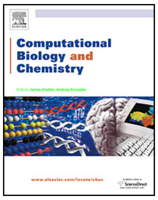
Computational Biology and Chemistry 57 (2015) 46–53

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem



Research Article

# Scaffold assembly based on genome rearrangement analysis

Sergey Aganezov[b,*], Nadia Sitdykova[a], Max A. Alekseyev[b], AGC Consortium[1]

[a] Academic University, St Petersburg, Russia
[b] The George Washington University, Washington, DC, USA

ARTICLE INFO

*Article history:*
Received 1 January 2015
Accepted 3 February 2015
Available online 3 March 2015

*Keywords:*
Scaffolding
Genome rearrangements
Breakpoint graph
MGRA
Genome assembly

ABSTRACT

Advances in DNA sequencing technology over the past decade have increased the volume of raw sequenced genomic data available for further assembly and analysis. While there exist many algorithms for assembly of sequenced genomic material, they often experience difficulties in constructing complete genomic sequences. Instead, they produce long genomic subsequences (*scaffolds*), which then become a subject to scaffold assembly aimed at reconstruction of their order along genome chromosomes. The balance between reliability and cost for scaffold assembly is not there just yet, which inspires one to seek for new approaches to address this problem. We present a new method for scaffold assembly based on the analysis of gene orders and genome rearrangements in multiple related genomes (some or even all of which may be fragmented). Evaluation of the proposed method on artificially fragmented mammalian genomes demonstrates its high reliability. We also apply our method for incomplete anophelinae genomes, which expose high fragmentation, and further validate the assembly results with referenced-based scaffolding. While the two methods demonstrate consistent results, the proposed method is able to identify more assembly points than the reference-based scaffolding.

## 1. Background

Genome sequencing technology has evolved over time, increasing availability of sequenced genomic data. Modern sequencers are able to identify only short subsequences (*reads*) in the supplied genomic material, which then become an input to genome assembly algorithms aimed at reconstruction of the complete genome. Such reconstruction is possible (but not guaranteed) only if each genomic region is covered by sufficiently many reads. Lack of comprehensive coverage (particularly severe in single-cell sequencing Chitsaz et al. (2011), Nikolenko et al. (2013)) and presence of long similar subsequences (*repeats*) in genomes pose major obstacles for existing assembly algorithms. They therefore often are able to reliably reconstruct only long subsequences of the genome (interspaced with low-coverage regions and repeats), called *scaffolds*.

The challenge of reconstructing a complete genomic sequence from scaffolds is known as the *scaffolds assembly* problem. It is often addressed technologically by generating so-called long-jump libraries Talkowski et al. (2012), Collins and Weissman (1984) or by using a related complete genome as a reference. Unfortunately, the technological solution may be expensive and inaccurate Hunt et al. (2014), while the reference-based approach is obfuscated with structural variations across the genomes Feuk et al. (2006).

In the current study, we assume that the constructed scaffolds are accurate and long enough to allow identification of orthologous genes. The scaffolds then can be represented as ordered sequences of genes and we pose the scaffolds assembly problem as the reconstruction of the global gene order (along genome chromosomes) from the gene sub-orders defined by the scaffolds. We view such gene sub-orders as the result of both evolutionary events and technological fragmentation in the genome. Evolutionary events that change gene orders are *genome rearrangements*, most common of which are *reversals*, *fusions*, *fissions*, and *translocations*. Technological fragmentation can be modeled by artificial "fissions" that break genomic chromosomes into scaffolds. Scaffold assembly can therefore be reduced to the search for "fusions" that revert technological "fissions" and glue scaffolds back into chromosomes. This observation inspires us to employ the genome rearrangement analysis techniques for scaffolding purposes.

Rearrangement analysis of multiple genomes relies on the concept of the breakpoint graph. While traditionally the breakpoint graph is constructed for complete genomes, it can also be constructed for fragmented genomes, where we treat scaffolds as "chromosomes". We will demonstrate that the breakpoint graph of multiple genomes possesses an important property that its connected components are robust with respect to genome fragmentation. In other words, connected components of the

[*] Corresponding author. Tel.: +1 7039498047.
*E-mail address:* aganezov@gwu.edu (S. Aganezov).
[1] See Appendix A.

http://dx.doi.org/10.1016/j.compbiolchem.2015.02.005
1476-9271/

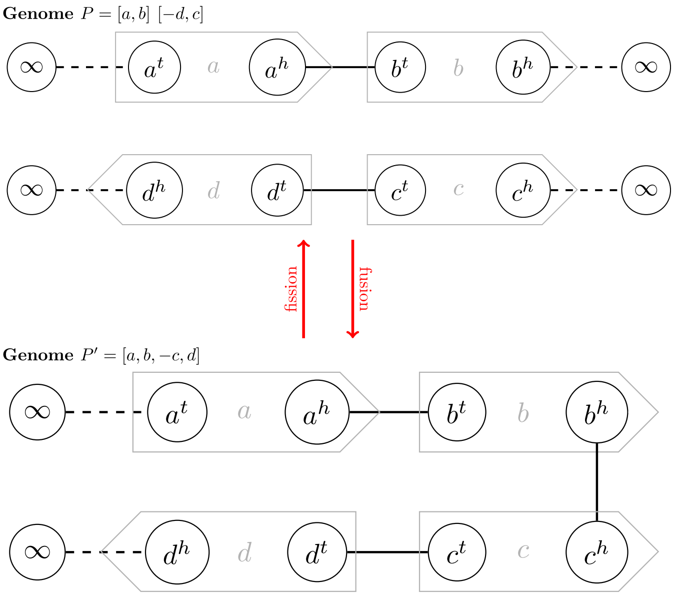


**Fig. 1.** Fusion/fission operations between the genome graphs of two-chromosomal genome $P = [a, b]\ [-d, c]$ and unichromosomal genome $P' = [a, b, -c, d]$, where regular and irregular edges are represented as solid and dashed, respectively. Grey boxes enclose pairs of vertices representing genes.

breakpoint graph mostly retain information about the complete genomes, even when the breakpoint graph is constructed on their scaffolds. We will show how to utilize connected components of the breakpoint graph for the scaffold assembly of fragmented genomes.

The paper is organized as follows. In Section 2, we provide background information about breakpoint graphs and genome rearrangements, discuss connected components of breakpoint graphs with respect to genome fragmentation, and describe our scaffold assembly algorithm. In Section 3, we evaluate our proposed algorithm on both simulated and real data. We summarize and discuss the paper results in Section 4.

## 2. Methods

### 2.1. Genome and breakpoint graphs

We start with defining a graph representation for a single genome, which may consist of multiple chromosomes and/or scaffolds commonly referred to as *fragments*. We represent each fragment with $n$ genes as an undirected graph on $2 \cdot n$ *regular* vertices representing gene extremities and several *irregular* vertices, labeled by $\infty$, encoding fragment ends (telomeres, if a fragment is a chromosome). A gene $a$ is represented by two regular vertices labeled as $a^t$ and $a^h$ denoting its *tail* and *head* extremities, respectively. Vertices corresponding to extremities of adjacent genes are connected by *regular* edges. Each vertex corresponding to a gene extremity at a fragment end is connected to an irregular vertex with an *irregular* edge. The *genome graph* of a genome is formed by the graph representing its fragments.

We remark that a single genome rearrangement affects the genome graph as follows: a pair of edges are removed and a new pair of edges on the same four vertices is created (Fig. 1). Genome rearrangements are therefore often modeled as DCJ Yancopoulos et al. (2005) or 2-break Alekseyev and Pevzner (2008) operations on graphs.

The *breakpoint graph* of $k$ genomes composed of the same $\ell$ genes consists of $2 \cdot \ell$ regular vertices (representing gene extremities), a number of irregular vertices (representing fragment ends) and undirected edges of $k$ colors (one color reserved for each of the genomes) encoding adjacencies between genes and/or fragment ends in the genomes. The breakpoint graph can be viewed as the superposition of $k$ genome graphs of individual genomes (Fig. 2). All edges connecting a pair of vertices in the breakpoint graph form a *multiedge*, whose *multicolor* is the set of individual colors in the multiedge[3] (e.g., in Fig. 2 vertices $a^h$ and $b^t$ are connected by a multiedge of the red–black multicolor).

We remark that traditionally breakpoint graphs are constructed on synteny blocks whose endpoints represent *breakpoints* (thus the name) in the genomes. In contrast, we construct the breakpoint graph directly on genes, whose extremities may or may not form breakpoints. Such graph therefore can contain *trivial multiedges* formed by parallel edges of all colors, which correspond to gene adjacencies shared across all the genomes and would be hidden within synteny blocks. Clearly, each trivial multiedge in the breakpoint graph forms its own connected component, which we also call *trivial*.

### 2.2. Connected components and fragmentation

Under a *connected component* in the breakpoint graph we will understand any largest set of regular vertices such that any two of them are connected by a path consisting of regular edges of any colors. The connected components form a partition of the regular vertices. We will show that this partition is robust with respect to fragmentation of the genomes. Namely, we observe that the connected components of the breakpoint graph of multiple genomes are strongly connected and can be hardly broken by technological “fissions”. To support this observation, we applied a number of random fissions[4] to six complete mammalian genomes and analyzed how such fissions affected the connected components of the breakpoint graph.

Using Ensembl BioMart tool Kasprzyk (2011), we obtained the following six complete mammalian genomes and pairwise orthologous gene mappings between them: *Homo sapiens* (GRCh38), *Mus musculus* (GRCm38.p2), *Rattus norvegicus* (Rnor_5.0), *Canis familiaris* (CanFam3.1), *Macaca mulatta* (MMUL_1.0), and *Pan troglodytes* (CHIMP2.1.4). From the orthologous gene mappings, we constructed gene families and filtered some of them so that each genome was represented as sequences of the same 11816 genes, each appearing in a single copy.

In order to determine how robustness of the connected components depends on the number of genomes, we analyzed different subsets of mammalian genomes of various sizes from 3 to all 6. For each subset of size $\ell$, we considered all possible combinations of $\ell$ mammalian genomes and constructed their breakpoint graph. After the same number of random fissions[5] was applied to every genome, we computed the averaged number of nontrivial connected components in the breakpoint graphs.

The results in Fig. 3 demonstrate that as the number of genomes grows, random fissions are less likely to break connected components into smaller ones. In other words, most connected components in the breakpoint graph of fragmented genomes are likely not affected by random fissions and thus represent connected components also in the breakpoint graph of complete genomes. We will employ this robustness property of the connected components and use them to guide our scaffolding algorithm.

[3] Each regular vertex may be adjacent to at most one irregular vertex (e.g., in Fig. 2 the vertex $a^t$ is connected to a single irregular vertex with red, black, and blue edges forming the red–black–blue mutiedge $(a^t, \infty)$).

[4] Unfortunately we do not have information about the actual mechanism of fragmentation of a genome into scaffolds.

[5] A random fission operation uniformly selects a regular edge in the genome graph performs a fission on this edge (Fig. 1).

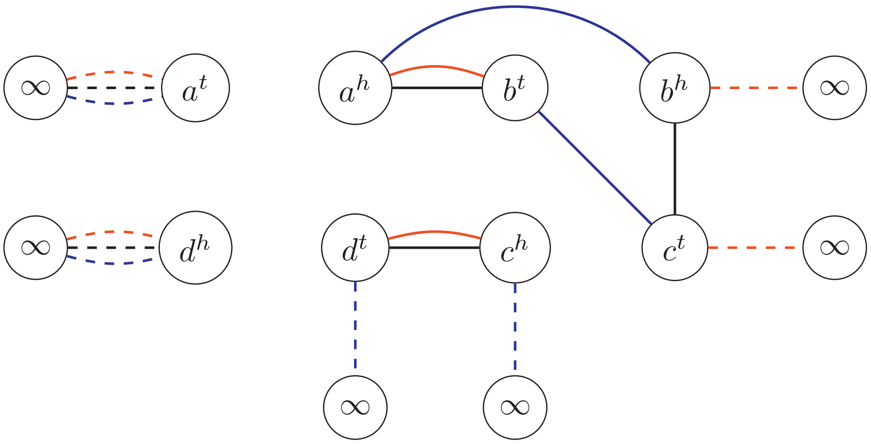


**Fig. 2.** Breakpoint graph $\mathbb{BG}$ of genomes $A = [a, b, c, d]$ (black edges), $B = [a, b][c, d]$ (red edges), and $C = [a, -b, c][d]$ (blue edges). Regular edges in each genome are shown as solid, while irregular edges are shown as dashed.

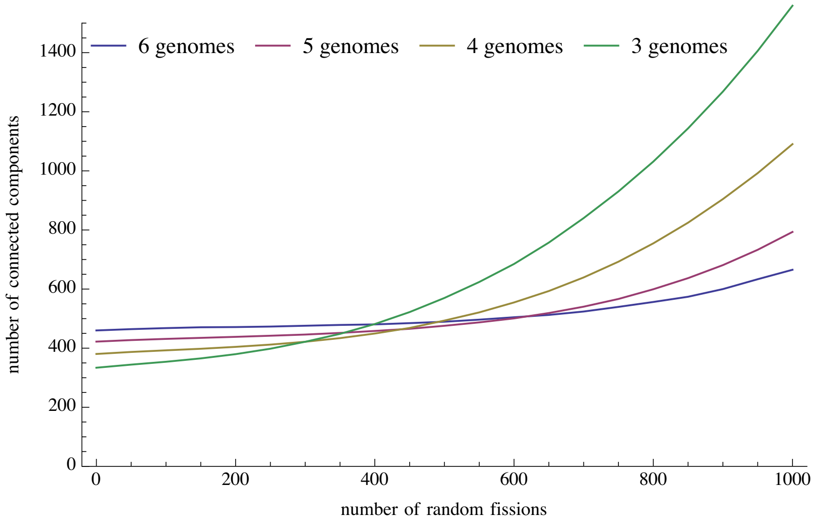


**Fig. 3.** Averaged number of connected components in the breakpoint graphs of multiple mammalian genomes fragmented by random "fissions". Statistics for groups of 3, 4, 5, and 6 genomes is shown in green, brown, purple, and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 2.3. Scaffold assembly algorithm

Our scaffold assembly algorithm takes as an input a set of fragmented genomes, whose scaffolds are represented as sequences of genes. In the current study, we focus only on gene families that are present in each genome exactly once.

The first step of our algorithm is a construction of the breakpoint graph $\mathbb{BG}$ of the given genomes.

From the breakpoint graph perspective, scaffold assembly corresponds performing "fusions" in $\mathbb{BG}$, i.e., adding new regular edges (*assembly edges*) connecting vertices that represent scaffold ends. Since the connected components in the breakpoint graph are robust with respect to genome fragmentation, our algorithm adds assembly edges only within existing connected components and thus preserves them. So the second step of our algorithm identifies pairs of *matching vertices* that will then be connected by assembly edges (Fig. 4). Namely, vertices $x, y : x \neq y$ form a matching pair in genome $P$ if they satisfy each of the following conditions:

connectivity: $x$ and $y$ belong to the same connected component $X$ of $\mathbb{BG}$;

extremity: there are multiedges $(x, \infty)$ and $(y, \infty)$ of multicolor $\{P\}$ in $\mathbb{BG}$;

uniqueness: there is no vertex $z \in X$, $z \neq x$, $z \neq y$ such that the multiedge $(z, \infty)$ has multicolor $\{P\}$; in other words, there are exactly two irregular edges of multicolor $\{P\}$ in $X$: $(x, \infty)$ and $(y, \infty)$.

```
input  : breakpoint graph $\mathbb{BG}$ on set $G$ of $k$ genomes
output: list of triples of $(x, y, P)$ where $x$ and $y$ is a matching pair in genome $P$
$result \leftarrow$ empty list;
$ccs \leftarrow$ connected components (sets of vertices) of $\mathbb{BG}$ on regular edges of all colors;
foreach $cc$ in $ccs$ do
    foreach genome $P$ in $G$ do
        $ies \leftarrow$ all irregular multiedges of multicolor $\{P\}$ connecting vertices from $cc$;
        if $|ies| = 2$ then
            $v1 \leftarrow$ regular vertex from first edge in $ies$;
            $v2 \leftarrow$ regular vertex from second edge in $ies$;
            if there is edge $(v1, v2)$ in $\mathbb{BG}$ then
                append $(v1, v2, P)$ to $result$;
            end
        end
    end
end
return $result$;
```

**Fig. 4.** Pseudocode for identification of matching pairs of vertices.

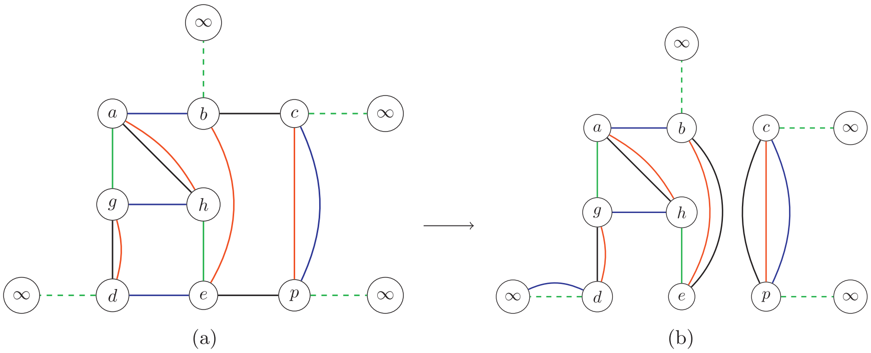


**Fig. 5. Left panel:** A connected component of the breakpoint graph of blue, black, red and green genomes. In this component the green genome has four irregular edges that correspond to fragment ends: $(b, \infty)$, $(c, \infty)$, $(d, \infty)$, and $(p, \infty)$. Since its endpoints $b, c, p$ do not satisfy the uniqueness property, while endpoint $d$ does not satisfy the extrimity property, our algorithm is not able to assemble any of corresponding fragments in the green genome. **Right panel:** A possible refinement of the connected component with MGRA with a genome rearrangement in the *black* genome. It results in a split of the connected component into two, which makes vertices $c$ and $p$ obtain the uniqueness property with respect to the green genome.

reliability: there exists a multiedge $(x, y)$ (of any multicolor) in $\mathbb{BG}$.

The **connectivity** condition preserves connected components in the breakpoint graph. The **extremity** condition ensures that each of $x$, $y$ represents a scaffold end in the genome $P$ but not in any other genome.[6] The **uniqueness** condition ensures that there is a unique way to create an assembly edge for genome $P$ inside the connected component $X$: if there is another multiedge $(z, \infty)$ of multicolor $\{P\}$, it would be unclear which pair out of $x$, $y$, $z$ to connect with an assembly edge. The **reliability** condition ensures that the new adjacency created by an assembly edge $(x, y)$ in genome $P$ is already present in some other genome(s).[7]

Once we obtained a list of matching vertex pairs for every genome $P$, we perform assembly of the corresponding fragment ends in $P$.

### 2.4. Integration with MGRA framework

We remark that our algorithm can be integrated with the MGRA framework Alekseyev and Pevzner (2009), which performs rearrangement analysis of multiple genomes, identifies reliable genome rearrangements and transforms their breakpoint graph into an *identity* breakpoint graph (of a single ancestral genome). The identity breakpoint graph consists of trivial multicycles, each forming its own connected component. In the process of this transformation MGRA can only break the connected components of the breakpoint graph into smaller ones, which can be viewed as a refinement of the original connected components. As a result, MGRA can make possible for two vertices to obtain the uniqueness property after a number of genome rearrangements (Fig. 5).

However, since the irregular edges in the breakpoint graph after a number of genome rearrangements may no longer correspond to fragment ends, the **extremity** condition does not anymore imply that $x$ and $y$ are fragment ends. Therefore, to integrate the scaffold assembly with MGRA, we modify the **extremity** condition to additionally test if $x$ and $y$ correspond to fragment ends in the genome $P$. By similar reasons, the **reliability** condition for vertices $x$, $y$ should be tested in the original breakpoint graph. So if $\mathbb{BG}$ denotes the original breakpoint graph, while $\mathbb{BG}'$ denotes this graph after a number of genome rearrangements performed by MGRA, then a matching pair $(x, y)$ in $\mathbb{BG}'$ should satisfy the following conditions:

connectivity': $x$ and $y$ belong to the same connected component $X$ of $\mathbb{BG}'$;

extremity': there are multiedges $(x, \infty)$ and $(y, \infty)$ of multicolor $\{P\}$ in both $\mathbb{BG}$ and $\mathbb{BG}'$;

uniqueness': for any vertex $z \in X$ such that $z \neq x$, $z \neq y$, and the multiedge $(z, \infty)$ in $\mathbb{BG}'$ has multicolor $\{P\}$, the multiedge $(z, \infty)$ either is not present in $\mathbb{BG}$ or has multicolor different from $\{P\}$;

reliability': there exists a multiedge $(x, y)$ (of any multicolor) in $\mathbb{BG}$;

Integration with MGRA allows us to obtain more matching vertices (as compared to what we can recover from the original breakpoint graph). We also take into consideration all pairs of vertices that are endpoints of fusions reported by MGRA. If such vertices correspond to fragment ends, we interpret their fusion as assembly the corresponding fragments.

## 3. Results and discussion

### 3.1. Artificially fragmented genomes

We start evaluation of the proposed scaffold assembly algorithm with running it on artificially fragmented mammalian genomes. We use the same set of six mammalian genomes and two different approaches for fragmenting them: random fragmentation and fragmentation based on repeats in the genomes. The random fragmentation allows us to overcome the lack of information about genome fragmentation mechanism. However, we may have better insight in the fragmentation model, if we assume that genome scaffolds were obtained from a conventional genome assembler having difficulties in reconstruction of the order of long DNA repeats. In this case, it becomes realistic to fragment the genomes based on locations of such repeats.

*Random fragmentation.*

To create instances of randomly fragmented genomes, we applied $k$ random artificial "fissions" to each of the genomes. For each value of $k$, we created 10 different sets of fragmented genomes, executed our algorithm on each of the sets (both with and without MGRA integration), and reported the following normalized values (averaged over the 10 sets):

[6] Under random fragmentation, it is more likely for two genomes to share a common chromosome end than a scaffold end, which is not a chromosome end. So, if $x$ or $y$ is a scaffold end in two or more genomes, it is more likely for this vertex to represent a chromosome end.

[7] This condition is optional. It shall be utilized when the given genomes are closely related; however if the genomes are rather diverse, this condition may result in only small number of conservative assembly edges.

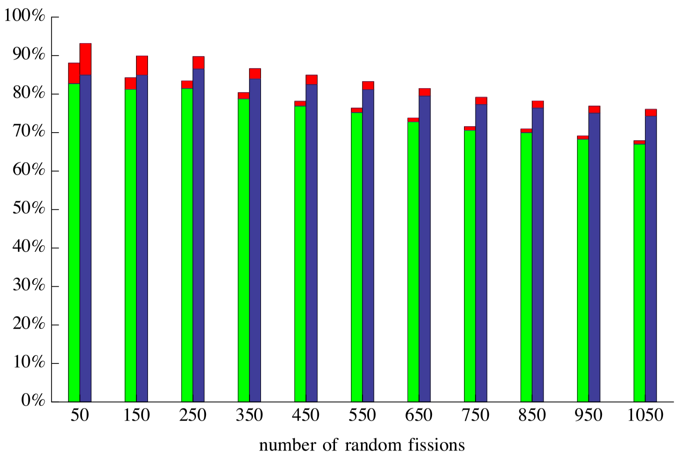


**Fig. 6.** Accuracy of the proposed algorithm on artificially fragmented mammalian genomes both with (blue bars) and without (green bars) integration with MGRA. For each number of random "fissions" (with step 100) applied to every genome, blue and green bars give normalized true positives, while red bar gives normalized false positives. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *true positive*: $|A \cap B|/|A \cup B| \cdot 100\,\%$, the percentage of assembly edges that correspond to the breakpoints of artificial "fissions".
- *false positive*: $|A \setminus B|/|A \cup B| \cdot 100\,\%$, the percentage of assembly edges that do not correspond to breakpoints of artificial "fissions" (such assembly edges may rather indicate the locations of actual fissions in mammalian evolution).

Here $A$ is the set of pairs of fragment ends assembled by our algorithm, and $B$ is the set of pairs of fragment ends resulted from the simulated fissions.

From the evaluation results in Fig. 6, we conclude that while our algorithm is not able to reconstruct complete genomes where fragmentation is high, it is able to reconstruct genomes almost completely, when the fragmentation is low. While the true positive rate of the assembly results decreases as fragmentation raises, the results still remain highly reliable with low false positive rate. This is not an unexpected property of our algorithm since it is based

**Table 1**
Statistics on the number of nonempty scaffolds in anophelinae genomes before and after gene family filtration.

| Genomes | # Scaffolds | |
|---|---|---|
| | Before filtration | After filtration |
| *An. gambiae* | 6 | 6 |
| *An. arabiensis* | 340 | 95 |
| *An. quadriannulatus* | 647 | 306 |
| *An. merus* | 1078 | 816 |
| *An. dirus* | 302 | 124 |
| *An. albimanus* | 57 | 39 |

**Table 2**
Statistics on the number of reported scaffold assemblies, both with and without integration with MGRA.

| Genomes | # Assemblies | |
|---|---|---|
| | Without MGRA | Integrated with MGRA |
| *An. gambiae* | 0 | 0 |
| *An. arabiensis* | 6 | 10 |
| *An. quadriannulatus* | 75 | 91 |
| *An. merus* | 466 | 550 |
| *An. dirus* | 30 | 45 |
| *An. albimanus* | 6 | 10 |

**Table 3**
Statistics on the number of fragments and coverage (in parenthesis) after removing all repeats of length at least *L* in the each of the six mammalian genomes.The column *Orth* accounts the fragments that contain at least one gene. Column *IDOrth* accounts for fragments that contain at least one non-duplicated gene. Similarly, column *UOrth* accounts for fragments that contain at least one gene present in every genome exactly once.

| *L* | Chimpanzee | | | Dog | | | Human | | | Macaca | | | Mouse | | | Rat | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth |
| 1K | 10631<br>(26.04) | 6234<br>(19.6%) | 1636<br>(7.7%) | 9067<br>(33.2%) | 4807<br>(24.2%) | 1547<br>(9.9%) | 20318<br>(36.8%) | 5539<br>(16.4%) | 1779<br>(6.9%) | 10255<br>(33.1%) | 5345<br>(22.8%) | 1673<br>(9.7%) | 12975<br>(38.3%) | 4293<br>(21.1%) | 1597<br>(10.1%) | 10399<br>(34.9%) | 5011<br>(27.3%) | 1469<br>(10.1%) |
| 1.5K | 8994<br>(39.01) | 5577<br>(30.2%) | 1939<br>(15.6%) | 7150<br>(52.1%) | 4066<br>(40.6%) | 1663<br>(21.6%) | 16835<br>(49.7%) | 5256<br>(25.9%) | 2147<br>(14.0%) | 8586<br>(47.8%) | 4884<br>(35.8%) | 1943<br>(19.4%) | 11088<br>(49.6%) | 3994<br>(30.4%) | 1868<br>(18.2%) | 8742<br>(46.4%) | 4246<br>(37.6%) | 1672<br>(18.8%) |
| 2K | 7496<br>(48.51) | 4781<br>(38.8%) | 2016<br>(23.8%) | 5579<br>(65.9%) | 3308<br>(53.8%) | 1643<br>(34.5%) | 13797<br>(60.6%) | 4769<br>(34.9%) | 2251<br>(21.9%) | 6865<br>(60.8%) | 4213<br>(48.5%) | 1978<br>(30.5%) | 9845<br>(56.4%) | 3697<br>(36.1%) | 2053<br>(25.2%) | 7527<br>(54.1%) | 3682<br>(44.2%) | 1754<br>(26.4%) |
| 2.5K | 6497<br>(55.9%) | 4264<br>(45.5%) | 2045<br>(30.5%) | 4389<br>(75.8%) | 2653<br>(64.2%) | 1518<br>(46.9%) | 11727<br>(67.5%) | 4425<br>(41.2%) | 2306<br>(28.2%) | 5385<br>(71.9%) | 3463<br>(59.2%) | 1848<br>(41.7%) | 9061<br>(60.5%) | 3530<br>(39.6%) | 2184<br>(29.9%) | 6651<br>(59.7%) | 3266<br>(49.1%) | 1759<br>(32.7%) |
| 3K | 5789<br>(64.9%) | 3899<br>(54.6%) | 2012<br>(38.9%) | 3605<br>(81.9%) | 2246<br>(70.9%) | 1382<br>(55.2%) | 10226<br>(72.3%) | 4121<br>(45.9%) | 2301<br>(33.1%) | 4273<br>(79.3%) | 2894<br>(67.5%) | 1682<br>(50.6%) | 8448<br>(63.5%) | 3368<br>(42.2%) | 2199<br>(33.1%) | 5997<br>(63.9%) | 2994<br>(52.4%) | 1730<br>(37.3%) |
| 3.5K | 5197<br>(69.3%6) | 3589<br>(58.6%) | 1965<br>(43.4%) | 3071<br>(85.6%) | 1908<br>(74.7%) | 1258<br>(61.8%) | 9005<br>(75.9%) | 3875<br>(50.3%) | 2278<br>(37.4%) | 3368<br>(84.9%) | 2377<br>(74.4%) | 1487<br>(58.6%) | 7909<br>(66.2%) | 3204<br>(44.3%) | 2189<br>(35.8%) | 5350<br>(67.9%) | 2725<br>(55.7%) | 1661<br>(41.55) |
| 4K | 4694<br>(73.6%) | 3292<br>(63.1%) | 1896<br>(48.1%) | 2683<br>(88.4%) | 1698<br>(78.1%) | 1175<br>(66.8%) | 8007<br>(79.2%) | 3631<br>(54.1%) | 2221<br>(41.2%) | 2635<br>(89.3%) | 1940<br>(80.1%) | 1285<br>(65.8%) | 7538<br>(68.3%) | 3125<br>(45.9%) | 2223<br>(38.1%) | 4779<br>(71.7%) | 2502<br>(58.9%) | 1601<br>(45.6%) |

on the robustness of the connected components of the breakpoint graph. Integration with the MGRA framework further yields additional number of highly reliable fragment assemblies.

*Repeat-based fragmentation.*

To create instances of repeat-based fragmented genomes, we removed all repeats longer that a fixed number of basepairs (from 1 K to 4 K bp with the step of 0.5 K) and partitioned the genomes into fragments with no long repeats. We used the same set of six mammalian genomes, for which we obtained the repeats locations from RepeatMasker Smit et al. (2010) database. We performed the following three experiments:

(i) de novo assembly of multiple genomes: all six genomes are fragmented (Table 3).
(ii) assembly of multiple genomes with a single reference: all genomes, but *dog* (the only representative of the carnivore clade) are fragmented (Table 4).
(iii) assembly of a single genome with multiple references: only *dog* genome is fragmented (Table 5).

In each experiment we considered only fragments that contain genes that are present exactly once in each genome. We evaluated the proposed algorithm in the same way as in the random fragmentation experiment, both with and without MGRA integration (Fig. 7).

Experiments (i) and (ii) demonstrate that while in the presence of a reference genome our algorithm yields more true fragment adjacencies, it still performs relatively well in the case, when no reference is known. Experiment (iii) shows that our algorithm can be used as a highly reliable step for assembly of a single fragmented genome, when several complete reference genomes are known. Since DNA repeats are subject to genome rearrangements in the course of evolution, integration with MGRA yields additional true adjacencies.

### 3.2. *Anophelinae genomes*

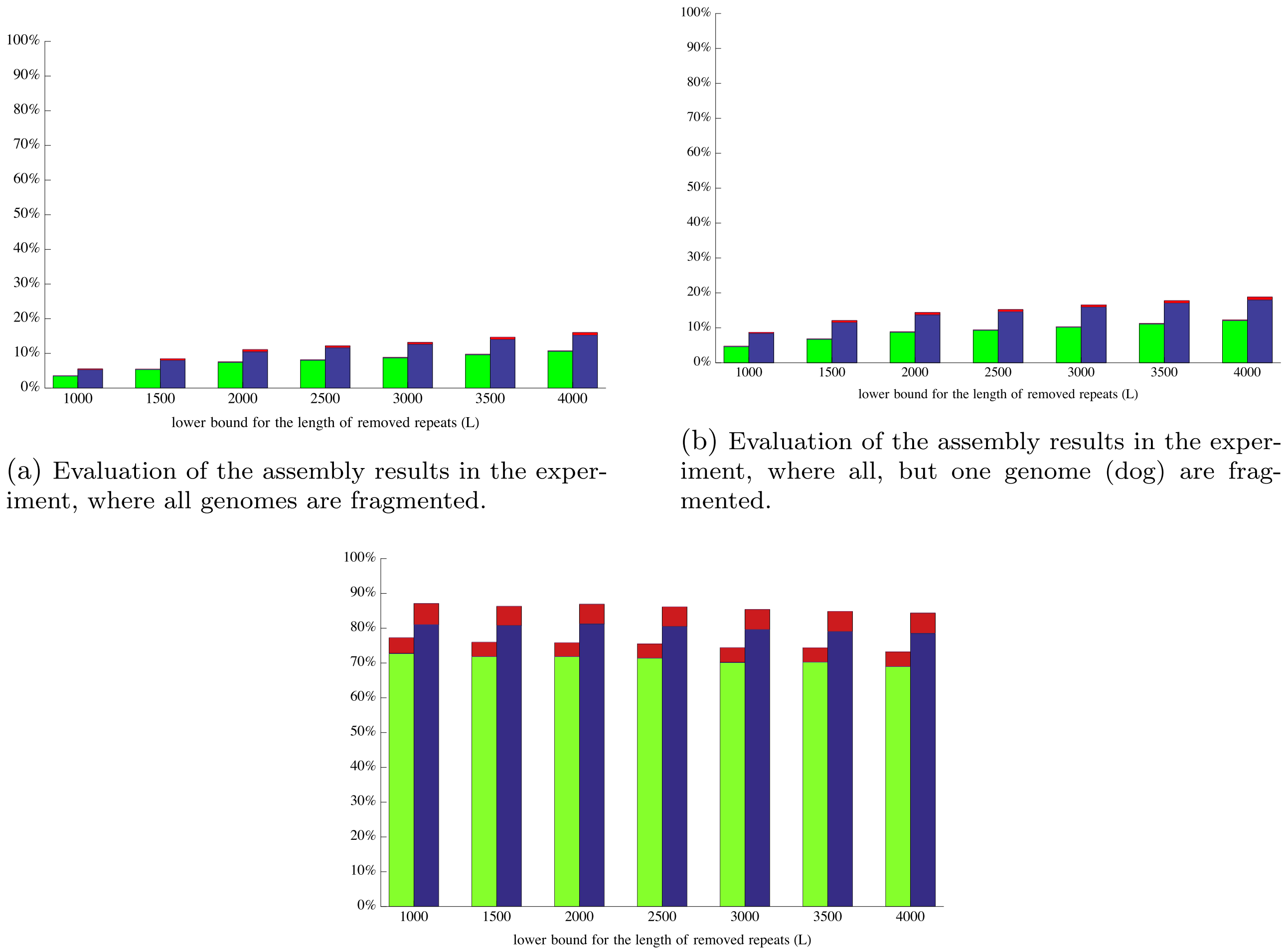
The second evaluation of our scaffold assembly algorithm was performed on highly fragmented genomes from anophelinae subfamily, followed by comparison of the results to a reference-based assembly approach. Namely, we considered six anophelinae genomes: *Anopheles gambiae*, *Anopheles arabiensis*, *An. quadriannulatus*, *An. merus*, *Anopheles dirus*, and *Anopheles albimanus*, for which we constructed gene families using orthologous gene mapping from OrthoDB Waterhouse et al. (2013). We then filtered out all gene families that are not present exactly once on each given genome, thus limiting ourselves to the case of uniform gene content across the genomes. After such filtration each genome was represented as sequences of the same 6837 genes. We remark that filtration eliminated all genes from some scaffolds and thus we exclude such scaffolds from assembly. Table 1 gives the scaffold statistics for anophelinae genomes before and after filtration.

Order and orientation of gene families in each given genome was determined from the corresponding GFF3 annotation obtained from VectorBase Megy et al. (2012), where a gene is represented by a sequence of coding exons of various length. We define the gene coordinate in a genomic fragment as the mean coordinate of all its coding exons start/end coordinates (i.e., (start + end)/2 averaged over all exons). Table 2 reports the number of scaffold assemblies obtained by the proposed algorithm.

As mentioned above, we compared our assembly results to another anophelinae study (*comparison study*) led by Dr. Igor Sharakhov at Virginia Tech University. The comparison study performed analysis of *An. gambiae*, *An. arabiensis* genomes from the same source, where *An. gambiae* represents a complete genome,

**Table 4**
Statistics on the number of fragments and coverage (in parenthesis) after removing all repeats of length at least *L* in the each of the six mammalian genomes, except *dog*.The column *Orth* accounts the fragments that contain at least one gene. Column *IDOrth* accounts for fragments that contain at least one non-duplicated gene. Similarly, column *UOrth* accounts for fragments that contain at least one gene present in every genome exactly once.

| *L* | Chimpanzee | | | Dog | Human | | | Macaca | | | Mouse | | | Rat | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Orth | IDOrth | UOrth | | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth | Orth | IDOrth | UOrth |
| 1K | 10613<br>(26.1%) | 6223<br>(19.6%) | 1728<br>(8.1%) | 40<br>(100%) | 20381<br>(36.9%) | 5528<br>(15.9%) | 1871<br>(7.4%) | 1055<br>(33.2%) | 5334<br>(22.8%) | 1766<br>(10.1%) | 12975<br>(38.3%) | 4283<br>(21.0%) | 1676<br>(10.5%) | 10399<br>(34.9%) | 4988<br>(27.3%) | 1538<br>(10.4%) |
| 1.5K | 8994<br>(39.1%) | 5572<br>(30.2%) | 2015<br>(16.1%) | 40<br>(100%) | 16835<br>(49.8%) | 5252<br>(25.4%) | 2221<br>(14.5%) | 8586<br>(47.8%) | 4878<br>(35.8%) | 2013<br>(20.1%) | 11088<br>(49.6%) | 3988<br>(30.4%) | 1937<br>(18.7%) | 8742<br>(46.4%) | 4230<br>(37.5%) | 1728<br>(21.5%) |
| 2K | 7469<br>(48.5%) | 4778<br>(38.8%) | 2078<br>(24.3%) | 40<br>(100%) | 13797<br>(60.6%) | 4767<br>(34.5%) | 2312<br>(22.4%) | 6865<br>(60.8%) | 4209<br>(48.5%) | 2034<br>(31.2%) | 9845<br>(56.5%) | 3691<br>(36.1%) | 2113<br>(25.7%) | 7527<br>(54.1%) | 3671<br>(44.1%) | 1805<br>(29.2%) |
| 2.5K | 6497<br>(55.9%) | 4262<br>(45.5%) | 2092<br>(30.9%) | 40<br>(100%) | 11727<br>(67.6%) | 4424<br>(41.2%) | 2361<br>(28.6%) | 5385<br>(71.9%) | 3460<br>(59.2%) | 1891<br>(42.2%) | 9061<br>(60.5%) | 3525<br>(39.5%) | 2232<br>(30.4%) | 6651<br>(59.7%) | 3260<br>(49.1%) | 1796<br>(35.4%) |
| 3K | 5789<br>(69.9%) | 3897<br>(54.6%) | 2050<br>(39.3%) | 40<br>(100%) | 10226<br>(72.3%) | 4120<br>(45.9%) | 2340<br>(33.4%) | 4723<br>(79.3%) | 2893<br>(67.5%) | 1712<br>(51.1%) | 8448<br>(63.6%) | 3363<br>(42.1%) | 2236<br>(33.4%) | 5997<br>(63.9%) | 2990<br>(52.4%) | 1762<br>(39.9%) |
| 3.5K | 5197<br>(69.2%) | 3588<br>(58.6%) | 1996<br>(43.8%) | 40<br>(100%) | 9005<br>(75.9%) | 3875<br>(50.3%) | 2307<br>(37.8%) | 3368<br>(84.9%) | 2376<br>(75.4%) | 1509<br>(59.1%) | 7909<br>(66.2%) | 3201<br>(44.3%) | 2215<br>(36.1%) | 5350<br>(67.9%) | 2723<br>(55.7%) | 1690<br>(44.2%) |
| 4K | 4694<br>(73.6%) | 3291<br>(63.1%) | 1919<br>(48.3%) | 40<br>(100%) | 8007<br>(79.2%) | 3630<br>(54%) | 2249<br>(41.5%) | 2635<br>(89.3%) | 1940<br>(80%) | 1298<br>(66.2%) | 7538<br>(68.3%) | 3122<br>(45.9%) | 2247<br>(38.4%) | 4779<br>(71.7%) | 2501<br>(58.9%) | 1625<br>(48.1%) |



(a) Evaluation of the assembly results in the experiment, where all genomes are fragmented.

(b) Evaluation of the assembly results in the experiment, where all, but one genome (dog) are fragmented.

(c) Evaluation of the assembly results in the experiment, where only one genome (dog) is fragmented.

**Fig. 7.** Accuracy of the proposed algorithm on artificially fragmented six mammalian genomes, that were broken at the positions of repeats of length at least *L*, with (blue bars) and without (green bars) integration with MGRA. For each value of *L* (with step 500 bp) blue and green bars give the *true positive* rate, while red bars give *false positive* rate for assembly results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

while *An. arabiensis* exposes rather high fragmentation. The genome data preparation was similar to ours. The relationships between these genes and their order on scaffold were visualized in genoPlotR Guy et al. (2010) and further compared to the cytogenetic Holt et al. (2002) and physical George et al. (2010) maps identifying breakpoints of fixed reversals. The *An. gambiae* genome assembly was used as a reference for scaffolding in *An. arabiensis*.

Among 10 assemblies in *An. arabiensis* genome identified by our algorithm, the comparison study was able to identify and confirm 6. For example, our algorithm suggested assembly of scaffolds

**Table 5**
Statistics on the number of fragments and coverage (in parenthesis) after removing all repeats of length at least *L* in the *dog* genome.The column *Orth* accounts the fragments that contain at least one gene. Column *IDOrth* accounts for fragments that contain at least one non-duplicated gene. Similarly, column *UOrth* accounts for fragments that contain at least one gene present in every genome exactly once.

| *L* | Chimpanzee | Dog | | | Human | Macaca | Mouse | Rat |
|---|---|---|---|---|---|---|---|---|
| | | Orth | IDOrth | UOrth | | | | |
| 1K | 26<br>(100%) | 9067<br>(33.3%) | 4675<br>(23.7%) | 3933<br>(21.4%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 1.5K | 26<br>(100%) | 7150<br>(52.1%) | 3990<br>(40.1%) | 3480<br>(37.4%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 2K | 26<br>(100%) | 5579<br>(65.9%) | 3265<br>(53.4%) | 2886<br>(50.4%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 2.5K | 26<br>(100%) | 4389<br>(75.8%) | 2628<br>(63.8%) | 2346<br>(60.9%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 3K | 26<br>(100%) | 3605<br>(81.9%) | 2228<br>(70.7%) | 2013<br>(68.1%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 3.5K | 26<br>(100%) | 3071<br>(85.6%) | 1896<br>(74.5%) | 1717<br>(72.3%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |
| 4K | 26<br>(100%) | 2683<br>(88.4%) | 1687<br>(77.9%) | 1533<br>(75.8%) | 24<br>(100%) | 22<br>(100%) | 21<br>(100%) | 22<br>(100%) |

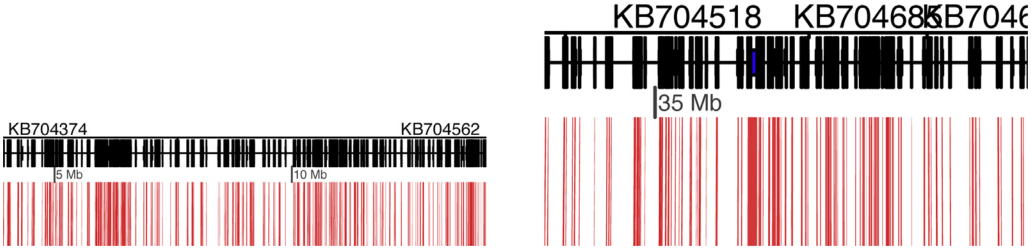


**Fig. 8.** genoPlotR visualization of gene order on scaffolds KB704374, KB704562, KB704518, and KB704685 for *A. arabiensis* genome. Courtesy of Dr. Igor Sharakhov.

KB704562 and KB704374 as well as of scaffolds KB704518 and KB704685 in the *A. arabiensis* genome, which was also identified by the comparison study with the gene reference-based (Fig. 8).

## 4. Conclusions

In current study, we proposed a scaffold assembly algorithm based on the genome rearrangement analysis, which can be used to assemble highly fragment genomes. The proposed algorithm relies on the properties of breakpoint graph of multiple genomes and can be further integrated with the MGRA framework. We evaluated the proposed algorithm by testing it on both real and simulated genomic data. In both cases, it significantly reduced fragmentation of the genomes and demonstrated high reliability.

While the proposed algorithm relies on unique gene content, we are currently expanding the algorithm with support of non-unique (inserted/deleted or duplicated) genes, which will potentially lead to even better quantity and quality of the scaffold assembly results. We implemented the proposed algorithm in a prototype software, which we plan to make user-friendly and publicly available in near future.

## Acknowledgements

The paper was prepared in collaboration with the Anopheles Genomes Cluster Consortium Neafsey et al. (2013). We are particularly indebted to Dr. Robert Waterhouse at MIT for the orthologous gene mapping data for Anopheles genomes. We are further thankful to Dr. Igor Sharakhov and members of his laboratory Xiaofang Jiang and Brantley Hall at Virginia Tech University for cross-validation of our scaffolding results. The first and fourth authors were supported by the National Science Foundation under grant no. IIS-1253614.

## Appendix A. Anopheles Genomes Cluster Consortium

Daniel E. Neafsey, George K. Christophides, Frank H. Collins, Scott J. Emrich, Michael C. Fontaine, William Gelbart, Matthew W. Hahn, Paul I. Howell, Fotis C. Kafatos, Daniel Lawson, Marc A. T. Muskavitch, Robert Waterhouse, and Nora J. Besansky.

## References

Alekseyev, Max A., Pevzner, Pavel A., 2009. Breakpoint graphs and ancestral genome reconstructions. Genome Res. 19 (5), 943–957.

Alekseyev, Max A., Pevzner, Pavel A., 2008. Multi-break rearrangements and chromosomal evolution. Theor. Comput. Sci. 395 (2-3), 193–202.

Chitsaz, H., Yee-Greenbaum, J.L., Tesler, G., Lombardo, M.J., Dupont, C.L., Badger, J.H., Novotny, M., Rusch, D.B., Fraser, L.J., Gormley, N.A., Schulz-Trieglaff, O., Smith, G.P., Evers, D.J., Pevzner, P.A., Lasken, R.S., 2011. Efficient de novo assembly of single-cell bacterial genomes from short-read data sets. Nat. Biotechnol. 29 (10), 915–921.

Collins, F.S., Weissman, S.M., 1984. Directional cloning of dna fragments at a large distance from an initial probe: a circularization method. Proc. Natl. Acad. Sci. U. S. A. 81 (21), 6812–6816.

Feuk, Lars, Carson, Andrew R., Scherer, Stephen W., 2006. Structural variation in the human genome. Nat. Rev. Genet. 7 (2), 85–97.

George, Phillip, Sharakhova, Maria V., Sharakhov, Igor V., 2010. High-resolution cytogenetic map for the african malaria vector anopheles gambiae. Insect Mol. Biol. 19 (5), 675–682.

Guy, Lionel, Kultima, Jens Roat, Andersson, Siv G.E., 2010. genoPlotR: comparative gene and genome visualization in R. Bioinformatics 26 (18), 2334–2335.

Holt, Robert A., Mani Subramanian, G., Halpern, Aaron, Sutton, Granger G., Charlab, Rosane, Nusskern, Deborah R., Wincker, Patrick, Clark, Andrew G., Ribeiro, JoséM.C., Wides, Ron, et al., 2002. The genome sequence of the malaria mosquito anopheles gambiae. Science 298 (5591), 129–149.

Hunt, Martin, Newbold, Chris, Berriman, Matthew, Otto, Thomas D., 2014. A comprehensive evaluation of assembly scaffolding tools. Genome Biol. 15 (3), R42.

Kasprzyk, A., 2011. BioMart: driving a paradigm change in biological data management. Database 2011.

Megy, Karine, Emrich, Scott J., Lawson, Daniel, Campbell, David, Dialynas, Emmanuel, Hughes, Daniel S.T., Koscielny, Gautier, Louis, Christos, MacCallum, Robert M., Redmond, Seth N., et al., 2012. VectorBase: improvements to a bioinformatics resource for invertebrate vector genomics. Nucleic Acids Res. 40 (D1), D729–D734.

Neafsey, Daniel E., Christophides, George K., Collins, Frank H., Emrich, Scott J., Fontaine, Michael C., Gelbart, William, Hahn, Matthew W., Howell, Paul I., Kafatos, Fotis C., Lawson, Daniel, et al., 2013. The evolution of the anopheles 16 genomes project. G3: Genes Genomes Genetics 3 (7), 1191–1194.

Nikolenko, Sergey I., Korobeynikov, Anton, Alekseyev, Max A, 2013. BayesHammer: Bayesian clustering for error correction in single-cell sequencing. BMC Genomics 14, S7.

Smit, A.F.A., Hubley, R., Green, P., 2010. RepeatMasker Open-3.0, 1996–2010, http://www.repeatmasker.org.

Talkowski, Michael E., Ordulu, Zehra, Pillalamarri, Vamsee, Benson, Carol B., Blumenthal, Ian, Connolly, Susan, Hanscom, Carrie, Hussain, Naveed, Pereira, Shahrin, Picker, Jonathan, et al., 2012. Clinical diagnosis by whole-genome sequencing of a prenatal sample. N. Engl. J. Med. 367 (23), 2226–2232.

Waterhouse, Robert M., Tegenfeldt, Fredrik, Li, Jia, Zdobnov, Evgeny M., Kriventseva, Evgenia V., 2013. OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. Nucleic Acids Res. 41 (D1), D358–D365.

Yancopoulos, Sophia, Attie, Oliver, Friedberg, Richard, 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21 (16), 3340–3346.