. For our purposes, let repeats first be defined as follows (using substring to denote a contiguous sequence of characters):

> Given a set of sequences $S = \{ s_i \mid i \in \{ 1, 2, \ldots, n \}$ for some $n \in \mathbb{N}$, $\text{len}(s_i) \geq k \}$ from the alphabet $\{ A, C, G, T \}$, where $\text{len}(s_i)$ represents the length of $s_i$,
> let $U = \{ u \mid u$ substring of $s_i \in S, i \in \{ 1, 2, \ldots, n \}, \text{len}(u) \geq k \}$, and
> occur($u$) represent the multiplicity, i.e. number of occurrences, in $S$ of a substring $u$.
> $R = \{ u \mid u \in U, \exists\ v \in U, i \in \mathbb{N}$ such that $u$, $v$ both substrings of $s_i$ and $\text{occur}(u) > \text{occur}(v) \}$
> $R$ is the set of repeats of length at least $k$ in $S$. A **repeat** is any $r \in$ R.

### 1.3.2 Effects of heterozygosity and repeats

Using this definition, a genome can be represented by $S$, with chromosomes as its members, and whether a sequence is a repeat relies on the existence of some sequence(s) that is connected to it and has lower multiplicity. A repeat (flanked by lower-multiplicity, i.e. lower copy number, sequences) induces path nonlinearity, and thus ambiguity, in the k-mer graph. For example:

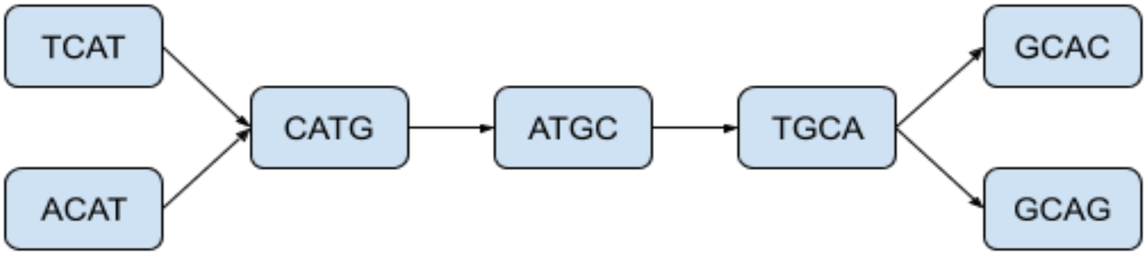

**Figure 3**: k-mer graph formed around the repeat motif "CATGCA", with *k* = 4.

There are two sets of two sequences compatible with the structure in Figure 3 (sometimes referred to as a "frayed rope" motif), one for each possible combination of the two entry and two exit paths: { TCATGCAC, ACATGCAG } and { TCATGCAG, ACATGCAC }. Without adequate supplementary information, e.g. from long-range data such as paired-end reads, the repeat cannot be resolved with reasonable certainty into the correct set of paths. In such a case, the contigs corresponding to the collapsed segment and the paths on either side cannot be merged and extended farther, affecting scaffold length and thus contiguity of assembly. That could have side effects on genome coverage, assembly size, and consequent genome size estimates, as illustrated in Figure 4: given that path copy number is unobserved, fully resolved repeats yield correct scaffolds with (locally) full genome coverage and correct assembly size, whereas unresolved repeats result in lower genome coverage and inaccurate assembly size.

In this thesis, we consider only haploid and diploid genomes. Continuing with the same example, without loss of generality, and using different multiplicities from those illustrated, suppose the structure in Figure 4 is induced by the set { TCATGCAC, ACATGCAG }. In a haploid genome, that corresponds to each of the sequences in the set occurring at separate loci in the genome. In a diploid genome, it could arise from the situation just described (e.g. at two separate loci on both members of a pair of homologous chromosomes), or from homozygosity (flanked by heterozygous sequence) at the same locus on homologous chromosomes, which is mathematically equivalent to a repeat. For compatibility with the standard practice of producing a haploid "consensus" instead of diploid (haplotype-aware) scaffolds, from now on "repeat" shall denote the shared motif only in the former. The two causes make a material distinction in conventional assembly: for repeats, the structure should be separated into paths of the correct multiplicities; otherwise, it should be collapsed into a single path. Incorrect resolution contributes inaccuracy in genome coverage and assembly size, and decreases contiguity as well. However, graph topology alone does not provide enough information to disambiguate the two, creating another challenge to which current solutions could be improved upon.

The effect of heterozygosity on the k-mer DBG is often exemplified by the "bubble" structure, i.e. a set of (at most four) disjoint paths sharing only their start and end nodes. Figure 5 shows a simple, balanced example (with equally long paths). The bubble can also result from sequence flanked by higher-copy-number segments, just as the frayed rope arises from multiple distinct causes. The two can be seen as duals, and often occur sequentially in k-mer DBGs. Indeed, the role of a bubble in confounding assembly is dependent on its overlapping with a frayed rope on one or both ends, causing the challenge of repeat resolution described earlier, possibly affecting assembly contiguity, coverage, and size. Moreover, the challenge of disambiguating between homozygosity and repeats is equivalent to that of distinguishing heterozygous paths from those of lower multiplicity relative to flanking segments. Otherwise, absent sequencing error (addressed next), a solitary bubble can be confidently separated into two correct paths.

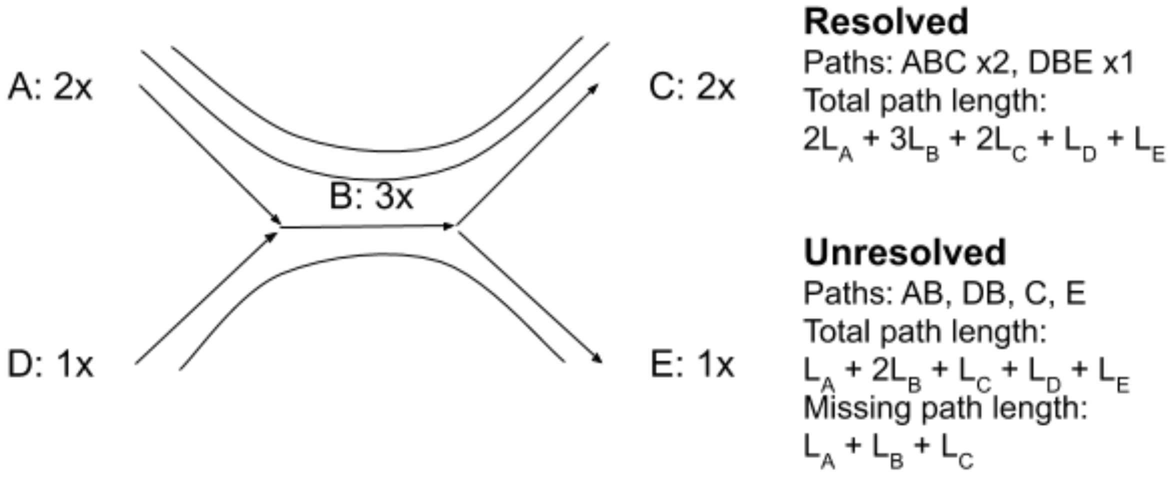

**Figure 4**: Effect of repeat resolution (or lack thereof) on assembly size, with contigs labelled A-E. Curves denote long-range support for the correct paths, ABC (of multiplicity 2) and DBE.

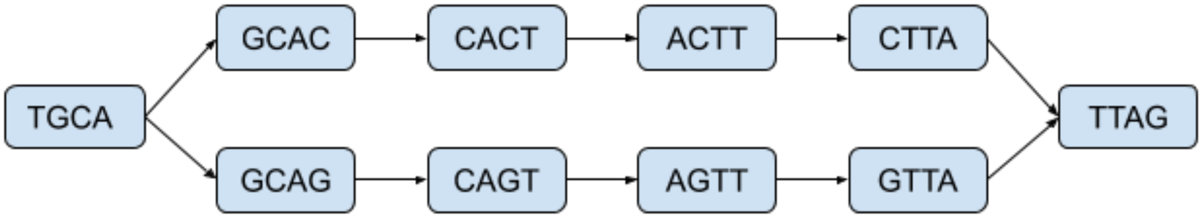

**Figure 5**: A simple bubble, induced by { TGCACTTAG, TGCAGTTAG }, equivalent to a SNP (single nucleotide polymorphism), or a base call error on the last character of GCAC or GCAG, or an inexact repeat (with paths having half the multiplicity of source and sink nodes).

### 1.3.3 Definition: Copy numbers

Apropos of the discussion about repeats, heterozygosity, and multiplicity in general, the concept of copy number (synonymous with multiplicity) is key to the rest of this thesis. We shall use the following operational definition for sequences of length greater than some reasonably large *k* (~20), which ignores some possibilities that are of no practical significance or extremely unlikely.

For a sequence from a haploid genome, the definition is simple: it has copy number 0 if it does not occur in the target genome; 1 if it occurs only once; 2 if it occurs twice (whether on the same or separate chromosomes); and so on. In other words, its copy number and number of occurrences are equal.

For a sequence from a diploid genome, there are complicating considerations. Technically, a unique heterozygous sequence occurs once in the genome, a unique homozygous sequence twice, a once-duplicated homozygous sequence with or without a heterozygous mutation in one locus three or four times respectively, and so on. However, beyond the important special cases of unique heterozygous and homozygous sequences, distinguishing between higher odd and even multiplicities is inconsistent with the standard practice of producing haploid "consensus" scaffolds. Thus, for present purposes, sequences having three occurrences are aggregated with those having four, and those having five with those having six, and so on. To maintain gap-free integer numbering, each unique sequence is assigned half its number of occurrences as its copy number; for non-unique sequences, this is rounded up to the nearest integer: 1 occurrence corresponds to copy number 0.5, 2 occurrences to 1, 3 and 4 to 2, 5 and 6 to 3, and so on.

### 1.3.4 Other complications

A bubble can also be caused by sequencing error; the discrepancy between correct and error reads induces sequences and k-mers resembling those from a heterozygous locus; in particular, a base call error looks like a SNP. Thus, an assembler needs to tell error apart from other competing reasons for the presence of a bubble and retain only the correct path in this case; alternatively, it can prevent bubble formation by error in many cases, albeit at a cost. Failure to do so results in base-level inaccuracy at best, or reduced contiguity, or possibly structural error.

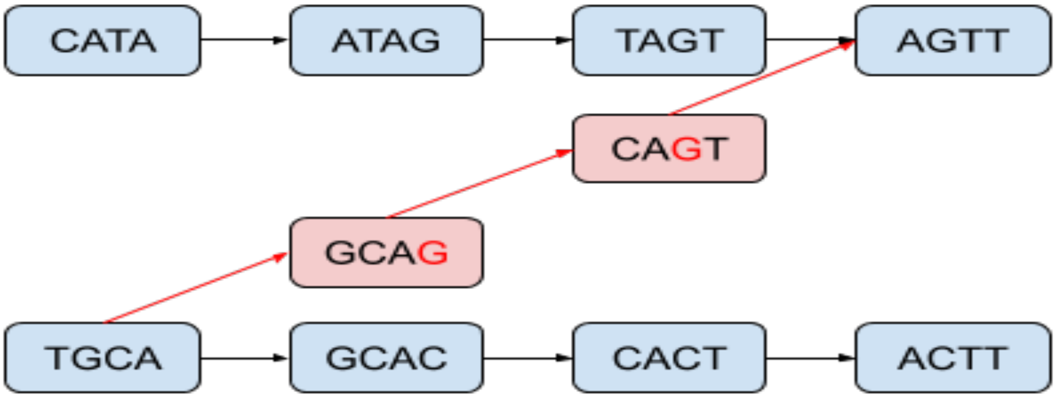

**Figure 6**: Spurious connection resulting from error-induced path, with base call error and spurious path in red. The spurious path could also reconverge on the correct path at the bottom to form a bubble, but that is omitted for clarity.

Last but not least, regions of low read depth also pose problems in assembly, affecting contiguity and coverage, and most insidiously, causing misassemblies. Detailed discussion is omitted here since statistical copy number estimation is expected to be of marginal utility in addressing these problems, both relative to the utility of other approaches, and to its utility in addressing some of the other issues discussed earlier. That said, it induces misassembly by causing the absence of k-mer graph edges that should exist in a correct representation of the target genome; that effect is illustrated in Figure 10 in relation to inappropriate error correction.

The (non-linear) DBG structures presented so far are by no means an exhaustive catalogue of what is encountered in practice. They merely illustrate the most basic outcomes possible from introducing vertices of in- or out-degree exceeding 1. In practice, DBGs are often far more complicated, with arbitrarily complex paths subject only to the constraint of node in- and out-degree not exceeding 4; for instance, exact tandem repeats induce cycles. Thus, a variety of approaches have been developed to address this complexity and its resulting problems, while room remains for additions to this toolbox.

## 1.4 Current approaches to challenges in short-read DBG assembly

Implicit in the discussion above is that the issues described (when not pre-empted—which comes with its own set of problems) apply to contigs in particular, and manifest during contig scaffolding. Some would benefit from accurate copy number estimation: Most simply, the effects of any given occurrence of sequencing error would disappear if spurious contigs arising from it were correctly identified as having copy number 0 and removed. A slightly less simple but arguably much more significant improvement on current practice would be, in general, reliable resolution of low copy number repeats into contigs of correct multiplicity, and in particular, the concomitant disambiguation of heterozygosity from other genomic features as a cause of DBG bubbles (contigs induced by the former have copy number 0.5, while the latter would be associated with higher copy numbers); these would increase accuracy of genome coverage and assembly size. Lastly, assembly contiguity could be a minor area of impact: First, an accurate repeat copy number estimate can be combined with graph topology to enable resolution of paths with insufficient long-range data support when an unambiguous solution to graph-based constraints exists; Figure 7 illustrates.

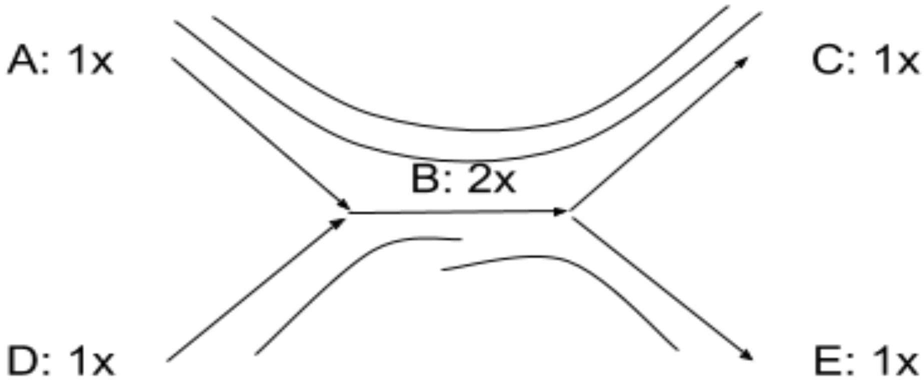

**Figure 7**: ABC has sufficient long-range support, but DBE doesn't. However, B has "remaining" copy number 1, which gives unambiguous support to DBE based on graph topology.

Perhaps more importantly, under certain conditions, accurate contig copy number estimation could improve scaffold contiguity, by preventing inappropriate bubble collapsing (merging separate paths into one), thus preserving long-range information associated with the separate paths for use in repeat resolution; Figure 8 illustrates.

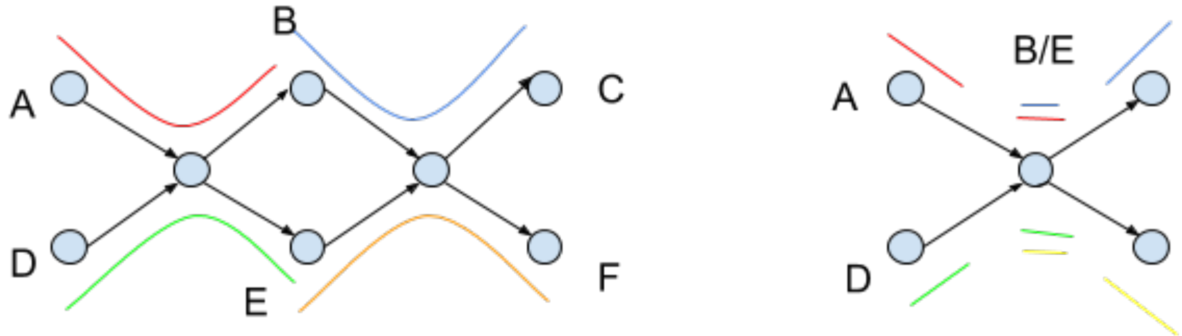

**Figure 8**: Nodes represent contigs. Each colour represents a linked read barcode (more barcodes are required in practice). Left: B has barcode overlap with both A and C, while E overlaps with D and F, thus resolving the subgraph into paths ABC and DEF. Right: with B, E, and their adjacent contigs merged, resulting node B/E has barcode overlaps with A, C, D, and F alike, so the paths can no longer be resolved. Note that this example does not apply to true long read information.

### 1.4.1 Sequencing error handling

Sequencing error-handling strategies can be classified into three categories, in increasing order of complexity: pre-assembly filtering of likely erroneous sequences, graph simplification, and pre-assembly error correction. The last is largely complementary to the first and second[19,56], though not always[54,55,57].

The simplest strategy, pre-assembly error filtering, excludes sequences from assembly by thresholding on read depth; it is used in conjunction with the other strategies. In ABySS[36] and Supernova[38], k-mers with read depth below an internal or user-specified threshold are excluded from k-mer DBG construction. This strategy is based on the likelihood that with high sequencing depth, most correct k-mers would be covered by a certain number of reads, and that given a low error rate, error-induced k-mers appear in a very small number of (spurious) reads[19]; of course, its sensitivity and precision also depend on an optimal threshold being chosen. However, it is also a dragnet that also removes many legitimate k-mers that happen to be in low-read-depth genomic regions, compromising graph continuity and even increasing the risk of misassembly, as alluded to in the last section.

Graph simplification consists mainly of dead-end branch removal (also called spur erosion or tip removal), bubble collapsing (of disjoint paths into one), and (lastly, in some assemblers) removal of low-depth graph connections; the depth of a path is usually taken to be the average depth of its constituent k-mers (nodes).

A dead-end branch (Figure 9) in the DBG is likely induced by a base call error in the first or last *k* bases of a read (otherwise it would diverge from the last correct k-mer preceding it in the read, and converge back onto the next error-free k-mer); those below some length (usually ~2*k*) threshold are removed[54,55], sometimes iteratively[36,49,50,53], and sometimes with the additional criterion that they have lower read depth than other paths diverging from their point of origin. Like a pre-assembly depth threshold, it is an effective tool for identifying and removing spurious sequences, but also risks removing legitimate sequences in low-read-depth regions, though likely less so because it is additionally based on graph topology (even when the spur's depth relative to alternative paths is not considered, it is implicitly based on read depth since dead ends are caused by insufficient read depth for continuing k-mer overlap). Unlike that threshold, however, it disrupts connectivity only locally[49]. Bubble collapsing is used for the general purpose of graph simplification, such that its effects include removal of error-induced paths as well as merging of heterozygous paths into a consensus sequence; thus, it shall be addressed separately below as part of heterozygosity identification and repeat resolution. Removal of low-depth graph connections[54,55] below some hard threshold as the final step[49,50,56] of graph simplification targets low-depth paths remaining after prior steps, with the rationale that by then, most legitimate low-depth segments would be subsumed into long unique paths with little effect on average depth[49]. This is an effective strategy that can be further improved by enhancing its precision using statistically informed contig copy number estimates as a more flexible tool than a hard cutoff.

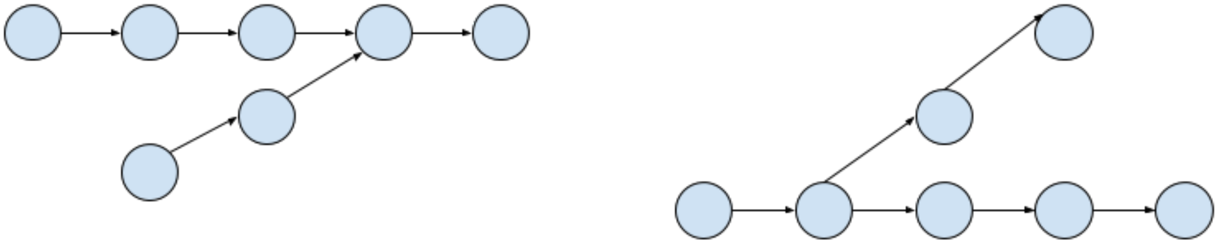

**Figure 9**: Dead-end branch examples

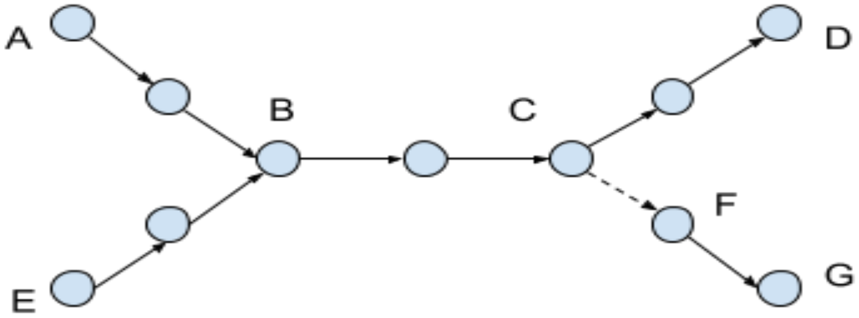

**Figure 10**: Misassembly caused by missing connection (dashed line; resulting from low read depth or removed k-mer). The correct set of contigs is { AB, BC, CD, EB, BC, CFG }, but the missing edge between C and and the node preceding F means { AB, EB, BCD, and FG } is produced instead.

Pre-assembly error correction is a relatively sophisticated method for handling sequencing error, specifically by pre-empting its effects. Most implementations[19,48,51,54,55,61] of this approach are based on counting the occurrences (read depth) of all k-mers across an entire sequencing dataset, followed by an attempt to correct reads containing k-mers with occurrences under some heuristic threshold (representing a minimum expected occurrence count for genuine k-mers). Reads for which the number of putative error k-mers can be reduced by a set of base-value changes meeting some criteria (e.g. sufficiently frequent alternatives, estimated likelihood of correctness) are changed; otherwise they are left unchanged or discarded. This procedure may be replaced in certain cases by more involved alternatives, e.g. a branch-and-bound tree (graph) algorithm[55].

However, these frequency-based procedures can discard true genetic variation when it coincides with low-depth regions. This effectively removes legitimate connections and paths, incurring the consequent effects on misassembly risk (see Figure 10) and contiguity (see Figure 8). Another error correction implementation[57] mitigates that by combining multiple sequence alignment of read pairs and base quality scores to identify low-frequency k-mers likely to represent genuine variation and exclude them from correction; however, as its authors state, this relies to some degree on PCR-free data.

### 1.4.2 Heterozygosity identification, repeat resolution, contiguity, and genome coverage

As explained earlier, heterozygosity identification and distinguishing unique homozygous sequence from repeats are equivalent problems. Therefore, they would ideally be solved together, i.e. by copy number estimation. That done, bubbles can be collapsed or retained as appropriate (corresponding to paths of copy number 0.5 or otherwise), before repeat resolution finally takes place.

However, at present, explicit copy number estimation in *de novo* WGS assemblers is generally unaddressed or basic, with arguably two exceptions that shall be described later. Where it is addressed, heuristics are used, such as assuming even coverage[51], or a user-specified depth threshold[58], or some multiple of the mean depth of all contigs[55]. A slightly more sophisticated example[50] adapts the Celera assembler's A statistic[43] into log-odds ratio for the contig being unique vs. having copy number 2; however, it only models repeat status, not a more precise copy number, and assumes Poisson-distributed read occurrences over the genome, whereas real sequencing data, especially high-depth SGS data, is often relatively over-dispersed[64]. Of those, two[55,58] target heterozygosity.

Instead, many assemblers use collapsing of simple bubbles, i.e. bubbles in which all intermediate nodes on all paths have in- and out-degree of 1, as a graph simplification measure meant to increase contiguity by reducing path ambiguity, collapse heterozygous paths into a consensus and, where applicable, discard spurious paths[36,49,50,53,54,56]. This is usually subject to criteria e.g. sufficiently similar sequence identity across paths, and retains the path with highest read depth as the consensus. However, given the well-documented existence of nested repeats in genomes[65], it can hardly be reliably assumed that bubble paths are heterozygous rather than of higher copy number. Besides, in a diploid genome, only two-path bubbles can indicate a correct choice of path in the presence of heterozygosity, which makes bubble collapsing inappropriate in other cases. The direct consequences of inappropriate bubble collapsing have been described, while its indirect effects are illustrated in Figure 8; not least, heterozygosity-induced bubbles that don't meet criteria for collapsing are retained and incorrectly treated as paths with higher copy number, affecting genome coverage and assembly size. Similarly, repeat resolution, genome coverage, and assembly size are determined entirely by contig scaffolding using long-range data, which as explained can yield inaccurate results for incorrectly or incompletely resolved repeats (see Figure 4).

## 1.5 Copy number estimation: related literature

There are few examples of any sort of copy number estimation that go beyond heuristics for assembly workflows. However, the sparse history of this endeavour reaches back at least as far as 2001, when an algorithm for a solution to the copy number problem was published along with EULER-DB[66]. It was formulated as the problem of finding a minimum flow in each connected component of the contig graph, satisfying unit lower capacity constraints on all edges (i.e. each edge has weight, corresponding to copy number, at least 1); for every node, the sum of the weights of inbound edges is constrained to equal that of outbound edges. This problem can be solved by an application of the min-flow max-cut theorem, a dual of the max-flow min-cut theorem. However, the time complexity of computing its solution is high, likely making it unsuitable for most or all SGS datasets.

To my knowledge, the only other example in a workflow for the express purpose of *de novo* WGS assembly arrived much later, when Unicycler, a tool for hybrid assembly of bacterial genomes from short and long reads, was published in 2017. Leveraging naïve usage of read depth information to inform a simple, computationally inexpensive graph-based approach, it starts by assigning copy number 1 to all ("seed") contigs satisfying both the following conditions: first, having median (as opposed to mean) read depth within 10% of the per-base median, and second, with both in- and out-degree at most 1. A greedy algorithm then propagates multiplicity where graph connections and depth are in close agreement, ending when a dead end is encountered or these conditions are no longer satisfied; this is repeated for all seed contigs. Figure 11 gives an abridged illustration. This is an elegant solution, but does not allow for copy number estimates that are independent of graph topology and make full use of depth distributional information. Its lack of independence and flexibility makes for many cases where estimates would be absent or incorrect, chiefly: its performance on repeat contigs is dependent on sufficient read depth across single-copy regions; no less, incorrect seed identification results in error propagation to all connected contigs.

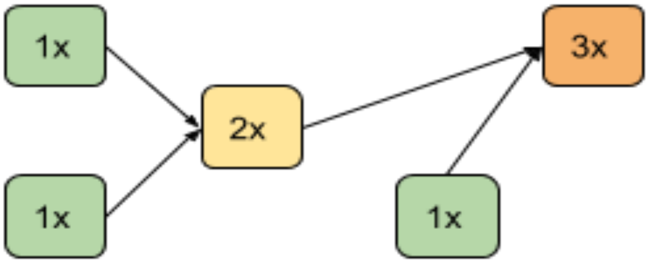

**Figure 11**: Each colour indicates one round of copy number propagation (green -> yellow -> orange).

There is, however, some relevant literature on *de novo* copy number estimation from whole-genome sequencing data, that takes more statistically informed approaches. One recent example is GenomeScope[67], a reference-free diploid genome profiler for short reads. It incorporates some mild postulates regarding heterozygosity and repeats into a negative binomial mixture model of the k-mer distribution as a function of depth, up to copy number 2:

$$\begin{aligned} f(x|\alpha, \beta, \gamma, \delta, \lambda, \rho, G) \\ &= G \cdot \{\ \alpha \mathrm{NB}(\mathrm{x}|\lambda, \lambda/\rho) + \beta \mathrm{NB}(\mathrm{x}|2\lambda, 2\lambda/\mathrm{rho}) + \gamma \mathrm{NB}(\mathrm{x}|3\lambda, 3\lambda/\rho) \\ &+ \delta \mathrm{NB}(\mathrm{x}|4\lambda, 4\lambda/\rho)\ \} \end{aligned}$$

where

*G* is a scaling parameter w.r.t genome size,

$\alpha$, $\beta$, $\gamma$, $\delta$ are mixing weights for each distribution (definitions omitted here)

$\lambda$ is the mean read depth of the distribution of heterozygous k-mers

$\rho$ is a common dispersion parameter

Usage of the negative binomial is motivated by the observation, mentioned earlier, that real sequencing data is often over-dispersed[64] relative to the Poisson distribution, which is popularly used to model sequencing read count data[43,68]; this allows variances to be controlled independently of the mean. This model is of limited utility for the purposes of contig copy number estimation due to its applicability to sequences of only one length, i.e. (some chosen) *k*, whereas contigs vary greatly in size; furthermore, it only models sequences up to copy number 2, whereas contig multiplicities can be arbitrarily high (within physical constraints). As well, it does not address haploid genomes, which should be entirely feasible whenever diploid genomes can be modelled.

In relation to that, an effort to estimate copy numbers of arbitrarily long contigs is not unprecedented in the literature: that task has been studied and applied for the purpose of *de novo* detection of copy number variation by co-assembly, in Magnolya, a tool published in 2012[69]. They model the number of reads $x_c$ that start on a contig *c* with a given copy number *i* as $p(x_c|i)$, with all contigs having $x_c$ exceeding some threshold collapsed into a geometric distribution representing outliers:

$$p(x_c|\pi,\lambda,\alpha) = \sum_{i=1}^{M} \pi_i \mathrm{Pois}(x_c|\theta_{i,c}) + \pi_{M+1} u(c) \mathrm{Geom}(x_c - \theta_{M+1,c}|\lambda,\alpha,M)$$

with

$L_c$ the length of contig c

$\lambda$ the number of reads starting at each base

$\theta_{i,c} = iL_c\lambda$ the Poisson rate parameter, with contig length $L_c$ and integer copy number i

u(c) an indicator function for c with $x_c \leq (M+1)L_c\lambda$

Geom() denotes the geometric distribution, with rate parameter $\alpha$

This seems to be largely a well-conceived model, though it still leaves room for improvement and adaptation. First, its assumption of shared mixture weights across contig lengths is incompatible with the probabilistically and empirically supported phenomenon that mixture weights differ with contig length; specifically, repeat probability decreases as length increases. This constrains its applicability to contigs from the same mixture population; for example, the article's authors fit it only on contigs ≥ 500bp. Second, it was designed for OLC contigs, for which read count data is naturally available; its discrete nature is unsuited to DBG assembly, for which collecting and retaining read counts or a similar discrete metric until the contig stage is infeasible. Last but not least, its assignment of contigs to the geometric component based on an observable characteristic (read count value), is inconsistent with maximum-likelihood assignment to the other components.

## 1.6 Statistically informed copy number estimation: Statement and aims

Thus, there is ample basis for a novel statistically informed copy number estimator in the context of *de novo* whole-genome shotgun short-read assembly. Specifically, we develop a statistically informed tool to estimate copy number for contigs from *de novo* WGS assembly of high-throughput short-read data for haploid and diploid genomes. This tool would supplement or supplant existing steps in assembly workflows as appropriate, for the following concomitant primary purposes:

1. Resolve multiplicity for heterozygous, unique, and low copy number repeat contigs.
2. Disambiguate heterozygosity-induced contigs from those of higher multiplicity.
3. Improve the accuracy of genome coverage, assembly size, and resulting genome size estimate in final assembly scaffolds. This would result from accomplishing (1) at the local (contig) level.

It would ideally fulfill the adjunct purpose of identifying spurious contigs (i.e. those having copy number 0), and thus potentially also serve as a computationally inexpensive alternative to error correction, particularly suitable for low-error datasets such as those from predominant Illumina machines. These improvements could also enhance assembly contiguity, as described earlier.

In order to achieve these aims, this tool would need to exploit the information from contig characteristics, primarily read depth and length, to the extent feasible. This can be done by building on existing examples in the literature, such as those just described, that approach problems related to those at hand by modelling copy number in sequencing data.