www.sciencemag.org/cgi/content/full/science.aal3327/DC1

Supplementary Materials for

De novo assembly of the *Aedes aegypti* genome using Hi-C yields chromosome-length scaffolds

Olga Dudchenko, Sanjit S. Batra, Arina D. Omer, Sarah K. Nyquist, Marie Hoeger, Neva C. Durand, Muhammad S. Shamim, Ido Machol, Eric S. Lander, Aviva Presser Aiden, Erez Lieberman Aiden*

*Corresponding author. Email: erez@erez.com

Published 23 March 2017 on *Science* First Release
DOI: 10.1126/science.aal3327

This PDF file includes:

Materials and Methods
Figs. S1 to S21
Tables S1 to S14
Captions for tables S15 to S20
References

Other Supplementary Materials for this manuscript includes the following:
(available at www.sciencemag.org/cgi/content/full/science.aal3327/DC1)

Tables S15 to S20 (Excel)

Materials and Methods

In situ Hi-C was performed as described previously (*5*) using two adult female *Ae. aegypti* mosquitoes. The resulting libraries were sequenced using an Illumina NextSeq instrument to yield approximately 40X coverage of the *Ae. aegypti* genome. *In situ* Hi-C was also performed on a female *Cx. quinquefasciatus* mosquito and sequenced using an Illumina NextSeq instrument to yield approximately 100X coverage of the *Cx. quinquefasciatus* genome. All three mosquitoes were obtained from Benzon Research (http://www.benzonresearch.com).

For the human assembly, we used a small fraction (100 million read pairs) of a dataset we recently published (HIC001, GEO accession GSM1551550, (*5*)).

Dependency of Hi-C contact frequency on one-dimensional distance between loci

Note that we use the term "Hi-C contact" as defined in (*5*) throughout the manuscript and the Supplementary Materials.

To illustrate the dependency of Hi-C contact frequency on the distance between loci, we analyzed the distribution of contacts in the HIC001 library, whose generation and initial analysis was reported in (*5*). In order to match the procedure used there, the analysis in this section was performed using hg19 as the human reference. HIC001 is also the library that was used to generate Hs2-HiC (see Main text).

We calculated the contact probability $I(s)$ using the method we employed in (*4*, *26*), focusing on chromosome 1 as a representative chromosome. It is helpful to define the "span" of an intrachromosomal contact as the distance between the two loci it connects. The relative intrachromosomal contact probabilities were estimated by taking the number of contacts whose span ranges from s-500 to s+500, and dividing by 1000*the total number of pairs of genomic positions separated by s on chromosome 1 (which is simply the length of chromosome 1, or 1,249,250,621, minus s) (*4*). The interchromosomal contact probability was estimated by taking the number of contacts between chromosome 1 and chromosome 10 and dividing by the product of the lengths of the two chromosomes in bp.

The typical distribution in absolute terms was calculated by generating a histogram using all contacts. Each intrachromosomal contact was assigned to a distance bin on the basis of its span: less than 10kb span; 10kb – 100kb span; 100kb – 1Mb span; 1Mb – 10Mb span; 10Mb – 100Mb span; and on the same chromosome, but more than 100Mb span. An additional bin included all interchromosomal contacts. This 7-bin histogram was then normalized.

Physical coverage achieved by Hi-C datasets

Again, we analyzed the HIC001 library. The physical coverage was computed by summing the span of all intrachromosomal Hi-C contacts. The sequencing coverage simply reflects the total amount of sequence generated using the HIC001 library, before any filtering.

Notably, the ratio between physical and sequence coverage for a Hi-C library – as for any paired-end library – depends greatly on the read length used for sequencing. So long as the reads are long enough to align well, reduction in read length – and the resulting reduction in sequence coverage – does not strongly impact the library's physical coverage. As such, Hi-C assembly costs can sometimes be reduced by sequencing shorter reads. In the case of HIC001, a 101bp paired-end sequencing strategy was used.

Pipeline description

The results reported in this paper are generated using a custom computational pipeline. The pipeline takes in as input the fasta file describing the draft assembly and the duplicate-free list of paired alignments of Hi-C reads to the draft fasta (merged_nodups.txt) as generated by the Juicer pipeline (*22*).

For the sake of generality, we will always refer to the inputs to our assembly algorithms as scaffolds. This is meant in a broad sense; the algorithms are agnostic as to whether the inputs are scaffolds (sequences that are permitted to contain gaps), or contigs (gap-free sequences). Input scaffolds can come from a wide variety of sources and technologies.

In characterizing sets of input scaffolds, it is also useful to define the "effective N50 length" of the input scaffolds. This is simply the N50 of the scaffolds after all misjoins they contain have been corrected. Of course, for a typical published set of scaffolds, the effective N50 is not known, since it may contain misjoins and other scaffolding errors that the authors were unaware of. Naturally, the actual N50 of the scaffold set furnishes an upper bound for the effective N50 – but the two are often not equal in practice.

The overall strategy of our assembler is to remove misjoins until the underlying scaffold set is largely free of misjoins. If there is a disparity between the actual N50 length of the input scaffolds and their effective N50 length, it will be greatly reduced by this step. After misjoin detection, the resulting input scaffolds are used to create the final ordered-and-oriented chromosome-length scaffolds.

An overview of the detailed workflow is schematically given in Fig. S1. We begin with a series of iterative steps whose goal is to eliminate misjoins in the input scaffolds. Each step begins with a scaffold pool (initially, this pool is the set of input scaffolds themselves); the scaffolding algorithm is used to order and orient these scaffolds; and the misjoin correction algorithm is applied to detect errors in the scaffold pool. Finally, the edited scaffold pool is used as an input for the next iteration of the misjoin correction algorithm. The ultimate effect of these iterations is to reliably detect misjoins in the input scaffolds without removing correctly assembled sequence. After the iterations are complete, the scaffolding algorithm is applied to the revised input scaffolds, and the output – a single "megascaffold" which concatenates all the chromosomes – is retained for post-processing. This post-processing includes four step: (i) a polishing algorithm, which is required for genomes in the Rabl configuration; (ii) a chromosome splitting algorithm, which is used to extract the chromosome-length scaffolds from the megascaffold; (iii) a sealing algorithm, which detects false positives in the misjoin correction process, and restores the erroneously removed sequence from the original scaffold; and (iv) a merge algorithm, which corrects misassembly errors due to undercollapsed heterozygosity in the input scaffolds. Step (ii) is omitted for genomes that are not in the Rabl configuration; step (iv) is omitted if the original scaffolds lack substantial undercollapsed heterozygosity.

Figure S2 explains the terminology we use to describe various scaffold subpopulations in the main text and supplement as well as relates these populations to various pipeline stages.

- *Preliminary Filtration*. First, a set of scaffolds (referred to as 'Tiny') is removed from the draft. Due to their small size, these scaffolds have relatively few Hi-C contacts, making them more difficult to reliably analyze. These are not processed further or included in the subsequent analysis.

- *Misjoin correction*. The input scaffolds are examined for Hi-C signal consistent with a misjoin. Scaffolds with no evidence of a misjoin are labeled as 'consistent.' Scaffolds with evidence of a misjoin are partitioned into segments; each segment is classified as either a 'consistent' scaffold or an 'inconsistent' scaffold on the basis of the Hi-C signal. Inconsistent scaffolds are not processed further. (This partitioning procedure makes it possible to remove errors while retaining the portions of a scaffold that are correctly assembled for subsequent steps.) Note that the terms above specifically refer to the results of the last round of misjoin correction, not the results of intermediate rounds.

- *Ordering and orientation.* The consistent scaffolds are then ordered and oriented; the results are polished (if needed), and chromosomes are extracted. As a result of this process, most of the consistent scaffolds are 'resolved', i.e., placed into a 'raw chromosomal scaffold', although a few remain 'unresolved'. The 'unresolved' scaffolds are not processed further. (However, note that although the unresolved scaffolds cannot be reliably localized within a chromosome-length scaffold, they can often be correctly associated with a particular chromosome. We do not do so in this manuscript.)

- *Overlap merging*. Pairs of resolved scaffolds in each raw chromosomal scaffold are examined for overlaps. When an overlap is detected, the scaffolds are merged. The result of this process is the final 'chromosome-length scaffold'. This step is crucial when assembling highly heterozygous genomes such as *Ae. aegypti* and *Cx. quinquefasciatus*.

The central components of the pipeline are the misjoin correction algorithm, the scaffolding algorithm, and the merging algorithm (see Figs. 1, S1). We describe each of the three blocks in detail below. We also include additional sections describing polishing, sealing and chromosome splitting algorithms.

All of these steps are fully automated and available as open-source code; more generally, furthermore, code is available which deploys each of these steps on input scaffolds from AaegL2, CpipJ2, and Hs1, producing AaegL4, CpipJ3, and Hs2-HiC (respectively) in a single click. The current version of the pipeline is written in the AWK programming language in combination with bash scripting. (We use AWK because the pipeline requires extremely rapid i/o.) It is optimized for speed using GNU Parallel shell tool (*27*), but can be run without parallelization. The pipeline is designed to take full advantage of the Juicebox visualization software for Hi-C data and produces Juicebox-compatible heatmaps as well as various supplementary annotation outputs at every step. Obligatory external dependencies are the Command line tools for Juicebox and Juicer (*22*, *23*) as well as the LASTZ sequence aligner (*28*).

I. Algorithm for misjoin correction

The misjoin correction algorithm consists of two parts: (i) misjoin detection and (ii) editing of input scaffolds (i.e., misjoin correction).

Our method for misjoin detection using Hi-C relies on the fact that sequences which have been erroneously concatenated in a scaffold form fewer contacts with one another than correctly joined sequences. This is because correctly joined sequences lie adjacent to one another in 1D, and are therefore proximate to one another in 3D, facilitating the formation of Hi-C contacts. Because they do not actually lie in close proximity in the one-dimensional (1D) sequence of the chromosome, misjoined sequences usually do not exhibit similar 3D proximity or similar contact frequency.

I.a. Calculating an expected model for contact frequency in the absence of an accurate genome

To detect this depletion in contact frequency, one must compare the observed contact frequency between adjacent genomic loci with an expected model that describes the contact frequency typically observed for correctly joined sequences. Given a genome assembly with chromosome-length scaffolds, calculating the expected frequency of contact for a typical pair of sequences at a particular distance during a given experiment is straightforward. Such calculations have been commonplace since our original paper on the Hi-C method (*4*).

However, the results of such contact probability calculations are influenced by disparate factors, ranging from the organism of interest, the cell population interrogated, the details of the experimental approach, the particular computational methods used to analyze the data, and seemingly random inter-experimental variability. For this reason, expected models derived from a particular experiment in a particular cell population in a particular organism cannot be reliably applied to all experiments in all cell populations in all species. Thus, in the absence of a genome assembly with chromosome-length scaffolds, it is unclear how to determine the relationship between contact probability and distance even if Hi-C data are available. To the best of our knowledge, no such calculations have been performed in the literature to date.

A second challenge is that the contact probability between a pair of loci varies greatly, with frequent “jackpot” effects where the number of contacts is markedly enhanced with respect to the background model. This variability makes raw contact probability a very noisy indicator of the presence of a misjoin.

To overcome these challenges, we developed a method that estimates contact probability, as a function of genomic distance, using data from a Hi-C experiment without utilizing a high quality genome. Instead, our method only assumes the availability of a collection of scaffolds that may be short and contain numerous errors. Specifically, we show that, even in this scenario, it is possible to calculate a lower bound for the expected number of contacts between a pair of loci at a given distance. Our estimation scheme relies on the fact that the frequency of contact between a pair of loci tends to decrease as the 1D distance between the loci increases. For this reason, pixels closer to the diagonal of a Hi-C matrix (which reflect contact frequency between loci that are nearby in 1D) tend to have higher contact counts than pixels further away from the diagonal.

Consider a $N\times N$ Hi-C contact matrix *M* generated using a known, correct reference genome (Fig. S3). To do so, the genome has been partitioned into *N* loci of fixed length that is matrix resolution r (measured in base pairs). Each pixel M_{ij} corresponds to all contacts between a pair of loci (in this case, the i^{th} locus and the j^{th} locus). Of course, *N* is simply the genome length divided by the matrix resolution, *r*. Note that, in such a setting, it is often convenient to measure 1D distance in terms of loci (which are all of fixed size *r*), which correspond to rows and columns of the matrix, rather than in terms of base pairs.

In such a matrix, we can consider the set of pixels that derive from pairs of loci that are within b loci of one another, i.e. the pixels M_{ij} such that $(i-b \ \leq j \leq i+b)$. Our principal goal is to estimate the function $Q(b)$, which is the minimum value of all these pixels: $Q(b)=\min(M_{ij}), i-b \ \leq j \leq i+b$. This function provides a lower bound for the values M_{ij} for pixels within *b* of the diagonal. $Q(b)$ is of obvious utility in identifying misjoins, for the following reason: if we were to align the Hi-C data against an incorrect reference genome, containing numerous misjoins, the presence of a value lower than $Q(b)$ within b pixels of the diagonal would indicate the presence of a misjoin at that position with complete certainty.

Before we address the estimation of $Q(b)$ in the general case, it is worth considering an idealized example. In Figure S3 (A) we show an idealized Hi-C matrix M' where contact probability decreases monotonically as the distance between a pair of loci increases, and the shape of this decay does not vary across the genome.

Notably, in such a matrix, the fraction of pixels M_{ij} that derive from pairs of loci that are within b loci of one another $(i-b \ \leq j \leq i+b)$ can be calculated by simply summing the lengths of the principal diagonal and $2\times b$ non-principal diagonals, and dividing by the size of the matrix as a whole (*N*^2). This yields:

$$F(b)=\left(N+b*(2N-b-1)\right)/N^2.$$

Thus, if we select a pixel from the matrix *M* at random, the probability that the pixel lies within *b* loci of the diagonal is exactly *F.*

Similarly, it is possible to determine the probability that a random pixel in the matrix contains a value larger than any arbitrary threshold *C,* denoted $F'(C)$, by simply counting the number of pixels that contain more than *C* contacts and again dividing by the size of the matrix (*N*^2).

It is therefore possible to define a function $C(b)$ so that $F'\left(C(b)\right)=F(b)$. In other words, $C(b)$ is the number of contacts such that the fraction of pixels in *M* that is larger than $C(b)$ is the same as the fraction of pixels in *M* that are within *b* of the diagonal. Furthermore, in an idealized Hi-C matrix such as the one shown in Fig. S3 (A), the pixels that lie within b of the diagonal will be exactly the pixels whose contact count is larger than $C(b)$.

It follows from the above that – for an idealized, perfectly monotonic Hi-C matrix – $Q(b)$ and $C(b)$ are exactly the same function.

In practice, this is relevant because, like the contact probability scaling, $Q(b)$ can be challenging to reliably estimate without an accurate genome assembly including chromosome-length scaffolds. By contrast, $F(b)$ can be calculated analytically using the formula above, without any experimental data at all.

Moreover, $F'(C)$ can be estimated for a given Hi-C experiment even assuming that a genome assembly with chromosome-length scaffolds is not available. In fact, $F'(C)$ can be accurately estimated using almost any reference genome assembly, so long as the effective scaffold N50 is much larger than the matrix resolution *r*.

A simple way to see why is that one can generate a proxy for the actual reference genome by concatenating all of the available scaffolds in an arbitrary order. In this proxy genome, the relative order and orientation of loci of size r will be entirely wrong. Nevertheless, most individual loci in the proxy genome will have a counterpart, containing the same sequence and having exactly the same size, in the true (albeit unknown) genome. For this reason, the vast majority of pairs of loci in the proxy genome will correspond to a pair of loci in the true (albeit unknown) genome. Thus, a Hi-C matrix generated with the proxy genome can be thought of as a permutation of the pixels of the Hi-C matrix that would be generated with the true genome. Consequently, the distribution of pixel values $F'(C)$ is unaffected by the use of a scrambled proxy genome. (Note that in practice $F'(C)$ can also be calculated from a raw scaffold set, without concatenation.)

Given estimates for $F'(C)$ and $F(b)$, estimating $C(b)$ is straightforward. Thus, it is possible to estimate $C(b)$ even with a relatively poor, and error-prone, input genome. Although $C(b)$ is not identical to $Q(b)$ for a real Hi-C matrix, it nevertheless provides a serviceable estimate for $Q(b)$. For this reason, $C(b)$ is useful in detecting misjoins.

I.b. Misjoin detection strategy

Consider a fragment of the Hi-C map shown in Fig. S3 (B). One possible misjoin score would be to place a triangular motif along the diagonal, summing the values of the pixels it contains to create a score associated with the particular genomic position:

$$S(X) = \sum_{i=X-d}^{X-1} \sum_{j=X+1}^{i+d+1} c_{ij}.$$

This score reflects the average contact frequency between a particular index locus being examined (*X*), and all other loci within d loci of the index locus. If the value of the misjoin score S is anomalously low, it suggests that the corresponding index locus spans a misjoin. Unfortunately, there is no simple and reliable way to calculate an expected value for this particular score. Thus, it is impossible to know whether the score is indeed anomalously low. Moreover, this score is extremely sensitive to "jackpot" effects, when a pixel with an anomalously high value (such as a loop or an alignment error) falls within the triangular motif.

By contrast, consider a slightly modified misjoin score. The score is calculated exactly as before, but with one change. Before calculating this score, we will apply a threshold C^* to the Hi-C heatmap, such that, whenever the value of a pixel is larger than C^*, we will change that value to exactly match C^*. Furthermore, we will exploit our ability to calculate $C(b)$ for a proxy genome in order to select C^* to be much less than $C(d)$, such that nearly all pixels in the triangle motif shown will have a value equal to C^*

in the saturated matrix – except in the case of a misjoin! When combined with our ability to calculate $C(b)$ for a low quality genome, this saturation step makes it simple to calculate an expected value for the misjoin score. Now we can obtain the following for the saturated score $S_{sat}(X)$ and the expected value (see Fig. S3 (C)):

$$S_{sat}(X) = \sum_{i=X-d}^{X-1} \sum_{j=X+1}^{i+d+1} \min\,(c_{ij}, C^{*});$$

$$S_{sat}^{ex} = \sum_{i=X-d}^{X-1} \sum_{j=X+1}^{i+d+1} C^{*} = d * (d+1) * C^{*}/2.$$

On this basis, we annotate a locus as a putative misjoin whenever the misjoin score for that locus satisfies $S_{sat}(X) < k * S_{sat}^{ex}$, where k is an arbitrary stringency parameter such that $0 \leq k < 1$. The availability of a reliable expected model greatly improves the sensitivity and specificity of such an approach. The approach is also much less susceptible to errors due to "jackpot" effects, since the impact of a single pixel is greatly dampened by the saturation step.

Note that, so long as $C^{*} < C(d)$, there is considerable latitude in selecting C^{*}. In practice, since the function $C(b)$, can only be estimated, rather than exactly calculated, it is useful to use $C(d)$ as an upper bound for C^{*}, but to choose values that are smaller, such as $C(2 * d)$. In the assemblies performed here, C^{*} is set to equal the 95th percentile of all nonzero pixels in the contact matrix.

I.c. Misjoin localization

In practice, we perform misassembly detection using two different values of the matrix resolution r. First, we annotate misassemblies at coarse resolution ($r = 25$kb), to eliminate noise. In areas flagged by the coarse resolution misassembly detection algorithm, we pinpoint the exact position of the misassembly by repeating the procedure at a higher matrix resolution ($r = 1$kb). This approach achieves high positional accuracy in misjoin identification with relatively few false positives.

I.d. Scaffolding during misjoin detection

Importantly, the misjoin detection algorithm is not performed directly on individual input scaffolds. Both misjoin detection and $C(b)$ estimation are more accurate the longer the effective N50 of the input scaffolds. Moreover, misjoin detection is significantly less sensitive if the effective N50 is less than $d \times r$. For this reason, we maximize the effective N50 of the scaffold set by running the scaffolding algorithm (see below) on the input scaffolds prior to misjoin detection. The input scaffolds are embedded in the resulting output scaffold, and thus misjoins detected in this output scaffold can be associated with misjoins in the input scaffolds.

I.e. Misjoin classification and correction

After misjoins are identified, we classify them based on whether the misjoin lies inside one of the input scaffolds – implying that there is an error in the input scaffold, which needs to be corrected – or whether the misjoin lies at the junction between two scaffolds, suggesting that the misjoin is a consequence of an error in the input sequence located at a different position. (Notably, our Hi-C based scaffolding step very rarely introduces a misjoin unless there is another, “causative” misjoin in one of the input scaffolds. For instance, a misjoin connecting loci from two different chromosomes can lead to the fusion of two large segments from those chromosomes into a single scaffold; as the scaffolder proceeds, this anomalous scaffold will tend to create many “non-causative” misjoins. Correction of the causative misjoin leads to resolution of the downstream misjoins.)

If a misjoin lies inside a scaffold, the scaffold is edited by excising sequence intervals flagged by the misjoin detection algorithm (see Fig. S4). The excised fragment is labeled as an additional, ‘inconsistent’ scaffold and excluded from subsequent assembly iterations, since its continued presence during the scaffolding process could lead to further misjoins. If the misjoin is sufficiently far from both ends of the scaffold, this results in splitting the affected scaffold into two scaffolds at the site of the misjoin (in addition to the formation of an inconsistent scaffold). Note that multiple misjoins can be identified in a single scaffold during a single round of misjoin detection, which could lead to repeatedly splitting one scaffold into multiple smaller scaffolds.

I.f. Pseudocode

The misassembly correction procedure can be described using the pseudocode listed in Table S13.

Overall, our misjoin detection algorithm is characterized by low false positive error rates and accurate localization. It is especially sensitive to large misassemblies that give rise to large-scale errors in the genome. Several examples of automatic misassembly detection are given in Fig. S5.

II. Algorithm for scaffolding

To transform a set of input scaffolds into chromosome-length scaffolds, three problems must be solved. “Anchoring” assigns each scaffold to a chromosome, thus partitioning the set of scaffolds into multiple subsets. “Ordering” assigns a relative position to each scaffold on each chromosome with respect to the other scaffolds assigned to the same chromosome. “Orienting” determines which of the two ends of each scaffold is adjacent to the preceding scaffold in the ordering, and which end is adjacent to the next scaffold in the ordering. (This step is equivalent to assigning each scaffold to one of the two complementary strands that comprise a chromosome.) Our algorithm for constructing chromosome-length scaffolds begins with a set of input scaffolds, and simultaneously anchors, orders, and orients them.

The algorithm we employ is iterative; the same steps are performed over and over, often thousands of times. In each step, subsets of the input scaffolds are ordered and oriented with respect to one another to create a new, longer set of scaffolds, which are then used as inputs for the next step. (These might be called “superscaffolds,” although we will not use that terminology here. Similarly, when the inputs to our algorithm are scaffolds, rather than contigs, the algorithm itself might be thought of as a “super-scaffolding” algorithm, rather than a “scaffolding” algorithm; again, we will not make that distinction here.) For the remainder of this section, we will use the term "input scaffolds" to refer to the scaffolds which are the inputs to each step; when needed, we will use the term "initial input scaffolds" to refer to the scaffolds which are the inputs to the iterative algorithm as a whole.

Each iterative step involves constructing and solving a graph optimization problem.

We assume that the input to a given step includes two or more scaffolds. (Otherwise, the scaffolding problem is already solved!)

We begin by splitting each input scaffold into two "hemi-scaffolds" by bisecting the scaffold sequence at the midpoint. The pair of hemi-scaffolds that derive from a single hemi-scaffold are dubbed "sister hemi-scaffolds."

Next, we construct the "density graph," *S*. Each hemi-scaffold is represented as a single vertex in the density graph. Then we append edges to the density graph as follows (see Fig. S6):

- First, we append edges between all pairs of vertices that do not correspond to sister hemi-scaffolds. We call these "non-sister" edges. The weight of each non-sister edge corresponds to the density of Hi-C contacts between the corresponding hemi-scaffolds. To calculate this density (i.e., the edge-weight), we count number of Hi-C contacts where one read is incident on one of the hemi-scaffolds, and the other read is incident on the other hemi-scaffold. We then take the resulting value and divide it by the product of the sequence length of the two hemi-scaffolds to arrive at the density. Note that, all else being equal, having an edge of greater weight between two hemi-scaffolds indicates that the two hemi-scaffolds tend to be more proximate in 3D, and thus are more likely to be nearby along the one-dimensional (1D) chromosome sequence as well.

- Next, we append edges between all pairs of vertices that correspond to sister hemi-scaffolds. All of these edges are assigned a weight of 2*MAXS, where MAXS is the maximum weight of all of the non-sister edges. This is done in order to encode the fact that sister hemi-scaffolds are adjacent to one another according to the input scaffold set, and that this evidence is – during each scaffolding iteration – regarded as more reliable than any evidence derived from Hi-C. (Of course, in the preceding section we described strategies for correcting scaffolds using Hi-C data. The results of these strategies influence the input scaffold set for any given step of the scaffolding algorithm. However, within the individual iterations, the accuracy of the input scaffold set is regarded as a constraint that takes precedence over the Hi-C data.)

Note that prior approaches for scaffolding using Hi-C rely directly on measures of contact density. This approach can be error-prone. For instance, high-coverage scaffolds, scaffolds containing loci engaged in strong long-range interactions, scaffolds containing repeat sequences, etc. might all display very frequent contacts with scaffolds that are far away from them along the 1D chromosome sequence. Conversely, input scaffolds from low-coverage regions of the genome might exhibit a relatively low contact density, even with scaffolds that are adjacent to them in 1D.

In order to reliably determine the relative positioning and orientation of scaffolds given these potential pitfalls, we have developed a method for identifying adjacent scaffolds that is not directly based on absolute read density. (We call a pair of input scaffolds "adjacent" if they are on the same chromosome, and no other input scaffold has a true genomic position that lies on that same chromosome, in between them.)

To accomplish this, we first use our density graph to define an "unfiltered confidence graph," C'. The vertices of the unfiltered confidence graph again correspond to the hemi-scaffolds. The edges of the unfiltered confidence graph are defined as follows (see Figure S6):

- If A and B are not sister homologs, then we append an edge between them whose weight is the ratio of the weight of the edge connecting them in the density graph (s_{AB}), and the weight of the second-largest non-sister edge incident on either A or B in the density graph. Note that $c_{AB}>1$ if and only if there is no hemi-scaffold whose contact density with either A or B exceeds the contact density between A and B. Informally, this means that, based on the Hi-C data, A is the best partner for B, and B is also the best partner for A. Thus we can be confident that A and B are adjacent. We therefore call any edge in the unfiltered confidence graph whose weight is greater than 1 "reliable." Edges whose weight is 1 or smaller are called "unreliable."

- Next, we append edges between all pairs of vertices that correspond to sister hemi-scaffolds. All of these edges are assigned a weight of 2*MAXC, where MAXC is the maximum weight of all of the non-sister edges in the unfiltered confidence graph. This is done in order to encode the fact that sister hemi-scaffolds are adjacent to one another according to the input scaffold set, and that this evidence is – during each scaffolding iteration – regarded as more reliable than any evidence derived from Hi-C.

If all of the non-sister edges are unreliable, then the iteration has failed, in the sense that no reliable adjacency information could be extracted from the Hi-C data. Therefore, if all the non-sister edges are unreliable, we remove the smallest input scaffold from the input scaffold set and repeat the step, constructing a new density graph, etc. Note that removing the smallest input scaffold and repeating the step might still not yield a reliable edge, in which case another scaffold is removed, and so on. Eventually, either a reliable edge will be found or there will only be one scaffold left (at which point the algorithm halts and outputs the remaining scaffold.)

Assuming there is a reliable non-sister edge in the unfiltered confidence graph, we next filter the unfiltered confidence graph by removing all edges whose weight is less than or equal to 1. The resulting graph is called the confidence graph. Note that every vertex is adjacent to 1 sister edge in the confidence graph, and to at most 1 non-sister edge. Thus, all vertices in the confidence graph have either degree 1 or degree 2. Hence (by an elementary fact of graph theory) the confidence graph is a collection of disjoint paths and cycles.

(Note that it is possible to use our methods to reconstruct circular plasmids and chromosomes, in which case the following steps must be modified; we will not do so here. Instead, for the sake of simplicity, we describe methods that apply to organisms with linear chromosomes.)

Vertices that are adjacent in the confidence graph are very likely to correspond to hemi-scaffolds that are adjacent in 1D. Therefore, each path in the confidence graph corresponds to a high-confidence scaffold. Furthermore, each path in the confidence graph whose length is greater than 2 corresponds to a new scaffold comprised of multiple input scaffolds whose relative order and orientation has been determined using Hi-C.

Cycles in the confidence graph are harder to interpret, since they contain multiple possible paths (i.e., new scaffolds) spanning all the vertices (hemi-scaffolds) in the cycle. Here the key is to identify the maximal path contained in the cycle, which corresponds to the new scaffold in which we have the highest confidence given the Hi-C data. This can be easily accomplished by removing the edge in each cycle whose weight is smallest. (Because of how the confidence graph is constructed, this edge will always be a non-sister edge.) In graph theoretic terms, this procedure can be thought of as a way to construct the maximal (in terms of total edge weight) vertex-disjoint path cover of the confidence graph.

A complementary way of formulating this procedure (which is mathematically guaranteed to produce exactly the same output) is to greedily select the highest weight edges from the confidence graph, ensuring that no vertex in the graph will be incident on two or more edges. (This constraint is equivalent to the rule that, after selecting non-sister edge AB, we must remove all other non-sister edges incident on either A or B.) Following this procedure, a maximum weight vertex-disjoint path cover is eventually obtained. Notably, for the special case of confidence graphs, this complementary formulation is exactly equivalent to Kruskal's algorithm for constructing a maximal spanning forest (*29*). (Because a confidence graph consists of disjoint paths and cycles, its maximal spanning forest is always a vertex-disjoint path cover.) In fact, it is this complementary procedure that is implemented in our code.

Since vertices that are adjacent in the confidence graph are very likely to correspond to hemi-scaffolds that are adjacent in 1D, and since each path in the confidence graph corresponds to a high-confidence scaffold, the maximum weight vertex-disjoint path cover in the confidence graph corresponds to a new set of scaffolds that is optimal with respect to the input scaffolds and the Hi-C data. Each of these new "output" scaffolds comprises one or more input scaffolds; within each output scaffold, the relative order and orientation of the input scaffolds has been determined using Hi-C.

Once the output scaffolds are obtained, the iteration ends. The next iteration can now begin; the output scaffolds from the previous iteration can be used as input scaffolds for the new iteration, and the density and confidence graphs are constructed for the new inputs. Note that reconstructing the graph from scratch allows more Hi-C data spanning larger scales to be incorporated into the analysis. Of course, reconstructing the density and confidence graphs is computationally expensive, and is therefore only done once no more information can be extracted from the reliable edges in the previous iteration; i.e., after the vertex-disjoint path cover of the confidence graph has been calculated, new scaffolds have been obtained, and the previous iteration is complete.

(Note that a more computationally efficient alternative to recalculating the density graph is to use a more permissive threshold for including edges in the confidence graph; i.e. require $c_{AB}>k$ where $k<1$. This would allow more scaffolding information to be extracted at each step, and thus fewer steps would be required to complete the scaffolding procedure. However, this strategy could also increase the frequency of errors, and it is not the strategy we employ here.)

If there is only one output scaffold, the process ends and the single output scaffold is reported.

To summarize we give pseudocode in Table S14.

Note that the algorithm does not rely on a preliminary Hi-C based clustering step to identify chromosomes (compare to (*10*)). This is particularly useful for species like mosquito, where loci that lie far apart on the same chromosome may not exhibit enhanced contact frequency relative to loci on different chromosomes (see Fig. 1 and Fig. S15).

III. Polishing

Polishing is an optional step in our pipeline that has been designed to address challenges associated with unusual 3D features that arise for organisms exhibiting strong telomere and centromere clustering. This can create false positives during scaffolding, since extremely strong off-diagonal 3D signals associated with telomere and centromere clustering can sometimes be strong enough to rival the contact frequencies observed for loci that are adjacent in 1D.

Figure S7 shows the Hi-C contact map built with respect to the *Ae. aegypti* genome assembly before and after the polishing step. The map suggests that chromosome 3 is very accurately assembled, but chromosomes 1 and 2 contain a type of error that is characteristic of assembly in genomes that exhibit strong telomere-to-telomere clustering. In this error, the enhanced proximity between the two telomeres is mistaken for 1D proximity. As a result, the raw chromosomal scaffolds corresponding to chromosomes 1 and 2 exhibit a cyclic permutation with respect to the true chromosome.

As an example, if the sequence of the true chromosome was ABCDEFG, where locus A and G are telomeres, then the erroneous sequence might be DEFGABC. We call this sort of error a “cycle break.” Note that, when a cycle break occurs, an off-diagonal peak linking the two putative ends of the chromosome (in the example, D and C) is still seen in the Hi-C map. However, this signal is actually due to the true 1D proximity between the two ends of the putative chromosome, rather than at true 3D signal. Conversely, the on-diagonal signal between A and G, which appears to reflect 1D proximity, is in fact due to the telomere clustering. (Similar errors may arise due to strong interaction between telomeres of two different chromosomes. They are addressed in the same way. See Fig. S7, chromosomes 2 and 3.)

Such errors can be corrected by a single additional round of misjoin correction, performed at extremely low resolution ($r \sim 1\text{Mb}$). The low-resolution misassembly detection identifies reliable “superscaffolds”, each of which is many megabases in length. These superscaffolds are then ordered and oriented using a version of the scaffolder that exploits the large size of the superscaffolds to more reliably distinguish 1D and 3D signal by utilizing Hi-C contacts incident only on the superscaffold ends, rather than on the whole superscaffold.

IV. Algorithm for extracting raw chromosomal scaffolds from the megascaffold

The scaffolding algorithm produces a single megascaffold that concatenates all the chromosomes.

For genomes that do not exhibit pronounced telomere clustering in the Hi-C map (such as human), we split the megascaffold into chromosomes by running a variant of the misassembly detector to identify the chromosome boundaries. This algorithm relies on the fact that the contact frequency between scaffolds that are adjacent on the megascaffold but which lie on different chromosomes is relatively low, since they are not actually in 1D proximity. Thus the boundaries between chromosomes generate a signal that is similar to a typical misjoin. Moreover, this effect is enhanced by the tendency of loci on the same chromosome to exhibit elevated contact frequency.

If the spatial clustering of telomeres is evident in the Hi-C map, the phenomenon can be exploited in the effort to partition the genome into chromosomes. In particular, the first scaffold in the megascaffold must come from the end of a chromosome, and therefore derives from a telomere. Identifying positions in the Hi-C matrix that have an enriched number of contacts with the megascaffold edge thus enables the detection of chromosome boundaries.

V. Algorithm for detection and correction of false positives that occurred during misjoin detection ("Sealing")

During the sealing step, sequences that were erroneously excised during misjoin correction may be re-introduced. In particular, if the two parts of a corrected scaffold remain adjacent to one another in the raw chromosomal scaffold, it suggests that the original scaffold was correct, since the independent contact patterns from both parts are consistent with the original scaffold. In this case, the misjoin that led to the correction is judged to be a false positive and the intervening sequence is restored.

VI. Algorithm for merging assembly errors due to undercollapsed heterozygosity

A frequent error modality found in draft haploid genome assemblies is undercollapsed heterozygosity. This is when there exists a subset of the scaffolds such that each scaffold accurately corresponds to a single locus in the genome, but these loci overlap one another. Consequently, there are individual loci in the genome that are covered multiple times by different scaffolds. This error is typically caused by the presence of multiple haplotypes in the input sample material, which are sufficiently different from one another that the contig and scaffold generation algorithms do not recognize them as emerging from a single locus. This class of error is frequent in AaegL2; the step can be omitted when assembling genomes of organisms with low heterozygosity such as Hs1.

Undercollapsed heterozygosity error leads to highly fragmented draft assemblies with a larger-than-expected total size (*30*, *31*). This, in turn, causes numerous problems in downstream analyses such as erroneous gene copy number estimates, fragmented gene models etc. The challenge remains significant even when special effort is taken to reduce the levels of heterozygosity in genomic data by inbreeding as has been done with the draft AaegL2 assembly (*18*). It is therefore important to ensure that the final genome reported by our assembler minimizes the number of assembly errors due to undercollapsed heterozygosity.

To specifically address this class of misassembly error, we have developed an algorithm whose goal is to merge these overlapping scaffolds into a single scaffold accurately incorporating the sequence from the individual scaffolds. The result of this is a merged haploid reference scaffold.

One assumption of the overlap merging algorithm is that, when multiple scaffolds correspond to multiple haplotypes, these scaffolds will exhibit extremely similar contact patterns, genome-wide. (Note that although some interesting examples of homolog-specific folding have been documented (*5*, *32*), the relative input from the differential signal is very small as compared to that coming from the 'diagonal', i.e. from 3D interaction associated with proximity in 1D, so the assumption seems to hold true for the vast majority of candidate loci.) Because they exhibit similar long-range contact patterns, the scaffolding algorithm tends to assign such scaffolds to nearby positions in the genome. Thus, the merge algorithm seeks to identify pairs of resolved scaffolds that (i) lie near one another in the raw chromosomal scaffolds, and (ii) exhibit long stretches of extremely high sequence identity.

Briefly, we search for undercollapsed loci by running a sliding window of fixed width along the raw chromosomal scaffolds. We then use LASTZ to do pairwise alignment of all pairs of resolved scaffolds that fall in the sliding window (*28*). The total score of all collinear alignment blocks (stanzas), normalized by the length of the overlap, is used as a primary filtering criterion to distinguish between alternative haplotypes and false positive sequence similarity. The location of the overlap relative to input scaffold boundaries is also taken into account in determining whether the scaffolds can be correctly merged (see Fig. S8).

We next construct a graph whose nodes are resolved scaffolds, and where edges reflect significant sequence overlap between resolved scaffolds that are proximate on the raw chromosomal scaffold. The resulting graph contains a series of connected components. Cycles in the graph are analyzed in order to filter out components with overlaps on conflicting strands.

Finally, we construct a tiling path through the scaffolds of each individual connected component, recursively aligning scaffolds to an already collapsed portion of the group, finding the highest scoring alignment block and switching from one haploid sequence to the other at the endpoints of the alignment.

Ideally the resolved scaffolds in each connected component are consecutive on the raw chromosomal scaffold, and with relative orientations that match those suggested by the pairwise alignments. In practice, however, this is not always the case. This can be due to differences in haplotype representation between the genomic data used to produce the draft assembly and that of the Hi-C experiment. For example, sequences belonging to different clusters may be intertwined. Similarly, the orientation of contigs/scaffolds within the cluster as suggested by pairwise alignment may not match those suggested by the scaffolding step. In such cases the relative position and orientation of the connected components with respect to the rest of the assembly is decided by majority vote with each input scaffold's contribution weighed by its length. Alternatively, assembly can be rerun using the merged components as input.

Note that although it is possible to add additional constraints when appropriate, such as the exact number of haplotypes present in the data, we do not rely on such knowledge in general, or in any of the assemblies we performed in this paper. This allows us to work with polymorphic assemblies, such as when multiple individuals were used to produce the draft assembly. In particular it allows us to handle cases where the degree of polymorphism is unknown.

Computational details

Each of the genomes reported in the paper were produced by running the pipeline on a single compute node: Intel Xeon E5-2683v3 for Hs2-HiC and AaegL4 and Power8 8335-GCA for CpipJ3 assembly. The number of iterations that were necessary to remove misassemblies in the input scaffold set was two for Hs2-HiC and CpipJ3, and nine iterations for AaegL2. Polish and merge steps were omitted for Hs2-HiC.

Hs1 draft genome

Hs1 was created using DISCOVAR *de novo* version 52488 (*13*, *33*), setting min_link_count to 10.

Small scaffolds

The small scaffolds, and more specifically the unresolved scaffolds (see Fig. S2) in Hs2-HiC are often associated with small scaffolds in hg38 that are not included in any chromosome. For example, 12.3% of all alignments (primary, secondary and chimeric) for unresolved scaffolds map to non-chromosomal scaffolds in hg38, as compared with 1.7% for resolved scaffolds. Unresolved scaffolds also have an elevated repeat content. The average alignment quality (mapq) for unresolved scaffolds is 31, as compared with 55 for resolved scaffolds. Finally, 9% of reads that align to unresolved scaffolds have mapq<10, as compared with 3% for resolved scaffolds. The alignments for this analysis were performed using BWA (*34*).

Proof-of-concept NA12878 genome assembly from Pacific Biosciences data

A proof-of-concept experiment was performed using contigs from the draft assembly of NA12878 published by Pendleton et al. (GenBank assembly accession: GCA_001013985, (*16*)). Crucially we did not use any scaffolding information from (*16*), which was derived from BioNano optical mapping data (*35*). Instead, the reported scaffolds were split to yield a contig set spanning 3.03Gb of sequenced bases with a contig N50 of 1.56Mb. After scaffolding these contigs using Hi-C we again obtained 23 chromosome-length scaffolds, which spanned 95% of the sequenced bases of hg38.

AaegL2 draft assembly

The AaegL2 assembly was downloaded from GenBank (GenBank accession number GCA_000004015.2).

AaegL4 genome assembly validation

Of the 2006 markers in the genetic linkage map (*19*), 1826 markers were assigned to chromosome-length scaffolds in AaegL4 (Table S15). Of these, 1814 have a position in AaegL4, which is in perfect agreement with the linkage map, in the sense that these 1814 markers appear in the same order in AaegL4 and in the linkage map. In 8 cases, a marker from one linkage group has a position in AaegL4 that is consistent with the immediately adjacent linkage group (*19*). These may reflect fine-structure ordering errors in AaegL4, small errors in the linkage map, or errors in the draft assembly (see below).

For the remaining 4 markers, AaegL4 and the linkage map disagree (i.e., AaegL4 is not consistent with placement of the marker in its original position in the linkage map, or in either of the neighboring linkage groups; see Fig. 2 and Fig. S10). We examined these cases of disagreement more closely. Specifically, we examined the scaffolds in the draft assembly (supercontigs 1.3, 1.32, 1.45, 1.90) that contain these markers (accession numbers AAGE02000222.1, AAGE02002163.1 AAGE02003018.1, AAGE02005396.1). Upon reexamination of the Hi-C data, we found that all of these draft scaffolds contained misassemblies. In all four cases, a small locus (23kb, 16kb, 125kb and 4kb in length) from a different 1D position was erroneously inserted into these larger scaffolds. These misassemblies had not been identified by our misassembly correction procedure. Crucially, these smaller loci contained the actual genetic marker.

Although these errors in AaegL2 are retained in the final version of AaegL4, we confirmed our assessment of the origin of the disparities between the linkage map and AaegL4 by manually removing the small marker loci from the misassembled larger scaffold. We then used the Hi-C data to specifically localize these small marker loci. The position suggested by the Hi-C data for both marker loci agreed with the position suggested by the genetic map.

We also compared misassembly errors identified in the linkage study with those that we detected. Notably, all 63 contigs identified as misassembled in (*19*) were identified as misassembled on the basis of Hi-C data. Furthermore, the breakpoints in the scaffolds are consistent across the two methods (see Fig. S11). However, the Hi-C signal allows the misjoins to be localized with much higher resolution.

Taken together, the above findings indicate outstanding agreement between AaegL4 and the best available genetic map of *Aedes aegypti*.

We have also examined the correlation between the position of 500 mapped BAC clones in the recently published physical map (*36*) and their positions on AaegL4. Each BAC end was treated independently. The results are presented in Fig. S12. Alignment data are listed in Table S16.

Out of 828 unambiguous alignments to chromosome-length scaffolds in AaegL4, 90% lie in a 50Mb band around the diagonal. Manual examination of a few selected markers from the remaining 10% reveals no obvious discrepancy between their placement in AaegL4 and the Hi-C signal.

Finally, we have analyzed the 29 AaegL2 scaffolds with inconsistent markers in the physical mapping study: 27 of these were flagged as misassembled on the basis of Hi-C data. The remaining two AaegL2 scaffolds, 1.209 and 1.302, had also been examined in the linkage mapping study (*19*). The latter study did not detect a misassembly. Thus, in this case, the Hi-C and linkage studies are consistent with one another, and the physical map is inconsistent with both.

Given the strong agreement of our genome assemblies with the best available genetic maps, we believe that the relatively higher rate of inconsistency seen in the physical maps may be due to strain-specific variation or inaccuracies in the physical map, but does not represent errors in AaegL4 scaffolding.

Note that the raw chromosomal scaffolds were used for the above analysis, which enabled us to employ alignments generated in prior mapping work, when such alignments were available; when they were not, BWA was used instead (*34*).

CpipJ2 draft assembly

The CpipJ2 assembly was downloaded from VectorBase Release VB-2016-06.

CpipJ3 genome assembly validation

We assessed the accuracy of our chromosome-length scaffolds by aligning markers from existing linkage and physical mapping studies to CpipJ3.

To the best of our knowledge, the most comprehensive extant genetic map of *Culex quinquefasciatus* is (*21*), a microsatellite map which characterizes 63 markers that aligned to the raw chromosomal scaffolds of CpipJ3. Of these, 61 out of 63 markers are in perfect agreement with CpipJ3; 1 marker corresponds to a position in CpipJ3 that is consistent with an immediately adjacent linkage group; and 1 marker is in disagreement with CpipJ3 (see Fig. S13).

We examined the lone marker that disagreed with CpipJ3 more closely. As in the case of earlier discrepancies (discussed above, see Supplemental Note 9), the CpipJ2 scaffold (supercontig 3.99) with the marker C99GTC1 (NCBI probe 32416933) contained a misassembly error. A small locus (13kb in length) from a different 1D position, which contained the marker, was erroneously inserted into the larger scaffold. This misassembly had not been identified by our procedure. Six out of nine scaffolds identified as misassembled from the mapping data in (*21*) were flagged as misassembled by the misassembly detector. One of the three remaining scaffolds is 3.99 with a 13kb insertion spanning the C99GTC1 marker. Manual analysis of the other two scaffolds (3.65, 3.177) revealed no large-scale inconsistencies in the Hi-C signal associated with these scaffolds.

Taken together, the above findings indicate outstanding agreement between CpipJ3 and the best available genetic map of *Culex quinquefasciatus*.

We also compared our findings with a lower-quality map containing fewer markers. For the RFLP map (*20*), we observed six differences with CpipJ3 (see Fig. S13). Three of these were relatively small: the marker's position in CpipJ3 corresponds to a position on the linkage map that is within two linkage groups of the group suggested by the map. These disagreements may be due to misassemblies that we were unable to detect, errors in the linkage map, strain-specific rearrangements, or to fine-scale ordering errors in CpipJ3.

Composite RFLP (*20*) and microsatellite linkage map (*21*) marker positions are compared to CpipJ3 placement in Figs. 2 and S13. Individual marker data are listed in supplementary Tables S17, S18.

The physical maps we analyzed represented two different approaches: polytene chromosome mapping (*37*), and mitotic chromosome mapping (*38*). CpipJ3 is in good agreement with both datasets (see Fig. S14).

For the polytene chromosome map (*37*) (see Fig. S14 (A) and Table S17), CpipJ3 is consistent with the positions of all but one marker CX51 (accession number GT056146). As in earlier cases, this disagreement was due to the marker locus being erroneously inserted into a larger scaffold (supercontig 3.28) in CpipJ2, which we did not correct. Notably six CpipJ2 scaffolds contained exactly two markers in this map; thus, the map suggests an orientation for these 6 scaffolds. In each case, this orientation matches the orientation in the CpipJ3 assembly (see Table S19).

A larger number of inconsistencies were observed between CpipJ3 and the mitotic physical map (*38*) (see Table S20). At least in the case of some of the disagreements, CpipJ3 is consistent with linkage data, which is inconsistent with the mitotic physical map. For example, the position of marker LF108 (accession number T58322) in CpipJ3 is inconsistent with the Naumenko et al. physical mapping data (*38*), but is consistent with the genetic linkage map which places this marker on chromosome 2 (*21*).

Given the strong agreement of our genome assemblies with the best available genetic maps, we believe that the relatively higher rate of inconsistency seen in the lower-quality maps may be due to strain-specific variation or inaccuracies in the physical map, but does not represent errors in CpipJ3 scaffolding.

Note that the raw chromosomal scaffolds were used for the above analysis, which enabled us to employ alignments generated in prior mapping work when such alignments were available; when they were not, BWA was used instead (*34*).

Comparative analyses with *An. gambiae* genome

Comparative analyses with *An. gambiae* genome relied on existing LASTZ_NET alignment data between the *An. gambiae* genome assembly AgamP4 and the AaegL2 and CpipJ2 draft genomes available from VectorBase (*28*, *39*). The alignment data were further processed in order to assign each alignment block a position on the raw chromosomal scaffolds of AaegL4 and CpipJ3. Thus, each block represents an alignment between one locus in a genome and a second, orthologous locus in another genome.

We process this list of ortholog pairings in various ways. For instance, we describe how often loci on a particular arm in one genome are paired with loci on a particular arm in another genome. Matrices describing such analyses for all pairs of arms are provided in Tables S8-S11. In Figures S17 and S18, we provide histograms at 1Mb resolution showing the frequency with which loci on a particular chromosome arm of AgamP4 align to all positions, genome-wide in AaegL4 and CpipJ3.

Note that although the results are in broad correspondence with several cytogenetic studies (*18–20*), the extent to which individual arms are conserved in *An. gambiae*, *Cx. quinquefasciatus* and *Ae. aegypti* was not apparent from prior genome assemblies, even after improvements using various anchoring and mapping strategies (Figs. S17 and S18).

Comparative analyses with *D. melanogaster* genome

Comparative analyses with *D. melanogaster* genome rely on LASTZ_NET alignment data between the *D. melanogaster* genome assembly BDGP6 and the AaegL2 and CpipJ2 draft genome assemblies available from Ensembl (*28*, *40*). The distribution of conserved sequences highlights several highly conserved chromosome arms (Fig. S19).

Cost estimate for Hi-C based assembly

Overall, our results show that incorporating Hi-C data into genome assembly provides a rapid, inexpensive methodology for generating highly accurate *de novo* assemblies with chromosome-length scaffolds for a wide variety of species. For example, using the Hs2-HiC strategy, the actual production costs in our lab for a *de novo* mammalian genome are less than $10,000. Since most of this cost is sequencing, and the quotes we rely on for our cost estimates are consistent with rates available to the broader community, there is no reason this strategy could not be employed in other labs at a similar cost.

The breakdown of projected expenses can be found in Table S12. The whole library preparation and quality control sequencing pipeline takes a single technician a full work-week from end-to-end. Assuming 20 samples are processed at a time (which is a reasonable estimate for a skilled technician) yields an estimate of ~$50 per library in terms of library preparation costs. The DNA-seq library preparation estimate is calculated based on Qiagen Genomic Tip ($233 per 25 columns), Qiagen Genomic DNA Buffer Set ($163 per 75 minipreps) and TruSeq® DNA PCR-Free HT Kit ($3000 for 96 samples) list prices in 2016. The Hi-C library preparation estimate is calculated based on the price for consumables listed in the *in situ* Hi-C protocol (*5*). The sequencing costs are listed based on estimates of $8,600 per HiSeq2500 flow cell for PE250 and $1,750 per HiSeqX lane, PE150. These estimates were obtained by comparing quotes from several institutions. It is worth noting that, instead of HiSeqX PE150, shorter read-length NextSeq instruments can be used to achieve comparable physical coverage by Hi-C (though the quality of the mapping will be somewhat reduced). This yields a total cost estimate of ~$9,450.

The cost can be much lower for smaller genomes, or if some draft sequencing data are already available. For example, the combined cost of AaegL4 and CpipJ3 was ~$4,000, which includes library preparation and sequencing for both projects. The sequencing run was performed on a NextSeq500 machine; using HiSeqX technology, the total cost for improving both assemblies would have been $2,000.

Comparison with existing methods: Hs1 genome assembly

We compare the algorithms introduced in the paper with LACHESIS (*10*), the most widely-used tool for large-scale Hi-C based genome assembly (*8*, *11*). LACHESIS was downloaded from GitHub (permalink 81ea957348ce0db454145399f2cfe1253c0ff427).

The input to both algorithms was the same: a set of DISCOVAR *de novo* scaffolds from 60X PE250 Illumina short reads and 6.7X of Hi-C data. Only scaffolds longer than 20kb were used, although the results were similar without this restriction. Since LACHESIS does not contain an error correction module we have also disabled misjoin detection in our pipeline for this comparison. In addition, LACHESIS was provided with the correct number of scaffolds (23; this number is required in order to run LACHESIS), whereas our algorithm was not.

The dotplots showing alignment of the resulting scaffolds to hg38 are shown in Fig. S20. As the dotplots illustrate, LACHESIS was able to anchor many of the scaffolds, but failed to generate chromosome-length scaffolds. In contrast, our algorithm correctly reconstructed the 23 chromosomes as evidenced by a near-perfect diagonal in the dotplot.

(Though several inversions remain due to the fact that we did not use our misjoin detection algorithm for this comparison.)

Comparison with existing methods: AaegL2 genome assembly

We also compare the results of running LACHESIS, our scaffolding algorithm (without misjoin correction), and our full pipeline (including misjoin correction) on AaegL2. Since an accepted reference does not exist for the *Aedes aegypti* genome, the quality of each algorithm's output is illustrated by comparison with a genetic linkage map (*19*). Neither LACHESIS nor our scaffolding algorithm alone can assemble the genome. In contrast, by employing misjoin correction, our full pipeline produces an extremely accurate assembly (see Fig. S21).

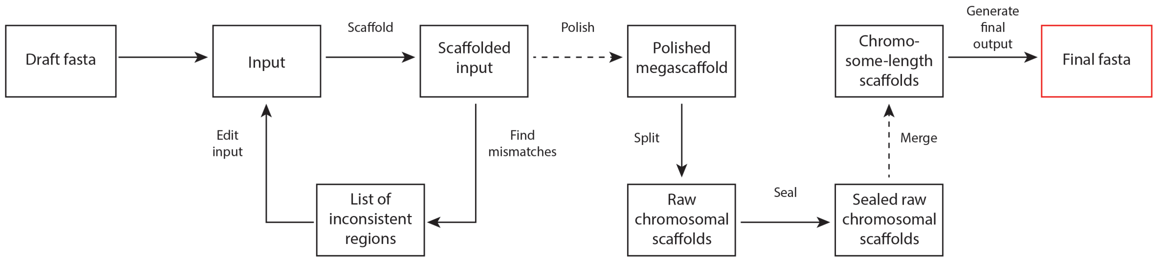

Fig. S1.

Workflow diagram describing the computational pipeline. The pipeline starts with scaffolding the input and assessing the result for misassemblies. The detected misassemblies are linked to misjoins in the draft contigs/scaffolds. Once the misjoins have been removed from the input, the scaffolding and misassembly analysis is repeated. Iterative scaffolding and misassembly detection constitute the core section of the pipeline. For some genomes polishing and merging is also employed. Additional steps include chromosome splitting and sealing (see Pipeline description).

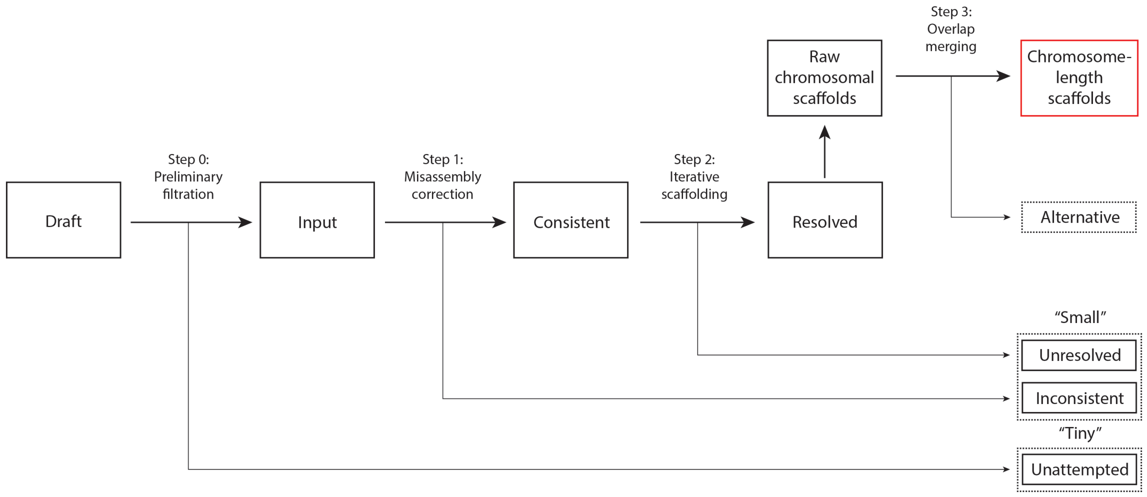

Fig. S2.

A workflow diagram illustrating the processing of various scaffold populations, beginning with draft scaffolds and ending with chromosome-length scaffolds. Each node corresponds to a set of scaffolds; the terminology used in the paper and supplement to refer to that set of scaffolds is shown.

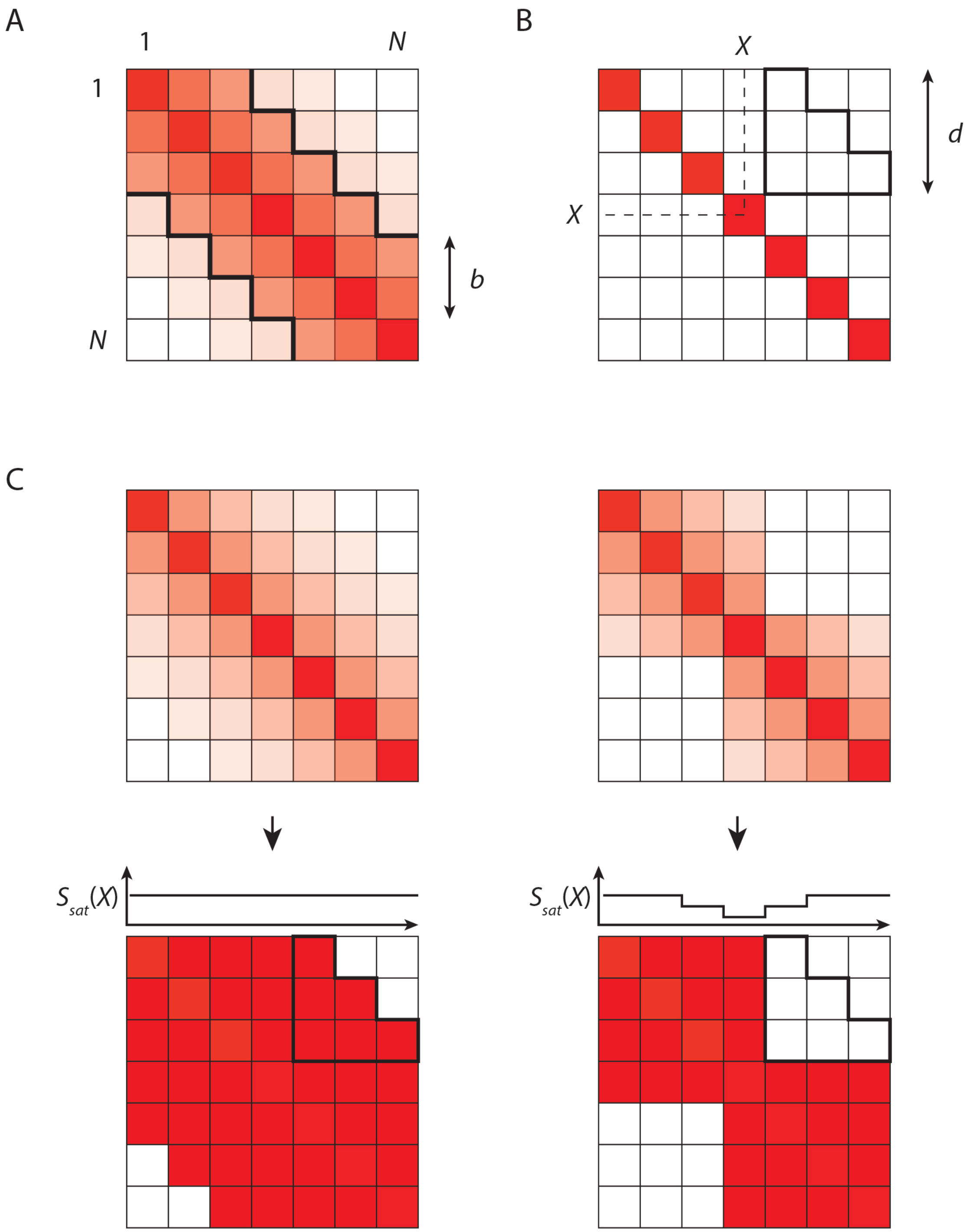

Fig. S3.

Misassembly detection notation and algorithm. (A) Calculating the number of bins in between the diagonals from $c_{1+b,1}$ to $c_{N,N-b}$ and from $c_{1,1+b}$ to $c_{N-b,N}$. (B) Triangular shape used to calculate the scores $S(X)$ and $S_{sat}(X)$ along the assembly. (C) Schematic representation of matrix saturation and the distribution of the score $S_{sat}(X)$ along the genome. Bright red signifies the highest scoring bin in a given matrix.

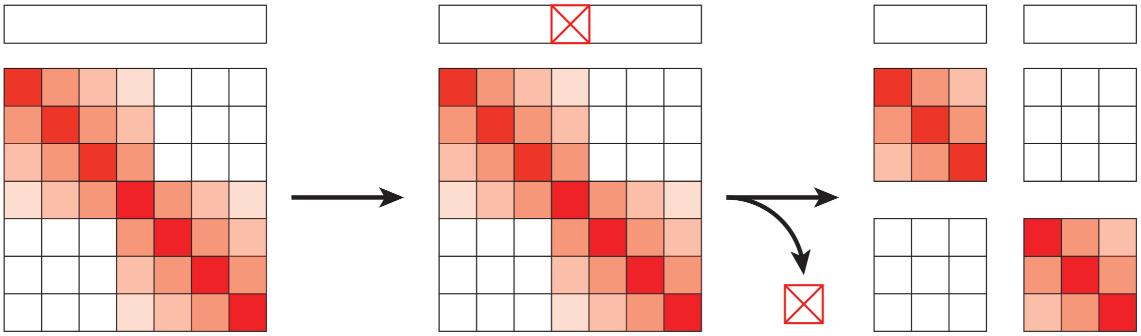

Fig. S4.

Misassembly correction. Once the misassembly detection algorithm has identified a problematic region that lies inside an input scaffold (a bin marked with an X), the region gets excised resulting in two internally consistent fragments of the original input scaffold. The third fragment that spans a misassembled region is labeled as inconsistent. Inconsistent fragments do not participate in the next round of scaffolding.

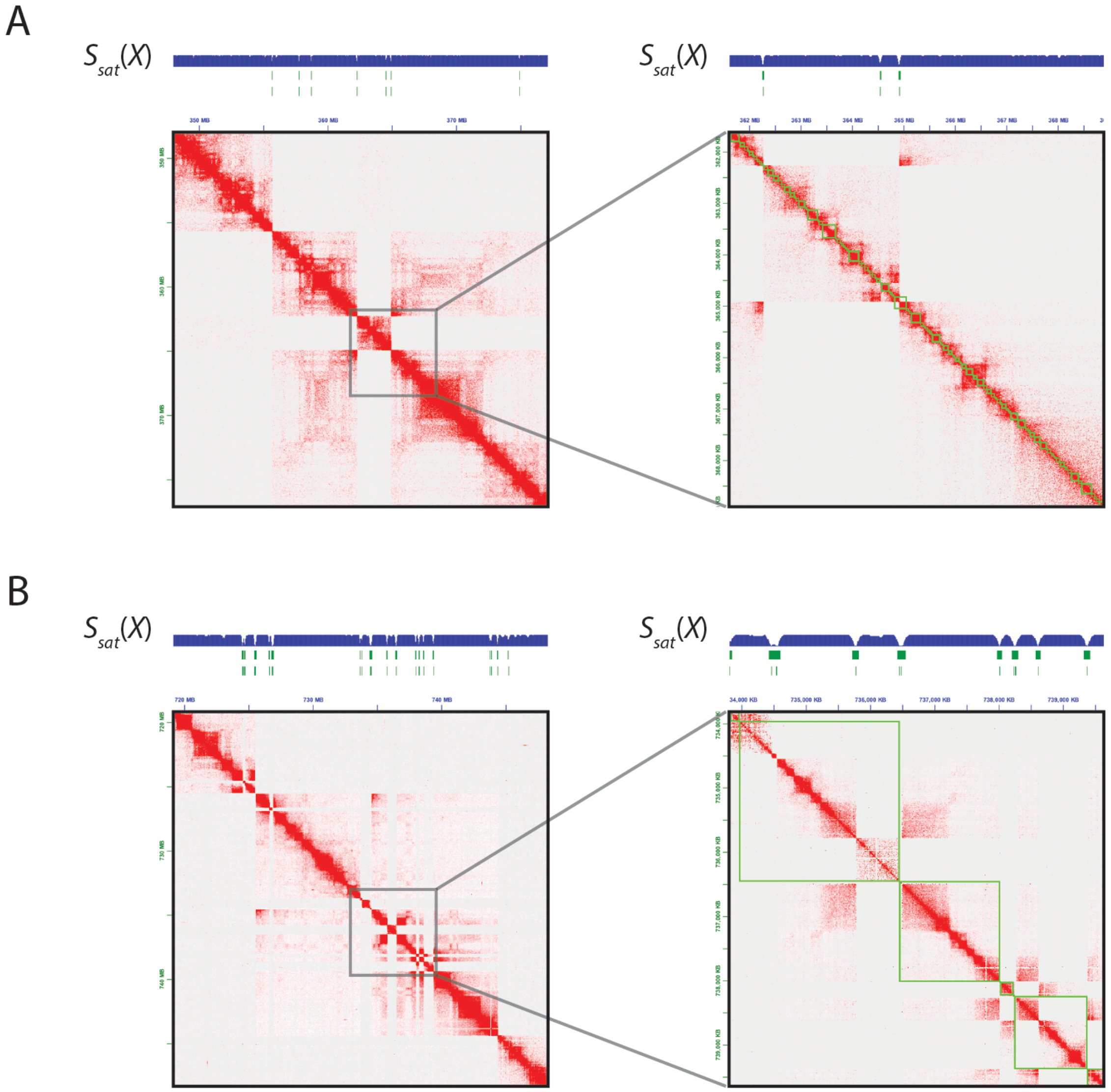

Fig. S5.

Misassembly detection algorithm performance on Hs1 (A) and AaegL2 (B) input. Left panel shows a fragment of the Hi-C map for the assembly obtained by scaffolding the original input scaffolds, without any editing. The tracks on top of the map show the distribution of $S_{sat}(X, r_1)$ along the assembly (blue) as well as coarse (top green track) and fine (bottom green track) positioning of misassembled sequences as identified by the misassembly detector. Right panel shows a zoom-in on a fragment of the map with input scaffold boundaries superimposed to assist in classifying the detected misassemblies. Intrascaffold misassemblies constitute a list of edits to be applied to the original scaffold set; misassemblies that overlap with scaffold boundaries are ignored. There is one intrascaffold misassembly in Hs1 and 5 intrascaffold misassemblies in AaegL2 in the corresponding fields of view.

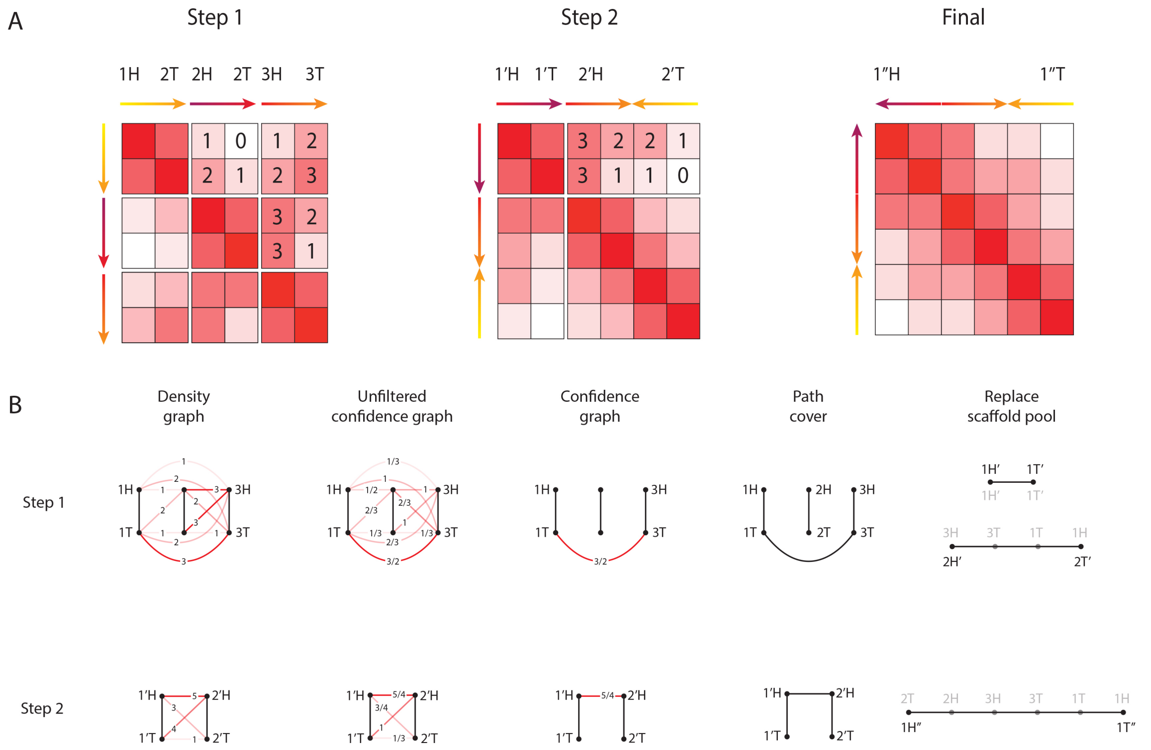

Fig. S6.

An example of applying an iterative scaffolding algorithm to a mock Hi-C dataset. The input scaffold pool consists of three scaffolds: 1, 2 and 3. The scaffolds are split into hemi-scaffolds. (To be able to distinguish between the hemi-scaffolds we annotate one as H for head and T for tail. The choice in each case is arbitrary.) The number of pairwise Hi-C contacts observed between all loci in all scaffolds is given as a Hi-C contact map. The assembly finishes in two steps. We show the intermediate results for both steps: density graph, unfiltered confidence graph, confidence graph, path cover and redefinition of scaffold pool. Note that to reduce cluttering the weights on the density graph are given without normalization. For the same reason the weights of sister edges are not shown in the density and confidence graphs; instead the sister edges are marked with black color.

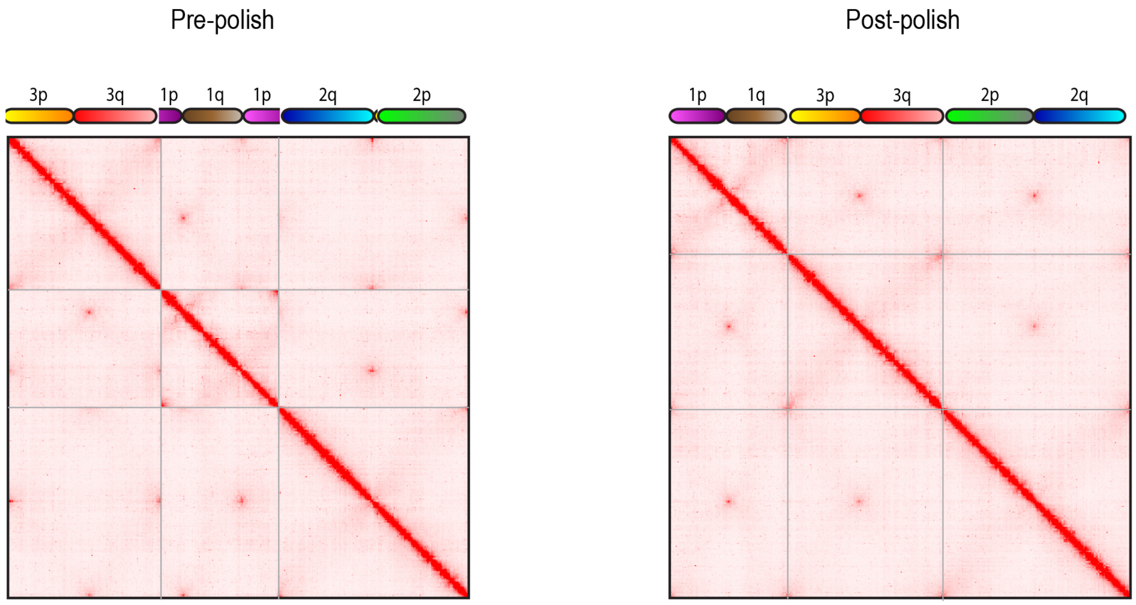

Fig. S7.

Polishing the assembly during the construction of AaegL4 genome. Clustering of telomeres and centromeres can create false positives during scaffolding, since extremely strong off-diagonal 3D signals associated with telomere and centromere clustering can sometimes be strong enough to rival the contact frequencies observed for loci that are adjacent in 1D. Such errors are corrected by low-resolution misassembly detection and reassembly of the resulting multimegabase fragments.

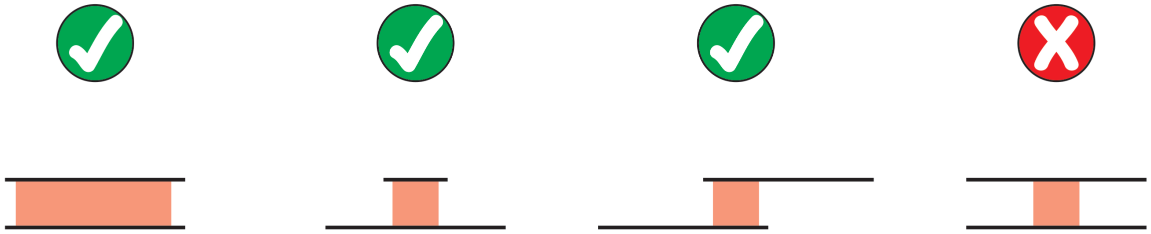

Fig. S8.

The location of the overlap relative to input scaffold boundaries is taken into account in determining whether the scaffolds can be correctly merged.

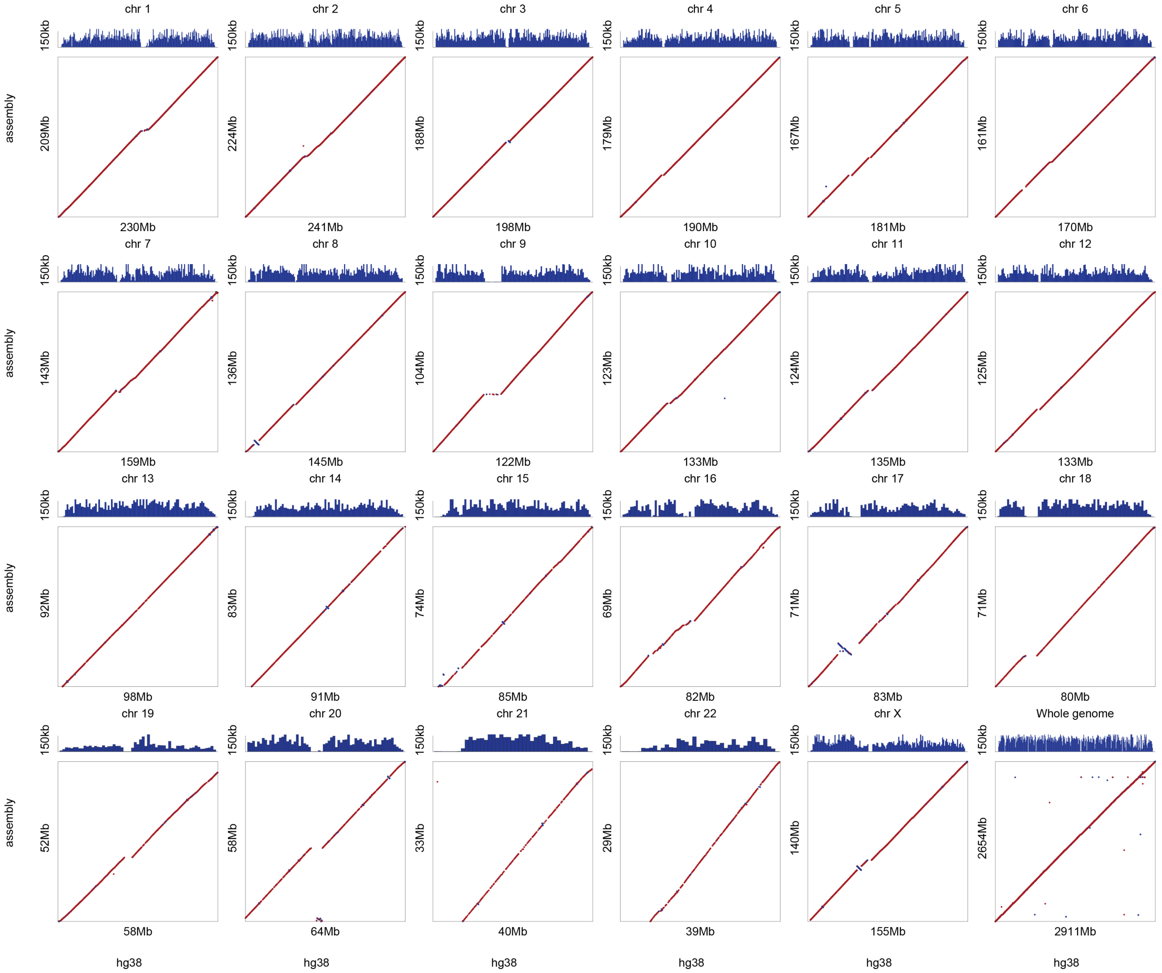

Fig. S9.

Dotplots showing alignment of Hs2-HiC chromosome-length scaffolds vs hg38 chromosome-length scaffolds. The hg38 reference (NCBI accession number GCA_000001405.23) is shown on the X axis. The Y axis shows the 23 largest scaffolds of the Hs2-HiC assembly; they have been ordered and oriented to match the chromosomes as defined in hg38 in order to facilitate comparison. (For the same reason, all gaps are removed in both assemblies.) Each dot represents the position of an individual resolved scaffold aligned to hg38. The color of the dots reflects the orientation of individual alignments with respect to hg38 (red indicates a match, whereas blue indicates disagreement). The track on top illustrates the scaffold N50 of the draft DISCOVAR *de novo* assembly Hs1 as a function of position (calculated in windows of 1Mb for individual chromosomes and 10Mb for the whole-genome graph). Alignment was performed using BWA (*34*). The dotplots illustrate excellent correspondence between hg38 and Hs2-HiC, with the exception of a few low-complexity regions of the human genome.

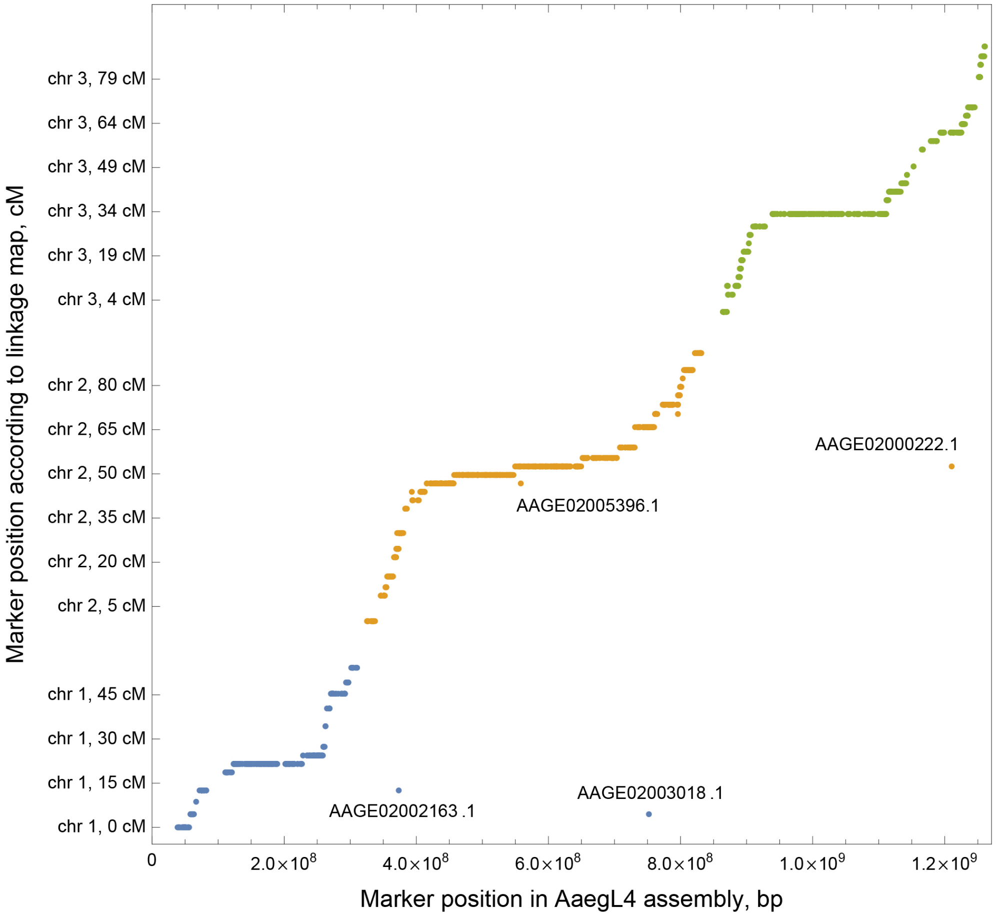

Fig. S10.

The correlation between the position of a scaffold on a genetic linkage map in centimorgans (cM) and its position in the AaegL4 assembly. Out of 1826 markers, only four are inconsistent. These inconsistencies are due to errors in the draft assembly (AaegL2) that were not flagged by our approach.

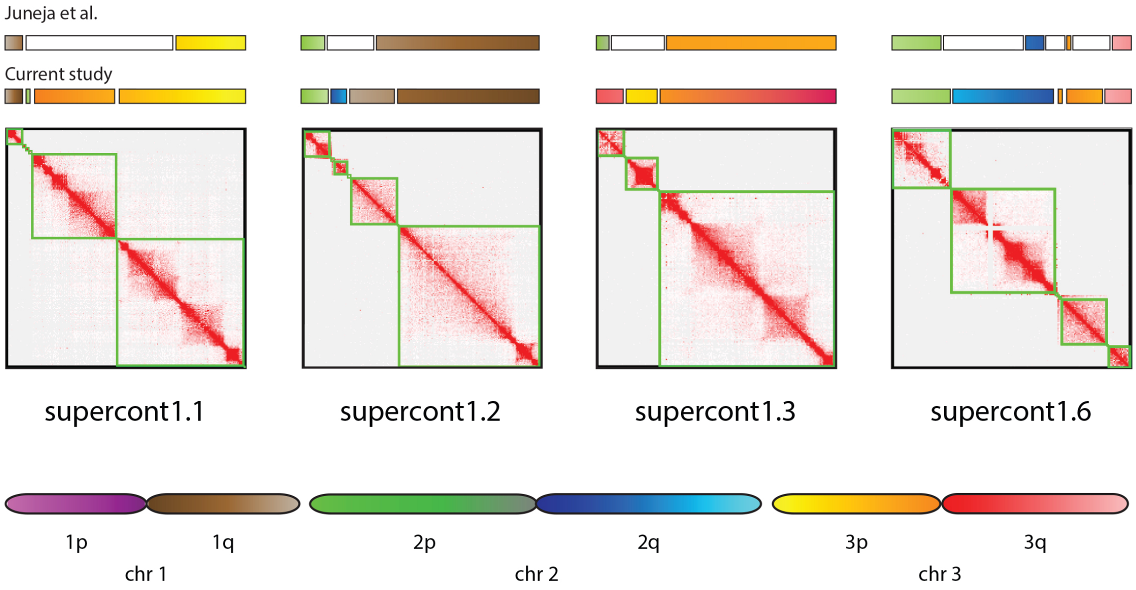

Fig. S11.

Misassembly detection using Hi-C: comparison with evidence from linkage mapping. Shown are contact maps for the first four AaegL2 scaffolds that were identified as misassembled in a genetic linkage mapping study (*19*). The boundaries of consistent fragments as identified via automatic misassembly detector are overlaid over the contact maps (green squares). The upper track shows the location of breakpoints as well as the position of the resulting scaffold fragments (indicated using color) along the *Ae. aegypti* chromosomes according to the linkage map (*19*). White bars indicate a lack of markers on the fragment, making more precise identification of breakage position on the basis of the genetic map impossible. The lower track illustrates the location of breakpoints as well as the position of the resulting scaffold fragments (indicated using color) according to current study. The overall coloring scheme used is shown; however note that the individual color gradients along each scaffold fragment were enhanced in order to heighten the contrast between nearby positions on the same chromosome. All 63 scaffolds that were identified as misassembled in (*19*) through linkage mapping were independently flagged as misassembled based on their Hi-C signal.

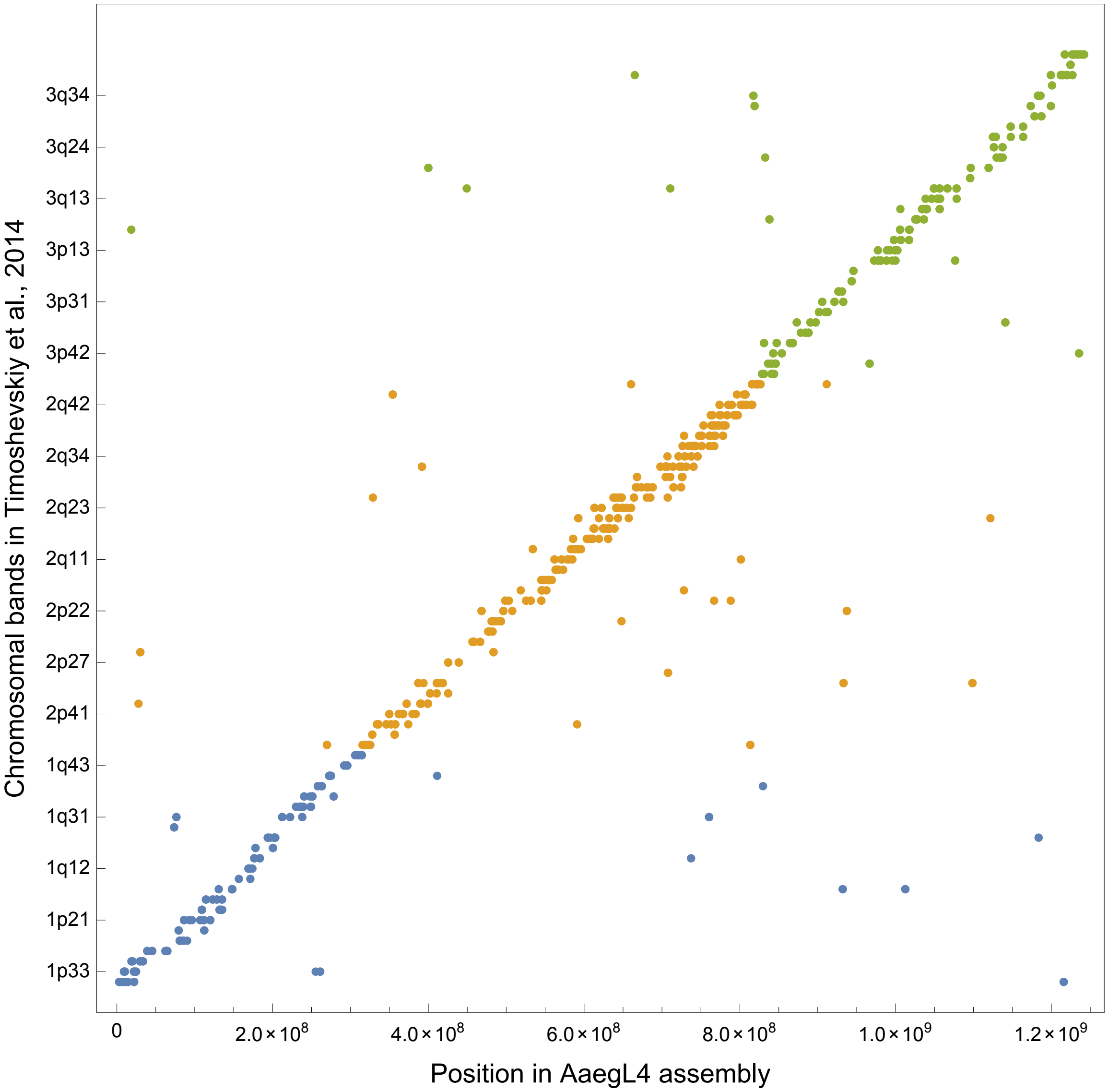

Fig. S12.

The correlation between chromosomal band assignment by physical mapping (*36*) and position in AaegL4 genome.

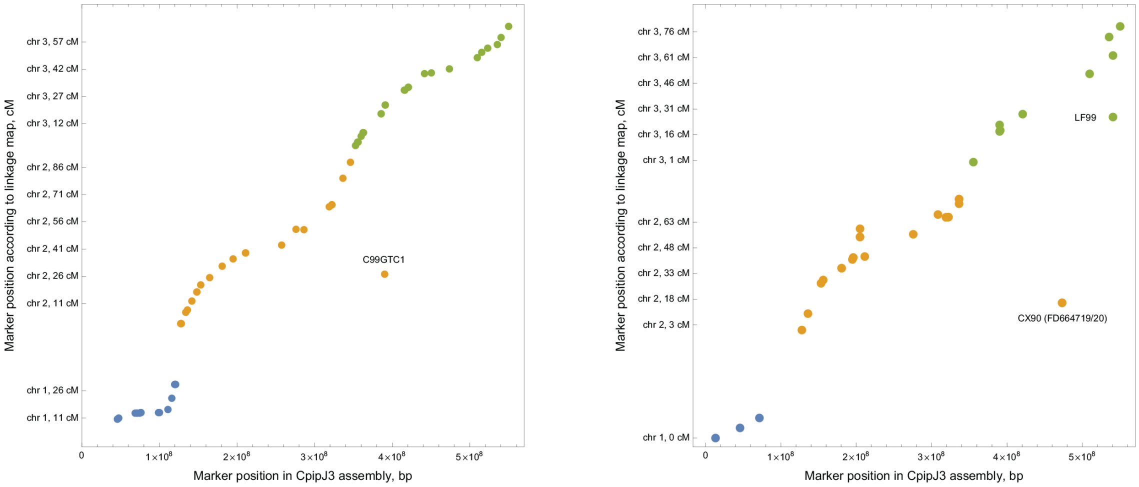

Fig. S13

The correlation between the position of a marker on a genetic linkage map in centimorgans (cM) and its position in the CpipJ3 assembly. (A) Map of microsatellite loci (*21*); (B) Map of restriction fragment length polymorphism (RFLP) markers (*20*).

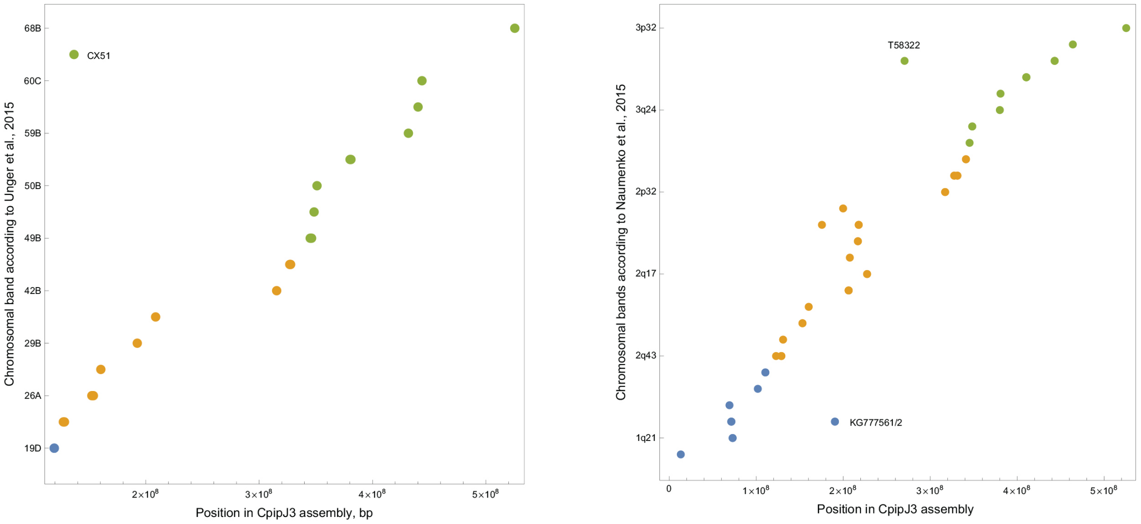

Fig. S14.

Correlation between chromosomal band assignment by physical mapping and position in CpipJ3 genome. (A) Physical mapping of polytene *Cx. quinquefasciatus* chromosomes (*37*); (B) Mitotic chromosome-based physical mapping (*38*).

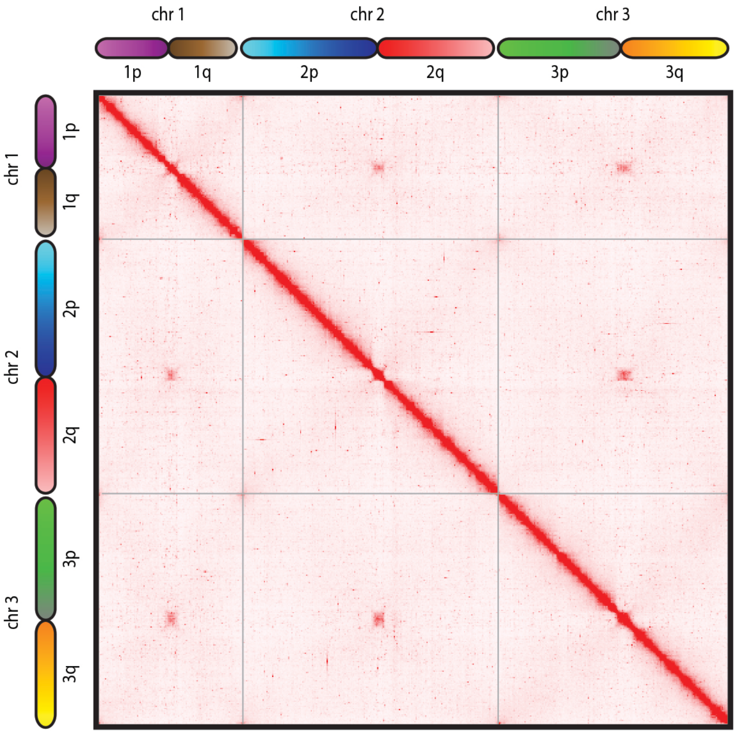

Fig. S15.

The 3D map of the *Cx. quinquefasciatus* genome. Both *Ae. aegypti* and *Cx. quinquefasciatus* genomes exhibit bright, off-diagonal peaks, which indicate the spatial clustering of telomeres and centromeres. These peaks facilitate the annotation of centromeric sequences for each chromosome (*41*).

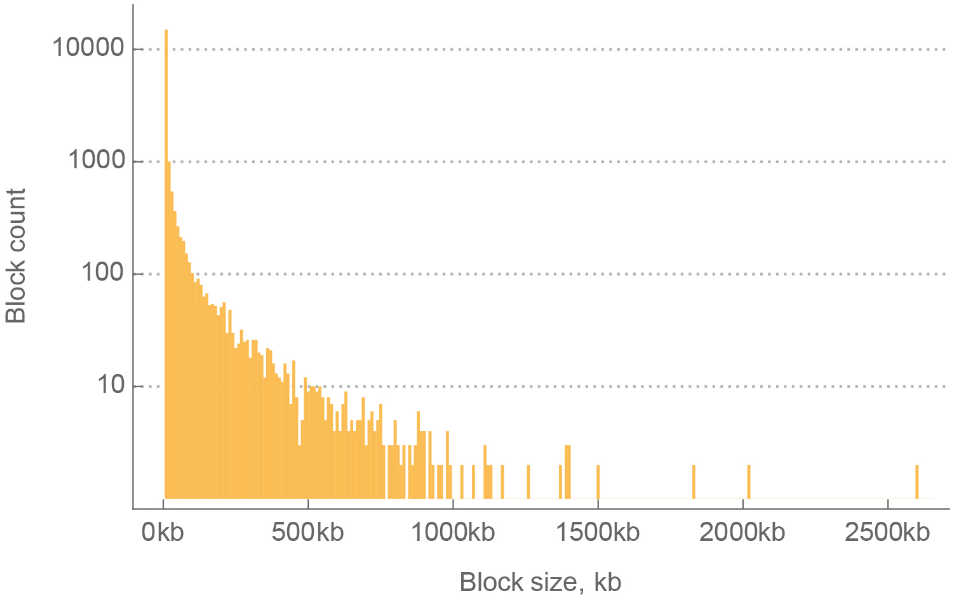

Fig. S16.

Size distribution for synteny blocks between *Ae. aegypti* and *An. gambiae*. The block sizes are measured with respect to the *Ae. aegypti* genome. The blocks are defined as chains of conserved sequence markers that are both consecutive and collinear in both genomes. The chain ends when two consecutive markers disagree with the rest of the chain; however, one marker in the wrong order and/or the wrong orientation does not break the chain.

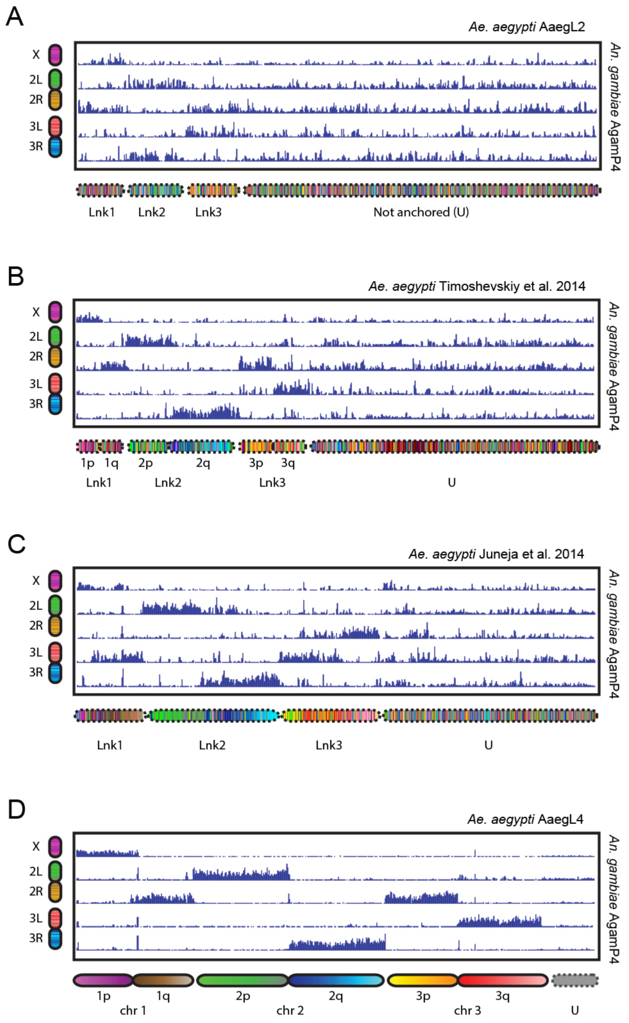

Fig. S17.

The AaegL4 genome assembly enables genome-wide analysis of conservation of synteny between *Ae. aegypti* and *An. gambiae*. (A) Density of conserved alignments between chromosomes of *Ae. aegypti* and *An. gambiae* as suggested by the AaegL2 assembly. Below the histogram tracks we show the linkage groups reported in AaegL2 (*18*). (B) The synteny analysis from panel A is repeated using improved linkage groups generated via physical mapping (*36*), which are indicated below the plot. (C) The synteny analysis from panel A is repeated using improved linkage groups generated via genetic linkage mapping (*19*), which are indicated below the plot. (D) Synteny analysis for the AaegL4 assembly. A one-to-one correspondence between the chromosome arms of *Ae. aegypti* and *An. gambiae* is apparent. Chromograms along the x- and y- axes indicate which portions of AaegL4 correspond to which positions in the other genomes and genome assemblies.

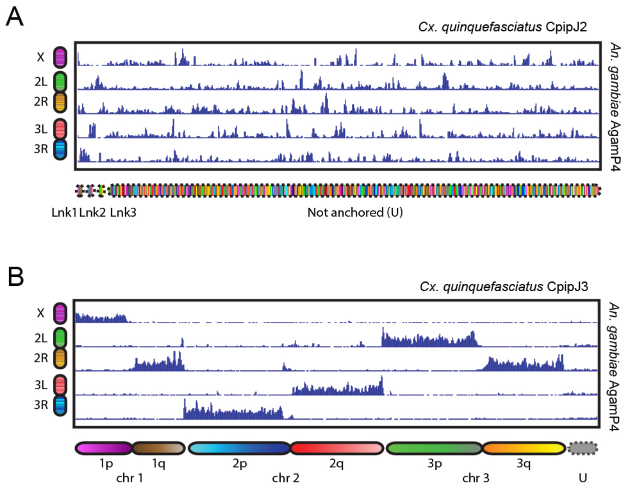

Fig. S18.

The CpipJ3 mapping reveals strong conservation of the contents of chromosome arms between *Cx. quinquefasciatus* and *An. gambiae*. A) Conservation of synteny between chromosomes of *Cx. quinquefasciatus* and *An. gambiae* as suggested by the CpipJ2 assembly. B) Conservation of synteny as represented by the CpipJ3 assembly. Chromograms along the x- and y- axes indicate which portions of CpipJ3 correspond to which positions in the other genomes and genome assemblies.

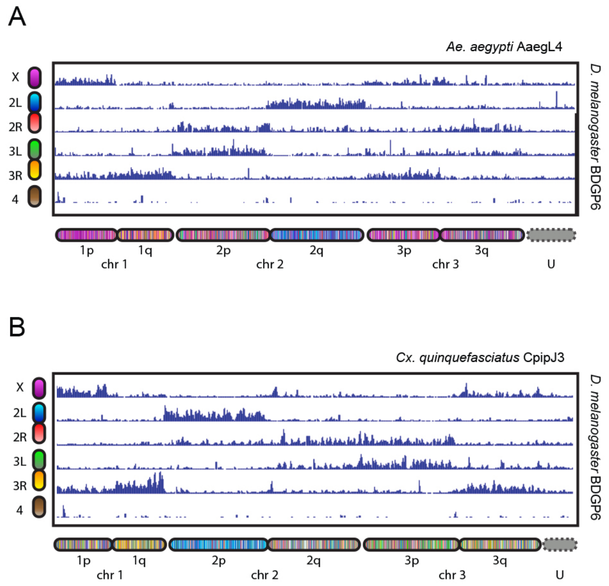

Fig. S19.

Conservation of synteny across dipterans. Several chromosome arms, such as *D. melanogaster* 2L, *Ae. aegypti* 2q, and *Cx. quinquefasciatus* 2p, show strong conservation of content. Chromograms along the x- and y- axes indicate which portions of BDGP6 correspond to which positions in the other genomes and genome assemblies.

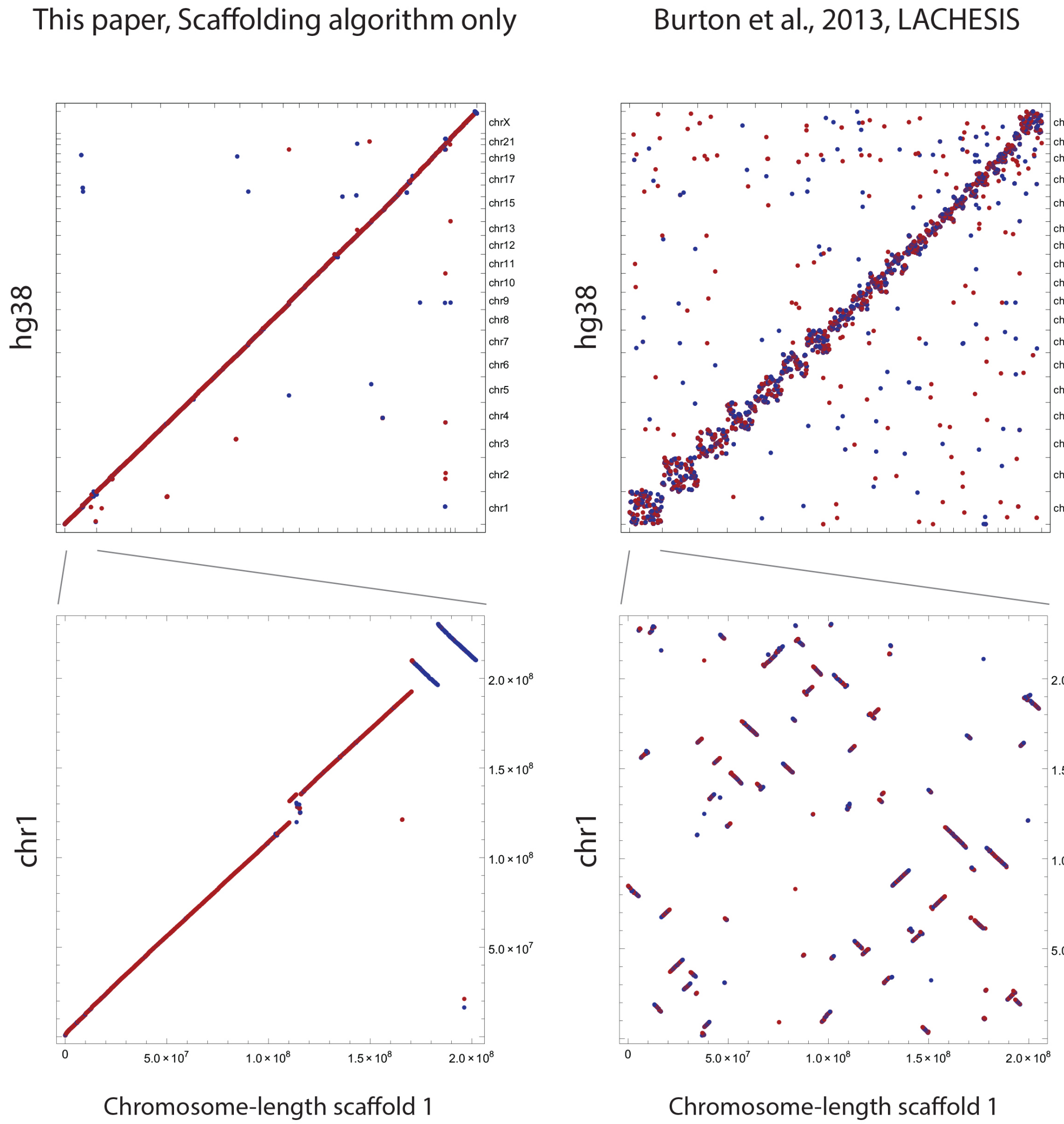

Fig. S20.

Comparison of scaffolding algorithms presented in (*10*) and the current paper. Misassembly detection has been disabled in our pipeline to focus the comparison specifically on scaffolding abilities. The algorithms have been given the same data as input: set of DISCOVAR *de novo* scaffolds from 60X PE250 Illumina short reads and 6.7X of Hi-C data. In both cases only scaffolds longer than 20kb have been used. Although clustering is clearly visible in the LACHESIS output individual chromosome-length scaffolds were not correctly reconstructed.

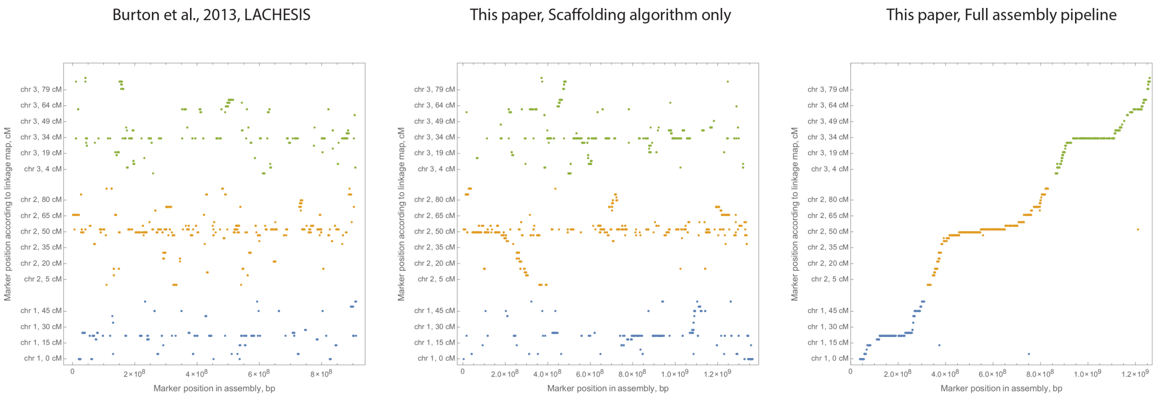

Fig. S21.

Comparison of the results of running LACHESIS (*10*), our scaffolding algorithm (without misjoin correction), and our full pipeline (including misjoin correction) on AaegL2 with the existing linkage map (*19*).

Table S1.

Chromosome-length scaffolds of the Hs2-HiC *de novo* short read assembly as compared to those in the current human genome reference, hg38. Comparison of the position of the first and last contig of Hs2-HiC to the hg38 chromosome termini demonstrates that the Hs2-HiC scaffolds are chromosome-length.

Chr	hg38 Sequenced Bases	Hs2-HiC Sequenced Bases	Hs2-HiC % of hg38	Hs2-HiC p-terminus vs hg38 p-terminus (% of chr length)	Hs2-HiC q-terminus vs hg38 q-terminus (% of chr length)
1	230,481,014	209,243,285	91%	0.3%	0.0%
2	240,548,237	223,996,452	93%	0.0%	0.0%
3	198,100,142	188,079,834	95%	0.0%	0.1%
4	189,752,667	179,103,191	94%	0.0%	0.1%
5	181,265,378	167,410,379	92%	0.0%	0.2%
6	170,078,523	160,862,810	95%	0.1%	0.1%
7	158,970,135	143,026,896	90%	0.0%	0.0%
8	144,768,136	135,766,704	94%	0.1%	0.0%
9	121,790,553	103,710,604	85%	0.1%	0.2%
10	133,262,998	122,608,161	92%	0.1%	0.1%
11	134,533,742	123,909,805	92%	0.1%	0.0%
12	133,137,819	124,764,976	94%	0.1%	0.0%
13	97,983,128	91,655,170	94%	2.8%	0.0%
14	90,568,149	83,363,733	92%	4.1%	0.2%
15	84,641,325	74,204,551	88%	3.4%	0.2%
16	81,805,944	69,176,130	85%	0.0%	0.3%
17	82,920,216	70,715,528	85%	0.4%	0.1%
18	80,089,605	70,899,228	89%	0.2%	0.0%
19	58,440,758	52,113,858	89%	0.4%	0.1%
20	63,944,257	58,053,966	91%	0.0%	0.1%
21	40,088,622	32,584,600	81%	2.3%	0.1%
22	39,159,782	28,584,318	73%	21.0%	0.4%
X	154,893,034	140,293,516	91%	0.2%	0.0%
All	2,911,224,164	2,654,127,695	91%	0.7%	0.1%

Table S2.

Statistics describing various scaffold populations. See Fig. S2, which illustrates the corresponding scaffold populations in the context of the assembly workflow.

Scaffold Type	Statistic	Hs2-HiC	AaegL4	CpipJ3
Draft	Sequenced Base Pairs	2,819,306,710	1,310,076,332	539,974,961
	# of Scaffolds	73,770	4,756	3,172
	Scaffold N50, bp	125,775	1,547,048	486,756
	Length of Longest Scaffold, bp	1,261,627	5,856,339	3,873,040
Unattempted	Sequenced Base Pairs	151,762,261	14,122,292	112,343
	% of Total Sequenced Base Pairs	5.4%	1.1%	0.02%
	# of Scaffolds	43,231	2,222	25
	Scaffold N50, bp	6,144	6,577	9,403
	Length of Longest Scaffold, bp	14,999	9,990	9,957
Input	Sequenced Base Pairs	2,667,544,449	1,295,954,040	539,862,618
	% of Total Sequenced Base Pairs	94.6%	98.9%	99.98%
	# of Scaffolds	30,539	2,534	3,147
	Scaffold N50, bp	134,113	1,557,198	486,756
	Length of Longest Scaffold, bp	1,261,627	5,856,339	3,873,040
Resolved	Sequenced Base Pairs	2,654,127,695	1,213,489,564	508,694,417
	% of Attempted Sequenced Base Pairs	99.5%	93.6%	94.2%
	% of Total Sequenced Base Pairs	94.1%	92.6%	94.2%
	# of Scaffolds	30,209	3,613	2,902
	Scaffold N50, bp	134,623	1,010,000	459,995
	Length of Longest Scaffold, bp	1,261,627	4,485,194	3,873,040
Unresolved & Inconsistent	Sequenced Base Pairs	13,416,754	82,464,476	31,168,201
	% of Attempted Sequenced Base Pairs	0.50%	6.36%	5.77%
	% of Total Sequenced Base Pairs	0.48%	6.29%	5.77%
	# of Scaffolds	832	3,987	1,227
	Scaffold N50, bp	28,016	65,155	45,079
	Length of Longest Scaffold, bp	231,347	474,197	323,163

Table S3.

Statistics describing the results of assembly using Hi-C, including only draft scaffolds that we attempted to scaffold further as input. The values describe the chromosome-length scaffolds, as well as other, smaller scaffolds generated during the Hi-C assembly process. Note that the total number of scaffolds, and their total length in base pairs, can change on account of the misjoin correction and scaffold merge steps.

Scaffold Type	Statistic	Hs2-HiC	AaegL4	CpipJ3
Input	Sequenced Base Pairs	2,667,544,449	1,295,954,040	539,862,618
	# of Scaffolds	30,539	2,534	3,147
	Scaffold N50, bp	134,113	1,557,198	486,756
	Length of Longest Scaffold, bp	1,261,627	5,856,339	3,873,040
Chr-length & Small	Sequenced Base Pairs	2,667,544,449	1,240,425,868	523,568,378
	% of Sequenced Base Pairs in Input	100%	95.7%	97.0%
	# of Scaffolds	834	3,984	1,227
	Scaffold N50, bp	141,244,516	404,248,146	190,989,159
	Length of Longest Scaffold, bp	225,222,252	471,868,560	212,641,822
Chr-length	Sequenced Base Pairs	2,654,127,695	1,157,961,392	492,400,177
	% of Sequenced Base Pairs in Input	99.5%	89.4%	91.2%
	# of Scaffolds	23	3	3
	Scaffold N50, bp	141,244,516	404,248,146	190,989,159
	Length of Longest Scaffold, bp	225,222,252	471,868,560	212,641,822
Small	Sequenced Base Pairs	13,416,754	82,464,476	31,168,201
	% of Sequenced Base Pairs in Input	0.50%	6.36%	5.77%
	# of Scaffolds	811	3,981	1,224
	Scaffold N50, bp	30,467	65,348	45,079
	Length of Longest Scaffold, bp	231,347	474,197	323,163

Table S4.

Statistics describing the results of assembly using Hi-C, including all draft scaffolds as input. The values describe the chromosome-length scaffolds, as well as other, smaller scaffolds generated during the Hi-C assembly process. Note that the total number of scaffolds, and their total length in base pairs, can change on account of the misjoin correction and scaffold merge steps.

Scaffold Type	Statistic	Hs2-HiC	AaegL4	CpipJ3
Draft	Sequenced Base Pairs	2,819,306,710	1,310,076,332	539,974,961
	# of Scaffolds	73,770	4,756	3,172
	Scaffold N50, bp	125,775	1,547,048	486,756
	Length of Longest Scaffold, bp	1,261,627	5,856,339	3,873,040
Chr-length, Small, & Tiny	Sequenced Base Pairs	2,819,306,710	1,254,548,160	523,680,721
	% of Sequenced Base Pairs in Draft	100%	95.8%	97.0%
	# of Scaffolds	44,065	6,206	1,252
	Scaffold N50, bp	141,244,516	404,248,146	190,989,159
	Length of Longest Scaffold, bp	225,222,252	471,868,560	212,641,822
Chr-length	Sequenced Base Pairs	2,654,127,695	1,157,961,392	492,400,177
	% of Sequenced Base Pairs in Draft	94.1%	88.4%	91.2%
	# of Scaffolds	23	3	3
	Scaffold N50, bp	141,244,516	404,248,146	190,989,159
	Length of Longest Scaffold, bp	225,222,252	471,868,560	212,641,822
Small & Tiny	Sequenced Base Pairs	165,179,015	96,586,768	31,280,544
	% of Sequenced Base Pairs in Draft	5.9%	7.4%	5.8%
	# of Scaffolds	44,042	6,203	1,249
	Scaffold N50, bp	6,869	57,616	44,425
	Length of Longest Scaffold, bp	231,347	474,197	323,163

Table S5.

Cumulative assembly statistics for the assemblies. The values describe the combined set of chromosome-length scaffolds, as well as small scaffolds; however, they exclude the tiny scaffolds from the draft assembly, which we did not attempt to analyze (see Fig. S2).

Statistics	Hs2-HiC	AaegL4	CpipJ3
Base Pairs	2,667,544,449	1,240,425,868	523,568,378
Number of contigs	37,466	35,001	46,660
Contig N50	108,337	85,367	28,747
Number of scaffolds	834	3,984	1,227
Scaffold N50	141,244,516	404,248,146	190,989,159
In chromosome-length scaffolds	99.5%	93.4%	94.0%

Table S6.

Cumulative assembly statistics for the assemblies. The values describe the combined set of chromosome-length scaffolds, as well as small and tiny scaffolds (see Fig. S2).

Statistics	Hs2-HiC	AaegL4	CpipJ3
Base Pairs	2,819,306,710	1,254,548,160	523,680,721
Number of contigs	80,725	37,224	46,721
Contig N50	102,793	84,074	28,735
Number of scaffolds	44,065	6,206	1,252
Scaffold N50	141,244,516	404,248,146	190,989,159
In chromosome-length scaffolds	94.1%	92.3%	94.0%

Table S7.
Chromosome-length scaffolds of AaegL4 and CpipJ3.

Assembly Name	AaegL4 Total length, bp	AaegL4 Sequenced bases, bp	CpipJ3 Total length, bp	CpipJ3 Sequenced bases, bp
Chromosome 1	307,202,349	299,394,366	119,556,434	112,137,428
Chromosome 2	471,868,560	460,653,950	212,641,822	201,346,683
Chromosome 3	404,248,146	397,913,076	190,989,159	178,916,066

Table S8.

Conservation of arm content between AgamP4 and AaegL4 based on whole-genome alignment. Each alignment block represents an alignment between one locus in a genome and a second, orthologous locus in another genome. Here, we show the number of orthologous loci associated with each AgamP4 chromosome arm, broken down by their chromosome arm on AaegL4. Some loci orthologous to AgamP4 are not assigned to chromosome-length scaffolds in AaegL4, but rather to small or tiny scaffolds; we therefore also show the total number of orthologs that are anchored to chromosome-length scaffolds. The percentages are calculated with respect to the number of anchored orthologous loci.

	AaegL4	**Total**	**Anchored**	**1p**	**1q**	**2p**	**2q**	**3p**	**3q**
AgamP4	**X**	5,296	4,831 91%	3,923 81%	276 6%	145 3%	174 4%	140 3%	173 4%
	2L	13,734	12,740 93%	454 4%	522 4%	10,220 80%	391 3%	235 2%	918 7%
	2R	16,841	16,118 96%	787 5%	5,643 35%	309 2%	644 4%	8,316 52%	419 3%
	3L	9,995	9,529 95%	376 4%	238 2%	237 2%	339 4%	245 3%	8,094 85%
	3R	13,060	12,494 96%	424 3%	268 2%	278 2%	10,749 86%	284 2%	491 4%

Table S9.

Conservation of arm content between AgamP4 and AaegL4 based on whole-genome alignment. Each alignment block represents an alignment between one locus in a genome and a second, orthologous locus in another genome. Here, we show the total length of the orthologous loci associated with each AgamP4 chromosome arm, broken down by their chromosome arm on AaegL4. Some loci orthologous to AgamP4 are not assigned to chromosome-length scaffolds in AaegL4, but rather to small or tiny scaffolds; we therefore also show the total length of orthologs that are anchored to chromosome-length scaffolds. The percentages are calculated with respect to the total length of anchored orthologous loci.

	AaegL4	Total	Anchored	1p	1q	2p	2q	3p	3q
AgamP4	X	2,089,304	1,890,087 90%	1,509,472 80%	99,495 5%	75,343 4%	70,506 4%	59,288 3%	75,983 4%
	2L	5,299,103	4,885,106 92%	177,858 4%	193,741 4%	3,948,264 81%	135,464 3%	85,130 2%	344,649 7%
	2R	6,771,249	6,467,981 96%	292,991 5%	2,231,191 34%	104,908 2%	207,432 3%	3,485,755 54%	145,704 2%
	3L	3,946,137	3,748,373 95%	125,450 3%	80,424 2%	93,035 2%	116,912 3%	71,548 2%	3,261,004 87%
	3R	5,060,186	4,855,922 96%	196,795 4%	104,498 2%	109,172 2%	4,133,038 85%	111,008 2%	201,411 4%

Table S10.

Conservation of arm content between AgamP4 and CpipJ3 based on whole-genome alignment. Each alignment block represents an alignment between one locus in a genome and a second, orthologous locus in another genome. Here, we show the number of orthologous loci associated with each AgamP4 chromosome arm, broken down by their chromosome arm on CpipJ3. All loci orthologous to AgamP4 get assigned to chromosome-length scaffolds in CpipJ3, as highlighted by the anchored alignment count. Note that the regions in the centromere-associated cluster observed in the Hi-C map of CpipJ3 are relatively large in comparison to those in AaegL4. Because loci lying in these regions cannot be assigned to a specific chromosome arm, they were excluded from the analysis in the last 6 columns.

	CpipJ3	**Total**	**Anchored**	**1p**	**1q**	**2p**	**2q**	**3p**	**3q**
AgamP4	**X**	4,696	4,526 96%	3,840 85%	77 2%	101 2%	89 2%	89 2%	83 2%
	2L	13,112	12,538 96%	318 3%	422 3%	400 3%	816 7%	9,832 78%	274 2%
	2R	16,884	16,310 97%	498 3%	5,756 35%	423 3%	441 3%	382 2%	7,814 48%
	3L	10,148	9,541 94%	352 4%	173 2%	430 5%	7,973 84%	232 2%	201 2%
	3R	12,933	12,477 96%	309 2%	160 1%	10,754 86%	289 2%	264 2%	239 2%

Table S11.

Conservation of arm content between AgamP4 and CpipJ3 based on whole-genome alignment. Each alignment block represents an alignment between one locus in a genome and a second, orthologous locus in another genome. Here, we show the total length of the orthologous loci associated with each AgamP4 chromosome arm, broken down by their chromosome arm on CpipJ3. All loci orthologous to AgamP4 get assigned to chromosome-length scaffolds in CpipJ3, as highlighted by the anchored alignment length. As in Table S10, pericentromeric regions were excluded from the analysis in the last 6 columns.

	CpipJ3	Total	Anchored	1p	1q	2p	2q	3p	3q
AgamP4	X	1,833,620	1,770,493 97%	1,553,607 88%	32,787 2%	31,127 2%	29,180 2%	25,623 1%	26,427 1%
	2L	5,030,653	4,824,352 96%	107,944 2%	147,390 3%	117,779 2%	329,880 7%	3,889,684 81%	82,639 2%
	2R	6,722,672	6,521,544 97%	160,791 2%	2,325,319 36%	123,219 2%	150,203 2%	105,720 2%	3,364,617 52%
	3L	3,959,192	3,730,270 94%	135,065 4%	48,100 1%	144,628 4%	3,227,489 87%	72,513 2%	57,471 2%
	3R	4,855,195	4,706,392 97%	116,050 2%	44,500 1%	4,163,259 88%	118,621 3%	71,312 2%	66,790 1%

Table S12.

Breakdown of the projected costs for generating a *de novo* human or mammalian genome with chromosome-length scaffolds following the Hs2-HiC strategy.

Library Preparation	$50
Reagents (DNA-Seq)	$50
Reagents (Hi-C)	$150
Sequencing (DNA-Seq)	$8600
Sequencing (Hi-C)	$600
Total Cost	$9450

Table S13.

Pseudocode for misassembly correction algorithm.

```
Misassembly correction:
1) Calculate $C(b)$ for the contact matrix at a coarse resolution $r_1$
2) Compute the saturated score function $S_{sat}(X, r_1)$ at the coarse resolution
3) Calculate $C(b)$ for the contact matrix at fine resolution $r_2$
4) Compute the saturated score function $S_{sat}(X, r_2)$
5) Flag misjoined loci satisfying $S_{sat}(X, r_1) < k * S_{sat}^{ex}$
6) For each misjoined locus identified:
    a. Localize the misjoin at resolution $r_2$ by finding the minimum of $S_{sat}(X, r_2)$ in the locus
    b. Compare localized misjoins with scaffold boundaries to distinguish scaffolds containing misjoins from misjoins that lie between scaffolds
    c. Correct the input scaffolds by excising misjoins inside scaffolds and labeling the excised fragment as inconsistent; in addition, if the misjoin is far from the ends of the scaffold, divide the input scaffold into two scaffolds by splitting it at the misjoin site
```

Table S14.

Pseudocode for iterative scaffolding algorithm.

```
Scaffolding:
  Initialize the scaffold pool using a set of input scaffolds
  While there is more than one scaffold in the scaffold pool:
    a. Construct the density graph for the scaffolds in the scaffold pool
    b. Transform the density graph into a confidence graph
    c. If the confidence graph does not contain edges linking hemi-scaffolds
       from distinct scaffolds in the pool ("non-sister edges"):
         i. Remove the smallest scaffold from the scaffold pool
    d. Else:
         i. Find maximum weight vertex-disjoint path cover of the
            confidence graph
        ii. Determine the corresponding output scaffolds
       iii. Replace the scaffold pool with the output scaffolds
```

Table S15 (provided online as a separate excel file).

AaegL4 alignments of markers from Juneja et al. *Ae. aegypti* genetic mapping study (*19*).

Table S16 (provided online as a separate excel file).

AaegL4 alignments of markers from Timoshevskiy et al. *Ae. aegypti* physical mapping study (*36*).

Table S17 (provided online as a separate excel file).

CpipJ3 alignments of microsatellite loci markers from Hickner et al. *Cx. quinquefasciatus* genetic mapping study (*21*).

Table S18 (provided online as a separate excel file).

CpipJ3 alignments of RFLP markers from Arensburger et al. *Cx. quinquefasciatus* genetic mapping study (*20*).

Table S19 (provided online as a separate excel file).

CpipJ3 alignments of markers from Unger et al. *Cx. quinquefasciatus* physical mapping study (*37*).

Table S20 (provided online as a separate excel file).

CpipJ3 alignments of markers from Naumenko et al. *Cx. quinquefasciatus* physical mapping study (*38*).

References and Notes

1. A. Harmon, "Team of Rival Scientists Comes Together to Fight Zika," *New York Times*, 30 March 2016; www.nytimes.com/2016/03/31/us/mapping-a-genetic-strategy-to-fight-the-zika-virus.html?_r=0.

2. S. Gnerre, I. Maccallum, D. Przybylski, F. J. Ribeiro, J. N. Burton, B. J. Walker, T. Sharpe, G. Hall, T. P. Shea, S. Sykes, A. M. Berlin, D. Aird, M. Costello, R. Daza, L. Williams, R. Nicol, A. Gnirke, C. Nusbaum, E. S. Lander, D. B. Jaffe, High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 1513–1518 (2011). doi:10.1073/pnas.1017351108 Medline

3. L. J. S. Williams, D. G. Tabbaa, N. Li, A. M. Berlin, T. P. Shea, I. Maccallum, M. S. Lawrence, Y. Drier, G. Getz, S. K. Young, D. B. Jaffe, C. Nusbaum, A. Gnirke, Paired-end sequencing of Fosmid libraries by Illumina. *Genome Res.* **22**, 2241–2249 (2012). doi:10.1101/gr.138925.112 Medline

4. E. Lieberman-Aiden, N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, J. Dekker, Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009). doi:10.1126/science.1181369 Medline

5. S. S. P. Rao, M. H. Huntley, N. C. Durand, E. K. Stamenova, I. D. Bochkov, J. T. Robinson, A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander, E. L. Aiden, A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014). doi:10.1016/j.cell.2014.11.021 Medline

6. N. Kaplan, J. Dekker, High-throughput genome scaffolding from in vivo DNA interaction frequency. *Nat. Biotechnol.* **31**, 1143–1147 (2013). doi:10.1038/nbt.2768 Medline

7. H. Marie-Nelly, M. Marbouty, A. Cournac, J.-F. Flot, G. Liti, D. P. Parodi, S. Syan, N. Guillén, A. Margeot, C. Zimmer, R. Koszul, High-quality genome (re)assembly using chromosomal contact data. *Nat. Commun.* **5**, 5695 (2014). doi:10.1038/ncomms6695 Medline

8. C. L. Peichel, S. T. Sullivan, I. Liachko, M. A. White, http://biorxiv.org/content/early/2016/08/09/068528 (2016).

9. N. H. Putnam, B. L. O'Connell, J. C. Stites, B. J. Rice, M. Blanchette, R. Calef, C. J. Troll, A. Fields, P. D. Hartley, C. W. Sugnet, D. Haussler, D. S. Rokhsar, R. E. Green, Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome Res.* **26**, 342–350 (2016). doi:10.1101/gr.193474.115 Medline

10. J. N. Burton, A. Adey, R. P. Patwardhan, R. Qiu, J. O. Kitzman, J. Shendure, Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nat. Biotechnol.* **31**, 1119–1125 (2013). doi:10.1038/nbt.2727 Medline

11. D. M. Bickhart *et al*., http://biorxiv.org/content/early/2016/07/18/064352 (2016).

12. A. M. Session, Y. Uno, T. Kwon, J. A. Chapman, A. Toyoda, S. Takahashi, A. Fukui, A. Hikosaka, A. Suzuki, M. Kondo, S. J. van Heeringen, I. Quigley, S. Heinz, H. Ogino, H. Ochi, U. Hellsten, J. B. Lyons, O. Simakov, N. Putnam, J. Stites, Y. Kuroki, T. Tanaka,

T. Michiue, M. Watanabe, O. Bogdanovic, R. Lister, G. Georgiou, S. S. Paranjpe, I. van Kruijsbergen, S. Shu, J. Carlson, T. Kinoshita, Y. Ohta, S. Mawaribuchi, J. Jenkins, J. Grimwood, J. Schmutz, T. Mitros, S. V. Mozaffari, Y. Suzuki, Y. Haramoto, T. S. Yamamoto, C. Takagi, R. Heald, K. Miller, C. Haudenschild, J. Kitzman, T. Nakayama, Y. Izutsu, J. Robert, J. Fortriede, K. Burns, V. Lotay, K. Karimi, Y. Yasuoka, D. S. Dichmann, M. F. Flajnik, D. W. Houston, J. Shendure, L. DuPasquier, P. D. Vize, A. M. Zorn, M. Ito, E. M. Marcotte, J. B. Wallingford, Y. Ito, M. Asashima, N. Ueno, Y. Matsuda, G. J. C. Veenstra, A. Fujiyama, R. M. Harland, M. Taira, D. S. Rokhsar, Genome evolution in the allotetraploid frog *Xenopus laevis*. *Nature* **538**, 336–343 (2016). doi:10.1038/nature19840 Medline

13. R. R. Love, N. I. Weisenfeld, D. B. Jaffe, N. J. Besansky, D. E. Neafsey, Evaluation of DISCOVAR de novo using a mosquito sample for cost-effective short-read genome assembly. *BMC Genomics* **17**, 187 (2016). doi:10.1186/s12864-016-2531-7 Medline

14. E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howland, L. Kann, J. Lehoczky, R. LeVine, P. McEwan, K. McKernan, J. Meldrim, J. P. Mesirov, C. Miranda, W. Morris, J. Naylor, C. Raymond, M. Rosetti, R. Santos, A. Sheridan, C. Sougnez, Y. Stange-Thomann, N. Stojanovic, A. Subramanian, D. Wyman, J. Rogers, J. Sulston, R. Ainscough, S. Beck, D. Bentley, J. Burton, C. Clee, N. Carter, A. Coulson, R. Deadman, P. Deloukas, A. Dunham, I. Dunham, R. Durbin, L. French, D. Grafham, S. Gregory, T. Hubbard, S. Humphray, A. Hunt, M. Jones, C. Lloyd, A. McMurray, L. Matthews, S. Mercer, S. Milne, J. C. Mullikin, A. Mungall, R. Plumb, M. Ross, R. Shownkeen, S. Sims, R. H. Waterston, R. K. Wilson, L. W. Hillier, J. D. McPherson, M. A. Marra, E. R. Mardis, L. A. Fulton, A. T. Chinwalla, K. H. Pepin, W. R. Gish, S. L. Chissoe, M. C. Wendl, K. D. Delehaunty, T. L. Miner, A. Delehaunty, J. B. Kramer, L. L. Cook, R. S. Fulton, D. L. Johnson, P. J. Minx, S. W. Clifton, T. Hawkins, E. Branscomb, P. Predki, P. Richardson, S. Wenning, T. Slezak, N. Doggett, J.-F. Cheng, A. Olsen, S. Lucas, C. Elkin, E. Uberbacher, M. Frazier, R. A. Gibbs, D. M. Muzny, S. E. Scherer, J. B. Bouck, E. J. Sodergren, K. C. Worley, C. M. Rives, J. H. Gorrell, M. L. Metzker, S. L. Naylor, R. S. Kucherlapati, D. L. Nelson, G. M. Weinstock, Y. Sakaki, A. Fujiyama, M. Hattori, T. Yada, A. Toyoda, T. Itoh, C. Kawagoe, H. Watanabe, Y. Totoki, T. Taylor, J. Weissenbach, R. Heilig, W. Saurin, F. Artiguenave, P. Brottier, T. Bruls, E. Pelletier, C. Robert, P. Wincker, D. R. Smith, L. Doucette-Stamm, M. Rubenfield, K. Weinstock, H. M. Lee, J. Dubois, A. Rosenthal, M. Platzer, G. Nyakatura, S. Taudien, A. Rump, H. Yang, J. Yu, J. Wang, G. Huang, J. Gu, L. Hood, L. Rowen, A. Madan, S. Qin, R. W. Davis, N. A. Federspiel, A. P. Abola, M. J. Proctor, R. M. Myers, J. Schmutz, M. Dickson, J. Grimwood, D. R. Cox, M. V. Olson, R. Kaul, C. Raymond, N. Shimizu, K. Kawasaki, S. Minoshima, G. A. Evans, M. Athanasiou, R. Schultz, B. A. Roe, F. Chen, H. Pan, J. Ramser, H. Lehrach, R. Reinhardt, W. R. McCombie, M. de la Bastide, N. Dedhia, H. Blöcker, K. Hornischer, G. Nordsiek, R. Agarwala, L. Aravind, J. A. Bailey, A. Bateman, S. Batzoglou, E. Birney, P. Bork, D. G. Brown, C. B. Burge, L. Cerutti, H. C. Chen, D. Church, M. Clamp, R. R. Copley, T. Doerks, S. R. Eddy, E. E. Eichler, T. S. Furey, J. Galagan, J. G. Gilbert, C. Harmon, Y. Hayashizaki, D. Haussler, H. Hermjakob, K. Hokamp, W. Jang, L. S. Johnson, T. A. Jones, S. Kasif, A. Kaspryzk, S. Kennedy, W. J. Kent, P. Kitts, E. V. Koonin, I. Korf, D. Kulp, D. Lancet, T. M. Lowe, A. McLysaght, T. Mikkelsen, J. V. Moran, N. Mulder, V. J. Pollara, C. P. Ponting, G.

Schuler, J. Schultz, G. Slater, A. F. Smit, E. Stupka, J. Szustakowki, D. Thierry-Mieg, J. Thierry-Mieg, L. Wagner, J. Wallis, R. Wheeler, A. Williams, Y. I. Wolf, K. H. Wolfe, S. P. Yang, R. F. Yeh, F. Collins, M. S. Guyer, J. Peterson, A. Felsenfeld, K. A. Wetterstrand, A. Patrinos, M. J. Morgan, P. de Jong, J. J. Catanese, K. Osoegawa, H. Shizuya, S. Choi, Y. J. Chen, J. Szustakowki, International Human Genome Sequencing Consortium, Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001). doi:10.1038/35057062 Medline

15. J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. Gabor Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. Di Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R. R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Wang, A. Wang, X. Wang, J. Wang, M. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y. H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigó, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y. H. Chiang, M. Coyne, C. Dahlke, A. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, X. Zhu, The sequence of the human genome. *Science* **291**, 1304–1351 (2001). doi:10.1126/science.1058040 Medline

16. M. Pendleton, R. Sebra, A. W. C. Pang, A. Ummat, O. Franzen, T. Rausch, A. M. Stütz, W. Stedman, T. Anantharaman, A. Hastie, H. Dai, M. H.-Y. Fritz, H. Cao, A. Cohain, G. Deikus, R. E. Durrett, S. C. Blanchard, R. Altman, C.-S. Chin, Y. Guo, E. E. Paxinos, J. O. Korbel, R. B. Darnell, W. R. McCombie, P.-Y. Kwok, C. E. Mason, E. E. Schadt, A. Bashir, Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nat. Methods* **12**, 780–786 (2015). doi:10.1038/nmeth.3454 Medline

17. D. B. Jaffe, J. Butler, S. Gnerre, E. Mauceli, K. Lindblad-Toh, J. P. Mesirov, M. C. Zody, E. S. Lander, Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome Res.* **13**, 91–96 (2003). doi:10.1101/gr.828403 Medline

18. V. Nene, J. R. Wortman, D. Lawson, B. Haas, C. Kodira, Z. J. Tu, B. Loftus, Z. Xi, K. Megy, M. Grabherr, Q. Ren, E. M. Zdobnov, N. F. Lobo, K. S. Campbell, S. E. Brown, M. F. Bonaldo, J. Zhu, S. P. Sinkins, D. G. Hogenkamp, P. Amedeo, P. Arensburger, P. W. Atkinson, S. Bidwell, J. Biedler, E. Birney, R. V. Bruggner, J. Costas, M. R. Coy, J. Crabtree, M. Crawford, B. Debruyn, D. Decaprio, K. Eiglmeier, E. Eisenstadt, H. El-Dorry, W. M. Gelbart, S. L. Gomes, M. Hammond, L. I. Hannick, J. R. Hogan, M. H. Holmes, D. Jaffe, J. S. Johnston, R. C. Kennedy, H. Koo, S. Kravitz, E. V. Kriventseva, D. Kulp, K. Labutti, E. Lee, S. Li, D. D. Lovin, C. Mao, E. Mauceli, C. F. M. Menck, J. R. Miller, P. Montgomery, A. Mori, A. L. Nascimento, H. F. Naveira, C. Nusbaum, S. O'leary, J. Orvis, M. Pertea, H. Quesneville, K. R. Reidenbach, Y.-H. Rogers, C. W. Roth, J. R. Schneider, M. Schatz, M. Shumway, M. Stanke, E. O. Stinson, J. M. C. Tubio, J. P. Vanzee, S. Verjovski-Almeida, D. Werner, O. White, S. Wyder, Q. Zeng, Q. Zhao, Y. Zhao, C. A. Hill, A. S. Raikhel, M. B. Soares, D. L. Knudson, N. H. Lee, J. Galagan, S. L. Salzberg, I. T. Paulsen, G. Dimopoulos, F. H. Collins, B. Birren, C. M. Fraser-Liggett, D. W. Severson, Genome sequence of *Aedes aegypti*, a major arbovirus vector. *Science* **316**, 1718–1723 (2007). doi:10.1126/science.1138878 Medline

19. P. Juneja, J. Osei-Poku, Y. S. Ho, C. V. Ariani, W. J. Palmer, A. Pain, F. M. Jiggins, Assembly of the genome of the disease vector *Aedes aegypti* onto a genetic linkage map allows mapping of genes affecting disease transmission. *PLOS Negl. Trop. Dis.* **8**, e2652 (2014). doi:10.1371/journal.pntd.0002652 Medline

20. P. Arensburger, K. Megy, R. M. Waterhouse, J. Abrudan, P. Amedeo, B. Antelo, L. Bartholomay, S. Bidwell, E. Caler, F. Camara, C. L. Campbell, K. S. Campbell, C. Casola, M. T. Castro, I. Chandramouliswaran, S. B. Chapman, S. Christley, J. Costas, E. Eisenstadt, C. Feschotte, C. Fraser-Liggett, R. Guigo, B. Haas, M. Hammond, B. S. Hansson, J. Hemingway, S. R. Hill, C. Howarth, R. Ignell, R. C. Kennedy, C. D. Kodira, N. F. Lobo, C. Mao, G. Mayhew, K. Michel, A. Mori, N. Liu, H. Naveira, V. Nene, N. Nguyen, M. D. Pearson, E. J. Pritham, D. Puiu, Y. Qi, H. Ranson, J. M. C. Ribeiro, H. M. Roberston, D. W. Severson, M. Shumway, M. Stanke, R. L. Strausberg, C. Sun, G. Sutton, Z. J. Tu, J. M. C. Tubio, M. F. Unger, D. L. Vanlandingham, A. J. Vilella, O. White, J. R. White, C. S. Wondji, J. Wortman, E. M. Zdobnov, B. Birren, B. M. Christensen, F. H. Collins, A. Cornel, G. Dimopoulos, L. I. Hannick, S. Higgs, G. C. Lanzaro, D. Lawson, N. H. Lee, M. A. T. Muskavitch, A. S. Raikhel, P. W. Atkinson, Sequencing of *Culex quinquefasciatus* establishes a platform for mosquito comparative genomics. *Science* **330**, 86–88 (2010). doi:10.1126/science.1191864 Medline

21. P. V. Hickner, A. Mori, D. D. Chadee, D. W. Severson, Composite linkage map and enhanced genome map for *Culex pipiens* complex mosquitoes. *J. Hered.* **104**, 649–655 (2013). doi:10.1093/jhered/est040 Medline

22. N. C. Durand, M. S. Shamim, I. Machol, S. S. P. Rao, M. H. Huntley, E. S. Lander, E. L. Aiden, Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. *Cell Syst.* **3**, 95–98 (2016). doi:10.1016/j.cels.2016.07.002 Medline

23. N. C. Durand, J. T. Robinson, M. S. Shamim, I. Machol, J. P. Mesirov, E. S. Lander, E. L. Aiden, Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell Syst.* **3**, 99–101 (2016). doi:10.1016/j.cels.2015.07.012 Medline

24. M. R. Hübner, D. L. Spector, Chromatin dynamics. *Annu. Rev. Biophys.* **39**, 471–489 (2010). doi:10.1146/annurev.biophys.093008.131348 Medline

25. M. A. Ferguson-Smith, V. Trifonov, Mammalian karyotype evolution. *Nat. Rev. Genet.* **8**, 950–962 (2007). doi:10.1038/nrg2199 Medline

26. A. L. Sanborn, S. S. P. Rao, S.-C. Huang, N. C. Durand, M. H. Huntley, A. I. Jewett, I. D. Bochkov, D. Chinnappan, A. Cutkosky, J. Li, K. P. Geeting, A. Gnirke, A. Melnikov, D. McKenna, E. K. Stamenova, E. S. Lander, E. L. Aiden, Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proc. Natl. Acad. Sci. U.S.A.* **112**, E6456–E6465 (2015). doi:10.1073/pnas.1518552112 Medline

27. O. Tange, GNU Parallel: The command-line power tool. *;login:* **36**, 42–47 (2011).

28. R. S. Harris, thesis, The Pennsylvania State University (2007).

29. J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Am. Math. Soc.* **7**, 48–50 (1956). doi:10.1090/S0002-9939-1956-0078686-7

30. J. P. Vinson, D. B. Jaffe, K. O'Neill, E. K. Karlsson, N. Stange-Thomann, S. Anderson, J. P. Mesirov, N. Satoh, Y. Satou, C. Nusbaum, B. Birren, J. E. Galagan, E. S. Lander, Assembly of polymorphic genomes: Algorithms and application to *Ciona savignyi*. *Genome Res.* **15**, 1127–1135 (2005). doi:10.1101/gr.3722605 Medline

31. C.-S. Chin, P. Peluso, F. J. Sedlazeck, M. Nattestad, G. T. Concepcion, A. Clum, C. Dunn, R. O'Malley, R. Figueroa-Balderas, A. Morales-Cruz, G. R. Cramer, M. Delledonne, C. Luo, J. R. Ecker, D. Cantu, D. R. Rank, M. C. Schatz, Phased diploid genome assembly with single-molecule real-time sequencing. *Nat. Methods* **13**, 1050–1054 (2016). doi:10.1038/nmeth.4035 Medline

32. E. M. Darrow, M. H. Huntley, O. Dudchenko, E. K. Stamenova, N. C. Durand, Z. Sun, S.-C. Huang, A. L. Sanborn, I. Machol, M. Shamim, A. P. Seberg, E. S. Lander, B. P. Chadwick, E. L. Aiden, Deletion of *DXZ4* on the human inactive X chromosome alters higher-order genome architecture. *Proc. Natl. Acad. Sci. U.S.A.* **113**, E4504–E4512 (2016). doi:10.1073/pnas.1609643113 Medline

33. N. I. Weisenfeld, S. Yin, T. Sharpe, B. Lau, R. Hegarty, L. Holmes, B. Sogoloff, D. Tabbaa, L. Williams, C. Russ, C. Nusbaum, E. S. Lander, I. MacCallum, D. B. Jaffe, Comprehensive variation discovery in single human genomes. *Nat. Genet.* **46**, 1350–1355 (2014). doi:10.1038/ng.3121 Medline

34. H. Li, R. Durbin, Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009). doi:10.1093/bioinformatics/btp324 Medline

35. H. Cao, A. R. Hastie, D. Cao, E. T. Lam, Y. Sun, H. Huang, X. Liu, L. Lin, W. Andrews, S. Chan, S. Huang, X. Tong, M. Requa, T. Anantharaman, A. Krogh, H. Yang, H. Cao, X. Xu, Rapid detection of structural variation in a human genome using nanochannel-based genome mapping technology. *Gigascience* **3**, 34 (2014). doi:10.1186/2047-217X-3-34 Medline

36. V. A. Timoshevskiy, N. A. Kinney, B. S. deBruyn, C. Mao, Z. Tu, D. W. Severson, I. V. Sharakhov, M. V. Sharakhova, Genomic composition and evolution of *Aedes aegypti* chromosomes revealed by the analysis of physically mapped supercontigs. *BMC Biol.* **12**, 27 (2014). doi:10.1186/1741-7007-12-27 Medline

37. M. F. Unger, M. V. Sharakhova, A. J. Harshbarger, P. Glass, F. H. Collins, A standard cytogenetic map of *Culex quinquefasciatus* polytene chromosomes in application for fine-scale physical mapping. *Parasit. Vectors* **8**, 307 (2015). doi:10.1186/s13071-015-0912-4 Medline

38. A. N. Naumenko, V. A. Timoshevskiy, N. A. Kinney, A. A. Kokhanenko, B. S. deBruyn, D. D. Lovin, V. N. Stegniy, D. W. Severson, I. V. Sharakhov, M. V. Sharakhova, Correction: Mitotic-chromosome-based physical mapping of the *Culex quinquefasciatus* genome. *PLOS ONE* **10**, e0127565 (2015). doi:10.1371/journal.pone.0127565 Medline

39. G. I. Giraldo-Calderón, S. J. Emrich, R. M. MacCallum, G. Maslen, E. Dialynas, P. Topalis, N. Ho, S. Gesing, G. Madey, F. H. Collins, D. Lawson, VectorBase Consortium, VectorBase: An updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. *Nucleic Acids Res.* **43**, D707–D713 (2015). doi:10.1093/nar/gku1117 Medline

40. A. Yates, W. Akanni, M. R. Amode, D. Barrell, K. Billis, D. Carvalho-Silva, C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. Girón, L. Gordon, T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan, I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag, M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat, D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa, E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J. Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino, P. Flicek, Ensembl 2016. *Nucleic Acids Res.* **44**, D710–D716 (2016). doi:10.1093/nar/gkv1157 Medline

41. N. Varoquaux, I. Liachko, F. Ay, J. N. Burton, J. Shendure, M. J. Dunham, J.-P. Vert, W. S. Noble, Accurate identification of centromere locations in yeast genomes using Hi-C. *Nucleic Acids Res.* **43**, 5331–5339 (2015). doi:10.1093/nar/gkv424 Medline